A procedure is reported for the compression of rank-deficient matrices. A matrix A of rank k is represented in the form $A = U \circ B \circ V$, where B is a $k \times k$ submatrix of A, and U, V are well-conditioned matrices that each contain a $k \times k$ identity submatrix. This property enables such compression schemes to be used in certain situations where the SVD cannot be used efficiently. Numerical examples are presented.

On the compression of low rank matrices

H. Cheng[†], Z. Gimbutas[†], P.G. Martinsson[‡], V. Rokhlin[‡]

Research Report YALEU/DCS/RR-1251
July 11, 2003

This research was supported in part by the Defense Advanced Research Projects Agency under contract #MDA972-00-1-0033, and by the Office of Naval Research under contract #N00014-01-0364.

[†] MadMax Optics Inc., 3035 Whitney Ave., Hamden CT 06518
[‡] Dept. of Mathematics, Yale University, New Haven CT 06511

Keywords: *Matrix compression, skeletons, model order reduction.*

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
11 JUL 2003		**00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
On the compression of low rank matrices	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Yale University,Department of Mathematics,New Haven,CT,06520	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A procedure is reported for the compression of rank-deficient matrices. A matrix A of rank k is represented in the form A = U o B o V where B is a k x k submatrix of A, and U, V are well-conditioned matrices that each contain a k x k identity submatrix. This property enables such compression schemes to be used in certain situations where the SVD cannot be used efficiently. Numerical examples are presented.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**16**	

On the compression of low rank matrices

H. Cheng, Z. Gimbutas, P.G. Martinsson, V. Rokhlin

Abstract: A procedure is reported for the compression of rank-deficient matrices. A matrix A of rank k is represented in the form $A = U \circ B \circ V$, where B is a $k \times k$ submatrix of A, and U, V are well-conditioned matrices that each contain a $k \times k$ identity submatrix. This property enables such compression schemes to be used in certain situations where the SVD cannot be used efficiently. Numerical examples are presented.

1. Introduction

In computational physics (and many other areas), one often encounters matrices whose ranks are (to high precision) much lower than their dimensionalities; even more frequently, one is confronted with matrices posessing large submatrices that are of low rank. An obvious source of such matrices is the potential theory, where discretization of integral equations almost always results in matrices of this type. Such matrices are also encountered in fluid dynamics, numerical simulation of electromagnetic phenomena, structural mechanics, multivariate statistics etc. In such cases, one is tempted to "compress" the matrices in question, so that they could be efficiently applied to arbitrary vectors; compression also facilitates the storage and any other manipulation of such matrices that might be desirable.

At this time, several classes of algorithms exist that use this observation. The so-called Fast Multipole Methods (FMMs) are algorithms for the application of certain classes of matrices to arbitrary vectors; FMMs tend to be extremely efficient, but are only applicable to very narrow classes of operators (see [7]). Another approach to the compression of operators is based on the wavelets and related structures (see, for example, [3, 2]); these schemes exploit the smoothness of the elements of the matrix viewed as a function of their indices, and tend to fail for highly oscillatory operators.

Finally, there is a class of compression schemes that are based purely on linear algebra, and are completely insensitive the the analytical origin of the operator. It consists of the Singular Value Decomposition (SVD), the so-called QR and QLP factorizations [8], and several others. Given an $m \times n$-matrix A of rank $k < \min(m, n)$, the SVD represents A in the form

$$A = U \circ D \circ V, \tag{1.1}$$

with U an $m \times k$, matrix whose columns are orthonormal, V a $k \times n$ matrix whose rows are orthonormal, and D a diagonal matrix whose diagonal elements are positive. The compression provided by the SVD is perfect in terms of accuracy (see, for example, [5]), and has a simple geometric interpretation: it expresses each of the columns of A as a linear combination of the k (orthonormal) columns of U; it also represents the rows of A as linear combinations of (orthonormal) rows of V; and the matrices U, V are chosen in such a manner that the rows of U are images (up to a scaling) under A of the columns of V.

In this paper, we propose a different matrix decomposition. Specifically, we represent the matrix A described above in the form

$$A = \mathcal{U} \circ B \circ \mathcal{V}, \tag{1.2}$$

where B is a $k \times k$-submatrix of A, and the norms of the matrices $\mathcal{U}, \mathcal{V}$ (of dimensionalities $n \times k$, $k \times m$ respectively) are reasonably close to 1 (see Theorem 3 in Section 3 below). Furthermore, each of the matrices $\mathcal{U}, \mathcal{V}$ contains a unity $k \times k$ submatrix.

Like (1.1), the representation (1.2) has a simple geometric interpretation: it expresses each of the columns of A as a linear combination of k selected columns of A, and each of the rows of A as a linear combination of k selected rows of A. This selection defines a $k \times k$ submatrix B of A, and in the resulting system of coordinates, the action of A is represented by the action of its submatrix B.

The representation (1.2) has the advantage that the bases used for the representation of the mapping A consists of the columns and rows of A, while each of the elements of the bases in the representation (1.1) is itself a linear combination of *all* rows (or columns) of the matrix A. In Section 5, we illustrate the advantages of the representation (1.2) by constructing an accelerated direct solver for integral equations of potential theory.

Another advantage of the representation (1.2) is that the numerical procedure for constructing it is considerably less expensive than that for the construction of the SVD (see Section 4), and that the cost of applying (1.2) to an arbitrary vector is

$$(n + m - k) \cdot k, \tag{1.3}$$

vs.

$$(n + m) \cdot k \tag{1.4}$$

for the SVD.

The obvious disadvantage of (1.2) vis-a-vis (1.1) is the fact that the norms of the the matrices $\mathcal{U}, \mathcal{V}$ are somewhat greater than 1, leading to some (though minor) loss of accuracy. Another disadvantage of the proposed factorization is its non-uniqueness; in this respect it is similar to the pivoted QR factorization.

Remark 1. In (1.2), the submatrix B of the matrix A is defined as the intersection of k columns with k rows. Denoting the sequence numbers of the rows by $i_1, i_2, \ldots, i_k$ and the sequence numbers of the columns by $j_1, j_2, \ldots, j_k$, we will be referring to the submatrix B of A as the skeleton of A, to the $k \times n$ matrix consisting of the rows of A numbered $i_1, i_2, \ldots, i_k$ as the row skeleton of A, and to the $m \times k$ matrix consisting of the columns of A numbered $j_1, j_2, \ldots, j_k$ as the column skeleton of A.

