Parallel Computation of the SVD

HALIL SNOPCE[1], ILIR SPAHIU[2]
[1]CST-Faculty, [2]Pedagogical Faculty "Kliment Ohridski"
[1]SEE-University, [2]St. Kiril and Methodius-University
[1]Ilindenska bb. Tetovo, [2] Bulevar "Krste Misirkov" bb
REPUBLIC OF MACEDONIA
[1]h.snopce@seeu.edu.mk , [2]i.spahiu@seeu.edu.mk

Abstract: - In this paper are investigated some methods for parallel computation of the Singular Value Decomposition (SVD) of a matrix. There is done a mathematical analysing of some known techniques. A mathematical approach is base on orthogonal rotations. Finally is given corresponding array for parallel computation based on systolic computation which on the other hand mathematically is based on Hestenes-Jacobi method.

Key-Words: - SVD of a matrix, Jacobi rotations, orthogonal matrix, Hestenes-Jacobi method, communication cost, systolic array.

1 Introduction

The computation of a singular value decomposition of an $m \times n$ matrix A is one of the most demanded tasks in various applications. There are many algorithms for computing the full or partial singular value decomposition. Among them, the Jacobi methods are reputable for the ability to compute the singular values as well as left and right singular vectors with high relative accuracy. The one-sided Jacobi methods for computing the SVD of a rectangular matrix are more efficient than their two sided counterparts, mainly due to the halving of the number of matrix-matrix products needed for updating the left and right singular vectors [10, 11]. Jacobi methods are more appropriate in accuracy point of view in comparison for example with QR methods. The well known sequential bidiagonalization based SVD algorithm takes $O(mn^2)$ time on an $m \times n, m \geq n$ matrix [1]. For large matrices an execution time may be unacceptable. Thus, it is essential to develop efficient parallel algorithms. Both two sided and one-sided techniques have been studied in this context [5,6,8,12]. An improved systolic arrays is discussed in [2].

2 The SVD of a Matrix

Definition: The singular-value decomposition (SVD) of an $m \times n$ matrix A is given by $A = U \Sigma V^T$ where U and V are orthogonal $m \times m$ and $n \times n$ matrices respectively, (i.e., $U^T U = I_m$ and $VV^T = I_n$), and Σ is a diagonal $m \times n$ matrix such that $\Sigma = diag(\sigma_1, \sigma_2, ...)$ and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq \sigma_{r+1} = ... = \sigma_k = 0$, where $r = rankA$. In the case $m = n$, Σ is a square diagonal matrix of order n. The σ_i-s are the singular values of A. The matrix U consists of m left singular vectors and the matrix V consists of n right singular vectors.

The SVD decomposition is often based on diagonalizing rotations which are orthogonal transformations which preserve Eigen values and Eigen vectors as well as singular values and singular vectors. The sequence of rotations A_k, such that $\lim_{k \to \infty} A_k = \Sigma$, is applied during the process.

3 Jacobi Rotations

The classical Jacobi rotation [4] has been used by Jacobi in the 19th century as a tool for solving the least square problem. This rotation is also called Given's rotation. This method uses a sequence of plane rotations to diagonalize a symmetric $n \times n$ real matrix A. We denote the Jacobi rotation of an angle θ in the (i, j) plane by $J(i, j, \theta)$. This is a square matrix equal to the identity matrix except the four additional elements in the intersection of i-th and j-th rows and columns:

$$J(i,j,\theta)=\begin{array}{c|ccccccc|} & & & i & & j & & \\ & 1 & & & & & & \\ & & \cdots & & & & & \\ i & & & c & \cdots & s & & \\ & & & & \cdots & & & \\ j & & & -s & \cdots & c & & \\ & & & & & & \cdots & \\ & & & & & & & 1 \end{array} \quad (1)$$

