Some Open Problems Linked to the Cosine-Sine Decompositions

Vjeran Hari [1] Josip Matejaš [2]

[1]Faculty of Science, Department of Mathematics, University of Zagreb, hari@math.hr

[2]Faculty of Economics and Business, University of Zagreb, jmatejas@efzg.hr

ApplMath 18
September 17–20, 2018, Solaris, Šibenik, Croatia

OUTLINE

OUTLINE

- Cosine-Sine Decomposition

This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.

OUTLINE

- Cosine-Sine Decomposition
- Known Algorithms

This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.

OUTLINE

- Cosine-Sine Decomposition
- Known Algorithms
- A New Approach

This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.

OUTLINE

- Cosine-Sine Decomposition
- Known Algorithms
- A New Approach
- Some Applications

This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.

OUTLINE

- Cosine-Sine Decomposition
- Known Algorithms
- A New Approach
- Some Applications
- Solving 4 Symmetric Eigenproblem with 6 Rotations

This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.

Cosine-Sine Decomposition (CSD) of Orthogonal Matrix Q

Let Q be orthogonal matrix of order n and let

$$Q = \begin{array}{c} \left[\begin{array}{cc} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{array} \right] \begin{array}{l} \} \, l \\ \} \, n-l \end{array} \\ \begin{array}{cc} \underbrace{\phantom{Q_{11}}}_{l} & \underbrace{\phantom{Q_{12}}}_{n-l} \end{array} \end{array} \tag{1}$$

be the partition of Q defined by l, $1 \leq l \leq n-1$.

Cosine-Sine Decomposition (CSD) of Orthogonal Matrix Q

Let Q be orthogonal matrix of order n and let

$$Q = \begin{array}{c@{}l} \begin{array}{cc} \left[\begin{array}{cc} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{array}\right] \\ \underbrace{\phantom{Q_{11}}}_{l} \; \underbrace{\phantom{Q_{12}}}_{n-l} \end{array} & \begin{array}{l} \} \, l \\ \} \, n-l \\ \end{array} \end{array} \qquad (1)$$

be the partition of Q defined by l, $1 \leq l \leq n-1$.

The Cosine-Sine decomposition of Q is read

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} U_1 & \\ & U_2 \end{bmatrix} \begin{bmatrix} C & -S \\ S & C \end{bmatrix} \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}^T$$

C, S diagonal, , $C^2 + S^2 = I$, $\quad U_1,\ U_2,\ V_1,\ V_2$ orthogonal.

A Bit of History

- C. C. Paige, M. Wei, History and Generality of the CS Decomposition, Linear Algebra and its Appl. 208 (1994) 303-326

A Bit of History

- C. C. Paige, M. Wei, History and Generality of the CS Decomposition, Linear Algebra and its Appl. 208 (1994) 303-326
- G. W. Stewart, Computing the CS decomposition of a partitioned orthogonal matrix, Numer. Math. 40 (1982) 297-306
- C. Van Loan. Computing the CS and the generalized singular value decompositions, Numer. Math. 46(4) (1985) 479491

A Bit of History

- C. C. Paige, M. Wei, History and Generality of the CS Decomposition, Linear Algebra and its Appl. 208 (1994) 303-326
- G. W. Stewart, Computing the CS decomposition of a partitioned orthogonal matrix, Numer. Math. 40 (1982) 297-306
- C. Van Loan. Computing the CS and the generalized singular value decompositions, Numer. Math. 46(4) (1985) 479491
- B. D. Sutton, Computing the complete CS decomposition, Numer. Algorithms 50 (209) 3365
- B. D. Sutton, Stable computation of the CS decomposition: simultaneous bidiagonalization, SIAM J. Matrix Anal. Appl. 33 (2012) 121
- D. Calvetti, L. Reichel, H. Xu, A CS decomposition for orthogonal matrices with application to eigenvalue computation, Linear Algebra and its Appl. 476 (2015) 10.1016/j.laa.2015.03.007

Applications of CSD

Z. Bai, The CSD, GSVD, their Applications and Computations 1999 (preprint)

Applications of CSD

Z. Bai, The CSD, GSVD, their Applications and Computations 1999
(preprint)

In matrix theory, CSD is used

- to define canonical angles between two subspaces of $\mathbf{R}^n$
- in the theory of orthogonal projections
- in solving GSVD
- in accelerating block Jacobi methods
- in quantum compiling
- . . .

A New Approach to CSD Computation

The first goal is to find out what happens if we make SVDs of the diagonal blocks of Q and extract the left and right singular vector matrices:

A New Approach to CSD Computation

The first goal is to find out what happens if we make SVDs of the diagonal blocks of Q and extract the left and right singular vector matrices:

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

A New Approach to CSD Computation

The first goal is to find out what happens if we make SVDs of the diagonal blocks of Q and extract the left and right singular vector matrices:

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

$$U = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix}.$$

A New Approach to CSD Computation

The first goal is to find out what happens if we make SVDs of the diagonal blocks of Q and extract the left and right singular vector matrices:

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

$$U = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix}.$$

We have

$$W = U^T Q V = \begin{bmatrix} C_1 & W_{12} \\ W_{21} & C_2 \end{bmatrix}, \tag{2}$$

where

$$W_{12} = U_{11}^T Q_{12} V_{22}, \qquad W_{21} = U_{22}^T Q_{21} V_{11}.$$

and C_1 and C_2 are diagonal.

CSD Computation in Exact Arithmetic

We can assume

$$C_1 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_l), \qquad C_2 = \mathrm{diag}\,(\gamma_{l+1}, \ldots, \gamma_n)$$

$$\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_l\,, \qquad \gamma_{l+1} \geq \gamma_{l+2} \geq \cdots \geq \gamma_n\,.$$

CSD Computation in Exact Arithmetic

We can assume

$$C_1 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_l), \qquad C_2 = \mathrm{diag}\,(\gamma_{l+1}, \ldots, \gamma_n)$$

$$\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_l\,, \qquad \gamma_{l+1} \geq \gamma_{l+2} \geq \cdots \geq \gamma_n\,.$$

Lemma

Let $n = 2l$. Then $C_1 = C_2$ and W_{12}, W_{21} are block diagonal. If $\gamma_n > 0$ then $W_{21} = -W_{12}^T$. Otherwise W_{21} and $-W_{12}^T$ can differ in the last diagonal blocks. In the special case when $C_2 = O$ or $C_1 = O$ these blocks are the whole matrices W_{12} and W_{21}.