The structure of this paper is as follows. Section 2 below summarizes several facts from numerical linear algebra to be used in the remainder of the paper. In Section 3, we prove the existence of a stable factorization of the form (1.2). In Section 4, we describe a reasonably efficient numerical algorithm for constructing such a factorization. In Section 5, we illustrate how the geometric properties of the factorization (1.2) can be utilized to construct an accelerated direct solver for integral equations of potential theory. In Section 6, we present the results of numerical experiments with the direct solver. Finally, Section 7 contains a discussion of other possible applications of the techniques of this paper.

2. Preliminaries

In this section we introduce our notation and summarize several facts from numerical linear algebra; these can all be found in [1].

Throughout the paper, we use upper case letters for matrices and lower case letters for vectors and scalars. We reserve Q for matrices that have orthonormal columns and P for permutation matrices. The canonical unit vectors in $\mathbb{C}^n$ are denoted by e_j. Given a matrix X, we let $X^\star$ denote its adjoint (the complex conjugate transpose), $\sigma_k(X)$ its k-th singular value, $||X||_2$ its l^2-norm and $||X||_{\mathrm{F}}$ its Frobenius norm. Finally, given matrices A, B, C and D we let

$$[A|\,B]\,, \qquad \left[\frac{A}{C}\right], \qquad \text{and} \qquad \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \tag{2.1}$$

denote larger matrices obtained by stringing the blocks A, B, C and D together.

The first result that we present asserts that given any matrix A, it is possible to reorder its columns to form a matrix AP, where P is a permutation matrix, with the following property: When AP is factorized into an orthonormal matrix Q and an upper triangular matrix R, so that $AP = QR$, then the singular values of the leading $k \times k$ submatrix of R are reasonably good approximations of the first k singular values of A. The theorem also says that the first k columns of AP form a well-conditioned basis for the column space of A to within accuracy $\sigma_{k+1}(A)$.

Theorem 1. [Gu & Eisenstat] *Suppose that A is an $m \times n$ matrix, $l = \min(m,n)$, and k is an integer such that $1 \leq k \leq l$. Then there exists a factorization*

$$AP = QR, \tag{2.2}$$

where P is an $n \times n$ permutation matrix, Q is an $m \times l$ matrix with orthonormal columns, and R is an $l \times n$ upper triangular matrix. Furthermore, splitting Q and R,

$$Q = \left[\begin{array}{c|c} Q_{11} & Q_{12} \\ \hline Q_{21} & Q_{22} \end{array}\right], \qquad \text{and} \qquad R = \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & R_{22} \end{array}\right], \tag{2.3}$$

in such a fashion that Q_{11} and R_{11} are of size $k \times k$, Q_{21} is $(m-k) \times k$, Q_{12} is $k \times (l-k)$, Q_{22} is $(m-k) \times (l-k)$, R_{12} is $k \times (n-k)$ and R_{22} is $(l-k) \times (n-k)$, results in the following inequalities:

$$\sigma_k(R_{11}) \geq \sigma_k(A) \frac{1}{\sqrt{1+k(n-k)}}, \tag{2.4}$$

$$\sigma_1(R_{22}) \leq \sigma_{k+1}(A) \sqrt{1+k(n-k)}, \tag{2.5}$$

and

$$||R_{11}^{-1} R_{12}||_{\mathrm{F}} \leq \sqrt{k(n-k)}. \tag{2.6}$$

Remark 2. In this paper we do not use the full power of Theorem 1 since we are only concerned with the case of very small $\varepsilon = \sigma_{k+1}(A)$. In this case, the inequality (2.5) implies that A can be well approximated by a low-rank matrix. In particular, (2.5) implies that

$$||A - \left[\frac{Q_{11}}{Q_{21}}\right] [R_{11}|\,R_{12}]\, P^\star||_2 \leq \varepsilon \sqrt{1+k(n-k)}. \tag{2.7}$$

Furthermore, the inequality (2.6) in this case implies that the first k columns of AP form a well-conditioned basis for the entire column space of A (within accuracy ε).

While Theorem 1 asserts the existence of a factorization (2.2) with the properties (2.4), (2.5), (2.6), it says nothing about the cost of constructing such a factorization numerically. The following theorem asserts that a factorization that satisfies bounds that are weaker than (2.4), (2.5), (2.6) by a factor of $\sqrt{n}$ can be computed in $O(mn^2)$ operations.

Theorem 2. **[Gu & Eisenstat]** *Given an $m \times n$ matrix A, a factorization of the form (2.2) that instead of (2.4), (2.5) and (2.6) satisfies the inequalities*

$$\sigma_k(R_{11}) \geq \frac{1}{\sqrt{1+nk(n-k)}}\sigma_k(A), \tag{2.8}$$

$$\sigma_1(R_{22}) \leq \sqrt{1+nk(n-k)}\sigma_{k+1}(A), \tag{2.9}$$

and

$$||R_{11}^{-1}R_{12}||_{\mathrm{F}} \leq \sqrt{nk(n-k)}, \tag{2.10}$$

can be computed in $O(mn^2)$ operations.

3. Analytical apparatus

In this section we prove that the factorization (1.2) exists by applying Theorem 1 to both the columns and the rows of the matrix A. Theorem 2 then guarantees that the factorization can be computed efficiently.

The following theorem is the principal analytic tool of this paper.

Theorem 3. *Suppose that A is an $m \times n$ matrix and let k be such that $1 \leq k \leq \min(m,n)$. Then there exists a factorization*

$$A = P_{\mathrm{L}} \begin{bmatrix} I \\ \hline S \end{bmatrix} A_{\mathrm{S}} \left[I \middle| T \right] P_{\mathrm{R}}^{\star} + X, \tag{3.1}$$

where $I \in \mathbb{C}^{k\times k}$ is the identity matrix, P_{L} and P_{R} are permutation matrices, and A_{S} is the top left $k \times k$ submatrix of $P_{\mathrm{L}}^{\star} A P_{\mathrm{R}}$. In (3.1), the matrices $S \in \mathbb{C}^{(m-k)\times k}$ and $T \in \mathbb{C}^{k\times(n-k)}$ satisfy the inequalities

$$||S||_{\mathrm{F}} \leq \sqrt{k(m-k)}, \qquad \text{and} \qquad ||T||_{\mathrm{F}} \leq \sqrt{k(n-k)}, \tag{3.2}$$

and the matrix X is small if the $(k+1)$-th singular value of A is small,

$$||X||_2 \leq \sigma_{k+1}(A)\sqrt{1+k(\min(m,n)-k)}. \tag{3.3}$$

Proof: The proof consists of two steps. First Theorem 1 is invoked to assert the existence of k columns of A that form a well-conditioned basis for the column space within accuracy $\sigma_{k+1}(A)$; these are collected in the $m \times k$ matrix A_{CS}. Then Theorem 1 is invoked again to prove that k of the rows of A_{CS} form a well-conditioned basis for its row space. Without loss of generality, we assume that $m \geq n$ and that $\sigma_k(A) \neq 0$.

