Implicit Jacobi Algorithms for the Symmetric Eigenproblem

Froilán M. Dopico

Departamento de Matemáticas, Universidad Carlos III de Madrid

15-th ILAS Conference, Cancún, México, 16-20 June, 2008

Jacobi Algorithm: Brief History

- The Jacobi algorithm computes **eigenvalues and eigenvectors of real symmetric matrices**.
- It is one of the earliest methods in numerical analysis, dating to 1846. **It is older than matrix theory itself**.
- It was the standard procedure in 1950s for solving dense symmetric eigenvalue problems before the faster QR algorithm was developed...
- It was forgotten but from the 1980s it came back to scene because of its adaptability to parallel computers, and
- from the 1990s **because sometimes, through special implementations, Jacobi algorithm is able to compute eigenvalues and eigenvectors much more accurately than any other method.**

Jacobi Algorithm: Brief History

- The Jacobi algorithm computes **eigenvalues and eigenvectors of real symmetric matrices**.
- It is one of the earliest methods in numerical analysis, dating to 1846. **It is older than matrix theory itself**.
- It was the standard procedure in 1950s for solving dense symmetric eigenvalue problems before the faster QR algorithm was developed...
- It was forgotten but from the 1980s it came back to scene because of its adaptability to parallel computers, and
- from the 1990s **because sometimes, through special implementations, Jacobi algorithm is able to compute eigenvalues and eigenvectors much more accurately than any other method.**

Jacobi Algorithm: Brief History

- The Jacobi algorithm computes **eigenvalues and eigenvectors of real symmetric matrices**.
- It is one of the earliest methods in numerical analysis, dating to 1846. **It is older than matrix theory itself**.
- It was the standard procedure in 1950s for solving dense symmetric eigenvalue problems before the faster QR algorithm was developed...
- It was forgotten but from the 1980s it came back to scene because of its adaptability to parallel computers, and
- from the 1990s **because sometimes, through special implementations, Jacobi algorithm is able to compute eigenvalues and eigenvectors much more accurately than any other method.**

Jacobi Algorithm: Brief History

- The Jacobi algorithm computes **eigenvalues and eigenvectors of real symmetric matrices**.
- It is one of the earliest methods in numerical analysis, dating to 1846. **It is older than matrix theory itself**.
- It was the standard procedure in 1950s for solving dense symmetric eigenvalue problems before the faster QR algorithm was developed...
- It was forgotten but from the 1980s it came back to scene because of its adaptability to parallel computers, and
- from the 1990s **because sometimes, through special implementations, Jacobi algorithm is able to compute eigenvalues and eigenvectors much more accurately than any other method.**

Jacobi Algorithm: Brief History

- The Jacobi algorithm computes **eigenvalues and eigenvectors of real symmetric matrices**.
- It is one of the earliest methods in numerical analysis, dating to 1846. **It is older than matrix theory itself**.
- It was the standard procedure in 1950s for solving dense symmetric eigenvalue problems before the faster QR algorithm was developed...
- It was forgotten but from the 1980s it came back to scene because of its adaptability to parallel computers, and
- from the 1990s **because sometimes, through special implementations, Jacobi algorithm is able to compute eigenvalues and eigenvectors much more accurately than any other method.**

Jacobi Algorithm: Basics (I)

It is very easy to diagonalize 2×2 symmetric matrices.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\tau = \frac{a_{ii} - a_{jj}}{2\, a_{ij}}$$

$$t = \frac{\mathrm{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}$$

$$\cos\theta = \frac{1}{\sqrt{1+t^2}} \quad , \quad \sin\theta = \frac{t}{\sqrt{1+t^2}}$$

$$\lambda_1 = a_{ii} + a_{ij}\, t$$

$$\lambda_2 = a_{jj} - a_{ij}\, t$$

Denote for simplicity $c \equiv \cos\theta$ and $s \equiv \sin\theta$.

Jacobi Algorithm: Basics (I)

It is very easy to diagonalize 2×2 symmetric matrices.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\tau = \frac{a_{ii} - a_{jj}}{2\, a_{ij}}$$

$$t = \frac{\mathrm{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}$$

$$\cos\theta = \frac{1}{\sqrt{1+t^2}} \quad , \quad \sin\theta = \frac{t}{\sqrt{1+t^2}}$$

$$\lambda_1 = a_{ii} + a_{ij}\, t$$

$$\lambda_2 = a_{jj} - a_{ij}\, t$$

Denote for simplicity $c = \cos\theta$ and $s = \sin\theta$.

Jacobi Algorithm: Basics (I)

It is very easy to diagonalize 2×2 symmetric matrices.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\tau = \frac{a_{ii} - a_{jj}}{2\, a_{ij}}$$

$$t = \frac{\mathsf{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}$$

$$\cos\theta = \frac{1}{\sqrt{1+t^2}} \quad , \quad \sin\theta = \frac{t}{\sqrt{1+t^2}}$$

$$\lambda_1 = a_{ii} + a_{ij}\, t$$

$$\lambda_2 = a_{jj} - a_{ij}\, t$$

Denote for simplicity $c = \cos\theta$ and $s = \sin\theta$.

