Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

THE NUMERICAL LEAST SQUARES SOLUTION OF LARGE LINEAR SYSTEMS–A COMPROMISE BETWEEN SPEED AND ACCURACY*

PHIL HOWLETT† AND ANATOLI TOROKHTI‡

Abstract. We propose and justify a numerical method called the Householder Unitary Transformation Scheme (HUTS) for the least squares solution of large linear systems in the form $YCC^* = AC^*$, where $A \in \mathbb{C}^{m\times n}$ with $\text{rank}(A) = h$ and $C \in \mathbb{C}^{h\times n}$, where $m, n \in \mathbb{N}$ are large and $n = h + s$, and where $h = \sum_{j=1}^{q} h_j$ with $h_j \in \mathbb{N}$ and $s \in \mathbb{N} - 1$. Direct solution of the least squares equation requires finding the Moore–Penrose inverse of a large $h \times h$ matrix. The recently proposed Elementary Block Operations Scheme (EBOS) [20] calculates the Moore–Penrose inverses of a collection of q matrices of size $h_j \times h_j$ and then uses these matrices to construct the Moore–Penrose inverses of an associated collection of $h \times h_j$ matrices. The new HUTS calculates the Moore–Penrose inverses of an alternative collection of q matrices of size $h_j \times h_j$ and then uses a process of back substitution to find the desired solution. In each case, the total computation time is substantially less than the direct method. In general, the HUTS is faster, but the EBOS is more accurate. We will compare the performance of all three methods in a range of typical examples.

Key word. Matrix approximation, Block matrix operations, Moore–Penrose inverse.

AMS subject classification. 15A18.

1. Introduction.

Our purpose in this paper is twofold. On the one hand, we describe a new numerical scheme—the Householder Unitary Transformation Scheme (HUTS)—to find the least squares solution for a large system of linear equations, and on the other hand, we compare the new scheme to another recently proposed scheme—the Elementary Block Operations Scheme (EBOS)—and to the direct MATLAB calculation for the same system. We use randomly generated matrices of various sizes to test each scheme.

1.1. Motivation.

A recent paper by Howlett and Torokhti [20] proposed the EBOS as an efficient numerical method for the least squares solution of large linear systems in the form $YCC^* = AC^*$, where $A \in \mathbb{C}^{m\times n}$ with $\text{rank}(A) = h$ and $C \in \mathbb{C}^{h\times n}$, where $m, n \in \mathbb{N}$ are large and $n = h + s$, where $h = \sum_{j=1}^{q} h_j$ with $h_j \in \mathbb{N}$ and $s \in \mathbb{N} - 1$. The EBOS uses a sequence of elementary block row operations on the matrix $C \to FC$ to reduce the original system to a block diagonal system $ZD_C = A(FC)^*$, where $Y = FZ$ and $D_C = (FC)(FC)^*$ is block diagonal. When $m = n$, $h_j = dh$ for each $j = 1, 2, \ldots, q$ and $n = q \cdot dh$, it was shown that EBOS is significantly faster than the direct method using the standard MATLAB routine to calculate $Y = AC^\dagger$, where $C^\dagger$ is the Moore–Penrose inverse of C.

In this paper, we describe an alternative scheme which we call the HUTS for solution of the system $YCC^* = AC^*$. The HUTS uses right multiplication by a sequence of augmented block Householder matrices to transform the matrix C into a matrix

*Received by the editors on April 10, 2025. Accepted for publication on August 18, 2025. Handling Editor: Nicolas Gillis. Corresponding Author: Anatoli Torokhti.

†Industrial AI Research Centre, UniSA STEM, University of South Australia, Mawson Lakes 5095, Australia (phil.howlett@unisa.edu.au).

‡Retired Academic, UniSA STEM, University of South Australia, Mawson Lakes 5095, Australia (anatoli.torokhti@gmail.com).

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

$$CK^{(1)}\cdots K^{(q-1)} = CQ^* = [L_1 \mid 0],$$

where the unitary matrix $K^{(1)}\cdots K^{(q-1)} = Q^* \in \mathbb{C}^{n\times n}$ is chosen in such a way that $L_1 \in \mathbb{C}^{h\times h}$ is block lower triangular[1]. We will also write $Q^* = [Q_1^* \mid Q_2^*]$, where $Q_1^* \in \mathbb{C}^{n\times h}$ and $Q_2^* \in \mathbb{C}^{n\times s}$. This transformation reduces the original system $YCC^* = AC^*$ to a simpler system $YL_1L_1^* = AQ_1^*L_1^*$. If $\mathrm{rank}(C) = h$, then L_1 is invertible and $YL_1 = AQ_1^*$. The simplified system can now be solved by inverting the block diagonal elements of L_1 and using a scheme of back substitution.

We test each scheme using randomly generated matrices A and C to compare the effectiveness of EBOS and HUTS algorithms against the direct method and each other. In our numerical trials, we assume that $m, n \in \mathbb{N}$ with $m \geq n$, $h_j = dh \in \mathbb{N}$ for all $j = 1, \ldots, q$, where $q \in \mathbb{N}$ and $n = q\cdot dh + s$ for some $s \in \mathbb{N} - 1$.

1.2. Contribution. In the first instance we propose and justify the HUTS—a new scheme designed to provide an efficient numerical scheme for solution of large linear least squares systems. In the second instance we compare the new scheme with an existing scheme—the EBOS—proposed recently in [20] and designed for the same general purpose.

An outline of the new scheme. The HUTS extends a transposed form of a basic scheme proposed and described by Rotella and Zambettakis [28, Theorem 2.2, pp 30–31]. If $A \in \mathbb{C}^{m\times n}$ with $\mathrm{rank}(A) = h$ and $C \in \mathbb{C}^{h\times n}$ and $n = h + s$, where $h = \sum_{j=1}^{q} h_j$ there are $q-1$ steps in the HUTS algorithm. The HUTS is constructed using a sequence of unitary transformations, where the transformation at Step 1 is defined using right multiplication by a block Householder matrix $K^{(1)} = H_1 \in \mathbb{C}^{(h+s)\times(h+s)}$. At Step r the essence of the HUTS transformation is right multiplication by a block Householder matrix $H_r \in \mathbb{C}^{(h_r+\cdots+h_q+s)\times(h_r+\cdots+h_q+s)}$ that transforms an initial submatrix

$$R^{(r)} = \left[\begin{array}{c|c} R_{1,1}^{(r)} & R_{1,2}^{(r)} \\ \hline R_{2,1}^{(r)} & R_{2,2}^{(r)} \end{array}\right] \in \mathbb{C}^{(h_r+\cdots+h_q)\times(h_r+\cdots+h_q+s)},$$

where $R_{1,1}^{(r)} \in \mathbb{C}^{h_r\times h_r}$ nonsingular to a modified submatrix in the form

$$R^{(r)}H_r = \left[\begin{array}{c|c} S_{1,1}^{(r)} & 0 \\ \hline S_{r,1}^{(r)} & S_{2,2}^{(r)} \end{array}\right] \in \mathbb{C}^{(h_r+\cdots+h_q)\times(h_r+\cdots+h_q+s)},$$

where $S_{1,1}^{(r)} \in \mathbb{C}^{h_r\times h_r}$ is nonsingular. At Step $r+1$, we define a smaller submatrix $R^{(r+1)} = S_{2,2}^{(r)} \in \mathbb{C}^{(h_{r+1}+\cdots+h_q)\times(h_{r+1}+\cdots+h_q+s)}$ and repeat the above procedure. In order to describe these transformations on the original matrix $C \in \mathbb{C}^{h\times h+s}$, we observe that the transformation at Step r can be defined using right multiplication of C by an augmented block Householder matrix

$$K^{(r)} = \begin{bmatrix} I_{h_1+\cdots+h_r} & 0 \\ 0 & H_r \end{bmatrix} \in \mathbb{C}^{n\times n}, \tag{1}$$

for each $r = 2, \ldots, q-1$. After $q-1$ steps we obtain a reduced matrix

$$CK^{(1)}K^{(2)}\cdots K^{(q-1)} = \left[\begin{array}{c|c} L_1 & 0 \end{array}\right] = L \in \mathbb{C}^{h\times n},$$

[1]This procedure is essentially equivalent to a standard LQ decomposition.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

where $L_1 \in \mathbb{C}^{h\times h}$ is block lower triangular with $L_{i,j} \in \mathbb{C}^{h_i \times h_j}$ for each $i, j = 1, \ldots, q$ and with $L_{i,j} = 0$ when $i < j$. If we define the unitary matrix $Q^* = K^{(1)}K^{(2)} \cdots K^{(q-1)} \in \mathbb{C}^{n\times n}$ and write $Q^* = [Q_1^* \mid Q_2^*]$, where $Q_1^* \in \mathbb{C}^{n\times h}$ and $Q_2^* \in \mathbb{C}^{n\times s}$, then the equations

$$QQ^* = I_n \iff \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right] \left[\begin{array}{c|c} Q_1^* & Q_2^* \end{array}\right] = \left[\begin{array}{c|c} Q_1Q_1^* & Q_1Q_2^* \\ \hline Q_2Q_1^* & Q_2Q_2^* \end{array}\right] = \left[\begin{array}{c|c} I_h & 0 \\ \hline 0 & I_s \end{array}\right],$$

and

$$C \left[\begin{array}{c|c} Q_1^* & Q_2^* \end{array}\right] = \left[\begin{array}{c|c} L_1 & 0 \end{array}\right] \iff CQ^* = L \iff C = LQ = \left[\begin{array}{c|c} L_1 & 0 \end{array}\right] \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right] = L_1Q_1,$$

show that the HUTS algorithm constructs a block LQ decomposition of the matrix C in the form $C = L_1Q_1$. It follows that

$$CC^* = CQ^*QC^* = \left[\begin{array}{c|c} L_1 & 0 \end{array}\right] \left[\begin{array}{c} L_1^* \\ \hline 0 \end{array}\right] = L_1L_1^* \quad \text{and} \quad C^* = Q_1^*L_1^*,$$

Thus, the system $YCC^* = AC^*$ is reduced to the equivalent system $YL_1L_1^* = AQ_1^*L_1^*$. If $\operatorname{rank}(C) = h$, then L_1 is invertible and the system further reduces to $YL_1 = AQ_1^*$. Thus, $Y = L_1^{-1}AQ_1^*$ can be found by inverting the block diagonal elements $L_{j,j}$ of L_1 and using back substitution. A more detailed description of the HUTS algorithm is given later in the paper.

Rationale. The original problem of finding the Moore–Penrose inverse of an $h \times h$ matrix is reduced to a collection of much smaller problems of calculating the Moore–Penrose inverses of an $h_j \times h_j$ matrix for each $j = 1, \ldots, q$. Thus, the number of flops needed for the numerical calculation of the respective Moore–Penrose inverses has been reduced from $\mathcal{O}(h^3)$ to $\mathcal{O}(\sum_{j=1}^{q} h_j^3) \sim q \cdot \mathcal{O}((h/q)^3) = \mathcal{O}(h^3)/q^2$.

Numerical calculations. The EBOS and HUTS algorithms can each be implemented in Matlab using the standard general purpose Matlab subroutines. In practice, these subroutines could possibly be replaced by more efficient bespoke techniques. See [13] for a general discussion of the relevant numerical methods. We have attached a Matlab program in Appendix A that implements the EBOS and HUTS algorithms. This program was used to test each procedure against the direct Matlab method. All numerical results reported here were obtained using this program.

Benefit. For large matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{(q\cdot dh)\times n}$ with $\operatorname{rank}(A) = q \cdot dh$, where $m, n, h, q, s \in \mathbb{N}$, $n = q \cdot dh + s$ and optimal sizes of the block partition $dh \in \mathbb{N}$ both algorithms are significantly faster than the direct method using the standard Matlab calculation $Y = AC^\dagger$, where $C^\dagger$ is the Moore–Penrose inverse of C. The HUTS algorithm is generally faster than the EBOS algorithm, but the EBOS algorithm is more accurate. For matrices $A, C \in \mathbb{R}^{6000\times 6000}$, where $A = A_\ell A_r$ and $A_\ell, A_r, C \in \mathbb{R}^{6000\times 6000}$ with independent randomly generated elements $a_{\ell,i,j}, a_{r,i,j}, c_{i,j} \in (0, 1)$ and with $q = 5$ the respective computation times for the EBOS, the HUTS and the direct method were $t_e \approx 41$ s, $t_h \approx 37$ s and $t_d \approx 64$ s. The Frobenius errors $\|YCC^* - AC^*\|_F$ were $\epsilon_e \sim \mathcal{O}(10^{-3})$, $\epsilon_h \sim \mathcal{O}(10^{-1})$ and $\epsilon_d \sim \mathcal{O}(10^{-3})$. Note that in the randomly generated matrix A, we consistently observed that $\max_{i,j} |a_{i,j}| \sim 1.6 \times 10^3$. For an error matrix $E \in \mathbb{R}^{6000\times 6000}$ with $|e_{i,j}| = \delta$ for each i, j, the Frobenius norm is given by $\|E\|_F = 6000\delta$. Thus, the average magnitude of the error for individual elements of E was less than $\mathcal{O}(10^{-4})$ for all three methods.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

2. Literature review. The solution of linear equations has a long history with contributions from many authors. The motivations for individual works are wide ranging, and the interconnections are many and varied. In many cases, the aim was to develop accurate solution procedures for large linear system. In other cases, the aim was simply to understand more about the basic structure of a matrix. Inevitably, the most effective methods for solution of large linear systems are also the methods that reveal the basic structure of the key matrices and are the same methods used for approximation of large matrices by smaller matrices of lesser rank. We list the most important historical references here but refer readers to the recent review in [20] for more details. The early work in [29, 32] was motivated by development of a systematic theory for solution of integral equations. The next contributions were motivated by applications to psychometrics [4, 5, 15, 16, 17, 30, 32, 33, 34], where an observed test score matrix was approximated by a lower rank matrix. The most notable results were the Eckart–Young theorem [4], which was essentially a finite-dimensional version of the Schmidt approximation theorem [29], and the principal component analysis of Hotelling [15, 16]. See also [14, 25]. The singular value decomposition (SVD) and the closely related QR and LQ algorithms [7, 8, 10, 11, 22, 31] were key motivations for much of the recent numerical work.

Development of a comprehensive theory of generalized inverse matrices [2, 3, 26, 27] was closely related to the conceptual development of the SVD and QR/LQ decompositions. These ideas are not only applied to solution of large linear systems but also to closely related problems of matrix approximation. See [13, 24] for further discussion and [9] for a recent solution of the generalized Eckart–Young problem. See also [23]. The work on matrix approximation has also found applications in signal processing [21], while more recent work has extended these approximations to bounded and unbounded linear maps on Hilbert space [6, 18, 19].

2.1. Review of the Elementary Block Operations Scheme (EBOS). Howlett and Torokhti [20] showed recently that the EBOS is an effective tool for solving the linear least squares matrix equation $YCC^* = AC^*$, where $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{h\times n}$ are given for large $m, n, h \in \mathbb{N}$ and where $n = h + s = \sum_{j=1}^{q} h_j + s$ with $h_j \in \mathbb{N}$ and $s \in \mathbb{N} - 1$. The EBOS algorithm is based on the following idea. If we write

$$C = \left[\begin{array}{c} C_1^{(0)} \\ \hline C_2^{(0)} \\ \vdots \\ C_q^{(0)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ \hline C_{2+}^{(0)} \end{array}\right] \in \mathbb{C}^{h\times n},$$

where $C_i^{(0)} \in \mathbb{C}^{h_i\times n}$ and $C_{2+}^{(0)} \in \mathbb{C}^{(h_2+\cdots+h_q)\times n}$ then

$$CC^* = \left[\begin{array}{c|ccc} C_1^{(0)}[C_1^{(0)}]^* & C_1^{(0)}[C_2^{(0)}]^* & \cdots & C_1^{(0)}[C_q^{(0)}]^* \\ \hline C_2^{(0)}[C_1^{(0)}]^* & C_2^{(0)}[C_2^{(0)}]^* & \cdots & C_2^{(0)}[Q_q^{(0)}]^* \\ \vdots & \vdots & \ddots & \vdots \\ C_q^{(0)}[C_1^{(0)}]^* & C_q^{(0)}[C_2^{(0)}]^* & \cdots & C_q^{(0)}[C_q^{(0)}]^* \end{array}\right] = \left[\begin{array}{c|c} C_1^{(0)}[C_1^{(0)}]^* & C_1^{(0)}[C_{2+}^{(0)}]^* \\ \hline C_{2+}^{(0)}[C_1^{(0)}]^* & C_{2+}^{(0)}[C_{2+}^{(0)}]^* \end{array}\right] \in \mathbb{C}^{h\times h}.$$

If we define

$$F_1 = \begin{bmatrix} I_{h_1} & 0 \\ -C_{2+}^{(0)}[C_1^{(0)}]^\dagger & I_{h_2+\cdots+h_q} \end{bmatrix},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

then

$$C^{(1))} = F_1 C^{(0)} = \left[\begin{array}{c} C_1^{(0)} \\ C_{2+}^{(0)}(I_{h_2+\cdots+h_q} - [C_1^{(0)}]^\dagger C_1^{(0)}) \end{array} \right] = \left[\begin{array}{c} C_1^{(0)} \\ C_{2+}^{(1)} \end{array} \right],$$

and since

$$\begin{aligned} C_{2+}^{(1)}[C_1^{(0)}]^* &= C_{2+}^{(0)}(I_{h_2+\cdots+h_q} - [C_1^{(0)}]^\dagger C_1^{(0)})[C_1^{(0)}]^* \\ &= C_{2+}^{(0)}(I_{h_2+\cdots+h_q} - [C_1^{(0)}]^*(C_1^{(0)}[C_1^{(0)}]^*)^\dagger C_1^{(0)})[C_1^{(0)}]^* \\ &= C_{2+}^{(0)}(I_{h_2+\cdots+h_q} - [C_1^{(0)}]^*[C_1^{(0)}]^*]^\dagger)[C_1^{(0)}]^* \\ &= 0, \end{aligned}$$

it follows that

$$C^{(1)}[C^{(1)}]^* = \left[\begin{array}{cc} C_1^{(0)}[C_1^{(0)}]^* & 0 \\ 0 & C_{2+}^{(1)}[C_{2+}^{(1)}]^* \end{array} \right],$$

or equivalently that

$$\left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \vdots \\ C_q^{(1)} \end{array} \right] \left[\begin{array}{cccc} [C_1^{(0)}]^* & [C_2^{(2)}]^* & \cdots & C_q^{(1)}]^* \end{array} \right] = \left[\begin{array}{c|ccc} C_1^{(0)}[C_1^{(0)}]^* & 0 & \cdots & 0 \\ \hline 0 & C_2^{(1)}[C_2^{(1)}]^* & \cdots & C_2^{(1)}[C_q^{(1)}]^* \\ \vdots & \vdots & \ddots & \vdots \\ 0 & C_q^{(1)}[C_2^{(1)}]^* & \cdots & C_q^{(1)}[C_q^{(1)}]^* \end{array} \right].$$

