Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

OPTIMAL APPROXIMATION OF A LARGE MATRIX BY A SUM OF PROJECTED LINEAR MAPPINGS ON PRESCRIBED SUBSPACES*

PHIL HOWLETT† AND ANATOLI TOROKHTI‡

Abstract. We propose and justify a matrix reduction method for calculating the optimal approximation of an observed matrix $A \in \mathbb{C}^{m\times n}$ by a sum $\sum_{i=1}^{p}\sum_{j=1}^{q} B_i X_{ij} C_j$ of matrix products where each $B_i \in \mathbb{C}^{m\times g_i}$ and $C_j \in \mathbb{C}^{h_j\times n}$ is known and where the unknown matrix kernels X_{ij} are determined by minimizing the Frobenius norm of the error. The sum can be represented as a bounded linear mapping BXC with unknown kernel X from a prescribed subspace $\mathcal{T} \subseteq \mathbb{C}^n$ onto a prescribed subspace $\mathcal{S} \subseteq \mathbb{C}^m$ defined, respectively, by the collective domains and ranges of the given matrices $C_1, \ldots, C_q$ and $B_1, \ldots, B_p$. We show that the optimal kernel is $X = B^\dagger A C^\dagger$ and that the optimal approximation $BB^\dagger AC^\dagger C$ is the projection of the observed mapping A onto a mapping from $\mathcal{T}$ to $\mathcal{S}$. If A is large, B and C may also be large and direct calculation of $B^\dagger$ and $C^\dagger$ becomes unwieldy and inefficient. The proposed method avoids this difficulty by reducing the solution process to finding the pseudo-inverses of a collection of much smaller matrices. This significantly reduces the computational burden.

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

AMS subject classifications. 15A18.

1. Introduction.

Let $A \in \mathbb{C}^{m\times n}$ be an observed matrix and let

$$\mathcal{S} = B_1(\mathbb{C}^{g_1}) \cup \cdots \cup B_p(\mathbb{C}^{g_p}) \subseteq \mathbb{C}^m \quad \text{and} \quad \mathcal{T} = C_1^*(\mathbb{C}^{h_1}) \cup \cdots \cup C_q^*(\mathbb{C}^{h_q}) \subseteq \mathbb{C}^n,$$

be prescribed subspaces defined by known matrices $B_i \in \mathbb{C}^{m\times g_i}$ with $g_i \in \mathbb{N}$ for each $i = 1, \ldots, p$ and $C_j \in \mathbb{C}^{h_j\times n}$ with $h_j \in \mathbb{N}$ for each $j = 1, \ldots, q$. In general, we assume that $m, n \in \mathbb{N}$ are large and that $g_1, \ldots, g_p \in \mathbb{N}$ and $h_1, \ldots, h_q \in \mathbb{N}$ are small. Consider the following problem.

PROBLEM 1. *Find $X_{ij} \in \mathbb{C}^{g_i\times h_j}$ for each $i = 1, \ldots, p$ and $j = 1, \ldots, q$ to solve*

$$\min_{\substack{X_{ij}\\ i=1,\ldots,p, j=1,\ldots,q}} \left\| A - \sum_{i=1}^{p}\sum_{j=1}^{q} B_i X_{ij} C_j \right\|_F, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. □

Let $g = g_1 + \ldots + g_p$ and $h + h_1 + \ldots + h_q$. If we define $B \in \mathbb{C}^{m\times g}$, $C \in \mathbb{C}^{h\times n}$, and $X \in \mathbb{C}^{g\times h}$ by setting

$$B = \begin{bmatrix} B_1 & \cdots & B_p \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ \vdots \\ C_q \end{bmatrix}, \quad X = \begin{bmatrix} X_{11} & \cdots & X_{1q} \\ \vdots & \ddots & \vdots \\ X_{p1} & \cdots & X_{pq} \end{bmatrix},$$

then Problem 1 can be stated more succinctly as follows.

*Received by the editors on April 18, 2024. Accepted for publication on August 12, 2024. Handling Editor: Nicolas Gillis. Corresponding Author: Anatoli Torokhti.

†Scheduling and Control Group (SCG), Centre for Industrial and Applied Mathematics (CIAM), UniSA STEM, University of South Australia, South Australia 5095, Australia (phil.howlett@unisa.edu.au).

‡Retired from Centre for Industrial and Applied Mathematics (CIAM), UniSA STEM, University of South Australia, South Australia 5095, Australia (anatoli.torokhti@gmail.com).

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

PROBLEM 2. *Find* $X \in \mathbb{C}^{g\times h}$ *to solve*

$$\min_X \|A - BXC\|_F, \tag{2}$$

where $\|\cdot\|_F$ *denotes the Frobenius norm.* □

1.1. Preliminaries. It is well known [15] that $X \in \mathbb{C}^{g\times h}$ is a solution to (2) if and only if it is also a solution to

$$B^*BXCC^* = B^*AC^*. \tag{3}$$

Equation (3) always has a solution, but in general the solution is not unique. If we write $B^\dagger$ and $C^\dagger$ for the respective Moore–Penrose inverse matrices, then the general solution is given by:

$$X = B^\dagger AC^\dagger + (I - B^\dagger B)M(I - CC^\dagger), \tag{4}$$

where $M \in \mathbb{C}^{g\times h}$ is arbitrary. For every solution X, we have

$$BXC = BB^\dagger AC^\dagger C. \tag{5}$$

Therefore the best approximation BXC is uniquely determined. Since $BB^\dagger$ is the projection of the space $\mathbb{C}^m$ onto the subspace $\mathcal{S} \subseteq \mathbb{C}^m$ and $C^\dagger C$ is the projection of the space $\mathbb{C}^n$ onto the subspace $\mathcal{T} \subseteq \mathbb{C}^n$ we can see that the optimal approximation $BXC \in \mathcal{B}(\mathcal{T}, \mathcal{S})$ is the projection of the bounded linear mapping $A \in \mathcal{B}(\mathbb{C}^n, \mathbb{C}^m)$ onto a bounded linear mapping from $\mathcal{T}$ to $\mathcal{S}$. Note that if $\boldsymbol{x} \in \mathcal{T}$ and $\boldsymbol{y} = A\boldsymbol{x} \in \mathcal{S}$ then

$$BXC\boldsymbol{x} = B(B^\dagger AC^\dagger)C\boldsymbol{x} = BB^\dagger A(C^\dagger C\boldsymbol{x}) = BB^\dagger A\boldsymbol{x} = BB^\dagger \boldsymbol{y} = \boldsymbol{y}.$$

In this case the approximate mapping is exact.

1.2. Motivation. Our matrix approximation problem is motivated by tasks arising in data processing. Suppose it is necessary to repeatedly transmit large data matrices from a remote location to a central storage facility. Although there may be numerous portable devices in the field that can transmit data it may also be the case that no single device has the capacity to transmit the complete data set in the time allowed. This scenario is considered, for example, in [28, 44] and the references therein. More specifically suppose we wish to transmit a large data matrix representing a digital message or an image, from one location to another using an array of transmitters. We assume that each individual transmitter can be used to transmit a small matrix $X_{ij} \in \mathbb{C}^{h_i\times g_j}$ for each $i = 1, \ldots, p$ and $j = 1, \ldots, q$. If $B_i \in \mathbb{C}^{m\times g_i}$ and $C_j \in \mathbb{C}^{h_j\times n}$ are known matrices to both sender and receiver, the sender can find each X_{ij} by solving Problem 1 and the receiver can reconstruct A approximately from the data array $X = [X_{ij}]$ according to the formula $A \approx \sum_{i=1}^p \sum_{j=1}^q B_i X_{ij} C_j$. We could regard $X = [X_{ij}]$ as a codified form of the intended message A in which case the matrices $B = [B_i]$ and $C = [C_j]$ would represent decoders.

Alternatively, we could imagine that A represents the kernel of a large linear system and that we wish to restrict our analysis to certain key input–output relationships. See, for example, [45] and the references therein. Suppose we are interested in those inputs that lie in the space $\mathcal{T}$ defined by the collective domains of the matrices C_j for $j = 1, \ldots, q$ and those outputs that lie in the space $\mathcal{S}$ defined by the collective ranges of the matrices B_i for $i = 1, \ldots, p$. The matrix X represents the kernel of an economized linear system BXC which is restricted to the designated input–output relationships.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

1.3. Aim. Our aim is to replace the direct calculation described in (4) by a more efficient solution procedure and hence reduce the computational burden associated with the direct calculation of generalized inverses of large matrices.