CSD Computation in Exact Arithmetic

Theorem

Let W be orthogonal matrix satisfying the above relations. If $2l \geq n$ then

$$W = \begin{array}{c} \left[\begin{array}{ccc} I & & \\ & C & S_1 \\ & S_2 & C \end{array} \right] \begin{array}{l} \}\, 2l-n \\ \}\, n-l \\ \}\, n-l \end{array} \\ \begin{array}{ccc} \underbrace{\qquad}_{2l-n} & \underbrace{\qquad}_{n-l} & \underbrace{\qquad}_{n-l} \end{array} \end{array}$$

If $2l < n$, then

$$W = \begin{array}{c} \left[\begin{array}{ccc} C & & S_1 \\ & I & \\ S_2 & & C \end{array} \right] \begin{array}{l} \}\, l \\ \}\, n-2l \\ \}\, l \end{array} \\ \begin{array}{ccc} \underbrace{\qquad}_{l} & \underbrace{\qquad}_{n-2l} & \underbrace{\qquad}_{l} \end{array} \end{array}$$

where C is diagonal with nonnegative diagonal elements arranged nonincreasingly, S_1 and S_2 are block-diagonal such that each diagonal block of S_1 and of S_2 is some multiple of an orthogonal matrix. The relation $S_1 = -S_2^T$ holds, except possibly for the last diagonal block. If all diagonal elements of C are distinct, S_1 and S_2 are diagonal and $S_1^2 = S_2^2 = I - C^2$ holds.

$$\begin{aligned} C &= \operatorname{diag}\,(\,\gamma^{(1)} I_{n_1},\ \ldots\ ,\ \gamma^{(p-1)} I_{n_{p-1}},\ \ \gamma^{(p)} I_{n_p}), \\ S_2 &= \operatorname{diag}\,(\,\sigma^{(1)} S_{11}, \ldots\ ,\ \ \sigma^{(p-1)} S_{p-1,p-1},\ \ \sigma^{(p)} S_{pp}), \\ S_1 &= \operatorname{diag}\,(-\sigma^{(1)} S_{11}^T, \ldots - \sigma^{(p-1)} \tilde{S}_{p-1,p-1}^T,\ -\sigma^{(p)} \tilde{S}_{pp}^T), \end{aligned}$$

To obtain the CSD of Q, we make the block-diagonal orthogonal matrices

$$\begin{aligned} \tilde{U} &= \begin{bmatrix} I_0 & \\ & \operatorname{diag}(S_{11}, \ldots, S_{pp}) \end{bmatrix}, \\ \tilde{V} &= \begin{bmatrix} I_0 & \\ & \operatorname{diag}(S_{11}^T, \ldots \tilde{S}_{pp}^T) \end{bmatrix}, \end{aligned}$$

where I_0 stands for I_{2l-n} (I_{n-2l}) provided that $2l > n$ ($2l < n$). It does not exist when $2l = n$. Then make the transformation $\tilde{W} = \tilde{U}^T W \tilde{V}$.
The matrix $\tilde{W}$ has the same form as W in Theorem 2, but C (resp. S_2, S_1) is replaced by Γ, (resp. Σ, $-\Sigma$). Here

$$\Gamma = \operatorname{diag}(\gamma^{(1)} I_{n_1}, \ldots, \gamma^{(p)} I_{n_p}), \quad \Sigma = \operatorname{diag}(\sigma^{(1)} I_{n_1}, \ldots, \sigma^{(p)} I_{n_p}),$$

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations
- Just storing a matrix in the computer generates small relative errors in the matrix elements

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations
- Just storing a matrix in the computer generates small relative errors in the matrix elements
- $\tilde{Q}^T\tilde{Q}$ and $\tilde{Q}\tilde{Q}^T$ are close to identity

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations
- Just storing a matrix in the computer generates small relative errors in the matrix elements
- $\tilde{Q}^T\tilde{Q}$ and $\tilde{Q}\tilde{Q}^T$ are close to identity
- Our goal here is to compute an approximate CS decomposition of $\tilde{Q}$ which is as accurate as the data warrant

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations
- Just storing a matrix in the computer generates small relative errors in the matrix elements
- $\tilde{Q}^T\tilde{Q}$ and $\tilde{Q}\tilde{Q}^T$ are close to identity
- Our goal here is to compute an approximate CS decomposition of $\tilde{Q}$ which is as accurate as the data warrant
- To this end $\tilde{Q}$ is partitioned, as earlier, so that the diagonal blocks are of order l and $n-l$.

Computing CSD in Finite Arithmetic

In computer we rarely have orthogonal matrix Q, rather we have $\tilde{Q}$ which is almost orthogonal.

- Usually, $\tilde{Q}$ computed as product of Householder reflectors or plane rotations
- Just storing a matrix in the computer generates small relative errors in the matrix elements
- $\tilde{Q}^T\tilde{Q}$ and $\tilde{Q}\tilde{Q}^T$ are close to identity
- Our goal here is to compute an approximate CS decomposition of $\tilde{Q}$ which is as accurate as the data warrant
- To this end $\tilde{Q}$ is partitioned, as earlier, so that the diagonal blocks are of order l and $n-l$.
- We assume that the two initial diagonalizations of the diagonal blocks $\tilde{Q}_{11}$ and $\tilde{Q}_{22}$ are already performed, so that these diagonal blocks are diagonal. By $\tilde{W}$ we denote the computed version of W from the preceding section

Computing CSD in Finite Arithmetic

Since $\tilde{W}$ is almost orthogonal, we can assume

$$\tilde{W}^T\tilde{W} = I + E, \quad \tilde{W}\tilde{W}^T = I + F, \quad \|E\|_2 \leq \varepsilon,\ \|F\|_2 \leq \varepsilon$$

ε is a small number, typically like $\mathcal{O}(n\mathbf{u})$ or $\mathcal{O}(n^2\mathbf{u})$, where $\mathbf{u}$ denotes the unit roundoff of the finite arithmetic used in the computation. The bound ε measures how close to orthogonality is $\tilde{W}$.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

We assume

$$\tilde{W} = \begin{bmatrix} \Gamma^{+} & \Psi^{T} & Y^{T} \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^{-} \end{bmatrix}$$

- Γ^{+} and Γ^{-} are of order l

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

We assume

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

- Γ^+ and Γ^- are of order l
- the central term is written in the form $I - \Delta'$ because we have

$$\Psi\Psi^T + (I - \Delta')^2 \approx I.$$

So, we expect that the diagonal elements of Δ' are nonnegative.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

We assume

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

- Γ^+ and Γ^- are of order l
- the central term is written in the form $I - \Delta'$ because we have

$$\Psi\Psi^T + (I - \Delta')^2 \approx I.$$

So, we expect that the diagonal elements of Δ' are nonnegative.

Set

$$\Gamma = \frac{1}{2}(\Gamma^+ + \Gamma^-)$$

then we have

$$\Gamma^+ = \Gamma(I + \Delta), \quad \Gamma^- = \Gamma(I - \Delta)$$

and we can assume

$$\|\Gamma\Delta\|_2 \leq \varepsilon_1\,, \;\; \|\Delta'\|_2 \leq \varepsilon_1\,.$$

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

$$\Gamma = \frac{1}{2}(\Gamma^+ + \Gamma^-), \quad \Gamma^+ = \Gamma(I + \Delta), \quad \Gamma^- = \Gamma(I - \Delta)$$

We can assume

$$\|\Gamma\Delta\|_2 \leq \varepsilon_1 \,, \; \|\Delta'\|_2 \leq \varepsilon_1 \,.$$

Here, ε_1 is a modest multiple of $\mathbf{u}$. Why?

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

$$\Gamma = \frac{1}{2}(\Gamma^+ + \Gamma^-), \quad \Gamma^+ = \Gamma(I + \Delta), \quad \Gamma^- = \Gamma(I - \Delta)$$

We can assume

$$\|\Gamma\Delta\|_2 \leq \varepsilon_1\,, \; \|\Delta'\|_2 \leq \varepsilon_1\,.$$

Here, ε_1 is a modest multiple of $\mathbf{u}$. Why?

