Diagonalization Methods for Solving Definite Generalized Eigenvalue Problem

Vjeran Hari

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

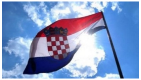
SIAM Annual Meeting
July 09–13, 2018, Portland, Oregon, USA

OUTLINE

OUTLINE

- GEP (DGEP, PGEP)

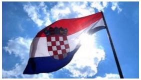
This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.



OUTLINE

- GEP (DGEP, PGEP)
- Derivation of Algorithms (real and complex)

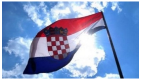
This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.



OUTLINE

- GEP (DGEP, PGEP)
- Derivation of Algorithms (real and complex)
- Properties (convergence, global and asymptotic),

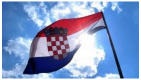
This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.



OUTLINE

- GEP (DGEP, PGEP)
- Derivation of Algorithms (real and complex)
- Properties (convergence, global and asymptotic),
- Stability (HRA: High Relative Accuracy)

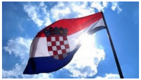
This work has been fully supported by Croatian Science Foundation under the project IP-09-2014-3670.



OUTLINE

- GEP (DGEP, PGEP)
- Derivation of Algorithms (real and complex)
- Properties (convergence, global and asymptotic),
- Stability (HRA: High Relative Accuracy)
- Block algorithms
- Global convergence of block algorithms

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



Hrvatska zaklada za znanost

Why Element-wise, Two-sided Jacobi Methods?

Why Element-wise, Two-sided Jacobi Methods?

- They can be used standalone or as kernel algorithms in block methods

Why Element-wise, Two-sided Jacobi Methods?

- They can be used standalone or as kernel algorithms in block methods
- As basic algorithms they can be "upgraded" to one-sided algorithms

Why Element-wise, Two-sided Jacobi Methods?

- They can be used standalone or as kernel algorithms in block methods
- As basic algorithms they can be "upgraded" to one-sided algorithms
- The theoretical aspects of one-sided methods can be better analysed and understood if they are considered/imagined as two-sided methods

Why Element-wise, Two-sided Jacobi Methods?

- They can be used standalone or as kernel algorithms in block methods
- As basic algorithms they can be "upgraded" to one-sided algorithms
- The theoretical aspects of one-sided methods can be better analysed and understood if they are considered/imagined as two-sided methods
- One-sided methods have problem with terminating the process. Stopping of the process can be costly, especially if the matrix dimension n is large.

Why Element-wise, Two-sided Jacobi Methods?

- They can be used standalone or as kernel algorithms in block methods
- As basic algorithms they can be "upgraded" to one-sided algorithms
- The theoretical aspects of one-sided methods can be better analysed and understood if they are considered/imagined as two-sided methods
- One-sided methods have problem with terminating the process. Stopping of the process can be costly, especially if the matrix dimension n is large.
- Two sided methods can smoothly, timely and cost effectively stop the process.

GEP, DGEP and PGEP

Let $A = A^*, \quad B = B^*.$

GEP, DGEP and PGEP

Let $A = A^*, \quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

GEP, DGEP and PGEP

Let $A = A^*, \quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

GEP, DGEP and PGEP

Let $A = A^*$, $\quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

If $\alpha A + \beta B \succ O$, $\alpha, \beta \in \mathbf{R}$, GEP is called Definite GEP(DGEP)

GEP, DGEP and PGEP

Let $A = A^*, \quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

If $\alpha A + \beta B \succ O$, $\alpha, \beta \in \mathbf{R}$, GEP is called Definite GEP(DGEP)
then (A, B) is called definite pair

GEP, DGEP and PGEP

Let $A = A^*$, $\quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

If $\alpha A + \beta B \succ O$, $\alpha, \beta \in \mathbf{R}$, GEP is called Definite GEP(DGEP)
then (A, B) is called definite pair

For a definite pair (A, B) there exists a nonsingular matrix F such that

$$F^* A F = \Lambda_A = \text{diag}(\alpha_1, \ldots, \alpha_n)\,, \quad F^* B F = \Lambda_B = \text{diag}(\beta_1, \ldots, \beta_n),$$

GEP, DGEP and PGEP

Let $A = A^*$, $B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

If $\alpha A + \beta B \succ O$, $\alpha, \beta \in \mathbf{R}$, GEP is called Definite GEP(DGEP)
then (A, B) is called definite pair

For a definite pair (A, B) there exists a nonsingular matrix F such that

$$F^* AF = \Lambda_A = \text{diag}(\alpha_1, \ldots, \alpha_n), \quad F^* BF = \Lambda_B = \text{diag}(\beta_1, \ldots, \beta_n),$$

The eigenpairs of (A, B) are: $(\alpha_i/\beta_i, Fe_i)$, $1 \leq i \leq n$;

GEP, DGEP and PGEP

Let $A = A^*, \quad B = B^*$.
We consider the Generalized Eigenvalue Problem (GEP)

$$Ax = \lambda Bx, \quad x \neq 0.$$

If $B \succ O$, GEP is usually called Positive definite GEP (PGEP).

If $\alpha A + \beta B \succ O$, $\alpha, \beta \in \mathbf{R}$, GEP is called Definite GEP(DGEP)
then (A, B) is called definite pair

For a definite pair (A, B) there exists a nonsingular matrix F such that

$$F^* AF = \Lambda_A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n), \quad F^* BF = \Lambda_B = \mathrm{diag}(\beta_1, \ldots, \beta_n),$$

The eigenpairs of (A, B) are: $(\alpha_i/\beta_i, Fe_i)$, $1 \leq i \leq n$;
here $I_n = [e_1, \ldots, e_n]$.

Why are Element-wise Methods Important?

Why are Element-wise Methods Important?

On contemporary GPU and CPU parallel computing machines, probably the best methods for solving full DGEP and PGEP are block diagonalization methods.

Why are Element-wise Methods Important?

On contemporary GPU and CPU parallel computing machines, probably the best methods for solving full DGEP and PGEP are block diagonalization methods.

They use kernel algorithms to perform at each step an intrinsic job - solving PGEP or DGEP with much smaller matrices (say, $n = 32 - 512$)

Why are Element-wise Methods Important?

On contemporary GPU and CPU parallel computing machines, probably the best methods for solving full DGEP and PGEP are block diagonalization methods.

They use kernel algorithms to perform at each step an intrinsic job - solving PGEP or DGEP with much smaller matrices (say, $n = 32 - 512$)

The block method will function well only if the kernel algorithm if globally convergent, fast and accurate.

Why are Element-wise Methods Important?

On contemporary GPU and CPU parallel computing machines, probably the best methods for solving full DGEP and PGEP are block diagonalization methods.

