Journal of Mathematics and Statistics 1 (2): 124-128, 2005
ISSN 1549-3644

Computing ULV Decomposition by Divide Conquer Method

[1]Hasan Erbay and [2]Jesse L. Barlow
[1]Department of Mathematics, Kırıkkale University, Kırıkkale, Turkey
[2]Department of Computer Science and Engineering
Pennsylvania State University, State College, PA USA

Abstract: The ULV decomposition (ULVD) is an important member of a class of rank-revealing two-sided orthogonal decompositions used to approximate the Singular Value Decomposition (SVD). The ULVD can be modified much faster than the SVD.
The accurate computation of the subspaces is required in applications in signal processing. In this study we introduce a divide conquer ULVD algorithm.

Key words: Two-sided Orthogonal Decompositions, ULV Decomposition, Subspace Estimation

INTRODUCTION

The SVD has been the main tool for extracting matrix information such as rank, subspaces, noise space of a given matrix A. However it is computationally expensive and difficult to modify [9]. Thus alternative decompositions have been considered which are nearly as reliable as the SVD but computationally inexpensive and easier to modify. Stewart [13] proposed rank-revealing ULV decomposition as an alternative. ULVD is guaranteed to reveal the numerical rank correctly.
Throughout the study $\|\cdot\|$ will denote the two norm and $\|\cdot\|_F$ will denote the Frobenius norm.
The ULVD is a special case of the two-sided orthogonal decompositions defined by Faddeev, Kublanovskaya and Faddeeva [5] and Hanson and Lawson [10]. The most familiar formulation is due to Stewart [13]. A slightly different formulation is given below by Barlow, Yoon and Zha [2].
Let $A \in \Re^{m\times n}$ with $m \geq n$ have the factorization

$$A = U\begin{pmatrix} C \\ 0 \end{pmatrix} V^T \tag{1.1}$$

where

$$U \in \Re^{m\times m},\ V \in \Re^{n\times n} \text{ orthogonal} \tag{1.2}$$

$$C = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} L & 0 \\ F & G \end{pmatrix} \end{array}, \ L, G \text{ lower triangular,} \tag{1.3}$$

$$\left\|L^{-1}\right\|^{-1} \geq \varepsilon,\ \left\|(F\ G)\right\| \leq \left(\Phi(n)\right)^2 \varepsilon \tag{1.4}$$

where ε is the tolerance or "noise level" and $\Phi(n)$ is a modestly growing function of n, typically n or $\frac{1}{2}\lceil \log n \rceil + 1$. The value k is usually referred to as ε-rank or the rank "revealed" by the ULVD. It is not numerical rank unless ε is very close to machine precision times the norm of A and, in practice, ε will often be larger than that value.
In this study, we let $\sigma_i(A)$ denote the i-th singular value of A.
The matrix V yields good approximations of two important right singular subspaces. Let W be the right singular vector matrix of A and let W and V be partitioned into

$$W = \begin{pmatrix} \overset{k}{W_1} & \overset{n-k}{W_2} \end{pmatrix},\ V = \begin{pmatrix} \overset{k}{V_1} & \overset{n-k}{V_2} \end{pmatrix} \tag{1.5}$$

In applications to least squares and total least squares problems, it is important that range(V_1) be "close" to range(W_1) or equivalently, that range(V_2) be "close" to range(W_2). To insure that, Fierro and Bunch [6] show that Davis-Kahan [4] sin θ measure of error in subspaces is bounded by

$$\left|\sin\theta\right| = \left\|W_1^T V_2\right\| \leq \frac{\left\|F^T G\right\|}{\text{gap}} \tag{1.6}$$

where

$$\text{gap} = \max\left\{\sigma_k^2\left(\begin{pmatrix} L \\ F \end{pmatrix}\right) - \sigma_{k+1}^2(A),\ \sigma_k^2(A) - \|G\|^2\right\}.$$

With current procedures, computing the ULVD from scratch saves a little time from computing the SVD. These ULVD algorithms start with an initial skinny QR factorization of A followed by steps involving condition estimation, deflation and refinement.

We introduce a divide conquer ULVD algorithm of cost $4m^2n + 8mn^2 - 4/3n^3 + 5n^2 + 16n^{3/2} + O(n)$ The rest of the study is organized as follows. In the next section we briefly outline the steps of the divide conquer ULVD algorithm. This section also includes some theoretical results of the algorithm. Finally, in §3 we present some numerical results obtained from using our algorithm.