- $\Gamma\Delta$ has as diagonal elements the means of the absolute errors of the singular values of $\tilde{Q}_{11}$ and $\tilde{Q}_{22}$ and they are tiny.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

$$\Gamma = \frac{1}{2}(\Gamma^+ + \Gamma^-), \quad \Gamma^+ = \Gamma(I + \Delta), \quad \Gamma^- = \Gamma(I - \Delta)$$

We can assume

$$\|\Gamma\Delta\|_2 \leq \varepsilon_1 \, , \; \|\Delta'\|_2 \leq \varepsilon_1 \, .$$

Here, ε_1 is a modest multiple of $\mathbf{u}$. Why?

- $\Gamma\Delta$ has as diagonal elements the means of the absolute errors of the singular values of $\tilde{Q}_{11}$ and $\tilde{Q}_{22}$ and they are tiny.
- We have assumed the same bound for $\|\Gamma\Delta\|_2$ and for $\|\Delta'\|_2$ because Δ' and Γ^- are parts of the same SVD computation of $\tilde{Q}_{22}$.

$$\tilde{W} = \begin{bmatrix} \Gamma^{+} & \Psi^{T} & Y^{T} \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^{-} \end{bmatrix}$$

Lemma

The following assertions hold.

(i)
$$\|\Phi\|_2 \leq \sqrt{\varepsilon} \quad \textit{and} \quad \|\Psi\|_2 \leq \sqrt{\varepsilon}. \tag{3}$$

(ii) *For all i, j, such that $\gamma_i \neq \gamma_j$,*

$$x_{ij} = \frac{\xi_{ij}}{\gamma_i - \gamma_j} \quad \textit{and} \quad y_{ij} = \frac{\eta_{ij}}{\gamma_i - \gamma_j}, \quad |\xi_{ij}| \leq |\varepsilon|, \; |\eta_{ij}| \leq |\varepsilon|. \tag{4}$$

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

These results are analogous to those of Lemma 1 and theorem 2. They tell us that tiny gap between successive diagonals of Γ can make the appropriate off-diagonal elements of X and Y large. Hence, in order to compute the CS decomposition we shall have to work on X and Y.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

These results are analogous to those of Lemma 1 and theorem 2. They tell us that tiny gap between successive diagonals of Γ can make the appropriate off-diagonal elements of X and Y large. Hence, in order to compute the CS decomposition we shall have to work on X and Y.

- We know that the possible large elements in X and Y form a small diagonal block (in earlier notation: a diagonal submatrix $-Y_i^T$ in X and Y_i in Y) which is close to a multiple of orthogonal matrix

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

These results are analogous to those of Lemma 1 and theorem 2. They tell us that tiny gap between successive diagonals of Γ can make the appropriate off-diagonal elements of X and Y large. Hence, in order to compute the CS decomposition we shall have to work on X and Y.

- We know that the possible large elements in X and Y form a small diagonal block (in earlier notation: a diagonal submatrix $-Y_i^T$ in X and Y_i in Y) which is close to a multiple of orthogonal matrix
- Note that the QL, QR, LQ, RQ factorizations of an almost diagonal matrix actually almost diagonalizes it

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W} = \begin{bmatrix} \Gamma^+ & \Psi^T & Y^T \\ \Phi & I - \Delta' & 0 \\ X & 0 & \Gamma^- \end{bmatrix}$$

These results are analogous to those of Lemma 1 and theorem 2. They tell us that tiny gap between successive diagonals of Γ can make the appropriate off-diagonal elements of X and Y large. Hence, in order to compute the CS decomposition we shall have to work on X and Y.

- We know that the possible large elements in X and Y form a small diagonal block (in earlier notation: a diagonal submatrix $-Y_i^T$ in X and Y_i in Y) which is close to a multiple of orthogonal matrix
- Note that the QL, QR, LQ, RQ factorizations of an almost diagonal matrix actually almost diagonalizes it
- Therefore, we can make the QL factorization of X, $X = Q_X L_X$ and of Y, $Y = Q_Y L_Y = Q_Y R^T$ and transform $\tilde{W}$ appropriately

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W}' = \begin{bmatrix} I & O \\ O & Q_X^T \end{bmatrix} \tilde{W} \begin{bmatrix} I & O \\ O & Q_Y \end{bmatrix} = \begin{bmatrix} \Gamma(I+\Delta) & \Psi^T & R \\ \Phi & I-\Delta' & O \\ L_X & O & Q_X^T \Gamma(I-\Delta) Q_Y \end{bmatrix}.$$

Computing CSD in Finite Arithmetic, Case: $2l \le n$

$$\tilde{W}' = \begin{bmatrix} I & O \\ O & Q_X^T \end{bmatrix} \tilde{W} \begin{bmatrix} I & O \\ O & Q_Y \end{bmatrix} = \begin{bmatrix} \Gamma(I+\Delta) & \Psi^T & R \\ \Phi & I-\Delta' & O \\ L_X & O & Q_X^T\Gamma(I-\Delta)Q_Y \end{bmatrix}.$$

- If the norm of Y is close to $\mathbf{u}$, the Q_Y becomes dependent on the errors in $\tilde{W}$ which are also of that order of magnitude

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W}' = \begin{bmatrix} I & O \\ O & Q_X^T \end{bmatrix} \tilde{W} \begin{bmatrix} I & O \\ O & Q_Y \end{bmatrix} = \begin{bmatrix} \Gamma(I+\Delta) & \Psi^T & R \\ \Phi & I-\Delta' & O \\ L_X & O & Q_X^T \Gamma(I-\Delta) Q_Y \end{bmatrix}.$$

- If the norm of Y is close to $\mathbf{u}$, the Q_Y becomes dependent on the errors in $\tilde{W}$ which are also of that order of magnitude
- In that case, the same is true for Q_X since $\|X\| \approx \|Y\|$. This can make $Q_X^T \Gamma(I-\Delta) Q_Y$ to be far from Γ. Therefore, we shall follow another way.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W}' = \begin{bmatrix} I & O \\ O & Q_X^T \end{bmatrix} \tilde{W} \begin{bmatrix} I & O \\ O & Q_Y \end{bmatrix} = \begin{bmatrix} \Gamma(I+\Delta) & \Psi^T & R \\ \Phi & I-\Delta' & O \\ L_X & O & Q_X^T \Gamma(I-\Delta) Q_Y \end{bmatrix}.$$

- If the norm of Y is close to $\mathbf{u}$, the Q_Y becomes dependent on the errors in $\tilde{W}$ which are also of that order of magnitude
- In that case, the same is true for Q_X since $\|X\| \approx \|Y\|$. This can make $Q_X^T \Gamma(I-\Delta) Q_Y$ to be far from Γ. Therefore, we shall follow another way.
- We split $\Gamma = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_l)$ into two parts,

$$\Gamma = \begin{bmatrix} \Gamma_1 & O \\ O & \Gamma_2 \end{bmatrix}, \quad \Gamma_1 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_m), \quad 0 \leq m \leq l,$$

where m is the largest index for which $\gamma_m \geq \frac{\sqrt{2}}{2}, \quad 0 \leq m \leq l$, holds. Thus, if $\gamma_l \geq \sqrt{2}/2$, then $\Gamma = \Gamma_1$ and if $\gamma_1 < \sqrt{2}/2$, then $\Gamma = \Gamma_2$.

