IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Aldana et al.	
U.S. Pat. No.:	8,416,862	Attorney Docket No.: 35548-0097IP1
Issue Date:	April 9, 2013	
Appl. Serial No.:	11/237,341	
Filing Date:	September 28, 2005	
Title:	EFFICIENT FEEDBACK OF CHANNEL INFORMATION IN A CLOSED LOOP BEAMFORMING WIRELESS COMMUNICATION SYSTEM	

Declaration of Jacob Robert Munford

1. My name is Jacob Robert Munford. I am over the age of 18, have personal knowledge of the facts set forth herein, and am competent to testify to the same.

2. I earned a Master of Library and Information Science (MLIS) from the University of Wisconsin-Milwaukee in 2009. I have over ten years of experience in the library/information science field. Beginning in 2004, I have served in various positions in the public library sector including Assistant Librarian, Youth Services Librarian and Library Director. I have attached my Curriculum Vitae as Appendix A.

3. During my career in the library profession, I have been responsible for materials acquisition for multiple libraries. In that position, I have cataloged, purchased and processed incoming library works. That includes purchasing materials directly from vendors, recording publishing data from the material in question, creating detailed material records for library catalogs and physically preparing that material for circulation. In addition to my experience in acquisitions, I was also responsible for analyzing large collections of library materials, tailoring library records for optimal catalog

search performance and creating lending agreements between libraries during my time as a Library Director.

4. I am fully familiar with the catalog record creation process in the library sector. In preparing a material for public availability, a library catalog record describing that material would be created. These records are typically written in Machine Readable Catalog (herein referred to as "MARC") code and contain information such as a physical description of the material, metadata from the material's publisher, and date of library acquisition. In particular, the 008 field of the MARC record is reserved for denoting the date of creation of the library record itself. As this typically occurs during the process of preparing materials for public access, it is my experience that an item's MARC record indicates the date of an item's public availability.

5. I have reviewed Exhibit EX1008, an article by B. Yang and J.F. Bohme entitled "Reducing The Computations of the Singular Value Decomposition Array Given By Brent and Luk" (hereto referred to as 'Yang') as presented in *SIAM Journal On Matrix Analysis and Applications* Volume 12, Issue 4.

6. Attached hereto as YA01 is a true and correct copy of the spine, publication data, title page and complete 'Yang' from *SIAM Journal On Matrix Analysis*

and Applications from the University of Pittsburgh library. In comparing YA01 to Exhibit EX1008, it is my determination that Exhibit EX1008 is a true and correct copy of 'Yang'.

7. Attached hereto as YA02 is a true and correct copy of the MARC record describing *SIAM Journal On Matrix Analysis and Application* from the University of Pittsburgh's library. I secured this record myself from the library's online catalog. The 008 field of this MARC record indicates *SIAM Journal On Matrix Analysis and Application* was first cataloged by the University of Pittsburgh library as of September 9, 1987. The item holdings indicate this journal was held in perpetuity since September 1987. This item record also indicates the library's collection includes the Volume 12, Issue 4 publication of *SIAM Journal On Matrix Analysis and Application* containing "Yang".

8. The date stamp on page 4 of YA01 indicates this journal was processed by library staff as of November 1991. Considering this information in concert with the record data from YA02, it is my determination that the Volume 12, Issue 4 edition of *SIAM Journal On Matrix Analysis and Application* was made available and accessible to the public by the University of Pittsburgh library shortly after initial publication and certainly no later than November

1991. Based on journal availability, it is my determination that ‘Yang’ was made available and accessible to the public shortly after initial publication via *SIAM Journal On Matrix Analysis and Application*.

9. I have been retained on behalf of the Petitioner to provide assistance in the above-illustrated matter in establishing the authenticity and public availability of the documents discussed in this declaration. I am being compensated for my services in this matter at the rate of $100.00 per hour plus reasonable expenses. My statements are objective, and my compensation does not depend on the outcome of this matter.

10. I declare under penalty of perjury that the foregoing is true and correct. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: 7/31/19

Jacob Robert Munford

APPENDIX A

Appendix A - Curriculum Vitae

Education

University of Wisconsin-Milwaukee - MS, Library & Information Science, 2009
Milwaukee, WI

- Coursework included cataloging, metadata, data analysis, library systems, management strategies and collection development.
- Specialized in library advocacy and management.

Grand Valley State University - BA, English Language & Literature, 2008
Allendale, MI

- Coursework included linguistics, documentation and literary analysis.
- Minor in political science with a focus in local-level economics and government.

Professional Experience

Library Director, February 2013 - March 2015
Dowagiac District Library
Dowagiac, Michigan

- Executive administrator of the Dowagiac District Library. Located in Southwest Michigan, this library has a service area of 13,000, an annual operating budget of over $400,000 and total assets of approximately $1,300,000.
- Developed careful budgeting guidelines to produce a 15% surplus during the 2013-2014 & 2014-2015 fiscal years.
- Using this budget surplus, oversaw significant library investments including the purchase of property for a future building site, demolition of existing buildings and building renovation projects on the current facility.
- Led the organization and digitization of the library's archival records.
- Served as the public representative for the library, developing business relationships with local school, museum and tribal government entities.

- Developed an objective-based analysis system for measuring library services - including a full collection analysis of the library's 50,000+ circulating items and their records.

November 2010 - January 2013
Librarian & Branch Manager, Anchorage Public Library
Anchorage, Alaska

- Headed the 2013 Anchorage Reads community reading campaign including event planning, staging public performances and creating marketing materials for mass distribution.
- Co-led the social media department of the library's marketing team, drafting social media guidelines, creating original content and instituting long-term planning via content calendars.
- Developed business relationships with The Boys & Girls Club, Anchorage School District and the US Army to establish summer reading programs for children.

June 2004 - September 2005, September 2006 - October 2013
Library Assistant, Hart Area Public Library
Hart, MI

- Responsible for verifying imported MARC records and original MARC cataloging for the local-level collection as well as the Michigan Electronic Library.
- Handled OCLC Worldcat interlibrary loan requests & fulfillment via ongoing communication with lending libraries.

Professional Involvement

Alaska Library Association - Anchorage Chapter

- Treasurer, 2012

Library Of Michigan

- Level VII Certification, 2008
- Level II Certification, 2013

Michigan Library Association Annual Conference 2014

- New Directors Conference Panel Member

Southwest Michigan Library Cooperative

- Represented the Dowagiac District Library, 2013-2015

Professional Development

Library Of Michigan Beginning Workshop, May 2008
Petoskey, MI

- Received training in cataloging, local history, collection management, children's literacy and reference service.

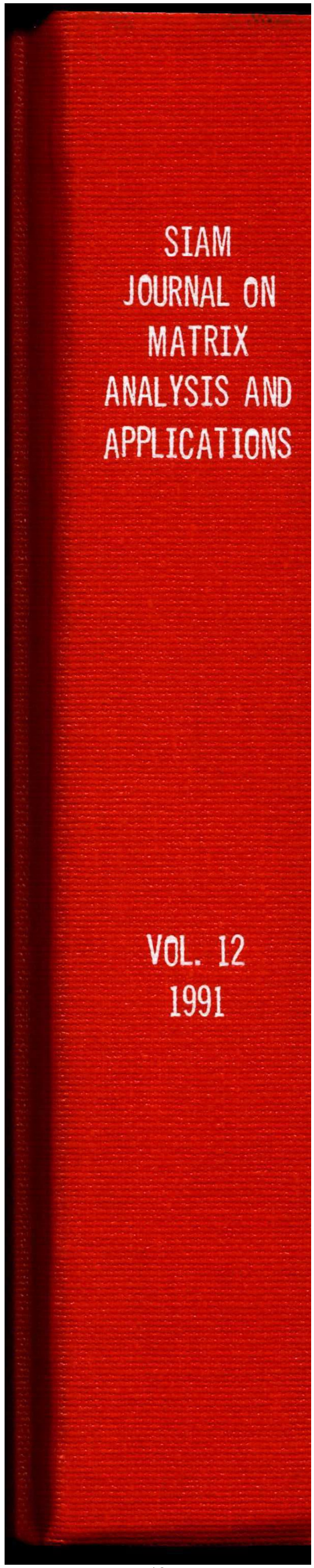
Public Library Association Intensive Library Management Training, October 2011
Nashville, TN