where $c=\cos\theta$ and $s=\sin\theta$. It is not difficult to verify that this is an orthogonal transformation because of the fact that $J(i,j,\theta)^T J(i,j,\theta)=I$ for each θ. The rotation angle θ is chosen so that it annihilates the symmetrically placed pairs $a_{ij}=a_{ji}$ of the $n(n-1)$ off-diagonal elements. Because only the rows and columns i and j are modified only the example with a matrix of order 2 is analyzed. Initially, let $A=A_1$ and at the $k-th$ iteration $A_{k+1}=J^T A_k J$. The 2×2 representation is as follows:

$$\begin{bmatrix} a_{ii}^{k+1} & 0 \\ 0 & a_{jj}^{k+1} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \cdot \begin{bmatrix} a_{ii}^k & a_{ij}^k \\ a_{ij}^k & a_{jj}^k \end{bmatrix} \cdot \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \quad (2)$$

From (2) we have: $csa_{ii}^k - s^2 a_{ij}^k + c^2 a_{ij}^k - csa_{jj}^k = 0$. After additional transformation of this expression and putting $\alpha = ctg2\theta$ and $t=tg\theta$ we get:

$$\frac{c^2-s^2}{cs} = \frac{a_{jj}^k - a_{ii}^k}{a_{ij}^k} \Rightarrow tg2\theta = \frac{a_{jj}^k - a_{ii}^k}{a_{ij}^k} \quad (3)$$

$$t_1 = -\operatorname{sgn}\alpha(|\alpha| + \sqrt{1+\alpha^2}); \quad t_2 = \frac{\operatorname{sgn}\alpha}{|\alpha| + \sqrt{1+\alpha^2}} \quad (4)$$

From $t = tg\theta = \dfrac{\sin\theta}{\cos\theta} = \dfrac{\sqrt{1-\cos^2\theta}}{\cos\theta}$ we get the values of c and s.

$$c = \cos\theta = \frac{1}{\sqrt{1+t^2}}; \; s = ct \quad (5)$$

In practice, the variant with t_2 is better because in that case the angle satisfies the condition $0 \le |\theta| \le \dfrac{\pi}{4}$.

The convergence is proved using the Frobenius norm $\|A\| = \sqrt{\sum a_{ij}}$ which remains unchanged under orthogonal rotations. In fact, the Jacobi transformation increases the norm of the diagonal elements and it decreases the norm of the off-diagonal elements so that $offA_k = \sqrt{\sum_{i\neq j} a_{ij}} \to 0$ [1]. Hence, A approaches the diagonal form.

The basic problem is to choose the ordering of pairs (i,j) which will be used for zeroing the off-diagonal elements. An objective is to go through all pairs exactly ones. Such a sequence is called a sweep and it consists of $C_n^2 = \dfrac{n(n-1)}{2}$ rotations. A simple cyclic ordering by the rows is the ordering:

$$(1,2),(1,3),\ldots,(1,n),(2,3),\ldots,(2,n),(3,4),\ldots,(n-1,n) \quad (6)$$

The procedure given above is for the case when A is a symmetric matrix. What about a nonsymmetrical matrix A? Even in that case annihilation of the off-diagonal elements can be made using two sided Jacobi method.

In the two sided Jacobi method for SVD of a nonsymetric matrix, the annihilation of the off-diagonal elements is done by using two pairs of rotations [7], such that:

$$\begin{bmatrix} a_{ii}^{k+1} & 0 \\ 0 & a_{jj}^{k+1} \end{bmatrix} = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix}^T \cdot \begin{bmatrix} a_{ii}^k & a_{ij}^k \\ a_{ji}^k & a_{jj}^k \end{bmatrix} \cdot \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} \quad (7)$$

If θ_1 and θ_2 are the angles generating the pairs (s_1,c_1) and (s_2,c_2) then the solution given in [7] is

$$tg(\theta_1+\theta_2) = \frac{a_{ji}^k + a_{ij}^k}{a_{jj}^k - a_{ii}^k}, \quad tg(\theta_2-\theta_1) = \frac{a_{ji}^k - a_{ij}^k}{a_{jj}^k + a_{ii}^k} \quad (8)$$

