11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

LARGE SVD COMPUTATIONS FOR ANALYSIS OF INVERSE PROBLEMS IN GEOPHYSICS

SERGEY A. SOLOVYEV[*] AND SEBASTIEN TORDEUX[†]

[*] Institute of Petroleum Geology and Geophysics SB RAS
3 Koptug pr., Novosibirsk, Russia, 630090, Russia
e-mail: solovevsa@ipgg.sbras.ru

[†]Inria Bordeaux Sud-Ouest, Equipe-Projet Magique-3D IPRA-LMA
Université de Pau et des Pays de l'Adour
BP 1155, 64013 Pau Cedex, Université de Pau, France
e-mail: sebastien.tordeux@gmail.com

Key words: SVD algorithm, Cross Approximation (CA), Low-rank approximation, High-performance computing, Parallel computations

Abstract. This paper describes the implementation of a new algorithm to compute the Truncated Singular Value Decomposition (T-SVD) of matrices with fast decreasing singular values (such as Born approximation matrices). This method is based on a Low-rank approximations which extracts the most important information contained in the matrix. The largest singular values and their left and right singular vectors can then be approximated numerically without performing any operation using the full matrix. This property decreases significantly the memory usage and increases the performance (FLOPS) while getting the T-SVD. The low-rank approximation is computed thanks to the Cross Approximation (CA) technique. Validations tests demonstrate the accuracy of the method, both in terms of singular values and singular vectors. High performance of matrix-matrix operations on intermediate steps is archived by using BLAS and LAPACK components from Intel Math Kernel Library (Intel MKL) that is optimized for Intel architecture and parallelized via OpenMP. Performance tests showed more than ten times performance on one-thread system. Algorithm has large opportunity for parallelization both on shared memory systems (using OMP parallelization) and on distributed ones (MPI parallelization).

1 INTRODUCTION

Linearized inverse theories appear to be powerful tools, lending interesting insight into the physical interpretation of various methods of inversion and migration. The SVD analysis is nowadays at the heart of solving and analyzing inverse problems in geophysics

[1, 2]. However, the computation of the T-SVD of large sized problem is very expensive. There are different ways to overcome this limitation. Most of them are using HPC cluster parallelization algorithms and some simplifications of the input model [3, 4, 5]. In this paper, we present an algorithm to compute the T-SVD of a matrix A, whose number of rows (m) is significantly greater than its number of columns (n) and whose singular values of A are decreasing very fast:

1. $m < n$;
2. $d_0 > d_1 > \ldots >> d_i >> d_m$

This is the type of matrices that are encountered in the problem of inversing seismic data with Born approximation.

So far, algorithms for computing SVD are based on robust arithmetic. These algorithms are implemented in many packages, optimized for high-performance computing, like Intel Math Kernel Library (MKL). However recently, the computational models become large and traditional SVD algorithms are pretty expensive. In this regard, considerable attention has recently been given to low rank arithmetic and matrix formats, like block low rank (BLR) and hierarchically semiseparable (HSS) one [6, 7]. Matrices with properties 2 are ideally suited for low-rank SVD algorithms, so in this paper we adopt this approach.

2 STATEMENT OF THE PROBLEM

Let us consider matrix $A \in R^{m \times n}$ with properties 2. Exact SVD is a factorization of the matrix A with the form

$$\begin{array}{l} A = UDV^T, \\ UU^T = I, \\ VV^T = I, \\ U = \{u_1, \ldots u_n\} \in R^{m \times n}, \\ V = \{v_1, \ldots v_n\} \in R^{n \times n}, \\ D = diag\{d_i\}_{i=1}^n \in R^n, \end{array} \tag{1}$$

where the matrices U and V, contain the left and right singular vectors u_i and v_i. The matrix D is diagonal and contains the singular values d_i. The T-SVD is an approximation (when property 2 is satisfied) of the exact SVD obtained by removing the singular values $d_{k+1}, d_{k+2}, \ldots$ which are lower than a small parameter ε ($\varepsilon > d_{k+1} > d_{k+1} > \ldots$)

$$\begin{array}{l} A \approx U_k D_k V_k^T, \\ U_k U_k^T = I, \\ V_k V_k^T = I, \\ U_k = \{u_1, \ldots u_k\} \in R^{m \times k}, \\ V_k = \{v_1, \ldots v_k\} \in R^{n \times k}, \\ D_k = diag\{d_i\}_{i=1}^k \in R^k, \end{array} \tag{2}$$