They use kernel algorithms to perform at each step an intrinsic job - solving PGEP or DGEP with much smaller matrices (say, $n = 32 - 512$)

The block method will function well only if the kernel algorithm if globally convergent, fast and accurate.

Most of the time, the kernel algorithm will operate on nearly diagonal matrices. On such matrices, the element-wise diagonalization methods are fast and highly accurate.

Why are Element-wise Methods Important?

On contemporary GPU and CPU parallel computing machines, probably the best methods for solving full DGEP and PGEP are block diagonalization methods.

They use kernel algorithms to perform at each step an intrinsic job - solving PGEP or DGEP with much smaller matrices (say, $n = 32 - 512$)

The block method will function well only if the kernel algorithm if globally convergent, fast and accurate.

Most of the time, the kernel algorithm will operate on nearly diagonal matrices. On such matrices, the element-wise diagonalization methods are fast and highly accurate.

Hence, probably the best choice for the kernel algorithm is some element-wise diagonalization method.

Jacobi Methods for DGEP and PGEP

So far, we know three "promising" real diagonalization methods:

So far, we know three "promising" real diagonalization methods:

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)

Jacobi Methods for DGEP and PGEP

So far, we know three "promising" real diagonalization methods:

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann method (shorter: HZ method)
  (Numerical Algorithms, to appear)

Jacobi Methods for DGEP and PGEP

So far, we know three "promising" real diagonalization methods:

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann method (shorter: HZ method)
  (Numerical Algorithms, to appear)
- Cholesky-Jacobi method (shorter: CJ method)
  (Numerical Algorithms, to appear)

Jacobi Methods for DGEP and PGEP

So far, we know three "promising" real diagonalization methods:

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann method (shorter: HZ method)
  (Numerical Algorithms, to appear)
- Cholesky-Jacobi method (shorter: CJ method)
  (Numerical Algorithms, to appear)

The methods are connected: the FL method can be viewed as the HZ or CJ method with "fast scaled" transformations.

Jacobi Methods for DGEP and PGEP

So far, we know three "promising" real diagonalization methods:

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann method (shorter: HZ method)
  (Numerical Algorithms, to appear)
- Cholesky-Jacobi method (shorter: CJ method)
  (Numerical Algorithms, to appear)

The methods are connected: the FL method can be viewed as the HZ or CJ method with "fast scaled" transformations.

We have also derived their "equally promising" complex counterparts.

The Real and Complex FL Method

Starting with a definite pair (A, B) of Hermitian matrices, FL generates a sequence of "congruent" matrix pairs

$$(A, B) = (A^{(0)}, B^{(0)}), (A^{(1)}, B^{(1)}), \ldots$$

by the rule

$$A^{(k+1)} = F_k^* A^{(k)} F_k \ , \qquad B^{(k+1)} = F_k^* B^{(k)} F_k \ , \quad k \geq 0.$$

Here F_k is an elementary plane matrix defined by the pivot pair $(i(k), j(k))$

$$F_k = \begin{bmatrix} I & & & & \\ & 1 & & \alpha_k & \\ & & I & & \\ & \beta_k & & 1 & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix} , \qquad \alpha_k, \beta_k \in \mathbf{C},$$

Derivation of the Complex FL Method

The goal is to compute complex numbers α_k, β_k such that the pivot elements $a_{ij}^{(k)}$, $b_{ij}^{(k)}$ of $A^{(k)}$, $B^{(k)}$ are annihilated.

Derivation of the Complex FL Method

The goal is to compute complex numbers α_k, β_k such that the pivot elements $a_{ij}^{(k)}$, $b_{ij}^{(k)}$ of $A^{(k)}$, $B^{(k)}$ are annihilated.

We simplify notation: $A = A^{(k)}$, $A' = A^{(k+1)}$, $F = F_k$, $(i,j) = (i(k), j(k))$.

Pivot submatrices $\hat{A}$, $\hat{B}$, $\hat{F}$ of A, B, F are 2×2 principal submatrices obtained on the intersection of pivot rows and columns i and j.

Derivation of the Complex FL Method

The goal is to compute complex numbers α_k, β_k such that the pivot elements $a_{ij}^{(k)}$, $b_{ij}^{(k)}$ of $A^{(k)}$, $B^{(k)}$ are annihilated.

We simplify notation: $A = A^{(k)}$, $A' = A^{(k+1)}$, $F = F_k$, $(i,j) = (i(k), j(k))$.

Pivot submatrices $\hat{A}$, $\hat{B}$, $\hat{F}$ of A, B, F are 2×2 principal submatrices obtained on the intersection of pivot rows and columns i and j.

We have

$$A' = F^* A F, \quad B' = F^* B F \qquad \left(\hat{A}' = \hat{F}^* \hat{A} \hat{F}, \quad \hat{B}' = \hat{F}^* \hat{B} \hat{F} \right)$$

and F is chosen to obtain $a'_{ij} = 0$ and $b'_{ij} = 0$.

Derivation of the Complex FL Method ($n = 2$)

Derivation of the Complex FL Method ($n = 2$)

The goal is to compute α and β which satisfy the matrix equations

$$\begin{bmatrix} 1 & \bar{\beta} \\ \bar{\alpha} & 1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix} = \begin{bmatrix} a'_{ii} & 0 \\ 0 & a'_{jj} \end{bmatrix}$$

$$\begin{bmatrix} 1 & \bar{\beta} \\ \bar{\alpha} & 1 \end{bmatrix} \begin{bmatrix} b_{ii} & b_{ij} \\ \bar{b}_{ij} & b_{jj} \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix} = \begin{bmatrix} b'_{ii} & 0 \\ 0 & b'_{jj} \end{bmatrix}.$$

Derivation of the Complex FL Method ($n = 2$)

The goal is to compute α and β which satisfy the matrix equations

$$\begin{bmatrix} 1 & \bar{\beta} \\ \bar{\alpha} & 1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix} = \begin{bmatrix} a'_{ii} & 0 \\ 0 & a'_{jj} \end{bmatrix}$$
$$\begin{bmatrix} 1 & \bar{\beta} \\ \bar{\alpha} & 1 \end{bmatrix} \begin{bmatrix} b_{ii} & b_{ij} \\ \bar{b}_{ij} & b_{jj} \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix} = \begin{bmatrix} b'_{ii} & 0 \\ 0 & b'_{jj} \end{bmatrix}.$$

This leads us to solving a system of two nonlinear equations

$$e_1 = a_{ii}\alpha + a_{jj}\bar{\beta} + \bar{a}_{ij}\alpha\bar{\beta} + a_{ij} = 0$$
$$e_2 = b_{ii}\alpha + b_{jj}\bar{\beta} + \bar{b}_{ij}\alpha\bar{\beta} + b_{ij} = 0.$$

