On Jacobi Methods for the Positive Definite Generalized Eigenvalue Problem

Vjeran Hari

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

SIAM Annual Meeting, July 10–14, 2017, Pittsburgh, USA

OUTLINE

OUTLINE

- GEP and PGEP
- Derivation of the algorithms

OUTLINE

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

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

OUTLINE

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

OUTLINE

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

- We have restricted our attention to element-wise, two-sided Jacobi-type methods for PGEP since they can be used standalone or as kernel algorithms for the block methods.

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

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)

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

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

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

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

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

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

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

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

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

where $I_n = [e_1, \ldots, e_n]$.

How to solve PGEP?

One can reduce PGEP to the standard EP for one symmetric matrix

$$(A, B) \mapsto (L^{-1}AL^{-T}, I), \quad B = LL^T.$$

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

How to solve PGEP?

One can reduce PGEP to the standard EP for one symmetric matrix

$$(A, B) \mapsto (L^{-1}AL^{-T}, I), \quad B = LL^T.$$

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

One can try to maximize the smallest 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 reduce PGEP to the standard EP for one symmetric matrix

$$(A, B) \mapsto (L^{-1}AL^{-T}, I), \quad B = LL^T.$$

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

One can try to maximize the smallest 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

So far we know of two diagonalization methods for PGEP

Jacobi Methods for PGEP

So far we know of 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

So far we know of 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):

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.*

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.*

Since the derivation of the HZ method has not yet been published, we shall devote few slides to its derivation.

Derivation of the HZ Method

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

Derivation of the HZ Method

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 $b_{11}^{(0)} = b_{22}^{(0)} = \cdots = b_{nn}^{(0)} = 1.$

Derivation of the HZ Method

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:

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

Derivation of the HZ Method

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

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

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

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

Each Z_k is a nonsingular elementary plane matrix

Derivation of the HZ Method

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

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

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

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

Each Z_k is a nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & c_k & & -s_k & \\ & & I & & \\ & \tilde{s}_k & & \tilde{c}_k & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}, \qquad c_k^2 + s_k^2 = \tilde{c}_k^2 + \tilde{s}_k^2 = 1/\sqrt{1 - b_{i(k)j(k)}^2},$$

Derivation of the HZ Method

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:

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

Each Z_k is a nonsingular elementary plane matrix

$$Z_k = \begin{bmatrix} I & & & & \\ & c_k & & -s_k & \\ & & I & & \\ & \tilde{s}_k & & \tilde{c}_k & \\ & & & & I \end{bmatrix} \begin{matrix} \\ i(k) \\ \\ j(k) \\ \\ \end{matrix}, \qquad c_k^2 + s_k^2 = \tilde{c}_k^2 + \tilde{s}_k^2 = 1/\sqrt{1 - b_{i(k)j(k)}^2},$$

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

Derivation of the HZ Method

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

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

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

Derivation of the HZ Method

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

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

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

Then $A' = Z^T A Z, \quad B' = Z^T B Z$ implies

$$\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z}.$$

Derivation of the HZ Method

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

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

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

Then $A' = Z^T A Z, \quad B' = Z^T B Z$ implies

$$\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z}.$$

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

Derivation of the HZ Method

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

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

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

Then $A' = Z^T A Z, \quad B' = Z^T B Z$ implies

$$\hat{A}' = \hat{Z}^T \hat{A} \hat{Z}, \quad \hat{B}' = \hat{Z}^T \hat{B} \hat{Z}.$$

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

$\downarrow$ $\hat{B} \to \mathrm{diag}$ $\qquad$ $\downarrow$ $\hat{B} \to I_2$ $\qquad$ $\downarrow$ $\hat{A} \to \mathrm{diag}$

$\hat{Z}$ 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 approaches yield the same algorithm.