There is another approach to do the diagonalization using two transformations. The first one is to symmetrize the given matrix A and then to diagonalize the matrix. These two steps are given below:

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}^T \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} p & q \\ q & r \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix}^T \cdot \begin{bmatrix} p & q \\ q & r \end{bmatrix} \cdot \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix} \quad (10)$$

From (9), the following relations can be obtained:

$$s_1 a + c_1 c = c_1 b - s_1 d \Rightarrow ctg\,\theta_1 = \frac{c_1}{s_1} = \frac{a+d}{b-c} = \delta$$
$$\Rightarrow s_1 = \sin\theta_1 = \frac{\operatorname{sgn}\delta}{\sqrt{1+\delta^2}};\ c_1 = \delta s_1 \qquad (11)$$

The first step (symmetrization) is done by using (11). The diagonalization is done by using (10). Afterwards the relations and equations are the same as in (3), (4) and (5).

4 Hestenes-Jacobi Method

Computationally better results offers the so called Hestenes-Jacobi method [3]. This method works with a smaller unit of computation (only with the rows of the matrix) compared with the first explained Jacobi method (which modifies both the rows and columns). Since Hestenes-Jacobi transformations are orthogonal transformations which leave singular values unchanged, it is not important how many such transformations are applied.

For a given $m \times n$ matrix A, the Hestenes-Jacobi method produces an orthogonal matrix U as a product of plane rotations and a matrix N whose rows are orthogonal.

$$UA = N = [\eta_1 \quad \eta_2 \quad \dots \quad \eta_n];\ \eta_i^T \eta_j = 0 \ \ for\ i \neq j \qquad (12)$$

The orthogonal matrix U may be taken as a plane rotation matrix $J(i, j, \theta)$. If $A_1 = A$ then $A_{k+1} = A_k J$. The iterations result in the matrix N defined by (12). Working with a submatrix of type 2×2 we have:

$$\begin{bmatrix} a_i^{k+1} & a_j^{k+1} \end{bmatrix} = \begin{bmatrix} a_i^k & a_j^k \end{bmatrix} \cdot \begin{bmatrix} c & -s \\ s & c \end{bmatrix} =$$
$$= \begin{bmatrix} a_i^k \cdot c + a_j^k \cdot s & -a_i^k \cdot s + a_j^k \cdot c \end{bmatrix} \qquad (13)$$

From the orthogonality condition we have:

$$\left(a_i^k \cdot c + a_j^k \cdot s\right)^T \cdot \left(-a_i^k \cdot s + a_j^k \cdot c\right) = 0 \qquad (14)$$

After a transformation of the relation (14), one can obtain the new relation:

$$\frac{cs}{c^2 - s^2} = \frac{\left(a_i^k\right)^T \cdot a_j^k}{\left(a_i^k\right)^T a_i^k - \left(a_j^k\right)^T a_j^k} \ . \qquad (15)$$

From these relations the iteration formulae for the updating the value of an inner product may be given with this relation:

$$\left(a_i^{k+1}\right)^T \cdot a_j^{k+1} = \left(c^2 - s^2\right)\left(a_i^k\right)^T \cdot a_j^k + cs\left[\left\|a_i^k\right\|^2 - \left\|a_j^k\right\|^2\right] \qquad (16)$$

Writing $g_{ij} = \left(a_i\right)^T \cdot a_j$ the following equality will be fulfilled:

$$\left(c^2 - s^2\right) g_{ij} + cs\left[\left\|a_i^k\right\|^2 - \left\|a_j^k\right\|^2\right] = 0$$

From the equality above there can be obtained the relation:

$$\lambda = ctg 2\theta = \frac{\cos 2\theta}{\sin 2\theta} = \frac{\left\|a_j^k\right\|^2 - \left\|a_i^k\right\|^2}{2 g_{ij}} \qquad (17)$$

