On Element-wise and Block-wise Jacobi Methods for PGEP

Vjeran Hari

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

Meeting of the International Linear Algebra Society
July 24–28, 2017, Ames, Iowa, USA

OUTLINE

OUTLINE

- GEP and PGEP

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

OUTLINE

- GEP and PGEP
- Known results, open problems

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

OUTLINE

- GEP and PGEP
- Known results, open problems
- Derivation of the algorithms

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

OUTLINE

- GEP and PGEP
- Known results, open problems
- Derivation of the algorithms
- Convergence, global and asymptotic

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

OUTLINE

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

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

OUTLINE

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

- Since we consider theoretical aspects of the methods, we have restricted our attention to element-wise, two-sided Jacobi-type methods for PGEP. 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 $A = A^*, \quad B = B^*.$

GEP and PGEP

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

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

GEP 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 or shorter PGEP.

GEP 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 or shorter PGEP.

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

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

GEP 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 or shorter PGEP.

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

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

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

GEP and PGEP

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

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

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

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

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

How to Solve PGEP?

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

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

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

(ask Bart Vandereycken about it) and then derive a method which works with the initial pair (A, B).

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. thesis 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. thesis 1984)

The two methods are connected: the FL method can be viewed as the HZ method with “fast scaled” transformations.

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. thesis 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.

The FL Method (what is known, what is not known?)

- The method was derived for real symmetric matrices

The FL Method (what is known, what is not known?)

- The method was derived for real symmetric matrices
- In SIAM J. MAA 12 (1991), I. Slapničar and V. Hari proved the asymptotic quadratic convergence of the method in the case of simple eigenvalues. They also proved that the method is well defined for definite matrix pairs ($\alpha A + \beta B \succ O$ for some real α, β)

The FL Method (what is known, what is not known?)

- The method was derived for real symmetric matrices
- In SIAM J. MAA 12 (1991), I. Slapničar and V. Hari proved the asymptotic quadratic convergence of the method in the case of simple eigenvalues. They also proved that the method is well defined for definite matrix pairs ($\alpha A + \beta B \succ O$ for some real α, β)
- In Numer. Algor. 68 (2015), J. Matejaš proved accuracy bounds for one step of the method

The FL Method (what is known, what is not known?)

- The method was derived for real symmetric matrices
- In SIAM J. MAA 12 (1991), I. Slapničar and V. Hari proved the asymptotic quadratic convergence of the method in the case of simple eigenvalues. They also proved that the method is well defined for definite matrix pairs ($\alpha A + \beta B \succ O$ for some real α, β)
- In Numer. Algor. 68 (2015), J. Matejaš proved accuracy bounds for one step of the method
- What is missing?
  1. Some global convergence proof
  2. Some proof of high relative accuracy of the method on well-behaved matrix pairs
  3. Complex algorithm for definite pairs of Hermitian matrices

The HZ Method (what is known, what is not known?)

- Originally, the method was derived in 1984 for the pair (A, B) of complex Hermitian matrices such that $B \succ O$

The HZ Method (what is known, what is not known?)

- Originally, the method was derived in 1984 for the pair (A, B) of complex Hermitian matrices such that $B \succ O$
- In that thesis the asymptotic quadratic convergence has been proved for the case of simple eigenvalues

The HZ Method (what is known, what is not known?)

- Originally, the method was derived in 1984 for the pair (A, B) of complex Hermitian matrices such that $B \succ O$
- In that thesis the asymptotic quadratic convergence has been proved for the case of simple eigenvalues
- In the thesis, the real method was derived directly from the complex one. It was also shown that the real FL method can be considered as the real HZ method with fast-scaled transformations

The HZ Method (what is known, what is not known?)

- Originally, the method was derived in 1984 for the pair (A, B) of complex Hermitian matrices such that $B \succ O$
- In that thesis the asymptotic quadratic convergence has been proved for the case of simple eigenvalues
- In the thesis, the real method was derived directly from the complex one. It was also shown that the real FL method can be considered as the real HZ method with fast-scaled transformations
- The method was named in the paper of V. Novaković, S. Singer, S. Singer (Parallel Comput. 2015) where its block version was numerically tested on parallel machines

The HZ Method (what is known, what is not known?)

- Originally, the method was derived in 1984 for the pair (A, B) of complex Hermitian matrices such that $B \succ O$
- In that thesis the asymptotic quadratic convergence has been proved for the case of simple eigenvalues
- In the thesis, the real method was derived directly from the complex one. It was also shown that the real FL method can be considered as the real HZ method with fast-scaled transformations
- The method was named in the paper of V. Novaković, S. Singer, S. Singer (Parallel Comput. 2015) where its block version was numerically tested on parallel machines
- What is missing?
  1. Some global convergence proof
  2. Some proof of high relative accuracy of the method
  3. Connection of the complex HZ method to the complex FL method

New Results

- Complex Falk-Langemayer algorithm derived (work in progress)

New Results

- Complex Falk-Langemayer algorithm derived (work in progress)
- New real algorithms for PGEP derived (LL^TJ, RR^TJ and CJ algorithm).

New Results

- Complex Falk-Langemayer algorithm derived (work in progress)
- New real algorithms for PGEP derived (LL^TJ, RR^TJ and CJ algorithm).
- The global convergence has been proved for all those PGEP methods, including some much more general PGEP methods which are called hybrid and general PGEP Jacobi method. The new results hold for the large class of generalized serial pivot strategies

New Results

- Complex Falk-Langemayer algorithm derived (work in progress)
- New real algorithms for PGEP derived (LL^TJ, RR^TJ and CJ algorithm).
- The global convergence has been proved for all those PGEP methods, including some much more general PGEP methods which are called hybrid and general PGEP Jacobi method. The new results hold for the large class of generalized serial pivot strategies
- Matejaš and Hari have made a detailed rounding error analysis of one step of the HZ method. This analysis sheds light to accuracy issues. Now it is understood why the method shows high relative accuracy property on well-behaved matrix pairs (work in progress)

New Results

- Complex Falk-Langemayer algorithm derived (work in progress)
- New real algorithms for PGEP derived (LL^TJ, RR^TJ and CJ algorithm).
- The global convergence has been proved for all those PGEP methods, including some much more general PGEP methods which are called hybrid and general PGEP Jacobi method. The new results hold for the large class of generalized serial pivot strategies
- Matejaš and Hari have made a detailed rounding error analysis of one step of the HZ method. This analysis sheds light to accuracy issues. Now it is understood why the method shows high relative accuracy property on well-behaved matrix pairs (work in progress)
- V. Hari and E. Begović Kovač have developed tools (block-Jacobi annihilators and operators) for proving the global convergence of real and complex block and element-wise Jacobi methods for PGEP and similar problems under the class of generalized serial strategies (ETNA 46, 2017, and one work in progress)

Some Open Problems

- Asymptotic quadratic convergence of any of those methods in the case of multiple or close eigenvalues of the pair (A, B).