The EBOS algorithm works by repeating the above procedure on the reduced matrix

$$C_{2+}^{(1)} = \left[\begin{array}{c} C_2^{(1)} \\ \hline C_3^{(1)} \\ \vdots \\ C_q^{(1)} \end{array} \right] \in \mathbb{C}^{(h_2+\cdots+h_q)\times(n-h_1)},$$

using left multiplication by an elementary block matrix F_2 to perform block row operations that remove the off-diagonal elements in the first block row and block column of the matrix

$$C_{2+}^{(1)}[C_{2+}^{(1)}]^* = \left[\begin{array}{ccc} C_2^{(1)}[C_2^{(1)}]^* & \cdots & C_2^{(1)}[C_q^{(1)}]^* \\ \vdots & \ddots & \vdots \\ C_q^{(1)}[C_2^{(1)}]^* & \cdots & C_q^{(1)}[C_q^{(1)}]^* \end{array} \right].$$

The algorithm terminates when the matrix CC^* has been reduced to the block diagonal form

$$C_1^{(q-1)}[C_1^{(q-1)}]^* = \left[\begin{array}{cccc} C_1^{(0)}[C_1^{(0)}]^* & 0 & \cdots & 0 \\ 0 & C_2^{(1)}[C_2^{(1)}]^* & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_q^{(q-1)}[C_q^{(q-1)}]^* \end{array} \right] \in \mathbb{C}^{h\times h},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

with $C_i^{(i-1)}[C_i^{(i-1)}]^* \in \mathbb{C}^{h_i \times h_i}$ for each $i = 1, 2, \ldots, q$. A Matlab program to implement the EBOS algorithm was published in [18]. The solution requires finding the Moore–Penrose inverse $\{C_j^{(j-1)}[C_j^{(j-1)}]^*\}^\dagger \in \mathbb{C}^{h_j \times h_j}$ of each block diagonal element in the reduced coefficient matrix and then subsequently constructing the related Moore–Penrose inverse $[C_j^{(j-1)}]^\dagger = [C_j^{(j-1)}]^*\{C_j^{(j-1)}[C_j^{(j-1)}]^*\}^\dagger \in \mathbb{C}^{h_j \times h_j}$. A new Matlab program has been used to compare the direct method and the EBOS algorithm with the proposed HUTS algorithm.

2.2. Notational summary for the EBOS algorithm. We have used the following notation to describe the EBOS algorithm.

- At stage 1, we write $C = C^{(0)} \in \mathbb{C}^{(h_1+\cdots+h_q)\times n}$ for the initial matrix. The matrix $C^{(0)}$ is written as a block column with blocks $C_i^{(0)} \in \mathbb{C}^{h_i \times n}$ for each $i = 1, \ldots, q$. The consolidated block column comprising blocks $C_2^{(0)} \in \mathbb{C}^{h_2 \times n}, \ldots, C_q^{(0)} \in \mathbb{C}^{h_q \times n}$ is then written as $C_{2+}^{(0)} \in \mathbb{C}^{h_2+\cdots+h_q, n}$.
- At stage 2, we write $C = C^{(1)} \in \mathbb{C}^{(h_2+\cdots+h_q)\times n}$ for the initial matrix. The matrix $C^{(1)}$ is written as a block column with blocks $C_i^{(1)} \in \mathbb{C}^{h_i \times n}$ for each $i = 2, \ldots, q$. The consolidated block column comprising blocks $C_3^{(1)} \in \mathbb{C}^{h_3 \times n}, \ldots, C_q^{(1)} \in \mathbb{C}^{h_q \times n}$ is then written as $C_{3+}^{(1)} \in \mathbb{C}^{h_3+\cdots+h_q, n}$.
- At stage j, we write $C = C^{(j)} \in \mathbb{C}^{(h_j+\cdots+h_q)\times n}$ for the initial matrix. The matrix $C^{(j)}$ is written as a block column with blocks $C_i^{(j)} \in \mathbb{C}^{h_i \times n}$ for each $i = j, \ldots, q$. The consolidated block column comprising blocks $C_{j+1}^{(j)} \in \mathbb{C}^{h_{j+1} \times n}, \ldots, C_q^{(j)} \in \mathbb{C}^{h_q \times n}$ is then written as $C_{j+1}^{(j)} \in \mathbb{C}^{h_{j+1}+\cdots+h_q, n}$.

3. The Householder Unitary Transformations Scheme (HUTS). The HUTS uses right multiplication by a sequence of augmented elementary block Householder matrices to reduce the system $CC^*Y = AC^*$ to block lower triangular form.

3.1. Theoretical basis. We begin with a result established by Rotella and Zambettakis [28] in 1999.

Theorem 1. *For any full column rank matrix $R \in \mathbb{C}^{m\times r}$ with*

$$R = \left[\begin{array}{c} R_1 \\ \hline R_2 \end{array}\right],$$

where $R_1 \in \mathbb{C}^{r\times r}$ is nonsingular there is a unitary matrix $P \in \mathbb{C}^{r\times r}$ and a diagonal matrix $D \in \mathbb{C}_{r\times r}$ with $d_{jj} > 0$ for all $j = 1, \ldots, r$ such that

$$I_r + [R_1^{-1}]^* R_2^* R_2 R_1^{-1} = P^* D P,$$

where $I_r \in \mathbb{C}^{r\times r}$ is the identity matrix. If we define associated matrices $U = P^*\sqrt{D}PR_1 \in \mathbb{C}^{r\times r}$ and

$$V = \left[\begin{array}{c} R_1 + U \\ \hline R_2 \end{array}\right],$$

and we also define a block Householder matrix $H = I_m - 2V(V^*V)^{-1}V^*$ then

$$HR = \left[\begin{array}{c} -U \\ \hline 0 \end{array}\right],$$

where $I_m \in \mathbb{C}^{m\times m}$ is the identity matrix and $0 \in \mathbb{C}^{(m-r)\times r}$ is a zero matrix.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

In practice, it may happen that $R \in \mathbb{C}^{m\times n}$ with $m > n$ has $\mathrm{rank}(R) = r < n$. If so, it is necessary to consider a slightly more general procedure. As a precursor, we need the following elementary lemma.

LEMMA 1. *Let $S \in \mathbb{C}^{m\times n}$, where $m > n$ and suppose that $\mathrm{rank}(S) = k$. There exists a permutation matrix $P \in \mathbb{C}^{m\times m}$ such that*

$$R = PS \iff R = \left[\begin{array}{c} R_1 \\ \hline FR_1 \end{array}\right],$$

where $R_1 \in \mathbb{C}^{n\times n}$ with $\mathrm{rank}(R_1) = k$ and $F \in \mathbb{C}^{(m-n)\times n}$.

Proof. We simply reorder the rows of S to ensure that the first k rows are independent. We can do this by pre-multiplying S by a suitable permutation matrix $P \in \mathbb{C}^{m\times m}$. Since $P^*P = I_m$ this is a unitary transformation. Therefore, $R^\dagger = (PS)^\dagger = S^\dagger P$. Define rows $\boldsymbol{r}_i^* = \boldsymbol{e}_i^* R \in \mathbb{C}^{1\times n}$ for each $i = 1,\dots,m$, where $\{\boldsymbol{e}_1,\dots,\boldsymbol{e}_m\} \in \mathbb{C}^{m\times 1}$ is the standard orthonormal basis for $\mathbb{C}^m$. For each $i = n+1,\dots,m$, the row vector $\boldsymbol{r}_i^*$ can be expressed as a linear combination of the row vectors $\boldsymbol{r}_1^*,\dots,\boldsymbol{r}_k^*$. Thus, we can find uniquely determined constants f_{ij} for each $i = n+1,\dots,m$ and $j = 1,\dots,k$ such that

$$\begin{bmatrix} \boldsymbol{r}_{n+1}^* \\ \vdots \\ \boldsymbol{r}_m^* \end{bmatrix} = \left[\begin{array}{ccc|ccc} f_{n1} & \cdots & f_{nk} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mk} & 0 & \cdots & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{r}_1^* \\ \vdots \\ \boldsymbol{r}_k^* \\ \hline \boldsymbol{r}_{k+1}^* \\ \vdots \\ \boldsymbol{r}_n^* \end{array}\right] \iff R_2 = FR_1 \iff R = \left[\begin{array}{c} R_1 \\ \hline FR_1 \end{array}\right].$$

We could write this process more compactly as follows. We wish to solve $R_2 = FR_1$, where $R_1 \in \mathbb{C}^{n\times n}$ and $R_2 \in \mathbb{C}^{(m-n)\times n}$ are known and $F \in \mathbb{C}^{(m-r)\times n}$ is unknown. The best approximate solution to this equation is $F_0 = R_2R_1^*(R_1R_1^*)^\dagger = R_2R_1^\dagger$. Since we know that a solution exists the best approximate solution must also be a true solution. Therefore, $R_2 = F_0R_1 = R_2R_1^\dagger R_1$. □

Now we can prove a more general version of Theorem 1.

THEOREM 2. *Suppose $m, n \in \mathbb{N}$ with $m > n$. Let $R \in \mathbb{C}^{m\times n}$ with*

$$R = \left[\begin{array}{c} R_1 \\ \hline FR_1 \end{array}\right],$$

where $R_1 \in \mathbb{C}^{n\times n}$, $F \in \mathbb{C}^{(m-n)\times n}$. There is a unitary matrix $P \in \mathbb{C}^{n\times n}$ and a diagonal matrix $D \in \mathbb{C}_{n\times n}$ with $d_{jj} > 0$ for all $j = 1,\dots,n$ such that

$$I_n + F^*F = PDP^*,$$

where $I_n \in \mathbb{C}^{n\times n}$ is the identity matrix. If we define associated matrices $U = P\sqrt{D}P^*R_1 \in \mathbb{C}^{n\times n}$ and

$$V = \left[\begin{array}{c} R_1 + U \\ \hline FR_1 \end{array}\right],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and we also define a block Householder matrix $H = I_m - 2VV^\dagger$ then

$$HR = \left[\frac{-U}{0}\right],$$

where $I_m \in \mathbb{C}^{m\times m}$ is the identity matrix and $0 \in \mathbb{C}^{(m-n)\times n}$ is a zero matrix.

Proof. Suppose for the moment that $U \in \mathbb{C}^{n\times n}$ is unknown. The definitions of H and V and the formula $V^\dagger = (V^*V)^\dagger V^*$ show that

$$HR = \left[\frac{R_1}{FR_1}\right] - 2\left[\frac{R_1+U}{FR_1}\right](V^*V)^\dagger V^*R = \left[\frac{R_1 - 2(R_1+U)(V^*V)^\dagger V^*R}{FR_1 - 2FR_1(V^*V)^\dagger V^*R}\right] = \left[\frac{E_1}{E_2}\right], \tag{2}$$

where

$$E_1 = R_1 - 2(R_1+U)(V^*V)^\dagger V^*R, \tag{3}$$

and

$$E_2 = FR_1 - 2FR_1(V^*V)^\dagger V^*R. \tag{4}$$

We wish choose U so that $E_1 = -U$ and $E_2 = 0$. We will begin by considering the equation

$$V^*V = 2V^*R. \tag{5}$$

By expanding both sides, we can rewrite (5) in the form

$$R_1^*R_1 + U^*R_1 + R_1^*U + U^*U + R_1^*F^*FR_1 = 2(R_1^*R_1 + U^*R_1 + R_1^*F^*FR_1).$$

The identity $R^*R = R_1^*(I_n + F^*F)R_1$ shows that this equation reduces to the simplified equation

$$R^*R - U^*U = R_1^*U - U^*R_1, \tag{6}$$

The matrix $R^*R - U^*U$ is hermitian. It follows from (6) that $R_1^*U - U^*R_1$ is also hermitian. Therefore, $R_1^*U = U^*R_1$ and so (5) is satisfied if and only if

$$U^*U = R^*R = R_1^*(I_n + F^*F)R_1. \tag{7}$$

The matrix $I_n + F^*F$ is hermitian and positive. Therefore, we can write $I_n + F^*F = PDP^*$, where P is unitary and D is diagonal and positive. If we now define $U = P\sqrt{D}P^*R_1$, then

$$\begin{aligned} U^*U &= R_1^*P\sqrt{D}P^* \cdot P\sqrt{D}P^*R_1 \\ &= R_1^*PDP^*R_1 \\ &= R_1^*(I_n + F^*F)R_1. \end{aligned}$$

Therefore, (7) is satisfied and $R_1^*U = U^*R_1$. Importantly, we also know that (5) is also satisfied. Now we also have

$$\begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} R_1 \\ FR_1 \end{bmatrix} = R_1 \quad \text{and} \quad \begin{bmatrix} 0 & I_{m-n} \end{bmatrix} \begin{bmatrix} R_1 \\ FR_1 \end{bmatrix} = FR_1,$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and so

$$\begin{aligned}
E_2 &= FR_1 - 2FR_1(V^*V)^\dagger V^*R \\
&= FR_1 - FR_1(V^*V)^\dagger V^*V \\
&= FR_1[I_n - V^\dagger V] \\
&= \begin{bmatrix} 0 & I_n \end{bmatrix} V[I_n - V^\dagger V] \\
&= 0. \qquad (8)
\end{aligned}$$

Finally, we can use (2) and (5) to deduce that

$$E_1 = (R_1 + U)[I - (V^*V)^\dagger V^*V] - U = \begin{bmatrix} I & 0 \end{bmatrix} V[I - V^\dagger V] - U = -U,$$

as required. □

REMARK 1. *If we choose* $F = R_2R_1^\dagger$, *then* $R_2 = FR_1$. *In this case,* $I + F^*F = I + [R_1^\dagger]^*R_2^*R_2R_1^\dagger$ *and* $U = P\sqrt{D}P^*R_1$. *Therefore, as long as the condition* $\operatorname{rank} R_1 = \operatorname{rank} R$ *is satisfied, the calculations can proceed as in the original theorem by Rotella and Zambettakis with* R_1^{-1} *replaced by* $R_1^\dagger$.

EXAMPLE 1. *Let*

$$R = \left[\begin{array}{c} R_1 \\ \hline R_2 \end{array}\right], \text{ where } R_1 = \begin{bmatrix} 1 & 1 & 1 & 4 \\ -1 & 0 & 2 & -2 \\ 0 & 1 & 3 & 2 \end{bmatrix} \text{ and } R_2 = \begin{bmatrix} 2 & 1 & -1 & 6 \\ 2 & 0 & -4 & 4 \\ 4 & 1 & -5 & 10 \end{bmatrix}.$$

Note that $\operatorname{rank}(R_1) = \operatorname{rank}(R) = 2$. *We calculate*

$$F = R_2R_1^\dagger = \begin{bmatrix} 1.0000 & -1.0000 & -0.0000 \\ 0.6667 & -1.3333 & -0.6667 \\ 1.6667 & -2.3333 & -0.6667 \end{bmatrix},$$

which allows us to define $R_1 = R_1$ *and to write* $R_2 = FR_1 = FR_1$. *Now we define*

$$T = I_3 + F^*F = \begin{bmatrix} 5.2222 & -5.7778 & -1.5556 \\ -5.7778 & 9.2222 & 2.4444 \\ -1.5556 & 2.4444 & 1.8889 \end{bmatrix}.$$

The singular value decomposition $T = PDP^*$ *is defined by*

$$P = \begin{bmatrix} -0.5620 & 0.5923 & 0.5774 \\ 0.7939 & 0.1906 & 0.5774 \\ 0.2319 & 0.7829 & -0.5774 \end{bmatrix} \text{ and } D = \begin{bmatrix} 14.0263 & 0 & 0 \\ 0 & 1.3071 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

Next we define

$$U = P\sqrt{D}P^*R_1 = \begin{bmatrix} 3.1261 & 1.6260 & -1.3743 & 9.5041 \\ -3.9444 & -0.6819 & 5.8431 & -9.2525 \\ -0.8183 & 0.9441 & 4.4688 & 0.2516 \end{bmatrix},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

$$V = \left[\begin{array}{c} R_1 + U \\ \hline FR_1 \end{array}\right] = \left[\begin{array}{rrrr} 4.1261 & 2.6260 & -0.3743 & 13.5041 \\ -4.9444 & -0.6819 & 7.8431 & -11.2525 \\ -0.8183 & 1.9441 & 7.4688 & 2.2516 \\ \hline 2.0000 & 1.0000 & -1.0000 & 6.0000 \\ 2.0000 & 0 & -4.0000 & 4.0000 \\ 4.0000 & 1.0000 & -5.0000 & 10.0000 \end{array}\right],$$

and

$$H = I_6 - 2VV^\dagger = \left[\begin{array}{rrr|rrr} -0.0578 & 0.3537 & -0.7041 & -0.4116 & -0.0408 & -0.4524 \\ 0.3537 & 0.1333 & -0.5130 & 0.2205 & 0.4002 & 0.6206 \\ -0.7041 & -0.5130 & -0.2171 & -0.1911 & 0.3593 & 0.1682 \\ \hline -0.4116 & 0.2205 & -0.1911 & 0.8307 & -0.0587 & -0.2280 \\ -0.0408 & 0.4002 & 0.3593 & -0.0587 & 0.7988 & -0.2598 \\ -0.4524 & 0.6206 & 0.1682 & -0.2280 & -0.2598 & 0.5122 \end{array}\right].$$