Essential Part of the Algorithm

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

Essential Part of the Algorithm

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

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

Essential Part of the Algorithm

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

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

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

Essential Part of the Algorithm

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

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

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

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

Essential Part of the Algorithm

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

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

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

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

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

There are more formulas!

$$2\rho\xi = b_{ij}, \quad |\xi| \le \sqrt{2}/2 \le \rho \le 1.$$

$$\begin{aligned}
\cos\phi\sin\psi &= \cos\theta\sin\theta - \rho\xi = 0.5\,(\sin 2\theta - b_{ij}),\\
\cos\psi\sin\phi &= \cos\theta\sin\theta + \rho\xi = 0.5\,(\sin 2\theta + b_{ij}),\\
\cos\phi\cos\psi &= \rho^2\cos^2\theta - \xi^2\sin^2\theta,\\
\sin\phi\sin\psi &= \rho^2\sin^2\theta - \xi^2\cos^2\theta.
\end{aligned}$$

$$\begin{aligned}
\min\{\cos\phi\,,\,\cos\psi\} &\ge \rho\cos\theta - \frac{|b_{ij}|}{2\rho}|\sin\theta| \ge (\rho - \frac{|b_{ij}|}{2\rho})\cos\theta > 0,\\
\max\{\cos\phi\,,\,\cos\psi\} &= \rho\cos\theta + |\xi\sin\theta| \ge \cos(\theta) \ge \frac{\sqrt{2}}{2}.
\end{aligned}$$

There are more formulas!

Let

$$\sin\gamma = b_{ij}, \qquad \cos\gamma = \sqrt{1-b_{ij}^2}.$$

Then we have

$$\frac{1}{\cos\gamma}\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj}\end{bmatrix}\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\psi & \cos\psi\end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix} a'_{ii} & \\ & a'_{jj}\end{bmatrix},$$

$$\frac{1}{\cos\gamma}\begin{bmatrix} 1 & b_{ij} \\ b_{ij} & 1\end{bmatrix}\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\psi & \cos\psi\end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\phi & \cos\phi\end{bmatrix},$$

$$\begin{aligned} \cos\gamma &= \frac{\cos\phi}{\cos\psi} + b_{ij}\tan\psi = \frac{\cos\psi}{\cos\phi} - b_{ij}\tan\phi, \\ 2\cos(\phi+\psi)a_{ij} &= a_{ii}\sin(2\phi) - a_{jj}\sin(2\psi). \end{aligned}$$

There are more formulas!

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

We also have

$$
\begin{aligned}
\phi + \psi &= 2\theta \\
\phi - \psi &= \gamma
\end{aligned}, \qquad \text{hence} \qquad
\begin{aligned}
\phi &= \theta + \gamma/2 \\
\psi &= \theta - \gamma/2
\end{aligned}.
$$

All these relations are used in the global convergence proof and in the proof of high relative accuracy of the method.

Algorithm HZ

select the pivot pair (i,j)

if $a_{ij} \neq 0$ or $b_{ij} \neq 0$ then

$\rho = 0.5\,(\sqrt{1+b_{ij}} + \sqrt{1-b_{ij}});\quad \xi = b_{ij}/(2\rho);$

$\tau = \sqrt{(1+b_{ij})(1-b_{ij})};\quad t2 = 2a_{ij} - (a_{ii}+a_{jj})b_{ij};$

if $t2 = 0$ then $t = 0;$

else

$ct2 = \tau\,(a_{ii} - a_{jj})/t2;$

$t = \mathrm{sign}(ct2)/(\mathrm{abs}(ct2) + (1 + \sqrt{1+ct2^2});$

end

$cs = 1/\sqrt{1+t^2};\quad sn = t/\sqrt{1+t^2};$

$c1 = (\rho \cdot cs - \xi \cdot sn)/\tau;\quad s1 = (\rho \cdot sn + \xi \cdot cs)/\tau;$

$c2 = (\rho \cdot cs + \xi \cdot sn)/\tau;\quad s2 = (\rho \cdot sn - \xi \cdot cs)/\tau;$

$\delta_i = (b_{ij}/\tau - s1)(b_{ij}/\tau + s1)a_{ii} + (2c1\,a_{ij} + s2\,a_{jj})\,s2;$

$\delta_j = (s2 - b_{ij}/\tau)(s2 + b_{ij}/\tau)\,a_{jj} + (2c2\,a_{ij} - s1\,a_{ii})\,s1;$

$a'_{ij} = (c1\,c2 - s1\,s2)a_{ij} + (c2\,s2\,a_{jj} - c1\,s1\,a_{ii});\quad a'_{ji} = a'_{ij};$

$b'_{ij} = 0;\quad b'_{ji} = b'_{ij};\quad a'_{ii} = a_{ii} + \delta_i;\quad a'_{jj} = a_{jj} - \delta_j;$

for $k = 1,\ldots,n,\ k \neq i,j$ do

$a'_{ki} = c1 \cdot a_{ki} + s2 \cdot a_{kj};\quad b'_{ki} = c1 \cdot b_{ki} + s2 \cdot b_{kj};\quad a'_{ik} = a'_{ki};\quad b'_{ik} = b'_{ki};$