For the first step we factor A into matrices Q and R as specified by Theorem 1, letting P_{R} denote the permutation matrix. Splitting Q and R into submatrices Q_{ij} and R_{ij} as in (2.3), we reorganize the factorization (2.2) as follows,

$$AP_{\mathrm{R}} = \begin{bmatrix} Q_{11} \\ \hline Q_{21} \end{bmatrix} \left[R_{11} \middle| R_{12} \right] + \begin{bmatrix} Q_{12} \\ \hline Q_{22} \end{bmatrix} \left[0 \middle| R_{22} \right] = \begin{bmatrix} Q_{11}R_{11} \\ \hline Q_{21}R_{11} \end{bmatrix} \left[I \middle| R_{11}^{-1}R_{12} \right] + \left[\begin{array}{c|c} 0 & Q_{12}R_{22} \\ \hline 0 & Q_{22}R_{22} \end{array}\right]. \tag{3.4}$$

We now define the matrix $T \in \mathbb{C}^{k\times(n-k)}$ via the formula

$$T = R_{11}^{-1}R_{12}; \tag{3.5}$$

T satisfies the inequality (3.2) by virtue of (2.6). We define the matrix $X \in \mathbb{C}^{m\times n}$ via the formula

$$X = \left[\begin{array}{c|c} 0 & Q_{12}R_{22} \\ \hline 0 & Q_{11}R_{22} \end{array}\right] P_{\mathrm{R}}^{\star}, \tag{3.6}$$

which satisfies the inequality (3.3) by virtue of (2.5). Defining the matrix $A_{\mathrm{CS}} \in \mathbb{C}^{m\times k}$ by

$$A_{\mathrm{CS}} = \left[\begin{array}{c} Q_{11}R_{11} \\ \hline Q_{21}R_{11} \end{array}\right], \tag{3.7}$$

we reduce equation (3.4) to the form

$$AP_{\mathrm{R}} = A_{\mathrm{CS}} \left[I \middle| T\right] + XP_{\mathrm{R}}. \tag{3.8}$$

An obvious interpretation of (3.8) is that A_{CS} consists of the first k columns of the matrix AP_{R} (since the corresponding columns of XP_{R} are identically zero).

The second step of the proof is to find k rows of A_{CS} forming a well-conditioned basis for its row-space. To this end, we factor the transpose of A_{CS} as specified by Theorem 1,

$$A_{\mathrm{CS}}^{\star}P_{\mathrm{L}} = \tilde{Q}\left[\tilde{R}_{11}\middle|\,\tilde{R}_{12}\right]. \tag{3.9}$$

Transposing (3.9) and rearranging the terms we have

$$P_{\mathrm{L}}^{\star}A_{\mathrm{CS}} = \left[\begin{array}{c} \tilde{R}_{11}^{\star} \\ \hline \tilde{R}_{12}^{\star} \end{array}\right] \tilde{Q}^{\star} = \left[\begin{array}{c} I \\ \hline \tilde{R}_{12}^{\star}(\tilde{R}_{11}^{\star})^{-1} \end{array}\right] \tilde{R}_{11}^{\star}\tilde{Q}^{\star}. \tag{3.10}$$

Multiplying (3.8) by $P_{\mathrm{L}}^{\star}$ and using (3.10) to substitute for $P_{\mathrm{L}}^{\star}A_{\mathrm{CS}}$ we obtain

$$P_{\mathrm{L}}^{\star}AP_{\mathrm{R}} = \left[\begin{array}{c} I \\ \hline \tilde{R}_{12}^{\star}(\tilde{R}_{11}^{\star})^{-1} \end{array}\right] \tilde{R}_{11}^{\star}\tilde{Q}^{\star}\left[I\middle|\,T\right] + P_{\mathrm{L}}^{\star}XP_{\mathrm{R}}. \tag{3.11}$$

We now convert (3.11) into (3.1) by defining the matrices $A_{\mathrm{S}} \in \mathbb{C}^{k\times k}$ and $S \in \mathbb{C}^{(n-k)\times k}$ via the formulæ

$$A_{\mathrm{S}} = \tilde{R}_{11}^{\star}\tilde{Q}^{\star}, \qquad \text{and} \qquad S = \tilde{R}_{12}^{\star}(\tilde{R}_{11}^{\star})^{-1}, \tag{3.12}$$

respectively. □

Remark 3. While the definition (3.5) serves its purpose within the proof of Theorem 3, it is somewhat misleading. Indeed, it is more reasonable to define T as a solution of the equation

$$||R_{11}T - R_{12}||_2 \leq \sigma_{k+1}(A)\sqrt{1 + k(n-k)}. \tag{3.13}$$

When the solution is non-unique we chose a solution that minimizes $||T||_{\mathrm{F}}$. From the numerical point of view, the definition (3.13) is much preferable to (3.5) since it is almost invariably the case that R_{11} is highly ill-conditioned, if not outright singular.

Introducing the notation

$$A_{\mathrm{CS}} = P_{\mathrm{L}}\left[\begin{array}{c} I \\ \hline S \end{array}\right] A_{\mathrm{S}} \in \mathbb{C}^{n\times k}, \qquad \text{and} \qquad A_{\mathrm{RS}} = A_{\mathrm{S}}\left[I\middle|\,T\right] P_{\mathrm{R}} \in \mathbb{C}^{k\times m}, \tag{3.14}$$

we observe that under the conditions of Theorem 3, the factorization (3.1) can be rewritten in the forms

$$A = A_{\mathrm{CS}}\left[I\middle|\,T\right] P_{\mathrm{R}}^{\star} + X, \tag{3.15}$$

and

$$A = P_{\mathrm{L}} \left[\frac{I}{S} \right] A_{\mathrm{RS}} + X. \tag{3.16}$$

The matrix A_{CS} consists of k of the columns of A, while A_{RS} consists of k of the rows. We refer to A_{S} as the skeleton of A, and to A_{CS} and A_{RS} as the column and row skeletons, respectively.

Remark 4. While Theorem 3 guarantees the existence of a well-conditioned factorization of the form (3.1), it says nothing about the cost of obtaining such a factorization. However, it follows immediately from Theorem 2 that a factorization (3.1) with the matrices S, T, and X satisfying the weaker bounds

$$\|S\|_2 \leq \sqrt{mk(m-k)}, \qquad \text{and} \qquad \|T\|_2 \leq \sqrt{nk(n-k)}, \tag{3.17}$$

and, with $l = \min(m, n)$,

$$\|X\|_2 \leq \sqrt{1 + lk(l-k)}\,\sigma_{k+1}(A), \tag{3.18}$$

can be constructed at the cost $O(mnl)$.