Jacobi Algorithm: Basics (I)

It is very easy to diagonalize 2×2 symmetric matrices.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\begin{aligned}
\tau &= \frac{a_{ii} - a_{jj}}{2\, a_{ij}} \\
t &= \frac{\text{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}} \\
\cos\theta &= \frac{1}{\sqrt{1+t^2}} \quad , \quad \sin\theta = \frac{t}{\sqrt{1+t^2}} \\
\lambda_1 &= a_{ii} + a_{ij}\, t \\
\lambda_2 &= a_{jj} - a_{ij}\, t
\end{aligned}$$

Denote for simplicity $c \equiv \cos\theta$ and $s \equiv \sin\theta$.

Jacobi Algorithm: Basics (II)

Then, given $A = A^T \in \mathbb{R}^{n\times n}$, the previous expressions can be used to compute a plane rotation

$$R(i,j,c,s) \quad = \quad \begin{array}{cc} & \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array} & \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & -s & & \\ & & & \ddots & & & \\ & & s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \end{array} \in \mathbb{R}^{n\times n},$$

such that

$$\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$$

The Jacobi Algorithm

INPUT: $A = A^T \in \mathbb{R}^{n \times n}$
OUTPUT: e-values, λ_k, and matrix of e-vectors, U, of A

$U = I_n$

repeat

choose a pair $i \neq j$

compute c and s such that $\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$

$A = R(i,j,c,s)^T A R(i,j,c,s)$

$U = U\, R(i,j,c,s)$

until A is sufficiently diagonal

$\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n.$

Remarks

- Each step costs $6\,n$ operations.
- Each step only modifies rows and columns i and j (parallelism).
- The steps do not preserve previous zeros.

The Jacobi Algorithm

INPUT: $A = A^T \in \mathbb{R}^{n\times n}$
OUTPUT: e-values, λ_k, and matrix of e-vectors, U, of A

$U = I_n$

repeat
  choose a pair $i \neq j$
  compute c and s such that $\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$
  $A = R(i,j,c,s)^T A R(i,j,c,s)$
  $U = U\, R(i,j,c,s)$
until A is sufficiently diagonal

$\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n.$

Remarks

- Each step costs $6\,n$ operations.
- Each step only modifies rows and columns i and j (parallelism).
- The steps do not preserve previous zeros.

How to pick (i,j) pairs

Classical strategy

- Choose in each step (i,j) such that $|a_{ij}| = \max_{k\neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2-n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r} (1,2),(1,3),\ldots,(1,n) \\ (2,3),\ldots,(2,n) \\ \cdots\cdots \\ (n-1,n) \end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i,j) pairs

Classical strategy

- Choose in each step (i,j) such that $|a_{ij}| = \max_{k\neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2-n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r} (1,2),(1,3),\ldots,(1,n) \\ (2,3),\ldots,(2,n) \\ \ldots\ldots \\ (n-1,n) \end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i,j) pairs

Classical strategy

- Choose in each step (i,j) such that $|a_{ij}| = \max_{k \neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2-n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$
\begin{array}{l}
(1,2),(1,3),\ldots,(1,n) \\
\qquad\quad (2,3),\ldots,(2,n) \\
\qquad\qquad\qquad \cdots\cdots \\
\qquad\qquad\qquad\quad (n-1,n)
\end{array}
$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i,j) pairs

Classical strategy

- Choose in each step (i,j) such that $|a_{ij}| = \max_{k\neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2-n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r} (1,2),(1,3),\dots,(1,n) \\ (2,3),\dots,(2,n) \\ \dots\dots \\ (n-1,n) \end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i, j) pairs

Classical strategy

- Choose in each step (i, j) such that $|a_{ij}| = \max_{k \neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2 - n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r}(1,2),(1,3),\ldots,(1,n)\\ (2,3),\ldots,(2,n)\\ \ldots\ldots\\ (n-1,n)\end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i, j) pairs

Classical strategy

- Choose in each step (i, j) such that $|a_{ij}| = \max_{k \neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2 - n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r}
(1,2),(1,3),\ldots,(1,n) \\
(2,3),\ldots,(2,n) \\
\ldots\ldots \\
(n-1,n)
\end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i, j) pairs

Classical strategy

- Choose in each step (i, j) such that $|a_{ij}| = \max_{k \neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2 - n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r} (1,2), (1,3), \ldots, (1,n) \\ (2,3), \ldots, (2,n) \\ \ldots\ldots \\ (n-1, n) \end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

How to pick (i, j) pairs

Classical strategy

- Choose in each step (i, j) such that $|a_{ij}| = \max_{k \neq l} |a_{kl}|$.
- **No practical:** $\frac{n^2-n}{2}$ search for cost $6\,n$ in each step.

Cyclic-by-row strategy

$$\begin{array}{r} (1,2),(1,3),\ldots,(1,n) \\ (2,3),\ldots,(2,n) \\ \ldots\ldots \\ (n-1,n) \end{array}$$

A whole cycle is called a **sweep**.

Convergence of Cyclic-by-row strategy

- It is globally convergent (Forsythe and Henrici (1960)).
- It is quadratically convergent (ultimately) (Wilkinson (1962)).

Stopping Criterion

INPUT: $A = A^T \in \mathbb{R}^{n\times n}$

$U = I_n$

repeat

choose a pair $i \neq j$

compute c and s such that $\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$

$A = R(i,j,c,s)^T A R(i,j,c,s)$

$U = U\, R(i,j,c,s)$

until A is sufficiently diagonal

$\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n.$

Two options

- $\sqrt{\sum_{k\neq l} |a_{kl}|^2} \leq \text{tol}\, \|A\|_F$ (**basic**)
- $\frac{|a_{ij}|}{\sqrt{|a_{ii} a_{jj}|}} \leq \text{tol}$ for all $i \neq j$ (**accurate**, it is used in this talk)

Usually $\text{tol} = O(\epsilon)$, where ϵ is the machine precision.