Computing CSD in Finite Arithmetic, Case: $2l \leq n$

$$\tilde{W}' = \begin{bmatrix} I & O \\ O & Q_X^T \end{bmatrix} \tilde{W} \begin{bmatrix} I & O \\ O & Q_Y \end{bmatrix} = \begin{bmatrix} \Gamma(I+\Delta) & \Psi^T & R \\ \Phi & I-\Delta' & O \\ L_X & O & Q_X^T \Gamma(I-\Delta) Q_Y \end{bmatrix}.$$

- If the norm of Y is close to $\mathbf{u}$, the Q_Y becomes dependent on the errors in $\tilde{W}$ which are also of that order of magnitude
- In that case, the same is true for Q_X since $\|X\| \approx \|Y\|$. This can make $Q_X^T \Gamma(I-\Delta) Q_Y$ to be far from Γ. Therefore, we shall follow another way.
- We split $\Gamma = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_l)$ into two parts,

$$\Gamma = \begin{bmatrix} \Gamma_1 & O \\ O & \Gamma_2 \end{bmatrix}, \quad \Gamma_1 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_m), \quad 0 \leq m \leq l,$$

  where m is the largest index for which $\gamma_m \geq \frac{\sqrt{2}}{2}, \quad 0 \leq m \leq l$, holds. Thus, if $\gamma_l \geq \sqrt{2}/2$, then $\Gamma = \Gamma_1$ and if $\gamma_1 < \sqrt{2}/2$, then $\Gamma = \Gamma_2$.
- Although the story complicates, now we can make sharp estimates for almost all blocks appearing in $\tilde{W}'$.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- In today's CPU and GPU eigenvalue and singular value computation, very efficient and accurate are the block diagonalization methods. They are generalizations of the element-wise Jacobi methods. Instead of annihilating two off-diagonal elements, they annihilate two off-diagonal blocks.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- In today's CPU and GPU eigenvalue and singular value computation, very efficient and accurate are the block diagonalization methods. They are generalizations of the element-wise Jacobi methods. Instead of annihilating two off-diagonal elements, they annihilate two off-diagonal blocks.
- The dimension of these blocks depends on the available cache memory of the computing machine. Typically these blocks are of order 16–256. For the annihilation of these blocks one employs a special method that is referred to as kernel algorithm.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- In today's CPU and GPU eigenvalue and singular value computation, very efficient and accurate are the block diagonalization methods. They are generalizations of the element-wise Jacobi methods. Instead of annihilating two off-diagonal elements, they annihilate two off-diagonal blocks.
- The dimension of these blocks depends on the available cache memory of the computing machine. Typically these blocks are of order 16–256. For the annihilation of these blocks one employs a special method that is referred to as kernel algorithm.
- So, the kernel algorithm diagonalizes a symmetric matrix of order 32–512. Typically, an element-wise Jacobi method is used.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- In today's CPU and GPU eigenvalue and singular value computation, very efficient and accurate are the block diagonalization methods. They are generalizations of the element-wise Jacobi methods. Instead of annihilating two off-diagonal elements, they annihilate two off-diagonal blocks.
- The dimension of these blocks depends on the available cache memory of the computing machine. Typically these blocks are of order 16–256. For the annihilation of these blocks one employs a special method that is referred to as kernel algorithm.
- So, the kernel algorithm diagonalizes a symmetric matrix of order 32–512. Typically, an element-wise Jacobi method is used.
- The kernel algorithm can be made more efficient if instead of two off-diagonal elements it annihilates two blocks of small order 2–4. If the blocks are of order 2 then there is a need for a simple, efficient and accurate eigenvalue algorithm for a symmetric matrix of order 4.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- In today's CPU and GPU eigenvalue and singular value computation, very efficient and accurate are the block diagonalization methods. They are generalizations of the element-wise Jacobi methods. Instead of annihilating two off-diagonal elements, they annihilate two off-diagonal blocks.
- The dimension of these blocks depends on the available cache memory of the computing machine. Typically these blocks are of order 16–256. For the annihilation of these blocks one employs a special method that is referred to as kernel algorithm.
- So, the kernel algorithm diagonalizes a symmetric matrix of order 32–512. Typically, an element-wise Jacobi method is used.
- The kernel algorithm can be made more efficient if instead of two off-diagonal elements it annihilates two blocks of small order 2–4. If the blocks are of order 2 then there is a need for a simple, efficient and accurate eigenvalue algorithm for a symmetric matrix of order 4.
- We shall show how the CSD can be used to shed some light how to the construction 6 plane rotations that diagonalize the given 4×4 symmetric matrix.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

be a symmetric matrix of order 4 and let Q be orthogonal and such that

$$A = Q\Lambda Q^\tau.$$

Recall the CSD of Q with $n = 4$, $l = 2$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} U_1 & \\ & U_2 \end{bmatrix} \begin{bmatrix} C & -S \\ S & C \end{bmatrix} \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}^T = UGV^T$$

C, S diagonal, , $C^2 + S^2 = I$, U_1, U_2, V_1, V_2 orthogonal.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U
- Then $\tilde{U} = UD_U$ has nonnegative diagonal, so $\tilde{U} = R_1 R_2$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U
- Then $\tilde{U} = UD_U$ has nonnegative diagonal, so $\tilde{U} = R_1 R_2$
- R_1 and R_2 are plane rotations with angles in $[-\pi/2, \pi/2]$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U
- Then $\tilde{U} = UD_U$ has nonnegative diagonal, so $\tilde{U} = R_1 R_2$
- R_1 and R_2 are plane rotations with angles in $[-\pi/2, \pi/2]$
- Then $\tilde{V} = D_V V D_U$ has nonnegative diagonal, so $\tilde{V}^T = R_5 R_6$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U
- Then $\tilde{U} = UD_U$ has nonnegative diagonal, so $\tilde{U} = R_1 R_2$
- R_1 and R_2 are plane rotations with angles in $[-\pi/2, \pi/2]$
- Then $\tilde{V} = D_V V D_U$ has nonnegative diagonal, so $\tilde{V}^T = R_5 R_6$
- R_5 and R_6 are plane rotations with angles in $[-\pi/2, \pi/2]$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

- Let D_U, D_V be special 4×4 diagonal matrices of signs
- Let D_U contain the signs of the diagonal elements of U
- Let D_V contain the signs of the diagonal elements of VD_U
- Then $\tilde{U} = UD_U$ has nonnegative diagonal, so $\tilde{U} = R_1 R_2$
- R_1 and R_2 are plane rotations with angles in $[-\pi/2, \pi/2]$
- Then $\tilde{V} = D_V V D_U$ has nonnegative diagonal, so $\tilde{V}^T = R_5 R_6$
- R_5 and R_6 are plane rotations with angles in $[-\pi/2, \pi/2]$
- R_1 and R_5 (R_2 and R_6) rotate in the $(1, 2)$-plane ($(3, 4)$-plane)

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Note that

$$Q = UGV^T = UD_U(D_U G D_U)(D_V V D_U)^T D_V = \tilde{U}\tilde{G}\tilde{V}^T D_V$$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Note that

$$Q = UGV^T = UD_U(D_U G D_U)(D_V V D_U)^T D_V = \tilde{U}\tilde{G}\tilde{V}^T D_V$$

Note also that

$$\tilde{G} = \begin{bmatrix} C & -D_1 S D_2 \\ D_2 S D_1 & C \end{bmatrix} = R_3 R_4$$

The plane rotations R_3 and R_4 rotate in the $(1,3)$ and $(2,4)$ planes with angles in $[-\pi/2, \pi/2]$.