Observation 1. The relations (3.1), (3.15), (3.16) have simple geometric interpretations. Specifically, (3.15) asserts that for a matrix A of rank k, it is possible to select k columns that form a well-conditioned basis of the entire column space. Let $j_1, \ldots, j_k \in \{1, \ldots, n\}$ denote the indices of those columns and let $X_k = \mathrm{span}(e_{j_1}, \ldots, e_{j_k}) \subseteq \mathbb{C}^n$ (thus, X_k is the space of vectors whose only non-zero coordinates are $x_{j_1}, \ldots, x_{j_k}$). According to Theorem 3, there exists an operator

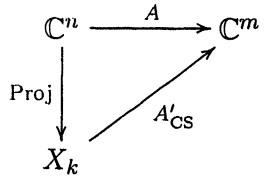
$$\mathrm{Proj} : \mathbb{C}^n \to X_k, \tag{3.19}$$

defined by the formula

$$\mathrm{Proj} = P_{\mathrm{R}} \left[\begin{array}{c|c} I & T \\ \hline 0 & \end{array} \right] P_{\mathrm{R}}^{\star}, \tag{3.20}$$

such that the diagram

(3.21)



is commutative. Here, A'_{CS} is the $m \times n$ matrix formed by setting all columns of A except $j_1, \ldots, j_k$ to zero. Furthermore, $\sigma_1(\mathrm{Proj})/\sigma_k(\mathrm{Proj}) \leq \sqrt{1 + k(n-k)}$. Similarly, equation (3.16) asserts the existence of k rows, say with indices $i_1, \ldots, i_k \in \{1, \ldots, m\}$, that form a well-conditioned basis for the entire row-space. Setting $Y_k = \mathrm{span}(e_{i_1}, \ldots, e_{i_k}) \subseteq \mathbb{C}^m$, there exists an operator

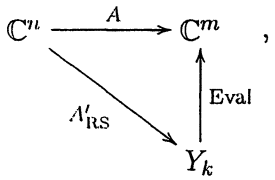
$$\mathrm{Eval} : Y_k \to \mathbb{C}^m, \tag{3.22}$$

defined by

$$\mathrm{Eval} = P_{\mathrm{L}} \left[\begin{array}{c|c} \frac{I}{S} & 0 \end{array} \right] P_{\mathrm{L}}^{\star}. \tag{3.23}$$

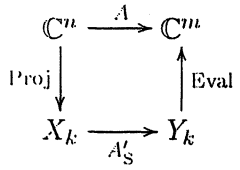
such that the diagram

(3.24)



is commutative. Here, A'_{RS} is the $m \times n$ matrix formed by setting all rows of A except $i_1, \ldots, i_k$ to zero. Furthermore, $\sigma_1(\mathrm{Eval})/\sigma_k(\mathrm{Eval}) \leq \sqrt{1 + k(m-k)}$. Finally, the geometric interpretation of (3.1) is the combination of the diagrams (3.21) and (3.24),

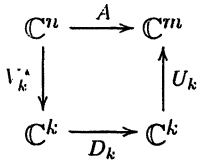
(3.25)



Here, A'_{S} is the $m \times n$ matrix formed by setting all entries of A, except those at the intersection of the rows $i_1, \ldots, i_k$ with the columns $j_1, \ldots, j_k$, to zero.

As a comparison, we consider the diagram

(3.26)



obtained when the SVD is used to compress the matrix $A \in \mathbb{C}^{m \times n}$. Here, D_k is the $k \times k$ diagonal matrix formed by the k largest singular values of A, and V_k and U_k are column matrices containing the corresponding right and left singular vectors, respectively. The factorization (3.25) has the advantage over (3.26) that the mappings Proj and Eval leave k of the coordinates invariant. This is gained at the price of non-orthonormality of these mappings.

4. Numerical apparatus

In this section, we present a simple and reasonably efficient procedure for computing the factorization (3.1). It has been extensively tested and consistently produces factorizations that satisfy the bounds (3.17). While there exist matrices for which this simple approach will not work well, they appear to be exceedingly rare.

Given an $m \times n$ matrix A, the first step (out of four) is to apply the pivoted Gram-Schmidt process to its columns. The process is halted when the column space has been exhausted to a preset accuracy ε, leaving a factorization

$$AP_{\mathrm{R}} = Q\left[R_{11} \middle| R_{12}\right], \tag{4.1}$$

where $P_{\mathrm{R}} \in \mathbb{C}^{n \times n}$ is a permutation matrix, $Q \in \mathbb{C}^{m \times k}$ has orthonormal columns, $R_{11} \in \mathbb{C}^{k \times k}$ is upper triangular, and $R_{12} \in \mathbb{C}^{k \times (n-k)}$.

The second step is to find a matrix $T \in \mathbb{C}^{k \times (n-k)}$ that solves the equation

$$R_{11}T = R_{12} \tag{4.2}$$

to within accuracy ε. When R_{11} is ill-conditioned, there is a large set of solutions; we pick one for which $||T||_{\mathrm{F}}$ is small.

Letting $A_{\mathrm{CS}} \in \mathbb{C}^{m\times k}$ denote the matrix formed by the first k columns of AP_{R}, we now have a factorization

$$A = A_{\mathrm{CS}} \left[I \middle| T\right] P_{\mathrm{R}}^{\star}. \tag{4.3}$$

The third and the fourth steps are entirely analogous to the first and the second, but are concerned with finding k rows of A_{CS} that form a basis for its row-space. They result in a factorization

$$A_{\mathrm{CS}} = P_{\mathrm{L}} \left[\frac{I}{S}\right] A_{\mathrm{S}}. \tag{4.4}$$

The desired factorization is now obtained by inserting (4.4) into (4.3):

$$A = P_{\mathrm{L}} \left[\frac{I}{S}\right] A_{\mathrm{S}} \left[I \middle| T\right] P_{\mathrm{R}}^{\star}. \tag{4.5}$$

For this technique to be successful, it is crucially important that the Gram-Schmidt factorization be performed accurately. Modified Gram-Schmidt or the method using Householder reflectors are not accurate enough. Instead, we use a technique that is based on modified Gram-Schmidt, but that at each step re-orthogonalizes the vector chosen to add to the basis before adding it. In exact arithmetic, this step would be superfluous, but in the presence of round-off error it greatly increases the quality of the factorization generated, see *e.g.* [6].