Solution if Matrices are Real and Symmetric

$$
\begin{aligned}
\Im_{ii} &= a_{ii}b_{ij} - a_{ij}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im_{jj} &= a_{jj}b_{ij} - a_{ij}b_{jj} = \begin{vmatrix} a_{jj} & b_{jj} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im_{ij} &= a_{ii}b_{jj} - a_{jj}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{jj} & b_{jj} \end{vmatrix} \\
\Im &= \Im_{ij}^2 + 4\Im_{ii}\Im_{jj} \\
\nu &= (\Im_{ij} + \mathrm{sgn}(\Im_{ij})\sqrt{\Im})/2 \\
\alpha &= \Im_j/\nu, \qquad \beta = -\Im_i/\nu
\end{aligned}
$$

Solution if Matrices are Real and Symmetric

$$
\begin{aligned}
\Im_{ii} &= a_{ii}b_{ij} - a_{ij}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im_{jj} &= a_{jj}b_{ij} - a_{ij}b_{jj} = \begin{vmatrix} a_{jj} & b_{jj} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im_{ij} &= a_{ii}b_{jj} - a_{jj}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{jj} & b_{jj} \end{vmatrix} \\
\Im &= \Im_{ij}^2 + 4\Im_{ii}\Im_{jj} \\
\nu &= (\Im_{ij} + \mathrm{sgn}(\Im_{ij})\sqrt{\Im})/2 \\
\alpha &= \Im_j/\nu, \qquad \beta = -\Im_i/\nu
\end{aligned}
$$

If $\begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix}$ and $\begin{bmatrix} b_{ii} & b_{ij} \\ \bar{b}_{ij} & b_{jj} \end{bmatrix}$ are proportional, all $\Im_{ii}$, $\Im_{jj}$, $\Im_{ij}$, $\Im$ and ν are zero and a special algorithm is required. This is the real FL algorithm.

The Complex FL Algorithm

$$
\begin{aligned}
\Im_{ii} &= a_{ii}b_{ij} - a_{ij}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im_{jj} &= a_{jj}b_{ij} - a_{ij}b_{jj} = \begin{vmatrix} a_{jj} & b_{jj} \\ a_{ij} & b_{ij} \end{vmatrix} \\
\Im'_{ij} &= a_{ii}b_{jj} - a_{jj}b_{ii} = \begin{vmatrix} a_{ii} & b_{ii} \\ a_{jj} & b_{jj} \end{vmatrix} \\
\imath\Im''_{ij} &= a_{ij}\bar{b}_{ij} - \bar{a}_{ij}b_{ij} = \begin{vmatrix} a_{ij} & b_{ij} \\ \bar{a}_{ij} & \bar{b}_{ij} \end{vmatrix} = -2\imath \begin{vmatrix} \mathrm{Re}(a_{ij}) & \mathrm{Re}(b_{ij}) \\ \mathrm{Im}(a_{ij}) & \mathrm{Im}(b_{ij}) \end{vmatrix} \\
\Im_{ij} &= \Im'_{ij} + \imath\Im''_{ij} \\
\Im &= \Im_{ij}^2 + 4\bar{\Im}_{ii}\Im_{jj} = (\Im'_{ij})^2 - (\Im''_{ij})^2 + 2\imath\Im'_{ij}\Im''_{ij} + 4\bar{\Im}_{ii}\Im_{jj} \\
\nu &= (\Im_{ij} + \mathrm{sgn}(\Im'_{ij})\sqrt{\Im})/2, \\
\alpha &= \Im_j/\nu, \qquad \beta = -\bar{\Im}_i/\nu
\end{aligned}
$$

The Main Characteristics of the FL Algorithms

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)
- Very accurate (HRA on well-behaved positive definite matrices)

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)
- Very accurate (HRA on well-behaved positive definite matrices)
- Well defined for general definite GEP

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)
- Very accurate (HRA on well-behaved positive definite matrices)
- Well defined for general definite GEP
- Problems with renormalizations ($\|A^{(k)}\| \nearrow \infty$, $\|B^{(k)}\| \nearrow \infty$, $\|F_1 F_2 \cdots F_k\| \nearrow \infty$)

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)
- Very accurate (HRA on well-behaved positive definite matrices)
- Well defined for general definite GEP
- Problems with renormalizations ($\|A^{(k)}\| \nearrow \infty$, $\|B^{(k)}\| \nearrow \infty$, $\|F_1 F_2 \cdots F_k\| \nearrow \infty$)
- Difficult and challenging for making a good numerical code (to many freedoms, all we have $\alpha A + \beta B \succ O$, when to stop iterations?)

The Main Characteristics of the FL Algorithms

- Very fast (SAXPY BLAS1 operations, Fused multiplyadd)
- Very accurate (HRA on well-behaved positive definite matrices)
- Well defined for general definite GEP
- Problems with renormalizations $(\|A^{(k)}\| \nearrow \infty$, $\|B^{(k)}\| \nearrow \infty$, $\|F_1 F_2 \cdots F_k\| \nearrow \infty)$
- Difficult and challenging for making a good numerical code (to many freedoms, all we have $\alpha A + \beta B \succ O$, when to stop iterations?)
- Theoretical results are lacking (all we have is quadratic asymptotic convergence result)

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

$D_0 = [\mathrm{diag}(B)]^{-\frac{1}{2}}$, so that $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

$D_0 = [\mathrm{diag}(B)]^{-\frac{1}{2}}$, so that $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

This property of $B^{(0)}$ is maintained during the iteration process:

$$A^{(k+1)} = Z_k^* A^{(k)} Z_k, \qquad B^{(k+1)} = Z_k^* B^{(k)} Z_k, \quad k \geq 0.$$

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

$D_0 = [\mathrm{diag}(B)]^{-\frac{1}{2}}$, so that $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

This property of $B^{(0)}$ is maintained during the iteration process:

$$A^{(k+1)} = Z_k^* A^{(k)} Z_k, \qquad B^{(k+1)} = Z_k^* B^{(k)} Z_k, \quad k \geq 0.$$

Each Z_k is nonsingular elementary plane matrix

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

$D_0 = [\mathrm{diag}(B)]^{-\frac{1}{2}}$, so that $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

This property of $B^{(0)}$ is maintained during the iteration process:

$$A^{(k+1)} = Z_k^* A^{(k)} Z_k, \qquad B^{(k+1)} = Z_k^* B^{(k)} Z_k, \quad k \geq 0.$$

Each Z_k is nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & * & & * & \\ & & I & & \\ & * & & * & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}$$

Derivation of the Real and Complex HZ Method, $B \succ O$

Preliminary transformation: $A^{(0)} = D_0 A D_0, \quad B^{(0)} = D_0 B D_0$

$D_0 = [\text{diag}(B)]^{-\frac{1}{2}}$, so that $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

This property of $B^{(0)}$ is maintained during the iteration process:

$$A^{(k+1)} = Z_k^* A^{(k)} Z_k, \qquad B^{(k+1)} = Z_k^* B^{(k)} Z_k, \quad k \geq 0.$$

Each Z_k is nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & * & & * & \\ & & I & & \\ & * & & * & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}$$

The selection of pivot pairs $(i(k), j(k))$ defines pivot strategy.