Some Open Problems

- Asymptotic quadratic convergence of any of those methods in the case of multiple or close eigenvalues of the pair (A, B).

  We understand behavior of the algorithms when iteration matrices are nearly diagonal: huge cancelation in the computation of certain angles, asymptotic convergence slowdown, termination criteria problems, ...

Some Open Problems

- Asymptotic quadratic convergence of any of those methods in the case of multiple or close eigenvalues of the pair (A, B).

  We understand behavior of the algorithms when iteration matrices are nearly diagonal: huge cancelation in the computation of certain angles, asymptotic convergence slowdown, termination criteria problems, ...
  However, we are confident that a robust algorithm with overall good properties can be found

Some Open Problems

- Asymptotic quadratic convergence of any of those methods in the case of multiple or close eigenvalues of the pair (A, B).

  We understand behavior of the algorithms when iteration matrices are nearly diagonal: huge cancelation in the computation of certain angles, asymptotic convergence slowdown, termination criteria problems, ...
  However, we are confident that a robust algorithm with overall good properties can be found

- To find the best possible candidate for the kernel algorithm for one-sided block Jacobi methods for GSVD, in the real and in the complex case

Some Open Problems

- Asymptotic quadratic convergence of any of those methods in the case of multiple or close eigenvalues of the pair (A, B).

  We understand behavior of the algorithms when iteration matrices are nearly diagonal: huge cancelation in the computation of certain angles, asymptotic convergence slowdown, termination criteria problems, ...
  However, we are confident that a robust algorithm with overall good properties can be found
- To find the best possible candidate for the kernel algorithm for one-sided block Jacobi methods for GSVD, in the real and in the complex case
- To make sound global and asymptotic convergence proofs of the block Jacobi methods for PGEP

One Block Jacobi Method for PGEP

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

One Block Jacobi Method 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 over the FL approach.

One Block Jacobi Method 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 over the FL approach.

> *When implemented as one-sided block algorithm for the GSVD, the HZ method 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.*

One Block Jacobi Method 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 over the FL approach.

> *When implemented as one-sided block algorithm for the GSVD, the HZ method 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.*

That paper deals with real block HZ method.

One Block Jacobi Method 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 over the FL approach.

> *When implemented as one-sided block algorithm for the GSVD, the HZ method 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.*

That paper deals with real block HZ method.

The problem with block methods is that they need best possible kernel algorithms: globally convergent, highly accurate and numerically fast.

Returning to Element-wise PGEP Jacobi Methods

So, the primary theoretical and practical task in this area seems to be finding an excellent kernel algorithm for the block methods.

So, the primary theoretical and practical task in this area seems to be finding an excellent kernel algorithm for the block methods.

In any block Jacobi method the pivot submatrices are most of the time nearly diagonal.

Returning to Element-wise PGEP Jacobi Methods

So, the primary theoretical and practical task in this area seems to be finding an excellent kernel algorithm for the block methods.

In any block Jacobi method the pivot submatrices are most of the time nearly diagonal. Hence the best choice for the kernel algorithm seems to be an element-wise PGEP Jacobi method. Such methods are very fast and accurate on pairs of nearly diagonal matrices.

Returning to Element-wise PGEP Jacobi Methods

So, the primary theoretical and practical task in this area seems to be finding an excellent kernel algorithm for the block methods.

In any block Jacobi method the pivot submatrices are most of the time nearly diagonal. Hence the best choice for the kernel algorithm seems to be an element-wise PGEP Jacobi method. Such methods are very fast and accurate on pairs of nearly diagonal matrices.

The next slides are devoted to new element-wise PGEP Jacobi algorithms. We start with the derivation of the real element-wise HZ method, since it has not yet been properly disclosed. Then we shall derive the new real algorithms.

Returning to Element-wise PGEP Jacobi Methods

So, the primary theoretical and practical task in this area seems to be finding an excellent kernel algorithm for the block methods.

In any block Jacobi method the pivot submatrices are most of the time nearly diagonal. Hence the best choice for the kernel algorithm seems to be an element-wise PGEP Jacobi method. Such methods are very fast and accurate on pairs of nearly diagonal matrices.

The next slides are devoted to new element-wise PGEP Jacobi algorithms. We start with the derivation of the real element-wise HZ method, since it has not yet been properly disclosed. Then we shall derive the new real algorithms.

For obvious reasons we shall try to escape derivation of complex algorithms!!!

Derivation of the Real HZ Method

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

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

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 $B^{(0)}$ will be 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 Real 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 $B^{(0)}$ will be 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 Real 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 $B^{(0)}$ will be 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 Real 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 $B^{(0)}$ will be 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: $\qquad A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

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

$\hat{A}$, $\hat{B}$, $\hat{Z}$ 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: $\quad A^{(k)} \mapsto A, \quad A^{(k+1)} \mapsto A', \quad Z_k \mapsto Z,$

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

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

Then $A' = Z^T A Z$, $B' = Z^T B Z$ 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:

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

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

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

$\hat{Z}$ 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 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

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

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

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

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

Essential Part of the Algorithm

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

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

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

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

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

There are more formulas!

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

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

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

There are more formulas!

Let

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

then

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

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

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

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 ≠ 0 or b_ij ≠ 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

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

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

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

Digression: Complex Matrices

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

Digression: Complex Matrices

If $A = A^*$, $B = B^* \succ O$ are complex, with $\mathrm{diag}(B) = I_n$, 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} and to maintain ones on the diagonal of B'.

Digression: Complex Matrices

If $A = A^*$, $B = B^* \succ O$ are complex, with $\mathrm{diag}(B) = I_n$, 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} and to maintain ones on the diagonal of B'.

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 $A = A^*$, $B = B^* \succ O$ are complex, with $\mathrm{diag}(B) = I_n$, 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} and to maintain ones on the diagonal of B'.