5. Application: An accelerated direct solver for integral equations

In this section we use the matrix compression technique presented in Section 3 to construct an accelerated direct solver for boundary integral equations with non-oscillatory kernels. Upon discretization, such equations lead to dense systems of linear equations, and iterative methods combined with fast matrix-vector multiplication techniques are commonly used to obtain the solution. Many such fast multiplication techniques take advantage of the fact that the off-diagonal blocks of the discrete system typically have low rank. Employing the matrix compression techniques presented in Section 3, we use this low-rank property to accelerate direct, rather than iterative, solution techniques. The method uses no machinery beyond what is described in Section 3 and is applicable to most integral equations involving non-oscillatory kernels.

For concreteness, we consider the equation

$$u(x) + \int_{\Gamma} K(x,y)u(y)\,dy = f(x), \qquad \text{for } x \in \Gamma, \tag{5.1}$$

where Γ is some contour and $K(x,y)$ is a non-oscillatory kernel. The function u represents an unknown "charge" distribution on Γ that is to be determined from the given function f. The method that we present works for almost any contour but for simplicity, we will assume that the contour consists of p disjoint pieces, $\Gamma = \Gamma_1 + \cdots + \Gamma_p$, where all pieces have similar size (an example is given in Fig. 3). In fact, to simplify the formulas, we will for the most part set $p = 3$.

Discretizing each contour Γ_i using n points, the equation (5.1) takes the form

$$\left[\begin{array}{c|c|c} M^{(1,1)} & M^{(1,2)} & M^{(1,3)} \\ \hline M^{(2,1)} & M^{(2,2)} & M^{(2,3)} \\ \hline M^{(3,1)} & M^{(3,2)} & M^{(3,3)} \end{array}\right] \left[\begin{array}{c} u^{(1)} \\ \hline u^{(2)} \\ \hline u^{(3)} \end{array}\right] = \left[\begin{array}{c} f^{(1)} \\ \hline f^{(2)} \\ \hline f^{(3)} \end{array}\right], \tag{5.2}$$

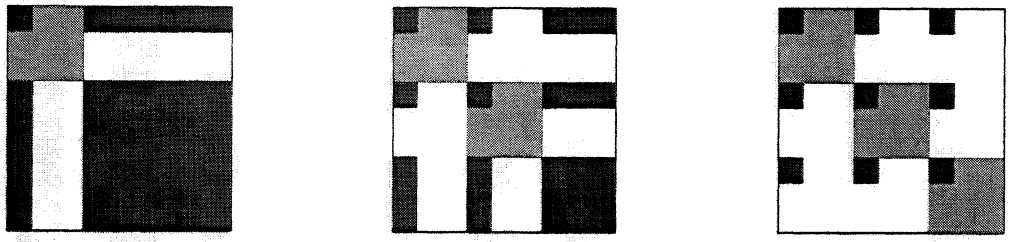


FIGURE 1. Zeros are introduced into the matrix in three steps: (a) interaction between Γ_1 and the other contours is compressed, (b) interaction with Γ_2 is compressed, (c) interaction with Γ_3 is compressed. The small black blocks are of size $k \times k$ and consist of entries that have not been changed beyond permutations, grey blocks refer to updated parts and white blocks are all zero entries.

where $u^{(i)} \in \mathbb{C}^n$ and $f^{(i)} \in \mathbb{C}^n$ are discrete representations of the unknown boundary charge distribution and the right hand side associated with Γ_i, and $M^{(i,j)} \in \mathbb{C}^{n\times n}$ is a dense matrix representing the evaluation of a potential on Γ_i caused by a charge distribution on Γ_j.

The interaction between Γ_1 and the rest of the contour is governed by the matrices

$$H^{(1)} = \left[M^{(1,2)} \middle| M^{(1,3)}\right] \in \mathbb{C}^{n\times 2n}, \qquad \text{and} \qquad V^{(1)} = \left[\frac{M^{(2,1)}}{M^{(3,1)}}\right] \in \mathbb{C}^{2n\times n}. \tag{5.3}$$

For non-oscillatory kernels, these matrices are typically highly rank-deficient. We let k denote an upper bound on their ranks (to within some preset level of accuracy ε). By virtue of (3.16), we know that there exist k rows of $H^{(1)}$ which form a well-conditioned basis for all the n rows. In other words, there exists a well-conditioned $n \times n$ matrix $L^{(1)}$ (see Remark 6) such that

$$L^{(1)}H^{(1)} = \left[\frac{H^{(1)}_{\text{RS}}}{Z}\right] + O(\varepsilon), \tag{5.4}$$

where $H^{(1)}_{\text{RS}}$ is a $k \times 2n$ matrix formed by k of the rows of $H^{(1)}$ and Z is the $(n-k) \times 2n$ zero matrix. There similarly exist an $n \times n$ matrix $R^{(1)}$ such that

$$V^{(1)}R^{(1)} = \left[V^{(1)}_{\text{CS}} \middle| Z^*\right] + O(\varepsilon), \tag{5.5}$$

where $V^{(1)}_{\text{CS}}$ is a $2n \times k$ matrix formed by k of the columns of $V^{(1)}$. For simplicity, we will henceforth assume that the off-diagonal blocks have *exact* rank at most k and ignore the error terms.

The relations (5.4) and (5.5) imply that by restructuring equation (5.2) as follows,

$$\left[\begin{array}{c|c|c} L^{(1)}M^{(1,1)}R^{(1)} & L^{(1)}M^{(1,2)} & L^{(1)}M^{(1,3)} \\ \hline M^{(2,1)}R^{(1)} & M^{(2,2)} & M^{(2,3)} \\ \hline M^{(3,1)}R^{(1)} & M^{(3,2)} & M^{(3,3)} \end{array}\right] \left[\begin{array}{c} (R^{(1)})^{-1}u^{(1)} \\ \hline u^{(2)} \\ \hline u^{(3)} \end{array}\right] = \left[\begin{array}{c} L^{(1)}f^{(1)} \\ \hline f^{(2)} \\ \hline f^{(3)} \end{array}\right], \tag{5.6}$$

we introduce large blocks of zeros in the matrix, as shown in Figure 1(a).

