https://dx.doi.org/10.22161/ijcmp.2.2.1 ISSN: 2456-866X

One Modification which Increases Performance of N-Dimensional Rotation Matrix Generation Algorithm

Ognyan Ivanov Zhelezov

Firm NET IT, bul. "Aleksandar Malinov" 91, 1715, Sofia, Bulgaria

Abstract*— *This article presents one modification of algorithm for generation of n-dimensional rotation matrix M, which rotates given n-dimensional vector X to the direction of given n-dimensional vector Y. Algorithm, named N-dimensional Rotation Matrix Generation Algorithm (NRMG) includes rotations of given vectors X and Y to the direction of coordinate axis x_1 using two-dimensional rotations in coordinate planes. Proposed modification decreases the number of needed two-dimensional rotations to 2(Lw-1) were Lw is the number of corresponding components of the two given vectors, that are not equal.

***Keywords*— Computations on matrices, Mathematics of computing, Mathematical analysis, Numerical analysis.**

I. INTRODUCTION

High-dimensional spaces frequently occur in mathematics and the sciences, in example n-dimensional feature space, which presents input signals of neural network or collection of n-dimensional parameters for multidimensional data analysis. Rotation is one of rigid transformations in geometrical space, which preserves length of vectors and can be presented using matrix operation like Y = M.X , where X and Y are input and output vector respectively, and M is rotation matrix. This article proposes one modification of n-dimensional rotation matrix generation algorithm, which increases performance if some of corresponding components of the two given vectors are equal.

II. SHORT DESCRIPTION OF ALGORITHMS FOR N-DIMENSIONAL ROTATION MATRIX GENERATION

Let's say that we have two n-dimensional vectors X and Y, having the same dimension, X, Y $\in R^n$. We want to obtain a rotation matrix M that satisfies the equation.

$$\tilde{Y} = M.X \quad (1)$$

where $\tilde{Y}$ has the same norm as X and the same direction as Y, i.e. $\|\tilde{Y}\| = \|X\|$ and $\cos(Y, \tilde{Y}) = 1$.

One of possibilities to generate rotation matrix M is to rotate given vectors in two-dimensional subspace, generated by vectors and map it back to R^n as follows [4],[19]:

Obtain two orthogonal vectors $u = X/\|X\|$ and $v = (Y-(u.Y)u)/\|(Y-(u.Y)u)\|$ that have the same norm. Then $P=uu^T+vv^T$ is a projection onto the space, generated by X and Y, and $Q=I - uu^T - vv^T$ is a projection onto the n-2 dimensional complementment subspace. So the rotation just has to take place on the range of P. This rotation has to be performed in R^2. Then result is mapped back to R^n by $[a,b]^T \rightarrow au+bv$. So matrix of rotation, which rotates X to the direction of Y is calculated as follows:

$$M = I - uu^T - vv^T + [u\ v]R(\theta)[u\ v]^T \quad (2)$$

where $R(\theta) = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix}$ and $\cos(\theta) = \frac{X^T Y}{\|X\|\|Y\|}$

Another way to generate rotation matrix M is to use Householder Reflection [5], [11], [18], [19]. If u and v are vectors with the same norm, in example obtained from input vectors X and Y as $u = X/\|X\|$ and $v = Y/\|Y\|$, there exists an orthogonal symmetric matrix P such that $v = P.u$ where $P = I - 2W.W^T$ and $W = (u-v)/\|u-v\|$. Matrix P is matrix of reflection (not a rotation) because det P = -1, (which gives the name to method) that's why to obtain matrix of rotation M, for which det M = 1, have to be performed two subsequent reflections. Matrix of rotation can be obtained as multiplication of two matrices of reflection P1 and P2 as M = P1.P2 .