The minimum norm solution to both (2) and (3) is uniquely defined by the formula $X_0 = B^\dagger AC^\dagger$ where $B \in \mathbb{C}^{m\times g}$ and $C \in \mathbb{C}^{h\times n}$. We have $B^\dagger = \{B^*B\}^\dagger B^*$ where $B^*B \in \mathbb{C}^{g\times g}$ and $C^\dagger = C^*\{CC^*\}^\dagger$ where $CC^* \in \mathbb{C}^{h\times h}$. Direct calculation of $B^\dagger$ and $C^\dagger$ using a standard singular value decomposition (SVD) algorithm will take, respectively, $\mathcal{O}(g^3)$ and $\mathcal{O}(h^3)$ floating point operations (flops). When g, h are large, this direct calculation becomes unwieldy and inefficient. We propose a more efficient alternative to this direct calculation. We are primarily interested in the reconstruction of A. In this regard, it is important to remember that $BXC = BX_0C$ for all X given by (4). Thus, it is not necessary to compute the minimum norm solution.

If the subspaces $\mathcal{S}_1 = B_1(\mathbb{C}^m), \ldots, \mathcal{S}_p = B_p(\mathbb{C}^m)$ are mutually orthogonal and the subspaces $\mathcal{T}_1 = C_1^*(\mathbb{C}^n), \ldots, \mathcal{T}_q = C_q^*(\mathbb{C}^n)$ are also mutually orthogonal, then the optimal matrices X_{ij} are determined independently with $X_{ij} = B_i^\dagger AC_j^\dagger$ for each $i = 1, \ldots, p$ and $j = 1, \ldots, q$. This is not true in general.

2. Contribution. We propose a new method for numerical solution of the matrix equation $B^*BXCC^* = B^*AC^*$ called the elementary block operations scheme (EBOS). A related algorithm and Matlab code are represented in Sections 4.4 and 6, respectively. To begin, we write the equation as two separate systems. We define Y as the solution to the equation $YCC^* = AC^*$ and define X as the solution to the equation $B^*BX = B^*Y$. The scheme is designed to greatly reduce the computational burden.

• **Elementary block operations scheme (EBOS).** We use a result established by Baksalary and Baksalary [2, Theorem 3, pp 21–22] to justify nonsingular transformations $E = E_1 \cdots E_{p-1}$ and $F = F_{q-1} \cdots F_1$ defined, respectively, by a product of nonsingular elementary block upper and lower triangular matrices so that the transformed matrices

$$B^{(p-1)} = BE = \begin{bmatrix} B_1 & B_2^{(1)} & \cdots & B_p^{(p-1)} \end{bmatrix} \quad \text{and} \quad C^{(q-1)} = FC = \begin{bmatrix} C_1 \\ C_2^{(1)} \\ \vdots \\ C_q^{(q-1)} \end{bmatrix},$$

have the special property that the Moore–Penrose inverses are given by:

$$[B^{(p-1)}]^\dagger = \begin{bmatrix} B_1^\dagger \\ [B_2^{(1)}]^\dagger \\ \vdots \\ [B_p^{(p-1)}]^\dagger \end{bmatrix} \quad \text{and} \quad [C^{(q-1)}]^\dagger = \begin{bmatrix} C_1^\dagger & [C_2^{(1)}]^\dagger & \cdots & [C_q^{(q-1)}]^\dagger \end{bmatrix}.$$

The reduction also ensures that the matrices $D_B = [B^{(p-1)}]^*B^{(p-1)}$ and $D_C = C^{(q-1)}[C^{(q-1)}]^*$ are block diagonal and that the systems $B^*BX = B^*Y \iff D_BE^{-1}X = [B^{(p-1)}]^*Y$ and $Y^*CC^* = AC^* \iff Y^*F^{-1}D_C = A[C^{(q-1)}]^*$ can be solved by calculating the Moore–Penrose inverses of the block diagonal elements of the transformed coefficients.

2.1. Benefit. We will show that EBOS can be easily implemented using the standard general-purpose MATLAB algorithms. In practice, we would expect these general-purpose algorithms to be replaced in the

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

proposed scheme by more efficient special numerical techniques considered in a number of works. We cite [16] as a particular instance and refer to [20] for a general discussion of the relevant numerical methods. The problem of calculating the Moore–Penrose inverses of a $g \times g$ matrix and an $h \times h$ matrix is reduced to a much smaller problem of calculating the Moore–Penrose inverses of a $g_i \times g_i$ matrix for each $i = 1, \ldots, p$ and 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 $\mathcal{O}(g^3) + \mathcal{O}(h^3)$ to $\mathcal{O}(\sum_{i=1}^{p} g_i^3) + \mathcal{O}(\sum_{j=1}^{q} h_j^3)$. We make a general comparison of our method with the direct Matlab method by counting the number of flops in a full range of different circumstances. In problems such as those in biology, ecology, finance, sociology, and medicine (see e.g., [23, 33, 1]), $m, n = O(10^4)$ and greater. For example, measurements of gene expression can contain tens of hundreds of samples. Each sample can contain tens of thousands of genes. As a result, in such cases, the associated matrices are very large. In particular, the well-known phenomenon of the "curse of dimensionality" [4] states that many problems become exponentially difficult in high dimensions.

3. Literature review. The topic of matrix approximation has a long history with contributions from many authors. The motivations for individual works are wide-ranging and the inter-connections are complex. The early work in the period 1905–1912 on integral equations [40, 47] was motivated by the development of a systematic theory for solution of integral equations. The work in [40, 47] essentially established a SVD for bounded integral operators on the infinite-dimensional space of square integrable functions and proved that the best k-dimensional approximation to the integral operator was the projected integral operator defined by the characteristic vectors associated with the k largest singular values. The next contributions in the period 1933–1941 were motivated by applications to psychometrics [10, 11, 25, 26, 27, 41, 47, 48, 49] where an observed test score matrix was approximated by a lower rank matrix. The underlying idea was that linear dependencies in the true score were obscured in the observed score by the addition of random errors. The most notable results were the Eckart–Young theorem [10] which was essentially the Schmidt approximation theorem [40] in matrix notation and the principal component analysis of Hotelling [25, 26]. See also later work by Green [24] and Schönemann [41] which was also motivated by psychometric considerations. The SVD was the mathematical foundation for all of this work. The next results were motivated by fundamental mathematical considerations [36] and by development of effective algorithms for numerical calculation of the SVD and the closely related QR decomposition [13, 14, 17, 18, 32]. A detailed description of the above work can be found in the 1993 survey paper by G W Stewart [43].

Development of a comprehensive theory of generalized inverse matrices was closely related to the conceptual development of the SVD. The *so-called* Moore–Penrose matrix inverse was first discovered and justified by E.H. Moore [5, 37] and later rediscovered and elaborated by R. Penrose [6, 38]. See [20, 35] for further discussion.

Friedland and Torokhti [15] solved the generalized Eckart–Young problem:

$$\min_{X \in \mathcal{R}(m,n,k)} \|A - BXC\|, \tag{6}$$

where $A \in \mathbb{C}^{m \times n}$, $B \in \mathbb{C}^{m \times g}$ and $C \in \mathbb{C}^{n \times h}$ are known and $\mathcal{R}(m, n, k)$ is the set of all $X \in \mathbb{C}^{m \times n}$ with rank $(X) \leq k$. Liu et al. [34] solved a similar problem for Hermitian or skew-Hermitian matrices. See also [46, 42] for similar forms of the rank constrained matrix approximation. Other forms of constrained matrix approximation include the so-called skeleton approximations [7, 8, 9, 22].

The work on matrix approximation has also found applications in signal processing where Hua and Liu [31] applied the Eckart–Young theorem to the dimensionality reduction of transmitted random signals.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

A similar approximation was established for estimation of random signals in infinite-dimensional Hilbert space in [12, 29] provided the generalized inverse auto-covariance operator is a bounded operator. This result was recently further extended by Howlett and Torokhti [30] to the case where the generalized inverse auto-covariance operator is closed but unbounded.

4. The elementary block operations scheme (EBOS). EBOS uses a sequence of elementary block row operations to reduce each of the separated systems $B^*BX = B^*Y$ and $YCC^* = AC^*$ to block diagonal form.

4.1. A theoretical basis for EBOS. We use a key result proved by Baksalary and Baksalary [2] in 2007. See also [3] for further discussion. It is convenient to state the key result in the following form.