Next, we compress the interaction between Γ_2 and the rest of the contour to obtain the matrix structure shown in Fig. 1(b). Repeating the process with Γ_3, we obtain the final structure shown

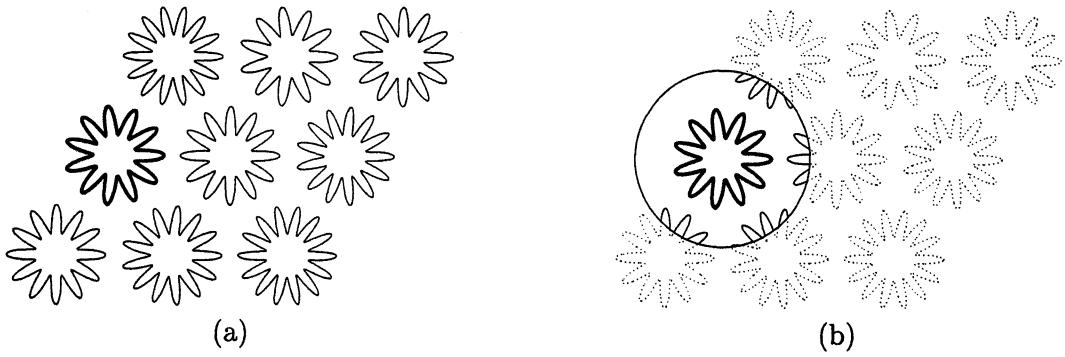


FIGURE 2. In order to determine the $R^{(i)}$ and $L^{(i)}$ that compress the interaction between Γ_i (shown in bold) and the remaining contours, it is sufficient to consider only the interactions between the contours drawn with a solid line in (b).

in Fig. 1(c). At this point, we have constructed matrices $R^{(i)}$ and $L^{(i)}$ and formed the new system

$$\left[\begin{array}{c|c|c} L^{(1)}M^{(1,1)}R^{(1)} & L^{(1)}M^{(1,2)}R^{(2)} & L^{(1)}M^{(1,3)}R^{(3)} \\ \hline L^{(2)}M^{(2,1)}R^{(1)} & L^{(2)}M^{(2,2)}R^{(2)} & L^{(2)}M^{(2,3)}R^{(3)} \\ \hline L^{(3)}M^{(3,1)}R^{(1)} & L^{(3)}M^{(3,2)}R^{(2)} & L^{(3)}M^{(3,3)}R^{(3)} \end{array}\right] \left[\begin{array}{c} (R^{(1)})^{-1}u^{(1)} \\ \hline (R^{(2)})^{-1}u^{(2)} \\ \hline (R^{(3)})^{-1}u^{(3)} \end{array}\right] = \left[\begin{array}{c} L^{(1)}f^{(1)} \\ \hline L^{(2)}f^{(2)} \\ \hline L^{(3)}f^{(3)} \end{array}\right], \tag{5.7}$$

whose matrix is shown in Figure 1(c). We emphasize that the $k \times k$ non-zero parts of the off-diagonal blocks are *submatrices* of the original $n \times n$ off-diagonal blocks. The parts of the matrix that are shown as grey in the figure represent interactions that are internal to each contour. These $n-k$ degrees of freedom per contour can be eliminated by performing a local, $O(n^3)$, operation for each contour. This leaves a dense system of 3×3 blocks, each of size $k \times k$. Thus, we have reduced the problem size by a factor of n/k.

Remark 5. For the algorithm presented above, the compression of the interaction between a fixed contour and its $p-1$ fellows is quite costly since it requires the construction and compression of the large matrices $H^{(i)} \in \mathbb{C}^{n\times(p-1)n}$ and $V^{(i)} \in \mathbb{C}^{(p-1)n\times n}$. In the numerical examples presented below, this step is avoided by constructing matrices $L^{(i)}$ and $R^{(i)}$ that satisfy (5.4) and (5.5) through an entirely local procedure. We illustrate how this is done by considering the contours in Fig. 2(a) and supposing that we want to find the transforms that compress the interaction of the contour Γ_i (drawn with a bold line) with the remaining ones. This can be done by compressing the interaction between Γ_i and an artificial contour Γ_{artif} that surrounds Γ_i (as shown in Fig. 2(b)) combined with the parts of the other contours that penetrate it. This procedure works for any potential problem for which the Green's identities hold. The computational cost for one compression is $O(kn^2)$ rather than the $O(pkn^2)$ cost for constructing and compressing the entire $H^{(i)}$ and $V^{(i)}$.

To sum up: The accelerated solver consists of four steps. For a problem involving p contours, each of which is discretized using n nodes and having off-diagonal blocks of rank at most k, they are:

(1) The off-diagonal blocks are skeletonized and the diagonal $n \times n$ blocks are updated at a cost of $O(pkn^2)$ using the technique described in Remark 5.

(2) The $n - k$ degrees of freedom that represent internal interactions for each contour are eliminated at a cost of $O(pn^3)$.
(3) The reduced $kp \times kp$ system is solved at a cost of $O(k^3p^3)$.
(4) The solution of the original system is reconstructed from the solution of the reduced problem through p local operations at a cost of $O(pn^2)$.

The third step is typically the most expensive one with an asymptotic cost of $t^{(\text{comp})} \sim ck^3p^3$. The cost of a solution of the uncompressed equations is $t^{(\text{uncomp})} \sim cn^3p^3$. Consequently;

$$\text{Speed-up} = \frac{t^{(\text{uncomp})}}{t^{(\text{comp})}} \sim \left(\frac{n}{k}\right)^3.$$

Remark 6. The existence of the matrices $L^{(1)}$ and $R^{(1)}$ are direct consequences of (3.16) and (3.15), respectively. Specifically, substituting $H^{(1)}$ for A in (3.16), we obtain

$$P_{\text{L}}^{\star} H^{(1)} = \left[\frac{I}{S}\right] H_{\text{RS}}^{(1)}, \tag{5.8}$$

where $H_{\text{RS}}^{(1)}$ is the $k \times 2n$ matrix consisting of the top k rows of $P_{\text{L}}^{\star} H^{(1)}$. The relation (5.4) now follows from (5.8) by defining

$$L^{(1)} = \left[\begin{array}{c|c} I & 0 \\ \hline -S & I \end{array}\right] P_{\text{L}}^{\star}. \tag{5.9}$$

We note that the largest and smallest singular values of $L^{(1)}$ satisfy

$$\begin{aligned} \sigma_1(L^{(1)}) &\leq \left(1 + ||S||_{l^2}^2\right)^{1/2}, \\ \sigma_n(L^{(1)}) &\geq \left(1 + ||S||_{l^2}^2\right)^{-1/2}. \end{aligned} \tag{5.10}$$

Thus $\text{cond}(L^{(1)}) \leq 1 + ||S||_{l^2}^2$, which is of moderate size according to Theorem 3. The matrix $R^{(1)}$ is similarly constructed by forming the column skeleton of $V^{(1)}$.