The next steps are same like in the relations (4) and (5). Summering this, the results are:

$$t = -\operatorname{sgn}\lambda\left(|\lambda| + \sqrt{1+\lambda^2}\right),\ c = \frac{1}{\sqrt{1+t^2}}\ ;\ s = ct \qquad (18)$$

5 A Parallel Approach

In general, the maximal number of rotation pairs of one sweep for a matrix of order $m \times n$ is $[\max\{m,n\} \cdot \{\max\{m,n\} - 1\}]/2$. If the matrix is $n \times n$ symmetric then one sweep consists of $n(n-1)/2$ transformations, which is the same with the number of elements above or below the main diagonal. Let $T(i, j)$ be the transformation which is used to annihilate the element in the position (i, j). Then, $T(i, j)A = J(i, j, \theta)^T AJ(i, j, \theta)$. For $n = 4$ each sweep consists of 6 transformations. These transformations are $T(1,2), T(1,3), T(1,4), T(2,3), T(2,4), T(3,4)$. So, the general transformation for a given matrix A will be:

$$T(3,4)T(2,4)T(2,3)T(1,4)T(1,3)T(1,2)A\ . \qquad (19)$$

In fig. 1 are presented the modifications of the rows and columns after applying the transformation (19). Annihilated elements are represented by putting the value 0 in the corresponding position:

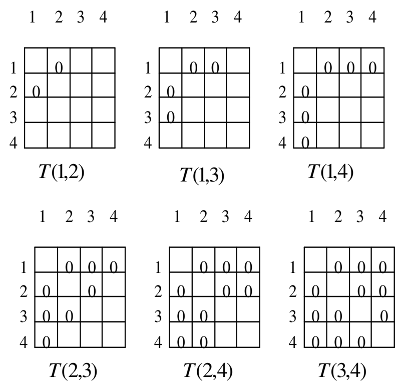


Fig. 1. Sweep of Jacobi transformation for a matrix of order 4

In this case it is important to mention that the off-diagonal elements converge to 0, although it is possible that after the annihilation of some element by using the corresponding transformation, the same element can be filled with non-zero by a subsequent transformation. So, in this case the fact that the process is convergent, allow us to fill this places by 0. In general, transformation $T(1,3)$ will fill the zero elements created by transformation $T(1,2)$ with non-zero elements, $T(1,4)$ will fill those created by $T(1,3)$, and so on. But the important fact here is that the off-diagonal elements decrease from sweep to sweep.

Concerning the parallelization, if n is an even number, there may be performed $n/2$ rotations simultaneously. For $n = 4$ simultaneously may be applied two rotations. It may be done by multiplying by the orthogonal matrix of the form:

$$J_{13,24} = \begin{bmatrix} c_1 & & s_1 & \\ & c_2 & & s_2 \\ -s_1 & & c_1 & \\ & -s_2 & & c_2 \end{bmatrix} \tag{20}$$

This implicates an annihilation which is equivalent by the annihilation using the rotations $J(1,3,\theta)$ and $J(2,4,\theta)$. Using similar multiplication matrices, the steps of the annihilations will be given by:

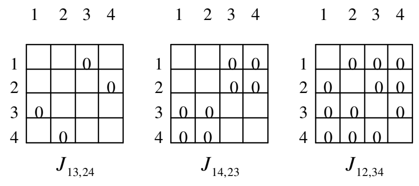


Fig. 2 Sweep of parallel Jacobi transformation for matrix of order 4

If $n = 8$, then four rotations can be performed simultaneously. The matrix used for the annihilation of the first 8 elements is:

$$J_{13,24,57,68} = \begin{bmatrix} c_1 & & s_1 & & & & & \\ & c_2 & & s_2 & & & & \\ -s_1 & & c_1 & & & & & \\ & -s_2 & & c_2 & & & & \\ & & & & c_3 & & s_3 & \\ & & & & & c_4 & & s_4 \\ & & & & -s_3 & & c_3 & \\ & & & & & -s_4 & & c_4 \end{bmatrix} \tag{21}$$

