On Some New Applications of the CS Decomposition

Vjeran Hari*[1]

[1] Department of Mathematics, University of Zagreb, P.O. Box 335, 10002 Zagreb, Croatia.

Received 15 November 2002, revised 30 November 2002, accepted 2 December 2002

In this report we consider some new applications of the cosine-sine decomposition of orthogonal matrices. These applications require new algorithms for this decomposition.

1 Introduction

Let Q be orthogonal matrix of order n, partitioned as follows

$$Q = \begin{array}{c} \left[\begin{array}{cc} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{array}\right] \begin{array}{l} \}\, l \\ \}\, n-l, \end{array} \\ \begin{array}{cc} \underbrace{\quad}_{l} & \underbrace{\quad}_{n-l} \end{array} \end{array} \quad 1 \le l \le n-1.$$

Then Q can be factored into three orthogonal matrices,

$$Q = U\Theta V^T = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix} \Theta \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix}^T,$$

where U_{11} and V_{11} are of order l, while U_{22} and V_{22} are of order $n-l$. Here

$$\Theta = \begin{array}{c} \left[\begin{array}{ccc} I & & \\ & C & -S \\ & S & C \end{array}\right] \begin{array}{l} \}\, 2l-n \\ \}\, n-l, \\ \}\, n-l \end{array} \\ \begin{array}{ccc} \underbrace{\quad}_{2l-n} & \underbrace{\quad}_{n-l} & \underbrace{\quad}_{n-l} \end{array} \end{array} \qquad \text{provided that } 2l \ge n$$

and

$$\Theta = \begin{array}{c} \left[\begin{array}{ccc} C & & -S \\ & I & \\ S & & C \end{array}\right] \begin{array}{l} \}\, l \\ \}\, n-2l, \\ \}\, l \end{array} \\ \begin{array}{ccc} \underbrace{\quad}_{l} & \underbrace{\quad}_{n-2l} & \underbrace{\quad}_{l} \end{array} \end{array} \qquad \text{provided that } 2l \le n$$

where C is diagonal with non-negative diagonal elements arranged in the nonincreasing ordering, S is diagonal with non-negative diagonal elements arranged in nondecreasing ordering, and $C^2 + S^2 = I$ holds. This decomposition is known as cosine-sine decomposition of orthogonal matrices or shorter, the *CS decomposition*.

So far, this decomposition has been used in the context of principal angles between two subspaces. Typically, the two subspaces $\mathcal{X}$ and $\mathcal{Y}$ are given by their bases, the orthonormal matrices X and Y. If

* Corresponding author: e-mail: hari@math.hr, Phone: +385 1 4605 748, Fax: +385 1 4680 335

$\mathcal{X}$ and $\mathcal{Y}$ are from $\mathbb{R}^n$, then their bases can be appended by additional columns to obtain the bases of the whole $\mathbb{R}^n$. This gives orthogonal matrices $[X, X^\perp]$ and $[Y, Y^\perp]$, whose cross-product matrix

$$Q = [X, X^\perp]^T [Y, Y^\perp] = \begin{bmatrix} X^T Y & X^T Y^\perp \\ [X^\perp]^T Y & [X^\perp]^T Y^\perp \end{bmatrix} \tag{1}$$

is also orthogonal. The singular values of $[X^\perp]^T Y$ are the sines and those of $X^T Y$ are the cosines of the principal angles between $\mathcal{X}$ and $\mathcal{Y}$. Note that the CS decomposition of this Q reveals these sines and cosines. The known algorithms of Stewart [2] and Van Loan [3] actually compute the decomposition

$$\begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix} \begin{bmatrix} C \\ S \end{bmatrix} V_{11}$$

which is sufficient for the purpose of computing the sines and cosines of the principal angles between two subspaces. The same decomposition is also used for solving the generalized singular value problem.

Here we propose another use of the CS decomposition in the form (1).