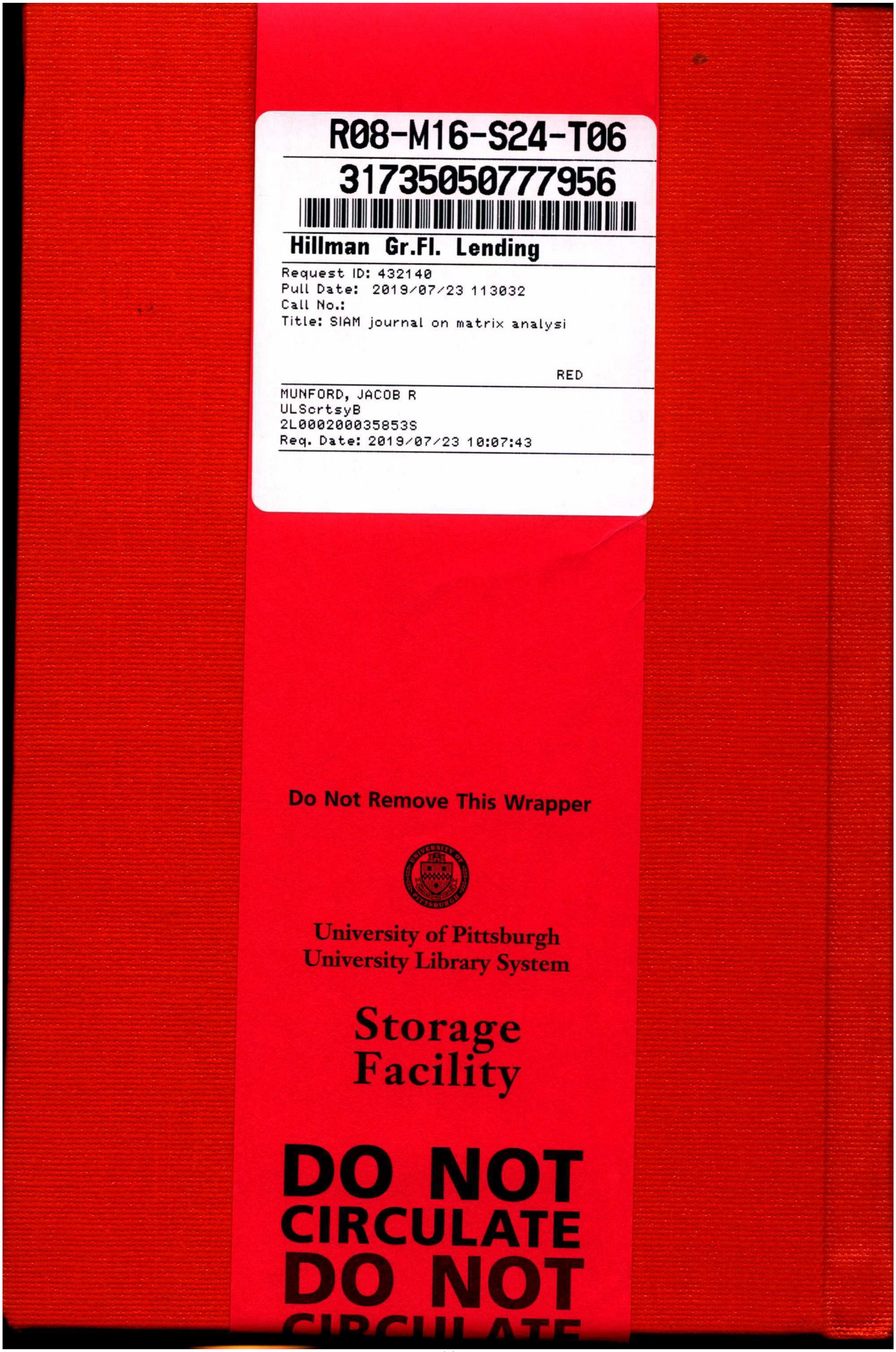
- Attended a five-day workshop focused on strategic planning, staff management, statistical analysis, collections and cataloging theory.

Alaska Library Association Annual Conference 2012 - Fairbanks, February 2012
Fairbanks, AK

- Attended seminars on EBSCO advanced search methods, budgeting, cataloging, database usage and marketing.

YA01


SIAM
JOURNAL ON
MATRIX
ANALYSIS AND
APPLICATIONS
VOL. 12
1991


R08-M16-S24-T06
31735050777956
Hillman Gr.Fl. Lending
Request ID: 432140
Pull Date: 2019/07/23 113032
Call No.:
Title: SIAM journal on matrix analysi
RED
MUNFORD, JACOB R
ULSortsyB
2L000200035853S
Req. Date: 2019/07/23 10:07:43
Do Not Remove This Wrapper
University of Pittsburgh
University Library System
Storage
Facility
DO NOT
CIRCULATE
DO NOT

SIAM JOURNAL ON

Matrix Analysis and Applications

VOLUME 12 1991

SIAM JOURNAL ON

Matrix Analysis and Applications

OCTOBER 1991 Volume 12, Number 4

Publisher: Vickie H. Kearn. **Production Manager:** J. Corey Gray. **Managing Editor:** Tricia Manning. **Senior Production Editor:** Crystal G. Norris. **Copy Editor:** Beth Gallagher. **Editorial Assistant:** David Livewell. **Production Coordinator:** Nancy H. Abbott. **Reprints Coordinator:** Jill M. Davis.

SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS (ISSN 0895-4798) is published quarterly in January, April, July, and October by the Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688. Second-class postage paid at Philadelphia, Pennsylvania and additional mailing offices. POSTMASTER: Send address change to *SIAM Journal on Matrix Analysis and Applications*, 3600 University City Science Center, Philadelphia, PA 19104-2688.

Subscriptions: Annual subscription list price: $130.00 (domestic), $155.00 (overseas); annual individual member subscription price: $40.00 (domestic), $43.00 (overseas). Subscriptions are available on a calendar-year basis only. Send orders to Customer Service, SIAM, 3600 University City Science Center, Philadelphia, PA 19104-2688. In Japan send orders to USACO Corporation, 13-12, Shimbashi, 1-chome, Minato-ku, Tokyo 105, tel. 03(502)-6471. All back volumes are available; prices will be provided on request. **Manuscript Submissions:** See Instructions to Authors on page iv of this journal. **Advertising:** Accepted for all publications. Inquiries should be directed to the Marketing Manager, SIAM Publications, 3600 University City Science Center, Philadelphia, PA 19104-2688.

TAPSCO, Inc., Akron, Pennsylvania, compositor; Lancaster Press, Lancaster, Pennsylvania, printer.

The *SIAM Journal on Matrix Analysis and Applications* is a continuation of the *SIAM Journal on Algebraic and Discrete Methods* (ISSN 0196-5212).

SIAM J. MATRIX ANAL. APPL.
Vol. 12, No. 4, pp. 713–725, October 1991

009

REDUCING THE COMPUTATIONS OF THE SINGULAR VALUE DECOMPOSITION ARRAY GIVEN BY BRENT AND LUK*

B. YANG† AND J. F. BÖHME†

Abstract. A new, efficient, two-plane rotation (TPR) method for computing two-sided rotations involved in singular value decomposition (SVD) is presented. It is shown that a two-sided rotation can be evaluated by only two plane rotations and a few additions. This leads to significantly reduced computations. Moreover, if coordinate rotation digital computer (CORDIC) processors are used for realizing the processing elements (PEs) of the SVD array given by Brent and Luk, the computational overhead of the diagonal PEs due to angle calculations can be avoided. The resulting SVD array has a homogeneous structure with identical diagonal and off-diagonal PEs. Similar results can also be obtained if the TPR method is applied to Luk's triangular SVD array and to Stewart's Schur decomposition array.