$a'_{kj} = c2 \cdot a_{kj} - s1 \cdot a_{ki};\quad b'_{kj} = c2 \cdot b_{kj} - s1 \cdot b_{ki};\quad a'_{jk} = a'_{kj};\quad b'_{jk} = b'_{kj};$

endfor

endif

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

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

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

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

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

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

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

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

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

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

Assuming $c > 0$, one obtains $\quad a = b_{ij}, \; c = \sqrt{1 - b_{ij}^2}, \quad$ 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 $\hat{B}$: $\hat{B} = \hat{L}\hat{L}^T$,

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

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

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

Let the first transformation be

$$\hat{F}_1 = \hat{L}^{-T}, \qquad \text{then} \quad \hat{F}_1^T \hat{B} \hat{F}_1 = I_2$$

The Algorithm Based on LL^T Factorization

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

The Algorithm Based on LL^T Factorization

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

The final $\hat{F}$,

$$\hat{F} \; = \; \hat{F}_1 \hat{R}, \qquad \hat{R} \text{ is Jacobi rotation which diagonalizes } \; \hat{F}_1^T \hat{A} \hat{F}_1.$$

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

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

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

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

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

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

If $a_{ii} = a_{jj}$, $a_{ij} = a_{ii}b_{ij}$ then ϑ is determined from $0/0$, 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

$$\hat{Z} = \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} \sqrt{1-b_{ij}^2} & -b_{ij} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_\vartheta & -s_\vartheta \\ s_\vartheta & c_\vartheta \end{bmatrix}$$

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

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

Algorithm $LL^T J$

```
select the pivot pair (i, j)
if a_ij ≠ 0 or b_ij ≠ 0 then
```

$$
\begin{aligned}
&\texttt{if } a_{ij} \neq 0 \texttt{ or } b_{ij} \neq 0 \texttt{ then}\\
&\quad \beta = b_{ij}\,,\ \ \tau = \texttt{sqrt}((1+\beta)(1-\beta)); \quad \alpha = a_{ij} - \beta a_{ii}\,;\\
&\qquad \texttt{if} \quad \alpha = 0 \quad \texttt{then} \quad t = 0;\\
&\qquad \texttt{else} \quad ct2 = (0.5\,(a_{ii} - a_{jj}) + \alpha\beta)/(\alpha\,\tau);\\
&\qquad\qquad t = \texttt{sign}(ct2)/(\texttt{abs}(ct2) + \texttt{sqrt}(1 + ct2^2));\\
&\qquad \texttt{endif}\\
&\qquad cs = 1/\texttt{sqrt}(1+t^2); \quad sn = t/\texttt{sqrt}(1+t^2);\\
&\qquad c1 = cs - sn\,\beta/\tau\,; \quad s1 = sn + cs\,\beta/\tau\,; \quad c2 = cs/\tau\,; \quad s2 = sn/\tau\,;\\
&\qquad \delta_i = t\alpha/\tau\,; \quad \delta_j = (\,t\alpha + (\beta/\tau)\cdot(2a_{ij} - (a_{ii} + a_{jj})\beta)\,)/\tau\,;\\
&\qquad a'_{ij} = (c1\,c2 - s1\,s2)\,a_{ij} + (c2\,s2\,a_{jj} - c1\,s1\,a_{ii}); \quad a'_{ji} = a'_{ij}\,;\\
&\qquad b'_{ij} = (c1\,c2 - s1\,s2)\,\beta + (c2\,s2 - c1\,s1); \quad b'_{ji} = b'_{ij}\,;\\
&\qquad a'_{ii} = a_{ii} + \delta_i\,; \quad a'_{j} = a_{jj} - \delta_j\,;\\
&\qquad \texttt{for } k = 1, \ldots, n,\ k \neq i, j \quad \texttt{do}\\
&\qquad\quad a'_{ki} = c1 \cdot a_{ki} + s2 \cdot a_{kj}; \quad b'_{ki} = c1 \cdot b_{ki} + s2 \cdot b_{kj}; \quad a'_{ik} = a'_{ki}; \quad b'_{ik} = b'_{ki}\\
&\qquad\quad a'_{kj} = c2 \cdot a_{kj} - s1 \cdot a_{ki}; \quad b'_{kj} = c2 \cdot b_{kj} - s1 \cdot b_{ki}; \quad a'_{jk} = a'_{kj}; \quad b'_{jk} = b'_{kj}\,;\\
&\qquad \texttt{endfor}\\
&\texttt{endif}
\end{aligned}
$$