THEOREM 1. *Let*

$$R = \left[\begin{array}{c} R_1 \\ \hline R_2 \end{array}\right] \in \mathbb{C}^{m\times n},$$

where $R_1 \in \mathbb{C}^{m_1\times n}$, $R_2 \in \mathbb{C}^{m_2\times n}$ *and* $m = m_1 + m_2$. *Suppose*

$$S = \left[\begin{array}{c|c} R_1^\dagger & R_2^\dagger \end{array}\right],$$

where $R_j^\dagger$ *denotes the Moore–Penrose inverse of* R_j *for each* $j = 1, 2$. *The following statements are equivalent*

1. $S = R^\dagger$,
2. $RSR = R$,
3. $R_2^{-1}(\{\mathbf{0}\})^\perp \subseteq R_1^{-1}(\{\mathbf{0}\}) \iff R_2^*(\mathbb{C}^n) \subseteq R_1^{-1}(\{\mathbf{0}\})$,
4. $R_1^{-1}(\{\mathbf{0}\})^\perp \subseteq R_2^{-1}(\{\mathbf{0}\}) \iff R_1^*(\mathbb{C}^n) \subseteq R_2^{-1}(\{\mathbf{0}\})$,

where $R_j^{-1}(\{\mathbf{0}\}) \subseteq \mathbb{C}^n$ *denotes the null space of* R_j, $R_j^{-1}(\{\mathbf{0}\})^\perp \subseteq \mathbb{C}^n$ *denotes the orthogonal complement of the null space of* R_j, *and* $R_j(\mathbb{C}^n) \subseteq \mathbb{C}^m$ *denotes the range space of* R_j *in* $\mathbb{C}^{m_j}$ *considered as a subspace of* $\mathbb{C}^m$ *for each* $j = 1, 2$. □

Suppose that R does not satisfy the conditions of Theorem 4.1. We can use elementary row operations to transform the matrix R into a matrix $R^{(1)}$ that does satisfy the conditions of Theorem 4.1. Define

$$R^{(1)} = F_1 R = \left[\begin{array}{c|c} I & 0 \\ \hline -R_2R_1^\dagger & I \end{array}\right] \left[\begin{array}{c} R_1 \\ \hline R_2 \end{array}\right] = \left[\begin{array}{c} R_1 \\ \hline R_2(I - R_1^\dagger R_1) \end{array}\right] = \left[\begin{array}{c} R_1^{(1)} \\ \hline R_2^{(1)} \end{array}\right].$$

Now $R_2^{(1)}[R_1^{(1)}]^\dagger = R_2(I - R_1^\dagger R_1)R_1^\dagger = 0$ implies $[R_1^{(1)}]^\dagger(\mathbb{C}^m) \subseteq [R_2^{(1)}]^{-1}(\{\mathbf{0}\}) \iff R_1^{-1}(\{\mathbf{0}\})^\perp \subseteq R_2^{-1}(\{\mathbf{0}\})$. Therefore, $[R^{(1)}]^\dagger = \left[\begin{array}{c|c} [R_1^{(1)}]^\dagger & [R_2^{(1)}]^\dagger \end{array}\right]$. For more information, see Luenberger [35, p 164, Figure 6.6].

EXAMPLE 1. *Let*

$$R = \left[\begin{array}{c} R_1 \\ \hline R_2 \end{array}\right] = \left[\begin{array}{rrrrrr} 1 & 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ \hline 1 & 0 & 1 & -1 & 2 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & -2 & -2 & 0 & 1 \end{array}\right].$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

Direct MATLAB *calculations show that* $R^\dagger \neq \left[R_1^\dagger \mid R_2^\dagger\right]$. *If we define* $R_2^{(1)} = R_2(I - R_1^\dagger R_1)$, *then* MATLAB *returns*

$$R_2^{(1)} = \begin{bmatrix} 0.0000 & 0 & 0.0000 & 0 & 0.0000 & 0 \\ 0.4000 & 1.0000 & 0.0000 & -0.2000 & -0.4000 & 0.2000 \\ 1.0000 & 2.0000 & -0.0000 & -1.0000 & -1.0000 & 0 \end{bmatrix}.$$

Therefore,

$$R^{(1)} = \begin{bmatrix} R_1^{(1)} \\ \hline R_2^{(1)} \end{bmatrix} = \left[\begin{array}{cccccc} 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 \\ -1.0000 & 0 & 1.0000 & -1.0000 & 0 & 1.0000 \\ 1.0000 & 0 & -1.0000 & 0 & 1.0000 & 0 \\ \hline 0.0000 & 0 & 0.0000 & 0 & 0.0000 & 0 \\ 0.4000 & 1.0000 & 0.0000 & -0.2000 & -0.4000 & 0.2000 \\ 1.0000 & 2.0000 & -0.0000 & -1.0000 & -1.0000 & 0 \end{array}\right].$$

Now by MATLAB,

$$[R_1^{(1)}]^\dagger = \begin{bmatrix} 0.3000 & -0.2000 & 0.1000 \\ 0.0000 & -0.0000 & -0.0000 \\ 0.5000 & -0.0000 & -0.5000 \\ 0.1000 & -0.4000 & -0.3000 \\ 0.2000 & 0.2000 & 0.4000 \\ -0.1000 & 0.4000 & 0.3000 \end{bmatrix}, \quad [R_2^{(1)}]^\dagger = \begin{bmatrix} 0.0000 & -0.2500 & 0.2500 \\ -0.0000 & 1.2500 & -0.2500 \\ -0.0000 & 0.0000 & -0.0000 \\ 0.0000 & 2.0000 & -1.0000 \\ 0.0000 & 0.2500 & -0.2500 \\ 0.0000 & 1.7500 & -0.7500 \end{bmatrix},$$

and

$$[R^{(1)}]^\dagger = \left[\begin{array}{ccc|ccc} 0.3000 & -0.2000 & 0.1000 & 0.0000 & -0.2500 & 0.2500 \\ -0.0000 & -0.0000 & -0.0000 & 0.0000 & 1.2500 & -0.2500 \\ 0.5000 & -0.0000 & -0.5000 & 0.0000 & 0.0000 & 0.0000 \\ 0.1000 & -0.4000 & -0.3000 & 0.0000 & 2.0000 & -1.0000 \\ 0.2000 & 0.2000 & 0.4000 & 0.0000 & 0.2500 & -0.2500 \\ -0.1000 & 0.4000 & 0.3000 & 0.0000 & 1.7500 & -0.7500 \end{array}\right],$$

that is,

$$[R^{(1)}]^\dagger = \left[\begin{array}{c|c} [R_1^{(1)}]^\dagger & [R_2^{(1)}]^\dagger \end{array}\right].$$

Thus, the Moore–Penrose inverse of $R^{(1)}$ *can be calculated by finding the Moore–Penrose inverse of each block component.* □

4.2. Application of EBOS to solve $YCC^* = AC^*$. We will begin by describing the solution procedure for the system $YCC^* = AC^*$. In other words, we wish to find Y_+ which satisfies $Y_+CC^* = AC^*$ and hence minimizes $\|A - YC\|_F$. The general solution to this problem is given by:

$$Y = Y_0 + K[I - (CC^*)^\dagger], \tag{7}$$

where $Y_0 = AC^*[CC^*]^\dagger$ is the minimum norm solution and $K \in \mathbb{C}^{m\times h}$ is arbitrary. Our procedure for finding the proposed solution Y_+ is as follows.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

Define $C^{(0)} = C$. Suppose that $C^{(r-1)}$, for $r = 1, \ldots, q-1$, has already been determined by:

$$C^{(r-1)} = \left[\begin{array}{c} C_1^{(0)} \\ \vdots \\ C_r^{(r-1)} \\ \hline C_{r+1,c}^{(r-1)} \end{array}\right] \quad \text{where} \quad C_{r+1,c}^{(r-1)} = \left[\begin{array}{c} C_{r+1}^{(r-1)} \\ \vdots \\ C_q^{(r-1)} \end{array}\right]. \tag{8}$$

To determine $C^{(r)}$, we define

$$C_{r+1,c}^{(r)} = C_{r+1,c}^{(r-1)} \left(I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)} \right), \tag{9}$$

and write

$$C^{(r)} = \left[\begin{array}{c} C_1^{(0)} \\ \vdots \\ C_r^{(r-1)} \\ \hline C_{r+1,c}^{(r)} \end{array}\right] \quad \text{where} \quad C_{r+1,c}^{(r)} = \left[\begin{array}{c} C_{r+1}^{(r)} \\ \vdots \\ C_q^{(r)} \end{array}\right]. \tag{10}$$

To define $C^{(r+1)}$, for $r = 0, \ldots, q-2$, the procedure in (9)–(10) is updated. Further, let

$$F_r = \left[\begin{array}{cccc|c} I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \\ \hline 0 & 0 & \cdots & -C_{r+1,c}^{(r-1)}[C_r^{(r-1)}]^\dagger & I \end{array}\right], \tag{11}$$

and $F = F_{q-1} \cdots F_1$.