Computing the exact SVD in full-arithmetic is a pretty expensive algorithm. In Intel MKL, it takes $(4/3)n^2(3m-n)$ arithmetic operations (FLOPS).

We aim in this paper in looking for matrices $\overline{U}_k = \{\bar{u}_1, \dots \bar{u}_k\} \in R^{m\times k}$, $\overline{V}_k = \{\bar{v}_1, \dots \bar{v}_k\} \in R^{n\times k}$ and $\overline{D}_k = diag\{\bar{d}_i\}_{i=1}^k \in R^k$ which approximate the T-SVD (1) in the following sense:

1. The difference between the approximate singular values and the exact singular values are smaller than δ

$$\frac{d_i - \bar{d}_i}{d_0} < \delta, \quad 1 < i \leq k \quad \text{ and } \quad d_i < \varepsilon,\ k < i \leq n \tag{3}$$

2. The angles between the approximated and exact left and right singular spaces are smaller than a the parameter δ

$$\angle(U_k, \overline{U}_k) < \delta \quad \text{ and } \quad \angle(V_k, \overline{V}_k) < \delta \tag{4}$$

   where $\angle(A, B) = \arccos(\sigma)$ with σ the smallest singular value of $A^T B$.

Matrices $\overline{U}_k$, $\overline{V}_k$ and $\overline{D}_k$ will be called Low-rank SVD of A.

3 Low-rank approximation of a matrix

In what follows, we will make an intensive use of the Low-rank approximation. We recall here some classical facts about this compression method. A Low-rank approximation of a matrix $A \in \mathbb{R}^{m\times n}$ with *eps*-rank k consists of the product of two matrices $B \in \mathbb{R}^{m\times k}$ and $C \in \mathbb{R}^{n\times k}$ which satisfies:

$$\frac{\|BC^T - A\|}{\|A\|} < \varepsilon \tag{5}$$

When $\|\cdot\|$ is the Euclidean matrix norm, it is given by the largest singular value of the matrix. As we have mentioned above, the computation of singular values is pretty expensive. So, in practice we prefer to use the maximum of the absolute values of the matrix elements as norm.

Matrices B and C should be chosen by minimizing the *eps*-rank k. In practice, k is less than m and n. Three techniques are commonly used to obtain obtain this factorization: the computation of the truncated SVD of the matrix, a QR factorization of the Matrix, or a Cross Approximation (CA) technique which is similar to the incomplete LU factorization.

4 DESCRIPTION OF THE ALGORITHM

The algorithm can be decomposed into four steps.

The **first step** consists in decomposing vertically the matrix A into blocks A_i, see Figure 1, and in performing a low-rank approximation of each block $A_i \in \mathbb{R}^{m_i \times n}$, see Figure 2.

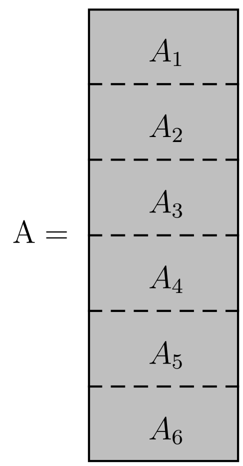

Figure 1: Decomposition of the matrix A by blocks

$$A_i \simeq B_i \, C_i^T \quad \text{with } B_i \in \mathbb{R}^{m_i \times k_i} \text{ and } C_i \in \mathbb{R}^{n \times k_i}$$

Due to the second properties of matrix A, integer number k_i is less than n and, in practice, very small.