Derivation of the Real and Complex HZ Method, $B \succ O$

At step k we denote: $\quad A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

$$\hat{A} = \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix}, \quad \hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix}.$$

$\hat{A}$, $\hat{B}$, $\hat{Z}$ are pivot submatrices of A, B, Z.

Derivation of the Real and Complex HZ Method, $B \succ O$

At step k we denote: $A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

$$\hat{A} = \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix}, \quad \hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix}.$$

$\hat{A}$, $\hat{B}$, $\hat{Z}$ are pivot submatrices of A, B, Z.

Then $A' = Z^* A Z, \quad B' = Z^* B Z$ implies $\hat{A}' = \hat{Z}^* \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^* \hat{B} \hat{Z}$.

Derivation of the Real and Complex HZ Method, $B \succ O$

At step k we denote: $A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

$$\hat{A} = \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix}, \quad \hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix}.$$

$\hat{A}$, $\hat{B}$, $\hat{Z}$ are pivot submatrices of A, B, Z.

Then $A' = Z^* A Z$, $B' = Z^* B Z$ implies $\hat{A}' = \hat{Z}^* \hat{A} \hat{Z}$, $\hat{B}' = \hat{Z}^* \hat{B} \hat{Z}$.

$\hat{Z}$ is chosen to diagonalize $\hat{A}'$ and to make $\hat{B}'$ identity matrix I_2.

Derivation of the Real and Complex HZ Method, $B \succ O$

At step k we denote: $\quad A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

$$\hat{A} = \begin{bmatrix} a_{ii} & a_{ij} \\ \bar{a}_{ij} & a_{jj} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix}, \quad \hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix}.$$

$\hat{A}$, $\hat{B}$, $\hat{Z}$ are pivot submatrices of A, B, Z.

Then $A' = Z^* A Z$, $B' = Z^* B Z$ implies $\hat{A}' = \hat{Z}^* \hat{A} \hat{Z}$, $\hat{B}' = \hat{Z}^* \hat{B} \hat{Z}$.

$\hat{Z}$ is chosen to diagonalize $\hat{A}'$ and to make $\hat{B}'$ identity matrix I_2.

$\hat{Z}$ is sought in the form of a product of two Jacobi rotations and one or two diagonal matrices.

Real Algorithm: $\hat{Z}$ is sought in the form:

(a) $$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1+b_{ij}}} & 0 \\ 0 & \frac{1}{\sqrt{1-b_{ij}}} \end{bmatrix} \begin{bmatrix} \cos(\theta-\frac{\pi}{4}) & -\sin(\theta-\frac{\pi}{4}) \\ \sin(\theta-\frac{\pi}{4}) & \cos(\theta-\frac{\pi}{4}) \end{bmatrix}$$

(b) $$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1-b_{ij}}} & 0 \\ 0 & \frac{1}{\sqrt{1+b_{ij}}} \end{bmatrix} \begin{bmatrix} \cos(\theta+\frac{\pi}{4}) & -\sin(\theta+\frac{\pi}{4}) \\ \sin(\theta+\frac{\pi}{4}) & \cos(\theta+\frac{\pi}{4}) \end{bmatrix}$$

$\downarrow$ $\hat{B} \to \text{diag}$ | $\downarrow$ $\hat{B} \to I_2$ | $\downarrow$ $\hat{A} \to \text{diag}$

Real Algorithm: $\hat{Z}$ is sought in the form:

(a) $$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1+b_{ij}}} & 0 \\ 0 & \frac{1}{\sqrt{1-b_{ij}}} \end{bmatrix} \begin{bmatrix} \cos(\theta - \frac{\pi}{4}) & -\sin(\theta - \frac{\pi}{4}) \\ \sin(\theta - \frac{\pi}{4}) & \cos(\theta - \frac{\pi}{4}) \end{bmatrix}$$

(b) $$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1-b_{ij}}} & 0 \\ 0 & \frac{1}{\sqrt{1+b_{ij}}} \end{bmatrix} \begin{bmatrix} \cos(\theta + \frac{\pi}{4}) & -\sin(\theta + \frac{\pi}{4}) \\ \sin(\theta + \frac{\pi}{4}) & \cos(\theta + \frac{\pi}{4}) \end{bmatrix}$$

$\downarrow$ $\hat{B} \to \text{diag}$ | $\downarrow$ $\hat{B} \to I_2$ | $\downarrow$ $\hat{A} \to \text{diag}$

Both approaches yield the same algorithm.

Essential Part of the Real Algorithm

$$\xi = \frac{b_{ij}}{\sqrt{1+b_{ij}} + \sqrt{1-b_{ij}}}, \quad \rho = \frac{1}{2}(\sqrt{1+b_{ij}} + \sqrt{1-b_{ij}}), \quad \xi^2 + \rho^2 = 1,$$

$$\tan(2\theta) = \frac{2a_{ij} - (a_{ii} + a_{jj})\, b_{ij}}{\sqrt{1-(b_{ij})^2}\,(a_{ii} - a_{jj})}, \qquad -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4},$$

$$\begin{aligned}
\cos\phi &= \rho\cos\theta - \xi\sin\theta \\
\sin\phi &= \rho\sin\theta + \xi\cos\theta \\
\cos\psi &= \rho\cos\theta + \xi\sin\theta \\
\sin\psi &= \rho\sin\theta - \xi\cos\theta
\end{aligned}$$

$$\boxed{\hat{Z} = \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} \cos\phi & -\sin\phi \\ \cos\psi & \sin\psi \end{bmatrix}.}$$