Stopping Criterion

INPUT: $A = A^T \in \mathbb{R}^{n \times n}$

$U = I_n$

repeat

choose a pair $i \neq j$

compute c and s such that $\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$

$A = R(i,j,c,s)^T A R(i,j,c,s)$

$U = U\, R(i,j,c,s)$

until A is sufficiently diagonal

$\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n.$

Two options

- $\sqrt{\sum_{k \neq l} |a_{kl}|^2} \leq \text{tol}\, \|A\|_F$ (**basic**)
- $\frac{|a_{ij}|}{\sqrt{|a_{ii} a_{jj}|}} \leq \text{tol}$ for all $i \neq j$ (**accurate**, it is used in this talk)

Usually tol $= O(\epsilon)$, where ϵ is the machine precision.

Stopping Criterion

INPUT: $A = A^T \in \mathbb{R}^{n\times n}$

$U = I_n$

repeat

choose a pair $i \neq j$

compute c and s such that $\left(R(i,j,c,s)^T A R(i,j,c,s)\right)_{ij} = 0$

$A = R(i,j,c,s)^T A R(i,j,c,s)$

$U = U\, R(i,j,c,s)$

until A is sufficiently diagonal

$\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n$.

Two options

- $\sqrt{\sum_{k\neq l} |a_{kl}|^2} \leq \text{tol}\, \|A\|_F$ (**basic**)
- $\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \text{tol}$ for all $i \neq j$ (**accurate**, it is used in this talk)

Usually tol $= O(\epsilon)$, where ϵ is the machine precision.

Errors in eigenvalues computed by Jacobi, QR,....

- We restrict to eigenvalues for simplicity in this talk, also results on eigenvectors.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, **Jacobi, QR, divide and conquer,...** are **backward stable**, i.e., the computed eigenvalues $\widehat{\lambda}_1 \geq \ldots \geq \widehat{\lambda}_n$ are the exact eigenvalues of

$$A + E, \quad \text{with } \|E\|_2 = O(\epsilon)\|A\|_2$$

  where $\epsilon \approx 10^{-16}$ in double precision.
- If $\lambda_1 \geq \ldots \geq \lambda_n$ are the eigenvalues of A then Weyl's perturbation theorem implies

$$|\widehat{\lambda}_i - \lambda_i| \leq \|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{for all } i$$

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\frac{\|A\|_2}{|\lambda_i|} \leq O(\epsilon)\kappa(A) \quad \text{for all } i,$$

  because $\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$.

Errors in eigenvalues computed by Jacobi, QR,....

- We restrict to eigenvalues for simplicity in this talk, also results on eigenvectors.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, **Jacobi, QR, divide and conquer,...** are **backward stable**, i.e., the computed eigenvalues $\widehat{\lambda}_1 \geq \ldots \geq \widehat{\lambda}_n$ are the exact eigenvalues of

$$A + E, \quad \text{with } \|E\|_2 = O(\epsilon)\|A\|_2$$

  where $\epsilon \approx 10^{-16}$ in double precision.
- If $\lambda_1 \geq \ldots \geq \lambda_n$ are the eigenvalues of A then Weyl's perturbation theorem implies

$$|\widehat{\lambda}_i - \lambda_i| \leq \|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{for all } i$$

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\frac{\|A\|_2}{|\lambda_i|} \leq O(\epsilon)\kappa(A) \quad \text{for all } i,$$

  because $\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$. Very large if $\kappa(A) \geq \frac{1}{\epsilon} \approx 10^{16}$

Errors in eigenvalues computed by Jacobi, QR,....

- We restrict to eigenvalues for simplicity in this talk, also results on eigenvectors.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, **Jacobi, QR, divide and conquer,...** are **backward stable**, i.e., the computed eigenvalues $\widehat{\lambda}_1 \geq \ldots \geq \widehat{\lambda}_n$ are the exact eigenvalues of

$$A + E, \quad \text{with } \|E\|_2 = O(\epsilon)\|A\|_2$$

where $\epsilon \approx 10^{-16}$ in double precision.
- If $\lambda_1 \geq \ldots \geq \lambda_n$ are the eigenvalues of A then Weyl's perturbation theorem implies

$$|\widehat{\lambda}_i - \lambda_i| \leq \|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{for all } i$$

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\frac{\|A\|_2}{|\lambda_i|} \leq O(\epsilon)\kappa(A) \quad \text{for all } i,$$

because $\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$. Very large if $\kappa(A) \geq \frac{1}{\epsilon} \approx 10^{16}$.

Errors in eigenvalues computed by Jacobi, QR,....

- We restrict to eigenvalues for simplicity in this talk, also results on eigenvectors.
- Given $A = A^T \in \mathbb{R}^{n\times n}$, **Jacobi, QR, divide and conquer,...** are **backward stable**, i.e., the computed eigenvalues $\widehat{\lambda}_1 \geq \ldots \geq \widehat{\lambda}_n$ are the exact eigenvalues of

$$A + E, \quad \text{with } \|E\|_2 = O(\epsilon)\|A\|_2$$

  where $\epsilon \approx 10^{-16}$ in double precision.
- If $\lambda_1 \geq \ldots \geq \lambda_n$ are the eigenvalues of A then Weyl's perturbation theorem implies

$$|\widehat{\lambda}_i - \lambda_i| \leq \|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{for all } i$$

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\frac{\|A\|_2}{|\lambda_i|} \leq O(\epsilon)\kappa(A) \quad \text{for all } i,$$

  because $\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$. Very large if $\kappa(A) \geq \frac{1}{\epsilon} \approx 10^{16}$.