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Note that

$$Q = UGV^T = UD_U(D_U G D_U)(D_V V D_U)^T D_V = \tilde{U}\tilde{G}\tilde{V}^T D_V$$

Note also that

$$\tilde{G} = \begin{bmatrix} C & -D_1 S D_2 \\ D_2 S D_1 & C \end{bmatrix} = R_3 R_4$$

The plane rotations R_3 and R_4 rotate in the $(1,3)$ and $(2,4)$ planes with angles in $[-\pi/2, \pi/2]$.

This way we have obtained

$$Q = R_1 R_2 R_3 R_4 R_5 R_6 D_V.$$

Hence

$$\begin{aligned} \Lambda = Q^T A Q &= D_V R_6^T R_5^T R_4^T R_3^T R_2^T R_1^T A R_1 R_2 R_3 R_4 R_5 R_6 D_V \\ &= R_6^T R_5^T R_4^T R_3^T R_2^T R_1^T A R_1 R_2 R_3 R_4 R_5 R_6. \end{aligned}$$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4)(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4)(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4)(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- Since R_1 and R_2 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4)(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- Since R_1 and R_2 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- So, the product $R_3 R_4$ has the task to zero A_{12}

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5R_6)^T(R_3R_4)^T(R_1R_2)^T A(R_1R_2)(R_3R_4)(R_5R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- Since R_1 and R_2 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- So, the product R_3R_4 has the task to zero A_{12}
- Hence the task of R_5 and R_6 is to annihilate $(1,2)$- and $(3,4)$-elements

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)\,(R_3 R_4)\,(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- Since R_1 and R_2 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- So, the product $R_3 R_4$ has the task to zero A_{12}
- Hence the task of R_5 and R_6 is to annihilate $(1,2)$- and $(3,4)$-elements
- Therefore, we can choose R_5 and R_6 as simple Jacobi rotations

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

Recall

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

$$\Lambda = (R_5 R_6)^T (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4)(R_5 R_6).$$

The rotations within each parenthesis commute. We can conclude:

- Since R_5 and R_6 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- Since R_1 and R_2 rotate in $(1,2)$ and $(3,4)$ planes they cannot change $\|A_{12}\|_F$
- So, the product $R_3 R_4$ has the task to zero A_{12}
- Hence the task of R_5 and R_6 is to annihilate $(1,2)$- and $(3,4)$-elements
- Therefore, we can choose R_5 and R_6 as simple Jacobi rotations
- The role of R_1 and R_2 is to prepare the matrix, in such a way that R_3 and R_4 can accomplish their task

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

- Since R_3 does not affect/change a_{24}, R_4 must be Jacobi rotation

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

- Since R_3 does not affect/change a_{24}, R_4 must be Jacobi rotation
- Since R_4 does not affect/change a_{13}, R_3 must be Jacobi rotation

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

- Since R_3 does not affect/change a_{24}, R_4 must be Jacobi rotation
- Since R_4 does not affect/change a_{13}, R_3 must be Jacobi rotation
- Since $R_5 R_6$ annihilates a_{14} and a_{23} R_5 or/and R_6 must be Givens rotation(s)

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array}\right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

- Since R_3 does not affect/change a_{24}, R_4 must be Jacobi rotation
- Since R_4 does not affect/change a_{13}, R_3 must be Jacobi rotation
- Since $R_5 R_6$ annihilates a_{14} and a_{23} R_5 or/and R_6 must be Givens rotation(s)
- Since $R_3 R_4 = R_4 R_3$ the angle of R_3 (R_4) does not depend on that of R_4 (R_3)

How to Diagonalize 4x4 Symmetric Matrix by 6 Rotations

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{array} \right] = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ \hline a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{array} \right],$$

Let us rename

$$A \leftarrow (R_3 R_4)^T (R_1 R_2)^T A (R_1 R_2)(R_3 R_4).$$

- Since R_3 does not affect/change a_{24}, R_4 must be Jacobi rotation
- Since R_4 does not affect/change a_{13}, R_3 must be Jacobi rotation
- Since $R_5 R_6$ annihilates a_{14} and a_{23} R_5 or/and R_6 must be Givens rotation(s)
- Since $R_3 R_4 = R_4 R_3$ the angle of R_3 (R_4) does not depend on that of R_4 (R_3)
- Once we know R_1 and R_2, we are done since all later rotations are Jacobi rotations. However, for R_3 and R_4 we must alow for the larger interval $[-\pi/2, \pi/2]$ for the angles.

Numerical Experiment in MATLAB

We have made a simple experiment in MATLAB to check the role of the rotations R_1—R_6. We shall explain the details by describing the steps of the experiment.

Numerical Experiment in MATLAB

We have made a simple experiment in MATLAB to check the role of the rotations R_1—R_6. We shall explain the details by describing the steps of the experiment.

- Everything starts with the symmetric matrix A,

$$A = \begin{bmatrix} 0 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 8 \end{bmatrix}$$

Numerical Experiment in MATLAB

We have made a simple experiment in MATLAB to check the role of the rotations R_1—R_6. We shall explain the details by describing the steps of the experiment.

- Everything starts with the symmetric matrix A,

$$A = \begin{bmatrix} 0 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 8 \end{bmatrix}$$

- Using the commands: `digits(dg);` $AA = \mathtt{vpa}(\mathtt{sym}(A,\text{'f'}))$
  A is converted to symbolic type, so we can apply to it variable precision arithmetic (`vpa`). We set `dg`$= 80$.

Numerical Experiment in MATLAB

We have made a simple experiment in MATLAB to check the role of the rotations R_1—R_6. We shall explain the details by describing the steps of the experiment.