Remark 7. Equations (5.4) and (5.5) have simple heuristic interpretations: Equation (5.4) says that it is possible to choose k points on the contour Γ_1 in such a way that when a field generated by charge distributions on the rest of the contour is known at those points, it is possible to extrapolate the field at the remaining points on Γ_1 from those values. Equation (5.5) says that it is possible to choose k points on Γ_1 in such a way that any field on the rest of the contour generated by charges on Γ_1, can be replicated by placing charges only on those k points.

Remark 8. It is sometimes advantageous to choose the same k points when constructing the skeletons of $H^{(i)}$ and $V^{(i)}$. This can be achieved by compressing the two matrices jointly, for instance by forming the row skeleton of $\left[H^{(i)} \middle| (V^{(i)})^{\star}\right]$. In this case $L^{(i)} = (R^{(i)})^{\star}$. When this is done, the compression ratio deteriorates since the singular values of $\left[H^{(i)} \middle| (V^{(i)})^{\star}\right]$ decay slower than those of either $H^{(i)}$ or $V^{(i)}$, as is seen by comparing Figures 4 and 5.

Remark 9. When the solution of equation (5.2) is sought for multiple right-hand sides, the cost of the first solve is $O(mnk)$. Subsequent solves can be preformed using $O(p^2k^2 + pn^2)$ operations rather than $O(p^2n^2)$ for an uncompressed solver.

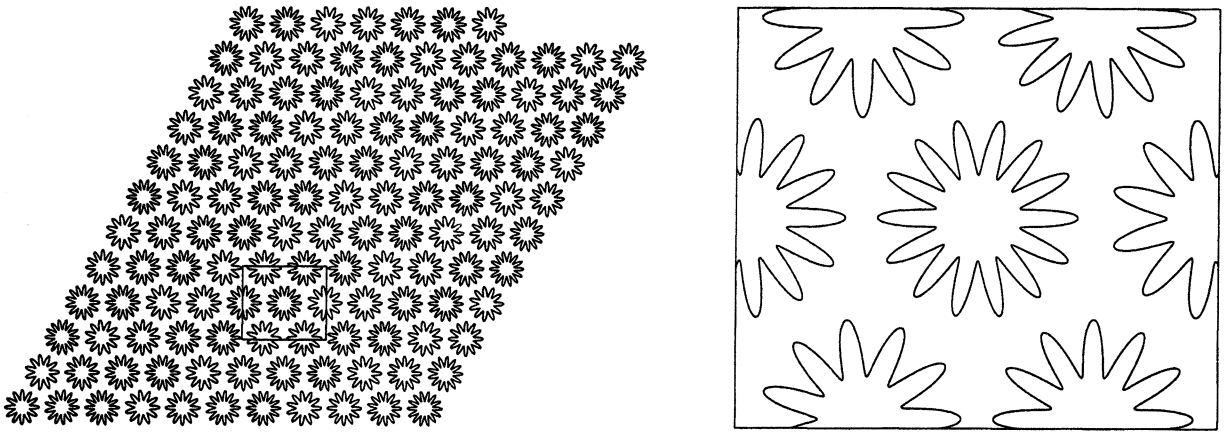


FIGURE 3. The contours used for the numerical calculations with $p = 128$. Picture (a) shows the full contour and a box (which is not part of the contour) that indicates the location of the close-up shown in (b).

Remark 10. The direct solver that we have presented has a computational complexity that scales cubically with the problem size N and is thus not a "fast" algorithm. However, by applying the techniques presented recursively, it is possible to reduce the asymptotic complexity to $O(N^{3/2})$, and possibly even $O(N \log N)$. This is a topic of current research.

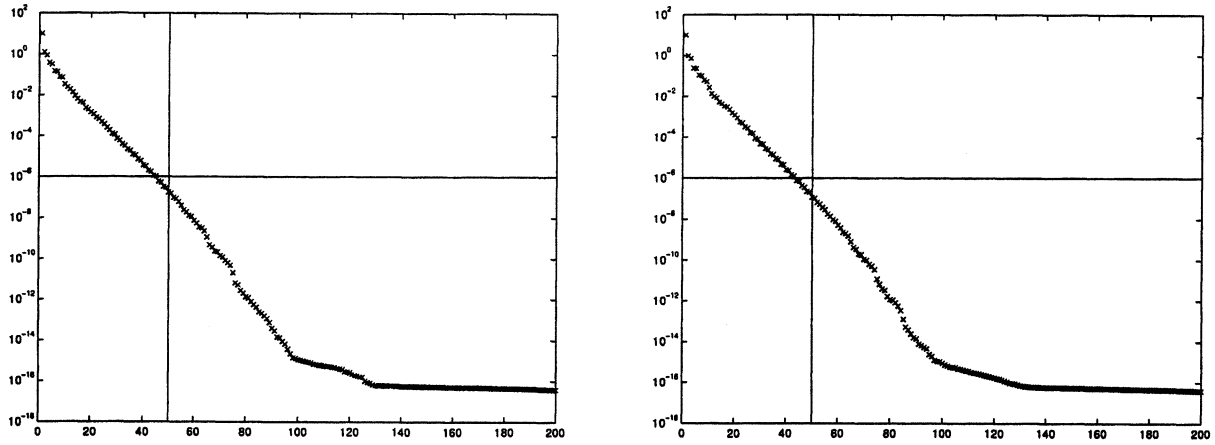
6. NUMERICAL RESULTS

The algorithm described in Section 5 has been computationally tested on the second kind integral equation obtained by discretizing an exterior Dirichlet boundary value problem using the double layer kernel. The contours used consisted of a number of jagged circles arranged in a skewed square as shown in Fig. 3. The number of contours p ranged from 8 to 128. For this problem, $n = 200$ points per contour were required to obtain a relative accuracy of $\varepsilon = 10^{-6}$. We found that to this level of accuracy, no $H^{(i)}$ or $V^{(i)}$ had rank exceeding $k = 50$. As an example, we show in Fig. 4 the singular values of the matrices $H^{(i)}$ and $V^{(i)}$ representing interactions between the highlighted contour in Fig. 2(a) and the remaining ones.

The algorithm described in Section 5 was implemented in FORTRAN and run on a 2.8GHz Pentium IV desktop PC with 512Mb RAM. The CPU times for a range of different problem sizes are presented in Table 1. The data presented supports the following claims for the compressed solver:

- For large problems, the CPU time speed-up approaches the estimated factor of $(n/k)^3 = 64$.
- The reduced memory requirement make large problems amenable to direct solution.

Remark 11. In the interest of simplicity, we forced the program to use the same compression ratio k/n for each contour. In general, it detects the required interaction rank of each contour as its interaction matrices are being compressed and uses different ranks for each contour.