Key words. singular value decomposition, systolic arrays, CORDIC, two-sided rotations, VLSI

AMS(MOS) subject classification. 15A18

1. Introduction. One important problem in linear algebra and digital signal processing is the singular value decomposition (SVD). Typical applications arise in beamforming and direction finding, spectrum analysis, digital image processing, etc. [1]. Recently, there has been a massive interest in parallel architectures for computing SVD because of the high computational complexity of SVD, the growing importance of real-time signal processing, and the rapid advances in very large scale integration (VLSI) that make low-cost, high-density and fast processing memory devices available.

There are different numerically stable methods for computing complete singular value and singular vector systems of dense matrices, for example, the Jacobi SVD method, the QR method, and the one-sided Hestenes method. For parallel implementations, the Jacobi SVD method is far superior in terms of simplicity, regularity, and local communications. Brent, Luk, and Van Loan have shown how the Jacobi SVD method with parallel ordering can be implemented by a two-dimensional systolic array [2], [3]. Various coordinate rotation digital computer (CORDIC) realizations of the SVD array have been reported by Cavallaro and Luk [4] and Delosme [5], [6].

The Jacobi SVD method is based on, as common for all two-sided approaches, applying a sequence of two-sided rotations to 2×2 submatrices of the original matrix. The computational complexity is thus determined by how to compute the two-sided rotations. In most previous works, a two-sided rotation is evaluated in a straightforward manner by four plane rotations, where two of them are applied from left to the two column vectors of the 2×2 submatrix and the other ones are applied from right to the row vectors, respectively. In the diagonal processing elements (PEs), additional operations for calculating rotation angles are required. This leads to an inhomogeneous array architecture containing two different types of PEs.

In this paper, we develop a two-plane rotation (TPR) method for computing two-sided rotations. We show that the above computational complexity can be reduced significantly because each two-sided rotation can be evaluated by only two plane rotations and a few additions. Moreover, the SVD array given by Brent and Luk becomes homogeneous with identical diagonal and off-diagonal PEs when CORDIC processors are

* Received by the editors September 28, 1989; accepted for publication (in revised form) August 2, 1990.
† Department of Electrical Engineering, Ruhr-Universität Bochum, 4630 Bochum, Germany.

used. In a recent work [6], Delosme has also indicated this possibility in connection with "rough rotations" independently. He has taken, however, a different approach that is based on encoding the rotation angles. He has still required four plane rotations on the off-diagonal PEs while diagonal and off-diagonal operations can be overlapped.

Our paper is organized as follows. In § 2, we briefly reexamine Jacobi's SVD method and Brent and Luk's SVD array. Then, we develop the TPR method in § 3. The CORDIC algorithm is described in § 4, where in particular CORDIC scaling correction techniques are discussed and examples of scaling-corrected CORDIC sequences are given. In § 5, a unified CORDIC SVD module for all PEs of the SVD array is presented. This module is compared to those proposed by Cavallaro, Luk, and Delosme in § 6. Finally, we stress the applicability of the TPR method to several other problems.

2. Jacobi SVD method. In this paper, we consider real, square, and nonsymmetric matrices. Let $M \in \mathbb{R}^{N \times N}$ be a matrix of dimension N. The SVD is given by

$$M = U \Sigma V^T, \tag{1}$$

where $U \in \mathbb{R}^{N \times N}$ and $V \in \mathbb{R}^{N \times N}$ are orthogonal matrices containing the left and right singular vectors, and $\Sigma \in \mathbb{R}^{N \times N}$ is a diagonal matrix of singular values, respectively. The superscript T denotes matrix transpose. Based on an extension of the Jacobi eigenvalue algorithm [7], Kogbetliantz [8] and Forsythe and Henrici [9] proposed to diagonalize M by a sequence of two-sided rotations,

$$M_0 = M, \qquad M_{k+1} = U_k^T M_k V_k \qquad (k = 0, 1, 2, \cdots). \tag{2}$$

U_k and V_k describe two rotations in the (i, j)-plane ($1 \leqq i < j \leqq N$), where the rotation angles are chosen to annihilate the elements of M_k at the positions (i, j) and (j, i). Usually, several sweeps are necessary to complete the SVD, where a sweep is a sequence of $N(N-1)/2$ two-sided rotations according to a special ordering of the $N(N-1)/2$ different index pairs (i, j).

For sequential computing on a uniprocessor system, possibly the most frequently used orderings are the cyclic orderings, namely, the cyclic row ordering

$$(i,j) = (1,2), (1,3), \cdots, (1,N), (2,3), \cdots, (2,N), \cdots, (N-1,N) \tag{3}$$

or the equivalent cyclic column ordering. Sameh [10] and Schwiegelshohn and Thiele [11] have shown how to implement the cyclic row ordering on a ring-connected or a mesh-connected processor array. Recently, a variety of parallel orderings have been developed. Luk and Park [12] have shown that these parallel orderings are essentially equivalent to the cyclic orderings and thus share the same convergence properties.

Brent and Luk have suggested a particular parallel ordering and developed a square systolic array consisting of $\lceil N/2 \rceil \times \lceil N/2 \rceil$ PEs for implementing the Jacobi SVD method (Fig. 1). To do this, the matrix M is partitioned into 2×2 submatrices. Each PE contains one submatrix and performs a two-sided rotation

$$B = R(\theta_1)^T A R(\theta_2), \tag{4}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \tag{5}$$

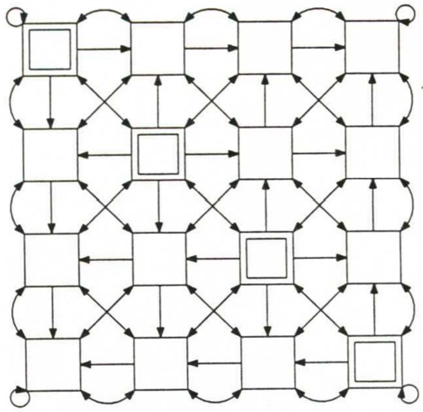


FIG. 1. *The* SVD *array given by Brent and Luk.*

denote the submatrix before and after the two-sided rotation, respectively, and

$$R(\theta)=\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{6}$$

describes a plane rotation through the angle θ. At first, the diagonal PEs (symbolized by a double square in Fig. 1) generate the rotation angles to diagonalize the 2×2 submatrices ($b_{12}=b_{21}=0$) stored in them. This means that θ_1 and θ_2 are first calculated from the elements of A and then relation (4) is used to compute b_{11} and b_{22}. We call this the generation mode. Then, the rotation angles are sent to all off-diagonal PEs in the following way: the angles associated to the left-side rotations propagate along the rows while the angles associated to the right-side rotations propagate along the columns. Once these angles are received, the off-diagonal PEs perform the two-sided rotations (4) on their stored data. We call this the rotation mode. Clearly, if we compute the rotation mode straightforwardly, we require four plane rotations. For the generation mode, additional operations for calculating θ_1 and θ_2 are required.

3. TPR method for computing two-sided rotations. In order to develop the TPR method for computing two-sided rotations more efficiently, we first discuss the commutative properties of two special types, the rotation-type and the reflection-type, of 2×2 matrices. We define

$$\mathcal{M}^{\text{rot}}=\left\{\begin{pmatrix} x & y \\ -y & x \end{pmatrix}\middle| x,y\in\mathbb{R}\right\} \quad\text{and}\quad \mathcal{M}^{\text{ref}}=\left\{\begin{pmatrix} x & y \\ y & -x \end{pmatrix}\middle| x,y\in\mathbb{R}\right\}. \tag{7}$$

The former is called rotation-type because it has the same matrix structure as a 2×2 plane rotation matrix. Similarly, the latter is called reflection-type because it has the same matrix structure as a 2×2 Givens reflection matrix [13]. Note that x and y must not be normalized to $x^2+y^2=1$. Using the above definitions, the following results can be shown by some elementary manipulations.