Essential Part of the Real Algorithm

$$\xi = \frac{b_{ij}}{\sqrt{1+b_{ij}}+\sqrt{1-b_{ij}}}, \quad \rho = \frac{1}{2}(\sqrt{1+b_{ij}}+\sqrt{1-b_{ij}}), \quad \xi^2+\rho^2=1,$$

$$\tan(2\theta) = \frac{2a_{ij}-(a_{ii}+a_{jj})\,b_{ij}}{\sqrt{1-(b_{ij})^2}\,(a_{ii}-a_{jj})}, \qquad -\frac{\pi}{4} \le \theta \le \frac{\pi}{4},$$

$$\begin{aligned}
\cos\phi &= \rho\cos\theta - \xi\sin\theta \\
\sin\phi &= \rho\sin\theta + \xi\cos\theta \\
\cos\psi &= \rho\cos\theta + \xi\sin\theta \\
\sin\psi &= \rho\sin\theta - \xi\cos\theta
\end{aligned}$$

$$\boxed{\hat{Z} = \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} \cos\phi & -\sin\phi \\ \cos\psi & \sin\psi \end{bmatrix}.}$$

$$\begin{aligned}
a'_{ii} &= a_{ii} + \frac{1}{1-b_{ij}^2}\left[(b_{ij}^2-\sin^2\phi)\,a_{ii} + 2\cos\phi\sin\psi\,a_{ij} + \sin^2\psi\,a_{jj}\right] \\
a'_{jj} &= a_{jj} - \frac{1}{1-b_{ij}^2}\left[(\sin^2\psi-b_{ij}^2)\,a_{jj} + 2\cos\psi\sin\phi\,a_{ij} + \sin^2\phi\,a_{ii}\right]
\end{aligned}$$

Complex Algorithm: $\hat{Z}$ is sought in the form:

$$\hat{B} \to \text{diag} \qquad\qquad \hat{B} \to I_2$$

$$\uparrow \qquad\qquad \uparrow$$

$$\hat{Z} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} e^{\imath \arg(b_{ij})} \\ \frac{\sqrt{2}}{2} e^{-\imath \arg(b_{ij})} & \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{1+|b_{ij}|}} & 0 \\ 0 & \frac{1}{\sqrt{1-|b_{ij}|}} \end{bmatrix}$$

$$\cdot \begin{bmatrix} \cos(\theta - \frac{\pi}{4}) & -e^{\imath\alpha} \sin(\theta - \frac{\pi}{4}) \\ e^{-\imath\alpha} \sin(\theta - \frac{\pi}{4}) & \cos(\theta - \frac{\pi}{4}) \end{bmatrix} \cdot \begin{bmatrix} e^{\imath\omega_i} & 0 \\ 0 & e^{\imath\omega_j} \end{bmatrix}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\hat{A} \to \text{diag} \qquad\qquad \text{diag}(\hat{Z}) \succ O$$

Essential Part of the Complex Algorithm

Let

$$b = |b_{ij}|, \quad t = \sqrt{1 - b^2}, \quad e = a_{jj} - a_{ii}, \qquad \epsilon = \begin{cases} 1, & e \geq 0 \\ -1, & e < 0 \end{cases},$$

Essential Part of the Complex Algorithm

Let

$$b = |b_{ij}|, \quad t = \sqrt{1-b^2}, \quad e = a_{jj} - a_{ii}, \quad \epsilon = \begin{cases} 1, & e \geq 0 \\ -1, & e < 0 \end{cases},$$

$$\begin{aligned}
u + \imath\, v &= e^{-\imath \arg(b_{ij})}\, a_{ij}, \qquad \tan\gamma = 2\tfrac{v}{e}, \quad -\tfrac{\pi}{2} < \gamma \leq \tfrac{\pi}{2} \\
\tan 2\theta &= \epsilon \tfrac{2u-(a_{ii}+a_{jj})b}{t\sqrt{e^2+4v^2}}, \quad -\tfrac{\pi}{4} < \theta \leq \tfrac{\pi}{4} \\
2\cos^2\phi &= 1 + b\sin 2\theta + t\cos 2\theta \cos\gamma, \qquad 0 \leq \phi \leq \tfrac{\pi}{2} \\
2\cos^2\psi &= 1 - b\sin 2\theta + t\cos 2\theta \cos\gamma, \qquad 0 \leq \psi \leq \tfrac{\pi}{2} \\
e^{\imath\alpha}\sin\phi &= \tfrac{e^{\imath \arg(b_{ij})}}{2\cos\psi}\left[\sin 2\theta - b - \imath t\cos 2\theta \sin\gamma\right] \\
e^{-\imath\beta}\sin\psi &= \tfrac{e^{-\imath \arg(b_{ij})}}{2\cos\phi}\left[\sin 2\theta + b + \imath t\cos 2\theta \sin\gamma\right].
\end{aligned}$$

Then

$$\hat{Z} = \frac{1}{\sqrt{1-b^2}} \begin{bmatrix} \cos\phi & e^{\imath\alpha}\sin\phi \\ -e^{-\imath\beta}\sin\psi & \cos\psi \end{bmatrix}$$

The Main Characteristics of the HZ Algorithms

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect
- Theoretical results exist (Global and asymptotic convergence is proved, much is known on the relative accuracy of the computed eigenvalues)

The Main Characteristics of the HZ Algorithms

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect
- Theoretical results exist (Global and asymptotic convergence is proved, much is known on the relative accuracy of the computed eigenvalues)
- It requires B to be positive definite (it solves PGEP)

Derivation of Complex CJ Method, $B \succ O$

Derivation of Complex CJ Method, $B \succ O$

- Cholesky-Jacobi is a hybrid algorithm

Derivation of Complex CJ Method, $B \succ O$

- Cholesky-Jacobi is a hybrid algorithm
- It is composed of two algorithms: LL^*J and RR^*J algorithms

Derivation of Complex CJ Method, $B \succ O$

- Cholesky-Jacobi is a hybrid algorithm
- It is composed of two algorithms: LL^*J and RR^*J algorithms
- In each step it chooses one which is more accurate for the given data

Derivation of Complex CJ Method, $B \succ O$

- Cholesky-Jacobi is a hybrid algorithm
- It is composed of two algorithms: LL^*J and RR^*J algorithms
- In each step it chooses one which is more accurate for the given data
- We derive the complex CJ algorithm

Derivation of Complex CJ Method, $B \succ O$

- Cholesky-Jacobi is a hybrid algorithm
- It is composed of two algorithms: LL^*J and RR^*J algorithms
- In each step it chooses one which is more accurate for the given data
- We derive the complex CJ algorithm
- The real CJ algorithm is obtained by simplifying the complex one

Derivation of Complex LL^*J Algorithm, $B \succ O$

Consider the Cholesky foctorization of $\hat{B}$: $\hat{B} = \hat{L}\hat{L}^*$,

$$\begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^* = \begin{bmatrix} 1 & 0 \\ \bar{a} & \bar{c} \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ \bar{a} & |a|^2 + |c|^2 \end{bmatrix}$$

Derivation of Complex LL^*J Algorithm, $B \succ O$

Consider the Cholesky foctorization of $\hat{B}$: $\hat{B} = \hat{L}\hat{L}^*$,

$$\begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^* = \begin{bmatrix} 1 & 0 \\ \bar{a} & \bar{c} \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ \bar{a} & |a|^2 + |c|^2 \end{bmatrix}$$

Assuming $c > 0$, one obtains $a = b_{ij}, \; c = \tau \equiv \sqrt{1 - |b_{ij}|^2}$.