Errors in eigenvalues computed by Jacobi, QR,....

- We restrict to eigenvalues for simplicity in this talk, also results on eigenvectors.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, **Jacobi, QR, divide and conquer,...** are **backward stable**, i.e., the computed eigenvalues $\widehat{\lambda}_1 \geq \ldots \geq \widehat{\lambda}_n$ are the exact eigenvalues of

$$A + E, \quad \text{with } \|E\|_2 = O(\epsilon)\|A\|_2$$

  where $\epsilon \approx 10^{-16}$ in double precision.
- If $\lambda_1 \geq \ldots \geq \lambda_n$ are the eigenvalues of A then Weyl's perturbation theorem implies

$$|\widehat{\lambda}_i - \lambda_i| \leq \|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{for all } i$$

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\frac{\|A\|_2}{|\lambda_i|} \leq O(\epsilon)\kappa(A) \quad \text{for all } i,$$

  because $\kappa(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$. Very large if $\kappa(A) \geq \frac{1}{\epsilon} \approx 10^{16}$.

Famous example: 100×100 Hilbert matrix

$$h_{ij} = \frac{1}{i+j-1}, \qquad 1 \leq i, j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.77970086283480 2 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

- **Can we do anything better?**

Famous example: 100×100 Hilbert matrix

$$h_{ij} = \frac{1}{i+j-1}, \qquad 1 \leq i,j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

- **Can we do anything better?**

Famous example: 100×100 Hilbert matrix

$$h_{ij} = \frac{1}{i+j-1}, \qquad 1 \le i,j \le 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

- Can we do anything better?

Famous example: 100×100 Hilbert matrix

$$h_{ij} = \frac{1}{i+j-1}, \qquad 1 \leq i, j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

- Can we do anything better?

Famous example: 100×100 Hilbert matrix

$$h_{ij} = \frac{1}{i+j-1}, \qquad 1 \leq i,j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

- **Can we do anything better?**

Outline

Outline

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,

  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.

- HRA is only possible for **special types of matrices**.

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,

  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.

- HRA is only possible for **special types of matrices**.

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,
  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.
- HRA is only possible for **special types of matrices**.

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,
  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.
- HRA is only possible for **special types of matrices**.

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,
  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.
- HRA is only possible for **special types of matrices**.

Overview

- In the last twenty years an intensive research effort has been made to compute eigenvalues and eigenvectors of $n \times n$ **symmetric matrices** **to high relative accuracy (hra)**.
- Given $A = A^T \in \mathbb{R}^{n \times n}$, we will say that an algorithm computes **all** its **eigenvalues** to **hra** if the computed eigenvalues satisfy

$$|\widehat{\lambda}_i - \lambda_i| = O(\epsilon)\,|\lambda_i| \quad \text{for all} \quad i$$

  and, in addition,
  1. the cost is $O(n^3)$ flops,
  2. and extra precision is not used.
- HRA is only possible for **special types of matrices**.

The beginning: Implicit zero-shift QR for singular values of bidiagonal matrices. Demmel-Kahan 1990

- All of the singular values of **any bidiagonal matrix** B can be computed with high relative accuracy.
- A variation of the QR iteration is needed (or `dqds` by Fernando and Parlett 1994).
- **Consequence:** the eigenvalues of **any positive definite tridiagonal matrix** $B^T B$ can be computed with high relative accuracy **if its Cholesky factor** B **is known.**
- If for **a positive definite tridiagonal matrix only its entries are known**, then **we cannot compute** its eigenvalues with guaranteed high relative accuracy.
- **General rule for accurate computations:** a good representation of the matrix is essential.

The beginning: Implicit zero-shift QR for singular values of bidiagonal matrices. Demmel-Kahan 1990

- All of the singular values of **any bidiagonal matrix** B can be computed with high relative accuracy.
- A variation of the QR iteration is needed (or `dqds` by Fernando and Parlett 1994).
- **Consequence:** the eigenvalues of **any positive definite tridiagonal matrix** $B^T B$ can be computed with high relative accuracy **if its Cholesky factor** B **is known.**
- If for **a positive definite tridiagonal matrix only its entries are known**, then **we cannot compute** its eigenvalues with guaranteed high relative accuracy.
- **General rule for accurate computations:** a good representation of the matrix is essential.

The beginning: Implicit zero-shift QR for singular values of bidiagonal matrices. Demmel-Kahan 1990

- All of the singular values of **any bidiagonal matrix** B can be computed with high relative accuracy.
- A variation of the QR iteration is needed (or `dqds` by Fernando and Parlett 1994).
- **Consequence:** the eigenvalues of **any positive definite tridiagonal matrix** $B^T B$ can be computed with high relative accuracy **if its Cholesky factor** B **is known.**
- If for **a positive definite tridiagonal matrix only its entries are known**, then **we cannot compute** its eigenvalues with guaranteed high relative accuracy.
- **General rule for accurate computations:** a good representation of the matrix is essential.