- Everything starts with the symmetric matrix A,

$$A = \begin{bmatrix} 0 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 8 \end{bmatrix}$$

- Using the commands: `digits(dg);` $AA = \mathtt{vpa}(\mathtt{sym}(A,\text{'f'}))$ A is converted to symbolic type, so we can apply to it variable precision arithmetic (`vpa`). We set `dg`$= 80$.
- Using the command $[Qv, Dv] = \mathrm{eig}(AA)$; the eigenvector matrix Qv and the eigenvalue diagonal matrix Dv are computed in vpa with 80 decimal digits

Numerical Experiment in MATLAB

Numerical Experiment in MATLAB

- Using the commands: $D = \text{double}(Dv)$; $Q = \text{double}(Qv)$; these `vpa` matrices are converted to double. Here are Q and D:

$$\left[\begin{array}{cc|cc} 2.931093180703362\text{e-}01 & -8.798579328718921\text{e-}02 & -2.278262800852170\text{e-}01 & 9.243595696169855\text{e-}01 \\ 4.193767196696980\text{e-}01 & 1.911371496763030\text{e-}01 & 8.815256669327826\text{e-}01 & 1.024805130325522\text{e-}01 \\ \hline 5.289096009714916\text{e-}01 & 7.298536860359306\text{e-}01 & -3.873541839913629\text{e-}01 & -1.937136214217473\text{e-}01 \\ 6.771002353120279\text{e-}01 & -6.504142427507317\text{e-}01 & -1.447909554690162\text{e-}01 & -3.123013983025982\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{rr|rr} 1.751525068292106\text{e-}01 & 0 & 0 & 0 \\ 0 & 3.389366899857361\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & -9.579649901340663\text{e-}02 & 0 \\ 0 & 0 & 0 & -1.758390873893391\text{e+}00 \end{array}\right]$$

Numerical Experiment in MATLAB

- Using the commands: $D = \text{double}(Dv)$; $Q = \text{double}(Qv)$; these `vpa` matrices are converted to double. Here are Q and D:

$$\left[\begin{array}{cc|cc} 2.931093180703362\text{e-}01 & -8.798579328718921\text{e-}02 & -2.278262800852170\text{e-}01 & 9.243595696169855\text{e-}01 \\ 4.193767196696980\text{e-}01 & 1.911371496763030\text{e-}01 & 8.815256669327826\text{e-}01 & 1.024805130325522\text{e-}01 \\ \hline 5.289096009714916\text{e-}01 & 7.298536860359306\text{e-}01 & -3.873541839913629\text{e-}01 & -1.937136214217473\text{e-}01 \\ 6.771002353120279\text{e-}01 & -6.504142427507317\text{e-}01 & -1.447909554690162\text{e-}01 & -3.123013983025982\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{cc|cc} 1.751525068292106\text{e-}01 & 0 & 0 & 0 \\ 0 & 3.389366899857361\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & -9.579649901340663\text{e-}02 & 0 \\ 0 & 0 & 0 & -1.758390873893391\text{e+}00 \end{array}\right]$$

- Thus Q is the eigenvector matrix for A which is accurate to the last decimal digit

Numerical Experiment in MATLAB

- Using the commands: $D = \text{double}(Dv)$; $Q = \text{double}(Qv)$; these `vpa` matrices are converted to double. Here are Q and D:

$$\left[\begin{array}{cc|cc} 2.931093180703362\text{e-}01 & -8.798579328718921\text{e-}02 & -2.278262800852170\text{e-}01 & 9.243595696169855\text{e-}01 \\ 4.193767196696980\text{e-}01 & 1.911371496763030\text{e-}01 & 8.815256669327826\text{e-}01 & 1.024805130325522\text{e-}01 \\ \hline 5.289096009714916\text{e-}01 & 7.298536860359306\text{e-}01 & -3.873541839913629\text{e-}01 & -1.937136214217473\text{e-}01 \\ 6.771002353120279\text{e-}01 & -6.504142427507317\text{e-}01 & -1.447909554690162\text{e-}01 & -3.123013983025982\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{rr|rr} 1.751525068292106\text{e-}01 & 0 & 0 & 0 \\ 0 & 3.389366899857361\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & -9.579649901340663\text{e-}02 & 0 \\ 0 & 0 & 0 & -1.758390873893391\text{e+}00 \end{array}\right]$$

- Thus Q is the eigenvector matrix for A which is accurate to the last decimal digit
- Indeed, we have checked the orthogonality of Q:

$$\|Q^T Q - I_4\|_2 = 2.470193220159530\text{e-}16, \quad \|QQ^T - I_4\|_2 = 2.489175293808355\text{e-}16$$

Numerical Experiment in MATLAB

- Using the commands: $D = \text{double}(Dv)$; $Q = \text{double}(Qv)$; these `vpa` matrices are converted to double. Here are Q and D:

$$\left[\begin{array}{cc|cc} 2.931093180703362\text{e-}01 & -8.798579328718921\text{e-}02 & -2.278262800852170\text{e-}01 & 9.243595696169855\text{e-}01 \\ 4.193767196696980\text{e-}01 & 1.911371496763030\text{e-}01 & 8.815256669327826\text{e-}01 & 1.024805130325522\text{e-}01 \\ \hline 5.289096009714916\text{e-}01 & 7.298536860359306\text{e-}01 & -3.873541839913629\text{e-}01 & -1.937136214217473\text{e-}01 \\ 6.771002353120279\text{e-}01 & -6.504142427507317\text{e-}01 & -1.447909554690162\text{e-}01 & -3.123013983025982\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{cc|cc} 1.751525068292106\text{e-}01 & 0 & 0 & 0 \\ 0 & 3.389366899857361\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & -9.579649901340663\text{e-}02 & 0 \\ 0 & 0 & 0 & -1.758390873893391\text{e+}00 \end{array}\right]$$

- Thus Q is the eigenvector matrix for A which is accurate to the last decimal digit
- Indeed, we have checked the orthogonality of Q:

$$\|Q^TQ - I_4\|_2 = 2.470193220159530\text{e-}16, \quad \|QQ^T - I_4\|_2 = 2.489175293808355\text{e-}16$$

- The remaining computation is performed in double

Numerical Experiment in MATLAB

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

Next, we compute the matrix W. First compute the SVDs of the diagonal blocks

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

Numerical Experiment in MATLAB

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

Next, we compute the matrix W. First compute the SVDs of the diagonal blocks

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

Then form the block diagonal U and V

$$U = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix}.$$

Numerical Experiment in MATLAB

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

Next, we compute the matrix W. First compute the SVDs of the diagonal blocks

$$Q_{11} = U_{11} C_1 V_{11}^T, \qquad Q_{22} = U_{22} C_2 V_{22}^T$$

Then form the block diagonal U and V

$$U = \begin{bmatrix} U_{11} & O \\ O & U_{22} \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix}.$$

and also

$$W = U^T Q V = \begin{bmatrix} C_1 & W_{12} \\ W_{21} & C_2 \end{bmatrix}, \tag{5}$$

where

$$W_{12} = U_{11}^T Q_{12} V_{22}, \qquad W_{21} = U_{22}^T Q_{21} V_{11}.$$

and C_1 and C_2 are diagonal.

Numerical Experiment in MATLAB

Numerical Experiment in MATLAB

- We have obtained: W is to working accuracy G, the central matrix in CSD!