Finally, we obtain

$$HR = \left[\begin{array}{rrrr} -3.1261 & -1.6260 & 1.3743 & -9.5041 \\ 3.9444 & 0.6819 & -5.8431 & 9.2525 \\ 0.8183 & -0.9441 & -4.4688 & -0.2516 \\ \hline 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 \\ 0.0000 & 0 & -0.0000 & 0.0000 \end{array}\right] = \left[\begin{array}{c} -U \\ \hline 0 \end{array}\right],$$

as desired. □

3.2. Solution of $YCC^* = AC^*$ using the block LQ decomposition of C. We wish to solve the equation $YCC^* = AC^*$, where $A \in \mathbb{C}^{m\times n}$ with rank $A = h$ and $C \in \mathbb{C}^{h\times n}$, where $n = h + s$ and $s \in \mathbb{N} - 1$. We assume that A and C are known and that $Y \in \mathbb{C}^{m\times h}$ is unknown. The solution to this system can be written in the form $Y = AC^\dagger$, where $C^\dagger$ is the Moore–Penrose inverse of C. When m, n are large we will show that the original system can be reduced to a simpler equivalent system $YL_1L_1^* = AQ^*L_1^*$, where $L_1 \in \mathbb{C}^{h\times h}$ is a block lower triangular matrix with blocks $L_{i,j} \in \mathbb{C}^{h_i\times h_j}$ for each $i, j = 1, \ldots, q$ such that $L_{i,j} = 0$ for $i < j$ and $Q \in \mathbb{C}^{n\times n}$ is a unitary matrix. The matrix Q is designed to provide a block LQ decomposition of the matrix C with

$$C = LQ \iff C = \left[\begin{array}{c|c} L_1 & 0 \end{array}\right] \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right] \iff C = L_1Q_1,$$

where $Q_1 \in \mathbb{C}^{h\times n}$ and $Q_2 \in \mathbb{C}^{s\times n}$. The identity $QQ^* = Q^*Q = I_n$ shows that $Q_1Q_1^* = I_h, Q_2Q_2^* = I_s$, $Q_1Q_2^* = 0_{h,s}$, $Q_2Q_1^* = 0_{s,h}$ and $Q_1^*Q_1 + Q_2^*Q_2 = I_n$. Therefore, $C^\dagger = Q_1^*L_1^\dagger$ and

$$YCC^* = AC^* \iff YL_1Q_1Q_1^*L_1 = AQ_1^*L_1^* \iff YL_1L_1^* = AQ_1^*L_1^*.$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

If $\operatorname{rank} Q = h$, then L_1^* is invertible and hence $YL_1 = AQ_1^*$. The coefficient matrix L_1 is lower triangular and so this system can be solved by back substitution. If we write

$$Y = \left[\begin{array}{cccc} Y_1 & Y_2 & \cdots & Y_q \end{array}\right], \quad L_1 = \left[\begin{array}{cccc} L_{1,1} & 0 & \cdots & 0 \\ L_{2,1} & L_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{q,1} & L_{q,2} & \cdots & L_{q,q} \end{array}\right], \quad Q_1 = \left[\begin{array}{c} Q_{1,1} \\ Q_{2,1} \\ \vdots \\ Q_{q,1} \end{array}\right],$$

where $Y_j \in \mathbb{C}^{m\times h_j}$, $L_{i,j}, C_{i,j} \in \mathbb{C}^{h_i\times h_j}$, and $Q_{i,1} \in \mathbb{C}^{h_i\times n}$ then the system $YL_1 = AQ_1^*$ can be rewritten as

$$\begin{aligned} Y_1L_{1,1} + Y_2L_{2,1} + Y_3L_{3,1} + \cdots + Y_1L_{q,1} &= AQ_{1,1}^* \\ Y_2L_{2,2} + Y_3L_{3,2} + \cdots + Y_qL_{q,2} &= AQ_{2,1}^* \\ Y_3L_{3,3} + \cdots + Y_qL_{q,3} &= AQ_{3,1}^* \\ \vdots &= \vdots \\ Y_qL_{q,q} &= AQ_{q,1}^*. \end{aligned}$$

Back substitution now shows that these equations have the solution

$$\begin{aligned} Y_q &= AQ_{q,1}^*L_{q,q}^{-1} \\ Y_{q-1} &= [AQ_{q-1,1}^* - Y_qL_{q,q-1}]L_{q-1,q-1}^{-1} \\ Y_{q-2} &= [AQ_{q-2,1}^* - Y_{q-1}L_{q-1,q-2} - Y_qL_{q,q-2}]L_{q-2,q-2}^{-1} \\ \vdots &= \vdots \\ Y_1 &= [AQ_{1,1}^* - Y_2L_{2,1} - Y_3L_{3,1} - Y_4L_{4,1} - \cdots - Y_qL_{q,1}]L_{1,1}^{-1}. \end{aligned}$$

This solution satisfies the least squares equation $YL_1L_1^* = AQ_1^*L_1^*$ and hence also satisfies the original equation $YCC^* = AC^*$. We will use our proposed HUTS algorithm to find this solution.

3.3. The HUTS algorithm. In essence, we wish to replace the equation $YC = A \iff YL^{(0)} = A$ with the equation $YL_1 = AQ_1^*$. We assume that $\operatorname{rank}(C) = h$. We will find a sequence $\{K^{(s)}\}_{s=1}^q$ of unitary transformations, where each $K^{(s)} \in \mathbb{C}^{n\times n}$ is defined by an augmented block Householder matrix, and a unitary matrix $Q^* = K^{(1)}\cdots K^{(q)}$ such that $L^{(q)} = L^{(0)}Q^* = L^{(0)}K^{(1)}\cdots K^{(q)} = [L_1 \mid 0]$, where $L_1 \in \mathbb{C}^{h\times h}$ is block lower triangular and nonsingular and $0 \in \mathbb{C}^{h\times s}$. Of course we must apply the same transformation to both sides of the equation $YL^{(0)} = A$ and so we get $YL^{(0)}K^{(1)}\cdots K^{(q)} = AK^{(1)}\cdots K^{(q)} \iff YL_1 = AQ_1^*$. More specifically, we have

$$C = \left[\begin{array}{ccccc} C_{1,1} & C_{1,2} & \cdots & C_{1,q} & C_{1,q+1} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,q} & C_{2,q+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{q,1} & C_{q,2} & \cdots & C_{q,q} & C_{q,q+1} \end{array}\right] = \left[\begin{array}{ccccc} L_{1,1}^{(0)} & L_{1,2}^{(0)} & \cdots & L_{1,q}^{(0)} & L_{1,q+1}^{(0)} \\ L_{2,1}^{(0)} & L_{2,2}^{(0)} & \cdots & L_{2,q}^{(0)} & L_{2,q+1}^{(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{q,1}^{(0)} & L_{q,2}^{(0)} & \cdots & L_{q,q}^{(0)} & L_{q,q+1}^{(0)} \end{array}\right] = L^{(0)},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

where $C_{i,j} \in \mathbb{C}^{h_i \times h_j}$ when $j \leq q$ and $C_{i,q+1} \in \mathbb{C}^{h_i \times s}$. We wish to reduce $L^{(0)}$ to the form

$$L^{(0)}Q^* = L^{(0)}K^{(1)} \cdots K^{(q)} = \begin{bmatrix} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{q,1}^{(1)} & L_{q,2}^{(2)} & \cdots & L_{q,q}^{(q)} & 0 \end{bmatrix} = L^{(q)} = \left[\begin{array}{c|c} L_1^{(q)} & 0 \end{array} \right],$$

where $L_1^{(0)} \in \mathbb{C}^{h \times h}$ is block lower triangular with $L_{i,j}^{(j)} \in \mathbb{C}^{h_i \times h_j}$ for $j \leq q$ and $L_{j,j}^{(j)}$ nonsingular and where $0 \in \mathbb{C}^{h \times s}$. We wish to use right multiplication by a sequence of augmented block Householder matrices to construct the desired LQ decomposition $C = LQ$. Thus, the basic procedure proposed by Rotella and Zambettakis must be transposed.

Stage 1. We begin the reduction by using consolidated partitions to define

$$L^{(0)} = \begin{bmatrix} L_{1,1}^{(0)} & L_{1,2+}^{(0)} \\ L_{2+,1}^{(0)} & L_{2+,2+}^{(0)} \end{bmatrix} \text{ and } A^{(0)} = \begin{bmatrix} A_1^{(0)} & A_{2+}^{(0)} \end{bmatrix}.$$

The transposed Rotella–Zambettakis transformation acts on the matrix blocks in $L^{(0)}$ in such a way that the block element $L_{1,2+}^{(0)}$ is eliminated. We will use right multiplication by a block Householder matrix to eliminate this term. It is necessary that $\operatorname{rank}(L_{1,1}^{(0)}) = \operatorname{rank}([L_{1,1}^{(0)} \mid L_{1,2+}^{(0)}])$. If not we must first permute the columns of $L^{(0)}$ to ensure that this condition is satisfied. This is always possible because $L_{1,1}^{(0)} \in \mathbb{C}^{h_1 \times h_1}$ and $\operatorname{rank}([L_{1,1}^{(0)} \mid L_{1,2+}^{(0)}]) = h_1$. We must also permute the columns of $A^{(0)}$ in the same way. Define

$$W_1 = I_{h_1} + \{L_{1,1}^{(0)}\}^\dagger L_{1,2+}^{(0)} [L_{1,2+}^{(0)}]^* \{[L_{1,1}^{(0)}]^*\}^\dagger,$$

and find a unitary block matrix P_1 such that $W_1 = P_1 D_1 P_1^*$, where D_1 is a positive diagonal matrix with $d_{1,j,j} \geq 1$ for all $j = 1, \ldots, h_1$. Set $L_{1,1}^{(1)} = L_{1,1}^{(0)} P_1 \sqrt{D_1} P_1^*$ and

$$V_1 = \left[\begin{array}{c|c} L_{1,1}^{(0)} - L_{1,1}^{(1)} & L_{1,2+}^{(0)} \end{array} \right] = \left[\begin{array}{c|c} V_{1,1} & V_{1,2} \end{array} \right],$$

and define the block Householder matrix

$$\begin{aligned} K^{(1)} &= \begin{bmatrix} I_{h_1} & 0 \\ 0 & I_{h_2+\cdots+h_n} \end{bmatrix} - 2 \begin{bmatrix} V_{1,1}^* \\ V_{1,2}^* \end{bmatrix} (V_1 V_1^*)^\dagger \left[\begin{array}{c|c} V_{1,1} & V_{1,2} \end{array} \right] \\ &= \begin{bmatrix} I_{h_1} - 2V_{1,1}^*(V_1V_1^*)^\dagger V_{1,1} & -2V_{1,1}^*(V_1V_1^*)^\dagger V_{1,2} \\ -2V_{1,2}^*(V_1V_1^*)^\dagger V_{1,1} & I_{h_2+\cdots+h_n} - 2V_{1,2}^*(V_1V_1^*)^\dagger V_{1,2} \end{bmatrix} = \begin{bmatrix} K_{1,1}^{(1)} & K_{1,2}^{(1)} \\ K_{2,1}^{(1)} & K_{2,2}^{(1)} \end{bmatrix}. \end{aligned}$$

It follows that

$$L^{(0)}K^{(1)} = \begin{bmatrix} L_{1,1}^{(0)} & L_{1,2+}^{(0)} \\ L_{2+,1}^{(0)} & L_{2+,2+}^{(0)} \end{bmatrix} \begin{bmatrix} K_{1,1}^{(1)} & K_{1,2}^{(1)} \\ K_{2,1}^{(1)} & K_{2,2}^{(1)} \end{bmatrix} = \begin{bmatrix} L_{1,1}^{(1)} & 0 \\ L_{2+,1}^{(1)} & L_{2+,2+}^{(1)} \end{bmatrix} = \left[\begin{array}{c|ccc} L_{1,1}^{(1)} & 0 & \cdots & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(1)} & \cdots & L_{2,q}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ L_{q,1}^{(1)} & L_{q,2}^{(1)} & \cdots & L_{q,q}^{(1)} \end{array} \right] = L^{(1)},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and

$$A^{(0)}K^{(1)} = \left[\begin{array}{cc} A_1^{(0)} & A_{2+}^{(0)} \end{array}\right] \left[\begin{array}{cc} K_{1,1}^{(1)} & K_{1,2}^{(1)} \\ K_{2,1}^{(1)} & K_{2,2}^{(1)} \end{array}\right] = \left[\begin{array}{cc} A_1^{(1)} & A_{2+}^{(1)} \end{array}\right] = \left[\begin{array}{c|ccc} A_1^{(1)} & A_2^{(1)} & \cdots & A_q^{(1)} \end{array}\right].$$

This completes Stage 1.

Stage 2. We use consolidated partitions to write

$$L^{(1)} = \left[\begin{array}{c|cc} L_{1,1}^{(1)} & 0 & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(1)} & L_{2,3+}^{(1)} \\ L_{3+,1}^{(1)} & L_{3+,2}^{(1)} & L_{3+,3+}^{(1)} \end{array}\right] \quad \text{and} \quad A^{(1)} = \left[\begin{array}{c|cc} A_1^{(1)} & A_2^{(1)} & A_{3+}^{(1)} \end{array}\right].$$

The transposed Rotella–Zambettakis transformation acts on the matrix blocks

$$L_{2+,2+}^{(1)} = \left[\begin{array}{cc} L_{22}^{(1)} & L_{2,3+}^{(1)} \\ L_{3+,2}^{(1)} & L_{3+,3+}^{(1)} \end{array}\right],$$

in such a way that the element $L_{1,2+}^{(0)}$ is eliminated. We require $\operatorname{rank}(L_{2+,2+}^{(1)}) = \operatorname{rank}([L_{2,2+}^{(1)} \mid L_{2,3+}^{(1)}]) = h_2$. If not we must permute the columns of $L_{2,2+}^{(1)}$ until the condition is satisfied. Define

$$W_2 = I_{h_2} + \{L_{2,2}^{(1)}\}^\dagger L_{2,3+}^{(1)} [L_{2,3+}^{(1)}]^* \{[L_{2,2}^{(1)}]^*\}^\dagger,$$

and find a unitary matrix P_2 such that $W_2 = P_2 D_2 P_2^*$, where D_2 is a positive diagonal matrix with $d_{2,j,j} \geq 1$ for all $j = 1, \ldots, h_2$. Set $L_{2,2}^{(2)} = L_{2,2}^{(1)} P_2 \sqrt{D_2} P_2^*$ and

$$V_2 = \left[\begin{array}{c|c} L_{2,2}^{(1)} - L_{2,2}^{(2)} & L_{23,c}^{(1)} \end{array}\right] = \left[\begin{array}{c|c} V_{2,1}^{(1)} & V_{2,2}^{(1)} \end{array}\right],$$

and define the block Householder transformation

$$\begin{aligned} \mathcal{K}^{(2)} &= \left[\begin{array}{cc} I_{h_2} & 0 \\ 0 & I_{h_3+\cdots+h_n} \end{array}\right] - 2 \left[\begin{array}{c} V_{2,1}^* \\ V_{2,2}^* \end{array}\right] (V_2 V_2^*)^\dagger \left[\begin{array}{c|c} V_{2,1} & V_{2,2} \end{array}\right] \\ &= \left[\begin{array}{cc} I_{h_2} - 2V_{2,1}^*(V_2V_2^*)^\dagger V_{2,1} & -2V_{2,1}^*(V_2V_2^*)^\dagger V_{2,2} \\ -2V_{2,1}^*(V_2V_2^*)^\dagger V_{1,1} & I_{h_3+\cdots+h_n} - 2V_{2,1}^*(V_2V_2^*)^\dagger V_{2,2} \end{array}\right] \\ &= \left[\begin{array}{cc} \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ \mathcal{K}_{2,1}^{(2)} & \mathcal{K}_{2,2}^{(2)} \end{array}\right]. \end{aligned}$$

Right multiplication by $\mathcal{K}^{(2)}$ on the restricted system now gives

$$L_{2+,2+}^{(1)}\mathcal{K}^{(2)} = \left[\begin{array}{cc} L_{2,2}^{(1)} & L_{2,3+}^{(1)} \\ L_{3+,2}^{(1)} & L_{3+,3+}^{(1)} \end{array}\right] \left[\begin{array}{cc} \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ \mathcal{K}_{2,1}^{(2)} & \mathcal{K}_{2,2}^{(2)} \end{array}\right] = \left[\begin{array}{cc} L_{2,2}^{(2)} & 0 \\ L_{3+,2}^{(2)} & L_{3+,3+}^{(2)} \end{array}\right] = \left[\begin{array}{c|ccc} L_{2,2}^{(2)} & 0 & \cdots & 0 \\ \hline L_{3,2}^{(2)} & L_{3,3}^{(2)} & \cdots & L_{3,q}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ L_{q,1}^{(2)} & L_{q,2}^{(2)} & \cdots & L_{q,q}^{(2)} \end{array}\right],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and

$$A_{2+}^{(1)}K^{(2)} = \begin{bmatrix} A_2^{(1)} & A_{3+}^{(1)} \end{bmatrix} \begin{bmatrix} K_{1,1}^{(2)} & K_{1,2}^{(2)} \\ K_{2,1}^{(2)} & K_{2,2}^{(2)} \end{bmatrix} = \begin{bmatrix} A_2^{(2)} & A_{3+}^{(2)} \end{bmatrix} = \left[\begin{array}{c|ccc} A_2^{(2)} & A_3^{(2)} & \cdots & A_q^{(2)} \end{array} \right].$$