(a) (b)

FIGURE 4. Plots of the singular values of (a) $V^{(i)}$ and (b) $H^{(i)}$ for a discretization of the double layer kernel associated with the Laplace operator on the nine contours depicted in Fig. 2(a). In the example shown, the contours were discretized using $n = 200$ points, giving a relative discretization error of about 10^{-6}. The plots show that to that level of accuracy, the matrices $V^{(i)} \in \mathbb{C}^{1600\times 200}$ and $H^{(i)} \in \mathbb{C}^{200\times 1600}$ have numerical rank less than $k = 50$ (to accuracy 10^{-6}).

p	$t^{(\text{uncomp})}$	$t^{(\text{comp})}$	$t^{(\text{comp})}_{\text{init}}$	$t^{(\text{comp})}_{\text{solve}}$	Error
8	5.6	2.0 (4.6)	1.6 (4.1)	0.05	$8.1 \cdot 10^{-7} (1.4 \cdot 10^{-7})$
16	50	4.1 (16.4)	3.1 (15.5)	0.4	$2.9 \cdot 10^{-6} (2.8 \cdot 10^{-7})$
32	451	13.0 (72.1)	6.4 (65.3)	5.5	$4.4 \cdot 10^{-6} (4.4 \cdot 10^{-7})$
64	*3700*	65 (270)	14 (220)	48	—
128	*30000*	480 (1400)	31 (960)	440	—

TABLE 1. CPU times in seconds for solving (5.2). p is the number of contours. $t^{(\text{uncomp})}$ is the CPU time required to solve the uncompressed equations; the numbers in italics are estimated since these problems did not fit in RAM. $t^{(\text{comp})}$ is the CPU time to solve the equations using the compression method; this time is split between $t^{(\text{comp})}_{\text{init}}$, the time to compress the equations, and $t^{(\text{comp})}_{\text{solve}}$, the time to solve the reduced system of equations. The error is the relative error incurred by the compression measured in the maximum norm when the right hand side is a vector of ones. Throughout the table, the numbers in parenthesis refer to numbers obtained when the technique of Remark 5 is not used.

7. CONCLUSIONS

We have described a "compression" scheme for low-rank matrices. For a matrix A of dimensionality $m \times n$ and rank k, the factorization can be applied to an arbitrary vector for the cost of $(n+m-k)\cdot k$ operations, after a significant initial factorization cost; this is marginally faster than

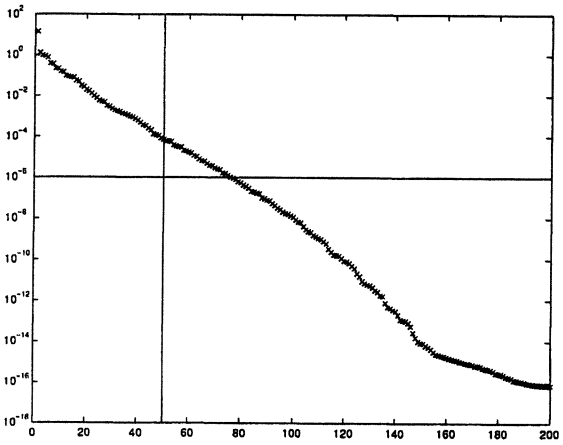


FIGURE 5. Plot of the singular values of $X^{(i)} = \left[H^{(i)} \middle| (V^{(i)})^*\right]$ where $H^{(i)}$ and $V^{(i)}$ are as in Figure 4. The numerical rank of $X^{(i)}$ is approximately 80, which is larger than the individual ranks of $H^{(i)}$ and $V^{(i)}$.

the cost $(n+m) \cdot k$ produced by the SVD. The factorization cost is roughly the same as that for the rank-revealing QR decomposition of A.

A more important advantage of the proposed decomposition is the fact that it expresses all of the columns of A as linear combinations of k appropriately selected columns of A, and all of the rows of A as linear combinations of k appropriately selected rows of A. Since each of the basis vectors (both row and column) produced by the SVD (or any other classical factorizations) is a linear combination of *all* rows (columns) of A, the decomposition we propose is considerably easier to manipulate; we illustrate this point by constructing an accelerated scheme for the direct solution of integral equations of potential theory in the plane.

A related advantage of the proposed decomposition is the fact that one frequently encounters collections of matrices such that the same selection of rows and columns can be used for each matrix to span its row and column space (in other words, there exist fixed P_{L} and P_{R} such that each matrix in the collection has a decomposition (3.1) with small matrices S and T). Once one matrix in such a collection has been factorized, the decomposition of the remaining ones is considerably simplified since the skeleton of the first can be reused. If it should happen that the skeleton of the first matrix that was decomposed is not a good choice for some other matrix, this is easily detected (since then no small matrices S and T can be computed) and the global skeleton can be extended as necessary.

We have constructed several other numerical procedures using the approach described in this paper. In particular, a code has been designed for the (reasonably) rapid solution of scattering problems in the plane based on the direct (as opposed to iterative) solution of the Lippman-Schwinger equation; the scheme utilizes the same idea as that used in [4], and has the same asymptotic CPU time estimate $O(N^{3/2})$ for a square region discretized into N nodes. However, the CPU times obtained by us are a significant improvement on these reported in [4]; the paper reporting this work is in preparation.

It also appears to be possible to utilize the techniques of this paper to construct an order $O(N \log N)$ (or possibly even order order $O(N)$ (!)) scheme for the solution of elliptic PDEs in

both two and three dimensions, provided that the associated Green's function is not oscillatory. This work is in progress, and if successful will be reported at a later date.

References

[1] Ming Gu and Stanley C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput. **17** (1996), no. 4, 848–869.

[2] B. Alpert, G. Beylkin, R. Coifman, V. Rokhlin, *Wavelet-like bases for the fast solution of second-kind integral equations*, SIAM J. Sci. Comput., vol. 14, pp. 159-184, 1993.

[3] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms I*, Communications on Pure and Applied Mathematics, 14:141-183 (1991).

[4] Yu Chen, *Fast direct solver for the Lippmann-Schwinger equation*, Advances in Computational Mathematics, vol. 16, pp. 175-190, 2002.

[5] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.

[6] Å Björck, *Numerics of Gram-Schmidt orthogonalization*, Linear Algebra Appl., vol. 197/198, pp. 297–316, 1994.

[7] G.Beylkin, *On multiresolution methods in numerical analysis*, Documenta Mathematica, Extra Volume ICM 1998, III, pp. 481-490, 1998.

[8] G.W. Stewart, *Matrix Algorithms, Vol. I*, SIAM, Philadelphia 1998.