The third way to generate rotation matrix M is to use rotation of given vectors to the direction of one of coordinate axes (i.e. axis $\vec{x}_1$). Algorithm, named N-dimensional Rotation Matrix Generation Algorithm (NRMG algorithm) [1] includes the following sequence of operations:

1) Obtaining rotation matrix M_X, which rotates given vector X to the direction of axis $\vec{x}_1$ as follows:

https://dx.doi.org/10.22161/ijcmp.2.2.1 ISSN: 2456-866X

$$M_X = \prod_{k=1}^{n-1} G\left(n-k, n-k+1, \theta_{n-k}\right) \quad (3)$$

where coefficients $\sin(\theta_k)$ and $\cos(\theta_k)$ of Givens matrices $G(k, k+1, \theta_k)$ are calculated using formulas:

$$\begin{cases} \sin(\theta_k) = -\dfrac{x_{k+1}}{\sqrt{x_k^2 + x_{k+1}^2}}, \\ \cos(\theta_k) = \dfrac{x_k}{\sqrt{x_k^2 + x_{k+1}^2}} \text{ if } x_k^2 + x_{k=1}^2 > 0 \\ \sin(\theta_k) = 0,\ \cos(\theta_k) = 1 \text{ if } x_k^2 + x_{k=1}^2 = 0 \end{cases} \quad (4)$$

2) Obtaining rotation matrix M_Y, which rotates given vector Y to the direction of axis using (3) and (4).
3) Obtaining rotation matrix M as multiplication of M_X by inverse matrix of M_Y given as $M=M_X^{-1}.M_Y$ [20].

III. DEFINITION OF THE TASK

Let's say that we have two n-dimensional vectors X and Y, having the same dimension and the same norm, X, Y $\in R^n$, $\|Y\| = \|X\|$ and let's some of their corresponding components x_i and y_i are equal: $x_i=y_i$ if $i \in K \subset \{1,\ldots,n\}$. We want to obtain a rotation matrix M that satisfies the equation.

$$Y = M.X \quad (5)$$

As it can be seen from this definition, proposed modification of NRMG algorithm needs the two given vectors X and Y to have the same norm, while the original NRMG algorithm[20] do not.

IV. DESCRIPTION OF MODIFIED NRMG ALGORITHM

Modified NRMG (MNRMG) algorithm for generation of N-dimensional rotation matrix M is NRMG algorithm [1], which uses rotations in coordinate planes only in subspace V, that is spanned by the unit vectors of axes, on which components of given vectors X и Y are not equal. MNRMG algorithm contains the following operations:

1) Comparing corresponding components of the two given vectors X и Y and obtaining vector , which consist of indexes of components that are not equal.
2) Obtaining rotation matrix M_X , which rotates projection of given vector X in subspace V to the direction of one of axes, having indexes in $\vec{w}$ (i.e. the first one)
3) Obtaining rotation matrix M_Y, which rotates projection of given vector Y in subspace V to the direction of the same axis as for projection of X.
4) Obtaining rotation matrix M, which rotates given vector X to the direction of given vector Y as multiplication of matrix M_X and inverse matrix of M_Y , as follows:

$$M = M_Y^{-1}.M_X = M_Y^T.M_X \quad (6)$$

Multiplication of vector by matrix M_X or by matrix M_Y do not change vector's components, which indexes are not in vector of indexes $\vec{w}$. Thus the length Lw of the vector $\vec{w}$ defines the number of rotations in coordinate planes (base operations of algorithm, described below), which coefficients have to be calculated to obtain searched matrix M. Below operations of proposed algorithm will be described subsequently

4.1. COMPARING CORRESPONDING COMPONENTS OF THE TWO GIVEN VECTORS X И Y AND OBTAINING VECTOR, WHICH CONTAINS INDEXES OF COMPONENTS THAT ARE NOT EQUAL

To obtain vector $\vec{w}$, which contains indexes of corresponding components of the two given vectors X and Y, that are not equal, have to be performed subsequent comparing of corresponding components of X and Y. Indexes of components, that are not equal, are saved subsequently in vector $\vec{w}$. In example if $X=[1\ 2\ 3\ 4]^T$ and $Y=[1\ 3\ 2\ 4]^T$ then after comparing of corresponding components vector $\vec{w}$ will have content $\vec{w}=[2\ 3]^T$. where 2 and 3 are indexes of second and third components, that are not equal. The length Lw of vector $\vec{w}$ is equal to the number of different corresponding components of vectors X и Y.

Vectors, that have zero values of components, which indexes are not in $\vec{w}$, generates subspace $V \subset R^n$. In example vectors $X2=[0\ 2\ 3\ 0]^T$ and $Y2=[0\ 3\ 2\ 0]^T$, obtained from vectors X and Y, given above, belongs to subspace $V \subset R4$. It is easy to see that X2 and Y2 are projections of vectors X and Y onto subspace V.