DIVIDE CONQUER ULVD ALGORITHM

For the sake of simplicity we consider A to be an order n matrix with $n = 2^{n_0}$ for some positive integer n_0.

Algorithm 2.1: (Divide Conquer ULVD Algorithm)

Step 1: Construct orthogonal matrices $U^{(0)}$, $V^{(0)} \in \Re^{n\times n}$ such that

$$U^{(0)^T} A V^{(0)} = \begin{pmatrix} \alpha_1 & & & \\ \phi_1 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \phi_{n-1} & \alpha_n \end{pmatrix}.$$

Step 2: Partition the matrix B as

$$B = \begin{array}{c} \\ r \\ 1 \\ n-r-1 \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{pmatrix} B_1 & 0 \\ \phi_{r+1} e_1^T & \alpha_{r+1} e_1^T \\ 0 & B_2 \end{pmatrix} \end{array}$$

Step 3: If $\left(r > n^{1/2}\right)$ then

take $B = B_1$ and go to step1

else compute the SVD of B_1 to obtain

$$B_1 = U^{(1)} \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{12} \end{pmatrix} V^{(1)^T}$$

such that $\varepsilon \left\| \Sigma_{11}^{-1} \right\| < 1$, $\left\| \Sigma_{12} \right\| < \varepsilon$

end if

Define $L_1 = \Sigma_{11}$, $G_1 = \Sigma_{12}$, $F_1 = 0$ and go to step 5.

If $\left(n - r > n^{1/2}\right)$ then

take $B = B_2$ and go to step1

else compute the SVD of B_2 to obtain

$$B_2 = U^{(2)} \begin{pmatrix} \Sigma_{21} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} V^{(2)^T}$$

such that $\varepsilon \left\| \Sigma_{21}^{-1} \right\| < 1$, $\left\| \Sigma_{22} \right\| < \varepsilon$

end if

Define $L_2 = \Sigma_{21}$, $G_2 = \Sigma_{22}$, $F_2 = 0$ and go to step 5.

Step 4: Construct orthogonal matrices $U^{(1)}$, $V^{(1)} \in \Re^{r\times r}$ and $U^{(2)}$, $V^{(2)} \in \Re^{n-r-1\times n-r-1}$ such that

$$B_1 = U^{(1)} \begin{pmatrix} L_1 & 0 \\ F_1 & G_1 \end{pmatrix} V^{(1)^T}$$

$$B_2 = U^{(2)} \begin{pmatrix} L_2 & 0 \\ F_2 & G_2 \end{pmatrix} V^{(2)^T}$$

with the conditions $\varepsilon \left\| L_1^{-1} \right\| < 1$, $\varepsilon \left\| L_2^{-1} \right\| < 1$.

Step 5: Merge the matrices B_1 and B_2 to get

$$\begin{pmatrix} U^{(1)^T} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U^{(2)^T} \end{pmatrix} B \begin{pmatrix} V^{(1)^T} & & \\ & & \\ & & V^{(2)^T} \end{pmatrix} = \begin{pmatrix} L_1 & 0 & 0 & 0 \\ F_1 & G_1 & 0 & 0 \\ z_1^T & z_2^T & z_3^T & z_4^T \\ 0 & 0 & L_2 & 0 \\ 0 & 0 & F_2 & G_2 \end{pmatrix} \quad (2.1)$$

Step 6: Reorder the matrix on the right side of (2.1) to obtain

$$\begin{pmatrix} U^{(3)^T} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U^{(4)^T} \end{pmatrix} B \begin{pmatrix} V^{(3)^T} & & \\ & & \\ & & V^{(4)^T} \end{pmatrix} = \begin{pmatrix} L_1 & 0 & 0 & 0 \\ 0_1 & L_2 & 0 & 0 \\ z_1^T & z_3^T & z_2^T & z_4^T \\ F_1 & 0 & G_1 & 0 \\ 0 & F_2 & 0 & G_2 \end{pmatrix}$$