THEOREM 2. *For, $r = 1, \ldots, q$, let $D_{C,rr} = C_r^{(r-1)}[C_r^{(r-1)}]^*$. Define*

$$D_C = \operatorname{diag}[D_{C,11}, \ldots, D_{C,qq}] \quad \textit{and} \quad D_C^\dagger = \operatorname{diag}[D_{C,11}^\dagger, \ldots, D_{C,qq}^\dagger]. \tag{12}$$

Then the matrix

$$Y_+ = AC^* F^* D_C^\dagger F, \tag{13}$$

satisfies $Y_+ CC^* = AC^*$ and hence also satisfies $\|A - Y_+ C\|_F = \min_Y \|A - YC\|$.

REMARK 1. *Matrix Y_+ is not necessarily equal to Y_0 where $\|Y_0\|_F = min\|Y\|_F$ subject to $YCC^* = AC^*$. The detailed notes 1–5 at the end of the following proof explain this observation.*

Proof. The first step is to define $C^{(0)} = C$ and explain the progressive block vector reduction

$$C = C^{(0)} = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(0)} \\ \vdots \\ C_{q-1}^{(0)} \\ C_q^{(0)} \end{array}\right] \quad \longrightarrow \quad C^{(q-1)} = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \vdots \\ C_{q-1}^{(q-2)} \\ C_q^{(q-1)} \end{array}\right],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

to a block vector of mutually orthogonal blocks satisfying $C_r^{(r-1)}[C_s^{(s-1)}]^* = 0$ for all $r \in \{1,\dots,q\}$ and $s \in \{1,\dots,q-1\}$ with $r > s$. Thus, the range $[C_s^{(s-1)}]^*(\mathbb{C}^n)$ of $[C_s^{(s-1)}]^*$ is a subspace of the null space $[C_r^{(r-1)}]^{-1}(\{\mathbf{0}\})$ of $C_r^{(r-1)}$ for all $r > s$. That is, $[C_s^{(s-1)}]^*(\mathbb{C}^n) \subseteq [C_r^{(r-1)}]^{-1}(\{\mathbf{0}\})$ for all $r > s$. At Stage 1, we write

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

and use the consolidated partition to define

$$F_1 = \left[\begin{array}{c|c} I & 0 \\ \hline -C_{2,c}^{(0)}[C_1^{(0)}]^\dagger & I \end{array}\right].$$

It follows that

$$C^{(1)} = F_1 C^{(0)} = \left[\begin{array}{c} C_1^{(0)} \\ \hline -C_{2,c}^{(0)}[C_1^{(0)}]^\dagger C_1^{(0)} + C_{2,c}^{(0)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ \hline C_{2,c}^{(1)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ \hline C_2^{(1)} \\ C_3^{(1)} \\ \vdots \\ C_q^{(1)} \end{array}\right],$$

where we have defined $C_{2,c}^{(1)} = C_{2,c}^{(0)}(I - [C_1^{(0)}]^\dagger C_1^{(0)})$. At Stage 2, we write

$$C^{(1)} = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \hline C_3^{(1)} \\ \vdots \\ C_q^{(1)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \hline C_{3,c}^{(1)} \end{array}\right],$$

and use the consolidated partition to define

$$F_2 = \left[\begin{array}{cc|c} I & 0 & 0 \\ 0 & I & 0 \\ \hline 0 & -C_{3,c}^{(1)}[C_2^{(1)}]^\dagger & I \end{array}\right].$$

It follows that

$$C^{(2)} = F_2 C^{(1)} = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \hline -C_{3,c}^{(1)}[C_2^{(1)}]^\dagger C_2^{(1)} + C_{3,c}^{(1)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \hline C_{3,c}^{(2)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \hline C_3^{(2)} \\ C_4^{(2)} \\ \vdots \\ C_q^{(2)} \end{array}\right],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

where we have defined $C_{3,c}^{(2)} = C_{3,c}^{(1)}(I - [C_2^{(1)}]^\dagger C_2^{(1)})$. In general, at Stage r, the matrices $C^{(r-1)}$ and F_r are defined, respectively, by the expressions in (8) and (11). It follows that

$$C^{(r)} = F_r C^{(r-1)} = \left[\begin{array}{c} C_1^{(0)} \\ \vdots \\ \hline C_r^{(r-1)} \\ \hline C_{r+1,c}^{(r-1)}(I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)}) \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ \vdots \\ \hline C_r^{(r-1)} \\ \hline C_{r+1,c}^{(r)} \end{array}\right] = \left[\begin{array}{c} C_1^{(0)} \\ \vdots \\ \hline C_r^{(r-1)} \\ \hline C_{r+1}^{(r)} \\ \vdots \\ C_q^{(r+1)} \end{array}\right],$$

where we have defined $C_{r+1,c}^{(r)} = C_{r+1,c}^{(r-1)}(I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)})$. The procedure continues until we obtain

$$C^{(q-1)} = \left[\begin{array}{c} C_1^{(0)} \\ C_2^{(1)} \\ \vdots \\ C_q^{(q-1)} \end{array}\right].$$

We can describe the final transformed matrix $C^{(q-1)}$ by the definitive relationships

$$C_{s+1}^{(s)} = C_{s+1}^{(s-1)}[I - \{C_s^{(s-1)}\}^\dagger C_s^{(s-1)}],$$

for each $s = 1, \ldots, q-1$. An important consequence of the reduction implemented by EBOS is that

$$\begin{aligned} C_{s+1,c}^{(s)}[C_s^{(s-1)}]^* &= C_{s+1,c}^{(s-1)}(I - [C_s^{(s-1)}]^\dagger C_s^{(s-1)})[C_s^{(s-1)}]^* \\ &= C_{s+1,c}^{(s-1)}(I - [C_s^{(s-1)}]^* \{C_s^{(s-1)}[C_s^{(s-1)}]^*\}^\dagger C_s^{(s-1)})[C_s^{(s-1)}]^* \\ &= C_{s+1,c}^{(s-1)}(I - [C_s^{(s-1)}]^* \{[C_s^{(s-1)}]^*\}^\dagger)[C_s^{(s-1)}]^* \\ &= 0, \end{aligned}$$

for each $s = 1, \ldots, q-1$. Therefore, $C_r^{(s-1)}[C_s^{(s-1)}]^* = 0$ for each $s = 1, \ldots, q-1$ and all $r = s+1, \ldots, q$. It follows that $C_{s+2,c}^{(s-1)}[C_{s+1}^{(s)}]^* = 0$ for each $s = 1, \ldots, q-2$ and hence we deduce that

$$C_{s+2,c}^{(s+1)}[C_s^{(s-1)}]^* = C_{s+2,c}^{(s)}(I - [C_{s+1}^{(s)}]^\dagger C_{s+1}^{(s)})[C_s^{(s-1)}]^* = 0,$$

from which it follows that $C_r^{(s+1)}[C_s^{(s-1)}]^* = 0$ for $r > s+1$ and in particular that $C_{s+2}^{(s+1)}[C_s^{(s-1)}]^* = 0$. By continuing this argument, we can show that $C_r^{(r-1)}[C_s^{(s-1)}]^* = 0$ for all $r > s$. It follows that $C_s^{(s-1)}[C_r^{(r-1)}]^* = 0$ for all $r > s$. Therefore,

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

If we define $F = F_{q-1} \cdots F_1$, then $FC = C^{(q-1)}$ and

$$FCC^*F^* = D_C. \tag{14}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

If we also define $Z = YF^{-1}$, then

$$YCC^* = AC^* \iff YF^{-1} \cdot FCC^*F^* = AC^*F^* \iff Z \cdot D_C = A[C^{(q-1)}]^*.$$

Let $\mathcal{Y} = \{Y \mid YCC^* = AC^*\}$ and $\mathcal{Z} = \{Z \mid ZD_C = A[C^{(q-1)}]^*\}$ denote the respective solution sets. We make the following observations:

1. $Y \in \mathcal{Y} \iff Z \in \mathcal{Z}$.
2. $Z_0 = A[C^{(q-1)}]^* D_C^\dagger \Longrightarrow Z_0 D_C = A[C^{(q-1)}]^* \iff Z_0F \cdot CC^* = AC^* \Longrightarrow Z_0F \in \mathcal{Y}$.
3. $Y_0 = AC^*(CC^*)^\dagger \Longrightarrow Y_0CC^* = AC^* \iff Y_0F^{-1}D_C = A[C^{(q-1)}]^* \Longrightarrow Y_0F^{-1} \in \mathcal{Z}$.
4. $\|Z_0\| = \min_{Z\in\mathcal{Z}} \|Z\| \not\Longrightarrow \|Z_0F\| = \min_{Y\in\mathbb{Y}} \|Y\| = \|Y_0\|$.
5. $\|Y_0\| = \min_{Y\in\mathbb{Y}} \|Y\| \not\Longrightarrow \|Y_0F^{-1}\| = \min_{Z\in\mathbb{Z}} \|Z\| = \|Z_0\|$.