4.2. OBTAINING ROTATION MATRICES M_X AND M_Y, WHICH ROTATES PROJECTIONS OF GIVEN VECTORS X AND Y IN V TO THE DIRECTION OF ONE OF AXES, HAVING INDEXES IN (I.E. THE FIRST ONE)

Rotation of projection of given vector X in V to the direction of the first of coordinate axes, having indexes in $\vec{w}$ can be performed by subsequent multiplications by Givens matrices [8] as follow:

$$X1 = \prod_{k=Lw-1}^{1} G\left(w_k, w_{k+1}, \theta_{w_k}\right).X \quad (7)$$

where w_k, w_{k+1} are components of vector $\vec{w}$. Givens matrices G(wk, wk+1, θwk), k = Lw-1, Lw-2, … 1 are defined as follows [8], [13]:

$$G\left(w_k, w_{k+1}, \theta_{w_k}\right)=$$

$$\begin{vmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & C_{w_k} & \cdots & -S_{w_k} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & S_{w_k} & \cdots & C_{w_k} & & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 1 \end{vmatrix} \quad (8)$$

where θwk is the angle of rotation and coefficients Cwk=cos(θwk) and Swk=sin(θwk) appears at the intersections of wk-th and wk+1-th rows and columns. As it is shown in [1] coefficients Cwk=cos(θwk) and Swk=sin(θwk) can be calculated using formula:

$$\begin{cases} \sin(\theta_{w_k}) = -\dfrac{x_{w_{k+1}}}{\sqrt{x_{w_k}^2 + x_{w_{k+1}}^2}}, \\ \cos(\theta_{w_k}) = \dfrac{x_k}{\sqrt{x_{w_k}^2 + x_{w_{k+1}}^2}} \text{ if } x_{w_k}^2 + x_{w_{k+1}}^2 > 0 \\ \sin(\theta_{w_k}) = 0,\ \cos(\theta_{w_k}) = 1 \text{ if } x_{w_k}^2 + x_{w_{k+1}}^2 = 0 \end{cases} \quad (9)$$

Every one multiplication of vector by Givens matrix G(wk, wk+1, θwk) performs rotation of its projection in coordinate plane $(x_{w_k}, x_{w_{k+1}})$, which changes values only of vector's coordinates $x_{w_k}, x_{w_{k+1}}$ to $\overline{x}_{w_k}, \overline{x}_{w_{k+1}}$ .This multiplication can be presented as multiplication of coordinates by sub matrix A(wk, wk+1, θwk) as follows:

$$\begin{vmatrix} \overline{x}_{w_k} \\ \overline{x}_{w_{k+1}} \end{vmatrix} = \quad (11)$$

$$A\left(w_k, w_{k+1}, \theta_{w_k}\right) \begin{vmatrix} x_{w_k} \\ x_{w_{k+1}} \end{vmatrix} = \begin{vmatrix} \cos(\theta_{w_k}) & -\sin(\theta_{w_k}) \\ \sin(\theta_{w_k}) & \cos(\theta_{w_k}) \end{vmatrix} \cdot \begin{vmatrix} x_{w_k} \\ x_{w_{k+1}} \end{vmatrix}$$

Taking in consideration that multiplication with Givens matrix G(wk, wk+1, θwk) changes only values of vector's coordinates $x_{w_k}, x_{w_{k+1}}$ (to $\overline{x}_{w_k}, \overline{x}_{w_{k+1}}$), schema of multiplication by Givens matrix G(wk, wk+1, θwk) can be presented as operator for two-dimensional rotation as follows:

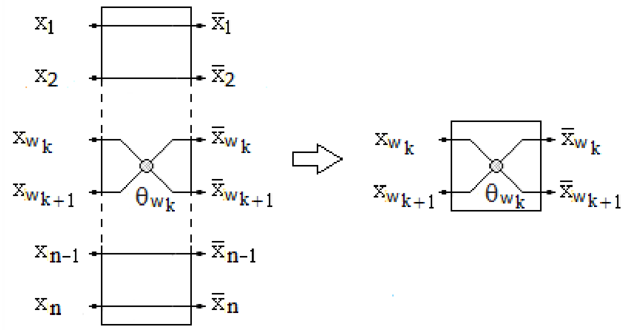


Fig. 1 Schema of two-dimensional rotation, performed by Givens matrix G(wk, wk+1, θwk)