The beginning: Implicit zero-shift QR for singular values of bidiagonal matrices. Demmel-Kahan 1990

- All of the singular values of **any bidiagonal matrix** B can be computed with high relative accuracy.
- A variation of the QR iteration is needed (or `dqds` by Fernando and Parlett 1994).
- **Consequence:** the eigenvalues of **any positive definite tridiagonal matrix** $B^T B$ can be computed with high relative accuracy **if its Cholesky factor** B **is known.**
- If for **a positive definite tridiagonal matrix only its entries are known**, then **we cannot compute** its eigenvalues with guaranteed high relative accuracy.
- **General rule for accurate computations:** a good representation of the matrix is essential.

The beginning: Implicit zero-shift QR for singular values of bidiagonal matrices. Demmel-Kahan 1990

- All of the singular values of **any bidiagonal matrix** B can be computed with high relative accuracy.
- A variation of the QR iteration is needed (or `dqds` by Fernando and Parlett 1994).
- **Consequence:** the eigenvalues of **any positive definite tridiagonal matrix** $B^T B$ can be computed with high relative accuracy **if its Cholesky factor** B **is known.**
- If for **a positive definite tridiagonal matrix** **only its entries are known**, then **we cannot compute** its eigenvalues with guaranteed high relative accuracy.
- **General rule for accurate computations:** a good representation of the matrix is essential.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \dots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\,\kappa(A)$.
- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \ldots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\,\kappa(A)$.
- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \ldots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

not $O(\epsilon)\,\kappa(A)$.

- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \dots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\, \kappa(DAD) \quad \text{for all} \quad i,$$

not $O(\epsilon)\, \kappa(A)$.

- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \operatorname{diag}\left(\frac{1}{\sqrt{a_{11}}}, \dots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\,\kappa(A)$.
- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \ldots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\, \kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\, \kappa(A)$.
- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \ldots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\,\kappa(A)$.
- $\kappa(DAD) \leq n \min_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

A second milestone: Jacobi can be more accurate than QR. Demmel-Veselić 1992

- Let $A = A^T$ be **positive definite.**
- Let $D = \text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \ldots, \frac{1}{\sqrt{a_{nn}}}\right)$.
- Then Jacobi algorithm computes the eigenvalues with errors

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} = O(\epsilon)\,\kappa(DAD) \quad \text{for all} \quad i,$$

  not $O(\epsilon)\,\kappa(A)$.
- $\kappa(DAD) \leq n \min\limits_{D' \text{ diagonal}} \kappa(D'AD')$.
- It is **one of the two types** of symmetric matrices for which **direct application of Jacobi** gives high relative accuracy. The other type is scaled diagonally dominant matrices (Matejaš, 2008).
- **Very restrictive.** It does not cover Hilbert matrix for instance.

Example: Jacobi on positive definite well scalable matrix

$$A = \begin{bmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{bmatrix} \qquad \kappa(A) = 1.019 \cdot 10^{40}$$

Computed Eigenvalues:

exacts	MATLAB	Jacobi
$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$
$9.900000000000005 \cdot 10^{19}$	$-8.100009764062724 \cdot 10^{19}$	$9.900000000000000 \cdot 10^{19}$
$9.818181818181818 \cdot 10^{-1}$	$-1.208925819614629 \cdot 10^{24}$	$9.818181818181818 \cdot 10^{-1}$

$$\begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} A \begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} = \begin{bmatrix} 1 & 10^{-1} & 10^{-1} \\ 10^{-1} & 1 & 10^{-1} \\ 10^{-1} & 10^{-1} & 1 \end{bmatrix} = B$$

$$\kappa(B) = 1.33$$

Example: Jacobi on positive definite well scalable matrix

$$A = \begin{bmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{bmatrix} \qquad \kappa(A) = 1.019 \cdot 10^{40}$$

Computed Eigenvalues:

exacts	MATLAB	Jacobi
$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$
$9.900000000000005 \cdot 10^{19}$	$-8.100009764062724 \cdot 10^{19}$	$9.900000000000000 \cdot 10^{19}$
$9.818181818181818 \cdot 10^{-1}$	$-1.208925819614629 \cdot 10^{24}$	$9.818181818181818 \cdot 10^{-1}$

$$\begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} A \begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} = \begin{bmatrix} 1 & 10^{-1} & 10^{-1} \\ 10^{-1} & 1 & 10^{-1} \\ 10^{-1} & 10^{-1} & 1 \end{bmatrix} \equiv B$$

$$\kappa(B) = 1.33$$

Example: Jacobi on positive definite well scalable matrix

$$A = \begin{bmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{bmatrix} \qquad \kappa(A) = 1.019 \cdot 10^{40}$$

Computed Eigenvalues:

exacts	MATLAB	Jacobi
$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$	$1.000000000000000 \cdot 10^{40}$
$9.900000000000005 \cdot 10^{19}$	$-8.100009764062724 \cdot 10^{19}$	$9.900000000000000 \cdot 10^{19}$
$9.818181818181818 \cdot 10^{-1}$	$-1.208925819614629 \cdot 10^{24}$	$9.818181818181818 \cdot 10^{-1}$

$$\begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} A \begin{bmatrix} 10^{-20} & & \\ & 10^{-10} & \\ & & 1 \end{bmatrix} = \begin{bmatrix} 1 & 10^{-1} & 10^{-1} \\ 10^{-1} & 1 & 10^{-1} \\ 10^{-1} & 10^{-1} & 1 \end{bmatrix} \equiv B$$

$$\kappa(B) = 1.33$$

Selected references for HRA algorithms for symmetric eigenproblems (SVDs)