In practice, we will calculate $Y_+ = Z_0F = A[C^{q-1)}]^*[D_C]^\dagger F = A[C^{(q-1)}]^\dagger F$. Note that in general $C^\dagger \neq [C^{(q-1)}]^\dagger F$. It follows from the observations above that $Y_+CC^* = AC^*$ and hence that

$$\|Z_0FC - A\| = \|Y_+C - A\| = \min_{Y\in\mathcal{S}} \|YC - A\| = \|Y_0C - A\|.$$

Therefore, $Z_0FC = Y_+C$ is an optimal reconstruction of A. Note that in general $Y_+ \neq Y_0$. ∎

4.3. Application of EBOS to solve $B^*BX = B^*Y$. We wish to find X_+ which satisfies $B^*BX_+ = B^*Y$ and hence minimizes $\|B^*Y - B^*BX\|_F$. The general solution to the equation $B^*BX = B^*Y$ is given by:

$$X = X_0 + [I - (B^*B)^\dagger]H, \tag{15}$$

where $X_0 = (B^*B)^\dagger B^*Y$ is the minimum norm solution and $H \in \mathbb{C}^{g\times h}$ is arbitrary. Our procedure for finding the proposed solution X_+ is as follows.

Define $B^{(0)} = B$. The transpose of the system $B^*BX = B^*Y \iff X^*B^*B = Y^*B$ has the same form as the system $YCC^* = AC^*$. Thus, we can implement an analogous reduction scheme. Consequently, we will restrict our remarks to a few key points. We simplify the block row vector B using right multiplication by a nonsingular elementary block matrix to implement the desired column operations. At Stage r, we obtain

$$B^{(r-1)} = \left[\begin{array}{ccc|c} B_1^{(0)} & \cdots & B_r^{(r-1)} & B_{r+1,c}^{(r-1)} \end{array}\right],$$

where $B_{r+1,c}^{(r-1)} = [B_{r+1}^{(r-1)} \cdots B_p^{(r-1)}]$ and we use the consolidated partitions to define

$$E_r = \left[\begin{array}{cccc|c} I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & -\{B_r^{(r-1)}\}^\dagger B_{r+1,c}^{(r-1)} \\ 0 & 0 & \cdots & 0 & I \end{array}\right].$$

Now we define $B^{(r)} = B^{(r-1)}E_r$. The procedure terminates with

$$B^{(p-1)} = [B_1^{(0)} B_2^{(1)} \cdots B_p^{(p-1)}].$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

The final transformed matrix satisfies the definitive relationships

$$B_{s+1}^{(r)} = [I - B_s^{(s-1)}\{B_s^{(s-1)}\}^\dagger]B_s^{(r)},$$

for each $s = 1, \ldots, p-1$ and each $r = s+1, \ldots, p-1$. Similar arguments to those used previously for the block row reduction of the matrix C can now be used to show that $[B_s^{(s-1)}]^* B_r^{(r-1)} = 0$ and $[B_r^{(r-1)}]^* B_s^{(s-1)} = 0$ for all $r > s$. Therefore,

$$[B^{(p-1)}]^* B^{(p-1)} = \text{diag}([B_1^{(0)}]^* B_1^{(0)}, [B_2^{(1)}]^* B_2^{(1)}, \cdots, [B_p^{(p-1)}]^* B_p^{(p-1)}) = D_B,$$

is a block diagonal matrix. If we define $E = E_1 \cdots E_{p-1}$, then $BE = B^{(p-1)}$ and $E^* B^* BE = D_B$. If we also define $W = E^{-1}X$, then

$$B^*BX = B^*Y \iff E^*B^*BE \cdot E^{-1}X = E^*B^*Y \iff D_B \cdot W = [B^{(p-1)}]^*Y.$$

We have the following theorem.

THEOREM 3. *Let $D_{B,rr} = [B_r^{(r-1)}]^* B_r^{(r-1)}$ for each $r = 1, \ldots, p$ and define*

$$D_B = \textit{diag}[D_{B,11}, \ldots, D_{B,pp}] \quad \textit{and} \quad {D_B}^\dagger = \textit{diag}[{D_{B,11}}^\dagger, \ldots, {D_{B,pp}}^\dagger], \tag{16}$$

then the matrix

$$X_+ = E{D_B}^\dagger E^* B^* Y, \tag{17}$$

satisfies $B^*BX_+ = B^*Y$ and hence minimizes $\|B^*Y - B^*BX\|_F$. If Y satisfies (13), then (17) solves the equation in (3). □

Proof. The transposed system $X^*B^*B = Y^*B \iff B^*BX = B^*Y$ has the same general form as the system $YCC^* = AC^*$. Therefore, the result follows by applying Theorem 4.3 to the transformed system. A direct proof along similar lines to the proof of Theorem 4.3 is left to the reader. □

Let $\mathcal{X} = \{X \mid B^*BX = B^*Y\}$ and $\mathcal{W} = \{W \mid D_B W = [B^{(p-1)}]^*Y\}$ denote the respective solution sets. We make the following observations:

1. $X \in \mathcal{X} \iff W \in \mathcal{W}$.
2. $W_0 = D_B^\dagger [B^{(p-1)}]^* Y \Rightarrow D_B W_0 = [B^{(p-1)}]^* Y \Leftrightarrow B^*B \cdot EW_0 = BY^* \Rightarrow EW_0 \in \mathcal{X}$.
3. $X_0 = (B^*B)^\dagger B^*Y \Rightarrow B^*BX_0 = B^*Y \Leftrightarrow D_B E^{-1} X_0 = [B^{(p-1)}]^*Y \Rightarrow E^{-1}X_0 \in \mathcal{W}$.
4. $\|W_0\| = \min_{W \in \mathcal{W}} \|W\| \Rightarrow \|EW_0\| = \min_{X \in \mathbb{X}} \|X\| = \|X_0\|$.
5. $\|X_0\| = \min_{X \in \mathbb{X}} \|X\| \Rightarrow \|E^{-1}X_0\| = \min_{W \in \mathbb{W}} \|W\| = \|W_0\|$.

In practice, we will define $W_0 = [D_B]^\dagger [B^{(p-1)}]^* Y = [B^{(p-1)}]^\dagger Y$ and define $X_+ = EW_0$. Note that in general $B^\dagger \neq E[B^{(p-1)}]^\dagger$. It follows from the observations above that $B^*BX_+ = B^*Y$ and hence that

$$\|BEW_0 - Y\| = \|BX_+ - Y\| = \min_{X \in \mathcal{X}} \|BX - Y\| = \|BX_0 - Y\|.$$

Therefore, $BEW_0 = BX_+$ is an optimal reconstruction of Y. □

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

4.4. EBOS algorithm. Here, we represent an algorithm for calculating X_+ as follows.

Algorithm 1 Calculate X_+

Require: $A, B, C, p, q, m, n, g_1, \ldots, g_p, h_1, \ldots, h_q$
Ensure: X_+
  for $r = 1, \ldots, q-1$ **do**
    Calculate $C^{(r-1)}, C^{(r)}_{r+1,c}, C^{(r)}, F_r, F$
  end for
  for $r = 1, \ldots, q$ **do**
    Calculate $D_{C,rr}$
  end for
  Calculate $D_C, D_C^\dagger, Y_+$
  for $r = 1, \ldots, p-1$ **do**
    Calculate $B^{(r-1)}, B^{(r)}_{r+1,c}, B^{(r)}, E_r, E$
  end for
  for $r = 1, \ldots, p$ **do**
    Calculate $D_{B,rr}$
  end for
  Calculate $D_B, D_B^\dagger, X_+$

4.5. Analysis of numerical load associated with EBOS. We wish to calculate $Y_+ = A[C^{(q-1)}]^* D_C F = A[\{C_1^{(0)}\}^\dagger, \cdots, \{C_q^{(q-1)}\}^\dagger]F$. We assume $h_r = h/q$ for each $r = 1, 2, \ldots, q$.