Derivation of Complex LL^*J Algorithm, $B \succ O$

Consider the Cholesky foctorization of $\hat{B}$: $\hat{B} = \hat{L}\hat{L}^*$,

$$\begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^* = \begin{bmatrix} 1 & 0 \\ \bar{a} & \bar{c} \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ \bar{a} & |a|^2 + |c|^2 \end{bmatrix}$$

Assuming $c > 0$, one obtains $\quad a = b_{ij}, \;\; c = \tau \equiv \sqrt{1 - |b_{ij}|^2}$.

$$\hat{L} = \begin{bmatrix} 1 & 0 \\ \bar{b}_{ij} & \tau \end{bmatrix}, \qquad \hat{L}^{-1} = \frac{1}{\tau} \begin{bmatrix} \tau & 0 \\ -\bar{b}_{ij} & 1 \end{bmatrix}, \qquad \hat{L}^{-*} = \frac{1}{\tau} \begin{bmatrix} \tau & -b_{ij} \\ 0 & 1 \end{bmatrix}.$$

Derivation of Complex LL^*J Algorithm, $B \succ O$

Consider the Cholesky foctorization of $\hat{B}$: $\hat{B} = \hat{L}\hat{L}^*$,

$$\begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^* = \begin{bmatrix} 1 & 0 \\ \bar{a} & \bar{c} \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ \bar{a} & |a|^2 + |c|^2 \end{bmatrix}$$

Assuming $c > 0$, one obtains $a = b_{ij}$, $c = \tau \equiv \sqrt{1 - |b_{ij}|^2}$.

$$\hat{L} = \begin{bmatrix} 1 & 0 \\ \bar{b}_{ij} & \tau \end{bmatrix}, \qquad \hat{L}^{-1} = \frac{1}{\tau} \begin{bmatrix} \tau & 0 \\ -\bar{b}_{ij} & 1 \end{bmatrix}, \qquad \hat{L}^{-*} = \frac{1}{\tau} \begin{bmatrix} \tau & -b_{ij} \\ 0 & 1 \end{bmatrix}.$$

Let $\hat{F}_1 = \hat{L}^{-*}$. Then $\hat{F}_1^* \hat{B} \hat{F}_1 = I_2$ and

$$\hat{F}_1^* \hat{A} \hat{F}_1 = \begin{bmatrix} a_{ii} & (a_{ij} - b_{ij} a_{ii})/\tau \\ (\bar{a}_{ij} - \bar{b}_{ij} a_{ii})/\tau & a_{jj} - \frac{a_{ij}\bar{b}_{ij} + \bar{a}_{ij} b_{ij} - (a_{ii} + a_{jj})|b_{ij}|^2}{1 - |b_{ij}|^2} \end{bmatrix}.$$

Derivation of Complex LL^*J Algorithm, $B \succ O$

The final $\hat{F}$ is obtained as product $\hat{F} = \hat{F}_1 \hat{R}_1$ where

$\hat{R}_1$ is the complex Jacobi rotation which diagonalizes $\hat{F}_1^* \hat{A} \hat{F}_1$.

Let us assume that the $(1,2)$-element of $\hat{R}_1$ is $-e^{\imath \epsilon_1} \sin \vartheta_1$. Then the angles ϑ_1 and ϵ_1 are determined by the formulas

$$\begin{aligned} \epsilon_1 &= \arg(a_{ij} - b_{ij} a_{ii}), \\ \tan(2\vartheta_1) &= \frac{2|a_{ij} - a_{ii} b_{ij}| \sqrt{1 - |b_{ij}|^2}}{a_{ii} - a_{jj} + a_{ij} \bar{b}_{ij} + \bar{a}_{ij} b_{ij} - 2a_{ii}|b_{ij}|^2}, \quad -\frac{\pi}{4} \leq \vartheta_1 \leq \frac{\pi}{4}. \end{aligned}$$

Derivation of Complex LL^*J Algorithm, $B \succ O$

The transformation formulas for the diagonal elements of A read

$$
\begin{aligned}
a'_{ii} &= a_{ii} + \tan\vartheta_1 \cdot \frac{|a_{ij} - a_{ii} b_{ij}|}{\sqrt{1 - b_{ij}^2}} \\
a'_{jj} &= a_{jj} - \frac{a_{ij}\bar{b}_{ij} + \bar{a}_{ij} b_{ij} - (a_{ii} + a_{jj})|b_{ij}|^2}{1 - |b_{ij}|^2} - \tan\vartheta_1 \cdot \frac{|a_{ij} - a_{ii} b_{ij}|}{\sqrt{1 - b_{ij}^2}}.
\end{aligned}
$$

In the case $a_{ii} = a_{jj}$, $a_{ij} = a_{ii} b_{ij}$, $\tan(2\vartheta_1)$ has the form $0/0$.

The transformation formulas for the diagonal elements of A read

$$\begin{aligned} a'_{ii} &= a_{ii} + \tan\vartheta_1 \cdot \frac{|a_{ij} - a_{ii} b_{ij}|}{\sqrt{1 - b_{ij}^2}} \\ a'_{jj} &= a_{jj} - \frac{a_{ij}\bar{b}_{ij} + \bar{a}_{ij} b_{ij} - (a_{ii} + a_{jj})|b_{ij}|^2}{1 - |b_{ij}|^2} - \tan\vartheta_1 \cdot \frac{|a_{ij} - a_{ii} b_{ij}|}{\sqrt{1 - b_{ij}^2}}. \end{aligned}$$

In the case $a_{ii} = a_{jj}$, $a_{ij} = a_{ii} b_{ij}$, $\tan(2\vartheta_1)$ has the form $0/0$.

Then we choose $\vartheta_1 = 0$, so that $a'_{ii} = a_{ii}$ and $a'_{jj} = a_{jj}$.