LEMMA 1. *If* $A_1\in\mathcal{M}^{\text{rot}}$ *and* $A_2\in\mathcal{M}^{\text{rot}}$, *then* $A_1A_2=A_2A_1\in\mathcal{M}^{\text{rot}}$.

LEMMA 2. *If* $A_1\in\mathcal{M}^{\text{ref}}$ *and* $A_2\in\mathcal{M}^{\text{rot}}$, *then* $A_1A_2=A_2^TA_1\in\mathcal{M}^{\text{ref}}$.

In particular, if we consider two plane rotations, we know the following.

LEMMA 3. *If* $R(\theta_1)$ *and* $R(\theta_2)$ *are plane rotations described by* (6), *then* $R(\theta_1)R(\theta_2)=R(\theta_1+\theta_2)$ *and* $R(\theta_1)^TR(\theta_2)=R(\theta_2-\theta_1)$.

Now, we give a theorem describing the rotation mode of the TPR method.

THEOREM. *If the* 2×2 *matrix* A *and the two rotation angles* θ_1 *and* θ_2 *are given, then the two-sided rotation* (4) *can be computed by two plane rotations, ten additions,* *and four scalings by* $\frac{1}{2}$:

$$(8)\qquad p_1=(a_{22}+a_{11})/2, \qquad p_2=(a_{22}-a_{11})/2,$$
$$q_1=(a_{21}-a_{12})/2, \qquad q_2=(a_{21}+a_{12})/2,$$

$$(9)\qquad \theta_- = \theta_2-\theta_1, \qquad \theta_+=\theta_2+\theta_1,$$

$$(10)\qquad \begin{pmatrix} r_1 \\ t_1 \end{pmatrix} = R(\theta_-)\begin{pmatrix} p_1 \\ q_1 \end{pmatrix}, \qquad \begin{pmatrix} r_2 \\ t_2 \end{pmatrix} = R(\theta_+)\begin{pmatrix} p_2 \\ q_2 \end{pmatrix},$$

$$(11)\qquad b_{11}=r_1-r_2, \qquad b_{12}=-t_1+t_2,$$
$$b_{21}=t_1+t_2, \qquad b_{22}=r_1+r_2.$$

Proof. Using (8), the matrix A can be reformulated as

$$A=A_1+A_2=\begin{pmatrix} p_1 & -q_1 \\ q_1 & p_1 \end{pmatrix}+\begin{pmatrix} -p_2 & q_2 \\ q_2 & p_2 \end{pmatrix}.$$

Clearly, $R(\theta_1)$, $R(\theta_2)$ in (4) and A_1 are elements of $\mathcal{M}^{\text{rot}}$ while A_2 belongs to $\mathcal{M}^{\text{ref}}$. This leads to the following reformulation of the matrix B by using Lemmas 1–3:

$$\begin{aligned} B &= R(\theta_1)^T A R(\theta_2) \\ &= R(\theta_1)^T A_1 R(\theta_2) + R(\theta_1)^T A_2 R(\theta_2) \\ &= R(\theta_1)^T R(\theta_2) A_1 + R(\theta_1)^T R(\theta_2)^T A_2 \\ &= R(\theta_2-\theta_1)A_1 + R(\theta_2+\theta_1)^T A_2 \\ &= R(\theta_-)\begin{pmatrix} p_1 & -q_1 \\ q_1 & p_1 \end{pmatrix} + R(\theta_+)^T\begin{pmatrix} -p_2 & q_2 \\ q_2 & p_2 \end{pmatrix} \\ &= \begin{pmatrix} r_1 & -t_1 \\ t_1 & r_1 \end{pmatrix} + \begin{pmatrix} -r_2 & t_2 \\ t_2 & r_2 \end{pmatrix}. \end{aligned}$$

This completes the proof.

The generation mode of the TPR method follows directly from the above theorem.

COROLLARY. *If the* 2×2 *matrix* A *is given, we can diagonalize* A *and calculate the corresponding rotation angles* θ_1 *and* θ_2 *by two Cartesian-to-polar coordinates conversions, eight additions, and four scalings by* $\frac{1}{2}$:

$$(12)\qquad p_1=(a_{22}+a_{11})/2, \qquad p_2=(a_{22}-a_{11})/2,$$
$$q_1=(a_{21}-a_{12})/2, \qquad q_2=(a_{21}+a_{12})/2,$$

$$(13)\qquad r_1=\text{sign}\,(p_1)\sqrt{p_1^2+q_1^2}, \qquad r_2=\text{sign}\,(p_2)\sqrt{p_2^2+q_2^2},$$
$$\theta_-=\arctan\,(q_1/p_1), \qquad \theta_+=\arctan\,(q_2/p_2),$$

$$(14)\qquad \theta_1=(\theta_+-\theta_-)/2, \qquad \theta_2=(\theta_++\theta_-)/2,$$

$$(15)\qquad b_{11}=r_1-r_2, \qquad b_{22}=r_1+r_2.$$

Proof. Regarding (11), $b_{12}=b_{21}=0$ is equivalent to $t_1=t_2=0$. Equation (13) follows then from (10). This completes the proof.

In equation (13), we choose the rotation through the smaller angle. All vectors lying in the first or the fourth quadrant are rotated onto the positive x-axis, and all vectors lying in the second and the third quadrant are rotated onto the negative x-axis. For vectors on the y-axis, the rotation direction is arbitrary. Thus, the generated rotation angles θ_- and θ_+ satisfy $|\theta_-|, |\theta_+| \leqq 90°$. This results in

$$|\theta_1| \leqq 90° \quad \text{and} \quad |\theta_2| \leqq 90°, \tag{16}$$

due to (14).

Equation (16) is important with respect to the convergence of the Jacobi SVD method. Forsythe and Henrici [9] have proven the convergence for cyclic orderings if the rotation angles θ_1 and θ_2 are restricted to a closed interval inside the open interval (-90°, 90°). They have also demonstrated that this condition may fail to hold, i.e., θ_1 and θ_2 may be $\pm 90°$, if the off-diagonal elements b_{12} and b_{21} in (5) have to be exactly annihilated. As a remedy, they suggested an under- or overrotation by computing the two-sided rotation (4) with angles $(1-\gamma)\theta_1$ and $(1-\gamma)\theta_2$ $(-1 < \gamma < 1)$ and proved its convergence. In practice, however, the finite machine accuracy in the real arithmetic allows only an approximative computation of the rotation angles and implies under- or overrotations. So the Jacobi SVD method converges without using under- or overrotations as shown by the experimental results of Brent, Luk, and Van Loan [3]. In case of CORDIC implementations, the effect of implicit under- or overrotations is more apparent. The angles $\pm 90°$ can never be exactly calculated because of the limited angle resolution arctan (2^{-p}) of the CORDIC algorithm, where p denotes the mantissa length.

4. The CORDIC algorithm. In the previous section, we have seen that the main operations of the TPR-method are plane rotations and Cartesian-to-polar coordinates conversions. These operations can be carried out by multiplier–adder-based processors supported by software or special hardware units. An alternative approach is the use of dedicated processors that usually map algorithms more effectively to hardware. The CORDIC processor is such a powerful one for calculating trigonometric functions.

The CORDIC algorithm was originally designed by Volder [14] as an iterative procedure for computing plane rotations and Cartesian-to-polar coordinates conversions. It was later generalized and unified by Walther [15], enabling a CORDIC processor to calculate more functions, including hyperbolic functions, as well as multiplications and divisions. In the following, we consider Volder's CORDIC algorithm because only trigonometric functions are involved in SVD applications.