• **Number of flops to calculate $C^{(q-1)}$ and F:** Assume that $C^{(r-1)}$ is known. To find $C^{(r)}$, we need to calculate

$$[C_r^{(r-1)}]^\dagger = [C_r^{(r-1)}]^* \{C_r^{(r-1)}[C_r^{(r-1)}]^*\}^\dagger \text{ and } C^{(r)}_{r+1,c} = C^{(r-1)}_{r+1,c}(I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)}),$$

for each $r = 1, \ldots, q-1$. We have the following calculations:

1. $C_r^{(r-1)}[C_r^{(r-1)}]^*$ requires $2(h/q) \times n \times (h/q)$ flops;
2. $\{[C_r^{(r-1)}]^* C_r^{(r-1)}\}^\dagger$ requires $21(h/q)^3$ flops;
3. $[C_r^{(r-1)}]^\dagger = [C_r^{(r-1)}]^* \{C_r^{(r-1)}[C_r^{(r-1)}]^*\}^\dagger$ takes $2 \times n \times (h/q) \times (h/q)$ flops.
4. $[C_r^{(r-1)}]^\dagger C_r^{(r-1)}$ requires $2 \times n \times (h/q) \times n$ flops;
5. $I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)}$ involves only diagonal elements and requires n flops; and
6. $C^{(r)}_{r+1,c} = C^{(r-1)}_{r+1,c}(I - [C_r^{(r-1)}]^\dagger C_r^{(r-1)})$ requires $2(q-r)h/q \times n \times n$ flops.

Therefore, stage r requires $21h^3/q^3 + 4nh^2/q^2 + 2(q-r)n^2h/q + n + 2n^2h/q$ flops. The total number of flops for all stages $r = 1, \ldots, q-1$ to calculate $C^{(q-1)}$ is

$$\mathcal{N}_1 = [21h^3/q^3 + 4nh^2/q^2 + 2n^2h/q + n^2h + n](q-1).$$

We must also calculate $F = F_{q-1} \cdots F_1$. We assume $Q_{r-1} = F_{r-1} \cdots F_1$ is known. We can use the natural block structure to write $F_r = [F_{r,(i,j)}]_{i,j=1}^q, Q_r = [Q_{r,(i,j)}]_{i,j=1}^q$ where $F_{r,(i,j)}, Q_{r,(i,j)} \in \mathbb{C}^{(h/q)\times(h/q)}$ for each $r = 1, \ldots, q-1$ and each $i, j = 1, \ldots, q$. If we define

$$F_r(r+1:q, r+1:q) = \begin{bmatrix} I_{h/q} & 0 \\ -C^{(r-1)}_{r+1,c}[C_r^{(r-1)}]^\dagger & I_{(q-r)h/q} \end{bmatrix} \in \mathbb{C}^{(q-r)h/q \times (q-r+1)h/q},$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

and

$$Q_{r-1}(r:q,1:r)=\begin{bmatrix} Q_{r-1,(r,1)} & \cdots & Q_{r-1,(r,r)} \\ \vdots & \vdots & \vdots \\ Q_{r-1,(q,1)} & \cdots & Q_{r-1,(q,r)} \end{bmatrix} \in \mathbb{C}^{(q-r+1)h/q\times rh/q},$$

then to calculate $Q_r = F_r Q_{r-1}$ we need to only calculate

$$Q_r(r+1:q,1:r+1) = F_r(r+1:q,r+1:q)\, Q_{r-1}(r:q,1:r),$$

for each $r=2,\ldots,q-1$. This calculation requires $2(h/q)^3 r(q-r)(q-r+1)$ flops. The total number of flops for all stages is

$$\begin{aligned} \mathcal{N}_2 &= 2(h/q)^3 \sum_{r=2}^{q-1} r(q-r)(q-r+1) \\ &= (h/q)^3 \left[q^4/6 + q^3/3 - 13q^2/6 + 5q/3\right]. \end{aligned}$$

• **Number of flops to calculate $\boldsymbol{Y_+ = A[C^{(q-1)}]^* D_C{}^\dagger F}$:** We have already calculated $\{C_r^{(r-1)}\}^\dagger$ and so all matrices in the product are known. The respective dimensions are $m\times n$, $n\times h$ and $h\times h$. Therefore, the number of flops required by the product is

$$\mathcal{N}_3 = 2mnh + \min\{2mh^2, 2nh^2\}.$$

• **Number of flops used by EBOS to calculate $\boldsymbol{Y_+}$:** The total number of flops is given by $\mathcal{N} = \mathcal{N}_1 + \mathcal{N}_2 + \mathcal{N}_3$.

• **Number of flops used by the direct method to calculate $\boldsymbol{Y_0 = AC^\dagger}$:** We have $C\in\mathbb{C}^{h\times n}$ and so we calculate $C^\dagger = C^*[CC^*]^\dagger$. Calculating CC^* requires $2h^2n$ flops and calculating $[CC^*]^\dagger$ takes $21h^3$ flops. Calculating $C^*[CC^*]^\dagger$ takes another $2nh^2$ flops and finally $AC^\dagger$ takes a further $2mnh$ flops. Therefore, the total number of flops for the direct method[1] is $\mathcal{F} = 21h^3 + 4nh^2 + 2mnh$.

• **A typical comparison:** We wish to compare typical values for $\mathcal{F}$ and $\mathcal{N}$. Suppose $m=n$ and $h=\epsilon n$ where $\epsilon\in(0,1]$ and $q=5$. We assume n is large and so we only include the highest order terms in n^3. For EBOS $\mathcal{N}_1 \approx (21\epsilon^3/q^3 + 4\epsilon^2/q^2 + 2\epsilon/q + \epsilon)n^3$, $\mathcal{N}_2 \approx \epsilon^3[q/6 + 1/3 - 13/(6q) + 5/(3q^2)]n^3$ and $\mathcal{N}_3 = (2\epsilon + 2\epsilon^2)n^3$. For the direct method, $\mathcal{F} = (21\epsilon^3 + 4\epsilon^2 + 2\epsilon)n^3$.

Table 1 shows that if $m=n$ and $h=\epsilon n$, the advantage for EBOS is greatest when $\epsilon\in[0.75,1]$. This is not surprising because the direct calculation becomes much easier when h is relatively small.

Example 2. *In this example illustrate the advantage of EBOS over the direct method (4) for large matrices of particular sizes. We used the Matlab code in Section 7 to compare the computation times for solution of the equation $YCC^* = AC^*$ with different randomly chosen matrices A and C. In the code we have used the notation $h_j = dh = h/q$ for each $j=1,\ldots,q$. The random choice of matrices means that the computation times vary from one experiment to the next, but the variation is relatively small compared to the execution time. In the tests we used $m=n$ with $h=n \iff \epsilon = 1$ and various $n\in[1000,15000]$*

[1] This is not necessarily a true indication of the time taken by Matlab to compute $Y_0 = AC^\dagger$.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

TABLE 1
Number of flops for EBOS ($\mathcal{N}$) and the direct method ($\mathcal{F}$) to solve $YCC^* = AC^*$ when $m = n$ and $h = \epsilon n$. Note that EBOS is less effective when ϵ is small and q is large.

ϵ	q	$\mathcal{N}_1/n^3$	$\mathcal{N}_2/n^3$	$\mathcal{N}_3/n^3$	$\mathcal{N}/n^3$	$\mathcal{F}/n^3$	$\mathcal{N}/\mathcal{F}$
1	2	5.62	1.67	4	11.29	27	0.4182
1	3	2.89	2.80	4	9.69	27	0.3587
1	5	1.73	4.97	4	10.69	27	0.3961
1	10	1.26	10.13	4	15.39	27	0.5702
0.75	2	3.17	0.70	2.63	6.50	12.61	0.5153
0.75	3	1.83	1.18	2.63	5.63	12.61	0.4467
0.75	5	1.21	2.10	2.63	5.93	12.61	0.4704
0.75	10	0.93	4.28	2.63	7.83	12.61	0.6211
0.5	2	1.58	0.21	1.5	3.29	4.63	0.7106
0.5	3	1.04	0.35	1.5	2.89	4.63	0.6251
0.5	5	0.76	0.62	1.5	2.88	4.63	0.6231
0.5	10	0.61	1.27	1.5	3.38	4.63	0.7307
0.25	2	0.60	0.03	0.63	1.25	1.08	1.1636
0.25	3	0.46	0.04	0.63	1.13	1.08	1.0437
0.25	5	0.36	0.08	0.63	1.07	1.08	0.9880
0.25	10	0.30	0.16	0.63	1.09	1.08	1.0075

TABLE 2
Computation time in seconds for the direct method and EBOS with $m = n = h = 3000$.

$m = n = h$	dh	q	$t_{\rm d}$ (secs)	$t_{\rm e}$ (secs)	$t_{\rm e}/t_{\rm d}$
3000	1500	2	7.2597	6.9437	0.9565
3000	1000	3	7.2267	6.2511	0.8650
3000	750	4	7.5824	6.0482	0.7977
3000	600	5	6.8432	5.8089	0.8489
3000	500	6	7.3092	6.2032	0.8487
3000	375	8	7.4798	6.8610	0.9173
3000	300	10	7.4431	7.6427	1.0268

and $q \in [2, 10]$. Selected typical results are given in Tables 2, 3, and 4. We have not bothered to illustrate solution of the second equation $B^*BX = B^*Y$ because it is essentially the same procedure used for the first equation.

