On Jacobi Methods for the Positive Definite Generalized Eigenvalue Problem

Vjeran Hari

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

Parallel Numerical Computing and Its Applications
Smolenice Castle, Slovakia

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

OUTLINE

OUTLINE

- GEP and PGEP

OUTLINE

- GEP and PGEP
- Derivation of the algorithms

OUTLINE

- GEP and PGEP
- Derivation of the algorithms
- Convergence, global and asymptotic

OUTLINE

- GEP and PGEP
- Derivation of the algorithms
- Convergence, global and asymptotic
- Stability and relative accuracy

OUTLINE

- GEP and PGEP
- Derivation of the algorithms
- Convergence, global and asymptotic
- Stability and relative accuracy
- Block algorithms

OUTLINE

- GEP and PGEP
- Derivation of the algorithms
- Convergence, global and asymptotic
- Stability and relative accuracy
- Block algorithms
- Global convergence of block algorithms

GEP and PGEP

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

GEP and PGEP

Let $A = A^T, \quad B = B^T$.
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 or shorter PGEP.

GEP and PGEP

Let $A = A^T, \quad B = B^T$.
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 or shorter PGEP.

For such a pair there is a nonsingular matrix F such that

$$F^T AF = \Lambda_A\,, \qquad F^T BF = \Lambda_B,$$

$\Lambda_A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n), \quad \Lambda_B = \mathrm{diag}(\beta_1, \ldots, \beta_n) \succ O.$

GEP and PGEP

Let $A = A^T, \quad B = B^T$.
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 or shorter PGEP.

For such a pair there is a nonsingular matrix F such that

$$F^T AF = \Lambda_A\,, \qquad F^T BF = \Lambda_B,$$

$\Lambda_A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n), \quad \Lambda_B = \mathrm{diag}(\beta_1, \ldots, \beta_n) \succ O.$

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

How to solve PGEP?

One can try with the transformation $(A, B) \mapsto (L^{-1}AL^{-T}, I)$, $B = LL^T$ and reduce PGEP to the standard EP for one symmetric matrix.

If L has small singular value(s), then the computed $L^{-1}AL^{-T}$ will have corrupt eigenvalues.

How to solve PGEP?

One can try with the transformation $(A, B) \mapsto (L^{-1}AL^{-T}, I)$, $B = LL^T$ and reduce PGEP to the standard EP for one symmetric matrix.

If L has small singular value(s), then the computed $L^{-1}AL^{-T}$ will have corrupt eigenvalues. Then one can try to maximize the minimum eigenvalue of B by rotating the pair

$$(A, B) \mapsto (A_\varphi, B_\varphi) = (A\cos\varphi + B\sin\varphi, -A\sin\varphi + B\cos\varphi),$$

or derive a method which works with the initial pair (A, B).

How to solve PGEP?

One can try with the transformation $(A, B) \mapsto (L^{-1}AL^{-T}, I)$, $B = LL^T$ and reduce PGEP to the standard EP for one symmetric matrix.

If L has small singular value(s), then the computed $L^{-1}AL^{-T}$ will have corrupt eigenvalues. Then one can try to maximize the minimum eigenvalue of B by rotating the pair

$$(A, B) \mapsto (A_\varphi, B_\varphi) = (A \cos\varphi + B \sin\varphi, -A \sin\varphi + B \cos\varphi),$$

or derive a method which works with the initial pair (A, B).

We follow the second path.

Jacobi methods for PGEP

We have two diagonalization methods for PGEP

Jacobi methods for PGEP

We have two diagonalization methods for PGEP

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

Jacobi methods for PGEP

We have two diagonalization methods for PGEP

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann variant of the FL method (shorter: HZ method)
  (Hari Ph.D. 1984)

Jacobi methods for PGEP

We have two diagonalization methods for PGEP

- Falk-Langemeyer method (shorter: FL method)
  (Elektronische Datenverarbeitung, 1960)
- Hari-Zimmermann variant of the FL method (shorter: HZ method)
  (Hari Ph.D. 1984)

The two methods are connected: the FL method can be viewed as the HZ method with “fast scaled” transformations. So, the FL method seems to be somewhat faster and the HZ method seems to be more robust.

Jacobi methods for PGEP

V. Novaković, S. Singer, S. Singer (Parallel Comput., 2015):

Jacobi methods for PGEP

V. Novaković, S. Singer, S. Singer (Parallel Comput., 2015):

Numerical tests on large matrices, on parallel machines, have confirmed the advantage of the HZ approach.

Jacobi methods for PGEP

V. Novaković, S. Singer, S. Singer (Parallel Comput., 2015):

Numerical tests on large matrices, on parallel machines, have confirmed the advantage of the HZ approach.

When implemented as one-sided block algorithm for the GSVD, it is almost perfectly parallelizable, so parallel shared memory versions of the algorithm are highly scalable, and their speedup almost solely depends on the number of cores used.

Derivation of the HZ Method

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

Derivation of the HZ Method

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

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

Derivation of the HZ Method

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

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

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

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

Derivation of the HZ Method

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

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

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

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

Each Z_k is a nonsingular elementary plane matrix

Derivation of the HZ Method

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

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

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

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

Each Z_k is a nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & c_k & & -s_k & \\ & & I & & \\ & \tilde{s}_k & & \tilde{c}_k & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}, \qquad i(k) < j(k) \text{ are pivot indices at step } k,$$

$$c_k^2 + s_k^2 = \tilde{c}_k^2 + \tilde{s}_k^2 = 1/\sqrt{1 - b_{i(k)j(k)}^2} \quad \text{(Gose 1979).}$$

Derivation of the HZ Method

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

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

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

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

Each Z_k is a nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & c_k & & -s_k & \\ & & I & & \\ & \tilde{s}_k & & \tilde{c}_k & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}, \qquad i(k) < j(k) \text{ are pivot indices at step } k,$$

$c_k^2 + s_k^2 = \tilde{c}_k^2 + \tilde{s}_k^2 = 1/\sqrt{1 - b_{i(k)j(k)}^2}$ (Gose 1979).

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

Derivation of the HZ Method

To describe step k, we assume: $A = A^{(k)}$, $A' = A^{(k+1)}$, $Z_k = Z$,

$$\hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix}$$ the pivot submatrix of Z.