- Demmel-Kahan (1990), Barlow-Demmel (1990), Demmel-Veselić (1992), Demmel-Gragg (1993), Demmel (1999)
- Veselić-Slapničar (1992, 93, 03)
- Fernando-Parlett (1994)
- Drmač (1998, 99), Drmač-Veselić (2008)
- Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač (1999)
- Demmel-Koev (2001, 04, 06), Koev (2005, 07)
- D-Molera-Moro(03),D-Koev(06,07),Peláez-Moro(06),D-Molera(08)
- Ye (2008)
- It has motivated Spectral Relative Perturbation Theory (Eisenstat, Ipsen, R.C. Li, Mathias)
- Improved Convergence analysis of Jacobi Algorithms (Drmač, Hari, Matejas).
- Application to MRRR $O(n^2)$-algorithm by Dhillon and Parlett.
- Analysis of block Jacobi methods (Hari, Drmač)...

Selected references for HRA algorithms for symmetric eigenproblems (SVDs)

- Demmel-Kahan (1990), Barlow-Demmel (1990), Demmel-Veselić (1992), Demmel-Gragg (1993), Demmel (1999)
- Veselić-Slapničar (1992, 93, 03)
- Fernando-Parlett (1994)
- Drmač (1998, 99), Drmač-Veselić (2008)
- Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač (1999)
- Demmel-Koev (2001, 04, 06), Koev (2005, 07)
- D-Molera-Moro(03),D-Koev(06,07),Peláez-Moro(06),D-Molera(08)
- Ye (2008)
- It has motivated Spectral Relative Perturbation Theory (Eisenstat, Ipsen, R.C. Li, Mathias)
- Improved Convergence analysis of Jacobi Algorithms (Drmač, Hari, Matejas).
- Application to MRRR $O(n^2)$-algorithm by Dhillon and Parlett.
- Analysis of block Jacobi methods (Hari, Drmač)...

Selected references for HRA algorithms for symmetric eigenproblems (SVDs)

- Demmel-Kahan (1990), Barlow-Demmel (1990), Demmel-Veselić (1992), Demmel-Gragg (1993), Demmel (1999)
- Veselić-Slapničar (1992, 93, 03)
- Fernando-Parlett (1994)
- Drmač (1998, 99), Drmač-Veselić (2008)
- Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač (1999)
- Demmel-Koev (2001, 04, 06), Koev (2005, 07)
- D-Molera-Moro(03),D-Koev(06,07),Peláez-Moro(06),D-Molera(08)
- Ye (2008)
- It has motivated Spectral Relative Perturbation Theory (Eisenstat, Ipsen, R.C. Li, Mathias)
- Improved Convergence analysis of Jacobi Algorithms (Drmač, Hari, Matejas).
- Application to MRRR $O(n^2)$-algorithm by Dhillon and Parlett.
- Analysis of block Jacobi methods (Hari, Drmač)...

Selected references for HRA algorithms for symmetric eigenproblems (SVDs)

- Demmel-Kahan (1990), Barlow-Demmel (1990), Demmel-Veselić (1992), Demmel-Gragg (1993), Demmel (1999)
- Veselić-Slapničar (1992, 93, 03)
- Fernando-Parlett (1994)
- Drmač (1998, 99), Drmač-Veselić (2008)
- Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač (1999)
- Demmel-Koev (2001, 04, 06), Koev (2005, 07)
- D-Molera-Moro(03),D-Koev(06,07),Peláez-Moro(06),D-Molera(08)
- Ye (2008)
- It has motivated Spectral Relative Perturbation Theory (Eisenstat, Ipsen, R.C. Li, Mathias)
- Improved Convergence analysis of Jacobi Algorithms (Drmač, Hari, Matejas).
- Application to MRRR $O(n^2)$-algorithm by Dhillon and Parlett.
- Analysis of block Jacobi methods (Hari, Drmač)...

Selected references for HRA algorithms for symmetric eigenproblems (SVDs)

- Demmel-Kahan (1990), Barlow-Demmel (1990), Demmel-Veselić (1992), Demmel-Gragg (1993), Demmel (1999)
- Veselić-Slapničar (1992, 93, 03)
- Fernando-Parlett (1994)
- Drmač (1998, 99), Drmač-Veselić (2008)
- Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač (1999)
- Demmel-Koev (2001, 04, 06), Koev (2005, 07)
- D-Molera-Moro(03),D-Koev(06,07),Peláez-Moro(06),D-Molera(08)
- Ye (2008)
- It has motivated Spectral Relative Perturbation Theory (Eisenstat, Ipsen, R.C. Li, Mathias)
- Improved Convergence analysis of Jacobi Algorithms (Drmač, Hari, Matejas).
- Application to MRRR $O(n^2)$-algorithm by Dhillon and Parlett.
- Analysis of block Jacobi methods (Hari, Drmač)...

Outline

Key unifying idea: Rank Revealing Decompositions (RRD) (Demmel et al. 1999)

- **The world of high relative accuracy algorithms** for computing eigenvalues of symmetric matrices and SVDs of general matrices **was a *jungle* until 1999**.
- There were **QR** methods for SVDs, **Jacobi** methods for positive definite matrices and SVDs, **bisection** methods for scaled diagonally dominant and for matrices with acyclic graphs, new implementations of the **dqds** method.....
- In 1999 Demmel et al. showed that every class of matrices for which its **SVD** can be accurately computed fits in the unifying framework of **computing first an accurate RRD and then use a Jacobi type algorithm on the decomposition**.