For optimal choices of q, the results in Tables 2, 3, and 4 suggest that EBOS may reduce computation time by as much as 40% compared to the direct method.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

Table 3
Computation time in seconds for the direct method and EBOS with $m = n = h = 9000$.

$m = n = h$	dh	q	t_{d} (secs)	t_{e} (secs)	$t_{\mathrm{e}}/t_{\mathrm{d}}$
9000	4500	2	215.9239	180.0193	0.8377
9000	3000	3	211.9872	155.5800	0.7339
9000	2250	4	222.0178	144.5274	0.6510
9000	1800	5	218.2690	138.2073	0.6332
9000	1500	6	217.9000	144.1707	0.6616
9000	1125	8	220.7669	155.2825	0.7034
9000	900	10	211.7547	173.5781	0.8197

Table 4
Computation time in seconds for the direct method and EBOS with $m = n = h = 15000$.

$m = n = h$	dh	q	t_{d} (secs)	t_{e} (secs)	$t_{\mathrm{e}}/t_{\mathrm{d}}$
15000	7500	2	1058.4895	930.0785	0.8787
15000	5000	3	1128.0494	827.3659	0.7334
15000	3750	4	1051.1613	706.4415	0.6721
15000	3000	5	1031.2660	691.3852	0.6704
15000	2500	6	1055.1534	706.5885	0.6697
15000	1875	8	1178.7413	843.8233	0.7159
15000	1500	10	1054.4458	903.6170	0.8570

Example 3. *Details of the EBOS computation for a particular choice of small matrices. Let*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}, \; B = [B_1 \mid B_2] = \left[\begin{array}{cc|cccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}\right],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

$$C = \left[\begin{array}{c} C_1 \\ \hline C_2 \\ \hline C_3 \end{array}\right] = \left[\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{array}\right].$$

Solution of $YCC^* = AC^*$ *We calculate*

$$F_1 = \left[\begin{array}{c|c} I & 0 \\ \hline -C_{2c}^{(0)}[C_1^{(0)}]^\dagger & I \end{array}\right] = \left[\begin{array}{cc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline -6/11 & -9/11 & 1 & 0 & 0 & 0 & 0 \\ -4/11 & -6/11 & 0 & 1 & 0 & 0 & 0 \\ -8/11 & -1/11 & 0 & 0 & 1 & 0 & 0 \\ -5/11 & -2/11 & 0 & 0 & 0 & 1 & 0 \\ -8/11 & -1/11 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

and $C^{(1)} = F_1 C$ *giving*

$$C^{(1)} = \left[\begin{array}{cc|cccccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline -5/11 & 2/11 & 5/11 & 2/11 & -4/11 & 5/11 & 0 & 0 \\ -4/11 & 5/11 & -4/11 & -6/11 & 1/11 & 7/11 & 0 & 0 \\ 3/11 & -1/11 & -8/11 & -1/11 & 2/11 & 3/11 & 1 & 1 \\ -5/11 & 9/11 & 6/11 & -2/11 & -7/11 & 6/11 & 0 & 0 \\ -8/11 & -1/11 & 3/11 & -1/11 & 2/11 & 3/11 & 0 & 0 \end{array}\right].$$

Next, we calculate

$$F_2 = \left[\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & I & 0 \\ \hline 0 & -C_{3c}^{(1)}[C_2^{(1)}]^\dagger & I \end{array}\right] = \left[\begin{array}{cc|ccc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & -26/25 & -2/5 & 2/25 & 1 & 0 \\ 0 & 0 & -13/25 & -1/5 & 1/25 & 0 & 1 \end{array}\right],$$

and $C^{(2)} = F_2 C^{(1)}$ *giving*

$$C^{(2)} = \left[\begin{array}{cc|ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline -6/11 & 2/11 & 5/11 & 2/11 & -4/11 & 5/11 & 0 & 0 \\ -4/11 & 5/11 & -4/11 & -6/11 & 1/11 & 7/11 & 0 & 0 \\ 3/11 & -1/11 & -8/11 & -1/11 & 2/11 & 3/11 & 1 & 1 \\ \hline 7/25 & 11/25 & 4/25 & -4/25 & -7/25 & -4/25 & 2/25 & 2/25 \\ -9/25 & -7/25 & 2/25 & -2/25 & 9/25 & -2/25 & 1/25 & 1/25 \end{array}\right].$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

Finally, we calculate $Y_+ = A[C^{(2)}]^*[D_C]^\dagger F_2 F_1 = A[C^{(2)}]^\dagger F$ *giving*

$$Y_+ = \left[\begin{array}{rr|rrrrr} 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 1 & 2 & -2 & -1 & 0 & 1 & 1 \\ -1 & 0 & 1 & -1 & 1 & 1 & 0 \\ 0.5 & -0.5 & 0.5 & 0.5 & 0.5 & -0.5 & -0.5 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & -0.5 & 0.5 & 0.5 & 0.5 & 0.5 & -0.5 \\ -0.5 & -0.5 & 1.5 & -0.5 & 0.5 & 0.5 & -0.5 \\ 1 & 0 & 1 & 0 & 0 & 0 & -2 \\ 0.5 & 2.5 & -2.5 & -0.5 & 0.5 & 1.5 & 0.5 \\ 0 & 1 & -1 & -1 & 1 & 1 & 0 \\ -1 & 1 & -1 & 0 & 1 & 1 & 1 \\ 0.5 & 1.5 & -0.5 & -1.5 & 0.5 & 1.5 & -0.5 \end{array}\right].$$

Solution of $\boldsymbol{B^*BX = B^*Y_+}$. *Define*

$$E_1 = \left[\begin{array}{rr|rrrr} 1 & 0 & -0.4737 & -0.4737 & 0.2632 & -0.1579 \\ 0 & 1 & -0.1579 & -0.1579 & -0.5789 & -0.0526 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

and calculate $B^{(1)} = BE_1$ *giving*

$$B^{(1)} = \left[\begin{array}{rr|rrrr} 1 & 1 & -0.6316 & -0.6316 & -0.3158 & -0.2105 \\ 0 & 1 & 0.8421 & -0.1579 & -0.5789 & 0.9474 \\ 0 & 1 & -0.1579 & -0.1579 & 0.4211 & -0.0526 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0.3684 & 0.3684 & -0.3158 & -0.2105 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -0.6316 & -0.6316 & -0.3158 & -0.2105 \\ 0 & 1 & -0.1579 & 0.8421 & 0.4211 & -0.0526 \\ 1 & 1 & 0.3684 & 0.3684 & 0.6842 & -0.2105 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0.5263 & 0.5263 & 0.2632 & 0.8421 \end{array}\right].$$

Define $E = E_1$. *Now* $X_+ = E[D_B]^\dagger[B^{(1)}]^* Y = E[B^{(1)}]^\dagger Y$ *which gives*

$$X_+ = \left[\begin{array}{rr|rrr|rr} -0.1987 & 0.1009 & 0.4826 & -0.5599 & 0.4527 & 0.3691 & -0.7823 \\ 0.2934 & 0.4700 & -0.2839 & 0.0647 & -0.0016 & 0.3123 & 0.1073 \\ \hline 0.4621 & -0.5284 & 0.0205 & 0.2981 & -0.0804 & -0.5726 & -0.0300 \\ -0.0126 & -0.0095 & 0.1735 & 0.0994 & -0.0268 & 0.3091 & -0.1767 \\ -0.5394 & 0.8454 & -0.8328 & -0.8770 & 0.7287 & 0.7161 & 0.4479 \\ 0.0110 & 1.3833 & -1.0268 & -0.7744 & 0.3360 & 0.9795 & 0.6546 \end{array}\right].$$

The partitions indicate the individual $X_{+,ij}$ *for* $i = 1, 2$ *and* $j = 1, 2, 3$. □

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

5. Conclusion. In this paper, we have developed a method called the EBOS for calculating the optimal approximation of an observed matrix $A \in \mathbb{C}^{m\times n}$. We targeted the case when matrix A was large. The approximant is represented by $\sum_{i=1}^{p}\sum_{j=1}^{q} B_i X_{ij} C_j$ where each $B_i \in \mathbb{C}^{m\times g_i}$ and $C_j \in \mathbb{C}^{h_j\times n}$ is known and matrix X_{ij} is unknown. Each X_{ij} should be determined from the solution of the problem of minimizing the Frobenius norm of the error. The problem is motivated by the task in optimal data processing. Our solutions are motivated by the observation that, for large A, the known approaches would involve considerable computational burden needed for calculation of large pseudo-inverse matrices. This would make computation unwieldy and inefficient.