The annihilation of all off diagonal elements will be done in 7 steps. Seven such transformations are the matrix $J_{13,24,57,68}$ (given in (21)) followed by the matrices $J_{14,23,58,67}$, $J_{12,43,56,78}$, $J_{15,26,47,38}$, $J_{16,25,48,37}$, $J_{17,35,28,47}$ and $J_{18,27,36,45}$. Generally, in such kind of parallel processing, the process will be finished in $n-1$ steps compared with the serial case where $n(n-1)/2$ steps are needed for performing. If n is an odd number, then the number of rotational transformations which can be used in parallel to annihilate the off diagonal elements is $(n-1)/2$ and the process will be done in n steps. Let us now take an example where $n = 5$. In this case the first matrix (of order 5) which can be used to annihilate the first 4 off diagonal elements may be defined by:

$$J_{14,25} = \begin{bmatrix} c_1 & & & s_1 & \\ & c_2 & & & s_2 \\ & & & & \\ -s_1 & & & c_1 & \\ & -s_2 & & & c_2 \end{bmatrix} \tag{22}$$

The annihilation will be finished in 5 steps choosing the transformations $J_{14,25}$, $J_{13,24}$, $J_{12,35}$, $J_{45,23}$ and $J_{15,34}$. The steps of annihilations are given by the representation:

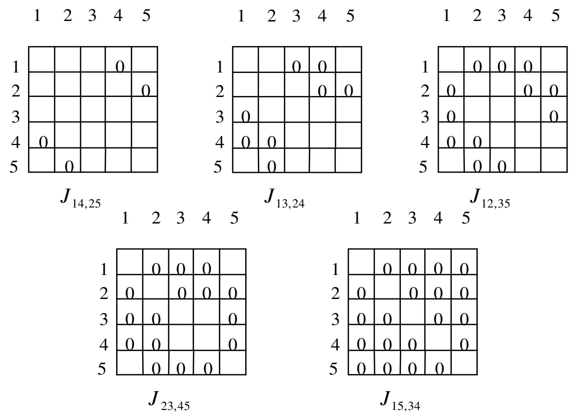


Fig. 3. Sweep of parallel Jacobi transformation for matrix of order 5

In general, for a randomly chosen square matrix of order n (with $n(n-1)$ off diagonal elements), it is possible to annihilate $2\cdot\lfloor n/2\rfloor$ elements at once. The number of transformations of the form $T(i,j)$ is

$$2\cdot\lfloor(n+1)/2\rfloor-1 \tag{23}$$

As a consequence of the analysis done above, a conclusion which determines the number of parallel iterations can be given with the following: **Corollary:** The number of parallel iterations k in a computation of SVD of a square matrix of order n is given by the relation:

$$k=\begin{cases} n, & \textit{if } n \textit{ is odd} \\ n-1, & \textit{if } \textit{ is even}\end{cases}$$

Proof: This is a consequence of the discussion given above. The number k is given by the formula (23). In the case where n is an even number we can use the notation $n=2m$ Therefore we have: $k=2\cdot\lfloor(2m+1)/2\rfloor-1=2\cdot 2m/2-1=n-1$. Otherwise, if n is an odd number then $n=2m+1$ and from (23) we have $k=2\cdot\lfloor(2m+1+1)/2\rfloor-1=n$.

5.1 A Linear Processors Array

The linear array, for $n=8$, is presented in fig. 4. This array consists of 4 processors working in parallel and annihilating the off diagonal elements in 7 steps.

What about the total communication cost (C) which in this case is equal to the total number of transmissions between processors? From fig. 4 we can see that each processor transmits only ones in each step. Taking into the consideration the fact that in the last $n-1'st$ step there is no more communication, the total cost will be:

$$C=(n-2)p=\frac{(n-2)n}{2} \tag{24}$$

The communication cost in this case is of order $O(n^2)$.

