

# New Class of Block Matrix Orderings for the Parallel Two-Sided Jacobi SVD Algorithm

Gabriel Okša<sup>\*</sup>   Ondrej Sýkora<sup>†</sup>   Marián Vajteršic<sup>‡</sup>

The computation of a singular value decomposition (SVD) belongs to the most intensive computational tasks in the numerical linear algebra. For this task, one- and two-sided block Jacobi SVD algorithms can be used. They are based on the decrease of the Frobenius norm of off-diagonal matrix blocks by means of one- or two-sided unitary (orthogonal) transforms which diagonalize the smaller 2-by-2 block subproblems.

The order in which the individual subproblems are solved is important for the convergence of the Jacobi algorithm, and it is given usually as a prescribed, static list according to which the off-diagonal matrix blocks are combined. During one *sweep* of the algorithm, each off-diagonal matrix block is nullified exactly once. The main disadvantage of such approach is the fact that it does not take into account the actual status of a matrix under consideration. It may happen that some off-diagonal blocks with too small Frobenius norms are combined during one iteration step so that the decrease of the overall off-diagonal Frobenius norm is not optimal. On the other hand, one would like to decrease the off-diagonal norm as much as possible in each iteration step because this approach would lead to the (substantial) reduction of the number of iteration steps needed for the convergence.

Based on these considerations, it has been designed, implemented and tested the parallel two-sided block-Jacobi algorithm with the so-called *dynamic ordering* of subproblems. On a parallel computer with  $p$  processors, one parallel iteration step corresponds to  $p$  serial iteration steps. Then the task of decreasing the Frobenius norm of the off-diagonal blocks as much as possible can be formulated in terms of graph theory as the *maximum-weight perfect matching problem*. The nodes of the complete graph are numbered from 0 to  $c - 1$ , where  $c$  is the blocking factor for the column-wise partition of matrix  $A$ , and the edge  $(i, j)$ ,  $i < j$  has the weight equal to the sum of the Frobenius norms of matrix blocks  $A_{ij}$  and  $A_{ji}$ . The complexity of the whole parallel algorithm was analyzed in detail for  $c = 2p$ . In particular, it was shown that the complexity of the greedy approach for the solution of the maximum-weight perfect matching problem is of order  $O(n^2/p + p^2 \log p)$  per

---

<sup>\*</sup>Institute of Mathematics, Slovak Academy of Sciences, Bratislava, Slovak Republic

<sup>†</sup>Loughborough University, Loughborough, United Kingdom

<sup>‡</sup>University of Salzburg, Salzburg, Austria

one iteration step, whereas the complexity of 2-by-2 SVD subproblems including the matrix multiplications is of order  $O(n^3/p^3 + n^3/p^2)$  per iteration step. Hence, the solution of the maximum-weight perfect matching problem does not represent any large overhead when compared with the number of arithmetic operations needed for the local SVD in each processor.

Recently, we have extended the concept of dynamic ordering further on. Our new strategy for accelerating the convergence of the parallel Jacobi SVD algorithm is based on the distinction between the *physical* and *logical* blocking factor. Suppose a matrix  $A$  is divided into  $p$  block columns so that each processor has one block column. For a fixed number of processors this is a fixed *physical* blocking factor. Then the logical blocking factor  $\ell$  which is connected to the physical blocking factor  $p$  via  $\ell = p/r$  for some integer  $r$ . Consider a complete graph with  $p$  nodes where the edge  $(i, j)$ ,  $i < j$ , has the weight equal to the sum of the Frobenius norms of matrix blocks  $A_{ij}$  and  $A_{ji}$ . Notice that the blocks on the level of physical blocking factor  $p$  are considered for the construction of the complete graph. In one parallel iteration step,  $\ell/2$  local SVDs are computed in parallel where each SVD comprises  $2r$  off-diagonal blocks of  $A$  defined by physical blocking factor  $p$ . To decrease the off-diagonal Frobenius norm of  $A$  maximally is equivalent to finding the partition of complete graph onto  $\ell/2$  disjunct cliques of size  $r$  where the weight of cliques (i.e. the sum of weights through the edges belonging to cliques) is maximized. This task is a generalization of the maximum-weight perfect matching problem on a complete graph which we have already used above. However, a huge step in the computational complexity arises. While the maximum-weight perfect matching problem on a complete graph has an optimal polynomial algorithm, the maximum-weight perfect clique problem on a complete graph, with the size of each clique larger than 2, is NP-hard.

To solve the problem approximately, we have designed and implemented a serial genetic algorithm for finding the cliques, which is called in processor 0 at the beginning of each parallel iteration step in the two-sided parallel block-Jacobi method. This partition into cliques provides the ordering of off-diagonal matrix blocks, which is then broadcasted to all processors. Processors are then grouped into  $\ell/2$  contexts, each context having  $r$  processors. One context is responsible for solving one SVD sub-problem of the block size  $r \times r$ .

The whole SVD algorithm was implemented on a cluster of personal computers using the MPI communication library. We report first results from our numerical experiments.

**Acknowledgement.** This work was supported in part by the VEGA Grant no. 2/4136/26 (Slovak Republic), and by the EPSRC Grant no. EP/C513053/1 (United Kingdom).