Let $c_{\vartheta_1} = \cos\vartheta_1$, $s^{\pm}_{\vartheta_1} = e^{\pm\imath\epsilon_1}\sin\vartheta_1$. Then

$$\hat{F} = \frac{1}{\sqrt{1-|b_{ij}|^2}} \begin{bmatrix} \sqrt{1-|b_{ij}|^2} & -b_{ij} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\vartheta_1} & -s^{+}_{\vartheta_1} \\ s^{-}_{\vartheta_1} & c_{\vartheta_1} \end{bmatrix}$$

$$= \frac{1}{\sqrt{1-|b_{ij}|^2}} \begin{bmatrix} c_{\tilde{\vartheta}_1} & -s_{\tilde{\vartheta}_1} \\ s^{-}_{\vartheta_1} & c_{\vartheta_1} \end{bmatrix} \qquad \begin{aligned} c_{\tilde{\vartheta}_1} &= c_{\vartheta_1}\sqrt{1-|b_{ij}|^2} - s^{-}_{\vartheta_1} b_{ij} \\ s_{\tilde{\vartheta}_1} &= c_{\vartheta_1} b_{ij} + s^{+}_{\vartheta_1}\sqrt{1-|b_{ij}|^2} \end{aligned}$$

$$= \begin{bmatrix} c1 & -s1 \\ s2 & c2 \end{bmatrix},$$

$$c1 = c_{\vartheta_1} - s^{-}_{\vartheta_1} b_{ij}/\sqrt{1-|b_{ij}|^2}, \qquad c2 = c_{\vartheta_1}/\sqrt{1-|b_{ij}|^2},$$

$$s1 = c_{\vartheta_1} b_{ij}/\sqrt{1-|b_{ij}|^2} + s^{+}_{\vartheta_1}, \qquad s2 = s^{-}_{\vartheta_1}/\sqrt{1-|b_{ij}|^2}.$$

Derivation of Complex RR^*J Algorithm, $B \succ O$

Instead of LL^*, one can use RR^* factorization of $\hat{B}$. Then we have

Instead of LL^*, one can use RR^* factorization of $\hat{B}$. Then we have

$$\begin{bmatrix} 1 & b_{ij} \\ \bar{b}_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{R}\hat{R}^* = \begin{bmatrix} c & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{c} & 0 \\ \bar{a} & 1 \end{bmatrix} = \begin{bmatrix} |a|^2 + |c|^2 & a \\ \bar{a} & 1 \end{bmatrix}.$$

Assuming positive c, one obtains $a = b_{ij}$, $c = \sqrt{1 - |b_{ij}|^2} = \tau$. Hence

$$\hat{R} = \begin{bmatrix} \tau & b_{ij} \\ 0 & 1 \end{bmatrix}, \qquad \hat{R}^{-1} = \frac{1}{\tau} \begin{bmatrix} 1 & -b_{ij} \\ 0 & \tau \end{bmatrix}, \qquad \hat{R}^{-*} = \frac{1}{\tau} \begin{bmatrix} 1 & 0 \\ -\bar{b}_{ij} & \tau \end{bmatrix}.$$

If we write $\hat{F}_2 = \hat{R}^{-*}$, then $\hat{F}_2^* \hat{B} \hat{F}_2 = \hat{R}^{-1} \hat{B} \hat{R}^{-*} = I_2$ and we have

The Algorithm RR^*J

$$\hat{F}_2^* \hat{A} \hat{F}_2 = \begin{bmatrix} a_{ii} - \frac{a_{ij}\bar{b}_{ij} + \bar{a}_{ij} b_{ij} - (a_{ii}+a_{jj})|b_{ij}|^2}{\tau^2} & (a_{ij} - a_{jj} b_{ij})/\tau \\ (\bar{a}_{ij} - a_{jj}\bar{b}_{ij})/\tau & a_{jj} \end{bmatrix}.$$

- The final transformation is $\hat{F} = \hat{F}_2 \hat{R}_2$,
- $\hat{R}_2$ is the Jacobi rotation which annihilates $(1,2)$-element of $\hat{F}_2^* \hat{A} \hat{F}_2$
- Let $(1,2)$-element of $\hat{R}_2$ be $-e^{\imath\epsilon_2}\sin\vartheta_2$

Then the parameters ϵ_2 and ϑ_2 are determined by the formulas

$$\begin{aligned} \epsilon_2 &= \arg(a_{ij} - b_{ij} a_{jj}), \\ \tan(2\vartheta_2) &= \frac{2|a_{ij} - a_{jj} b_{ij}|\sqrt{1 - |b_{ij}|^2}}{a_{ii} - a_{jj} - (a_{ij}\bar{b}_{ij} + \bar{a}_{ij} b_{ij}) + 2a_{jj}|b_{ij}|^2}, \quad -\frac{\pi}{4} \leq \vartheta_2 \leq \frac{\pi}{4}. \end{aligned}$$

The Algorithm RR^*J

The transformation formulas for the diagonal elements of A:

$$
\begin{aligned}
a'_{ii} &= a_{ii} - \frac{a_{ij}\bar{b}_{ij} + \bar{a}_{ij}b_{ij} - (a_{ii} + a_{jj})|b_{ij}|^2}{1 - |b_{ij}|^2} + \tan\vartheta_2 \cdot \frac{|a_{ij} - a_{jj}b_{ij}|}{\sqrt{1 - b_{ij}^2}}, \\
a'_{jj} &= a_{jj} - \tan\vartheta_2 \cdot \frac{|a_{ij} - a_{jj}b_{ij}|}{\sqrt{1 - b_{ij}^2}}.
\end{aligned}
$$

If $a_{ii} = a_{jj}$, $a_{ij} = a_{jj}b_{ij}$, ϑ_2 is not well defined and we choose $\vartheta_2 = 0$.

In that case a'_{ii} and a'_{jj} reduce to a_{ii} and a_{jj}, respectively.

The Algorithm RR^*J

Let $c_{\vartheta_2} = \cos\vartheta_2$, $s_{\vartheta_2}^{\pm} = e^{\pm\imath\epsilon_2}\sin\vartheta_2$. Then

$$
\begin{aligned}
\hat{F} &= \frac{1}{\sqrt{1-|b_{ij}|^2}} \begin{bmatrix} 1 & 0 \\ -\bar{b}_{ij} & \sqrt{1-|b_{ij}|^2} \end{bmatrix} \begin{bmatrix} c_{\vartheta_2} & -s_{\vartheta_2}^{+} \\ s_{\vartheta_2}^{-} & c_{\vartheta_2} \end{bmatrix} \\
&= \frac{1}{\sqrt{1-|b_{ij}|^2}} \begin{bmatrix} c_{\vartheta_2} & -s_{\vartheta_2}^{+} \\ s_{\tilde{\vartheta}_2} & c_{\tilde{\vartheta}_2} \end{bmatrix}, \qquad \begin{aligned} c_{\tilde{\vartheta}_2} &= c_{\vartheta_2}\sqrt{1-|b_{ij}|^2} + s_{\vartheta_2}^{+}\bar{b}_{ij} \\ s_{\tilde{\vartheta}_2} &= s_{\vartheta_2}^{-}\sqrt{1-b_{ij}^2} - c_{\vartheta_2}\bar{b}_{ij} \end{aligned} \\
&= \begin{bmatrix} c1 & -s1 \\ s2 & c2 \end{bmatrix}, \quad \text{It is easy to check that } c_{\tilde{\vartheta}}^2 + s_{\tilde{\vartheta}}^2 = 1.
\end{aligned}
$$

$$
c1 = c_{\vartheta_2}/\sqrt{1-b_{ij}^2}, \qquad c2 = c_{\vartheta_2} + s_{\vartheta_2}^{+}\bar{b}_{ij}/\sqrt{1-b_{ij}^2},
$$

$$
s1 = s_{\vartheta_2}^{+}/\sqrt{1-b_{ij}^2}^{\,+}, \qquad s2 = s_{\vartheta_2}^{-} - c_{\vartheta_2}\bar{b}_{ij}/\sqrt{1-b_{ij}^2}.
$$

We can postmultiply $\hat{F}$ by $\mathrm{diag}(1\,,\, \bar{c}_{\tilde{\vartheta}_2}/|c_{\tilde{\vartheta}_2}|)$ provided that $c_{\tilde{\vartheta}_2} \neq 0$. This ensures that (the updated) $\hat{F}$ has nonnegative diagonal elements.