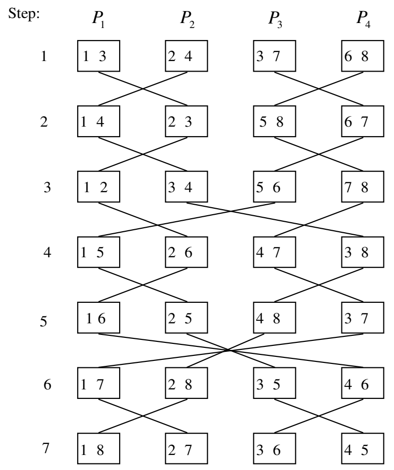
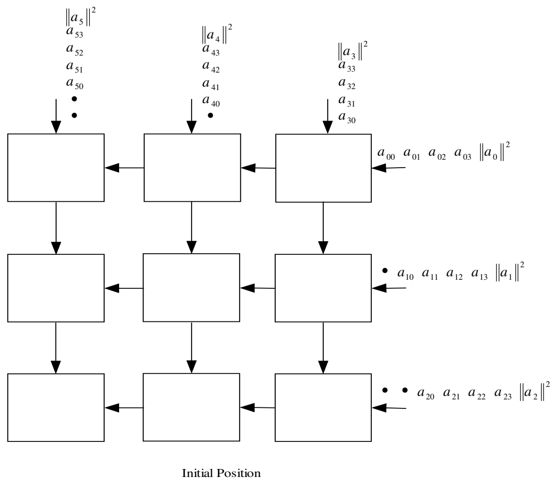


Fig. 4. A linear systolic array

5.2 A Systolic Array Based on Hestenes-Jacobi Method

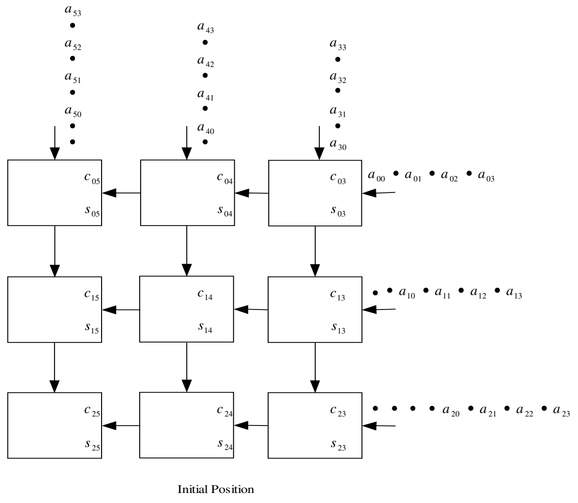
The proposed systolic array using Hestenes method is consisting of $RxR=R^2$ processors. At first it'll compute the value $g_{ij}=(a_i)^T\cdot a_j$, and after the receiving the row norms $\|a_i\|^2$ and $\|a_j\|^2$, each processor computes the rotation values s_{ij} and c_{ij} according the formulas (16) and (17). When the values c_{ij} and s_{ij} are computed, the same array can be used for second step of computation which is consisted in obtaining the correspondent values of matrix A_{k+1}. The initial arrays (for both steps) are presented in fig. 5 and fig.6.

The systolic array shown in fig. 5 and fig. 6 is more efficient in the parallel point of view, although it uses more steps for achieving the convergence. In the presented case there are required 13 time steps in order to compute the values of c_{ij} and s_{ij} (fig. 5), as well as 15 time steps in order to compute the values of A_{k+1} (fig. 6). In general, the total number of time steps is $N+1+4+2(R-1)=N+2R+3$ (fig.5).In the case of fig.6 there are used more number of steps because one needs to use two multiplicative and additive operations. So, the number of time steps in the case of fig.6 is is $4(N+R)+1$. So, the total number of steps will be $5N+6R+4$.