Derivation of the HZ Method

To describe step k, we assume: $\quad A = A^{(k)}$, $A' = A^{(k+1)}$, $Z_k = Z$,

$$\hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix} \qquad \text{the pivot submatrix of } Z.$$

We have

$$A' = Z^T A Z, \quad B' = Z^T B Z \qquad \left(\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z} \right).$$

Derivation of the HZ Method

To describe step k, we assume: $A = A^{(k)}$, $A' = A^{(k+1)}$, $Z_k = Z$,

$$\hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix} \qquad \text{the pivot submatrix of } Z.$$

We have

$$A' = Z^T A Z, \quad B' = Z^T B Z \qquad \left(\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z} \right).$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

Derivation of the HZ Method

To describe step k, we assume: $A = A^{(k)}$, $A' = A^{(k+1)}$, $Z_k = Z$,

$$\hat{Z} = \begin{bmatrix} c & -s \\ \tilde{s} & \tilde{c} \end{bmatrix} \qquad \text{the pivot submatrix of } Z.$$

We have

$$A' = Z^T A Z, \quad B' = Z^T B Z \qquad \left(\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z} \right).$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

$\hat{Z}$ is sought in the form of a product of two Jacobi rotations and one diagonal matrix. We have two possibilities:

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

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

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

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

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

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

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

The both possibilities yield the same algorithm.

Essential Part of the Algorithm

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

Essential Part of the Algorithm

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

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

Essential Part of the Algorithm

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

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

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

Essential Part of the Algorithm

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

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

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

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

Essential Part of the Algorithm

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

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

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

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

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

Digression: Complex Matrices

If $A = A^*$ and $B = B^*$ are complex, with $B \succ O$ and $\mathrm{diag}(B) = I_n$, then one step of the HZ method uses the transformation

Digression: Complex Matrices

If $A = A^*$ and $B = B^*$ are complex, with $B \succ O$ and $\mathrm{diag}(B) = I_n$, then one step of the HZ method uses the transformation

$$A' = Z^* A Z, \quad B' = Z^* B Z,$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

Digression: Complex Matrices

If $A = A^*$ and $B = B^*$ are complex, with $B \succ O$ and $\mathrm{diag}(B) = I_n$, then one step of the HZ method uses the transformation

$$A' = Z^* A Z, \quad B' = Z^* B Z,$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

It is proved that that pivot submatrix of Z has form

$$\hat{Z} = \begin{bmatrix} c & \bar{s} \\ -\tilde{s} & \tilde{c} \end{bmatrix}.$$

Digression: Complex Matrices

If $A = A^*$ and $B = B^*$ are complex, with $B \succ O$ and $\text{diag}(B) = I_n$, then one step of the HZ method uses the transformation

$$A' = Z^* A Z, \quad B' = Z^* B Z,$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

It is proved that that pivot submatrix of Z has form

$$\hat{Z} = \begin{bmatrix} c & \bar{s} \\ -\tilde{s} & \tilde{c} \end{bmatrix}.$$

We obtain $\hat{A}' = \hat{Z}^* \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^* \hat{B} \hat{Z}.$

Digression: Complex Matrices

If $A = A^*$ and $B = B^*$ are complex, with $B \succ O$ and $\mathrm{diag}(B) = I_n$, then one step of the HZ method uses the transformation

$$A' = Z^* A Z, \quad B' = Z^* B Z,$$

Z is chosen to annihilate the pivot elements a_{ij} and b_{ij}.

It is proved that that pivot submatrix of Z has form

$$\hat{Z} = \begin{bmatrix} c & \bar{s} \\ -\tilde{s} & \tilde{c} \end{bmatrix}.$$

We obtain $\hat{A}' = \hat{Z}^* \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^* \hat{B} \hat{Z}$. $\hat{Z}$ is sought as product of two complex Jacobi rotations and two diagonal matrices.

$\hat{Z}$ 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 Algorithm

Let

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

Essential Part of the Algorithm

Let

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

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

Essential Part of the 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

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

New Algorithms: Based on LL^T and RR^T Factorizations

Consider the Cholesky foctorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^T = \begin{bmatrix} 1 & 0 \\ a & c \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & a^2 + c^2 \end{bmatrix}.$$

New Algorithms: Based on LL^T and RR^T Factorizations

Consider the Cholesky foctorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^T = \begin{bmatrix} 1 & 0 \\ a & c \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & a^2 + c^2 \end{bmatrix}.$$

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

New Algorithms: Based on LL^T and RR^T Factorizations

Consider the Cholesky foctorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^T = \begin{bmatrix} 1 & 0 \\ a & c \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & a^2 + c^2 \end{bmatrix}.$$

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

$$\hat{L} = \begin{bmatrix} 1 & 0 \\ b_{ij} & \sqrt{1 - b_{ij}^2} \end{bmatrix}, \quad \hat{L}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{b_{ij}}{\sqrt{1-b_{ij}^2}} & \frac{1}{\sqrt{1-b_{ij}^2}} \end{bmatrix}.$$

New Algorithms: Based on LL^T and RR^T Factorizations

Consider the Cholesky foctorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{L}\hat{L}^T = \begin{bmatrix} 1 & 0 \\ a & c \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & c \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & a^2 + c^2 \end{bmatrix}.$$

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

$$\hat{L} = \begin{bmatrix} 1 & 0 \\ b_{ij} & \sqrt{1 - b_{ij}^2} \end{bmatrix}, \quad \hat{L}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{b_{ij}}{\sqrt{1-b_{ij}^2}} & \frac{1}{\sqrt{1-b_{ij}^2}} \end{bmatrix}.$$

If we write $\hat{F}_1 = \hat{L}^{-T}$, then $\hat{F}_1^T \hat{B} \hat{F}_1 = I_2$ and

The Algorithm Based on LL^T Factorization

$$
\begin{aligned}
\hat{F}_1^T \hat{A} \hat{F}_1 &= \begin{bmatrix} 1 & 0 \\ f_{ij} & f_{jj} \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} 1 & f_{ij} \\ 0 & f_{jj} \end{bmatrix} \\
&= \begin{bmatrix} a_{ii} & f_{ij} a_{ii} + f_{jj} a_{ij} \\ f_{ij} a_{ii} + f_{jj} a_{ij} & f_{ij}^2 a_{ii} + 2 f_{ij} f_{jj} a_{ij} + f_{jj}^2 a_{jj} \end{bmatrix} \\
&= \begin{bmatrix} a_{ii} & \frac{a_{ij} - b_{ij} a_{ii}}{\sqrt{1-b_{ij}^2}} \\ \frac{a_{ij} - b_{ij} a_{ii}}{\sqrt{1-b_{ij}^2}} & a_{jj} - \frac{2a_{ij} - (a_{ii}+a_{jj}) b_{ij}}{1-b_{ij}^2} b_{ij} \end{bmatrix},
\end{aligned} \tag{1}
$$

where we have used $f_{ij} = -b_{ij}/\sqrt{1 - b_{ij}^2}$, $f_{jj} = 1/\sqrt{1 - b_{ij}^2}$.