Key unifying idea: Rank Revealing Decompositions (RRD) (Demmel et al. 1999)

- **The world of high relative accuracy algorithms** for computing eigenvalues of symmetric matrices and SVDs of general matrices **was a *jungle* until 1999**.
- There were **QR** methods for SVDs, **Jacobi** methods for positive definite matrices and SVDs, **bisection** methods for scaled diagonally dominant and for matrices with acyclic graphs, new implementations of the **dqds** method.....
- In 1999 Demmel et al. showed that every class of matrices for which its **SVD** can be accurately computed fits in the unifying framework of **computing first an accurate RRD and then use a Jacobi type algorithm on the decomposition**.

Key unifying idea: Rank Revealing Decompositions (RRD) (Demmel et al. 1999)

- **The world of high relative accuracy algorithms** for computing eigenvalues of symmetric matrices and SVDs of general matrices **was a *jungle* until 1999**.
- There were **QR** methods for SVDs, **Jacobi** methods for positive definite matrices and SVDs, **bisection** methods for scaled diagonally dominant and for matrices with acyclic graphs, new implementations of the **dqds** method.....
- In 1999 Demmel et al. showed that every class of matrices for which its **SVD** can be accurately computed fits in the unifying framework of **computing first an accurate RRD and then use a Jacobi type algorithm on the decomposition**.

Symmetric Rank Revealing Decompositions (RRD)

We restrict in this talk to symmetric RRDs of $A = A^T \in \mathbb{R}^{n \times n}$.

- Compute first an **accurate** RRD

$$A = XDX^T,$$

  X is **well-conditioned** and D is **diagonal and nonsingular**.

  Remark: Accuracy is only possible for special types of matrices through structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP.

- Compute eigenvalues and eigenvectors with **high relative accuracy** from the **factors X and D** through a **Jacobi-type** algorithms.

These Jacobi algorithms are the main purpose of this talk!!

Symmetric Rank Revealing Decompositions (RRD)

We restrict in this talk to symmetric RRDs of $A = A^T \in \mathbb{R}^{n\times n}$.

- Compute first an **accurate** RRD

$$A = XDX^T,$$

X is **well-conditioned** and D is **diagonal and nonsingular**.

Remark: **Accuracy** is only possible for special types of matrices through structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP.

- Compute eigenvalues and eigenvectors with **high relative accuracy** from the **factors X and D** through a **Jacobi-type** algorithms.

These Jacobi algorithms are the main purpose of this talk!!

Symmetric Rank Revealing Decompositions (RRD)

We restrict in this talk to symmetric RRDs of $A = A^T \in \mathbb{R}^{n \times n}$.

- Compute first an **accurate** RRD

$$A = XDX^T,$$

  X is **well-conditioned** and D is **diagonal and nonsingular**.

  Remark: Accuracy is only possible for special types of matrices through structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP.

- Compute eigenvalues and eigenvectors with **high relative accuracy** from the **factors** X **and** D through a **Jacobi-type** algorithms.

These Jacobi algorithms are the main purpose of this talk!!

Symmetric Rank Revealing Decompositions (RRD)

We restrict in this talk to symmetric RRDs of $A = A^T \in \mathbb{R}^{n\times n}$.

- Compute first an **accurate** RRD

$$A = XDX^T,$$

  X is **well-conditioned** and D is **diagonal and nonsingular**.

  Remark: Accuracy is only possible for special types of matrices through structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP.

- Compute eigenvalues and eigenvectors with **high relative accuracy** from the **factors** X **and** D through a **Jacobi-type** algorithms.

These Jacobi algorithms are the main purpose of this talk!!

Symmetric Rank Revealing Decompositions (RRD)

We restrict in this talk to symmetric RRDs of $A = A^T \in \mathbb{R}^{n\times n}$.

- Compute first an **accurate** RRD

$$A = XDX^T,$$

  X is **well-conditioned** and D is **diagonal and nonsingular**.

  Remark: Accuracy is only possible for special types of matrices through structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP.

- Compute eigenvalues and eigenvectors with **high relative accuracy** from the **factors X and D** through a **Jacobi-type** algorithms.

These Jacobi algorithms are the main purpose of this talk!!

A symmetric RRD determines accurately its eigenvalues: Example

$$A = XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

$$= \begin{bmatrix} 1 & -2\cdot 10^{50} - 1 & 10^{50}+1 \\ -2\cdot 10^{50} - 1 & 1 & -3\cdot 10^{50} - 1 \\ 10^{50}+1 & -3\cdot 10^{50} - 1 & 3\cdot 10^{50}+1 \end{bmatrix} \qquad (\kappa(X) = 7.21)$$

We consider the **exact eigenvalues** of TWO perturbations of A

- $\widetilde{\mathbf{A}}$: $\widetilde{a}_{33} = (1+10^{-3})\, a_{33}$.
- $\widehat{\mathbf{A}}$: $\widehat{d}_{33} = (1+10^{-3})\, d_{33}$.