The target of multiplication by Givens matrix G(wk, wk+1, θwk) is to set to zero coordinate $\overline{x}_{w_{k+1}}$. It is easy to find that equation $\overline{x}_{w_{k+1}} = 0$ and equations (10) are satisfied simultaneously when sin(θw_k) and cos(θw_k) are calculated using formula (11). Thus searched matrix M_X can be calculated as multiplication of Givens matrices as it is given in (7). Calculation of angles of two-dimensional rotations θk, k=Lw-1, Lw-2,…,1 really is not needed. Schema for rotation of vector X to the direction of the first of axes, which indexes are in $\vec{w}$ using two-dimensional rotations can be presented as follows:

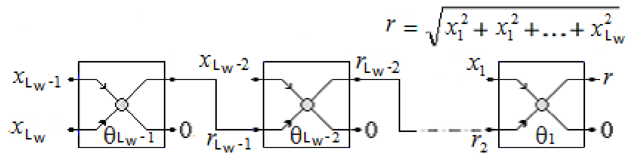


Fig. 2 Schema for rotation of vector X to the direction of first of axes, having indexes in $\vec{w}$

If calculation uses parallel execution of two-dimensional rotations [4],[16] (named Accelerated Rotation AR in [20]) matrices Mx and My are calculated as multiplication of $\lceil \log_2 Lw \rceil$ matrices of stages for which coefficients Cw_k and Sw_k are calculated using (9). The proposed modification of NRMG algorithm avoids execution of base operations for vector's components, which indexes are not in vector $\vec{w}$. In example let's we have two 8-dimensional vectors X and Y, that have two equal components, i.e. $x_3=y_3$ and $x_6=y_6$, Then schema, which perform rotation of X to the direction of Y using MNRMG algorithm and AR will look as follows:

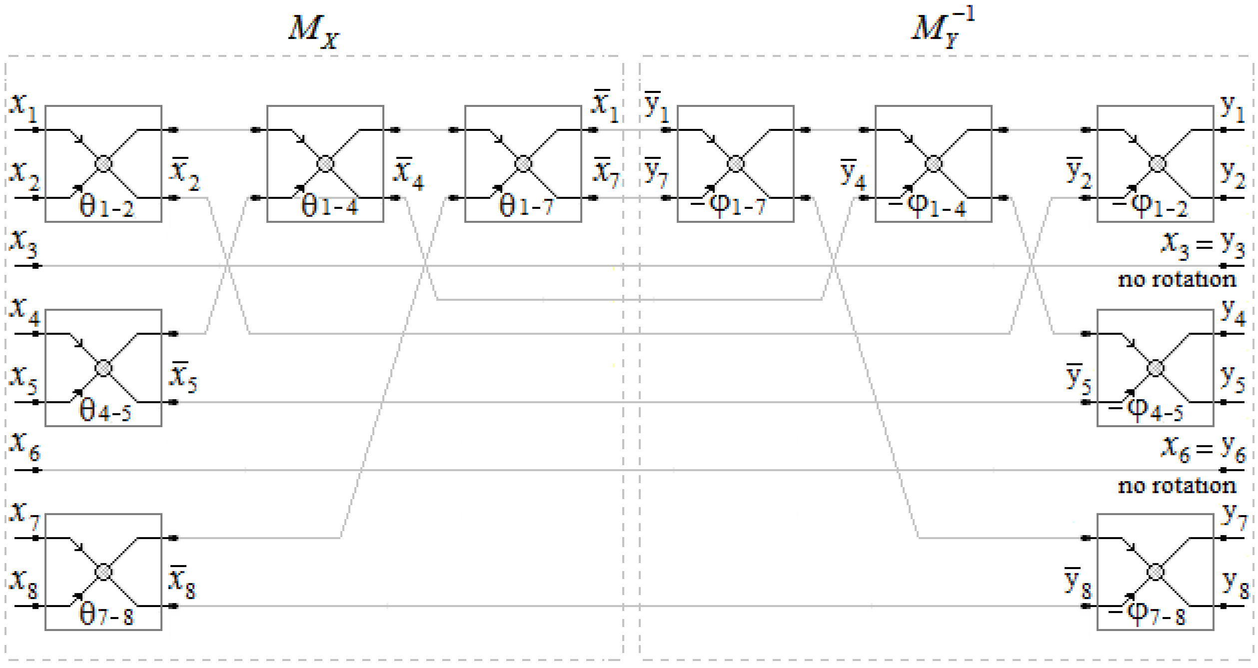


Fig.3: Schema for rotation of 8-dimensional vector X to the direction of vector Y using Modified NRMG algorithm and AR if $\|Y\| = \|X\|$, *x_3=y_3 and x_6=y_6*