Numerical Experiment in MATLAB

- We have obtained: W is to working accuracy G, the central matrix in CSD!
- Then we employ the procedure which transforms 2×2 reflectors into rotations with angles from the interval $[-\pi/2, \pi/2]$

Numerical Experiment in MATLAB

- We have obtained: W is to working accuracy G, the central matrix in CSD!
- Then we employ the procedure which transforms 2×2 reflectors into rotations with angles from the interval $[-\pi/2, \pi/2]$
- This makes the decomposition $Q = \tilde{U}\tilde{G}\tilde{V}D_V$. Here are $\tilde{U}$, $\tilde{G}$, $\tilde{V}$, D_V:

$$\left[\begin{array}{rr|rr} 5.057945141284034\text{e-}01 & -8.626539917473362\text{e-}01 & 0 & 0 \\ 8.626539917473362\text{e-}01 & 5.057945141284033\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & 8.012598003675062\text{e-}01 & -5.983165820157615\text{e-}01 \\ 0 & 0 & 5.983165820157615\text{e-}01 & 8.012598003675064\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{rr|rr} 5.240444959932522\text{e-}01 & 0 & -8.516908865422823\text{e-}01 & -1.110223024625157\text{e-}16 \\ 2.775557561562891\text{e-}17 & 1.773194331327221\text{e-}01 & -5.551115123125783\text{e-}17 & 9.841533511772902\text{e-}01 \\ \hline 8.516908865422820\text{e-}01 & -1.387778780781446\text{e-}16 & 5.240444959932518\text{e-}01 & 0 \\ 5.551115123125783\text{e-}17 & -9.841533511772899\text{e-}01 & 1.387778780781446\text{e-}17 & 1.773194331327220\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{rr|rr} 9.732572143939268\text{e-}01 & -2.297180764114445\text{e-}01 & 0 & 0 \\ 2.297180764114445\text{e-}01 & 9.732572143939268\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & 7.575733907232336\text{e-}01 & -6.527499962988148\text{e-}01 \\ 0 & 0 & 6.527499962988148\text{e-}01 & 7.575733907232336\text{e-}01 \end{array}\right]$$

$$\left[\begin{array}{rr|rr} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{array}\right]$$

Numerical Experiment in MATLAB

Numerical Experiment in MATLAB

- Now is the time to "extract" the 6 rotations from the CSD

Numerical Experiment in MATLAB

- Now is the time to “extract” the 6 rotations from the CSD
- To ensure the best orthogonality of the rotations we make a procedure that takes the mean of two diagonal elements and similar for the off-diagonal elements. So, each “updated rotation” will have equal diagonal and opposite off-diagonal elements

Numerical Experiment in MATLAB

- Now is the time to "extract" the 6 rotations from the CSD
- To ensure the best orthogonality of the rotations we make a procedure that takes the mean of two diagonal elements and similar for the off-diagonal elements. So, each "updated rotation" will have equal diagonal and opposite off-diagonal elements
- We have checked
  $\|Q - R_q R_2 R_3 R_4 R_5 R_6 D_V\|_2 = 3.885944763273228e-16$

Numerical Experiment in MATLAB

- Now is the time to "extract" the 6 rotations from the CSD
- To ensure the best orthogonality of the rotations we make a procedure that takes the mean of two diagonal elements and similar for the off-diagonal elements. So, each "updated rotation" will have equal diagonal and opposite off-diagonal elements
- We have checked $\|Q - R_q R_2 R_3 R_4 R_5 R_6 D_V\|_2 = 3.885944763273228e - 16$
- Next, we successively apply these 6 rotations R_1–R_6 to the symmetric matrix A. We shall have 6 steps to display, each one of the form

$$A^{(k)} = R_k^T A^{(k-1)} R_k, \quad k = 1, 2, 3, 4, 5, 6, \quad A^{(0)} = A$$

Step 1 $\quad A^{(1)} = R_1^T A R_1$

$$A = \left[\begin{array}{cc|cc} 0 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ \hline 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 8 \end{array}\right]$$

$$R_1 = \left[\begin{array}{rr|cc} 5.057945141284034\text{e-}01 & -8.626539917473362\text{e-}01 & 0 & 0 \\ 8.626539917473362\text{e-}01 & 5.057945141284033\text{e-}01 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

$$A^{(1)} = R_1^T A R_1$$

$$\left[\begin{array}{rr|rr} 3.977818354899925\text{e}{+}00 & 3.322893319398632\text{e-}01 & 4.967999509374554\text{e}{+}00 & 6.336448015250294\text{e}{+}00 \\ 3.322893319398633\text{e-}01 & -9.778183548999233\text{e-}01 & -5.647839187283954\text{e-}01 & -9.216433963473283\text{e-}01 \\ \hline 4.967999509374554\text{e}{+}00 & -5.647839187283954\text{e-}01 & 5.000000000000000\text{e}{+}00 & 6.000000000000000\text{e}{+}00 \\ 6.336448015250294\text{e}{+}00 & -9.216433963473283\text{e-}01 & 6.000000000000000\text{e}{+}00 & 8.000000000000000\text{e}{+}00 \end{array}\right]$$

Step 2, $\quad A^{(2)} = R_2^T A^{(1)} R_2$

$$A^{(1)}$$

$$\left[\begin{array}{cc|cc}
3.977818354899925e{+}00 & 3.322893319398632e{-}01 & 4.967999509374554e{+}00 & 6.336448015250294e{+}00 \\
3.322893319398633e{-}01 & -9.778183548999233e{-}01 & -5.647839187283954e{-}01 & -9.216433963473283e{-}01 \\
\hline
4.967999509374554e{+}00 & -5.647839187283954e{-}01 & 5.000000000000000e{+}00 & 6.000000000000000e{+}00 \\
6.336448015250294e{+}00 & -9.216433963473283e{-}01 & 6.000000000000000e{+}00 & 8.000000000000000e{+}00
\end{array}\right]$$

$$R_2 = \left[\begin{array}{cc|cc}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 8.012598003675062e{-}01 & -5.983165820157615e{-}01 \\
0 & 0 & 5.983165820157615e{-}01 & 8.012598003675064e{-}01
\end{array}\right]$$

$$A^{(2)} = R_2^T A^{(1)} R_2$$

$$\left[\begin{array}{cc|cc}
3.977818354899925e{+}00 & 3.322893319398632e{-}01 & 7.771860213712436e{+}00 & 2.104704585833569e{+}00 \\
3.322893319398633e{-}01 & -9.778183548999233e{-}01 & -1.003973176711023e{+}00 & -4.005562199362495e{-}01 \\
\hline
7.771860213712436e{+}00 & -1.003973176711023e{+}00 & 1.182683249769528e{+}01 & 3.142428287407269e{+}00 \\
2.104704585833569e{+}00 & -4.005562199362495e{-}01 & 3.142428287407268e{+}00 & 1.173167502304712e{+}00
\end{array}\right]$$

Step 3, $A^{(3)} = R_3^T A^{(2)} R_3$

$$A^{(2)}$$

$$\left[\begin{array}{rr|rr}
3.977818354899925\text{e}{+}00 & 3.322893319398632\text{e-}01 & 7.771860213712436\text{e}{+}00 & 2.104704585833569\text{e}{+}00 \\
3.322893319398633\text{e-}01 & -9.778183548999233\text{e-}01 & -1.003973176711023\text{e}{+}00 & -4.005562199362495\text{e-}01 \\
\hline
7.771860213712436\text{e}{+}00 & -1.003973176711023\text{e}{+}00 & 1.182683249769528\text{e}{+}01 & 3.142428287407269\text{e}{+}00 \\
2.104704585833569\text{e}{+}00 & -4.005562199362495\text{e-}01 & 3.142428287407268\text{e}{+}00 & 1.173167502304712\text{e}{+}00
\end{array}\right]$$

$$R_3 = \left[\begin{array}{rr|rr}
5.240444959932520\text{e-}01 & 0 & -8.516908865422821\text{e-}01 & 0 \\
0 & 1 & 0 & 0 \\
\hline
8.516908865422821\text{e-}01 & 0 & 5.240444959932520\text{e-}01 & 0 \\
0 & 0 & 0 & 1
\end{array}\right]$$

$$A^{(3)} = R_3^T A^{(2)} R_3$$

$$\left[\begin{array}{rr|rr}
1.660884981522208\text{e}{+}01 & -6.809404094573227\text{e-}01 & -3.552713678800501\text{e-}15 & 3.779336387895281\text{e}{+}00 \\
-6.809404094573226\text{e-}01 & -9.778183548999233\text{e-}01 & -8.091344130886772\text{e-}01 & -4.005562199362495\text{e-}01 \\
\hline
-2.664535259100376\text{e-}15 & -8.091344130886771\text{e-}01 & -8.041989626268715\text{e-}01 & -1.457854665489189\text{e-}01 \\
3.779336387895281\text{e}{+}00 & -4.005562199362495\text{e-}01 & -1.457854665489193\text{e-}01 & 1.173167502304712\text{e}{+}00
\end{array}\right]$$