The Algorithm Based on LL^T Factorization

$$\hat{F}_1^T \hat{A} \hat{F}_1 = \begin{bmatrix} 1 & 0 \\ f_{ij} & f_{jj} \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} 1 & f_{ij} \\ 0 & f_{jj} \end{bmatrix}$$

$$= \begin{bmatrix} a_{ii} & f_{ij} a_{ii} + f_{jj} a_{ij} \\ f_{ij} a_{ii} + f_{jj} a_{ij} & f_{ij}^2 a_{ii} + 2 f_{ij} f_{jj} a_{ij} + f_{jj}^2 a_{jj} \end{bmatrix}$$

$$= \begin{bmatrix} a_{ii} & \frac{a_{ij} - b_{ij} a_{ii}}{\sqrt{1-b_{ij}^2}} \\ \frac{a_{ij} - b_{ij} a_{ii}}{\sqrt{1-b_{ij}^2}} & a_{jj} - \frac{2a_{ij} - (a_{ii}+a_{jj}) b_{ij}}{1-b_{ij}^2} b_{ij} \end{bmatrix}, \qquad (1)$$

where we have used $f_{ij} = -b_{ij}/\sqrt{1 - b_{ij}^2}$, $f_{jj} = 1/\sqrt{1 - b_{ij}^2}$.

The final $\hat{F}$ has the form $\hat{F} = \hat{F}_1 \hat{R}$, where $\hat{R}$ is the Jacobi transformation which diagonalizes $\hat{F}_1^T \hat{A} \hat{F}_1$. Its angle ϑ is determined by the formula

The Algorithm Based on LL^T Factorization

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

The Algorithm Based on LL^T Factorization

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

The transformation formulas for the diagonal elements of A read

$$a'_{ii} = a_{ii} + \tan\vartheta \cdot \frac{a_{ij} - a_{ii}b_{ij}}{\sqrt{1 - b_{ij}^2}} \tag{2}$$

$$a'_{jj} = a_{jj} - \frac{2a_{ij}b_{ij} - b_{ij}^2(a_{ii} + a_{jj})}{1 - b_{ij}^2} - \tan\vartheta \cdot \frac{a_{ij} - a_{ii}b_{ij}}{\sqrt{1 - b_{ij}^2}} \tag{3}$$

The Algorithm Based on LL^T Factorization

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

The transformation formulas for the diagonal elements of A read

$$a'_{ii} = a_{ii} + \tan\vartheta \cdot \frac{a_{ij} - a_{ii}b_{ij}}{\sqrt{1-b_{ij}^2}} \qquad (2)$$

$$a'_{jj} = a_{jj} - \frac{2a_{ij}b_{ij} - b_{ij}^2(a_{ii} + a_{jj})}{1-b_{ij}^2} - \tan\vartheta \cdot \frac{a_{ij} - a_{ii}b_{ij}}{\sqrt{1-b_{ij}^2}} \qquad (3)$$

If $a_{ii} = a_{jj}$, $a_{ij} = a_{ii}b_{ij}$ then ϑ is determined from expression $0/0$, so we choose $\vartheta = 0$. In this case a'_{ii} and a'_{jj} reduce to a_{ii} and a_{jj}, respectively.

The Algorithm Based on LL^T Factorization

This leads to a simpler matrix

$$\begin{aligned}\hat{Z} &= \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} \sqrt{1-b_{ij}^2} & -b_{ij} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_\vartheta & -s_\vartheta \\ s_\vartheta & c_\vartheta \end{bmatrix} \\ &= \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} c_{\tilde{\vartheta}} & -s_{\tilde{\vartheta}} \\ s_\vartheta & c_\vartheta \end{bmatrix}, \qquad \begin{aligned} c_{\tilde{\vartheta}} &= c_\vartheta \sqrt{1-b_{ij}^2} - s_\vartheta b_{ij}, \\ s_{\tilde{\vartheta}} &= c_\vartheta b_{ij} + s_\vartheta \sqrt{1-b_{ij}^2}. \end{aligned}\end{aligned}$$

It is easy to check that $c_{\tilde{\vartheta}}^2 + s_{\tilde{\vartheta}}^2 = 1$.

The Algorithm Based on RR^T Factorizations

Consider the RR^T factorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{R}\hat{R}^T = \begin{bmatrix} c & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c & 0 \\ a & 1 \end{bmatrix} = \begin{bmatrix} a^2 + c^2 & a \\ a & 1 \end{bmatrix}.$$

Assuming positive c, one obtains $a = b_{ij}$, $c = \sqrt{1 - b_{ij}^2}$, hence

$$\hat{R} = \begin{bmatrix} \sqrt{1 - b_{ij}^2} & b_{ij} \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \hat{R}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{1-b_{ij}^2}} & -\frac{b_{ij}}{\sqrt{1-b_{ij}^2}} \\ 0 & 1 \end{bmatrix}.$$

The Algorithm Based on RR^T Factorizations

Consider the RR^T factorization of $\hat{B}$:

$$\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1 \end{bmatrix} = \hat{B} = \hat{R}\hat{R}^T = \begin{bmatrix} c & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c & 0 \\ a & 1 \end{bmatrix} = \begin{bmatrix} a^2 + c^2 & a \\ a & 1 \end{bmatrix}.$$

Assuming positive c, one obtains $a = b_{ij}$, $c = \sqrt{1 - b_{ij}^2}$, hence

$$\hat{R} = \begin{bmatrix} \sqrt{1 - b_{ij}^2} & b_{ij} \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \hat{R}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{1-b_{ij}^2}} & -\frac{b_{ij}}{\sqrt{1-b_{ij}^2}} \\ 0 & 1 \end{bmatrix}.$$

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

The Algorithm Based on RR^T Factorization

$$
\begin{aligned}
\hat{F}_2^T \hat{A} \hat{F}_2 &= \begin{bmatrix} f_{ii} & f_{ji} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} f_{ii} & 0 \\ f_{ji} & 1 \end{bmatrix} \\
&= \begin{bmatrix} f_{ii}^2 a_{ii} + 2 f_{ii} f_{ji} a_{ij} + f_{ji}^2 a_{jj} & f_{ii} a_{ij} + f_{ji} a_{jj} \\ f_{ii} a_{ij} + f_{ji} a_{jj} & a_{jj} \end{bmatrix} \\
&= \begin{bmatrix} a_{ii} - \frac{2a_{ij} - (a_{ii} + a_{jj}) b_{ij}}{1 - b_{ij}^2} b_{ij} & \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} \\ \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} & a_{jj} \end{bmatrix},
\end{aligned}
\qquad (4)
$$

where we have used $f_{ii} = 1/\sqrt{1 - b_{ij}^2}$, $f_{ji} = -b_{ij}/\sqrt{1 - b_{ij}^2}$.

$$\hat{F}_2^T \hat{A} \hat{F}_2 = \begin{bmatrix} f_{ii} & f_{ji} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} f_{ii} & 0 \\ f_{ji} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} f_{ii}^2 a_{ii} + 2 f_{ii} f_{ji} a_{ij} + f_{ji}^2 a_{jj} & f_{ii} a_{ij} + f_{ji} a_{jj} \\ f_{ii} a_{ij} + f_{ji} a_{jj} & a_{jj} \end{bmatrix}$$

$$= \begin{bmatrix} a_{ii} - \frac{2a_{ij} - (a_{ii} + a_{jj}) b_{ij}}{1 - b_{ij}^2} b_{ij} & \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} \\ \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} & a_{jj} \end{bmatrix}, \qquad (4)$$

where we have used $f_{ii} = 1/\sqrt{1 - b_{ij}^2}$, $f_{ji} = -b_{ij}/\sqrt{1 - b_{ij}^2}$.