It is proved that that pivot submatrix of Z has form

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

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

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

$$
\begin{array}{ccc}
 & \hat{B} \to \text{diag} & \hat{B} \to I_2 \\
 & \uparrow & \uparrow
\end{array}
$$

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

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

$$
\begin{array}{cc}
\downarrow & \downarrow \\
\hat{A} \to \text{diag} & \text{diag}(\hat{Z}) \succ O
\end{array}
$$

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

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

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

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 $\quad a = b_{ij}, \;\; c = \sqrt{1 - b_{ij}^2}, \quad$ hence

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

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

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

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

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

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

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

If we write

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

and

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 $\hat{F}$ 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

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

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

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

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

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

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

Algorithm LL^TJ

select the pivot pair (i, j)

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

$\beta = b_{ij}\,,\ \tau = \mathtt{sqrt}((1+\beta)(1-\beta));\quad \alpha = a_{ij} - \beta a_{ii}\,;$

if $\alpha = 0$ then $t = 0;$

else $ct2 = (0.5\,(a_{ii} - a_{jj}) + \alpha\beta)/(\alpha\,\tau);$

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

endif

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

$c1 = cs - sn\,\beta/\tau\,;\quad s1 = sn + cs\,\beta/\tau\,;\quad c2 = cs/\tau\,;\quad s2 = sn/\tau\,;$

$\delta_i = t\alpha/\tau\,;\quad \delta_j = (\,t\alpha + (\beta/\tau)\cdot(2a_{ij} - (a_{ii}+a_{jj})\beta)\,)/\tau\,;$

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

$b'_{ij} = (c1\,c2 - s1\,s2)\,\beta + (c2\,s2 - c1\,s1);\quad b'_{ji} = b'_{ij}\,;$

$a'_{ii} = a_{ii} + \delta_i\,;\quad a'_j = a_{jj} - \delta_j\,;$

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

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

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

The final $\hat{F}$ 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:

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

The Algorithm Based on RR^T Factorization

The transformation formulas for the diagonal elements of A read

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

The Algorithm Based on RR^T Factorization

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 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
```

$$\beta = b_{ij}, \ \tau = \mathtt{sqrt}((1+\beta)(1-\beta)); \quad \alpha = a_{ij} - \beta a_{jj};$$

$\mathtt{if}\ \ \alpha = 0 \quad \mathtt{then}\ \ t = 0;$

$\mathtt{else}\ \ ct2 = (0.5\,(a_{ii} - a_{jj}) - \alpha\beta)/(\alpha\,\tau);$

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

$\mathtt{endif}$

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

$$c1 = cs/\tau; \quad s1 = sn/\tau; \quad c2 = cs + sn\,\beta/\tau; \quad s2 = sn - cs\,\beta/\tau;$$

$$\delta_j = t\alpha/\tau; \quad \delta_i = (\,t\alpha - (\beta/\tau)\cdot(2a_{ij} - (a_{ii}+a_{jj})\beta)\,)/\tau;$$

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

$$b'_{ij} = (c1\,c2 - s1\,s2)\,\beta + (c2\,s2 - c1\,s1); \quad b'_{ji} = b'_{ij};$$

$$a'_{ii} = a_{ii} + \delta_i; \quad a'_{j} = a_{jj} - \delta_j;$$

$\mathtt{for}\ \ k = 1, \ldots, n,\ k \neq i, j \quad \mathtt{do}$

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

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

$\mathtt{endfor}$

$\mathtt{endif}$

Definition of a Hybrid and a General Method

Definition

Let $\mathcal{H}$ denote 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 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 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, $LL^T J$ 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
- 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

$$\hat{Z} = \frac{1}{\sqrt{1 - b_{ij}^2}} \left[\begin{array}{cc} \cos\phi & -\sin\phi \\ \cos\psi & \sin\psi \end{array} \right].$$

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

Global Convergence of the General PGEP Jacobi Method

We have used the following measure in the convergence analysis:

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

Global Convergence of the General PGEP Jacobi Method

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 general PGEP method is globally convergent 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 $B \succ O$.

Global Convergence of the General PGEP Jacobi Method

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 general PGEP method is globally convergent 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 $k \to \infty$.

Global Convergence of the General PGEP Jacobi Method

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 general PGEP method is globally convergent 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 class of

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 general PGEP method is globally convergent 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 $k \to \infty$.

We have proved the global convergence for the class of

- serial pivot strategies

Global Convergence of the General PGEP Jacobi Method

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 general PGEP method is globally convergent 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 class of

- serial pivot strategies
- generalized serial strategies which includes all weakly wavefront strategies and many others (Hari, Begović Kovač, ETNA 46 (2017) 107-147)

Asymptotic Convergence for the HZ Method

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 for the HZ Method

Let (A, B) have simple eigenvalues:

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

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

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:

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

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

In the case of multiple eigenvalues, the method is not quadratically convergent, but can be modified to be such.

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, LL^TJ and RR^TJ methods

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, $LL^T J$ and $RR^T J$ methods
- For that we need a detailed error analysis (J. Matejaš, V. Hari)

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, $LL^T J$ and $RR^T J$ methods
- For that we need a detailed error analysis (J. Matejaš, V. Hari)