The CORDIC algorithm consists of iterative shift-add operations on a three-component vector,

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} x_i - \sigma_i \delta_i y_i \\ y_i + \sigma_i \delta_i x_i \end{pmatrix} = \frac{1}{\cos(\alpha_i)} \begin{pmatrix} \cos(\alpha_i) & -\sigma_i \sin(\alpha_i) \\ \sigma_i \sin(\alpha_i) & \cos(\alpha_i) \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \tag{17}$$

$$z_{i+1} = z_i - \varepsilon \sigma_i \alpha_i \qquad (0 < \delta_i < 1;\ \sigma_i = \pm 1;\ \varepsilon = \pm 1;\ i = 0, 1, \cdots, n-1), \tag{18}$$

in which the iteration stepsize δ_i is defined by

$$\delta_i = \tan(\alpha_i) = 2^{-S(i)}. \tag{19}$$

The set of integers $\{S(i)\}$ parametrizing the iterations is called CORDIC sequence. Equation (17) can be interpreted, except for a scaling factor of

$$k_i = \frac{1}{\cos(\alpha_i)} = \sqrt{1+\delta_i^2}, \tag{20}$$

as a rotation of $(x_i, y_i)^T$ through the angle α_i, where the sign $\sigma_i = \pm 1$ gives the rotation direction. After n iterations, the results are given by

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = K \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \tag{21}$$

$$z_n = z_0 - \varepsilon\alpha, \tag{22}$$

with the overall scaling factor $K = \prod_i k_i$ and the total rotation angle $\alpha = \sum_i \sigma_i \alpha_i$. Now, if the CORDIC sequence satisfies the following convergence condition

$$\alpha_i - \sum_{j=i+1}^{n-1} \alpha_j \leqq \alpha_{n-1} \qquad (i = 0, 1, \cdots, n-2), \tag{23}$$

we can choose the sign parameter

$$\sigma_i = \begin{cases} -\text{sign}\,(x_i y_i) & \text{for } y_n \to 0, \\ \text{sign}\,(\varepsilon z_i) & \text{for } z_n \to 0 \end{cases} \tag{24}$$

to force y_n or z_n to zero, provided that the input data x_0, y_0, and z_0 lie in the convergence region

$$C = \sum_{i=0}^{n-1} \alpha_i \geqq \begin{cases} |\arctan\,(y_0/x_0)| & \text{for } y_n \to 0, \\ |z_0| & \text{for } z_n \to 0. \end{cases} \tag{25}$$

In this way, two different types of CORDIC trigonometric functions can be computed (Table 1). In the mode $y_n \to 0$, the Cartesian coordinate (x_0, y_0) of a plane vector is converted to its polar representation, where the parameter $\varepsilon = \pm 1$ determines the sign of the phase angle calculated. When $z_n \to 0$, a given plane vector is rotated through the angle z_0, where $\varepsilon = \pm 1$ controls the rotation direction.

In Table 1, the principal value $|\arctan\,(y_0/x_0)| \leqq 90°$ of the inverse tangent function is calculated when computing Cartesian-to-polar coordinates conversions. Correspondingly, x_n may be positive or negative according to the sign of x_0. So, it is guaranteed that a vector is always rotated through the smaller angle onto the x-axis in accordance with (13). In this case, a convergence region of $C \geqq 90°$ is sufficient for the generation mode of the two-sided rotation.

One main drawback of the CORDIC algorithm is the need of correcting the scaling factor K that arises during the iterations (17). For example, if we use Volder's CORDIC sequence

$$\{S(i)\} = \{0, 1, 2, 3, \cdots, p-1, p\}, \tag{26}$$

with $n = p + 1$ CORDIC iterations for a mantissa accuracy of 2^{-p}, the scaling factor is $K \approx 1.64676$. Compensating this undesired scaling effect with a minimum number of computations is of particular importance.

Clearly, multiplying x_n and y_n in Table 1 by K^{-1} will degrade the algorithm performance substantially. Most of the scaling correction issues are based on shift-add operations. For a two-sided rotation that is implemented by four plane rotations, each matrix element undergoes two plane rotations so that the total scaling factor to be corrected is K^2. In this case, Cavallaro and Luk [16] have pointed out that there is a simple systematic approach for scaling correction when using the CORDIC sequence (26). They proposed to use $\lceil p/4 \rceil$ scaling iterations of the type $x \leftarrow x - 2^{-2j}x$ with $j \in J = \{1, 3, 5, \cdots, 2\lceil p/4 \rceil - 1\}$ and one shift operation 2^{-1}. The remaining scaling error is

TABLE 1
CORDIC *trigonometric functions* ($\varepsilon = \pm 1$).

$y_n \to 0$	$z_n \to 0$
$x_n = K\,\text{sign}\,(x_0)\sqrt{x_0^2 + y_0^2}$ $z_n = z_0 + \varepsilon \arctan\,(y_0/x_0)$	$x_n = K(x_0 \cos z_0 - \varepsilon y_0 \sin z_0)$ $y_n = K(\varepsilon x_0 \sin z_0 + y_0 \cos z_0)$

bounded by 2^{-p-1},[1]

(27) $$\left|1-2^{-1}\prod_{j\in J}(1-2^{-2j})\cdot K^2\right|=\left|1-2^{-1}\prod_{j\in J}(1-2^{-2j})\cdot\prod_{i=0}^{p}(1+2^{-2i})\right|<2^{-p-1}.$$

This approach, however, fails in the TPR method. Here, each matrix element undergoes only one plane rotation. The scaling factor to be corrected is thus K rather than K^2. In order to solve this more difficult problem, different techniques have been developed in the literature. Haviland and Tuszynski [17] used similar scaling iterations as Cavallaro and Luk. Ahmed [18] repeated some CORDIC iterations to force K to a power of the machine radix. Delosme [19] combined both methods of Haviland, Tuszynski, and Ahmed for minimizing the number of computations. Deprettere, Dewilde, and Udo [20] suggested the double-shift concept.

We designed a computer program [21] for a systematic search of CORDIC sequences. We allow shifts parameters $S(i)$ $(i = 0, 1, \cdots, n-1)$ with differences $S(i+1) - S(i) \in \{0, 1, 2\}$ to provide more optimization freedom. For an efficient scaling correction, we require that the scaling factor K be corrected by a sequence of n_k shift-add operations,

(28) $$2^{-T(0)}\prod_{j=1}^{n_k}(1+\eta(j)2^{-T(j)})\cdot K = 1+\Delta K \qquad (T(j)\,\text{integers}, \eta(j)=\pm 1).$$

These additional scaling iterations are parametrized by the set of signed integers $\{T(0), \eta(1)T(1), \cdots, \eta(n_k)T(n_k)\}$. The total number of iterations is $L = n + n_k$.

In (28), ΔK denotes the remaining relative scaling error after the scaling correction. We emphasize that this is a systematic error with a constant sign. By contrast, the other two types of CORDIC errors, the angular error due to the limited angle resolution and the rounding error, are of statistical nature because they may be positive or negative. The scaling error is thus more critical with respect to error accumulation when repeated CORDIC operations on the same data have to be computed as in SVD applications. Roughly speaking, the total scaling error after k CORDIC function calls increases linearly with k, a fact that has been verified by our numerical experiments. For this reason, we require $|\Delta K|$ to be much smaller than 2^{-p}.

We found catalogues of CORDIC sequences with complexity comparable to those of Cavallaro and Luk. In the following, five examples for different mantissa lengths p = 16, 20, 24, 28, and 32, including the total number of iterations $L = n + n_k$, the convergence region C, and the remaining scaling error ΔK are given:

$p=16$: $\{S(i)\}=\{0\,1\,2\,3\,\cdots\,15\,16\}$,

$\{\eta(j)T(j)\}=\{1\,+2\,-5\,+9\,+10\}$, $L=17+4$, $C\approx 100°$, $\Delta K\approx -2^{-16.01}$,[2]

$p=20$: $\{S(i)\}=\{0\,1\,2\,3\,\cdots\,19\,20\}$,

$\{\eta(j)T(j)\}=\{1\,+2\,-5\,+9\,+10\,+16\}$,

$L=21+5$, $C\approx 100°$, $\Delta K\approx 2^{-23.05}$,

[1] When replacing $\lceil p/4\rceil$ by $\lceil (p-1)/4\rceil$ or $\lceil (p+1)/4\rceil$, the upper bound in (27) becomes 2^{-p} or 2^{-p-2}, respectively.