Step 7: Construct orthogonal matrices $U^{(5)}$, $V^{(5)} \in \Re^{n\times n}$ such that

$$\begin{pmatrix} \hat{L} & 0 \\ \hat{z}^T & \gamma e_1^T \\ \hat{F} & \hat{G} \end{pmatrix} = U^{(5)^T} \begin{pmatrix} L_1 & 0 & 0 & 0 \\ 0_1 & L_2 & 0 & 0 \\ z_1^T & z_3^T & z_2^T & z_4^T \\ F_1 & 0 & G_1 & 0 \\ 0 & F_2 & 0 & G_2 \end{pmatrix} V^{(5)}$$

where

$$\hat{L} = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \quad (2.2)$$

$$\hat{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \ \gamma = \left\| \begin{pmatrix} z_2 \\ z_4 \end{pmatrix} \right\|, \quad (2.3)$$

$$\begin{pmatrix} \hat{F} & \hat{G} \end{pmatrix} = \begin{pmatrix} F_1 & 0 & G_1 & 0 \\ 0 & F_2 & 0 & G_2 \end{pmatrix} V^{(5)}. \qquad (2.4)$$

Step 8: Construct orthogonal matrices $U^{(6)}$, $V^{(6)} \in \Re^{n\times n}$ such that

$$\begin{pmatrix} L & 0 \\ F & G \end{pmatrix} = U^{(6)^T} \begin{pmatrix} \hat{L} & 0 \\ \hat{z}^T & \gamma e_1^T \\ \hat{F} & \hat{G} \end{pmatrix} V^{(6)}$$

with $\varepsilon \left\| L^{-1} \right\| < 1$

Using the definition of a flop as an addition and a multiplication the complexity of the algorithm (2.1) is $4m^2n + 8mn^2 - 4/3n^3 + 5n^2 + 16n^{3/2} + O(n)$ We will show this using Proposition 2.2 in Proposition 2.3.
The following proposition gives the number of levels in which the problem is divided into.

Proposition 2.2: Assume the terminology of Algorithm 2.1. Then the number of levels l is given by

$$l = \frac{1}{2}\lfloor \log_2 n \rfloor \qquad (2.5)$$

In the following proposition we prove the complexity of Algorithm 2.1.

Proposition 2.3: Let a matrix A of size $m\times n$ be applied to the algorithm (2.1). Then it takes $4m^2n + 8mn^2 - 4/3n^3 + 5n^2 + 16n^{3/2} + O(n)$ to obtain ULVD of A.

Proof: Assume l is the number of levels when A is applied to the algorithm (2.1). Then the cost of all SVD's in the Step 3 of the algorithm is given by

$$2^l \times \begin{pmatrix} \text{the cost of each SVD} \\ \text{of a matrix of size } \le \dfrac{n}{2^l} \end{pmatrix} \qquad (2.6)$$

Moreover, at level i for $i \le l - 1$, there are 2^i updates of a matrix of size $\le n/2^i$ Thus the updates, due to Step 8 adds

$$\sum_{i=0}^{l-1} 2^i \times \begin{pmatrix} \text{the cost of update of} \\ \text{a matrix of size } \le \dfrac{n}{2^i} \end{pmatrix} \qquad (2.7)$$

where for a matrix of size n_0 having rank k_0, each update costs $\frac{5}{2}\left[n_0^2 - n_0(2k_0 - 1) + \left(k_0^2 - k_0\right)\right]$. Taking $l = \frac{1}{2}\lfloor \log_2 n \rfloor$ and adding the cost of bidiagonalization step to (2.6) suffice to complete the proof.

Lemma 2.4: Let

$$\begin{pmatrix} F & G \end{pmatrix} = \begin{pmatrix} F_1 & 0 & G_1 & 0 \\ 0 & F_2 & 0 & G_2 \end{pmatrix} \qquad (2.8)$$

be a given matrix. Then we have

$$\left\| \begin{pmatrix} F & G \end{pmatrix} \right\| = \max\left\{ \left\| \begin{pmatrix} F_1 & G_1 \end{pmatrix} \right\|, \left\| \begin{pmatrix} F_2 & G_2 \end{pmatrix} \right\| \right\} \qquad (2.9)$$

$$\left\| \begin{pmatrix} F & G \end{pmatrix} \right\|_F^2 = \left\| \begin{pmatrix} F_1 & G_1 \end{pmatrix} \right\|_F^2 + \left\| \begin{pmatrix} F_2 & G_2 \end{pmatrix} \right\|_F^2 \qquad (2.10)$$

Theorem 2.5 tells us about the quality of the decomposition obtained by Algorithm 2.1.