The final $\hat{F}$ has the form $\hat{F} = \hat{F}_2 \hat{J}$, where $\hat{J}$ is the Jacobi transformation which diagonalizes $\hat{F}_2^T \hat{A} \hat{F}_2$. Its angle ϑ is determined by the formula

The Algorithm Based on RR^T Factorization

$$\tan(2\vartheta) = \frac{2(a_{ij} - b_{ij}a_{jj})\sqrt{1 - b_{ij}^2}}{a_{ii} - a_{jj} - 2(a_{ij} - b_{ij}a_{jj})b_{ij}}, \quad -\frac{\pi}{4} \le \vartheta \le \frac{\pi}{4}.$$

The Algorithm Based on RR^T Factorization

$$\tan(2\vartheta) = \frac{2(a_{ij} - b_{ij}a_{jj})\sqrt{1-b_{ij}^2}}{a_{ii} - a_{jj} - 2(a_{ij} - b_{ij}a_{jj})b_{ij}}, \quad -\frac{\pi}{4} \le \vartheta \le \frac{\pi}{4}.$$

The transformation formulas for the diagonal elements of A read

$$a'_{ii} = a_{ii} - \frac{2a_{ij} - (a_{ii} + a_{jj})b_{ij}}{1 - b_{ij}^2} b_{ij} + \tan\vartheta \cdot \frac{a_{ij} - a_{jj}b_{ij}}{\sqrt{1-b_{ij}^2}}$$

$$a'_{jj} = a_{jj} - \tan\vartheta \cdot \frac{a_{ij} - a_{jj}b_{ij}}{\sqrt{1-b_{ij}^2}}$$

The Algorithm Based on RR^T Factorization

$$\tan(2\vartheta) = \frac{2(a_{ij} - b_{ij}a_{jj})\sqrt{1-b_{ij}^2}}{a_{ii} - a_{jj} - 2(a_{ij} - b_{ij}a_{jj})b_{ij}}, \quad -\frac{\pi}{4} \leq \vartheta \leq \frac{\pi}{4}.$$

The transformation formulas for the diagonal elements of A read

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

If $a_{ii} = a_{jj}$, $a_{ij} = a_{jj}b_{ij}$ then ϑ is determined from expression $0/0$, so we choose $\vartheta = 0$. In this case a'_{ii} and a'_{jj} reduce to a_{ii} and a_{jj}, respectively.

The Algorithm Based on RR^T Factorization

This leads to a simpler matrix

$$\begin{aligned}\hat{Z} &= \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} 1 & 0 \\ -b_{ij} & \sqrt{1-b_{ij}^2} \end{bmatrix} \begin{bmatrix} c_\vartheta & -s_\vartheta \\ s_\vartheta & c_\vartheta \end{bmatrix} \\ &= \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} c_\vartheta & -s_\vartheta \\ s_{\tilde{\vartheta}} & c_{\tilde{\vartheta}} \end{bmatrix}, \qquad \begin{aligned} c_{\tilde{\vartheta}} &= c_\vartheta \sqrt{1-b_{ij}^2} + s_\vartheta b_{ij}, \\ s_{\tilde{\vartheta}} &= s_\vartheta \sqrt{1-b_{ij}^2} - c_\vartheta b_{ij}. \end{aligned}\end{aligned}$$

It is easy to check that $c_{\tilde{\vartheta}}^2 + s_{\tilde{\vartheta}}^2 = 1$.

Some Remarks

- The algorithms based on LL^T and RR^T factorizations can be generalized to work with complex matrices

Some Remarks

- The algorithms based on LL^T and RR^T factorizations can be generalized to work with complex matrices
- All real algorithms have the form

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

This follows from a result of Gose (ZAMM 59, 1979), who found the general form of a matrix $\hat{Z}$ which diagonalizes a positive definite symmetric matrix $\hat{B}$ of order 2 via the congruence transformation $\hat{B} \mapsto \hat{Z}^T \hat{B} \hat{Z}$.

Some Remarks

- The algorithms based on LL^T and RR^T factorizations can be generalized to work with complex matrices
- All real algorithms have the form

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

This follows from a result of Gose (ZAMM 59, 1979), who found the general form of a matrix $\hat{Z}$ which diagonalizes a positive definite symmetric matrix $\hat{B}$ of order 2 via the congruence transformation $\hat{B} \mapsto \hat{Z}^T \hat{B} \hat{Z}$.
If we assume $b_{11} = \cdots = b_{nn}$ and the same condition for $\hat{Z}^T \hat{B} \hat{Z}$, then this form of $\hat{Z}$ is just the Gose's theorem. Later Hari generalized that result to complex matrices.

Global Convergence (Real and Complex Algorithm)

We have used the following measure in the convergence analysis:

$$S^2(A) = \|A - \text{diag}(A)\|_F^2, \quad S(A,B) = \left[S^2(A) + S^2(B)\right]^{1/2}.$$

The HZ method converges globally if

$$A^{(k)} \to \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n), \quad B^{(k)} \to I_n \qquad \text{as} \quad k \to \infty,$$

holds for any initial pair of symmetric matrices (A, B) with $B \succ O$.

Global Convergence (Real and Complex Algorithm)

We have used the following measure in the convergence analysis:

$$S^2(A) = \|A - \text{diag}(A)\|_F^2, \quad S(A,B) = \left[S^2(A) + S^2(B)\right]^{1/2}.$$

The HZ method converges globally if

$$A^{(k)} \to \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n), \quad B^{(k)} \to I_n \qquad \text{as} \quad k \to \infty,$$

holds for any initial pair of symmetric matrices (A, B) with $B \succ O$.

Actually, it is sufficient to show that $S(A,B) \to 0$ as $k \to \infty$.