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

1. select the pivot pair (i, j)

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

1. select the pivot pair (i, j)
2. if $a_{ii} \leq a_{jj}$ then employ the LL^*J algorithm

   else employ the RR^*J algorithm

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

1. select the pivot pair (i, j)
2. if $a_{ii} \leq a_{jj}$ then employ the LL^*J algorithm

   else employ the RR^*J algorithm

Our numerical tests show that neither LL^*J nor RR^*J is indicated as a high relative accurate algorithm on pairs of well-behaved positive definite matrices.

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

1. select the pivot pair (i, j)
2. if $a_{ii} \leq a_{jj}$ then employ the LL^*J algorithm

   else employ the RR^*J algorithm

Our numerical tests show that neither LL^*J nor RR^*J is indicated as a high relative accurate algorithm on pairs of well-behaved positive definite matrices.

The same can be said for the hybrid algorithm that selects the LL^*J and RR^*J algorithms in the opposite way, i.e. selects the RR^*J (LL^*J) algorithm when $a_{ii} \leq a_{jj}$ ($a_{ii} > a_{jj}$).

The Cholesky-Jacobi Algorithm

The CJ is a hybrid algorithm which can be briefly defined as follows:

1. select the pivot pair (i,j)
2. if $a_{ii} \leq a_{jj}$ then employ the LL^*J algorithm

   else employ the RR^*J algorithm

Our numerical tests show that neither LL^*J nor RR^*J is indicated as a high relative accurate algorithm on pairs of well-behaved positive definite matrices.

The same can be said for the hybrid algorithm that selects the LL^*J and RR^*J algorithms in the opposite way, i.e. selects the RR^*J (LL^*J) algorithm when $a_{ii} \leq a_{jj}$ ($a_{ii} > a_{jj}$).

Only the above definition warrants the high relative accuracy of the algorithm and it is in complete agreement with the behavior of the real CJ method.

The Main Characteristics of the CJ Method

The Main Characteristics of the CJ Method

- Fast (Quadratic asymptotic convergence)

The Main Characteristics of the CJ Method

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)

The Main Characteristics of the CJ Method

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code

The Main Characteristics of the CJ Method

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect

The Main Characteristics of the CJ Method

- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect
- Theoretical results exist (Global convergence is proved, much is known on the asymptotic convergence and on the relative accuracy of the computed eigenvalues)

The Main Characteristics of the CJ Method

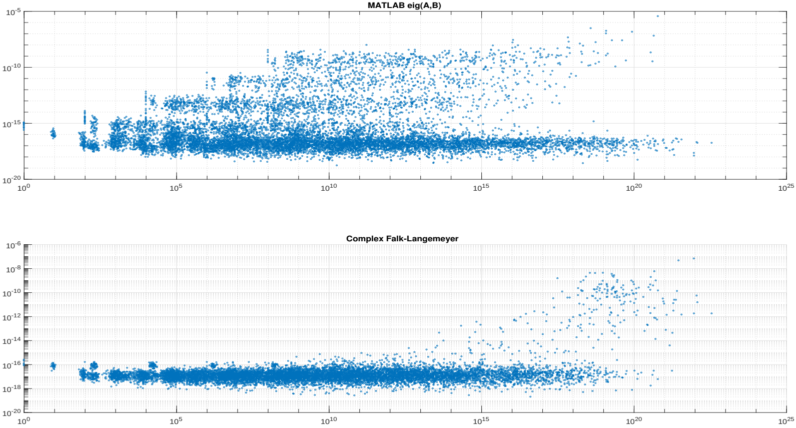
- Fast (Quadratic asymptotic convergence)
- Very accurate (HRA on well-behaved positive definite matrices)
- No problem with renormalizations, easy to code
- Unit diagonal in B has a stabilizing effect
- Theoretical results exist (Global convergence is proved, much is known on the asymptotic convergence and on the relative accuracy of the computed eigenvalues)
- It requires B to be positive definite (it solves PGEP)

Relative Accuracy, Assume: $A \succ O$, $B \succ O$

$$A_S = [\mathrm{diag}(A)]^{-1/2} A [\mathrm{diag}(A)]^{-1/2}, \quad B_S = [\mathrm{diag}(B)]^{-1/2} B [\mathrm{diag}(B)]^{-1/2}$$

$$\begin{aligned}
\varrho_{(A,B)} &= \max_{1 \le i \le n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \Big/ \sqrt{\kappa_2^2(A_S) + \kappa_2^2(B_S)} \\
\chi_{(A,B)} &= \sqrt{\kappa_2^2(A^{(0)}) + \kappa_2^2(B^{(0)})} \\
\mathcal{E} &= \{(\chi_{(A,B)}\ ,\ \varrho_{(A,B)}) : (A,B) \in \Upsilon\}.
\end{aligned}$$

Relative errors: *CFL* vs. MATLAB eig(A,B)



Some References and Thank You for Your Patience!

- V. Hari, *Globally convergent Jacobi methods for positive definite matrix pairs*, Numer. Algor. (2017). https://doi.org/10.1007/s11075-017-0435-5
- V. Hari, *Complex Cholesky-Jacobi Algorithm for PGEP*, proposed for publ. in AIP Conference Proceedings of ICNAAM 2018
- V. Hari, *Complex Falk-Langemeyer Method*, proposed for publ. in Numer. Algor.
- V. Hari, E. Begović Kovač, *Convergence of the Cyclic and Quasi-cyclic Block Jacobi Methods*. Electron. T. Numer. Ana. (ETNA), 46 (2017) 107-147
- V. Novaković, S. Singer, S. Singer, *Blocking and Parallelization of the Hari-Zimmermann Variant of the Falk-Langemeyer Algorithm for the Generalized SVD*, Parallel Comput., 49 (2015) 136-152.