Theorem 2.5: Let $A \in \Re^{m\times n}$, $m \ge n$. Then Algorithm 2.1 produces a decomposition of A such that

$$A = U \begin{pmatrix} L & 0 \\ F & G \end{pmatrix} V^T \qquad (2.11)$$

where

$$U \in \Re^{m\times m}, V \in \Re^{n\times n} \text{ orthogonal}, \qquad (2.12)$$

$$\varepsilon \| L^{-1} \| < 1 \qquad (2.13)$$

$$\begin{aligned} &\left\| \begin{pmatrix} F & G \end{pmatrix} \right\|_F \le \Phi_F(n)^{1/2} \varepsilon, \\ &\Phi_F(n) = n, \Phi_F\left(\left\lceil n^{1/2} \right\rceil \right) = n^{1/2} \end{aligned} \qquad (2.14)$$

$$\begin{aligned} &\left\| \begin{pmatrix} F & G \end{pmatrix} \right\| \le \Phi(n)^{1/2} \varepsilon, \\ &\Phi(n) = \tfrac{1}{2}\lceil \log_2 n \rceil + 1, \Phi\left(\left\lceil n^{1/2} \right\rceil \right) = 1 \end{aligned} \qquad (2.15)$$

Proof: Step 8 of Algorithm 2.1 constructs L so that $\varepsilon \| L^{-1} \| < 1$, thus (2.13) is obvious. In doing so it reduces

$$\begin{pmatrix} \hat{L} & 0 \\ \hat{z}^T & \gamma e_1^T \\ \hat{F} & \hat{G} \end{pmatrix} \to \begin{pmatrix} L & 0 \\ F & G \end{pmatrix}.$$

By Lemma 2.4 and orthogonal invariance of two norm and Frobenius norm we have

$$\left\| \begin{pmatrix} \hat{F} & \hat{G} \end{pmatrix} \right\| = \max\left\{ \left\| \begin{pmatrix} F_1 & G_1 \end{pmatrix} \right\|, \left\| \begin{pmatrix} F_2 & G_2 \end{pmatrix} \right\| \right\} \qquad (2.16)$$

$$\left\| \begin{pmatrix} \hat{F} & \hat{G} \end{pmatrix} \right\|_F^2 = \left\| \begin{pmatrix} F_1 & G_1 \end{pmatrix} \right\|_F^2 + \left\| \begin{pmatrix} F_2 & G_2 \end{pmatrix} \right\|_F^2 \qquad (2.17)$$

We observe that since $\varepsilon \| L^{-1} \| < 1$ then

$$\begin{pmatrix} \hat{L} & 0 \\ z^T & \gamma \end{pmatrix} \tag{2.18}$$

has either k or $k+1$ singular values greater than ε. We first consider the case where (2.18) has $k+1$ singular values greater than ε. Then

$$\|(F \quad G)\| = \left\|\left(\hat{F} \quad \hat{G}\right)\right\| , \ \|(F \quad G)\|_F = \left\|\left(\hat{F} \quad \hat{G}\right)\right\|_F .$$

Thus, by (2.16) and (2.17),

$$\Phi(n) = \Phi\left(\left\lceil \tfrac{n}{2} \right\rceil\right) , \ \Phi_F(n) = 2\Phi_F\left(\left\lceil \tfrac{n}{2} \right\rceil\right). \tag{2.19}$$

We second consider that (2.18) has k singular values greater than ε. Let ω be the smallest singular value of (2.18). Then

$$\|(F \quad G)\|^2 = \left\|\left(\hat{F} \quad \hat{G}\right)\right\|^2 + \omega^2 \le \left(\Phi\left(\left\lceil \tfrac{n}{2} \right\rceil\right)+1\right)\varepsilon^2 \tag{2.20}$$

$$\|(F \quad G)\|_F^2 = \left\|\left(\hat{F} \quad \hat{G}\right)\right\|_F^2 + \omega^2 \le \left(2\Phi_F\left(\left\lceil \tfrac{n}{2} \right\rceil\right)+1\right)\varepsilon^2 \tag{2.21}$$

Moreover Step 3 gives us that

$$B = U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} V^T$$

where $\|\Sigma_2\| \le \varepsilon$, $\|\Sigma_2\|_F \le n^{1/4}\varepsilon$. Hence,

$$\Phi(r) = 1 , \ \Phi_F(r) = n^{1/2} \tag{2.22}$$

$$\text{for } \tfrac{1}{2} n^{1/2} \le r \le n^{1/2}.$$

So far we presented the divide conquer ULVD algorithm and related theoretical results. However, it might be necessary to improve the quality of the computed subspaces. It follows the bound in (1.6) that this can be accomplished by reducing the norm of off diagonal matrix.