Global Convergence (Real and Complex Algorithm)

We have used the following measure in the convergence analysis:

$$S^2(A) = \|A - \mathrm{diag}(A)\|_F^2, \quad S(A,B) = \left[S^2(A) + S^2(B)\right]^{1/2}.$$

The HZ method converges globally if

$$A^{(k)} \to \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad B^{(k)} \to I_n \qquad \text{as} \quad k \to \infty,$$

holds for any initial pair of symmetric matrices (A, B) with $B \succ O$.

Actually, it is sufficient to show that $S(A,B) \to 0$ as $k \to \infty$.

We have proved the global convergence for the serial pivot strategies.

Global Convergence (Real and Complex Algorithm)

We have used the following measure in the convergence analysis:

$$S^2(A) = \|A - \mathrm{diag}(A)\|_F^2, \quad S(A,B) = \left[S^2(A) + S^2(B)\right]^{1/2}.$$

The HZ method converges globally if

$$A^{(k)} \to \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad B^{(k)} \to I_n \qquad \text{as} \quad k \to \infty,$$

holds for any initial pair of symmetric matrices (A, B) with $B \succ O$.

Actually, it is sufficient to show that $S(A, B) \to 0$ as $k \to \infty$.

We have proved the global convergence for the serial pivot strategies.

We are adapting the proof to hold for a new much larger class of generalized serial strategies which includes the class of weak wavefront strategies.

Asymptotic Convergence (Real and Complex Algorithm)

Let (A, B) have simple eigenvalues:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_n, \qquad \mu = \max\{|\lambda_1|\,,\,|\lambda_n|\},$$

$$3\delta_i = \min_{\substack{1\le i\le n\\ j\ne i}} |\,\lambda_i - \lambda_j\,|, \quad 1 \le i \le n; \qquad \delta = \min_{1\le i\le n} \delta_i\,.$$

Asymptotic Convergence (Real and Complex Algorithm)

Let (A, B) have simple eigenvalues:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_n, \qquad \mu = \max\{|\lambda_1|, |\lambda_n|\},$$

$$3\delta_i = \min_{\substack{1 \le i \le n \\ j \ne i}} |\lambda_i - \lambda_j|, \quad 1 \le i \le n; \qquad \delta = \min_{1 \le i \le n} \delta_i .$$

Theorem

If $S(B^{(0)}) < \dfrac{1}{n(n-1)}$ *and* $S(A^{(0)}, B^{(0)}) < \dfrac{\delta}{2\sqrt{1+\mu^2}}$,
then for the general cyclic and for the serial strategies it holds, respectively:

$$S(A^{(N)}, B^{(N)}) \le \sqrt{N(1+\mu^2)}\,\frac{S^2(A^{(0)}, B^{(0)})}{\delta}, \quad N = n(n-1)/2$$

$$S(A^{(N)}, B^{(N)}) \le \sqrt{1+\mu^2}\,\frac{S^2(A^{(0)}, B^{(0)})}{\delta}.$$

Multiple Eigenvalues

The situation complicates because the positive definite pair (A, B) with multiple eigenvalues, and with nearly diagonal matrices, has special structure.

Multiple Eigenvalues

The situation complicates because the positive definite pair (A, B) with multiple eigenvalues, and with nearly diagonal matrices, has special structure.

Let $\quad A = A^*$ with $a_{11} \geq a_{22} \geq \cdots \geq a_{nn}$,

$\quad\quad B = B^*$ with $B \succ O$, $\mathrm{diag}(B) = I_n$.

Multiple Eigenvalues

The situation complicates because the positive definite pair (A, B) with multiple eigenvalues, and with nearly diagonal matrices, has special structure.

Let $\quad A = A^*$ with $a_{11} \geq a_{22} \geq \cdots \geq a_{nn}$,

$\quad B = B^*$ with $B \succ O$, $\mathrm{diag}(B) = I_n$.

Let

$$\lambda_1 = \cdots = \lambda_{s_1} > \lambda_{s_1+1} = \cdots = \lambda_{s_2} > \cdots > \lambda_{s_{p-1}+1} = \cdots = \lambda_{s_p},$$

where $s_p = n$.

Multiple Eigenvalues

The situation complicates because the positive definite pair (A, B) with multiple eigenvalues, and with nearly diagonal matrices, has special structure.

Let $\quad A = A^*$ with $a_{11} \geq a_{22} \geq \cdots \geq a_{nn}$,

$\quad B = B^*$ with $B \succ O$, $\mathrm{diag}(B) = I_n$.

Let

$$\lambda_1 = \cdots = \lambda_{s_1} > \lambda_{s_1+1} = \cdots = \lambda_{s_2} > \cdots > \lambda_{s_{p-1}+1} = \cdots = \lambda_{s_p},$$

where $s_p = n$. Then

$$n_i = s_i - s_{i-1}, \qquad 1 \leq i \leq p \quad (s_0 = 0),$$

n_i is the multiplicity of λ_{s_i}. Again, let $\mu = \max\{|\lambda_{s_1}|, |\lambda_{s_p}|\}$.

Multiple Eigenvalues

The minimum distance between two distinct eigenvalues plays special role in the analysis. Let δ_r be the absolute gap (separation) of λ_{s_r} from other eigenvalues,

$$3\delta_r = \min_{\substack{1\le t\le p \\ t\ne r}} |\ \lambda_{s_r} - \lambda_{s_t}\ |, \quad 1 \le r \le p.$$

Multiple Eigenvalues

The minimum distance between two distinct eigenvalues plays special role in the analysis. Let δ_r be the absolute gap (separation) of λ_{s_r} from other eigenvalues,

$$3\delta_r = \min_{\substack{1 \leq t \leq p \\ t \neq r}} \mid \lambda_{s_r} - \lambda_{s_t} \mid, \quad 1 \leq r \leq p.$$

Then $$\delta = \min_{1 \leq r \leq p} \delta_r$$ is the minimum absolute gap.

Multiple Eigenvalues

Next we consider the following matrix block-partition

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{p1} & \cdots & A_{pp} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & \cdots & B_{1p} \\ \vdots & \ddots & \vdots \\ B_{p1} & \cdots & B_{pp} \end{bmatrix},$$

A_{rt}, B_{rt} are $n_r \times n_t$ blocks.
For a square matrix $X = (X_{rt})$ partitioned according to $n_1, \ldots, n_p$, let

$$\tau(X) = \|X - \mathrm{diag}(X_{11}, \ldots, X_{pp})\|_F.$$

For our positive definite pair (A, B), let

$$\tau(A, B) = \left[\tau^2(A) + \tau^2(B)\right]^{1/2}$$

Theorem (Hari 91)

Let $D_r + E_r = A - \lambda_{s_r} B$, $diag(E_r) = 0$, $1 \leq r \leq p$. *If*

$$\|E_r\|_2 < \delta_r, \quad 1 \leq r \leq p,$$

then

$$\|A_{rr} - \lambda_{s_r} B_{rr}\|_F \leq \frac{1}{\delta_r} \sum_{\substack{t=1 \\ t \neq r}}^{p} \|A_{rt} - \lambda_{s_r} B_{rt}\|_F^2, \quad 1 \leq r \leq p$$

and

$$\sum_{s=1}^{n} \left| \frac{a_{ss}}{b_{ss}} - \lambda_s \right|^2 \leq \sum_{r=1}^{p} \|A_{rr} - \lambda_{s_r} B_{rr}\|_F^2 \leq \left[\frac{(1+\mu^2)\tau^2(A,B)}{\delta} \right]^2.$$

Multiple Eigenvalues

Let us return to the method.