[2] When appending an additional scaling iteration with $\eta(5)T(5) = +16$, the scaling accuracy can be enhanced to $\Delta K \approx 2^{-23}$.

$$p=24: \quad \{S(i)\}=\{1\,1\,2\,3\,3\,4\,5\,5\,6\,6\,7\,8\,8\,9\,10 \cdots 23\,24\},$$
$$\{\eta(j)T(j)\}=\{0\,-2\,+6\}, \qquad L=29+2, \qquad C\approx 91°, \qquad \Delta K\approx -2^{-29.13},$$
$$p=28: \quad \{S(i)\}=\{1\,1\,2\,3\,3\,4\,5\,5\,6\,6\,7\,8\,8\,9\,10\,11\,12\,13\,14\,14\,15 \cdots 27\,28\},$$
$$\{\eta(j)T(j)\}=\{0\,-2\,+6\}, \qquad L=34+2, \qquad C\approx 91°, \qquad \Delta K\approx 2^{-32.53},$$
$$p=32: \quad \{S(i)\}=\{0\,0\,1\,3\,3\,3\,4\,5\,6\,7\,8\,9\,9\,10 \cdots 31\,32\},$$
$$\{\eta(j)T(j)\}=\{1\,-3\,-8\,+16\,-25\,-27\},$$
$$L=36+5, \qquad C\approx 145°, \qquad \Delta K\approx -2^{-39.93}.$$

Remember that in order to meet the convergence condition (23) and to provide a convergence region $C \geqq 90°$, the minimum number of CORDIC iterations is $p+1$. So, for all CORDIC sequences given above, the number $L-(p+1)$ of additional iterations for scaling corrrection is $p/4$. Moreover, except for the first CORDIC sequence, the remaining scaling error $|\Delta K|$ is significantly smaller than 2^{-p}. This leads to improved numerical properties compared with other CORDIC sequences reported in the literature. We also remark that if the symmetric eigenvalue problem is considered for which a convergence region of $C \geqq 45°$ is sufficient [2], the total number of CORDIC iterations L can be further reduced. An example that is nearly identical to the last CORDIC sequence given above is

$$p=32: \quad \{S(i)\}=\{1\,3\,3\,3\,4\,5\,6\,7\,8\,9\,9\,10 \cdots 31\,32\},$$
$$\{\eta(j)T(j)\}=\{0\,-3\,-8\,+16\,-25\,-27\},$$
$$L=34+5, \qquad C\approx 55°, \qquad \Delta K\approx -2^{-39.93}.$$

For comparison, Delosme [5] has also given an optimized CORDIC sequence for the same situation. His sequence requires one iteration more ($L=40$) and achieves a scaling accuracy of $\Delta K \approx 2^{-33.16}$.

We suspect that similar results can also be obtained by using Deprettere's double-shift concept. However, this method requires a slightly increased hardware complexity and will not be discussed in this paper.

5. CORDIC implementation of the SVD PEs. For easy illustration, we first introduce a CORDIC processor symbol as shown in Fig. 2. The descriptions inside the box determine uniquely the function mode of the CORDIC processor according to Table 1. The output data x and y are assumed to be scaling corrected.

It is now simple to map the operations (8)–(11) and (12)–(15) of the TPR method onto a two CORDIC processor architecture. In Fig. 3, the diagonal PEs of the SVD array are implemented by two CORDIC processors and eight adders. The dotted inputs of the adders represent negated inputs. Because the diagonal PEs work in the generation mode,

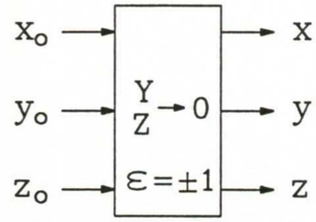


FIG. 2. CORDIC *symbol*.

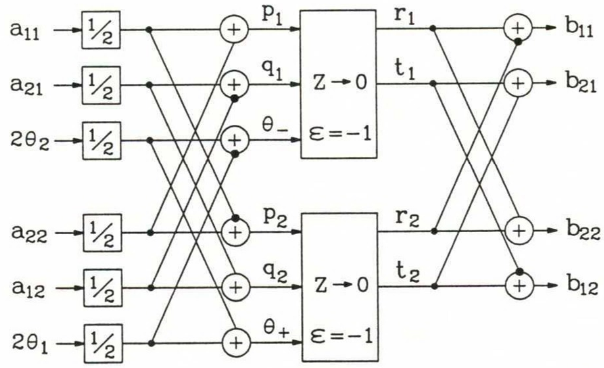


FIG. 3. CORDIC *implementation of the diagonal* PE *of the* SVD *array*.

both CORDIC processors are driven in the "$y \to 0$" mode for computing Cartesian-to-polar coordinates conversions. In Fig. 4, the off-diagonal PEs working in the rotation mode are implemented by two CORDIC processors and ten adders. Here, the CORDIC processors are driven in the "$z \to 0$" mode for performing plane rotations.

Obviously, both CORDIC implementations have nearly the same architecture. All PEs of the SVD array can thus be implemented by one unified CORDIC SVD module (Fig. 5) without considerably increased hardware cost. The different computation modes of the diagonal and off-diagonal PEs are easily "programmed" by one control bit. The resulting SVD array is similar to that in Fig. 1, but homogeneous with identical PEs.

We remark that Fig. 5 is more a "graphic program" describing the sequence of operations to be computed rather than a hardware block diagram. We show in the following that the 12 adders that are paired into three pre-butterflys and three post-butterflys can be integrated into the two CORDIC processors without separate hardware realizations. The Jacobi SVD method is a recursive method. Each PE of the SVD array has to exchange data with its diagonal neighbors. Because of this data dependency, only recursive CORDIC processors can be used here. This is an arithmetic unit consisting of mainly three adders and two barrel-shifters. It carries out the L iterations of the CORDIC algorithm in L cycles by using data feedback. The two CORDIC processors contained in one CORDIC SVD module require six adders altogether. So, it is natural to modify the CORDIC processor architecture slightly and to use the existing six adders for computing both the pre-butterfly and the post-butterfly operations. The resulting CORDIC SVD module has

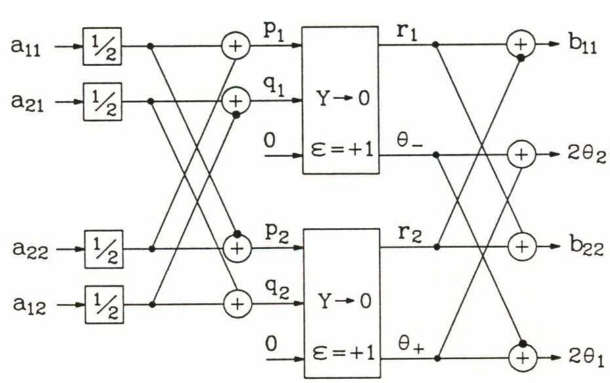


FIG. 4. CORDIC *implementation of the off-diagonal* PE *of the* SVD *array*.

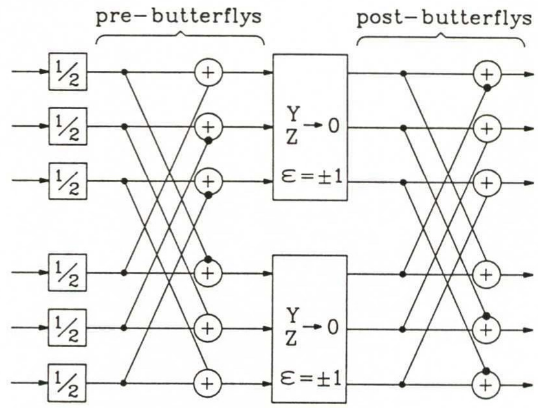


FIG. 5. *A unified* CORDIC SVD *module for implementing all* PEs *of the* SVD *array.*

the hardware complexity of two recursive CORDIC processors and requires a total computation time of $L + 2$ iterations.