Step 4, $\qquad A^{(4)} = R_4^T A^{(3)} R_4$

$$A^{(3)}$$

$$\left[\begin{array}{rr|rr}
1.660884981522208\text{e}{+}01 & -6.809404094573227\text{e-}01 & -3.552713678800501\text{e-}15 & 3.779336387895281\text{e}{+}00 \\
-6.809404094573226\text{e-}01 & -9.778183548999233\text{e-}01 & -8.091344130886772\text{e-}01 & -4.005562199362495\text{e-}01 \\
\hline
-2.664535259100376\text{e-}15 & -8.091344130886771\text{e-}01 & -8.041989626268715\text{e-}01 & -1.457854665489189\text{e-}01 \\
3.779336387895281\text{e}{+}00 & -4.005562199362495\text{e-}01 & -1.457854665489193\text{e-}01 & 1.173167502304712\text{e}{+}00
\end{array}\right]$$

$$R_4 = \left[\begin{array}{lr|lr}
1 & 0 & 0 & 0 \\
0 & 1.773194331327221\text{e-}01 & 0 & 9.841533511772900\text{e-}01 \\
\hline
0 & 0 & 1 & 0 \\
0 & -9.841533511772900\text{e-}01 & 0 & 1.773194331327221\text{e-}01
\end{array}\right]$$

$$A^{(4)} = R_4^T A^{(3)} R_4$$

$$\left[\begin{array}{rr|rr}
1.660884981522208\text{e}{+}01 & -3.840190538775552\text{e}{+}00 & -3.552713678800501\text{e-}15 & 5.551115123125783\text{e-}16 \\
-3.840190538775552\text{e}{+}00 & 1.245337557684717\text{e}{+}00 & 1.637578961322106\text{e-}15 & -2.775557561562891\text{e-}16 \\
\hline
-2.664535259100376\text{e-}15 & 1.221245327087672\text{e-}15 & -8.041989626268715\text{e-}01 & -8.221629404815349\text{e-}01 \\
6.661338147750939\text{e-}15 & -2.498001805406602\text{e-}16 & -8.221629404815352\text{e-}01 & -1.049988410279928\text{e}{+}00
\end{array}\right]$$

Step 5, $A^{(5)} = R_5^T A^{(4)} R_5$

$$A^{(4)}$$

$$\left[\begin{array}{rr|rr}
1.660884981522208\text{e}{+}01 & -3.840190538775552\text{e}{+}00 & -3.552713678800501\text{e-}15 & 5.551115123125783\text{e-}16 \\
-3.840190538775552\text{e}{+}00 & 1.245337557684717\text{e}{+}00 & 1.637578961322106\text{e-}15 & -2.775557561562891\text{e-}16 \\
\hline
-2.664535259100376\text{e-}15 & 1.221245327087672\text{e-}15 & -8.041989626268715\text{e-}01 & -8.221629404815349\text{e-}01 \\
6.661338147750939\text{e-}15 & -2.498001805406602\text{e-}16 & -8.221629404815352\text{e-}01 & -1.049988410279928\text{e}{+}00
\end{array}\right]$$

$$R_5 = \left[\begin{array}{rr|rr}
9.732572143939268e-01 & 2.297180764114445e-01 & 0 & 0 \\
-2.297180764114445e-01 & 9.732572143939268e-01 & 0 & 0 \\
\hline
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right]$$

$$A^{(5)} = R_5^T A^{(4)} R_5$$

$$\left[\begin{array}{rr|rr}
1.751525068292106\text{e}{+}01 & 4.440892098500626\text{e-}16 & -3.833885707535341\text{e-}15 & 6.040258585524866\text{e-}16 \\
5.412337245047638\text{e-}16 & 3.389366899857359\text{e-}01 & 7.776629859117751\text{e-}16 & -1.426139932733765\text{e-}16 \\
\hline
-2.873820291291477\text{e-}15 & 5.764939108819395\text{e-}16 & -8.041989626268715\text{e-}01 & -8.221629404815349\text{e-}01 \\
7.057031579426399\text{e-}16 & -9.009684930535088\text{e-}17 & -8.221629404815352\text{e-}01 & -1.049988410279928\text{e}{+}00
\end{array}\right]$$

Step 6, $A^{(6)} = R_6^T A^{(5)} R_6$

$$A^{(5)}$$

$$\left[\begin{array}{rr|rr}
1.751525068292106\text{e}{+}01 & 4.440892098500626\text{e-}16 & -3.833885707535341\text{e-}15 & 6.040258585524866\text{e-}16 \\
5.412337245047638\text{e-}16 & 3.389366899857359\text{e-}01 & 7.776629859117751\text{e-}16 & -1.426139932733765\text{e-}16 \\
\hline
-2.873820291291477\text{e-}15 & 5.764939108819395\text{e-}16 & -8.041989626268715\text{e-}01 & -8.221629404815349\text{e-}01 \\
7.057031579426399\text{e-}16 & -9.009684930535088\text{e-}17 & -8.221629404815352\text{e-}01 & -1.049988410279928\text{e}{+}00
\end{array}\right]$$

$$R_6 = \left[\begin{array}{cc|rr}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\hline
0 & 0 & 7.575733907232336\text{e-}01 & 6.527499962988148\text{e-}01 \\
0 & 0 & -6.527499962988148\text{e-}01 & 7.575733907232336\text{e-}01
\end{array}\right]$$

$$A^{(6)} = R_6^T A^{(5)} R_6$$

$$\left[\begin{array}{rr|rr}
1.751525068292106\text{e}{+}01 & 4.440892098500626\text{e-}16 & -3.298727672037416\text{e-}15 & -2.044974963655654\text{e-}15 \\
5.412337245047638\text{e-}16 & 3.389366899857359\text{e-}01 & 6.822280686584934\text{e-}16 & 3.995789447269441\text{e-}16 \\
\hline
-2.637777516138035\text{e-}15 & 4.955471648487312\text{e-}16 & -9.579649901340658\text{e-}02 & 1.526556658859590\text{e-}16 \\
-1.341264250297271\text{e-}15 & 3.080514225727404\text{e-}16 & -2.220446049250313\text{e-}16 & -1.758390873893392e + 00
\end{array}\right]$$