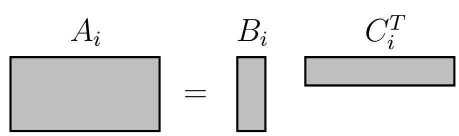

Figure 2: Low-rank approximation of A

To compute the low-rank approximation of blocks we use the Cross Approximation (CA) technique [8], which is more effective than SVD decomposition or QR one (decomposition of a matrix into a product of an orthogonal matrix and an upper triangular matrix). Numerical experiments show the performance superiority of the CA compression method over the SVD and QR techniques. Let us note that apart from standard CA algorithm we have developed a panel CA algorithm, based on modification the process of searching the maximum element of block (in sub-matrix). The search of the maximum is the performance bottle-neck of CA algorithm. Moreover, column-vector multiplications in CA were replaced by product set of columns and set of rows. It significantly improved performance due to block operations more effective than vectors ones.

The result of the first step is depicted in Figure 3. In this picture and next ones, the "plotted" parts of matrices means dense non-zero blocks. The "white" blocks means zero fill-in.

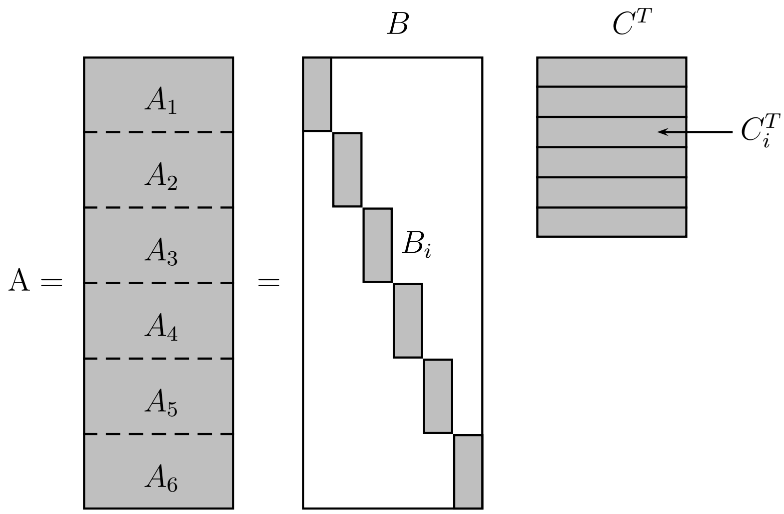

Figure 3: Low-rank approximation of the matrix A

In the **second step**, we orthogonalize the matrices B and C. More precisely, we perform a QR decomposition of matrices B_i and C

$$B_i = \widetilde{B}_i \, R_i \quad and \quad C^T = L \, \widetilde{C}^T,$$

with $\widetilde{B}_i \in \mathbb{R}^{m_i \times k_i}$, $\widetilde{C} \in \mathbb{R}^{n \times k}$ orthogonal, $R_i \in \mathbb{R}^{k_i \times k_i}$ upper triangular and $L \in \mathbb{R}^{k \times k}$ lower triangular where $k = \sum k_i$. The matrices $\widetilde{B}_i$ and R_i are collected into the orthogonal matrix B and into the upper triangular matrix R, see Figure 4. It results that the Low-rank approximation of A has been rewritten as

$$A \simeq \widetilde{B} \; R \; L \; \widetilde{C}^T.$$

During the **third step**, a robust T-SVD with accuracy δ of the product RL is performed

$$RL = U_{RL} D_{RL} V_{RL}^T \qquad \text{with} \quad \frac{\| U_{RL} D_{RL} V_{RL}^T - RL \|}{\| RL \|} < \delta$$

Matrices R and L are much smaller than the initial matrix A. The computation of the robust T-SVD of the product RL is less expensive than the computation of the T-SVD of A. The result of the third step is presented in Figure 4.