The Algorithm Based on RR^T Factorization

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

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 Factorization

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

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

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

Let $\hat{F}_2 = \hat{R}^{-T}$, then $\hat{F}_2^T \hat{B} \hat{F}_2 = I_2$

The Algorithm Based on RR^T Factorization

$$\hat{F}_2^T \hat{A} \hat{F}_2 = \begin{bmatrix} a_{ii} - \frac{2a_{ij} - (a_{ii} + a_{jj}) b_{ij}}{1 - b_{ij}^2} b_{ij} & \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} \\ \frac{a_{ij} - b_{ij} a_{jj}}{\sqrt{1 - b_{ij}^2}} & a_{jj} \end{bmatrix}.$$

The final $\hat{F}$,

$$\hat{F} = \hat{F}_2 \hat{J}, \qquad \hat{J} \text{ is Jacobi rotation which diagonalizes } \hat{F}_2^T \hat{A} \hat{F}_2.$$

Its angle ϑ is determined by the formula:

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

The Algorithm Based on RR^T Factorization

The transformation formulas for the diagonal elements of A:

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

The Algorithm Based on RR^T Factorization

The transformation formulas for the diagonal elements of A:

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

If $a_{ii} = a_{jj}$, $a_{ij} = a_{jj}b_{ij}$ then we choose $\vartheta = 0$ and then a'_{ii} and a'_{jj} reduce to a_{ii} and a_{jj}, respectively.

The Algorithm Based on RR^T Factorization

This leads to the transformation matrix

$$\hat{Z} = \frac{1}{\sqrt{1-b_{ij}^2}} \begin{bmatrix} 1 & 0 \\ -b_{ij} & \sqrt{1-b_{ij}^2} \end{bmatrix} \begin{bmatrix} c_\vartheta & -s_\vartheta \\ s_\vartheta & c_\vartheta \end{bmatrix}$$

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

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

Algorithm RR^TJ

```
select the pivot pair (i, j)
if a_ij ≠ 0 or b_ij ≠ 0 then
```

$$
\begin{array}{l}
\texttt{select the pivot pair } (i,j) \\
\texttt{if } a_{ij} \neq 0 \texttt{ or } b_{ij} \neq 0 \texttt{ then} \\
\quad \beta = b_{ij}\,,\ \tau = \texttt{sqrt}((1+\beta)(1-\beta));\quad \alpha = a_{ij} - \beta a_{jj}\,; \\
\quad \texttt{if}\quad \alpha = 0 \quad \texttt{then}\quad t = 0; \\
\quad \texttt{else}\quad ct2 = (0.5\,(a_{ii} - a_{jj}) - \alpha\beta)/(\alpha\,\tau); \\
\qquad\qquad t = \texttt{sign}(ct2)/(\texttt{abs}(ct2) + \texttt{sqrt}(1 + ct2^2)); \\
\quad \texttt{endif} \\
\quad cs = 1/\texttt{sqrt}(1+t^2);\quad sn = t/\texttt{sqrt}(1+t^2); \\
\quad c1 = cs/\tau\,;\quad s1 = sn/\tau\,;\quad c2 = cs + sn\,\beta/\tau\,;\quad s2 = sn - cs\,\beta/\tau\,; \\
\quad \delta_j = t\alpha/\tau\,;\quad \delta_i = (\,t\alpha - (\beta/\tau)\cdot(2a_{ij} - (a_{ii}+a_{jj})\beta)\,)/\tau\,; \\
\quad a'_{ij} = (c1\,c2 - s1\,s2)\,a_{ij} + (c2\,s2\,a_{jj} - c1\,s1\,a_{ii});\quad a'_{ji} = a'_{ij}\,; \\
\quad b'_{ij} = (c1\,c2 - s1\,s2)\,\beta + (c2\,s2 - c1\,s1);\quad b'_{ji} = b'_{ij}\,; \\
\quad a'_{ii} = a_{ii} + \delta_i\,;\quad a'_{j} = a_{jj} - \delta_j\,; \\
\quad \texttt{for}\ \ k = 1,\ldots,n,\ k \neq i,j \quad \texttt{do} \\
\qquad a'_{ki} = c1\cdot a_{ki} + s2\cdot a_{kj};\quad b'_{ki} = c1\cdot b_{ki} + s2\cdot b_{kj};\quad a'_{ik} = a'_{ki};\quad b'_{ik} = b'_{ki} \\
\qquad a'_{kj} = c2\cdot a_{kj} - s1\cdot a_{ki};\quad b'_{kj} = c2\cdot b_{kj} - s1\cdot b_{ki};\quad a'_{jk} = a'_{kj};\quad b'_{jk} = b'_{kj}\,; \\
\quad \texttt{endfor} \\
\texttt{endif}
\end{array}
$$