$\|a_5\|^2$ a_{53} a_{52} a_{51} a_{50}
$\|a_4\|^2$ a_{43} a_{42} a_{41} a_{40}
$\|a_3\|^2$ a_{33} a_{32} a_{31} a_{30}
a_{00} a_{01} a_{02} a_{03} $\|a_0\|^2$
a_{10} a_{11} a_{12} a_{13} $\|a_1\|^2$
a_{20} a_{21} a_{22} a_{23} $\|a_2\|^2$
Initial Position

Fig. 1 Systolic computation of rotation angles for R=3 and n=4



a_{53} a_{52} a_{51} a_{50}
a_{43} a_{42} a_{41} a_{40}
a_{33} a_{32} a_{31} a_{30}
c_{05} s_{05} c_{04} s_{04} c_{03} s_{03}
c_{15} s_{15} c_{14} s_{14} c_{13} s_{13}
c_{25} s_{25} c_{24} s_{24} c_{23} s_{23}
a_{00} a_{01} a_{02} a_{03}
a_{10} a_{11} a_{12} a_{13}
a_{20} a_{21} a_{22} a_{23}
Initial Position

Fig. 6 Obtaining the values of A_{k+1} using the values c_{ij} and s_{ij}

6 Conclusion

In this paper is done a theoretical analysis of some parallel methods for computation of the SVD. There is emphasized the parallel systolic computation which is consisted on two basic steps which are dependant. The proposed systolic array doesn't depend of the number of columns. The discussion can be extended making some comparison analysis as well as simulations about the efficiency of each method followed by the analysing of the advantages and disadvantages of each one.

References:

[1] A.Anderson, Z.Bai, C.Bischof, J.Demmel, J.Dongarra, J.Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *Lapack Users'Guide*, second ed., SIAM, Philadelphia, 1999.

[2] Ahmedsaid, A., Amira, A., and Bouridane, A.: *Improved SVD systolic array and implementation on FPGA*, in: IEEE International Conference on Field Programmable Technologie, pp. 3-42, 2003.

[3] Becka, M., Vajtersic, M.: *Block-Jacobi SVD Algorithms for Distributed Memory Systems I: Hypercubes and Rings*, Parallel Algorithms Appl. 13, 265-287 (1999).

[4] Becka, M., Vajtersic, M.: *Block-Jacobi SVD Algorithms for Distributed Memory Systems II: Meshes*. Parallel algorithms Appl. 14, 37-56 (1999).

[5] R.P. Brent and F.T. Luk, "The solution of Singular-Value and Symmetric eigenvalue problems on multiprocessor arrays", SIAM Journal Sci. Stat. Comput., 1985, pp. 69-84.

[6] Z.Drmac, K.Veselic, New fast and accurate Jacobi SVD algorithm: I., LAPACK Working Note 169, August 2005.

[7] George E. Forsythe and Peter Henrici. The cyclic Jacobi Method for Computing the Principal Values of a Complex Matrix. Transactions of the American Mathematical Society, 94(1): 1-23, 1966.

[8] Golub, G., Van Loan, C. : Matrix Computations, 2nd edn. John Hopkins University Press, Baltimore and London 1993.

[9] Magnus R.Hestenes. Inversion of Matrices by Biorthogonalization and Related Results. Journal of the Society for Industrial and Applied Mathematics, 6(1): 51-90, March 1958.

[10] Carl G.J. Jacobi. Uber eine neue Auflosungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen. Astronomishe Nachricten, 22, 1845. English translation by G.W. Stewart, Technical Report 2877, Department of Computer Science, University of Maryland, April 1992.

[11] K.Veselic and V.Hari, a note on a one-sided Jacobi Algorithm, Numer. Math. 56 (1989) 627-633.

[12] Zhou, B., Brent, R.: A Parallel Ring Ordering algorithm for Efficient One-sided SVD Computations. Journal of Parallel and Distributed Computing 42, 1-10 (1997).