Numerical Examples: In this section we analyze the performance of our algorithm. The results were obtained using Matlab on a Ultra5 SUN station in IEEE Standard double precision with machine epsilon $\approx 10^{-16}$.

$$\text{Let } \ A = \tilde{U}\Sigma W^T \ , \ W = \left(\overset{k}{W_1} \quad \overset{n-k}{W_2}\right)$$

be the SVD of A of numerical rank k obtained by the Matlab's SVD algorithm. Also, let

$$A = \breve{U}\breve{C}\breve{V}^T \ , \ \breve{V} = \left(\overset{k}{\breve{V}_1} \quad \overset{n-k}{\breve{V}_2}\right)$$

be ULVD of A computed by traditional way. And ULVD of A by our divide conquer algorithm is given by

$$A = UCV^T \ , \ V = \left(\overset{k}{V_1} \quad \overset{n-k}{V_2}\right).$$

Define the angles between subspaces by

$$\sin\theta_1 = \left\|W_1^T V_2\right\| \ , \ \sin\theta_2 = \left\|W_1^T \breve{V}_2\right\| \tag{3.1}$$

The angles $\sin\theta_1$ and $\sin\theta_2$ represent error between the noise subspaces from Matlab's SVD and the one obtained by traditional way and by divide conquer algorithm, respectively.

Table 1: Results from First Set of Matrices

	Divide conquer algorithm			Traditional ULVD algorithm		
n	Dec. Err.	$error_1$	$\sin\theta_1$	Dec. Err.	$error_2$	$\sin\theta_2$
Results without any refinement						
50	1.6806e-14	1.3768e-09	2.2758e-15	1.9924e-14	6.5721e-10	2.0505e-15
100	4.1523e-14	2.3551e-09	3.3393e-15	6.3926e-14	1.2770e-09	4.5402e-15
150	1.1059e-13	2.7901e-09	4.7793e-15	1.0100e-13	1.3591e-09	4.0696e-15
200	2.1651e-13	3.3787e-09	6.6035e-15	1.6343e-13	1.6223e-09	5.3578e-15
250	2.5751e-13	3.5982e-09	6.0715e-15	2.5230e-13	1.8572e-09	7.0443e-15
300	4.6100e-13	3.8086e-09	8.8898e-15	3.5068e-13	1.6105e-09	6.9187e-15
Results with refinement						
50	2.5412e-14	3.7223e-23	1.9307e-15	5.9529e-14	7.1138e-23	2.3624e-15
100	7.5532e-14	2.4815e-24	3.1304e-15	1.3512e-13	1.7371e-23	3.5428e-15
150	1.2885e-13	1.1746e-22	3.9587e-15	2.0399e-13	1.4972e-22	4.8671e-15
200	2.0675e-13	4.0615e-22	6.5683e-15	3.4168e-13	3.4080e-22	6.4560e-15
250	2.4713e-13	2.3161e-22	6.1558e-15	8.0126e-13	3.5403e-22	8.4351e-15
300	3.8083e-13	7.5819e-21	6.4523e-15	6.7611e-13	1.9789e-19	8.3802e-15

Table 2: Results from Second Set of Matrices

	Divide conquer algorithm			Traditional ULVD algorithm		
n	Dec. Err.	$error_1$	$\sin\theta_1$	Dec. Err.	$error_2$	$\sin\theta_2$
Results without any refinement						
50	2.8665e-15	2.4899e-09	6.2554e-12	2.9380e-15	8.0483e-16	4.5764e-05
100	1.4867e-15	7.5672e-09	3.2217e-11	4.7703e-15	2.8691e-14	2.6420e-04
150	2.3815e-15	4.9852e-09	1.4367e-07	6.5517e-15	4.9376e-14	5.2104e-04
200	2.0544e-15	5.9237e-09	1.7014e-10	8.2519e-15	1.1137e-13	8.1981e-04
250	2.6867e-15	5.3665e-09	4.0178e-10	9.2295e-15	1.5452e-13	1.1078e-03
300	1.6876e-15	4.1915e-09	9.7344e-10	8.5009e-15	1.6433e-13	1.3774e-03
Results with refinement						
50	2.9448e-15	0	6.2458e-12	3.7194e-15	8.2718e-25	8.6853e-12
100	1.6302e-15	1.6544e-24	3.2218e-11	7.8506e-15	1.6544e-24	1.8867e-10
150	2.6496e-15	8.2718e-25	1.7388e-09	8.3047e-15	5.7903e-24	1.3487e-10
200	2.7747e-15	8.2718e-24	1.7042e-10	8.3902e-15	1.6544e-24	3.5150e-10
250	3.4913e-15	3.3087e-24	4.0178e-10	1.4076e-14	9.0990e-24	3.6362e-10
300	1.8043e-15	0	9.7470e-10	1.5035e-14	9.0990e-24	3.3719e-10