The augmented block Householder matrix

$$\begin{aligned} K^{(2)} &= \begin{bmatrix} I_{h_1} & 0 \\ 0 & I_{h_2+\cdots+h_q} - 2V_2(V_2^*V_2)^{\dagger}V_2^* \end{bmatrix} \\ &= \begin{bmatrix} I_{h_1} & 0 & 0 \\ 0 & I_{h_2} - 2V_{2,1}^*(V_1V_1^*)^{\dagger}V_{2,1} & -2V_{2,1}^*(V_2V_2^*)^{\dagger}V_{2,2} \\ 0 & -2V_{2,1}^*(V_1V_1^*)^{\dagger}V_{1,1} & I_{h_3+\cdots+h_n} - 2V_{2,1}^*(V_1V_1^*)^{\dagger}V_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} I_{h_1} & 0 & 0 \\ 0 & \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ 0 & \mathcal{K}_{2,1}^{(2)} & \mathcal{K}_{2,2}^{(2)} \end{bmatrix}, \end{aligned}$$

simply implements the restricted reduction on the whole system. Indeed, we have

$$\begin{aligned} L^{(1)}K^{(2)} &= \left[\begin{array}{c|cc} L_{1,1}^{(1)} & 0 & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(1)} & L_{2,3+}^{(1)} \\ L_{3+,1}^{(1)} & L_{3+,2}^{(1)} & L_{3+,3+}^{(1)} \end{array} \right] \left[\begin{array}{c|cc} I_{h_1} & 0 & 0 \\ \hline 0 & \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ 0 & \mathcal{K}_{2,1}^{(2)} & \mathcal{K}_{,2}^{(2)} \end{array} \right] \\ &= \left[\begin{array}{c|cc} L_{1,1}^{(1)} & 0 & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(2)} & 0 \\ L_{3+,1}^{(1)} & L_{3+,2}^{(2)} & L_{3+,3+}^{(2)} \end{array} \right] \\ &= \left[\begin{array}{c|cccc} L_{1,1}^{(1)} & 0 & 0 & \cdots & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(2)} & 0 & \cdots & 0 \\ L_{3,1}^{(1)} & L_{3,2}^{(2)} & L_{3,3}^{(2)} & \cdots & L_{3,q}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{q,1}^{(1)} & L_{q,2}^{(2)} & L_{q,3}^{(2)} & \cdots & L_{q,q}^{(2)} \end{array} \right] = L^{(2)}, \end{aligned}$$

and

$$\begin{aligned} A^{(1)}K^{(2)} &= \left[\begin{array}{c|cc} A_1^{(1)} & A_2^{(1)} & A_{3+}^{(1)} \end{array} \right] \left[\begin{array}{c|cc} I_{h_1} & 0 & 0 \\ \hline 0 & \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ 0 & \mathcal{K}_{2,1}^{(1)} & \mathcal{K}_{2,2}^{(2)} \end{array} \right] \\ &= \left[\begin{array}{c|cc} A_1^{(1)} & A_2^{(2)} & A_{3+}^{(2)} \end{array} \right]. \end{aligned}$$

This completes Stage 2.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

Stage r. In general, we have

$$L^{(r-1)} = \left[\begin{array}{cccc|cc} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L_{r-1,1}^{(1)} & L_{r-1,2}^{(2)} & \cdots & L_{r-1,r-1}^{(r-1)} & 0 & 0 \\ \hline L_{r,1}^{(1)} & L_{r,2}^{(2)} & \cdots & L_{r,r-1}^{(r-1)} & L_{r,r}^{(r-1)} & L_{r,r+1+}^{(r-1)} \\ L_{r+1+,1}^{(1)} & L_{r+1+,2}^{(2)} & \cdots & L_{r+1+,r-1}^{(r-1)} & L_{r+1+,r}^{(r-1)} & L_{r+1+,r+1+}^{(r-1)} \end{array}\right],$$

and

$$A^{(r-1)} = \left[\begin{array}{cccc|cc} A_1^{(1)} & A_2^{(2)} & \cdots & A_{r-1}^{(r-1)} & A_r^{(r-1)} & A_{r+1+}^{(r-1)} \end{array}\right].$$

The Rotella–Zambettakis transformation acts on the matrix blocks

$$L_{r+,r+}^{(r-1)} = \begin{bmatrix} L_{r,r}^{(r-1)} & L_{r,r+1+}^{(r-1)} \\ L_{r+1+,r}^{(r-1)} & L_{r+1+,r+1+}^{(r-1)} \end{bmatrix},$$

and is designed to eliminate the term $L_{r,r+1+}^{(r-1)}$. If necessary, we must permute the columns of $L_{r,r+}^{(r-1)} = [L_{rr}^{(r-1)} \mid L_{r,r+1+}^{(r-1)}]$ to ensure $\mathrm{rank}([L_{rr}^{(r-1)}]) = \mathrm{rank}([L_{r,r+}^{(r-1)}]) = h_r$. Now define

$$W_r = I_{h_r} + \{L_{r,r}^{(r-1)}\}^\dagger L_{r,r+1+}^{(r-1)} [L_{r,r+1+}^{(r-1)}]^* \{[L_{r,r}^{(r-1)}]^*\}^\dagger,$$

and find a unitary matrix P_r such that $W_r = P_r D_r P_r^*$, where D_r is a positive diagonal matrix with $d_{r,j,j} \geq 1$ for all $j = 1, \ldots, h_r$. Set $L_{r,r}^{(r)} = L_{r,r}^{(r-1)} P_r \sqrt{D_r} P_r*$ and

$$V_r = \left[\begin{array}{c|c} L_{r,r}^{(r-1)} - L_{r,r}^{(r)} & L_{r,r+1+}^{(r-1)} \end{array}\right] = \left[\begin{array}{c|c} V_{r,1} & V_{r,2} \end{array}\right],$$

and define the block Householder matrix

$$\begin{aligned} \mathcal{K}^{(r)} &= \begin{bmatrix} I_{h_r} & 0 \\ 0 & I_{h_{r+1}+\cdots+h_n} \end{bmatrix} - 2 \begin{bmatrix} V_{r,1}^* \\ V_{r,2}^* \end{bmatrix} (V_r V_r^*)^\dagger \left[\begin{array}{c|c} V_{r,1} & V_{r,2} \end{array}\right] \\ &= \begin{bmatrix} I_{h_r} - 2V_{r,1}^*(V_r V_r^*)^\dagger V_{r,1} & -2V_{r,1}^*(V_r V_r^*)^\dagger V_{r,2} \\ -2V_{r,1}^*(V_r V_r^*)^\dagger V_{r,1} & I_{h_{r+1}+\cdots+h_n} - 2V_{r,1}^*(V_r V_r^*)^\dagger V_{r,2} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{K}_{1,1}^{(r)} & \mathcal{K}_{1,2}^{(r)} \\ \mathcal{K}_{2,1}^{(r)} & \mathcal{K}_{2,2}^{(r)} \end{bmatrix}. \end{aligned}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

Right multiplication by $\mathcal{K}^{(r)}$ on the restricted system now gives

$$
\begin{aligned}
L^{(1)}_{r+,r+}\mathcal{K}^{(r)} &= \begin{bmatrix} L^{(r-1)}_{r,r} & L^{(r-11)}_{r,r+1+} \\ L^{(r-1)}_{r+1+,r} & L^{(r-1)}_{r+1+,r+1+} \end{bmatrix} \begin{bmatrix} \mathcal{K}^{(r)}_{1,1} & \mathcal{K}^{(r)}_{1,2} \\ \mathcal{K}^{(r)}_{2,1} & \mathcal{K}^{(r)}_{2,2} \end{bmatrix} \\
&= \begin{bmatrix} L^{(r)}_{r,r} & 0 \\ L^{(r)}_{r+1+,r} & L^{(r)}_{r+1+,r+1+} \end{bmatrix} \\
&= \left[\begin{array}{c|ccc} L^{(r)}_{r,r} & 0 & \cdots & 0 \\ \hline L^{(r)}_{r+1,r} & L^{(r)}_{r+1,r+1} & \cdots & L^{(r)}_{r+1,q} \\ \vdots & \vdots & \ddots & \vdots \\ L^{(2)}_{q,r} & L^{(r)}_{q,r+1} & \cdots & L^{(r)}_{q,q} \end{array}\right],
\end{aligned}
$$

and

$$
A^{(r-1)}_{r+}\mathcal{K}^{(r)} = \begin{bmatrix} A^{(r-1)}_{r} & A^{(r-1)}_{r+1+} \end{bmatrix} \begin{bmatrix} \mathcal{K}^{(r)}_{1,1} & \mathcal{K}^{(r)}_{1,2} \\ \mathcal{K}^{(r)}_{2,1} & \mathcal{K}^{(r)}_{2,2} \end{bmatrix} = \begin{bmatrix} A^{(r)}_{r} & A^{(r)}_{r+1+} \end{bmatrix} = \left[\begin{array}{c|ccc} A^{(r)}_{r} & A^{(r)}_{r+1} & \cdots & A^{(r)}_{q} \end{array}\right].
$$

The augmented block Householder matrix

$$
\begin{aligned}
K^{(r)} &= \begin{bmatrix} I_{h_1+\cdots+h_{r-1}} & 0 \\ 0 & I_{h_r+\cdots+h_q} - 2V_r(V_r^*V_r)^\dagger V_r^* \end{bmatrix} \\
&= \begin{bmatrix} I_{h_1+\cdots+h_{r-1}} & 0 & 0 \\ 0 & I_{h_r+\cdots+h_q} - 2V_{r,1}^*(V_rV_r^*)^\dagger V_{r,1} & -2V_{r,1}^*(V_rV_r^*)^\dagger V_{r,2} \\ 0 & -2V_{r,2}^*(V_rV_r^*)^\dagger V_{r,1} & I_{h_r+\cdots+h_q} - 2V_{r,2}^*(V_rV_r^*)^\dagger V_{r,2} \end{bmatrix} \\
&= \begin{bmatrix} I_{h_1+\cdots+h_{r-1}} & 0 & 0 \\ 0 & \mathcal{K}^{(2)}_{1,1} & \mathcal{K}^{(2)}_{1,2} \\ 0 & \mathcal{K}^{(2)}_{2,1} & \mathcal{K}^{(2)}_{2,2} \end{bmatrix},
\end{aligned}
$$

simply implements the restricted reduction on the whole system. Indeed, we have

$$
\begin{aligned}
&L^{(r-1)}K^{(r)} \\
&= \left[\begin{array}{cccc|cc} L^{(1)}_{1,1} & 0 & \cdots & 0 & 0 & 0 \\ L^{(1)}_{2,1} & L^{(2)}_{2,2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L^{(1)}_{r-1,1} & L^{(2)}_{r-1,2} & \cdots & L^{(r-1)}_{r-1,r-1} & 0 & 0 \\ \hline L^{(1)}_{r,1} & L^{(2)}_{r,2} & \cdots & L^{(r-1)}_{r,r-1} & L^{(r-1)}_{r,r} & L^{(r-1)}_{r,r+1+} \\ L^{(1)}_{r+1+,1} & L^{(2)}_{r+1+1,2} & \cdots & L^{(r-1)}_{r+1+,r-1} & L^{(r-1)}_{r+1+,r} & L^{(r-1)}_{r+1+,r+1+} \end{array}\right] \left[\begin{array}{cccc|cc} I_{h_1} & 0 & \cdots & 0 & 0 & 0 \\ 0 & I_{h_2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I_{h_{r-1}} & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & \mathcal{K}^{(2)}_{1,1} & \mathcal{K}^{(2)}_{1,2} \\ 0 & 0 & \cdots & 0 & \mathcal{K}^{(2)}_{2,1} & \mathcal{K}^{(2)}_{,2} \end{array}\right]
\end{aligned}
$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

$$= \left[\begin{array}{cccc|cc} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L_{r-1,1}^{(1)} & L_{r-1,2}^{(2)} & \cdots & L_{r-1,r-1}^{(r-1)} & 0 & 0 \\ \hline L_{r,1}^{(1)} & L_{r,2}^{(2)} & \cdots & L_{r,r-1}^{(r-1)} & L_{r,r}^{(r)} & 0 \\ L_{r+1+,1}^{(1)} & L_{r+1+1,2}^{(2)} & \cdots & L_{r+1+,r-1}^{(r-1)} & L_{r+1+,r}^{(r)} & L_{r+1+,r+1+}^{(r)} \end{array}\right]$$

$$= \left[\begin{array}{ccccc|cccc} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ L_{r-1,1}^{(1)} & L_{r-1,2}^{(2)} & \cdots & L_{r-1,r-1}^{(r-1)} & 0 & 0 & 0 & \cdots & 0 \\ L_{r,1}^{(1)} & L_{r,2}^{(2)} & \cdots & L_{r,r-1}^{(r-1)} & L_{r,r}^{(r)} & 0 & 0 & \cdots & 0 \\ \hline L_{r+1,1}^{(1)} & L_{r+1,2}^{(2)} & \cdots & L_{r+1,r-1}^{(r-1)} & L_{r+1,r}^{(r)} & L_{r+1,r+1}^{(r)} & 0 & \cdots & 0 \\ L_{r+2,1}^{(1)} & L_{r+2,2}^{(2)} & \cdots & L_{r+2,r-1}^{(r-1)} & L_{r+2,r}^{(r)} & L_{r+2,r+1}^{(r)} & L_{r+2,r+2}^{(r)} & \cdots & L_{r+2,q}^{(r)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ L_{q,1}^{(1)} & L_{q,2}^{(2)} & \cdots & L_{q,r-1}^{(r-1)} & L_{q,r}^{(r)} & L_{q,r+1}^{(r)} & L_{q,r+2}^{(r)} & \cdots & L_{q,q}^{(r)} \end{array}\right] = L^{(r)},$$

and

$$A^{(r-1)}K^{(r)} = \left[\begin{array}{cccc|cc} A_1^{(1)} & A_2^{(2)} & \cdots & A_{r-1}^{(r-1)} & A_r^{(r-1)} & A_{r+1+}^{(r-1)} \end{array}\right] \left[\begin{array}{cccc|cc} I_{h_1} & 0 & \cdots & 0 & 0 & 0 \\ 0 & I_{h_2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I_{h_{r-1}} & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & \mathcal{K}_{1,1}^{(2)} & \mathcal{K}_{1,2}^{(2)} \\ 0 & 0 & \cdots & 0 & \mathcal{K}_{2,1}^{(2)} & \mathcal{K}_{,2}^{(2)} \end{array}\right]$$

$$= \left[\begin{array}{cccccc} A_1^{(1)} & A_2^{(2)} & \cdots & A_{r-1}^{(r-1)} & A_r^{(r)} & A_{r+1+}^{(r)} \end{array}\right].$$

This completes Stage r.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

Stage q. The process terminates at Stage q. We begin with

$$L^{(q-1)} = \left[\begin{array}{cccc|cc} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L_{q-1,1}^{(1)} & L_{q-1,2}^{(2)} & \cdots & L_{q-1,q-1}^{(q-1)} & 0 & 0 \\ \hline L_{q,1}^{(1)} & L_{q,2}^{(2)} & \cdots & L_{q,q-1}^{(q-1)} & L_{q,q}^{(q-1)} & L_{q,q+1}^{(q-1)} \end{array}\right].$$

The Householder transformation acts on the matrix

$$\left[\begin{array}{cc} L_{q,q}^{(q-1)} & L_{q,q+1}^{(q-1)} \end{array}\right],$$

to eliminate the term $L_{q,q+1}^{(q-1)}$. Thus, we finish with

$$L^{(q-1)}K^{(q)} = \left[\begin{array}{ccccc|c} L_{11}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ L_{21}^{(1)} & L_{22}^{(2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L_{q-1,1}^{(1)} & L_{q-1,2}^{(2)} & \cdots & L_{q-1,q-1}^{(q-1)} & 0 & 0 \\ L_{q,1}^{(1)} & L_{q,2}^{(2)} & \cdots & L_{q,q-1}^{(q-1)} & L_{q,q}^{(q)} & 0 \end{array}\right] = L^{(q)}.$$

If we define $Q^* = K^{(1)}K^{(2)}\cdots K^{(q)}$, then Q^* is a product of unitary matrices and hence is also unitary. If we define $L = L^{(q)}$, then we have shown that $CQ^* = L \iff C = LQ$. This is the desired LQ decomposition of C. It is convenient to write $L = [L_1 \mid 0]$, where $L_1 \in \mathbb{C}^{h\times h}$ is block lower triangular and $0 \in \mathbb{C}^{q\times s}$. Therefore, $LL^* = L_1L_1^*$. We also write

$$Q = \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right],$$

where $Q_1 \in \mathbb{C}^{h\times n}$ and $Q_2 \in \mathbb{C}^{s\times n}$. Now it follows that $C = LQ = L_1Q_1$ and so

$$YCC^* = AC^* \iff YLL^* = AQ^*L^* \iff YL_1L_1^* = AQ_1^*L_1^* \iff YL_1 = AQ_1^*.$$

Thus, we can solve the original equation $YCC^* = AC^*$ by solving the reduced equation $YL_1 = AQ_1$. The coefficient L_1 is block lower triangular and nonsingular. Hence, the reduced system can be solved by back substitution.