- We present here only the results of numerical tests

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, $LL^T J$ and $RR^T J$ methods
- For that we need a detailed error analysis (J. Matejaš, V. Hari)
- We present here only the results of numerical tests
  First theoretical background for the tests, and then the test results

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, $LL^T J$ and $RR^T J$ methods
- For that we need a detailed error analysis (J. Matejaš, V. Hari)
- We present here only the results of numerical tests
  First theoretical background for the tests, and then the test results
- High relative accuracy of the methods can be obtained only for well-behaved initial pairs (A, B)

Stability and High Relative Accuracy

- We inspect high relative accuracy of HZ, LL^TJ and RR^TJ methods
- For that we need a detailed error analysis (J. Matejaš, V. Hari)
- We present here only the results of numerical tests
  First theoretical background for the tests, and then the test results
- High relative accuracy of the methods can be obtained only for well-behaved initial pairs (A, B)
- An example of such pairs are the pairs for which the condition numbers $\kappa_2(\Delta_A A \Delta_A)$ and $\kappa_2(\Delta_B B \Delta_B)$ are small for some diagonal matrices Δ_A and Δ_B.

Theorem

Let $A = A^T \succ O$, $B = B^T \succ O$ *and* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, $\lambda_i \in \sigma(A, B)$.
Let $A_S = D_A^{-1/2} A D_A^{-1/2}$, $B_S = D_B^{-1/2} B D_B^{-1/2}$, $D_A = diag(A)$, $D_B = diag(B)$

Theorem

Let $A = A^T \succ O$, $B = B^T \succ O$ *and* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, $\lambda_i \in \sigma(A, B)$.
Let $A_S = D_A^{-1/2} A D_A^{-1/2}$, $B_S = D_B^{-1/2} B D_B^{-1/2}$, $D_A = diag(A)$, $D_B = diag(B)$

Let δA, δB *be symmetric perturbations and* $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n$ *the eigenvalues of* $(A + \delta A, B + \delta B)$.

Theorem

Let $A = A^T \succ O$, $B = B^T \succ O$ *and* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, $\lambda_i \in \sigma(A, B)$.
Let $A_S = D_A^{-1/2} A D_A^{-1/2}$, $B_S = D_B^{-1/2} B D_B^{-1/2}$, $D_A = diag(A)$, $D_B = diag(B)$

Let δA, δB *be symmetric perturbations and* $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n$ *the eigenvalues of* $(A + \delta A, B + \delta B)$.

Let

$$\varepsilon_{A_S} = \|(\delta A)_S\|_2 / \|A_S\|_2, \quad \varepsilon_{B_S} = \|(\delta B)_S\|_2 / \|B_S\|_2$$

where $(\delta A)_S = D_A^{-1/2} \delta A D_A^{-1/2}$, $(\delta B)_S = D_B^{-1/2} \delta B D_B^{-1/2}$.

Theorem

Let $A = A^T \succ O$, $B = B^T \succ O$ *and* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, $\lambda_i \in \sigma(A,B)$.
Let $A_S = D_A^{-1/2} A D_A^{-1/2}$, $B_S = D_B^{-1/2} B D_B^{-1/2}$, $D_A = diag(A)$, $D_B = diag(B)$

Let δA, δB *be symmetric perturbations and* $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n$ *the eigenvalues of* $(A + \delta A, B + \delta B)$.

Let

$$\varepsilon_{A_S} = \|(\delta A)_S\|_2 / \|A_S\|_2, \quad \varepsilon_{B_S} = \|(\delta B)_S\|_2 / \|B_S\|_2$$

where $(\delta A)_S = D_A^{-1/2} \delta A D_A^{-1/2}$, $(\delta B)_S = D_B^{-1/2} \delta B D_B^{-1/2}$.

If

$$\varepsilon_{A_S} \kappa_2(A_S) < 1 \quad \text{and} \quad \varepsilon_{B_S} \kappa_2(B_S) < 1,$$

then

$$\max_{1 \leq i \leq n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \leq \frac{\varepsilon_{A_S} \kappa_2(A_S) + \varepsilon_{B_S} \kappa_2(B_S)}{1 - \varepsilon_{B_S} \kappa_2(B_S)}.$$

Theoretical Background

- The initial normalization $B \mapsto B_S = B^{(0)}$, simplifies the algorithm. Moreover, it has a stabilizing effect on the iterative process, because it almost optimally reduces the condition of B and all $B^{(k)}$, $k \geq 1$.
  Van der Sluis, A. Numer. Math. 14 (1969)

Theoretical Background

- The initial normalization $B \mapsto B_S = B^{(0)}$, simplifies the algorithm. Moreover, it has a stabilizing effect on the iterative process, because it almost optimally reduces the condition of B and all $B^{(k)}$, $k \geq 1$.
  Van der Sluis, A. Numer. Math. 14 (1969)

- For those well-behaved pairs we have to find out what methods generate at every step only tiny relative errors $\varepsilon_{A_S^{(k)}}$, $\varepsilon_{B_S^{(k)}}$ and in the same time matrices with small or modest $\kappa_2(A_S^{(k)})$ and $\kappa_2(B^{(k)})$.