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k.

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B).

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$.

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \leq \epsilon$.

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \le \epsilon$.
Then the theorem implies

$$A_{rr} = \lambda_{s_r} B_{rr} + F_{rr}, \qquad \|F_r\|_F = \mathcal{O}(\tau^2), \quad 1 \le r \le p.$$

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \leq \epsilon$.
Then the theorem implies

$$A_{rr} = \lambda_{s_r} B_{rr} + F_{rr}, \qquad \|F_r\|_F = \mathcal{O}(\tau^2), \quad 1 \leq r \leq p.$$

If the pivot element a_{ij} (b_{ij}) is within the diagonal block A_{rr} (B_{rr}), then we shall have:

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \leq \epsilon$.
Then the theorem implies

$$A_{rr} = \lambda_{s_r} B_{rr} + F_{rr}, \qquad \|F_r\|_F = \mathcal{O}(\tau^2), \quad 1 \leq r \leq p.$$

If the pivot element a_{ij} (b_{ij}) is within the diagonal block A_{rr} (B_{rr}), then we shall have:

- Huge cancelations in the numerator and denominator when computing

$$\tan(2\theta) = \frac{2a_{ij} - (a_{ii} + a_{jj})\, b_{ij}}{\sqrt{1 - (b_{ij})^2}\,(a_{ii} - a_{jj})} = \frac{\mathcal{O}(\tau^2)}{\mathcal{O}(\tau^2)}$$

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \leq \epsilon$.
Then the theorem implies

$$A_{rr} = \lambda_{s_r} B_{rr} + F_{rr}, \qquad \|F_r\|_F = \mathcal{O}(\tau^2), \quad 1 \leq r \leq p.$$

If the pivot element a_{ij} (b_{ij}) is within the diagonal block A_{rr} (B_{rr}), then we shall have:

- Huge cancelations in the numerator and denominator when computing

$$\tan(2\theta) = \frac{2a_{ij} - (a_{ii} + a_{jj})\, b_{ij}}{\sqrt{1 - (b_{ij})^2}\,(a_{ii} - a_{jj})} = \frac{\mathcal{O}(\tau^2)}{\mathcal{O}(\tau^2)}$$

- Possibly large θ when ϵ and τ are tiny.

Multiple Eigenvalues

Let us return to the method. Let (A, B) be obtained at step k. Suppose that k is large enough, so that the last theorem holds for (A, B). Let $\tau = \tau(A, B)$, $\epsilon = S(A, B)$. Note that $\tau \leq \epsilon$.
Then the theorem implies

$$A_{rr} = \lambda_{s_r} B_{rr} + F_{rr}, \qquad \|F_r\|_F = \mathcal{O}(\tau^2), \quad 1 \leq r \leq p.$$

If the pivot element a_{ij} (b_{ij}) is within the diagonal block A_{rr} (B_{rr}), then we shall have:

- Huge cancelations in the numerator and denominator when computing

$$\tan(2\theta) = \frac{2a_{ij} - (a_{ii} + a_{jj})\, b_{ij}}{\sqrt{1 - (b_{ij})^2}\, (a_{ii} - a_{jj})} = \frac{\mathcal{O}(\tau^2)}{\mathcal{O}(\tau^2)}$$

- Possibly large θ when ϵ and τ are tiny.

This impacts asymptotic convergence and accuracy of the algorithm.

Multiple Eigenvalues

$$N = \frac{n(n-1)}{2}, \qquad M = N - \sum_{r=1}^{p} \frac{n_r(n_r-1)}{2}, \qquad n_{max} = \max_{1 \le r \le p} n_r$$

Let ϵ_N and τ_N denote ϵ and τ for the pair obtained after applying one sweep of the column-cyclic HZ method. If (A, B) satisfies $n \geq 3$, $p \geq 2$,

$$S(B) < \frac{1}{n(n-1)}, \quad \sqrt{1+\mu^2}\epsilon < \min\left\{\frac{1}{2}, \sqrt{\frac{\delta}{\mu+1}}\right\}\delta,$$

then

- $\tau_N \leq \frac{3}{2}\sqrt{2.31^M \cdot n_{max}(1+\mu^2)}\,\frac{\epsilon}{\delta}\tau$
- $\tau_N \leq \frac{3}{2}\sqrt{n_{max}(1+\mu^2)}\,\frac{\epsilon^2}{\delta}$
- if $n_{max} = 2$ then $\epsilon_N \leq \frac{18}{17}\sqrt{1+\mu^2}\frac{\epsilon^2}{\delta}$.

Goals

Goals

- To solve relative accuracy and quadratic convergence issues/problems

Goals

- To solve relative accuracy and quadratic convergence issues/problems
- To derive a sound quadratically convergent method

Goals

- To solve relative accuracy and quadratic convergence issues/problems
- To derive a sound quadratically convergent method
- To prove high relative accuracy of such a method when the both matrices A and B are positive definite

Goals

- To solve relative accuracy and quadratic convergence issues/problems
- To derive a sound quadratically convergent method
- To prove high relative accuracy of such a method when the both matrices A and B are positive definite
- To prove the global convergence of such a method under the large class of generalized serial strategies

Goals

- To solve relative accuracy and quadratic convergence issues/problems
- To derive a sound quadratically convergent method
- To prove high relative accuracy of such a method when the both matrices A and B are positive definite
- To prove the global convergence of such a method under the large class of generalized serial strategies
- To derive a block method for PGEP

Goals

- To solve relative accuracy and quadratic convergence issues/problems
- To derive a sound quadratically convergent method
- To prove high relative accuracy of such a method when the both matrices A and B are positive definite
- To prove the global convergence of such a method under the large class of generalized serial strategies
- To derive a block method for PGEP
- To prove at least the global convergence of the block method

Block-Matrix Partition

To define a block method for PGEP we start from a partition π of n

$$\pi = (n_1, n_2, \ldots, n_m), \qquad n_1 + n_2 + \cdots + n_m = n, \quad n_i \geq 1.$$

Block-Matrix Partition

To define a block method for PGEP we start from a partition π of n

$$\pi = (n_1, n_2, \ldots, n_m), \qquad n_1 + n_2 + \cdots + n_m = n, \quad n_i \geq 1.$$

The partition π defines block-matrix partition of any square matrix A of order n:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{bmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_m \end{matrix} \quad , \quad A_{ij} \in \mathbf{R}^{i \times j}$$

Elementary Block Matrix

Elementary block matrix $\mathbf{E}_{ij}$ is a nonsingular $n \times n$ matrix

$$\mathbf{E}_{ij} = \begin{bmatrix} I & & & & \\ & E_{ii} & & E_{ij} & \\ & & I & & \\ & E_{ji} & & E_{jj} & \\ & & & & I \end{bmatrix} \begin{matrix} \\ n_i \\ \\ n_j \\ \\ \end{matrix},$$

which carries the block-matrix partition defined by π.