3.4. Notational summary for the HUTS algorithm. We have used the following notation to describe the HUTS algorithm.

- The initial matrix is $C \in \mathbb{C}^{h\times n}$, where $n = h + s$ and where $\operatorname{rank}(C) = h$. We define $L^{(0)} = C$ and write $L^{(0)} = [L_{i,j}^{(0)}]_{(i,j)=(1,1)}^{(q,q+1)}$ in block matrix form, where $L_{i,j}^{(0)} \in \mathbb{C}^{h_i\times h_j}$ for each $(i,j) \in [1,q]\times[1,q]$ and where $L_{i,q+1}^{(0)} \in \mathbb{C}^{h_i\times s}$ for each $i = 1,\dots,q$.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

- We write

$$L^{(0)} = \begin{bmatrix} L_{1,1}^{(0)} & L_{1,2+}^{(0)} \\ L_{2+,1}^{(0)} & L_{2+,2+}^{(0)} \end{bmatrix} \quad \text{and} \quad A^{(0)} = \begin{bmatrix} A_1^{(0)} & A_{2+}^{(1)} \end{bmatrix},$$

in terms of the consolidated partitions

$$L_{1,2+}^{(0)} = \begin{bmatrix} L_{1,2}^{(0)} & \cdots & L_{1,q}^{(0)} \end{bmatrix}, \quad L_{2+,1}^{(0)} = \begin{bmatrix} L_{2,1}^{(0)} \\ \vdots \\ L_{q,1}^{(0)} \end{bmatrix}, \quad \text{and} \quad L_{2+,2+}^{(0)} = \begin{bmatrix} L_{2,2}^{(0)} & \cdots & L_{2,q}^{(0)} \\ \vdots & \ddots & \vdots \\ L_{q,1}^{(0)} & \cdots & L_{q,q}^{(0)}n \end{bmatrix}.$$

- At stage r, we write

$$L^{(r-1)} = \left[\begin{array}{cccc|cc} L_{1,1}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ L_{r-1,1}^{(1)} & L_{r-1,2}^{(2)} & \cdots & L_{r-1,r-1}^{(r-1)} & 0 & 0 \\ \hline L_{r,1}^{(1)} & L_{r,2}^{(2)} & \cdots & L_{r,r-1}^{(r-1)} & L_{r,r}^{(r-1)} & L_{r,r+1+}^{(r-1)} \\ L_{r+1+,1}^{(1)} & L_{r+1+,2}^{(2)} & \cdots & L_{r+1+,r-1}^{(r-1)} & L_{r+1+,r}^{(r-1)} & L_{r+1+,r+1+}^{(r-1)} \end{array}\right],$$

and

$$A^{(r-1)} = \left[\begin{array}{cccc|cc} A_1^{(1)} & A_2^{(2)} & \cdots & A_{r-1}^{(r-1)} & A_r^{(r-1)} & A_{r+1+}^{(r-1)} \end{array}\right].$$

in terms of the consolidated partitions

$$L_{r+1+,j}^{(j)} = \begin{bmatrix} L_{r+1,j}^{(j)} \\ \vdots \\ L_{q,j}^{(j)} \end{bmatrix} (j = 1, \ldots, r-1), \quad L_{r,r+1+}^{(r-1)} = \begin{bmatrix} L_{r,r+1}^{(r-1)} & \cdots & L_{r,q}^{(r-1)} \end{bmatrix},$$

$$L_{r+1+,r-1}^{(r-1)} = \begin{bmatrix} L_{r+1,r}^{(r-1)} \\ \vdots \\ L_{q,r}^{(r-1)} \end{bmatrix}, \quad L_{r+1+,r+1+}^{(r-1)} = \begin{bmatrix} L_{r+1,r+1}^{(r-1)} & \cdots & L_{r+1,q}^{(r-1)} \\ \vdots & \ddots & \vdots \\ L_{q,r+1}^{(r-1)} & \cdots & L_{q,q}^{(r-1)} \end{bmatrix},$$

and

$$A_{r+1+}^{(r-1)} = \begin{bmatrix} A_{r+1}^{(r-1)} & \cdots & A_q^{(r-1)} \end{bmatrix}.$$

.

3.5. Application in a numerical example.

Example 2. *Let A and C be the matrices defined by*

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

$$A = \left[\begin{array}{cc|cc|ccc|c}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}\right], \quad C = \left[\begin{array}{cc|cc|ccc|c}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0
\end{array}\right],$$

where $m = 12$, $n = 8$, $h = 7$, and $s = 1$. We have set $q = 3$ and defined $\boldsymbol{h} = [h_1, h_2, h_3] = [2, 2, 3]$ and $s = 1$. We have written

$$A = \left[\begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array}\right] \quad \text{and} \quad C = \left[\begin{array}{cccc} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} \\ C_{3,1} & C_{3,2} & C_{3,3} & C_{3,4} \end{array}\right],$$

where $A_j \in \mathbb{C}^{m \times h_j}$ for $j \le 3$ and $A_{q+1} \in \mathbb{C}^{m \times s}$ and where $C_{i,j} \in \mathbb{C}^{h_i \times h_j}$ for $j \le q$ and $C_{i,q+1} \in \mathbb{C}^{h_i \times s}$.

Solution of $\boldsymbol{YCC^* = AC^*}$. *To prepare for Stage 1 define $L^{(0)} = C$ and $A^{(0)} = A$ and write*

$$L^{(0)} = \left[\begin{array}{c|c} L_{1,1}^{(0)} & L_{1,2+}^{(0)} \\ \hline L_{2+,1}^{(0)} & L_{2+,2+}^{(0)} \end{array}\right] \quad \text{and} \quad A^{(0)} = \left[\begin{array}{c|c} A_1^{(0)} & A_{2+}^{(0)} \end{array}\right],$$

where $L_{1,1}^{(0)} \in \mathbb{C}^{2\times 2}$ and $A_1^{(0)} \in \mathbb{C}^{12 \times 2}$. For Stage 1, we define the block Householder matrix

$$K^{(1)} = \left[\begin{array}{rr|rrrrrr}
0.5158 & -0.0817 & 0.5158 & -0.0817 & 0.4342 & 0.5158 & 0 & 0 \\
-0.0817 & 0.5975 & -0.0817 & 0.5975 & 0.5158 & -0.0817 & 0 & 0 \\ \hline
0.5158 & -0.0817 & 0.3773 & 0.3522 & -0.2705 & -0.6227 & 0 & 0 \\
-0.0817 & 0.5975 & 0.3522 & 0.0252 & -0.6227 & 0.3522 & 0 & 0 \\
0.4342 & 0.5158 & -0.2705 & -0.6227 & 0.1068 & -0.2705 & 0 & 0 \\
0.5158 & -0.0817 & -0.6227 & 0.3522 & -0.2705 & 0.3773 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000
\end{array}\right],$$

in order to eliminate $L_{1,2+}^{(0)}$. We calculate

$$L^{(0)}K^{(1)} = \left[\begin{array}{rr|rrrrrr}
1.9816 & 0.2708 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0 & 0 \\
0.2708 & 1.7107 & 0.0000 & -0.0000 & 0 & 0.0000 & 0 & 0 \\ \hline
1.3025 & 1.5474 & -0.2453 & 0.7043 & -0.5410 & -0.2453 & 1.0000 & 0 \\
0.3525 & 1.1133 & -0.3522 & -0.0252 & 0.6227 & -0.3522 & 1.0000 & 0 \\
1.4658 & 0.3525 & -0.3773 & -0.3522 & 0.2705 & 0.6227 & 1.0000 & 1.0000 \\
0.9500 & 0.4342 & -0.3270 & 1.3018 & -0.0252 & -0.3270 & 0 & 1.0000 \\
1.4658 & 0.3525 & -0.5158 & 0.0817 & -0.4342 & -0.5158 & 1.0000 & 0
\end{array}\right] = L^{(1)},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and

$$A^{(0)}K^{(1)} = \left[\begin{array}{rr|rrrrrr} 1.8183 & 1.4658 & 0.2705 & 0.6227 & -0.1068 & 0.2705 & 1.0000 & 1.0000 \\ 1.4658 & 0.3525 & 0.6227 & -0.3522 & 0.2705 & -0.3773 & 0 & 0 \\ 0.8683 & 1.0316 & 0.0252 & 1.3270 & -0.6478 & 0.0252 & 1.0000 & 1.0000 \\ 1.4658 & 0.3525 & -0.3773 & -0.3522 & 0.2705 & 0.6227 & 0 & 1.0000 \\ 1.9816 & 0.2708 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0 & 0 \\ 0.4342 & 0.5158 & -0.7043 & 0.9496 & 0.2453 & 0.2957 & 0 & 1.0000 \\ 0.8683 & 1.0316 & 0.0252 & 1.3270 & -0.6478 & 0.0252 & 1.0000 & 0 \\ 0.3525 & 1.1133 & 0.7863 & 0.5410 & 0.3273 & 0.7863 & 0 & 0 \\ 0.8683 & 1.0316 & 0.1637 & -0.1068 & 1.0568 & 0.1637 & 0 & 1.0000 \\ 0.5158 & -0.0817 & 0.5158 & -0.0817 & 0.4342 & 0.5158 & 1.0000 & 1.0000 \\ 0.8683 & 1.0316 & -0.9748 & 0.3270 & 0.3522 & 0.0252 & 1.0000 & 1.0000 \\ 0.8683 & 1.0316 & 1.1637 & 0.8932 & 0.0568 & 0.1637 & 1.0000 & 0 \end{array}\right] = A^{(1)}.$$

To prepare for Stage 2*, we write*

$$L^{(1)} = \left[\begin{array}{c|c|c} L^{(1)}_{1,1} & 0 & 0 \\ \hline L^{(1)}_{2,1} & L^{(1)}_{2,2} & L^{(1)}_{2,3+} \\ \hline L^{(1)}_{3+,1} & L^{(1)}_{3+,2} & L^{(1)}_{3+,3+} \end{array}\right] \quad \text{and} \quad A^{(1)} = \left[\begin{array}{c|c|c} A^{(1)}_1 & A^{(1)}_2 & A^{(1)}_{3+} \end{array}\right],$$

where $L^{(1)}_{1,1}, L^{(1)}_{2,2} \in \mathbb{C}^{2\times 2}$ *and* $A^{(1)}_1, A^{(1)}_2 \in \mathbb{C}^{12\times 2}$*. For Stage 2, we define the augmented block Householder matrix*

$$K^{(2)} = \left[\begin{array}{rr|rr|rrrr} 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0.2792 & -0.0320 & -0.4235 & 0.2792 & -0.8147 & 0 \\ 0 & 0 & -0.0320 & 0.5848 & -0.7361 & -0.0320 & 0.3377 & 0 \\ \hline 0 & 0 & -0.4235 & -0.7361 & -0.4924 & 0.0870 & 0.1696 & 0 \\ 0 & 0 & 0.2792 & -0.0320 & 0.0870 & 0.8871 & 0.3557 & 0 \\ 0 & 0 & -0.8147 & 0.3377 & 0.1696 & 0.3557 & -0.2587 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \end{array}\right],$$

in order to eliminate $L^{(1)}_{2,3+}$*. Now we calculate*

$$L^{(1)}K^{(2)} = \left[\begin{array}{rr|rr|rrrr} 1.9816 & 0.2708 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & 0 \\ 0.2708 & 1.7107 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0 \\ \hline 1.3025 & 1.5474 & -0.7451 & 1.1635 & 0.0000 & 0 & -0.0000 & 0 \\ 0.3525 & 1.1133 & -1.2742 & -0.1128 & 0.0000 & -0.0000 & 0 & 0 \\ \hline 1.4658 & 0.3525 & -0.8495 & -0.0752 & 0.5096 & 0.8375 & 0.1972 & 1.0000 \\ 0.9500 & 0.4342 & -0.2136 & 0.8008 & -0.8359 & -0.4252 & 0.5854 & 1.0000 \\ 1.4658 & 0.3525 & -0.9215 & 0.7381 & 0.4968 & -0.2862 & -0.0680 & 0 \end{array}\right] = L^{(2)},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and

$$A^{(1)}K^{(2)} = \left[\begin{array}{rr|rr|rrrr} 1.8183 & 1.4658 & -0.6383 & 0.7632 & -0.3272 & 0.6420 & -0.1907 & 1.0000 \\ 1.4658 & 0.3525 & -0.0348 & -0.4129 & -0.1705 & -0.1261 & -0.7146 & 0 \\ 0.8683 & 1.0316 & -0.5688 & 1.5890 & -0.4968 & 0.2862 & 0.0680 & 1.0000 \\ 1.4658 & 0.3525 & -0.0348 & -0.4129 & 0.3400 & 0.4818 & 0.4559 & 1.0000 \\ 1.9816 & 0.2708 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & 0 \\ 0.4342 & 0.5158 & -0.2484 & 0.3878 & -0.4959 & 0.0566 & 1.0413 & 1.0000 \\ 0.8683 & 1.0316 & -0.5688 & 1.5890 & -0.4968 & 0.2862 & 0.0680 & 0 \\ 0.3525 & 1.1133 & 0.2832 & 0.0251 & -0.8240 & 0.9282 & -0.1227 & 0 \\ 0.8683 & 1.0316 & -0.3527 & -0.8509 & -0.4968 & 0.2862 & 0.0680 & 1.0000 \\ 0.5158 & -0.0817 & -0.7079 & -0.0627 & -0.1576 & 0.9977 & -0.4494 & 1.0000 \\ 0.8683 & 1.0316 & -1.2394 & 0.3001 & 0.1705 & 0.1261 & 0.7146 & 1.0000 \\ 0.8683 & 1.0316 & -0.4968 & 0.7757 & -0.9944 & 0.8021 & -0.8373 & 0 \end{array}\right] = A^{(2)}.$$

To prepare for Stage 3 *we write*

$$L^{(2)} = \left[\begin{array}{c|c|c|c} L_{1,1}^{(1)} & 0 & 0 & 0 \\ \hline L_{2,1}^{(1)} & L_{2,2}^{(2)} & 0 & 0 \\ \hline L_{3,1}^{(1)} & L_{3,2}^{(2)} & L_{3,3}^{(2)} & L_{3,4}^{(2)} \end{array}\right] \quad \textit{and} \quad A^{(2)} = \left[\begin{array}{c|c|c|c} A_1^{(1)} & A_2^{(2)} & A_3^{(2)} & A_4^{(2)} \end{array}\right],$$

where $L_{2,2}^{(2)} \in \mathbb{C}V^{2\times 2}$, $L_{3,3}^{(2)} \in \mathbb{C}^{3\times 3}$, $A_2^{(2)} \in \mathbb{C}^{12\times 2}$, *and* $A_3^{(2)} \in \mathbb{C}^{12\times 3}$. *For Stage* 3, *we define the augmented block Householder matrix*

$$K^{(3)} = \left[\begin{array}{rr|rr|rrr|r} 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0.9764 & -0.0121 & -0.1210 & 0.1782 \\ 0 & 0 & 0 & 0 & -0.0121 & 0.9938 & -0.0623 & 0.0917 \\ 0 & 0 & 0 & 0 & -0.1210 & -0.0623 & 0.3779 & 0.9158 \\ 0 & 0 & 0 & 0 & 0.1782 & 0.0917 & 0.9158 & -0.3482 \end{array}\right],$$

in order to eliminate $L_{3,4}^{(2)}$. *We calculate*

$$L^{(2)}K^{(3)} = \left[\begin{array}{rr|rr|rrr|r} 1.9816 & 0.2708 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.2708 & 1.7107 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ \hline 1.3025 & 1.5474 & -0.7451 & 1.1635 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.3525 & 1.1133 & -1.2742 & -0.1128 & 0.0000 & -0.0000 & -0.0000 & 0.0000 \\ \hline 1.4658 & 0.3525 & -0.8495 & -0.0752 & 0.6418 & 0.9055 & 0.8764 & -0.0000 \\ 0.9500 & 0.4342 & -0.2136 & 0.8008 & -0.7037 & -0.3572 & 1.2647 & -0.0000 \\ 1.4658 & 0.3525 & -0.9215 & 0.7381 & 0.4968 & -0.2862 & -0.0680 & 0.0000 \end{array}\right] = L^{(3)},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

and

$$A^{(2)}K^{(3)} = \left[\begin{array}{rr|rr|rrr|r} 1.8183 & 1.4658 & -0.6383 & 0.7632 & -0.1260 & 0.7455 & 0.8433 & -0.5222 \\ 1.4658 & 0.3525 & -0.0348 & -0.4129 & -0.0784 & -0.0787 & -0.2416 & -0.6963 \\ 0.8683 & 1.0316 & -0.5688 & 1.5890 & -0.3186 & 0.3779 & 0.9838 & -0.3482 \\ 1.4658 & 0.3525 & -0.0348 & -0.4129 & 0.4492 & 0.5380 & 1.0169 & 0.1741 \\ 1.9816 & 0.2708 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.4342 & 0.5158 & -0.2484 & 0.3878 & -0.4327 & 0.0891 & 1.3658 & 0.5222 \\ 0.8683 & 1.0316 & -0.5688 & 1.5890 & -0.4968 & 0.2862 & 0.0680 & -0.0000 \\ 0.3525 & 1.1133 & 0.2832 & 0.0251 & -0.8010 & 0.9401 & -0.0045 & -0.1741 \\ 0.8683 & 1.0316 & -0.3527 & -0.8509 & -0.3186 & 0.3779 & 0.9838 & -0.3482 \\ 0.5158 & -0.0817 & -0.7079 & -0.0627 & 0.0666 & 1.1131 & 0.7029 & -0.6963 \\ 0.8683 & 1.0316 & -1.2394 & 0.3001 & 0.2566 & 0.1704 & 1.1573 & 0.3482 \\ 0.8683 & 1.0316 & -0.4968 & 0.7757 & -0.8794 & 0.8613 & -0.2460 & -0.8704 \end{array}\right] = A^{(3)}.$$