Every one of matrices M_X and M_Y is calculated by multiplication of $\lceil \log_2 Lw \rceil$ matrices of stages (in this example $\lceil \log_2 6 \rceil = 3$) for which coefficients Cw and Sw are calculated using (9). It is important to note that Modified NRMG algorithm needs that two given vectrors X and Y have the same norm $\|Y\| = \|X\|$ while the original NRMG algorithm [20] do not.

V. NUMERICAL EXPERIMENTS

Has been created Matlab program to test proposed Modified NRMG algorithm. Program contains function for accelerated rotation fnMAR and code, that uses this function to obtain matrix M, which rotates given vector X to the direction of given vector Y.

```
function R = fnMAR(X,w)
N = length(X);    %X have to be row vector (transposed)
Lw= length(w);  %w have to be row vector (transposed)
if N<=Lw      %Length of X can't exceed the length of w
  R= eye(N);  %Initial rotation matrix = Identity matrix
  step = 1;           %Initial step
  while(step<N)   %Loop to create matrices of stages
    A= eye(N);
    n=1;
    while(n<=N-step && w(n+step) > 0)
      r2 = X(w(n))*X(w(n)) +
X(w(n+step))*X(w(n+step));
      if r2 > 0
        r = sqrt(r2);
        pcos = X(w(n))/r;    %Calculation of coefficients
        psin = -X(w(n+step))/r;
        % Base 2-dimensional rotation
        A(w(n), w(n)) = pcos;
        A(w(n), w(n+step)) = -psin;
        A(w(n+step), w(n)) = psin;
        A(w(n+step), w(n+step)) =  pcos;
        X(w(n+step))=0;
        X(w(n))=r;
      end;
      n=n+2*step;  % Move to the next base operation
    end;
    step = step*2;
    R= A*R;  % Multiply R by current matrix of stage A
  end;
end;
```

Example 1 Code of Matlab function for accelerated rotation of vector X to the direction of first axis having index in $\vec{w}$ *. Function returns matrix of rotation* M_X

Matlab code, that uses this function to obtain matrix M, which rotates given vector X to the direction of given vector Y is given below:

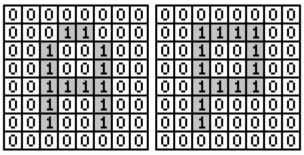
```
N = length(X);       %X and Y have to be row vectors
normX= norm(X);  %(transposed)
normY= norm(Y);
if normX ~= normY        %Set norm of Y equal to norm
  Y=( normX/normY)*Y;  % of X if they are different
end;
w=zeros(1,N);  m=1;  % Initialization of vector w
for(n=1:N)   %Loop to create vector of indexes w
  if X(n) ~= Y(n)
```

https://dx.doi.org/10.22161/ijcmp.2.2.1 *ISSN: 2456-866X*

```
    w(m)=n;   %save in w index of not equal elenents
    m=m+1;
  end
end
Mx=fnMAR(X,w);
My= fnMAR(Y,w);
M= My'*Mx;
Z=M*X';
if max(abs(Z'-Y)) < 10^-6
disp('Z and Y are identical');
end
```

Example 2 Matlab code, which creates matrix of rotation M using fnMAR function

For test data have been used the following two images:



X Y

Images can be presented as 64-dimensional vectors as follows (written using Matlab Language syntax for row vectors):

X =[0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0];

Y =[0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0];

Test program, given above, obtain vector Z to the direction of vector Y using Modified NRMG algorithm. Program displays message “Z and Y are identical”, which shows that vectors Z and Y are identical at least with precision of 10^{-6}. In this example Modified NRMG algorithm decreases the number of executed base operations to 4 (which is the number of different corresponding components of the two vectors X and Y) while for the same vectors NRMG algorithm without modification executes 70 base operations.