Definition of a Hybrid and a General Method

Definition

Let $\mathcal{H}$ denote a collection of Jacobi methods for PGEP $Ax = \lambda Bx$ which satisfy the following two rules:

1. at step k, $\hat{A}^{(k)}$ is diagonalized and $\hat{B}^{(k)}$ is transformed to I_2,
2. at least one diagonal element of $\hat{F}_k$ is not smaller than $\sqrt{2}/2$.

An element of $\mathcal{H}$ is called a general PGEP Jacobi method.
A hybrid Jacobi method is any method from $\mathcal{H}$ that uses at each step either the HZ, $LL^T J$ or $RR^T J$ algorithm.

Definition of a Hybrid and a General Method

Definition

Let $\mathcal{H}$ denote a collection of Jacobi methods for PGEP $Ax = \lambda Bx$ which satisfy the following two rules:

1. at step k, $\hat{A}^{(k)}$ is diagonalized and $\hat{B}^{(k)}$ is transformed to I_2,
2. at least one diagonal element of $\hat{F}_k$ is not smaller than $\sqrt{2}/2$.

An element of $\mathcal{H}$ is called a general PGEP Jacobi method.
A hybrid Jacobi method is any method from $\mathcal{H}$ that uses at each step either the HZ, $LL^T J$ or $RR^T J$ algorithm.

In this definition the pivot strategy is not specified, hence any can be used.

Definition of a Hybrid and a General Method

Definition

Let $\mathcal{H}$ denote a collection of Jacobi methods for PGEP $Ax = \lambda Bx$ which satisfy the following two rules:

1. at step k, $\hat{A}^{(k)}$ is diagonalized and $\hat{B}^{(k)}$ is transformed to I_2,
2. at least one diagonal element of $\hat{F}_k$ is not smaller than $\sqrt{2}/2$.

An element of $\mathcal{H}$ is called a general PGEP Jacobi method.
A hybrid Jacobi method is any method from $\mathcal{H}$ that uses at each step either the HZ, LL^TJ or RR^TJ algorithm.

In this definition the pivot strategy is not specified, hence any can be used. If a method uses only the HZ (LL^TJ, RR^TJ) algorithm, it will be called the HZ (LL^TJ, RR^TJ) method.

Some Remarks

- It is easy to show that HZ, $LL^T J$ and $RR^T J$ methods belong to $\mathcal{H}$

Some Remarks

- It is easy to show that HZ, $LL^T J$ and $RR^T J$ methods belong to $\mathcal{H}$
- Algorithms based on LL^T and RR^T factorizations are called $LL^T J$ and $RR^T J$ algorithm, because LL^T and RR^T factorizations are followed by one step of the standard Jacobi method

Some Remarks

- It is easy to show that HZ, LL^TJ and RR^TJ methods belong to $\mathcal{H}$
- Algorithms based on LL^T and RR^T factorizations are called LL^TJ and RR^TJ algorithm, because LL^T and RR^T factorizations are followed by one step of the standard Jacobi method
- The general (PGEP) Jacobi method can use at each step any conceivable algorithm which satisfies the above two rules. For example, it can use the FL method combined with normalization of the elements of B

Some Remarks

- 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 $\hat{B} \succ O$ via the congruence transformation $\hat{B} \mapsto \hat{Z}^T \hat{B} \hat{Z}$.

Some Remarks

- 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 $\hat{B} \succ O$ via the congruence transformation $\hat{B} \mapsto \hat{Z}^T \hat{B} \hat{Z}$.

If we assume $b_{11} = \cdots = b_{nn})1$ and the same for $\hat{Z}^T \hat{B} \hat{Z}$, then this form of $\hat{Z}$ is just the Gose's theorem.

Thank you for your attention