This completes the reduction. We write

$$L^{(3)} = \begin{bmatrix} L_{1,1}^{(1)} & 0 & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & 0 & 0 \\ L_{3,1}^{(1)} & L_{3,2}^{(2)} & L_{3,3}^{(3)} & 0 \end{bmatrix} \quad \textit{and} \quad A^{(3)} = \begin{bmatrix} A_1^{(1)} & A_2^{(2)} & A_3^{(3)} & A_4^{(3)} \end{bmatrix},$$

where $L_{2,2}^{(2)} \in \mathbb{C}^{2\times 2}$, $L_{3,3}^{(3)} \in \mathbb{C}^{3\times 3}$, $A_2^{(2)} \in \mathbb{C}^{12\times 2}$, *and* $A_3^{(3)} \in \mathbb{C}^{12\times 3}$. *We define the unitary matrix* $Q^* = K^{(1)}K^{(2)}K^{(3)}$ *and note that* $L = L^{(3)} = CQ^*$ *and* $A^{(3)} = AQ^*$. *Finally, we define the nonsingular block lower triangular matrix*

$$L_1 = \begin{bmatrix} L_{1,1}^{(1)} & 0 & 0 \\ L_{2,1}^{(1)} & L_{2,2}^{(2)} & 0 \\ L_{3,1}^{(1)} & L_{3,2}^{(2)} & L_{3,3}^{(3)} \end{bmatrix} \quad \textit{and} \quad Y = \begin{bmatrix} Y_1 & Y_2 & Y_3 \end{bmatrix},$$

and use back substitution to solve the equation $YL_1 = AQ_1^*$. *Thus, we obtain*

$$Y = \begin{bmatrix} Y_1 & Y_2 & Y_3 \end{bmatrix} = \left[\begin{array}{rr|rr|rrr} 0.2727 & -0.1818 & 1.0909 & -0.0000 & 0.6364 & 0.1818 & -0.8182 \\ 1.0303 & 0.0909 & -0.2121 & 0.3333 & -0.1515 & -0.0909 & -0.0909 \\ -0.4848 & -0.4545 & 1.3939 & -0.3333 & 0.4242 & 0.4545 & -0.5455 \\ 0.2424 & 0.7273 & -0.6970 & -0.3333 & 0.7879 & 0.2727 & 0.2727 \\ 1.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.2727 & 0.1818 & -0.0909 & 0.0000 & 0.3636 & 0.8182 & -0.1818 \\ -0.0000 & -1.0000 & 2.0000 & -0.0000 & 0.0000 & -0.0000 & -1.0000 \\ 0.7576 & -0.7273 & 1.6970 & 0.3333 & 0.2121 & -0.2727 & -2.2727 \\ 0.5152 & 0.5455 & -0.6061 & 0.6667 & 0.4242 & 0.4545 & -0.5455 \\ 0.0303 & -0.9091 & 0.7879 & 0.3333 & 0.8485 & -0.0909 & -1.0909 \\ -0.5152 & 0.4545 & -0.3939 & 0.3333 & 0.5758 & 0.5455 & 0.5455 \\ 0.7879 & -1.6364 & 2.4848 & 0.6667 & 0.0606 & -0.3636 & -2.3636 \end{array}\right].$$

4. Numerical trials.

We used the MATLAB program in Appendix A to solve the equation $YCC^* = AC^*$ for randomly generated matrices $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{(q\cdot h)\times n}$, where $m \geq n$ with $n = q\cdot h + s$, $h_j = h$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

for each $j = 1, \ldots, q$ and $s \in \mathbb{N} - 1$ for various values of $m, n, q, dh \in \mathbb{N}$. We use the following notation. The computation time is measured in seconds. For EBOS, the solution is denoted by Y_e, the computation time is t_e, and the error is $E_e = Y_e CC^* - AC^*$. The magnitude of the error is measured by the Frobenius norm $\epsilon_e = \|E_e\|_F$. For HUTS, the corresponding quantities are Y_h, t_h, $E_h = Y_h CC^* - AC^*$ and $\epsilon_h = \|E_h\|_F$. For the direct method, we write $Y_d = AC^\dagger$, t_d, $E_d = Y_d CC^* - AC^*$ and $\epsilon_d = \|E_d\|_F$. The Frobenius norm for the error matrix $E = [e_{i,j}] \in \mathbb{C}^{m \times n}$ is defined by

$$\epsilon = \|E\|_F = \sqrt{\sum_{i=1}^m \left[\sum_{j=1}^n e_{i,j}^2\right]}.$$

The Frobenius error is directly proportional to $\sqrt{mn}$. If $|e_{i,j}| = \delta$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$ then $\|E\|_F = \sqrt{mn\delta^2} = \sqrt{mn} \cdot \delta$.

All trials in this section were implemented in MATLAB on a MacBook Pro with 8 GB memory using the macOS Ventura 13.6 operating system. Throughout the numerical trials we observed various outliers in the tabulated results. These outliers are marked by an asterisk and shown in red on the tables.

4.1. Numerical trials for $m = n = q \cdot dh = 4000$ for various values of q, dh. We considered the case with $m = n = 4000$, $h_j = dh$ for all $j = 1, \ldots, q$ with $q \cdot dh = n$ for various values of q and dh to serve as an archetypal example. We calculated the computation times (t_e, t_h) and the magnitudes of the errors (ϵ_e, ϵ_h) for both EBOS and HUTS with $q \in \{2, 4, 5, 8, 10, 20\}$. The results are shown in Table 1. The computation times t_d and the errors ϵ_d for the direct method are also shown. The trials suggest that the best results are obtained when $(dh, q) = (1000, 4)$ and $(dh, q) = (800, 5)$ for EBOS and when $(dh, q) = (800, 5)$ for HUTS. HUTS is significantly faster than EBOS, which in turn is significantly faster than the direct method. On the other hand, the errors with EBOS and the direct method are more consistent and are usually less than the errors with HUTS. In each trial, all three methods are tested on the same data. The time advantage for both EBOS and HUTS is eventually lost as the number of subdivisions increases. Nevertheless, EBOS retains a consistently high level of accuracy.

4.2. Numerical trials for $m = n = q \cdot dh \in \{1000, 2000\}$ with $q = 4$. Our preliminary trials indicated that for $m = n = q \cdot dh \in \{1000, 2000\}$ the best results for both EBOS and HUTS are obtained with $(dh, q) = (250, 4)$ and $(dh, q) = (500, 4)$. The results are shown in Table 2.

For $m = n = 1000$, $dh = 250$, and $q = 4$, there is no time advantage using either EBOS or HUTS. All methods gave good results but the errors ϵ_e for EBOS and ϵ_d for the direct method were generally one order of magnitude smaller than the error ϵ_h for HUTS.

For $m = n = 2000$, $dh = 500$, and $q = 4$, the computation times t_e for EBOS and t_h for HUTS were significantly less than the computation time t_d for the direct method with $(\sum t_e)/(\sum t_d) \approx 0.89$ and $(\sum t_h)/(\sum t_d) \approx 0.83$. The errors ϵ_e for EBOS and ϵ_d for the direct method were generally two orders of magnitude smaller than the error ϵ_h for HUTS. Nevertheless, ϵ_h is small with the mean elemental error given approximately by $\delta_h = \epsilon_h/\sqrt{mn} \sim \mathcal{O}(10^{-6})$.

4.3. Numerical trials for $m = n = q \cdot dh \in \{3000, 4000, 5000, 6000\}$ with $q = 5$. Our preliminary trials indicated that for $m = n = q \cdot dh \in \{3000, 4000, 5000, 6000\}$ very good results for both EBOS and HUTS would be obtained with $q = 5$. The results are shown in Table 3.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

Table 1
Consecutive trials with $m = n = q \cdot dh = 4000$ *and* $q \in \{2, 4, 5, 8, 10, 20\}$.

m, n	dh	q	T	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
4000	2000	2	T_1	14.9219	16.6208	15.6737	1.8014×10^{-3}	$1.3418 \times 10^{+1}$	4.5840×10^{-4}
			T_2	15.0969	16.6104	16.0404	2.0682×10^{-3}	9.4605×10^{-1}	1.8473×10^{-3}
			T_3	15.0535	16.7200	16.7682	1.6539×10^{-3}	$1.6115 \times 10^{+1}$	5.7340×10^{-4}
4000	1000	4	T_1	12.6784	10.9434	16.6804	1.4223×10^{-3}	1.7789×10^{-2}	5.4461×10^{-3}
			T_2	12.8582	10.8932	15.8880	2.3415×10^{-3}	$*1.4938 \times 10^{+6}$	7.0562×10^{-3}
			T_3	13.4297	12.7373	17.0134	8.5362×10^{-4}	7.2841×10^{-3}	6.4894×10^{-4}
4000	800	5	T_1	13.3307	11.4244	17.8337	4.3119×10^{-4}	5.8716×10^{-3}	6.6309×10^{-4}
			T_2	13.2501	10.6771	15.9371	8.2113×10^{-3}	1.2639×10^{-1}	8.3219×10^{-4}
			T_3	12.4822	10.5275	16.0372	5.1980×10^{-4}	3.2762×10^{-3}	3.8119×10^{-4}
4000	500	8	T_1	14.1575	11.6662	17.1610	2.8201×10^{-4}	4.2373×10^{-3}	3.8649×10^{-4}
			T_2	14.4988	11.6386	16.1954	2.9991×10^{-4}	2.3255×10^{-2}	5.2592×10^{-4}
			T_3	14.0933	11.5634	16.6658	5.8226×10^{-4}	1.1256×10^{-1}	1.4159×10^{-3}
4000	400	10	T_1	15.1485	12.8771	16.4236	3.8681×10^{-4}	2.8836×10^{-3}	4.1385×10^{-4}
			T_2	15.2254	13.0421	16.3409	2.4063×10^{-4}	1.6611×10^{-2}	7.7839×10^{-4}
			T_3	15.2524	12.7326	16.7839	3.2256×10^{-4}	2.5529×10^{-2}	4.6023×10^{-4}
4000	200	20	T_1	24.7386	21.6125	16.3273	1.1901×10^{-4}	1.0138×10^{-2}	2.7609×10^{-3}
			T_2	23.9537	20.8439	16.3725	5.1631×10^{-4}	7.0395×10^{-2}	$*1.7653 \times 10^{-2}$
			T_3	23.4003	20.8876	16.1717	2.6385×10^{-4}	2.3850×10^{-2}	1.4256×10^{-3}

Table 2
Consecutive trials for $m = n = q \cdot dh \in \{1000, 2000\}$ *with* $q = 4$.

m, n	dh	q	T	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
1000	250	4	T_1	0.3875	0.3306	0.3340	3.8451×10^{-6}	2.3753×10^{-5}	2.1145×10^{-6}
			T_2	0.3556	0.3469	0.3677	3.0791×10^{-6}	1.4452×10^{-5}	3.3645×10^{-6}
			T_3	0.3578	0.3572	0.3750	2.8861×10^{-6}	1.5908×10^{-4}	7.3122×10^{-6}
			T_4	*0.5162	0.3302	0.3799	3.1329×10^{-6}	2.9772×10^{-4}	3.4782×10^{-6}
			T_5	0.3530	0.3412	0.3283	2.6437×10^{-6}	9.1305×10^{-6}	2.6329×10^{-8}
2000	500	4	T_1	1.8879	1.6884	2.0576	1.4550×10^{-4}	5.3505×10^{-2}	5.4339×10^{-4}
			T_2	1.8205	1.7264	2.1508	7.1023×10^{-5}	1.3238×10^{-3}	3.2106×10^{-5}
			T_3	1.8608	1.7317	2.1688	6.5977×10^{-5}	7.1926×10^{-3}	4.3326×10^{-5}
			T_4	1.8532	1.7297	2.0994	5.4622×10^{-5}	1.2789×10^{-2}	4.1029×10^{-5}
			T_5	1.9076	1.7053	2.1872	6.7353×10^{-5}	4.0117×10^{-4}	1.6517×10^{-4}

For $m = n = 3000$, $dh = 600$, and $q = 5$, the computation times t_e for EBOS and t_h for HUTS were significantly less than the computation time t_d for the direct method with $t_e/t_d \approx 0.87$ and $t_h/t_d \approx 0.72$. The errors ϵ_e for EBOS and ϵ_d for the direct method were generally between one and two orders of magnitude smaller than the error ϵ_h for HUTS. The mean elemental error for HUTS is given approximately by $\delta_h = \epsilon_h/\sqrt{mn} \sim \mathcal{O}(10^{-6})$. For $m = n = 4000$, $dh = 800$, and $q = 5$, the computation times t_e for EBOS and t_h for HUTS were significantly less than the computation time t_d for the direct method with $t_e/t_d \approx 0.78$ and $t_h/t_d \approx 0.67$. The errors ϵ_e for EBOS and ϵ_d for the direct method were at least two orders of magnitude

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

TABLE 3
Consecutive trials for $m = n = q \cdot dh \in \{3000, 4000, 5000, 6000\}$ *with* $q = 5$.

m, n	dh	q	T	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
3000	600	5	T_1	6.2888	4.8238	6.7366	2.2003×10^{-4}	3.8048×10^{-2}	1.5591×10^{-3}
			T_2	5.9592	4.8059	7.0946	2.9970×10^{-4}	2.8396×10^{-3}	2.6302×10^{-4}
			T_3	6.2034	4.8863	7.0921	1.9156×10^{-4}	5.2783×10^{-3}	1.5707×10^{-4}
			T_4	5.9266	4.8515	7.0850	1.6384×10^{-4}	6.2948×10^{-4}	1.3912×10^{-4}
			T_5	5.9408	4.8531	6.9766	1.5699×10^{-4}	1.0009×10^{-3}	2.1075×10^{-4}
4000	800	5	T_1	12.8454	11.2084	16.3007	1.1281×10^{-3}	1.3753×10^{-1}	5.2947×10^{-3}
			T_2	12.5709	10.6866	16.3938	8.9676×10^{-4}	2.5670×10^{-2}	4.7473×10^{-3}
			T_3	12.5781	10.6506	16.2745	4.2713×10^{-4}	1.1485×10^{-1}	3.4248×10^{-4}
			T_4	12.7755	10.7250	16.1110	4.9351×10^{-4}	2.1498×10^{-2}	9.1718×10^{-4}
			T_5	12.7460	10.6425	16.2819	4.2535×10^{-4}	$*1.1593 \times 10^{+2}$	1.0613×10^{-3}
5000	1000	5	T_1	*34.2219	24.6636	37.6202	1.2830×10^{-3}	6.9198×10^{-2}	1.1450×10^{-3}
			T_2	24.4187	21.3685	37.3421	1.2884×10^{-3}	1.1735×10^{-2}	1.7134×10^{-3}
			T_3	24.4205	21.4141	37.3095	1.1786×10^{-3}	6.4645×10^{-3}	4.5687×10^{-3}
			T_4	24.8651	21.1045	37.7728	1.0371×10^{-3}	4.5363×10^{-1}	9.5277×10^{-4}
			T_5	25.0451	20.9566	39.6188	1.0785×10^{-3}	5.6042×10^{-3}	1.7076×10^{-3}
6000	1200	5	T_1	41.2768	35.4972	62.2803	2.0329×10^{-3}	$*5.4629 \times 10^{+2}$	2.3146×10^{-3}
			T_2	41.7664	36.9280	62.6430	4.0189×10^{-3}	9.8147×10^{-1}	2.8032×10^{-2}
			T_3	*43.6625	36.4559	64.4931	2.2714×10^{-3}	9.8728×10^{0}	1.3405×10^{-2}
			T_4	40.7285	35.4109	62.3772	1.9970×10^{-3}	1.1805×10^{-1}	4.1214×10^{-3}
			T_5	41.0228	35.6987	62.0906	2.0021×10^{-3}	6.4781×10^{-3}	3.2094×10^{-3}

smaller than the error ϵ_h for HUTS. If we disregard one outlying result, the mean elemental error for HUTS is given approximately by $\delta_h = \epsilon_h/\sqrt{mn} \sim \mathcal{O}(10^{-5})$. For $m = n = 5000$, $dh = 1000$, and $q = 5$, the computation times t_e for EBOS (if we disregard one clear outlier) and t_h for HUTS were again significantly less than the computation time t_d for the direct method with $t_e/t_d \approx 0.67$ and $t_h/t_d \approx 0.58$. The errors ϵ_e for EBOS and ϵ_d for the direct method were approximately two orders of magnitude smaller than the error ϵ_h for HUTS. The mean elemental error for HUTS is given approximately by $\delta_h = \epsilon_h/\sqrt{mn} \sim \mathcal{O}(10^{-4})$. For $m = n = 6000$, $dh = 1200$, and $q = 5$, the computation times t_e for EBOS (if we disregard one outlier) and t_h for HUTS were again significantly less than the computation time t_d for the direct method with $t_e/t_d \approx 0.66$ and $t_h/t_d \approx 0.56$. The errors ϵ_e for EBOS and ϵ_d for the direct method were approximately two orders of magnitude smaller than the error ϵ_h for HUTS. The mean elemental error for HUTS is given approximately by $\delta_h = \epsilon_h/\sqrt{mn} \sim \mathcal{O}(10^{-3})$.

4.4. Numerical trials for $m = 6000$, $n = 5000$, $q \cdot dh \le n$, and $q = 5$. A key part of the program in Appendix A is that the random matrix A is generated in two parts. We use the MATLAB program `rand` to generate random matrices $A_L \in \mathbb{C}^{m \times (q \cdot dh)}$ and $A_R \in \mathbb{C}^{(q \cdot dh) \times n}$, where $m, n \ge q \cdot dh$. This ensures that the product matrix $A = A_L A_R \in \mathbb{C}^{m \times n}$ has $\text{rank}(A) \le q \cdot dh$. We tested EBOS and HUTS in an archetypal trial with $m = 6000 > n = 5000$ and $dh \in \{800, 900, 1000\}$ and $q = 5$. We conducted 3 consecutive trials for each value of dh. The results are shown in Table 4.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

Table 4
Consecutive trials for $m = 6000$, $n = 5000$, $dh \in \{800, 900, 1000\}$ and $q = 5$.

$(6000, 5000, 800, 5)$	T_1	18.7631	21.9193	21.4343	1.2575×10^{-3}	2.6710×10^{-3}	2.7311×10^{-5}
	T_2	18.6471	21.1824	21.5457	6.3838×10^{-4}	2.1987×10^{-1}	3.7990×10^{-5}
	T_3	18.3260	21.0940	21.3468	7.5598×10^{-4}	1.9105×10^{-2}	4.3829×10^{-5}
$(6000, 5000, 900, 5)$	T_1	22.0430	22.6164	26.3210	1.1723×10^{-3}	1.2624×10^{-4}	4.9936×10^{-5}
	T_2	21.7071	21.6511	26.4148	9.1179×10^{-4}	1.5268×10^{-3}	4.8308×10^{-5}
	T_3	23.0181	22.3049	26.1878	8.4917×10^{-4}	9.6197×10^{-4}	6.6867×10^{-5}
$(6000, 5000, 1000, 5)$	T_1	25.5931	23.3866	38.3929	3.3033×10^{-3}	4.9060×10^{-2}	1.2893×10^{-3}
	T_2	25.9993	22.7892	38.6663	1.6026×10^{-3}	4.2558×10^{-1}	4.5789×10^{-3}
	T_3	25.6197	22.9583	38.6963	1.1214×10^{-3}	7.8074×10^{-2}	1.4596×10^{-3}

Table 5
Consecutive trials for $m = 6000$, $n = 5000$, $dh \in \{900, 910, 920, \dots, 1000\}$ and $q = 5$.