The idea of our approach is to reduce the solution procedure to finding the pseudo-inverses of a collection of much smaller matrices. In EBOS, this idea is realized by using a sequence of elementary block row operations to reduce the associated systems to block diagonal form. As a result, the computational time is reduced by as much as 40%. The proposed scheme can be easily implemented using the standard general-purpose MATLAB algorithms.

REFERENCES

[1] F. Artoni, A. Delorme, and S. Makeig. Applying dimension reduction to EEG data by Principal Component Analysis reduces the quality of its subsequent Independent Component decomposition. *Neuroimage*, 175:176–187, 2018.
[2] 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.
[3] O.M. Baksalary and G. Trenkler. On formulae for the Moore–Penrose inverse of a columnwise partitioned matrix. *Appl. Math. Computat.*, 403:125913, 2021. https://doi.org/10.1016/j.amc.2020.125913.
[4] R.E. Bellman. *Dynamic Programming* (2nd edition). Courier Corporation, 2003.
[5] A. Ben-Israel. Generalized inverses of matrices: A perspective of the work of Penrose. *Math. Proc. Cambridge Philos. Soc.*, 100(3):407–425, 1986. https://doi.org/10.1017/S0305004100066172.
[6] 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.
[7] C. Boutsidis and D.P. Woodruff. Optimal CUR matrix decompositions. *SIAM J. Comput.*, 46(2), 543–589, 2017. http://dx.doi.org/10.1145/2591796.2591819.
[8] J. Chiu and L. Demanet. Sublinear randomized algorithms for skeleton decompositions. *SIAM J. Matrix Anal. Appl.*, 34:1361–1383, 2013. https://doi.org/10.1137/110852310.
[9] P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30:844–881. https://doi.org/10.1137/07070471X.
[10] 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.
[11] 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.
[12] 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.
[13] J.G.F. Francis. The QR Transformation, I. *Comput. J.*, 4(3):265–271, 1961. https://doi.org/10.1093/comjnl/4.3.265.
[14] J.G.F. Francis. The QR Transformation, II. *Comput. J.*, 4(4):332–345, 1962. https://doi.org/10.1093/comjnl/4.4.332.
[15] S. Friedland and A. Torokhti. Generalized rank-constrained matrix approximations. *SIAM J. Matrix Analy. Appl.*, 29, 656–659, 2007. https://doi.org/10.1137/06065551.
[16] N. Gillis and Y. Shitov. Low-rank matrix approximation in the infinity norm. *Linear Algebra Appl.*, 481:367–382, 2019. https://doi.org/10.1016/j.laa.2019.07.017.
[17] 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.
[18] 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.
[19] 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.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

[20] 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.
[21] G.H. Golub and C.F. Van Loan. *Matrix Computations* 3^{th} *Ed.*, Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.
[22] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin. A theory of pseudo-skeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997. https://doi.org/10.1016/S0024-3795(96)00301-1.
[23] N. Goela and M. Gastpar. Reduced-Dimension Linear Transform Coding of Correlated Signals in Networks. *IEEE Trans. Sig. Process.*, 60(6):3174–3187, 2012. https://doi.org/10.1109/TSP.2012.2188716.
[24] 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.
[25] H. Hotelling. Analysis of a complex of statistical variables into principal components (1). *Journal of Educational Psychology*, 24(6):417–441, 1933. https://doi.org/10.1037/h0071325.
[26] H. Hotelling. Analysis of a complex of statistical variables into principal components (2). *Journal of Educational Psychology*, 24(7):498–520, 1933. https://doi.org/10.1037/h0070888.
[27] 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.
[28] P. Howlett, A. Torokhti, P. Pudney, and P. Soto-Quiros. Multilinear Karhunen-Loève transforms. *IEEE Trans. Sig. Process.*, 70:5148–5163, 2022. https://doi.org/10.1109/TSP.2022.3214684.
[29] 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. https://doi.org/10.1017/S1446181100012888.
[30] 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.
[31] Y. Hua and W.Q. Liu. Generalized Karhunen-Loeve transform. *IEEE Signal Process. Lett.*, 5:141–142, 1998. https://doi.org/10.1109/97.681430.
[32] V.N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computat. Math. Math. Phys.*, 1(3): 637–657, 1962. https://doi.org/10.1016/0041-5553(63)90168-X.
[33] M. Leclercq, B. Vittrant, M.L. Martin-Magniette, M.P.S. Boyer, O. Perin, A. Bergeron, Y. Fradet, and A. Droit. Large-scale automatic feature selection for biomarker discovery in high-dimensional OMICs data. *Front. Genet.*, 10, 2019. https://doi.org/10.3389/fgene.2019.00452.
[34] X. Liu, W. Li, and H. Wang. Rank constrained matrix best approximation problem with respect to (skew) Hermitian matrices. *J. Computat. Appl. Math.*, 319:77–86, 2017. http://dx.doi.org/10.1016/j.cam.2016.12.029.
[35] D.G. Luenberger. *Optimization by Vector Space Methods*. Wiley, 1968. Reprinted in 1997.
[36] 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.
[37] 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.
[38] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51:406–413, 1955. https://doi.org/10.1017/S0305004100030401.
[39] 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.
[40] 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.
[41] P.H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966. https://doi.org/10.1007/BF02289451.
[42] P. Soto-Quiros, J. Chavarria-Molina, J. Fallas-Monge, and A. Torokhti. Fast multiple rank-constrained matrix approximation. (submitted). *SeMA J. Bull. Span. Soc. Appl. Math.*
[43] 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.
[44] A.Torokhti and P. Howlett. Optimal estimation of distributed highly noisy signals within KLT-Wiener archetype. *Digital Sig. Process.,* 143, Article no. 104225:1–10, 2023. https://doi.org/10.1016/j.dsp.2023.104225.
[45] A. Torokhti and P. Soto-Quiros. Optimal modeling of nonlinear systems: Method of variable injections. *Proyecciones J. Math.*, 43(1):163–198, 2024.
[46] H. Wang. Rank constrained matrix best approximationproblem. *Appl. Math. Lett.*, 50:98–104, 2015. https://doi.org/10.1016/j.aml.2015.06.009.
[47] H. Weyl. Das asymptotische Verteilungsgesetz der Eigenwert linearer partieller Differentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung). *Math. Ann.*, 71:441–479, 1912.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.

[48] 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.
[49] G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6:49–53, 1941. https://doi.org/10.1007/bF02288574.

Appendix A. MATLAB code for comparison of the direct method and EBOS.

```
clear all \\

%solve Y *C*C' = A*C', size(A) = (m,n), size(C) = (q*dh,n). \\

%define the parameters \\

m = 18; n = 18; dh = 3; q = 6; \\
$\%$m = 3000; n=3000; dh = 750; q = 4; \\

%define the random matrices \\

tic \\
A = rand(m, q*dh)*rand(q*dh,n); \\
C = rand(q*dh,n); \\
trand = toc \\ }

%set C0 = C for checking

C0 = C;

%calculate Y by the direct method

tic
Ydirect = A*pinv(C);
tdirect = toc
edirect = A - Ydirect*C;
ndirect = max(max(abs(edirect)))

%calculate Y by EBOS

C = C0;
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 r=2:q-1
```

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 40, pp. 585-605, September 2024.


```
F(r*dh+1:q*dh,1:r*dh) = [- C(r*dh+1:q*dh,:)*pinv(C((r-1)*dh+1:r*dh,:)),eye
    ((q-r)*dh)]*...
     F((r-1)*dh+1:q*dh,1:r*dh);
C(r*dh+1:q*dh,:) = C(r*dh+1:q*dh,:)*(eye(n) - pinv(C((r-1)*dh+1:r*dh,:))*C
    ((r-1)*dh+1:r*dh,:));
D((r-1)*dh+1:r*dh,(r-1)*dh+1:r*dh) = C((r-1)*dh+1:r*dh,:)*C((r-1)*dh+1:r*
    dh,:)';
Ddag = blkdiag(Ddag,pinv(D((r-1)*dh+1:r*dh,(r-1)*dh+1:r*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 r=1:q
     ACpDd = [ACpDd ACp(:,(r-1)*dh+1:r*dh)*Ddag((r-1)*dh+1:r*dh,(r-1)*dh+1:
        r*dh)];
end
Yebos = [];
for r=1:q
Yebos = [Yebos ACpDd(1:m,(r-1)*dh+1:q*dh)*F((r-1)*dh+1:q*dh,(r-1)*dh+1:r*
    dh)];
end
tebos = toc
eebos = Yebos*C0 - A;
nebos = max(max(abs(eebos)))
```