Define also

$$error_1 = \left|\sigma_{k+1}(C) - \left\|\begin{pmatrix} F & G \end{pmatrix}\right\|\right|,$$
$$error_2 = \left|\sigma_{k+1}(\breve{C}) - \left\|\begin{pmatrix} \breve{F} & \breve{G} \end{pmatrix}\right\|\right|. \quad (3.2)$$

We tested our algorithm on two sets of matrices of orders 50, 100, 150, 200, 250 and 300. In the first set of matrices there is a sufficiently big gap between the singular values σ_k and σ_{k+1} where k is the numerical rank of the matrix. However, the singular values are clustered in the second set of matrices. Both Table 1 and Table 2 show the decomposition error and the quantities defined in (3.2) as well as the subspace angles as defined in (3.1).

Table 1 presents the results by the first set of matrices. Both of traditional ULVD algorithm and divide conquer algorithm decompose A almost the same accuracy.

Table 2 presents results by second set of examples. As seen our method performs better in this case.

CONCLUSION

We have introduced a divide conquer ULVD algorithm which is faster than all available SVD algorithms. Theoretical and numerical results show that our algorithm is promising.

REFERENCE

1. Barlow, J.L. and P.A. Yoon, 1997. Solving recursive TLS problems using rank-revaling ULV decomposition, In Van Huffel, S. Ed, Proc. Workshop on TLS and Errors-In-Variables, Philadelphia, PA, SIAM Publications, pp: 117-126.
2. Barlow, J.L., P.A. Yoon and H. Zha, 1996. An algorithm and a stability theory for downdating the ULV decomposition. BIT, 36: 15-40.
3. Barlow, J.L., H. Erbay and Z. Zhang, 1999. A modified gram-schmidt based downdating technique for the ULV decompositions with applications to recursive TLS problems. Adv. Sign. Proc. Alg., Arch. and Impl. IX, F.T. Luk, Editor, SPIE Proc., Bellingham, WA, 3807: 247-257.
4. Davis, C. and W.M. Kahan, 1970. The rotation of eigenvectors by a perturbation III, SIAM J. Num. Anal. 7: 1-46.
5. Faddeev, D.K., V.N. Kublanovskaya and V.N. Faddeeva, 1968. Solution of linear algebraic systems with rectangular matrices. Proc. Steklov Inst. Math. 96: 93-111.
6. Fierro, R. and J. Bunch, 1995. Bounding the subspaces from rank revealing two-sided orthogonal decompositions. SIAM J. Matrix Anal. Appl., 16: 743-759.
7. Fierro, R., L. Vanhamme and S. Van Huffel, 1997. Total least squares algorithms based on rank-revealing complete orthogonal decompositions. SIAM Publications, pp: 99-116.
8. Golub, G.H. and C.F. Van Loan, 1996. Matrix Computations, 3rd Edn., The John Hopkins University Press, Baltimore, MD.
9. Gu, M. and S. Eisenstat, 1995. Downdating The singular value decomposition, SIAM J. Matrix Anal. Appl., 16: 793-810.
10. Hanson, R.J. and C.L. Lawson, 1969. Extensions and applications of the householder algorithm for solving linear least squares problems. Math. Comp. 23: 787-812.
11. Mathias, R. and G.W. Stewart, 1993. A block QR algorithm and the singular value decomposition. Lin. Alg. Appl., 182: 91-100.
12. Rudin, W., 1976. Principles of Mathematical Analysis, 3rd Edn., McGraw Hill, pp: 47-55.
13. Stewart, G.W., 1993. Updating a rank-revealing ULV decomposition. SIAM J. Matrix Anal. Appl., 14: 494-499.
14. Yoon, P.A. and J.L. Barlow, 1998. An efficient rank detection procedure for modifying the ULV decomposition, BIT, 38: 781-801.