Theoretical Background

- The initial normalization $B \mapsto B_S = B^{(0)}$, simplifies the algorithm. Moreover, it has a stabilizing effect on the iterative process, because it almost optimally reduces the condition of B and all $B^{(k)}$, $k \geq 1$.
  Van der Sluis, A. Numer. Math. 14 (1969)

- For those well-behaved pairs we have to find out what methods generate at every step only tiny relative errors $\varepsilon_{A_S^{(k)}}$, $\varepsilon_{B_S^{(k)}}$ and in the same time matrices with small or modest $\kappa_2(A_S^{(k)})$ and $\kappa_2(B^{(k)})$.

Nonetheless, this is a demanding task, so we shall go for a shortcut.

How to detect high relative accuracy of a method?

For all considered methods the starting matrix $B^{(0)}$ is just B_S. Therefore

$$\max_{1\leq i\leq n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \leq \frac{\varepsilon_{A_S}\kappa_2(A_S) + \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}{1 - \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}, \text{ and it implies}$$

How to detect high relative accuracy of a method?

For all considered methods the starting matrix $B^{(0)}$ is just B_S. Therefore

$$\max_{1\leq i\leq n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \leq \frac{\varepsilon_{A_S}\kappa_2(A_S) + \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}{1 - \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}, \text{ and it implies}$$

$$\varrho_{(A,B)} \equiv \frac{\max_{1\leq i\leq n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i}}{\sqrt{\kappa_2^2(A_S^{(0)}) + \kappa_2^2(B^{(0)})}} \leq \frac{\sqrt{\varepsilon_{A_S}^2 + \varepsilon_{B^{(0)}}^2}}{1 - \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})} \approx \max\{|\varepsilon_{A_S}|, |\varepsilon_{B^{(0)}}|\},$$

How to detect high relative accuracy of a method?

For all considered methods the starting matrix $B^{(0)}$ is just B_S. Therefore

$$\max_{1\le i\le n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \le \frac{\varepsilon_{A_S}\kappa_2(A_S) + \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}{1 - \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}, \text{ and it implies}$$

$$\varrho_{(A,B)} \equiv \frac{\max_{1\le i\le n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i}}{\sqrt{\kappa_2^2(A_S^{(0)}) + \kappa_2^2(B^{(0)})}} \le \frac{\sqrt{\varepsilon_{A_S}^2 + \varepsilon_{B^{(0)}}^2}}{1 - \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})} \approx \max\{|\varepsilon_{A_S}|, |\varepsilon_{B^{(0)}}|\},$$

We can check numerically whether the inequality

$$\varrho_{(A,B)} \le f(n)\mathbf{u}, \tag{3}$$

holds for a larger sample Υ of pairs (A, B).

How to detect high relative accuracy of a method?

For all considered methods the starting matrix $B^{(0)}$ is just B_S. Therefore

$$\max_{1\le i\le n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \le \frac{\varepsilon_{A_S}\kappa_2(A_S) + \varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}{1-\varepsilon_{B^{(0)}}\kappa_2(B^{(0)})}, \text{ and it implies}$$

$$\varrho_{(A,B)} \equiv \frac{\max_{1\le i\le n}\frac{|\tilde{\lambda}_i-\lambda_i|}{\lambda_i}}{\sqrt{\kappa_2^2(A_S^{(0)}) + \kappa_2^2(B^{(0)})}} \le \frac{\sqrt{\varepsilon_{A_S}^2 + \varepsilon_{B^{(0)}}^2}}{1-\varepsilon_{B^{(0)}}\kappa_2(B^{(0)})} \approx \max\{|\varepsilon_{A_S}|, |\varepsilon_{B^{(0)}}|\},$$

We can check numerically whether the inequality

$$\varrho_{(A,B)} \le f(n)\mathbf{u}, \tag{3}$$

holds for a larger sample Υ of pairs (A,B). Here

- $\tilde{\lambda}_i$, $1 \le i \le n$ are computed eigenvalues of $(A^{(0)}, B^{(0)})$
- $f(n)$ is a slowly growing function of n and $\mathbf{u}$ is the unit round off
- The relation (3) should hold irrespectively of how large $\kappa_2(A^{(0)})$ is.

Therefore, we are interested in how $\varrho_{(A,B)}$ behaves with respect to $\chi_{(A,B)}$,

$$\chi_{(A,B)} \equiv \kappa_2(A^{(0)}, B^{(0)}) = \sqrt{\kappa_2^2(A^{(0)}) + \kappa_2^2(B^{(0)})}.$$

Therefore, we are interested in how $\varrho_{(A,B)}$ behaves with respect to $\chi_{(A,B)}$,

$$\chi_{(A,B)} \equiv \kappa_2(A^{(0)}, B^{(0)}) = \sqrt{\kappa_2^2(A^{(0)}) + \kappa_2^2(B^{(0)})}.$$

- For the given sample of pairs Υ, and for each method, we shall make its graph of relative errors:

$$\mathcal{E} = \{(\chi_{(A,B)}\ ,\ \varrho_{(A,B)}) \,:\, (A, B) \in \Upsilon\}.$$

Therefore, we are interested in how $\varrho_{(A,B)}$ behaves with respect to $\chi_{(A,B)}$,

$$\chi_{(A,B)} \equiv \kappa_2(A^{(0)}, B^{(0)}) = \sqrt{\kappa_2^2(A^{(0)}) + \kappa_2^2(B^{(0)})}.$$

- For the given sample of pairs Υ, and for each method, we shall make its graph of relative errors:

$$\mathcal{E} = \{(\chi_{(A,B)}\ ,\ \varrho_{(A,B)})\ :\ (A,B) \in \Upsilon\}.$$

- Then we shall depict that graph $\mathcal{E}$ using the M-function `scatter(x,y,3)`

Therefore, we are interested in how $\varrho_{(A,B)}$ behaves with respect to $\chi_{(A,B)}$,

$$\chi_{(A,B)} \equiv \kappa_2(A^{(0)}, B^{(0)}) = \sqrt{\kappa_2^2(A^{(0)}) + \kappa_2^2(B^{(0)})}.$$

- For the given sample of pairs Υ, and for each method, we shall make its graph of relative errors:

$$\mathcal{E} = \{(\chi_{(A,B)}\ ,\ \varrho_{(A,B)})\ :\ (A,B) \in \Upsilon\}.$$

- Then we shall depict that graph $\mathcal{E}$ using the M-function `scatter(x,y,3)`

- The method will be indicated high relative accurate if the ordinates of the points on the graph are of order $\mathcal{O}(\mathbf{u})$ where $\mathbf{u} \approx 2.2 \cdot 10^{-16}$.

How to generate matrix pairs?

The starting pair $(A^{(0)}, B^{(0)})$ is generated by

- 4 the diagonal matrices : Δ_A, Δ_B, Σ, Δ and
- 2 orthogonal matrices U, V of order n.

It is done in two steps:

1: $F = U\Sigma V^T, \quad A = F^T \Delta_A F, \quad B = F^T \Delta_B F,$

2: $B^{(0)} = B_S = D_B^{-1/2} B D_B^{-1/2}, \quad A^{(0)} = \Delta A_S \Delta,\ A_S = D_A^{-1/2} A D_A^{-1/2},$

where D_A and D_B are the diagonal parts of A and B. Then $\kappa_2(A_S^{(0)})$ and $\kappa_2(B^{(0)})$ can be controlled by the diagonal elements of Δ_A, Δ_B, Σ, since

$$\kappa_2(A_S^{(0)}) \leq n\kappa_2^2(\Sigma)\kappa_2(\Delta_A) \quad \text{and} \quad \kappa_2(B^{(0)}) \leq n\kappa_2^2(\Sigma)\kappa_2(\Delta_B),$$

although most often $\kappa_2(A_S^{(0)})$ and $\kappa_2(B^{(0)})$ are much smaller than these bounds.

How to generate matrix pairs?

To simplify the construction we set $\Delta_B = I_n$.

If the method is high relative accurate, then $\varrho_{(A,B)}$ from the relation (3) should not depend on $\kappa_2(\Delta)$.

Note that

$$\kappa_2(A^{(0)}) \leq \kappa_2(A_S^{(0)})\kappa_2^2(\Delta).$$

If we set $\Delta = I_n$ i $(A^{(0)}, B^{(0)}) = (D_B^{-1/2} A D_B^{-1/2}, B_S)$, then we know in advance the eigenvalues of $(A^{(0)}, B^{(0)})$ These are the quotients

$$(\Delta_A)_{jj}/(\Delta_B)_{jj}, \qquad 1 \leq j \leq n.$$

This way can be used when considering behavior of the methods on pairs with multiple eigenvalues.

More Details

- Diagonal matrices are constructed by help of the M-function `diag(d)`
- `d` is a vector, and vectors are constructed by the M-function `logspace(x1,x2,n)`. We use it for the diagonal matrices Σ and Δ_A.
- For the construction of Δ we use our m-function

  `scalvec(k1,k2,k3,n,k)`

  which generates vector of length n, $d = [10^{\mathtt{k1}}, \ldots, 10^{\mathtt{k2}}, \ldots, 10^{\mathtt{k3}}]$ where `k` determines the position of $10^{\mathtt{k2}}$ within the components of d.
- To compute Δ, the function `scalvec` is used within triple loop controlled by the indices `k1`, `k2` and `k3`
- Orthogonal matrices U and V are computed by the command

  `[Q,~]=qr(rand(n))`
- We have generated the sample Υ of 18900 pairs of matrices of order 10. As "exact eigenvalues" we have used the eigenvalues computed by the M-function `eig(A,B)` in variable precision arithmetic (VPA) using 80 decimal digits.

The Methods and Their Variants

For each of the methods, HZ, LL^TJ, RR^TJ, we have made two additional variants. Let us explain it for the case of the HZ method.