(m, n, dh, q)	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
$(6000, 5000, 900, 5)$	22.4888	22.7782	26.2713	8.9842×10^{-4}	1.2625×10^{-3}	4.8031×10^{-5}
$(6000, 5000, 910, 5)$	23.3992	24.1901	27.1170	9.3715×10^{-4}	1.9655×10^{-3}	4.5275×10^{-5}
$(6000, 5000, 920, 5)$	23.6394	23.8896	27.9431	9.5021×10^{-4}	8.8280×10^{-3}	5.3528×10^{-5}
$(6000, 5000, 930, 5)$	23.6468	23.4833	28.1987	1.1806×10^{-3}	3.8852×10^{-4}	5.2517×10^{-5}
$(6000, 5000, 940, 5)$	23.7526	23.4271	28.6199	1.1153×10^{-3}	3.6608×10^{-3}	1.0397×10^{-4}
$(6000, 5000, 950, 5)$	24.6170	24.8775	29.6094	9.6616×10^{-4}	2.4365×10^{-3}	6.8875×10^{-5}
$(6000, 5000, 960, 5)$	25.1160	24.2718	37.1884	1.0992×10^{-3}	$*6.8731 \times 10^{0}$	8.2817×10^{-5}
$(6000, 5000, 970, 5)$	24.6679	24.6454	37.1702	1.3197×10^{-3}	9.9469×10^{-2}	1.1485×10^{-4}
$(6000, 5000, 980, 5)$	25.3903	24.9435	38.1266	2.1800×10^{-3}	9.8414×10^{-4}	1.1338×10^{-4}
$(6000, 5000, 990, 5)$	25.5006	25.9452	39.0390	1.3751×10^{-3}	$*4.4655 \times 10^{-4}$	1.5739×10^{-4}
$(6000, 5000, 1000, 5)$	25.8031	24.0052	39.0027	1.1712×10^{-3}	2.4387×10^{-1}	4.3082×10^{-3}

When $q \cdot dh \in \{3500, 4000\}$, there is no significant time advantage for EBOS and HUTS over the direct method. When $q \cdot dh = 4500$, there is a moderate time advantage for EBOS and HUTS with $(\sum t_e)/(\sum t_d) \approx (\sum t_h)/(\sum t_d) \approx 0.84$. When $q \cdot dh = 5000$, there is a much more substantial time advantage for EBOS and HUTS with $(\sum t_e)/(\sum t_d) \approx 0.66$ and $(\sum t_h)/(\sum t_d) \approx 0.59$. As $q \cdot dh \uparrow n$, the direct computation time rises significantly. The error with the direct method is consistently lower with $\epsilon_d \sim \mathcal{O}(10^{-5}$ than the errors $\epsilon_e \sim \mathcal{O}(10^{-4}$ and $\epsilon_h \sim \mathcal{O}(10^{-3})$ when $q \cdot dh \in \{3500, 4000, 4500\}$ but when $q \cdot dh = 5000 = n$ the error with the direct method increases dramatically to $\epsilon_d \sim \mathcal{O}(10^{-3})$, while the errors ϵ_e and ϵ_h remain at a similar level.

We conducted a more detailed trial for $m = 6000$, $n = 5000$, and $q = 5$ with $dh \in \{900, 910, 920, \dots, 1000\}$ with one trial for each value of dh. Although the values obtained are all random values, we have checked that the single trial described here is a true representation of what one could expect if the trial were repeated. The results of the trial are shown in Table 5.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

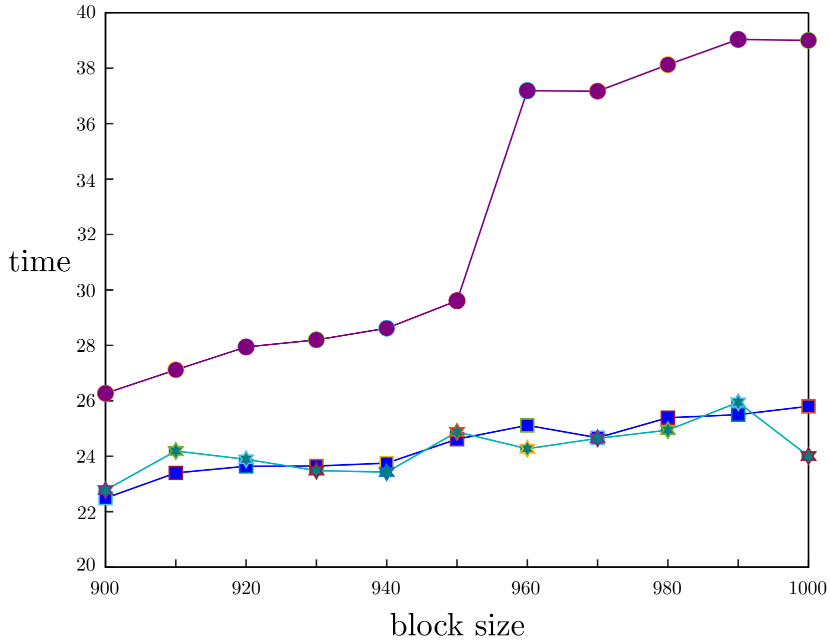

FIG. 1. *Graphs of t_e (squares), t_h (hexagrams), and t_d (circles) against $dh \in \{900, 910, \ldots, 1000\}$ when $m = 6000$, $n = 5000$ and $q = 5$. Note the consistent gradual increase in t_e and t_h as dh increases and the rapid increase in t_d when $dh \in [950, 960]$.*

We have used the graphs (dh, t_e), (dh, t_h), and (dh, t_d) of the computation times and the graphs $(dh, \log_e \epsilon_e)$, $(dh, \log_e \epsilon_h)$, and $(dh, \log_e \epsilon_d)$ of the errors to illustrate the results in Table 5.

The graphs (dh, t_e) and (dh, t_h) for the computation times against the block size in Fig. 1 show that t_e and t_h are close together for $dh \leq 990$ and $q \cdot dh < n$. The times increase steadily and consistently with some limited random variation for $dh \leq 990$. For $dh = 1000$ and $q \cdot dh = n$, the value of t_h drops significantly below the value for t_e with $t_e(1000) \approx 25.80$ and $t_h(1000) \approx 24.01$. Thus, $t_h/t_e \approx 0.92$. The decrease in t_h is not entirely due to random variation. If we look back at Table 4 for $m = 6000$, $n = 5000$, $dh = 1000$, and $q = 5$, we see that $(\sum t_h)/(\sum t_e) \approx 0.9$. For all of the trials with $m = n = q \cdot dh \geq 3000$, we have observed that t_h is consistently less than t_e. The graph (dh, t_d) is quite different. The value of t_d increases steadily and consistently with limited random variation for $dh \leq 950$. The value of t_d increases rapidly from $t_d(950) \approx 29.61$ to $t_d(960) \approx 37.19$. The value of t_e again increases steadily and consistently with some limited random variation as dh increases from $t_d(960) \approx 37.19$ to $t_d(1000) \approx 39.00$.

The graphs $(dh, \log_e \epsilon_e)$ and $(dh, \log_e \epsilon_h)$ in Fig. 2 of the natural logarithms of the errors against the block sizes show that the average values of ϵ_e and ϵ_h are an order of magnitude larger than ϵ_d for $dh \leq 990$. In these trials, $q \cdot dh < n$. The graph shows that $\log_e \epsilon_e \in [-7.02, -6.12]$ is reasonably steady but that $\log_e \epsilon_h \in [-7.86, 1.93]$ is quite volatile. The graph $(dh, \log_e \epsilon_d)$ is reasonably steady for $dh \in [900, 990]$ with $\log_e \epsilon_d \in [-10.01, -8.75]$. The value $\log_e \epsilon_d$ then increases significantly to $\log_e e_d(1000) \approx -5.45 \iff e_d(1000) \approx 4.3082 \times 10^{-3}$. This increase is not due to random variation. If we look back to Table 4, we can see that for $m = 6000$, $n = 5000$, $dh = 1000$, and $q = 5$, the average of the three values for $\epsilon_d(1000)$ is $\overline{\epsilon}_d = (\sum \epsilon_d)/3 \approx 2.43 \times 10^{-3} \iff \log_e(\overline{\epsilon}_d) \approx -6.02$.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

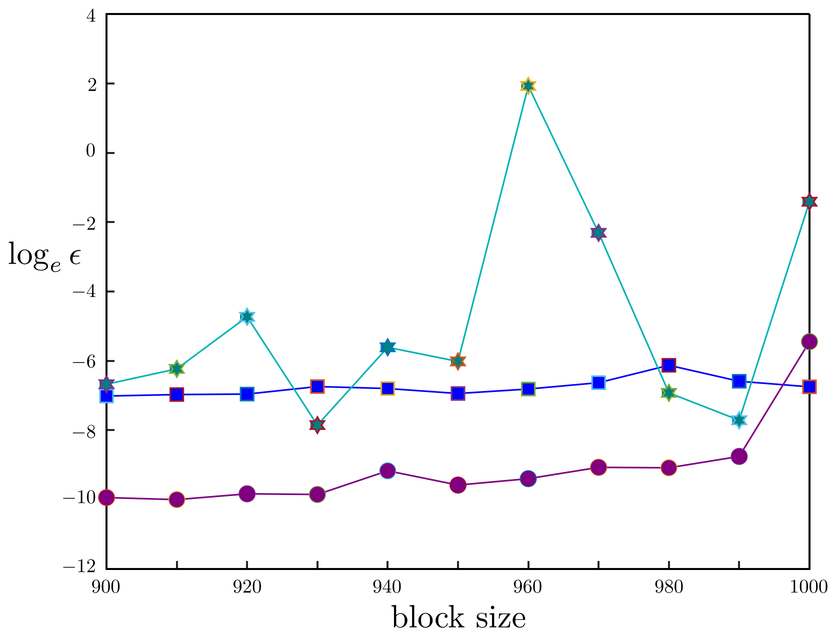

FIG. 2. *Graphs of* $\log_e \epsilon_e$ *(squares),* $\log_e \epsilon_h$ *(hexagrams), and* $\log_e \epsilon_d$ *(circles) against dh when* $m = 6000$*,* $n = 5000$*,* $dh \in \{900, 910, \ldots, 1000\}$*, and* $q = 5$*. Note the consistency of* $\log_e \epsilon_e$*, the volatility of* $\log_e \epsilon_h$*, and the rapid increase in* $\log_e \epsilon_d$ *for* $dh \in [990, 1000]$.

5. The errors and key matrix condition numbers in the HUTS trials. We conducted further trials with HUTS to investigate whether the observed volatility in the error ϵ_h could be linked to random variations in the condition numbers of the diagonal blocks in the matrix L_1. Errors in a particular block Y_j of the solution linked to the inversion of a badly conditioned block matrix $L_{1,j,j}$could propagated back to another block of the solution *via* the back substitution process used by HUTS. EBOS does not propagate errors of this type because the different blocks in the solution are found by solving different independent block subsystems. We used 10 consecutive trials with $m = n = q \cdot dh = 4000$ with $dh = 800$ and $q = 5$ to investigate a possible link between variation in ϵ_h and variation in the condition numbers $c_{j,j}$ of block $L_{1,j,j}$ for each $j = 1, \ldots, q$. We found a remarkably consistent structure for the values of the condition numbers with $c_{1,1} \approx 88$, $c_{2,2} \approx 3.15$, $c_{3,3} \approx 3.70$, and $c_{4,4} \approx 5.70$ with the only substantial variation in the final condition number $c_{5,5} \in [1173, 118765]$. We observed only a loose correlation between the final condition number and the errors. Our results are shown in Table 6.

The three largest condition numbers for the final block are all associated with larger than average values of both ϵ_h and ϵ_d in trials T_2, T_9, and T_{10}. The largest value $\epsilon_h \approx 6.9891 \times 10^{+5}$ in trial T_{12} is an extreme outlier, but is associated with a condition number $c_{5,5} \approx 4591.81$ that is certainly not extreme. It is true that the two smallest values for ϵ_h in trials T_3 and T_{11} are associated with the two smallest values for $c_{5,5}$. One wonders whether partial block pivoting could be used in the HUTS algorithm to ameliorate the volatility of ϵ_h and potentially decrease the errors. However, the most appropriate criterion for pivoting is not clear. Any such modification would increase the computation time.

6. Supplementary numerical trials. When m, n are small, there is no time advantage for EBOS and HUTS over the direct method. Nevertheless as part of our due diligence, we have tested both schemes

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

TABLE 6
Errors for HUTS and the direct method with condition numbers for the diagonal blocks of L_1 in 15 consecutive trials with $m = n = q \cdot dh = 4000$, $dh = 800$, and $q = 5$.

T	ϵ_h	ϵ_d	$c_{1,1}$	$c_{2,2}$	$c_{3,3}$	$c_{4,4}$	$c_{5,5}$
T_1	3.2538×10^{-2}	6.5448×10^{-4}	88.54	3.16	3.74	5.76	1758.72
T_2	9.0021×10^{0}	2.3163×10^{-2}	88.03	3.10	3.63	5.65	*64593.62
T_3	7.0003×10^{-3}	4.4589×10^{-4}	87.74	3.12	3.69	5.74	1446.41
T_4	1.1601×10^{-2}	8.8015×10^{-4}	85.41	3.15	3.67	5.69	2102.91
T_5	5.0073×10^{-1}	8.4695×10^{-4}	87.75	3.11	3.66	5.71	2373.01
T_6	1.9614×10^{-1}	2.9232×10^{-3}	87.98	3.23	3.68	5.84	9667.62
T_7	5.6054×10^{-2}	6.5719×10^{-4}	88.87	3.14	3.67	5.77	1550.85
T_8	1.1323×10^{-1}	7.4458×10^{-4}	88.74	3.13	3.66	5.71	1871.93
T_9	3.3587×10^{-1}	3.1191×10^{-2}	87.35	3.11	3.66	5.73	*118764.87
T_{10}	2.8543×10^{0}	1.5183×10^{-2}	87.01	3.11	3.64	5.80	*53297.97
T_{11}	7.4625×10^{-3}	4.8470×10^{-4}	88.49	3.16	3.77	5.82	1173.30
T_{12}	*$6.9891 \times 10^{+5}$	2.5876×10^{-3}	88.37	3.13	3.67	5.67	4591.81
T_{13}	1.3028×10^{0}	8.9172×10^{-4}	88.12	3.16	3.68	5.64	2469.25
T_{14}	3.8645×10^{-1}	1.0066×10^{-3}	88.58	3.16	3.69	5.76	3406.70
T_{15}	3.2569×10^{-2}	7.0183×10^{-4}	88.06	3.15	3.70	5.70	2393.96

in several potentially scenarios with smaller matrices. In particular, we considered some cases where the number of partitions was large and the block sizes were very small. All calculations in Tables 7, 8 and 9 were performed in a MATLAB environment on a MacBook Pro with 8 GB memory using the macOS Ventura 13.6 operating system.

The results for $m = n = 100$ with $(dh, q) \in \{(50, 2), (25, 4), (10, 10), (5, 20)\}$ are shown in Table 7. EBOS and HUTS were both comparable in accuracy to the direct method with $\epsilon = \|YCC^* - AC^*\|_F \sim \mathcal{O}(10^{-9})$ in all cases. The direct method was faster than HUTS which, in turn, was much faster than EBOS but all computation times were less than 0.05 seconds.

The results for $m = n = 100$ with $(dh, q) \in \{(50, 2), (25, 4), (10, 10), (5, 20)\}$ are shown in Table 7. EBOS and HUTS were both comparable in accuracy to the direct method with $\epsilon = \|YCC^* - AC^*\|_F \sim \mathcal{O}(10^{-9})$ in all cases. The direct method was faster than HUTS which, in turn, was much faster than EBOS but all computation times were less than 0.05 seconds

7. Notes on the numerical calculations. Similar calculations were also performed in an Octave environment on a Dell AS3760:2003 computer. GNU Octave (https://ftp.gnu.org/gnu/octave) is a scientific programming language primarily designed for numerical computation and is mostly compatible with MATLAB. Octave is free software under the terms of the GNU General Public License and uses less memory than MATLAB. Consequently, the execution times in Octave are generally much slower. In particular, the direct calculation $Y_d = AC^\dagger$ becomes very slow for large matrices. For instance, when $m = n = q \cdot dh = 2000$, the average execution times in Octave on the Dell computer were $t_e \approx 16$, $t_h \approx 17$, and $t_d \approx 126$ seconds,

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

TABLE 7
Consecutive trials for $m = n = 100$ *with* $(dh, q) \in \{(50, 2), (25, 4), (10, 10), (5, 20)\}$.

m, n	dh	q	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
100	50	2	0.0123	0.0061	0.0063	1.3851×10^{-9}	1.1629×10^{-8}	9.7496×10^{-10}
			0.0121	0.0060	0.0087	1.6485×10^{-9}	3.8828×10^{-8}	1.8187×10^{-9}
			0.0156	0.0060	0.0098	1.4676×10^{-9}	1.1187×10^{-8}	3.7638×10^{-10}
100	25	4	0.0190	0.0063	0.0080	3.2145×10^{-10}	2.8462×10^{-9}	6.2038×10^{-10}
			0.0177	0.0125	0.0081	8.3817×10^{-10}	5.8395×10^{-9}	3.0996×10^{-10}
			0.0219	0.0141	0.0083	3.4399×10^{-10}	1.2070×10^{-9}	4.8449×10^{-10}
100	10	10	0.0554	0.0255	0.0058	8.5847×10^{-10}	1.0194×10^{-9}	1.3039×10^{-8}
			0.0396	0.0125	0.0085	5.0372×10^{-10}	1.0379×10^{-9}	3.7481×10^{-10}
			0.0386	0.0140	0.0097	1.0102×10^{-9}	1.1149×10^{-8}	7.1089×10^{-10}
100	5	20	0.0394	0.0144	0.0075	1.7614×10^{-10}	9.4024×10^{-11}	5.5858×10^{-10}
			0.0419	0.0140	0.0086	3.2427×10^{-10}	4.8774×10^{-9}	3.3769×10^{-9}
			0.0453	0.0255	0.0086	5.0857×10^{-10}	1.4437×10^{-10}	1.2129×10^{-9}