In Fig. 6, the principal architecture of such a two CORDIC processor SVD module is shown. The dashed lines and boxes represent the additional hardware components

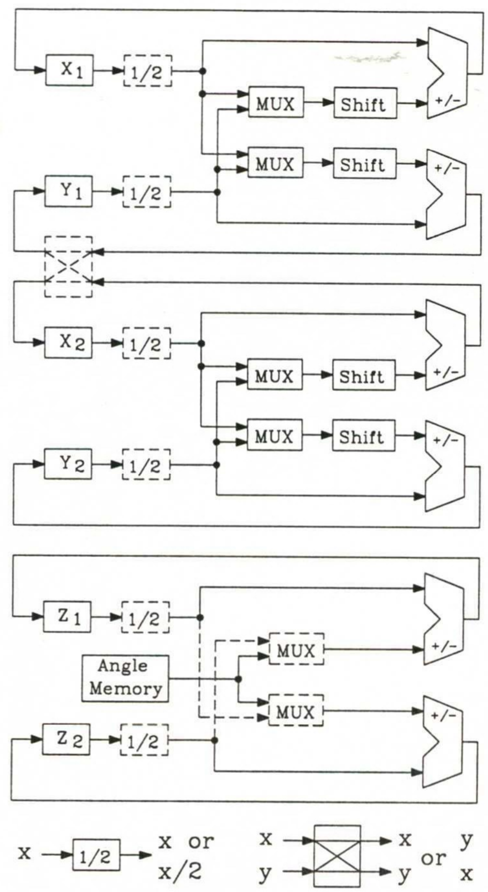


FIG. 6. *The principal architecture of the unified* CORDIC SVD *module.*

enabling the CORDIC processors to compute the butterfly operations. It is easily verified that the upper four adders devoted to x and y can perform the following types of operations $2^{-1}x \pm 2^{-1}y$ (pre-butterfly), $x \pm y$ (post-butterfly), $x \pm 2^{-s}y$ (CORDIC iteration) and $x \pm 2^{-s}x$ (scaling iteration) while the lower two adders devoted to z can compute $2^{-1}z_1 \pm 2^{-1}z_2$ (pre-butterfly), $z_1 \pm z_2$ (post-butterfly), and $z \pm \alpha$ (CORDIC iteration), respectively. The cross switch between the registers "Y_1" and "X_2" is needed to exchange data when the CORDIC SVD module switches from the pre-butterfly operations into the CORDIC iterations or from the CORDIC iterations into the post-butterfly operations, respectively. Then, we see from Fig. 3 that the output data pairs of the pre-butterflys are (p_1, p_2) and (q_1, q_2), while the desired input data pairs for the CORDIC iterations are (p_1, q_1) and (p_2, q_2), respectively. So, p_2 and q_1 have to be exchanged.

6. Comparisons. We now compare the new CORDIC SVD module with those proposed by Cavallaro and Luk [4] and Delosme [5]. Let A_{csvd} and T_{csvd} denote the area and time complexity of a CORDIC SVD module and A_{cordic} and T_{cordic} those of a CORDIC processor, respectively. Cavallaro and Luk have shown that their most efficient parallel diagonalization method requires $A_{\text{csvd}} \approx 2A_{\text{cordic}}$ and $T_{\text{csvd}} \approx 3T_{\text{cordic}}$ for the diagonal PEs and $A_{\text{csvd}} \approx 2A_{\text{cordic}}$ and $T_{\text{csvd}} \approx 2T_{\text{cordic}}$ for the off-diagonal PEs. By using the TPR method, we require $T_{\text{csvd}} \approx 2A_{\text{cordic}}$ and $T_{\text{csvd}} \approx T_{\text{cordic}}$ for all PEs. In other words, having approximately the same hardware complexity, the computation time is reduced by more than 50 percent.

A comparison to Delosme's method is more difficult because he follows a quite different approach. Therefore, only rough performance estimates are given here. In our method, we compute the rotation angles explicitly. After these computations have been completed in the diagonal PEs, the angles propagate to the off-diagonal ones. We assume that the propagation from one PE to its neighbors takes one cycle T_{cycle}, the time required for computing one CORDIC iteration. This implies local communications without broadcasting data. At the beginning of the second propagation cycle, the angles reach the diagonal neighbors of the diagonal PEs which complete their computations after T_{csvd}. This means that the diagonal PEs have to wait for a delay time $T_{\text{delay}} = T_{\text{cycle}} + T_{\text{csvd}}$ before they can exchange data with their diagonal neighbors.[3] The total time elapsed between two adjacent activities at each PE is thus $T_{\text{csvd}} + T_{\text{delay}} = 2T_{\text{csvd}} + T_{\text{cycle}} \approx 2T_{\text{cordic}}$ because T_{cycle} is negligible with respect to $T_{\text{cordic}} = L \cdot T_{\text{cycle}}$.

Delosme does not compute the rotation angles explicitly. He rather calculates encodings of the angles, i.e., sequences of signs ± 1, and sends them to the off-diagonal PEs. This enables overlap of diagonal and off-diagonal rotations because the encoding signs are recursively obtained and become available before the completion of diagonal operations. Accordingly, no delay time is required ($T_{\text{delay}} = 0$), provided that the SVD array size (the half of the matrix size) is smaller than the number of CORDIC scaling iterations n_k (for details, see [5]). The drawback is, however, that the TPR method cannot be applied to the off-diagonal PEs. Four plane rotations are hence required, resulting in $T_{\text{cvsd}} = 2T_{\text{cordic}}$ for two CORDIC processors in one module. In other words, the time complexities $T_{\text{csvd}} + T_{\text{delay}}$ of both methods are nearly identical and equal $2T_{\text{cordic}}$. If, however, multiple problems are interleaved, the fraction of idle time that is 50 percent in our case can be reduced to almost zero. In such a situation, our method provides the double speed compared with Delosme's one.

[3] If the propagation time is assumed to be T_{csvd}, we get the well-known result $T_{\text{delay}} = 2T_{\text{csvd}}$ given by Brent and Luk [2].

In terms of area complexity, both CORDIC SVD modules contain two CORDIC processors. Our module consists of essentially six adders, four barrel shifters, and one ROM table containing n angle values. Delosme's architecture requires four carry-save adders, four adders, and eight barrel shifters. So, as a rough estimate, both SVD modules have the same order of area complexities.

Perhaps the most important advantage of Delosme's approach is the 2-bit wide horizontal and vertical data connections for sending angle encodings serially rather than sending the full angle values in parallel. The prices are the upper bound of the SVD array size depending on the number of CORDIC scaling iterations, the relatively complicated timing, and a nonregular CORDIC architecture design. We also mention that while Delosme's method presumes a CORDIC implementation, the TPR method is applicable to other computing architectures.

7. Other applications of the TPR method. Another advantage of the TPR method seems to be the relatively wide range of applications. We indicate some of them in the following.

For the SVD of a rectangular matrix, a well-known method is first to triangularize the matrix by QR decomposition and then to apply the Jacobi SVD procedure to the triangular factor. Luk [22] has shown that both steps can be implemented by one triangular systolic array. Each PE contains a 2×2 submatrix. It applies two plane rotations (through the same angle) to the two column vectors at the QR step and a two-sided rotation at the SVD step. For computing the SVD step, the PE can be realized by the CORDIC SVD module, as before. On the other side, the two CORDIC processors contained in the module are also appropriate to perform the two-plane rotations of the QR step. The CORDIC SVD module presented in this paper thus provides a suitable PE for Luk's triangular SVD array.

Stewart [23] has proposed a square systolic array for computing the Schur decomposition (SD) of a non-Hermitian matrix which, for example, is useful for evaluating the eigenvalues of the matrix. His approach is similar to the Jacobi SVD method. It is based on applying a sequence of two-sided rotations to 2×2 submatrices, where the left and right rotation angles are identical to make the diagonal submatrices upper triangular. While the diagonal PEs perform operations different from those in SVD, the off-diagonal PEs have exactly the same computational task as in SVD computing. Therefore, the CORDIC SVD module can also be used in Stewart's SD array.