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \bullet & 1 & 2 & 3 & 4 & 5 \\ & \bullet & 6 & 7 & 8 & 9 \\ & & \bullet & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \bullet & 6 & 7 & 8 & 9 \\ & & \bullet & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, LL^TJ, RR^TJ, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \bullet & 6 & 7 & 8 & 9 \\ & & \bullet & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \bullet & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & \color{red}{6} & \color{red}{7} & \color{red}{8} & \color{red}{9} \\ & & \bullet & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & \color{red}{10} & \color{red}{11} & \color{red}{12} \\ & & & \bullet & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, LL^TJ, RR^TJ, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \circ & 13 & 14 \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \circ & \color{red}{13} & \color{red}{14} \\ & & & & \bullet & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, LL^TJ, RR^TJ, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \circ & 13 & 14 \\ & & & & \circ & 15 \\ & & & & & \bullet \end{bmatrix}$$

The Methods and Their Variants

For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \circ & 13 & 14 \\ & & & & \circ & 15 \\ & & & & & \bullet \end{bmatrix}$$

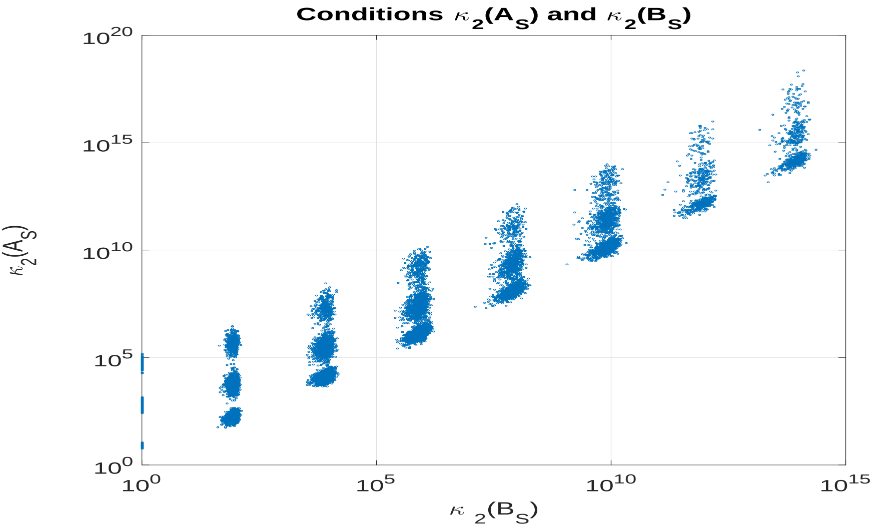
For each of the methods, HZ, $LL^T J$, $RR^T J$, we have made two additional variants. Let us explain it for the case of the HZ method.

We have used deRijk serial strategy. This is modified row-cyclic strategy:

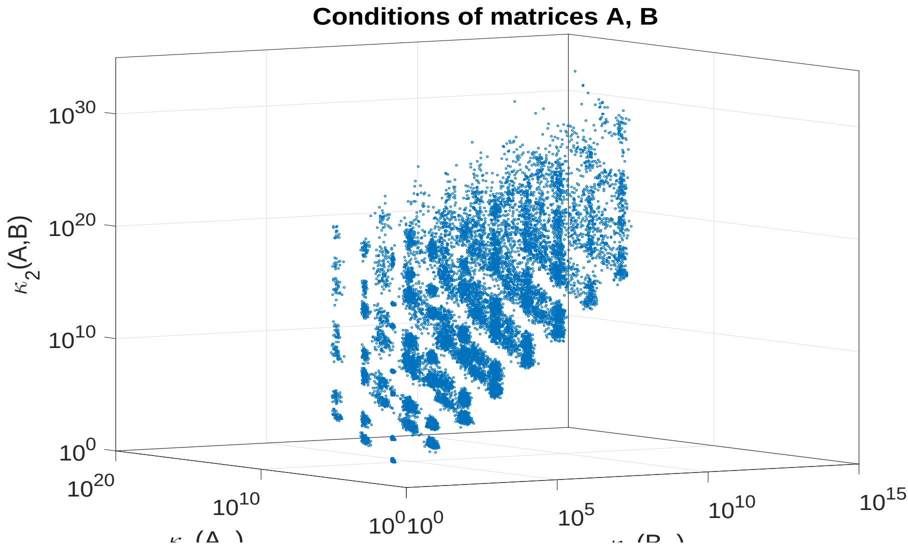
$$\begin{bmatrix} \circ & 1 & 2 & 3 & 4 & 5 \\ & \circ & 6 & 7 & 8 & 9 \\ & & \circ & 10 & 11 & 12 \\ & & & \circ & 13 & 14 \\ & & & & \circ & 15 \\ & & & & & \circ \end{bmatrix}$$

We call it descending HZ method or shorter HZD method, because the diagonal elements tend to end in descending order. In the similar way is defined ascending HZ method or HZA method.

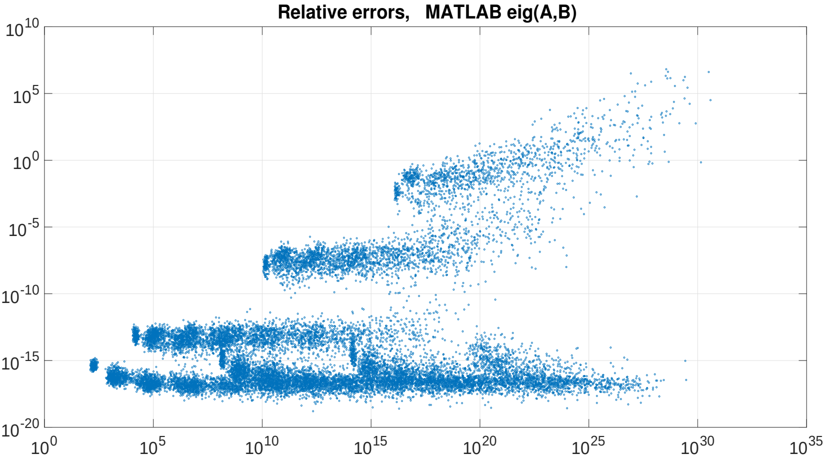
Matrix conditions



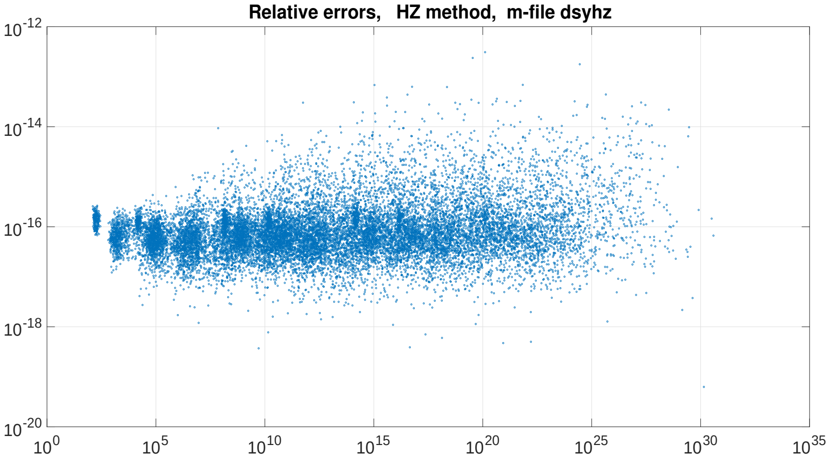
Matrix conditions



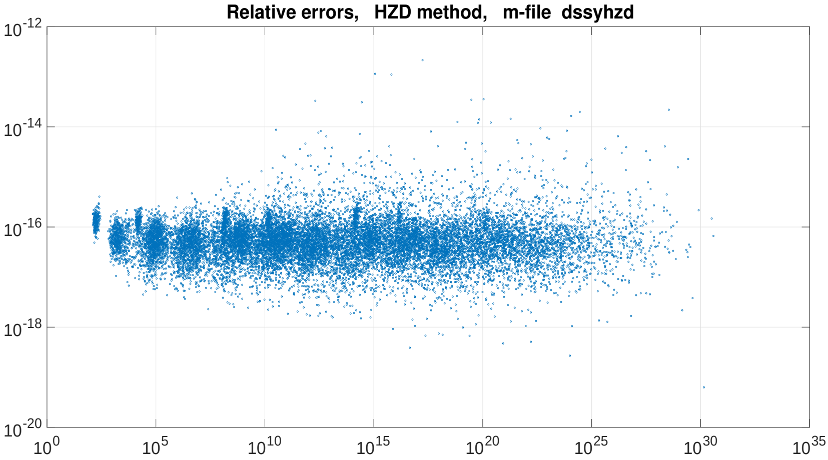
Relative errors: MATLAB eig function



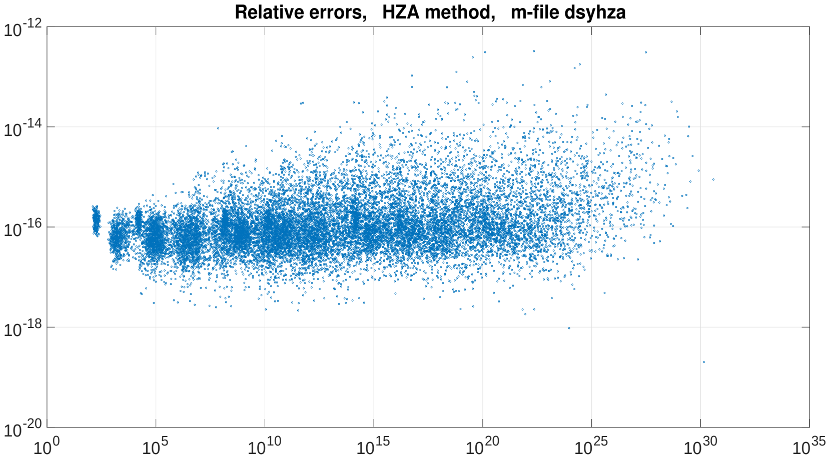
Relative errors: HZ method



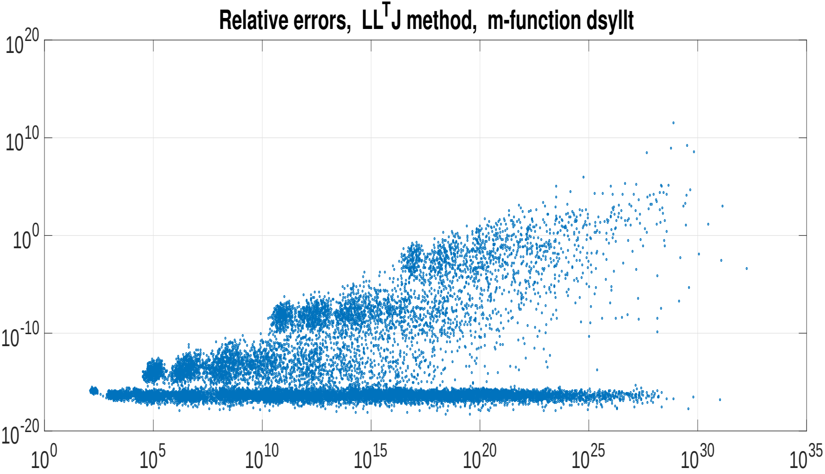
Relative errors: HZD method



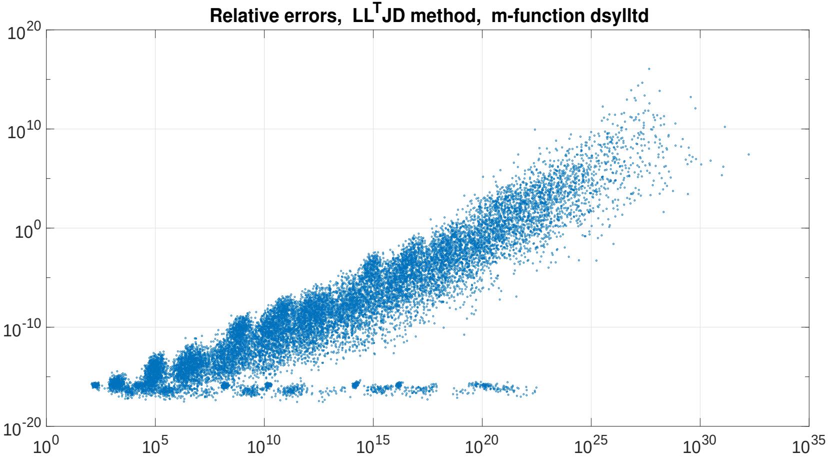
Relative errors: HZA method



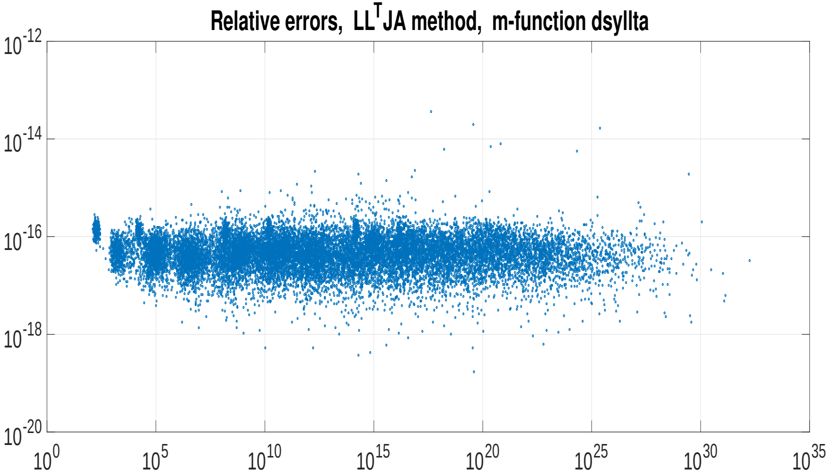
Relative errors: $LL^T J$ method



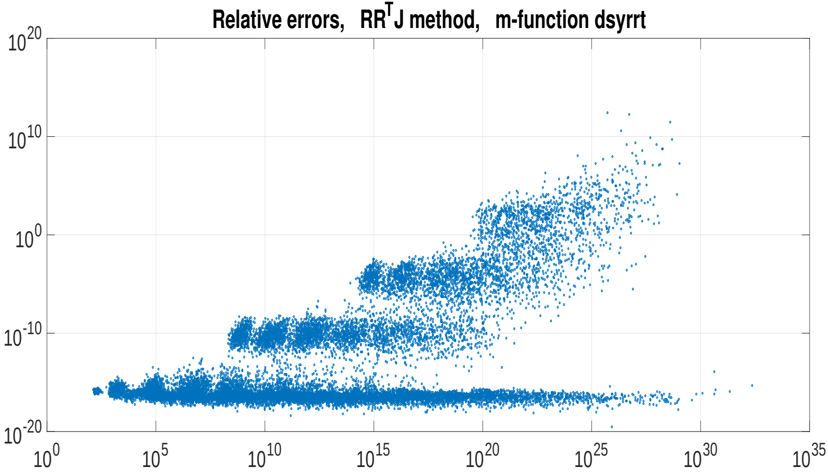
Relative errors: Descending LL^TJ method



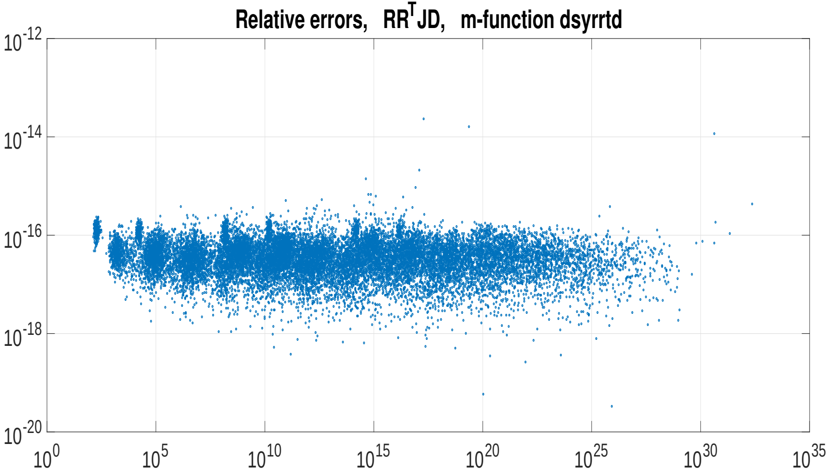
Relative errors: Ascending LL^TJ method



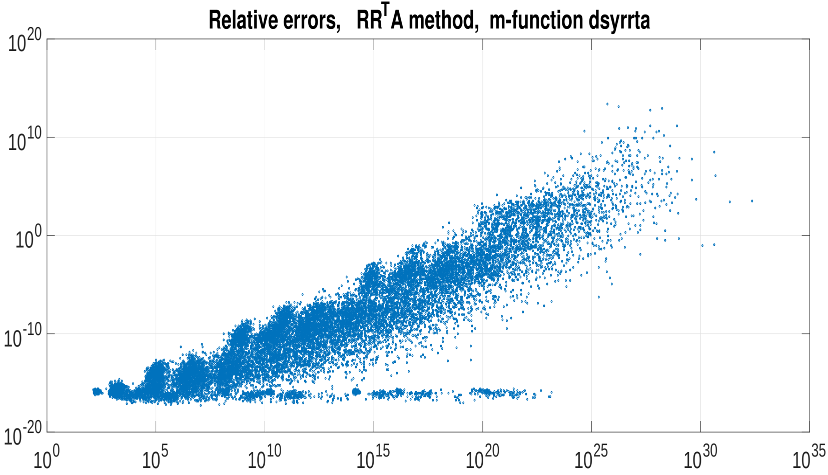
Relative errors: RR^TJ method



Relative errors: Descending RR^TJ method



Relative errors: Ascending RR^TJ method



How to define an accurate hybrid method?

We see that just one variant of LL^TJ method (LL^TJA) and just one variant of RR^TJ method (RR^TJD) is indicated as relatively accurate.

How to define an accurate hybrid method?

We see that just one variant of LL^TJ method (LL^TJA) and just one variant of RR^TJ method (RR^TJD) is indicated as relatively accurate.

This shows how to define a highly accurate hybrid method, call it *Cholesky-Jacobi method* or shorter CJ method:

How to define an accurate hybrid method?

We see that just one variant of $LL^T J$ method ($LL^T JA$) and just one variant of $RR^T J$ method ($RR^T JD$) is indicated as relatively accurate.

This shows how to define a highly accurate hybrid method, call it *Cholesky-Jacobi method* or shorter CJ method:

%%% Algorithm CJ

choose the pivot pair (i, j)

if $a_{ii} \geq a_{jj}$ **then** select $LL^T J$ algorithm
else select $RR^T J$ algorithm

endif

How to define an accurate hybrid method?

We see that just one variant of LL^TJ method (LL^TJA) and just one variant of RR^TJ method (RR^TJD) is indicated as relatively accurate.

This shows how to define a highly accurate hybrid method, call it *Cholesky-Jacobi method* or shorter CJ method:

%%% Algorithm CJ

choose the pivot pair (i, j)
if $a_{ii} \geq a_{jj}$ **then** select LL^TJ algorithm
else select RR^TJ algorithm
endif

Its global convergence is proved in an earlier theorem.

How to define an accurate hybrid method?

We see that just one variant of LL^TJ method (LL^TJA) and just one variant of RR^TJ method (RR^TJD) is indicated as relatively accurate.

This shows how to define a highly accurate hybrid method, call it *Cholesky-Jacobi method* or shorter CJ method:

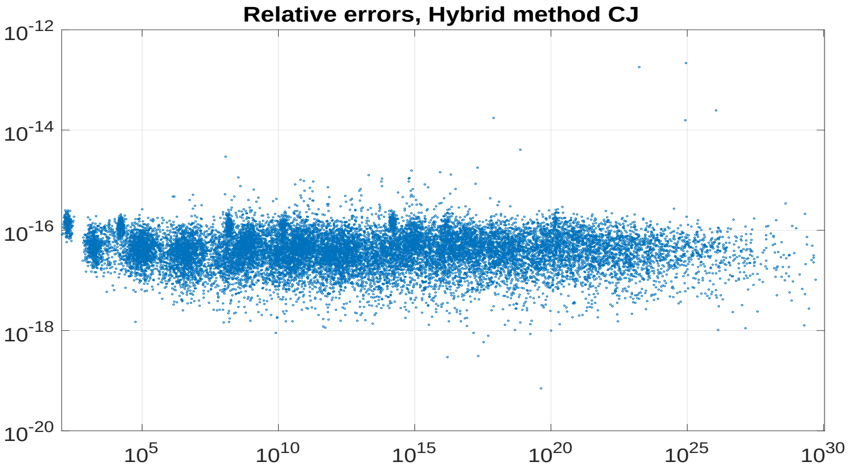
%%% Algorithm CJ

choose the pivot pair (i, j)
if $a_{ii} \geq a_{jj}$ **then** select LL^TJ algorithm
else select RR^TJ algorithm
endif

Its global convergence is proved in an earlier theorem.

We end this presentation with the graph associated with the CJ method.

Relative errors: *CJ* method



Thank you for your time