During the **fourth step**, we construct the final matrices by computing the products $\overline{U} = \widetilde{B} U_{RL}$, $\overline{V}^T = V_{RL}^T \tilde{C}^T$ and $\overline{D} = D_{RL}$. As result, matrices $\overline{U}$ and $\overline{V}$ have orthogonal columns. Our statement is that decomposition $\overline{U} \; \overline{D} \; \overline{V}^T$ approximates the exact T-SVD of A in sense (3) and (4).

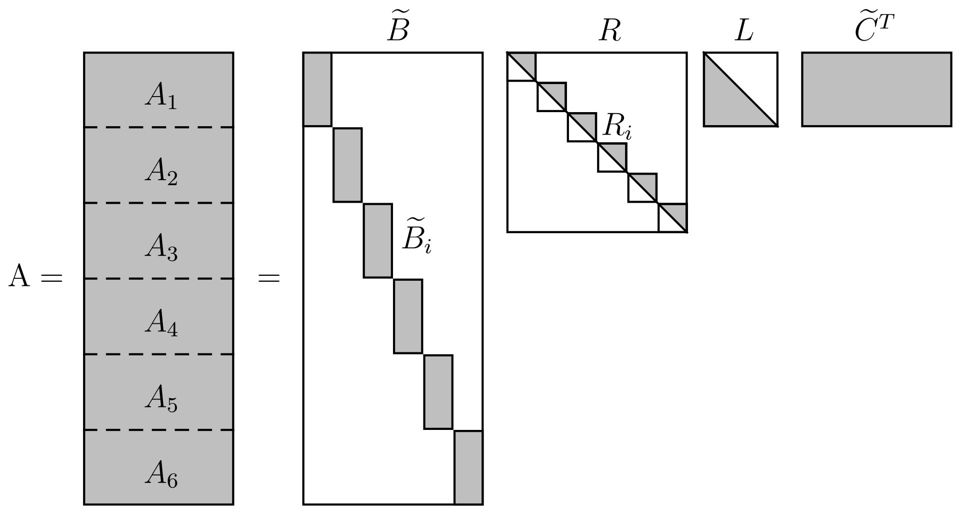

Figure 4: Result of the second step

5 NUMERICAL EXPERIMENTS

The following statements were checked:

1. Dependance of the angle between two subspaces (spanned on columns $\overline{U}$ and U_ε) with respect to the accuracy of low-rank approximation δ and to the threshold ε of cropping the exact SVD, see Figure 5, left image.

2. Dependance of error $\frac{||D_\varepsilon - \overline{D}||}{||\overline{D}||}$ with respect to the accuracy of low-rank approximation δ and threshold ε, see Figure 5, right image.

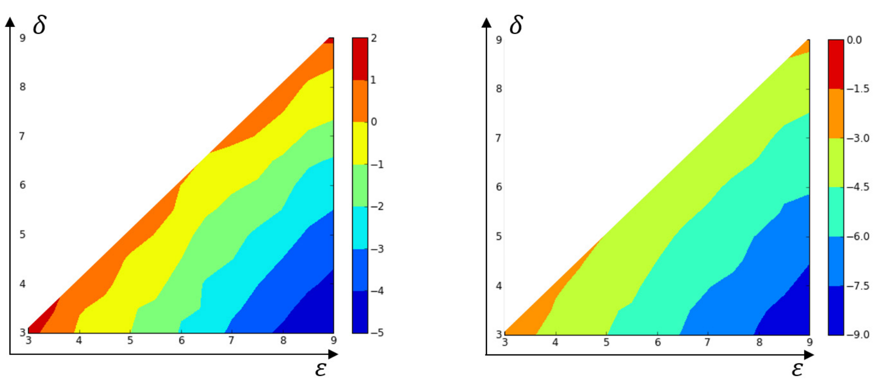

Figure 5: Dependency in $\log_{10}$-scale of the error committed by the low-rank SVD: Angle between subspaces spanned on singular vectors of the cropped exact SVD and Low-rank SVD (left image); singular value error (right image).