Even in sequential computations on a uniprocessor system, one can still apply the TPR method to reduce the computational complexity of two-sided rotations.

8. Conclusion. We have investigated a novel algorithm for computing two-sided rotations requiring only two plane rotations and a few additions. This results in significantly reduced computations of various SVD and SD methods. For parallel implementations, we have presented a unified CORDIC SVD module for implementing all PEs of the SVD array given by Brent and Luk. This leads to a homogeneous array architecture that is simpler in hardware and offers twice the computational speed of that of Cavallaro and Luk. Moreover, we have pointed out that the same CORDIC SVD module can be efficiently used in other array architectures, such as Luk's triangular SVD array and Stewart's SD array.

Acknowledgments. The authors thank the referees and the editor for their valuable comments on this paper.

REFERENCES

[1] J. M. Speiser, *Signal processing computational needs*, in Proc. SPIE Advanced Algorithms and Architectures for Signal Processing I, Vol. 696, Society of Photo-optical Instrumentation Engineers, 1986, pp. 2–6.

[2] R. P. Brent and F. T. Luk, *The solution of singular value and symmetric eigenvalue problems on multiprocessor arrays*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 69–84.

[3] R. P. Brent, F. T. Luk, and C. F. Van Loan, *Computation of the singular value decomposition using mesh-connected processors*, J. VLSI Comput. Syst., 1 (1985), pp. 242–270.

[4] J. R. Cavallaro and F. T. Luk, *Architectures for a* CORDIC SVD *processor*, Proc. SPIE Real-Time Signal Processing IX, Vol. 698, Society of Photo-optical Instrumentation Engineers, 1986, pp. 45–53.

[5] J. M. Delosme, *A processor for two-dimensional symmetric eigenvalue and singular value arrays*, in Proc. 21st Asilomar Conference on Circuits, Systems and Computers, November 1987.

[6] ——, CORDIC *algorithms: Theory and extensions*, Proc. SPIE Advanced Algorithms and Architectures for Signal Processing IV, Vol. 1152, Society of Photo-optical Instrumentation Engineers, August 1989, pp. 131–145.

[7] C. G. J. Jacobi, *Über ein leichtes Verfahren die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen*, J. Reine Angew. Math., 30 (1846), pp. 51–95.

[8] E. G. Kogbetliantz, *Solution of linear equations by diagonalization of coefficient matrices*, Quart. Appl. Math., 13 (1955), pp. 123–132.

[9] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc., 94 (1960), pp. 1–23.

[10] A. H. Sameh, *Solving the linear least squares problem on a linear array of processors*, in Algorithmically Specialized Parallel Computers, L. Synder et al., eds., Academic Press, New York, 1985, pp. 191–200.

[11] U. Schwiegelshohn and L. Thiele, *A systolic algorithm for cyclic-by-rows* SVD, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 1987, pp. 768–770.

[12] F. T. Luk and H. Park, *On the equivalence and convergence of parallel Jacobi* SVD *algorithms*, in Proc. SPIE Advanced Algorithms and Architectures for Signal Processing II, Vol. 826, Society of Photo-optical Instrumentation Engineers, 1987, pp. 152–159.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983, p. 44.

[14] J. E. Volder, *The* CORDIC *trigonometric computing technique*, IRE Trans. Electronic Comput., 8 (1959), pp. 330–334.

[15] J. S. Walther, *A unified algorithm for elementary functions*, in Proc. Spring Joint Computer Conference, AFIPS Press, New Jersey, 1971, pp. 379–385.

[16] J. R. Cavallaro and F. T. Luk, CORDIC *arithmetic for an* SVD *processor*, J. Parallel Distributed Comput., 5 (1988), pp. 271–290.

[17] G. L. Haviland and A. A. Tuszynski, *A* CORDIC *arithmetic processor chip*, IEEE Trans. Comput., 29 (1980), pp. 68–79.

[18] H. M. Ahmed, *Signal processing algorithms and architectures*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, Stanford, CA, December 1981.

[19] J. M. Delosme, VLSI *implementation of rotations in pseudo-Euclidean spaces*, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Boston, 1983, pp. 927–930.

[20] E. F. Deprettere, P. Dewilde, and R. Udo, *Pipelined* CORDIC *architectures for fast* VLSI *filtering and array processing*, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, San Diego, CA, March 1984, pp. 41A.6.1–41A.6.4.

[21] D. König and J. F. Böhme, *Optimizing the* CORDIC *algorithm for processors with pipeline architecture*, in Proc. EUSIPCO, North-Holland, Elsevier Science Publishers, Amsterdam, 1990, pp. 1391–1394.

[22] F. T. Luk, *A triangular processor array for computing singular values*, Linear Algebra Appl., 77 (1986), pp. 259–273.

[23] G. W. Stewart, *A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 853–864.

YA02

Search Request: Journal Title = siam journal
Search Results: Displaying 3 of 35 entries

SIAM journal on matrix analysis and applications : a publicatio of the...

000 02118mas a2200493 a 450
001 705634
005 20121114190113.0
008 870909c19889999pauqx1p |||| 0 a0eng d
010 __ **|a** sn 87002493
022 0_ **|a** 0895-4798
030 __ **|a** SJMAEL
032 __ **|a** 530050 **|b** USPS
035 __ **|a** (OCoLC)16655630
035 __ **|9** ADP8392C1
035 __ **|a** 705634
040 __ **|a** NSD **|c** NSD **|d** PIT **|d** WaOLN
042 __ **|a** nsdp **|a** lcd
050 __ **|a** QA263 **|b** .S45
210 0_ **|a** SIAM j. matrix anal. appl.
212 0_ **|a** Society for Industrial and Applied Mathematics journal on matrix analysis and applications
222 _0 **|a** SIAM journal on matrix analysis and applications
245 00 **|a** SIAM journal on matrix analysis and applications : **|b** a publicatio of the Society for Industrial and Applied Mathematics.
246 10 **|a** Journal on matrix analysis and applications
246 10 **|a** Matrix analysis and applications
260 __ **|a** [Philadelphia, Pa. : **|b** The Society, **|c** c1988-
265 __ **|a** SIAM Publications, 117 S. 17th St., Philadelphia, PA 19103-5052
300 __ **|a** v.
310 __ **|a** Four no. a year
362 0_ **|a** Vol. 9, no. 1 (Jan. 1988)-
510 1_ **|a** Mathematical reviews **|x** 0025-5629 **|b** 1988-
510 2_ **|a** Electronics and communications abstracts journal (Riverdale) **|x** 0361-3313 **|b** 1988-
510 2_ **|a** ISMEC bulletin **|x** 0306-0039 **|b** 1988-

510 2_ |**a** Pollution abstracts with indexes |**x** 0032-3624 |**b** 1988-

510 2_ |**a** Safety science abstracts journal |**x** 0160-1342 |**b** 1988-

510 2_ |**a** Computer & control abstracts |**x** 0036-8113 |**b** March 1981-

510 2_ |**a** Electrical & electronics abstracts |**x** 0036-8105 |**b** March 1981-

510 2_ |**a** Physics abstracts. Science abstracts. Series A |**x** 0036-8091 |**b** March 1981-

530 __ |**a** Also available on the World Wide Web.

650 _0 |**a** Matrices |**v** Periodicals.

650 _0 |**a** Mathematics |**v** Periodicals.

710 2_ |**a** Society for Industrial and Applied Mathematics.

780 00 |**t** SIAM journal on algebraic and discrete methods |**x** 0196-5212 |**w** (OCoLC) 5842627

856 7_ |**u** http://pitt.idm.oclc.org/login?url=http://epubs.siam.org:80/sam-bin/dbq/toclist/SIMAX |**z** Access for Pitt-affiliated users only

948 __ |**a** per:st

Print/Save/Email			
Select Download Format	Full Record ▼	Print or Save	Save Search to My Account
Enter your email address:		Email	
Save results for later:	Save to Bookbag		