Block Jacobi Method for PGEP

Block Jacobi method for PGEP is iterative process of the form

$$A^{(k+1)} = F_k^T A^{(k)} F_k, \quad B^{(k+1)} = F_k^T B^{(k)} F_k, \quad k \geq 0;$$

where

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

and F_k, $k \geq 0$, are elementary block matrices.

Block Jacobi Method for PGEP

Block Jacobi method for PGEP is iterative process of the form

$$A^{(k+1)} = F_k^T A^{(k)} F_k, \quad B^{(k+1)} = F_k^T B^{(k)} F_k, \quad k \geq 0;$$

where

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

and F_k, $k \geq 0$, are elementary block matrices.

Here (A, B) is the initial positive definite pair of symmetric matrices:

$$A = A^T, \qquad B = B^T, \qquad B \succ O$$

Block Jacobi method for PGEP is iterative process of the form

$$A^{(k+1)} = F_k^T A^{(k)} F_k, \quad B^{(k+1)} = F_k^T B^{(k)} F_k, \quad k \geq 0;$$

where

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

and F_k, $k \geq 0$, are elementary block matrices.

Here (A, B) is the initial positive definite pair of symmetric matrices:

$$A = A^T, \qquad B = B^T, \qquad B \succ O$$

All matrices carry block-matrix partition defined by π: $A^{(k)} = (A_{rs}^{(k)})$, $B^{(k)} = (B_{rs}^{(k)})$, $F_k = (F_{rs}^{(k)})$, $k \geq 0$.

Block Jacobi Method for PGEP

At step k, the pivot pair (i,j), $i < j$ is selected according to a given pivot strategy. Note that $i = i(k)$, $j = j(k)$.

Block Jacobi Method for PGEP

At step k, the pivot pair (i, j), $i < j$ is selected according to a given pivot strategy. Note that $i = i(k)$, $j = j(k)$.

The pivot pair selects the pivot submatrices $\hat{A}_{ij}^{(k)}$, $\hat{B}_{ij}^{(k)}$ which are to be diagonalized:

Block Jacobi Method for PGEP

At step k, the pivot pair (i, j), $i < j$ is selected according to a given pivot strategy. Note that $i = i(k)$, $j = j(k)$.

The pivot pair selects the pivot submatrices $\hat{A}_{ij}^{(k)}$, $\hat{B}_{ij}^{(k)}$ which are to be diagonalized:

$$\begin{bmatrix} A_{ii}^{(k+1)} & 0 \\ 0 & A_{jj}^{(k+1)} \end{bmatrix} = \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}^T \begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ (A_{ij}^{(k)})^T & A_{jj}^{(k)} \end{bmatrix} \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}$$

$$\hat{A}_{ij}^{(k+1)} = \hat{F}_k^* \, \hat{A}^{(k)} \, \hat{F}_k,$$

Block Jacobi Method for PGEP

At step k, the pivot pair (i, j), $i < j$ is selected according to a given pivot strategy. Note that $i = i(k)$, $j = j(k)$.

The pivot pair selects the pivot submatrices $\hat{A}_{ij}^{(k)}$, $\hat{B}_{ij}^{(k)}$ which are to be diagonalized:

$$\begin{bmatrix} A_{ii}^{(k+1)} & 0 \\ 0 & A_{jj}^{(k+1)} \end{bmatrix} = \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}^T \begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ (A_{ij}^{(k)})^T & A_{jj}^{(k)} \end{bmatrix} \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}$$

$$\hat{A}_{ij}^{(k+1)} = \hat{F}_k^* \, \hat{A}^{(k)} \, \hat{F}_k,$$

and similar for $\hat{B}^{(k)}$:

$$\hat{B}_{ij}^{(k+1)} = \hat{F}_k^T \, \hat{B}^{(k)} \, \hat{F}_k, \quad k \geq 0;$$

Block Jacobi Method for PGEP

At step k, the pivot pair (i, j), $i < j$ is selected according to a given pivot strategy. Note that $i = i(k)$, $j = j(k)$.

The pivot pair selects the pivot submatrices $\hat{A}_{ij}^{(k)}$, $\hat{B}_{ij}^{(k)}$ which are to be diagonalized:

$$\begin{bmatrix} A_{ii}^{(k+1)} & 0 \\ 0 & A_{jj}^{(k+1)} \end{bmatrix} = \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}^T \begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ (A_{ij}^{(k)})^T & A_{jj}^{(k)} \end{bmatrix} \begin{bmatrix} F_{ii}^{(k)} & F_{ij}^{(k)} \\ F_{ji}^{(k)} & F_{jj}^{(k)} \end{bmatrix}$$

$$\hat{A}_{ij}^{(k+1)} = \hat{F}_k^* \, \hat{A}^{(k)} \, \hat{F}_k,$$

and similar for $\hat{B}^{(k)}$:

$$\hat{B}_{ij}^{(k+1)} = \hat{F}_k^T \, \hat{B}^{(k)} \, \hat{F}_k, \quad k \geq 0;$$

$n_1 = n_2 = \cdots = n_m = 1 \longrightarrow$ standard (element-wise) method

Preliminary Transformation

Recall: element-wise methods maintained: $b_{11} = \cdots = b_{nn} = 1.$

Preliminary Transformation

Recall: element-wise methods maintained: $b_{11} = \cdots = b_{nn} = 1$.
At each step of the block method a much smaller PGEP has to be solved:

$$\hat{A}^{(k)} x = \lambda \hat{B}^{(k)} x \quad \text{i. e.} \quad (\hat{A}^{(k)}, \hat{B}^{(k)}) \mapsto (\text{diag}, I_{n_i + n_j}).$$

Preliminary Transformation

Recall: element-wise methods maintained: $b_{11} = \cdots = b_{nn} = 1$.
At each step of the block method a much smaller PGEP has to be solved:

$$\hat{A}^{(k)}x = \lambda \hat{B}^{(k)}x \quad \text{i. e.} \quad (\hat{A}^{(k)}, \hat{B}^{(k)}) \mapsto (\text{diag}, I_{n_i+n_j}).$$

To simplify the algorithm we need a preliminary transformation which makes both:

- transforms the diagonal elements of B to ones and
- diagonalizes all diagonal blocks of A and B.