The measured time is presented in Table 1. On the preliminary step, matrix A ($m = 29000$, $n = 7200$) separated on $p = 10$ blocks. Accuracy δ of low-rank approximation is 10^{-6}, threshold ε of the cropped exact SVD $U_A D_A V_A^*$ is 10^{-6}. On intermediate steps of our algorithm, we use LAPACK and BLAS functions from Intel MKL. To show the performance gain of the CA approach overs SVD and QR ones (on the first step of proposed algorithm), results of them are presented in the different columns. Computing SVD of full matrix A in robust arithmetic (using LAPACK Intel MKL) take about 970 s. Performance measurement was achieved on Intel Core i7-3770K CPU 3.5 GHz, (Ivy Bridge). To make clear experiments we try to avoid impact of OMP parallelization of all MKL functions by switching off threading (set OMP_NUM_TREADS=1).

Table 1: Performance comparison algorithms of Low-rank SVD based on SVD/QR/ACA approaches

Steps, description \ Approaches	SVD	QRpiv	ACA	Panel ACA
1-st, Get SVD/QR/ACA of all A_i	$610s$	$255s$	$90s$	$18s$
2-nd, Make QR of B and C:	$18s$	$19s$	$19s$	$25s$
3-rd, Perform SVD of RL:	$16s$	$18s$	$18s$	$25s$
4-th, Gathering $\overline{U}$,$\overline{D}$,$\overline{V}$	$7s$	$7s$	$8s$	$8s$
Total time	$651s$	$299s$	$135s$	$76s$

Ranks of truncated matrix are presented in the Table 2:

Table 2: Rank of matrices on intermediate steps

Matrix \ Approaches	SVD	QRpiv	ACA	Panel ACA
After the 1-st step	2366	2424	2454	2795
After the 4-th step	1964	1964	1964	1964

6 CONCLUSIONS

We present a Low-rank SVD algorithm which is more effective than traditional SVD one with robust arithmetic. Validation tests showed that proposed Low-rank SVD approximates very well singular values and spaces. Performance tests showed more than ten time performance gain on one-thread system. Algorithm has large opportunity for parallelization both on shared memory systems (using OMP parallelization) and on distributed ones (MPI parallelization)

7 ACKNOWLEDGEMENTS

The research described was partially supported by RFBR grants 14-01-31340, 14-05-31222, 14-05-00049.

REFERENCES

[1] Menke, W. Geophysical data analysis: Discrete inverse theory. Academic Press (Elsevier), (1994)

[2] Vogel, C. Computational methods for inverse problems: SIAM (2002).

[3] Lebrun, D., V. Richard, D. Mace, and M. Cuer. SVD for multioffset linearized inversion: Resolution analysis in multicomponent acquisition: Geophysics, 66, 871-882 (2001).

[4] Osypov, K., D. Nichols, M. Woodwart, O. Zdraveva, and C. E. Yarman. Uncertainty and resolution analysis for anisotropic tomography using iterative eigendecomposition. 78th Annual International Meeting, SEG, Expanded Abstracts, 3244-3249 (2008).

[5] Silvestrov, I., Neklyudov, D., Puckett, M., and Tcheverda, V. Resolution and stability analysis of offset VSP acquisition scenarios with applications to full-waveform inversion. SEG Technical Program Expanded Abstracts (2012)

[6] Hackbusch, W., A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices: Computing, 62, no. 2, 89108 (1999)

[7] Weisbecker, C., P. Amestoy, O. Boiteau, R. Brossier, A. Buttari, J. LExcellent, S. Operto, and J. Virieux, 3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver, SEG Technical Program Expanded Abstracts 2013, pages 3411-3416

[8] Rjasanow, S. Adaptive Cross Approximation of Dense Matrices. IABEM 2002, International Association for Boundary Element Methods, UT Austin, TX, USA, May 28-30 (2002).