Let $G \in \mathbb{R}^{m \times n}$,

$$G = \begin{array}{c} \begin{bmatrix} G_{11} & \cdots & G_{1p} \\ \vdots & \ddots & \vdots \\ G_{q1} & \cdots & G_{qp} \end{bmatrix} \begin{array}{l} \} \, m_1 \\ \} \, \vdots \\ \} \, m_q \end{array} \\ \begin{array}{ccc} \underbrace{\quad}_{n_1} & \cdots & \underbrace{\quad}_{n_p} \end{array} \end{array}$$

where p (q) denote the number of block- (row-) columns. Consider the general iterative process

$$G^{(k+1)} = W^{(k)} G^{(k)} Q^{(k)}, \quad k \geq 1, \tag{2}$$

where $G^{(1)} = G \in \mathbb{R}^{m \times n}$ is the starting matrix. For each k, the orthogonal matrices $Q^{(k)}$ and $W^{(k)}$ are computed from the elements of $G^{(k)}$. Each $Q^{(k)}$ (the same for $W^{(k)}$) is of the form

$$Q^{(k)} = \begin{bmatrix} I & O & O & O & O \\ O & Q_{ii}^{(k)} & O & Q_{ij}^{(k)} & O \\ O & O & I & O & O \\ O & Q_{ji}^{(k)} & O & Q_{jj}^{(k)} & O \\ O & O & O & O & I \end{bmatrix},$$

where the pivot indices $i = i(k)$ and $j = j(k)$ are chosen by some pivot strategy, $Q_{ii}^{(k)} \in \mathbb{R}^{n_i \times n_i}$, $Q_{jj}^{(k)} \in \mathbb{R}^{n_j \times n_j}$, $Q_{ij}^{(k)} \in \mathbb{R}^{n_i \times n_j}$ and $Q_{ji}^{(k)} \in \mathbb{R}^{n_j \times n_i}$ (similar, $W_{rt}^{(k)} \in \mathbb{R}^{m_r \times m_t}$ for $r, t \in \{i, j\}$).

Matrices $W^{(k)}$ ($Q^{(k)}$) can be used, say, to annihilate the (j,i)- ((i,j)-) block of $G^{(k)}$ and then the process (2) can be viewed as block (i.e., BLAS 3) algorithm for the known factorizations like QR, QL (RQ, LQ), or block-bidiagonalization. In the latter case the both matrices $W^{(k)}$ and $Q^{(k)}$ are generally non-trivial. These are finite processes.

As an example of infinite process, one can assume that all $W^{(k)}$ are identities, while each $Q^{(k)}$ makes the i'th and j'th block columns $G^{(k)}$ orthogonal to each other. This leads to block one-sided Jacobi-type algorithm for computing the singular value decomposition of G. Evenmore, if the initial G is first brought to a block-triangular form (as mentioned above), then one can derive the block form of the known Kogbetliantz algorithm for triangular matrices.

Block algorithms are important because they can better exploit cache memory and they can better fetch the matrix entries that are needed in each (block-) step of the algorithm.

Here we briefly report how CS decomposition can be used to accelerate the algorithms of the form (2). To make the exposition shorter, we shall consider only the case $W^{(k)} = I_m$ for all k. Then step k can be written in the form

$$\left[G_i^{(k+1)}, G_j^{(k+1)}\right] = \left[G_i^{(k)}, G_j^{(k)}\right] = \hat{Q}^{(k)}, \quad \hat{Q}^{(k)} = \begin{bmatrix} Q_{ii}^{(k)} & Q_{ij}^{(k)} \\ Q_{ji}^{(k)} & Q_{jj}^{(k)} \end{bmatrix}, \tag{3}$$

where $G_i^{(k)}$ and $G_j^{(k)}$ are the i'th and j'th block-column of $G^{(k)}$, $k \geq 1$.

Let us assume that $G^{(k)} = B^{(k)}\Phi^{(k)}$, where $\Phi^{(k)} \in \mathbb{R}^{n\times n}$ is block-diagonal and orthogonal. Let $[B_1^{(k)}, \ldots, B_p^{(k)}]$ be the block-column partition of $B^{(k)}$ and $\mathrm{diag}(\Phi_{11}^{(k)}, \ldots, \Phi_{pp}^{(k)})$ the corresponding block-partition of $\Phi^{(k)}$. For $k = 1$, we assume $\Phi^{(1)} = I_n$.