VI. CONCLUSION

In this article we proposes one modification of NRMG algorithm for generation of n-dimensional rotation matrix M, which rotates given n-dimensional vector X to the direction of given vector Y having the same dimension and the same norm. Modification decreases the number of base operations to 2(Lw-1) where Lw is the number of corresponding components of given vectors, that are not equal. Thus when the two given vectors X and Y have significant number of corresponding components that are equal, Modified NRMG algorithm can be more effective than existing algorithms, shortly described above.

It is important to note, that no one of algorithms for n-dimensional matrix generation, described in p.2 do not provide possibility to decrease needed calculations when exists some linear or non-linear dependence between given vectors X and Y. Proposed modification of NRMG algorithm provides decreasing of needed calculations using that NRMG algorithm consists of sequence of identical operations – multiplications of vector by Givens matrices. If some of corresponding components of given vectors are equal, they are not changed, which decreases the number of needed base operations.

REFERENCES

[1] O. I. Zhelezov, 2017, N-dimensional Rotation Matrix Generation Algorithm, American Journal of Computational and Applied Mathematics , Vol. 7 No. 2, 2017, pp. 51-57. doi: 10.5923/j.ajcam. 20170702.04.

[2] H. G. Golub, J. M. Ortega, 1993, Scientific Computing and Introduction with Parallel Computing, Academic Press, Inc., San Diego.

[3] G. A. Korn, T. M. Korn, 1961, Mathematical Handbook for Scientists and Engineers (1st ed.), New York: McGraw-Hill. pp. 55–79.

[4] H. Friedberg, A. Insel, L. Spence, 1997, Linear Algebra (3rd ed), Prentice Hall.

[5] D. A. Harville, 1997, Matrix Algebra from Statistician’s Perspective, Softcover.

[6] B. Buchberger, 1985, Multidimensional Systems Theory - Progress Directions and Open Problems in Multidimensional Systems, Reidel Publishing Company.

[7] G. H. Golub, C. F. Van Loan, , 1996, Matrix Computations, 4rd edition. Johns Horkins University Press, Baltimore

[8] M. Cosnard, Y. Robert. Complexity of parallel QR factorization. J. ACM 33, 4 (August 1986), 712-723. DOI=10.1145/6490.214102, 1986.

[9] A. H. Sameh, D. J. Kuck. On Stable Parallel Linear System Solvers. J. ACM 25, 1 (January 1978), 81-91. DOI= http://dx.doi.org/10.1145/322047.322054, 1978.

[10] N. J. Higham, 1996, Accuracy and Stability of Numerical Algorithms, SIAM, Philadelfia.

[11] G. W.Steward, 1976, The economical storage of plane rotations, Numer. Math, 25, 2 1976, 137-139

[12] Matlock, H., and Reese, L.C., 1960, Generalized solutions for laterally loaded piles., Journal of Soil Mechanics and Foun-dation, 86(5), 63–91.

[13] N. K. Bose, 1985, Multidimensional Systems Theory: Progress, Directions, and Open Problems. D.Reidel Publishing Co., Dordrecht, The Netherland.

[14] A. S. Householder, 1958, “Unitary triangularization of a nonsimetric matrix”, J. ACM 5, 339-342, 1958.

[15] E. Anderson, 2000, "Discontinuous Plane Rotations and the Symmetric Eigenvalue Problem" LAPACK Working Note 150, University of Tennessee, UT-CS-00-454, December 4, 2000. page 2.
[16] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, 1995, "Computer Graphics, Principles and Practice", 2nd edition in C, Addison-Wesley, ISBN 0-201-84840-6.
[17] G. Strang, 2006, Linear Algebra and its Applications, Thomson Learning Ink, pages 69-135, ISBN 0-03-010567.
[18] St. Roman, Advanced Linear Algebra, second ed., 2005 Springer-Verlag, New York. pages 59-85, ISBN: 978-1-4757-2180-5
[19] J. E. Gentle, 2007, Matrix Algebra: Theory, Computations, and Applications in Statistics, Springer, page 180, ISBN 978-0-387-70872-0
[20] I. R. Shafarevich, A. Remizov, 2013, Linear Algebra and Geometry, Springer, pages 133-160, ISBN 978-3-642-30993-9