TABLE 8
Consecutive trials for $m = n = 200$ *with* $(dh, q) \in \{(100, 2), (50, 4), (20, 10), (10, 20)\}$.

m, n	dh	q	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
200	50	4	0.0342	0.0255	0.0133	5.1286×10^{-9}	7.7471×10^{-6}	7.9459×10^{-9}
			0.0306	0.0197	0.0131	1.2932×10^{-8}	2.5699×10^{-7}	3.2265×10^{-8}
			0.0335	0.0239	0.0170	4.7659×10^{-9}	1.6653×10^{-6}	7.7290×10^{-9}
200	20	10	0.0434	0.0312	0.0511	3.5909×10^{-9}	7.8988×10^{-10}	7.5803×10^{-9}
			0.0390	0.0225	0.0143	3.4228×10^{-8}	3.8764×10^{-5}	1.0778×10^{-7}
			0.0391	0.0215	0.0153	6.1297×10^{-9}	3.6979×10^{-7}	1.1479×10^{-7}
200	10	20	0.0901	0.0320	0.0104	3.0476×10^{-8}	8.5766×10^{-7}	1.7911×10^{-7}
			0.0563	0.0307	0.0137	2.4435×10^{-9}	2.9275×10^{-8}	9.0102×10^{-8}
			0.0702	0.0518	0.0151	1.4877×10^{-9}	9.1476×10^{-8}	2.1703×10^{-8}
200	5	40	0.0950	0.0501	0.0123	3.2031×10^{-9}	8.5266×10^{-9}	2.2631×10^{-8}
			0.0852	0.0482	0.0122	8.3473×10^{-10}	2.5522×10^{-9}	1.4366×10^{-8}
			0.0836	0.0496	0.0210	2.1176×10^{-9}	1.7537×10^{-9}	7.1464×10^{-9}

whereas the average execution times in MATLAB on the MacBook Pro for the same parameters were $t_e \approx 1.86$, $t_h \approx 1.71$, and $t_d \approx 0.35$ seconds. For smaller problems with $m, n \leq 1000$, both EBOS and HUTS performed well with good accuracy. HUTS was substantially faster than EBOS in all cases but once again was less accurate. Our results with Octave suggest that both EBOS and HUTS provide a viable alternative to the direct method.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

TABLE 9
Consecutive trials for $m = n = 400$ with $(dh, q) \in \{(100, 4), (50, 8), (20, 20), (5, 80)\}$.

m, n	dh	q	t_e	t_h	t_d	ϵ_e	ϵ_h	ϵ_d
400	100	4	0.0952	0.0587	0.0622	4.0219×10^{-7}	9.1451×10^{-5}	1.1085×10^{-6}
			0.0932	0.0697	0.0424	1.1042×10^{-7}	4.0510×10^{-6}	3.0966×10^{-8}
			0.0886	0.0667	0.0419	1.7819×10^{-7}	1.8577×10^{-3}	1.1117×10^{-6}
400	50	8	0.1040	0.0567	0.0470	1.6676×10^{-6}	1.3080×10^{-4}	4.6794×10^{-5}
			0.1275	0.0720	0.0514	9.5459×10^{-8}	8.8970×10^{-7}	1.6224×10^{-7}
			0.0985	0.0566	0.0390	5.7813×10^{-8}	8.7645×10^{-6}	2.5045×10^{-7}
400	20	20	0.1228	0.0719	0.0457	2.4101×10^{-8}	2.1085×10^{-7}	3.5435×10^{-7}
			0.1299	0.0751	0.0395	1.2841×10^{-8}	1.1216×10^{-7}	1.3129×10^{-7}
			0.1229	0.0747	0.0500	3.5637×10^{-8}	4.2854×10^{-8}	2.0027×10^{-7}
400	10	40	0.2235	0.1180	0.0631	8.0690×10^{-8}	2.5807×10^{-7}	9.1892×10^{-7}
			0.1799	0.1158	0.0439	2.5827×10^{-8}	1.4366×10^{-6}	6.6482×10^{-7}
			0.1829	0.1121	0.0491	4.3011×10^{-8}	1.0953×10^{-7}	1.3387×10^{-7}
400	5	80	0.3142	0.2007	0.0459	6.9575×10^{-8}	3.3250×10^{-8}	1.4328×10^{-7}
			0.3325	0.2071	0.0369	8.1913×10^{-9}	8.6800×10^{-9}	6.8199×10^{-8}
			0.3104	0.1983	0.0446	1.2978×10^{-8}	4.0828×10^{-6}	5.3380×10^{-7}

8. Conclusion. The recently proposed EBOS [20] for the least squares solution of large linear systems significantly reduces computational time compared to that needed by the direct method applied to the same systems. Nevertheless, in some cases, the EBOS algorithm may still be regarded as too slow. Our motivation was to find a method which would further reduce the computation time needed. To this end, we have extended another recognized numerical procedure to develop the *so-called* HUTS algorithm. The HUTS finds the least squares solution of large linear systems in the form $YCC^* = AC^*$, where $A \in \mathbb{C}^{m\times n}$ with $\text{rank}(A) = h$ and $C \in \mathbb{C}^{h\times n}$, where $m, n \in \mathbb{N}$ are *large* and $n = h + s$, where $h = \sum_{j=1}^{q} h_j$ with $h_j \in \mathbb{N}$ and $s \in \mathbb{N} - 1$. The original system $YCC^* = AC^*$ is transformed to a simpler system $YL_1L_1^* = AQ_1^*L_1^*$, where L_1 is block lower triangular and $Q \in \mathbb{C}^{n\times n}$ is unitary. If $\text{rank}(C) = h$, then L_1 is invertible and $YL_1 = AQ_1^*$. This system can be solved by inverting the block diagonal elements of L_1 and using a scheme of back substitution.

While the direct method requires finding, the Moore–Penrose inverse of a large matrix both the EBOS and HUTS is reduced to a collection of much smaller problems where calculation of the Moore–Penrose inverse matrices is much quicker. Thus, the number of flops needed by both the EBOS and HUTS is much reduced compared to the number of flops used by the direct method. It turns out that the HUTS is much faster than the EBOS. At the same time, our tests with randomly generated matrices showed that while both schemes gave accurate results the EBOS is more accurate and more reliable than the HUTS. If speed of calculation is more important than accuracy, then the HUTS should be used. If accuracy is more important than the speed of calculation, then the EBOS should be applied.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

REFERENCES

[1] J.K. Baksalary and O.M. Baksalary. Particular formulae for the Moore–Penrose inverse of a columnwise partitioned matrix. *Linear Algebra Appl*, 421:16–23, 2007. https://doi.org/10.1016/j.laa.2006.03.031.

[2] A. Ben-Israel. Generalized inverses of matrices: A perspective of the work of Penrose. *Math. Proc. Cambr. Philos. Soc.*, 100(3):407–425, 1986. https://doi.org/10.1017/S0305004100066172.

[3] A. Ben-Israel. The Moore of the Moore–Penrose Inverse. *Electron. J. Linear Algebra*, 9:150–157, 2002. https://doi.org/10.13001/1081-3810.1083.

[4] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. https://doi.org/10.1007/BF02288367.

[5] C. Eckart and G. Young. A principal axis transformation for non-hermitian matrices. *Bull. Amer. Math. Soc.*, 45(2):118–122, 1939. https://doi.org/10.1090/S0002-9904-1939-06910-3.

[6] V.N. Fomin and M.V. Ruzhansky. Abstract optimal linear filtering. *SIAM J. Control Optim.*, 38:1334–1352, 2000. https://doi.org/10.1137/S036301299834778X.

[7] J.G.F. Francis. The QR Transformation, I. *Comput. J.*, 4(3):265–271, 1961. https://doi.org/10.1093/comjnl/4.3.265.

[8] J.G.F. Francis. The QR transformation, II. *Comput. J.*, 4(4):332–345, 1962. https://doi.org/10.1093/comjnl/4.4.332.

[9] S. Friedland and A. Torokhti. Generalized rank-constrained matrix approximations. *SIAM J. Mat. Anal. Appl.*, 29:656–659, 2007. https://doi.org/10.1137/06065551.

[10] G.H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal.*, 2:205–224, 1965. www.jstor.org/stable/2949777.

[11] G.H. Golub and C. Reinsch. Singular value decomposition and least squares solution. *Numer. Math.*, 14:403–420, 1970. https://doi.org/10.1007/BF02163027.

[12] G.H. Golub, A. Hoffman, and G.W. Stewart. A generalization of the Eckart–Young–Mirsky matrix approximation theorem. *Linear Algebra Appl.*, 88–89:317–327, 1987. https://doi.org/10.1016/0024-3795(87)90114-5.

[13] G.H. Golub and C. F. Van Loan. *Matrix Computations* 4^{th} *Ed.*, Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

[14] B.F. Green. The orthogonal approximation of the oblique structure in factor analysis. *Psychometrika*, 17:429–440, 1952. https://psycnet.apa.org/doi/10.1007/BF02288918.

[15] H. Hotelling. Analysis of a complex of statistical variables into principal components (1). *J. Educ. Psychol.*, 24(6):417–441, 1933. https://doi.org/10.1037/h0071325.

[16] H. Hotelling. Analysis of a complex of statistical variables into principal components (2). *J. Educ. Psychol.*, 24(7):498–520, 1933. https://doi.org/10.1037/h0070888.

[17] A.S. Householder and G. Young. Matrix approximation and latent roots. *Amer. Math. Monthly*, 45(3):165–171, 1938. https://doi.org/10.1080/00029890.1938.11990787.

[18] P.G. Howlett, C.E.M. Pearce, and A.P. Torokhti. An optimal linear filter for random signals with realisations in Hilbert Space. *ANZIAM J.*, 44:485–500, 2003. https://doi.org/10.1017/S1446181100012888.

[19] P. Howlett and A. Torokhti. An optimal linear filter for estimation of random functions in Hilbert space. *ANZIAM J.*, 62(3):274–301, 2020. https://doi.org/10.1017/S1446181120000188.

[20] P. Howlett and A. Torokhti. Optimal approximation of a large matrix by a sum of projected linear mappings on prescribed subspaces. *Electron. J. Linear Algebra*, 40:585–605, 2024. https://doi.org/10.13001/ela.2024.8645.

[21] Y. Hua and W.Q. Liu. Generalized Karhunen-Loeve transform. *IEEE Sig. Process. Lett.*, 5:141–142, 1998. https://doi.org/10.1109/97.681430.

[22] V.N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Comput. Math. Math. Phys.*, 1(3):637–657, 1962. https://doi.org/10.1016/0041-5553(63)90168-X.

[23] X. Liu, W. Li, and H. Wang. Rank constrained matrix best approximation problem with respect to (skew) Hermitian matrices. *J. Comput. Appl. Math.*, 319:77–86, 2017. http://dx.doi.org/10.1016/j.cam.2016.12.029.

[24] D.G. Luenberger. *Optimization by Vector Space Methods*. Wiley, New York, 1968. Reprinted in 1997.

[25] L. Mirsky. Symmetric gauge functions and unitarily invariant matrix norms. *Quart. J. Math.*, 11:50–59, 1960. https://doi.org/10.1093/qmath/11.1.50.

[26] E.H. Moore. On the reciprocal of the general algebraic matrix. *Bull. Amer. Math. Soc.*, 26(9):394–395, 1920. In A. Dresden. The Fourteenth Western Meeting of the American Mathematical Society, 26(9), 385–396, 1920. https://doi.org/10.1090/S0002-9904-1920-03322-7.

[27] R Penrose, (1955). A generalized inverse for matrices, *Proc. Cambr. Philos. Soc.*, 51, 406–413. https://doi.org/10.1017/S0305004100030401.

[28] F. Rotella and I. Zambettakis. Block householder transformation for parallel QR factorization. *Appl. Math. Lett.*, 12:29–34, 1999. https://doi.org/10.1016/S0893-9659(99)00028-2.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.

[29] E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. I Teil. Entwicklung willkürlichen Funktionen nach System vorgeschriebener. *Math. Ann.*, 63:433–476, 1907. https://doi.org/10.1007/BF01449770.

[30] P.H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966. https://doi.org/10.1007/BF02289451.

[31] G.W. Stewart. On the early history of the singular value decomposition. *SIAM Rev.*, 35(4):551–566, 1993. https://www.jstor.org/stable/2132388.

[32] H. Weyl. Das asymptotische Verteilungsgesetz der Eigenwert linearer partieller Differentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung). *Math. Ann.*, 71:441–479, 1912.

[33] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938. https://doi.org/10.1007/BF02287916.

[34] G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6:49–53, 1941. https://doi.org/10.1007/bF02288574.

Appendix A. MATLAB code for solution of $YCC^* = AC^*$.

```
%this program compares solution of Y*C*C' = A*C' by a direct MATLAB
%calculation with solutions using the EBOS and HUTS algorithms
clear all

%size(A) = (m,n), size(C) = (q*dh,n) where q*dh <= n
m = 4000;  n = 4000;  dh = 800;  q = 5;

%construct the matrices and time the construction
tic
AL = rand(m,q*dh);
AR = rand(q*dh,n);
A = AL*AR;
C = rand(q*dh,n);
vh = [];
for u=1:q
    vh = [vh dh];
end
tr = toc;

%remember the initial matrices
A0 = A;
C0 = C;

%calculate the solution Yd and the elapsed time td for the direct method
tic
Yd = A0*pinv(C0);
td = toc;

%calculate the error with the direct method
edf = norm(Yd*C0*C0'-A0*C0',"fro");

%calculate the solution Ye and the elapsed time te for EBOS
C = C0;
```

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.


```
tic
F = eye(q*dh);
F(dh+1:q*dh,1:dh) = - C(dh+1:q*dh,:)*pinv(C(1:dh,:));
C(dh+1:q*dh,:) = C(dh+1:q*dh,:)*(eye(n) - pinv(C(1:dh,:))*C(1:dh,:));
D(1:dh,1:dh) = C(1:dh,1:n)*C(1:dh,1:n)';
Ddag = pinv(D(1:dh,1:dh));
for j=2:q-1
F(j*dh+1:q*dh,1:j*dh) = [- C(j*dh+1:q*dh,:)*pinv(C((j-1)*dh+1:j*dh,:)),eye
    ((q-j)*dh)]*...
     F((j-1)*dh+1:q*dh,1:j*dh);
C(j*dh+1:q*dh,:) = C(j*dh+1:q*dh,:)*(eye(n) - pinv(C((j-1)*dh+1:j*dh,:))*C
    ((j-1)*dh+1:j*dh,:));
D((j-1)*dh+1:j*dh,(j-1)*dh+1:j*dh) = C((j-1)*dh+1:j*dh,:)*C((j-1)*dh+1:j*
    dh,:)';
Ddag = blkdiag(Ddag,pinv(D((j-1)*dh+1:j*dh,(j-1)*dh+1:j*dh)));
end
D((q-1)*dh+1:q*dh,(q-1)*dh+1:q*dh) = C((q-1)*dh+1:q*dh,:)*C((q-1)*dh+1:q*
    dh,:)';
Ddag = blkdiag(Ddag,pinv(D((q-1)*dh+1:q*dh,(q-1)*dh+1:q*dh)));
ACp = A*C';
ACpDd = [];
for j=1:q
     ACpDd = [ACpDd ACp(:,(j-1)*dh+1:j*dh)*Ddag((j-1)*dh+1:j*dh,(j-1)*dh+1:
         j*dh)];
end
Ye = [];
for j=1:q
Ye = [Ye ACpDd(1:m,(j-1)*dh+1:q*dh)*F((j-1)*dh+1:q*dh,(j-1)*dh+1:j*dh)];
end
te = toc;

%calculate the error for EBOS
eef = norm(Ye*C0*C0'-A0*C0',"fro");

%define the initial settings for HUTS
index = 0;
Lh = C0;
Ah = A0;
Lhtemp = Lh;
Ahtemp = Ah;

%use HUTS to reduce Lh to block lower triangular form
tic
for j=1:q-1+sign(n-q*dh)
     G1 = Lhtemp(1:vh(j),1:vh(j));
```

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.


```
    G2 = Lhtemp(1:vh(j),vh(j)+1:n-index);
    W = eye(vh(j)) + pinv(G1)*G2*G2'*pinv(G1');
    [P,D,Pprime] = svd(W);
    X = G1*P*sqrt(D)*P';
    V = [G1 - X, G2];
    K = eye(n-index) - 2*V'*pinv(V*V')*V;
    Lhtemp = Lhtemp*K;
    Ahtemp = Ahtemp*K;
    Lh(index+1:q*dh,index+1:n) = Lhtemp;
    Ah(:,index+1:n) = Ahtemp;
    index = index+vh(j);
    Lhtemp = Lh(index+1:q*dh,index+1:n);
    Ahtemp = Ah(:,index+1:n);
end

thred = toc;

%use back substitution to find Yh satisfying Yh*Lh = Ah
tic
Lh = Lh(:,1:q*dh);
Ah = Ah(:,1:q*dh);
Ysnew = Ah(:,q*dh-vh(q)+1:q*dh)*pinv(Lh(q*dh-vh(q)+1:q*dh,q*dh-vh(q)+1:q*
   dh));
Ytemp = Ysnew;
index = q*dh-vh(q);
for k = 1:q-1
    Y = (Ah(:,index-vh(q-k)+1:index)...
        -Ytemp*Lh(index+1:q*dh,...
        index-vh(q-k)+1:index))...
        *pinv(Lh(index-vh(q-k)+1:index,...
        index-vh(q-k)+1:index));
    Ytemp = [Y Ytemp];
    index = index-vh(q-k);
end
Yh = Ytemp;
thback = toc;

%calculate the elapsed time for HUTS
th = thred + thback;

%calculate the error for HUTS
ehf = norm(Yh*C0*C0'-A*C0',"fro");

%record the times for each method
te
```

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 41, pp. 538-574, October 2025.


```
th
td

%record the errors for each method
eef
ehf
edf

%record the discrepancies in the solutions
Yed = norm(Ye-Yd,"fro")
Yhd = norm(Yh-Yd,"fro")
Yeh = norm(Ye-Yh,"fro")
```