At step k, $k \geq 1$, we compute $\hat{Q}^{(k)}$ from the elements of $B^{(k)}$ and $\Phi^{(k)}$. Then we compute the CS decomposition of $\hat{Q}^{(k)}$,

$$\hat{Q}^{(k)} = U^{(k)}\Theta^{(k)}[V^{(k)}]^T = \begin{bmatrix} U_{11}^{(k)} & O \\ O & U_{22}^{(k)} \end{bmatrix} \Theta^{(k)} \begin{bmatrix} V_{11}^{(k)} & O \\ O & V_{22}^{(k)} \end{bmatrix}^T.$$

Now, the relation (3) can be written as

$$\begin{aligned} \left[B_i^{(k+1)}, \; B_j^{(k+1)}\right] \begin{bmatrix} \Phi_{ii}^{(k+1)} & O \\ O & \Phi_{jj}^{(k+1)} \end{bmatrix} &= \left[B_i^{(k)}, B_j^{(k)}\right] \begin{bmatrix} \Phi_{ii}^{(k)} & O \\ O & \Phi_{jj}^{(k)} \end{bmatrix} \hat{Q}^{(k)} \\ &= \left[B_i^{(k)}, B_j^{(k)}\right] \left(\begin{bmatrix} \Phi_{ii}^{(k)} & O \\ O & \Phi_{jj}^{(k)} \end{bmatrix} \begin{bmatrix} U_{11}^{(k)} & O \\ O & U_{22}^{(k)} \end{bmatrix} \Theta^{(k)} \right) \begin{bmatrix} V_{11}^{(k)} & O \\ O & V_{22}^{(k)} \end{bmatrix}^T \end{aligned}$$

and we can set

$$\left[B_i^{(k+1)}, B_j^{(k+1)}\right] = \left(\left[B_i^{(k)}, B_j^{(k)}\right] \begin{bmatrix} \Phi_{ii}^{(k)} U_{11}^{(k)} & O \\ O & \Phi_{jj}^{(k)} U_{22}^{(k)} \end{bmatrix} \right) \Theta^{(k)}, \tag{4}$$

$$\Phi_{ii}^{(k+1)} = [V_{11}^{(k)}]^T, \quad \Phi_{jj}^{(k+1)} = [V_{22}^{(k)}]^T. \tag{5}$$

It is presumed that block-partition takes into account the cache memory capacity, i.e., that all small matrices $U_{11}^{(k)}$, $U_{22}^{(k)}$, $V_{11}^{(k)}$, $V_2^{(k)}$, $\Phi_{ii}^{(k)}$, $\Phi_{jj}^{(k)}$, $\Theta^{(k)}$, $\hat{Q}^{(k)}$ are contained in the cache memory which is several times faster than the main memory. So, all the bookkeeping of these matrices including the CS decomposition is performed in the cache. This extra cost in computational time is small compared to the gain in updating the block-columns, which is the slowest part of the algorithm. Indeed, once $X = \Phi_{ii}^{(k)} U_{11}^{(k)}$ and $Y = \Phi_{jj}^{(k)} U_{22}^{(k)}$ are computed, the first multiplication $[B_i^{(k)}, B_j^{(k)}]\,\mathrm{diag}(X, Y)$ halves the number of operations and CPU time involved in (3). The second multiplication with $\Theta^{(k)}$ involves at most $\min\{n_i, n_j\}$ rotations which is negligible in comparison with the cost of the first multiplication. Computational tests show (see [1]) that using the CS decomposition in this way, the total CPU time of one step can be reduced (theoretically up to 40%, but in practice usually) between 10% and 30%.

Note that the above modification of the original algorithm can be seen as the block version of the fast-scaled rotations. In contrast to the fast-scaled rotations, there is no danger here of the potential growth of the elements of $\mathrm{diag}(\Phi_{11}^{(k)}, \ldots, \Phi_{pp}^{(k)})$ since these matrices are orthogonal.

Such block-fast-scaled transformation can be derived for the general two-sided process of the form (2). In the case of complex matrices, such fast-scaling is possible, since unitary matrices also have the CS decomposition. Even some structured matrices like J-orthogonal (or J-unitary) allow the CS decomposition which gives further applications of this approach.

However, for such purposes the existing algorithms for computing the CS decomposition are not appropriate and new algorithms are needed. A way how to compute the full CS decomposition of orthogonal matrices is proposed in [1], and a further research to find the most appropriate algorithm is under way.

References

[1] V. Hari, Accelerating the SVD block-Jacobi method. Proposed for publication in Computing.

[2] G. W. Stewart, Computing the CS decomposition of a partitioned orthonormal matrix, Numer. Math., **40** (1982), pp. 297–306.

[3] C. F. Van Loan, Computing CS and the generalized singular value decomposition., Numer. Math., **46** (1985), pp. 479–492.