Preliminary Transformation

- Set: $D^{(0)} = \mathrm{diag}(\frac{1}{\sqrt{b_{11}}}, \ldots, \frac{1}{\sqrt{b_{nn}}})$ (i.e. $\mathrm{diag}(D^{(0)}BD^{(0)}) = I_n$).

Preliminary Transformation

- Set: $D^{(0)} = \mathrm{diag}(\frac{1}{\sqrt{b_{11}}}, \ldots, \frac{1}{\sqrt{b_{nn}}})$ (i.e. $\mathrm{diag}(D^{(0)} B D^{(0)}) = I_n$).
- Let $D^{(0)} = \mathrm{diag}(D_{11}^{(0)}, D_{22}^{(0)}, \ldots, D_{mm}^{(0)})$ and

$$\tilde{A}_{rr} = D_{rr}^{(0)} A_{rr} D_{rr}^{(0)}, \quad \tilde{B}_{rr} = D_{rr}^{(0)} B_{rr} D_{rr}^{(0)}, \quad 1 \leq r \leq m.$$

Preliminary Transformation

- Set: $D^{(0)} = \text{diag}(\frac{1}{\sqrt{b_{11}}}, \ldots, \frac{1}{\sqrt{b_{nn}}})$ (i.e. $\text{diag}(D^{(0)} B D^{(0)}) = I_n$).
- Let $D^{(0)} = \text{diag}(D_{11}^{(0)}, D_{22}^{(0)}, \ldots, D_{mm}^{(0)})$ and

$$\tilde{A}_{rr} = D_{rr}^{(0)} A_{rr} D_{rr}^{(0)}, \quad \tilde{B}_{rr} = D_{rr}^{(0)} B_{rr} D_{rr}^{(0)}, \quad 1 \le r \le m.$$

Apply to each pair $(\tilde{A}_{rr}, \tilde{B}_{rr})$ the HZ (or similar) method to obtain F_{rr}:

$$F_{rr}^T \tilde{A}_{rr} F_{rr} = A_{rr}^{(0)} = \text{diag}, \qquad F_{rr}^T \tilde{B}_{rr} F_{rr} = I_{n_r}, \quad 1 \le r \le m.$$

Set: $F_0 = \text{diag}(F_{11}, F_{22}, \ldots, F_{mm})$.

Preliminary Transformation

- Set: $D^{(0)} = \text{diag}(\frac{1}{\sqrt{b_{11}}}, \ldots, \frac{1}{\sqrt{b_{nn}}})$ (i.e. $\text{diag}(D^{(0)} B D^{(0)}) = I_n$).
- Let $D^{(0)} = \text{diag}(D_{11}^{(0)}, D_{22}^{(0)}, \ldots, D_{mm}^{(0)})$ and

$$\tilde{A}_{rr} = D_{rr}^{(0)} A_{rr} D_{rr}^{(0)}, \quad \tilde{B}_{rr} = D_{rr}^{(0)} B_{rr} D_{rr}^{(0)}, \quad 1 \le r \le m.$$

Apply to each pair $(\tilde{A}_{rr}, \tilde{B}_{rr})$ the HZ (or similar) method to obtain F_{rr}:

$$F_{rr}^T \tilde{A}_{rr} F_{rr} = A_{rr}^{(0)} = \text{diag}, \qquad F_{rr}^T \tilde{B}_{rr} F_{rr} = I_{n_r}, \quad 1 \le r \le m.$$

Set: $F_0 = \text{diag}(F_{11}, F_{22}, \ldots, F_{mm})$.

- Perform: $A^{(0)} = F_0^T D_0 A D_0 F_0, \qquad B^{(0)} = F_0^T D_0 B D_0 F_0.$

Preliminary Transformation

The preliminary transformation ensures that

$$(\hat{A}^{(k)}, \hat{B}^{(k)}) = \left(\begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ A_{ji}^{(k)} & A_{jj}^{(k)} \end{bmatrix}, \begin{bmatrix} I_{n_i} & B_{ij}^{(k)} \\ B_{ji}^{(k)} & I_{n_j} \end{bmatrix} \right), \quad k \geq 0,$$

with diagonal blocks $A_{ii}^{(k)}$ and $A_{jj}^{(k)}$. This form makes it easier to apply the element-wise HZ (or similar) algorithm which we call here the kernel algorithm.

Preliminary Transformation

The preliminary transformation ensures that

$$(\hat{A}^{(k)}, \hat{B}^{(k)}) = \left(\begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ A_{ji}^{(k)} & A_{jj}^{(k)} \end{bmatrix}, \begin{bmatrix} I_{n_i} & B_{ij}^{(k)} \\ B_{ji}^{(k)} & I_{n_j} \end{bmatrix} \right), \quad k \geq 0,$$

with diagonal blocks $A_{ii}^{(k)}$ and $A_{jj}^{(k)}$. This form makes it easier to apply the element-wise HZ (or similar) algorithm which we call here the kernel algorithm.

For the complete description of the block HZ method one has to specify the pivot strategy and the stopping criterion.

Preliminary Transformation

The preliminary transformation ensures that

$$(\hat{A}^{(k)}, \hat{B}^{(k)}) = \left(\begin{bmatrix} A_{ii}^{(k)} & A_{ij}^{(k)} \\ A_{ji}^{(k)} & A_{jj}^{(k)} \end{bmatrix}, \begin{bmatrix} I_{n_i} & B_{ij}^{(k)} \\ B_{ji}^{(k)} & I_{n_j} \end{bmatrix} \right), \quad k \geq 0,$$

with diagonal blocks $A_{ii}^{(k)}$ and $A_{jj}^{(k)}$. This form makes it easier to apply the element-wise HZ (or similar) algorithm which we call here the kernel algorithm.

For the complete description of the block HZ method one has to specify the pivot strategy and the stopping criterion.

For the latter, one can try with $S(A, B) \leq \|A\|_F \epsilon$ or with $S(A_S, B) \leq \epsilon$ where $A_S = \Delta A \Delta$ with diagonal Δ which makes $\mathrm{diag}(|A_S|) = I_n$. However, these are yet open problems as are all those concerning the global and asymptotic convergence and high relative accuracy.

THANK YOU.