A	$\widetilde{\mathbf{A}}$	$\widehat{\mathbf{A}}$
$5.53112887 \cdot 10^{50}$	$5.53291828 \cdot 10^{50}$	$5.53080731 \cdot 10^{50}$
$2.85714285 \cdot 10^{-1}$	$8.56985368 \cdot 10^{46}$	$2.85714285 \cdot 10^{-1}$
$-2.53112887 \cdot 10^{50}$	$-2.53077527 \cdot 10^{50}$	$-2.53380731 \cdot 10^{50}$

A symmetric RRD determines accurately its eigenvalues: Example

$$A = XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

$$= \begin{bmatrix} 1 & -2\cdot 10^{50}-1 & 10^{50}+1 \\ -2\cdot 10^{50}-1 & 1 & -3\cdot 10^{50}-1 \\ 10^{50}+1 & -3\cdot 10^{50}-1 & 3\cdot 10^{50}+1 \end{bmatrix} \qquad (\kappa(X) = 7.21)$$

We consider the **exact eigenvalues** of TWO perturbations of A

- $\widetilde{\mathbf{A}}$: $\widetilde{a}_{33} = (1+10^{-3})\, a_{33}$.
- $\widehat{\mathbf{A}}$: $\widehat{d}_{33} = (1+10^{-3})\, d_{33}$.

A	$\widetilde{\mathbf{A}}$	$\widehat{\mathbf{A}}$
$5.53112887 \cdot 10^{50}$	$5.53291828 \cdot 10^{50}$	$5.53080731 \cdot 10^{50}$
$2.85714285 \cdot 10^{-1}$	$8.56985368 \cdot 10^{46}$	$2.85714285 \cdot 10^{-1}$
$-2.53112887 \cdot 10^{50}$	$-2.53077527 \cdot 10^{50}$	$-2.53380731 \cdot 10^{50}$

A symmetric RRD determines accurately its eigenvalues: Example

$$A = XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

$$= \begin{bmatrix} 1 & -2\cdot 10^{50}-1 & 10^{50}+1 \\ -2\cdot 10^{50}-1 & 1 & -3\cdot 10^{50}-1 \\ 10^{50}+1 & -3\cdot 10^{50}-1 & 3\cdot 10^{50}+1 \end{bmatrix} \qquad (\kappa(X) = 7.21)$$

We consider the **exact eigenvalues** of TWO perturbations of A

- $\widetilde{\mathbf{A}}$: $\widetilde{a}_{33} = (1+10^{-3})\, a_{33}$.
- $\widehat{\mathbf{A}}$: $\widehat{d}_{33} = (1+10^{-3})\, d_{33}$.

A	$\widetilde{\mathbf{A}}$	$\widehat{\mathbf{A}}$
$5.53112887 \cdot 10^{50}$	$5.53291828 \cdot 10^{50}$	$5.53080731 \cdot 10^{50}$
$2.85714285 \cdot 10^{-1}$	$8.56985368 \cdot 10^{46}$	$2.85714285 \cdot 10^{-1}$
$-2.53112887 \cdot 10^{50}$	$-2.53077527 \cdot 10^{50}$	$-2.53380731 \cdot 10^{50}$

A symmetric RRD determines accurately its eigenvalues: Example

$$A = XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

$$= \begin{bmatrix} 1 & -2\cdot 10^{50}-1 & 10^{50}+1 \\ -2\cdot 10^{50}-1 & 1 & -3\cdot 10^{50}-1 \\ 10^{50}+1 & -3\cdot 10^{50}-1 & 3\cdot 10^{50}+1 \end{bmatrix} \qquad (\kappa(X) = 7.21)$$

We consider the **exact eigenvalues** of TWO perturbations of A

- $\widetilde{\mathbf{A}}$: $\widetilde{a}_{33} = (1 + 10^{-3})\, a_{33}$.
- $\widehat{\mathbf{A}}$: $\widehat{d}_{33} = (1 + 10^{-3})\, d_{33}$.

A	$\widetilde{\mathbf{A}}$	$\widehat{\mathbf{A}}$
$5.53112887 \cdot 10^{50}$	$5.53291828 \cdot 10^{50}$	$5.53080731 \cdot 10^{50}$
$2.85714285 \cdot 10^{-1}$	$8.56985368 \cdot 10^{46}$	$2.85714285 \cdot 10^{-1}$
$-2.53112887 \cdot 10^{50}$	$-2.53077527 \cdot 10^{50}$	$-2.53380731 \cdot 10^{50}$

A symmetric RRD determines accurately its eigenvalues: Theorem

Theorem (D, Koev (2006))

Let $A = A^T = XDX^T$ *be an RRD, where* $X \in \mathbb{R}^{n\times r}$, $n \geq r$, *and* $D = \operatorname{diag}(d_1, \ldots, d_r) \in \mathbb{R}^{r\times r}$. *Let* $\widehat{X}$ *and* $\widehat{D} = \operatorname{diag}(\widehat{d}_1, \ldots, \widehat{d}_r)$ *be perturbations of* X *and* D *such that*

$$\frac{\|\widehat{X} - X\|_2}{\|X\|_2} \leq \delta \quad \text{and} \quad \frac{|\widehat{d}_i - d_i|}{|d_i|} \leq \delta \quad \text{for } i = 1, \ldots, r,$$

where $\delta < 1$. *Let* $\lambda_1 \geq \cdots \geq \lambda_n$ *be the eigenvalues of* A *and* $\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_n$ *be the eigenvalues of* $\widehat{X}\widehat{D}\widehat{X}^T$ *then, for all* i,

$$\left|\frac{\lambda_i - \widehat{\lambda}_i}{\lambda_i}\right| \leq \kappa(X)\left(4\delta + 2\delta^2 + \kappa(X)\left(2\delta + \delta^2\right)^2\right) \approx 4\,\delta\,\kappa(X) + O(\delta^2)$$

A symmetric RRD determines accurately its eigenvalues: Proof and multiplicative perturbation theory

Write

$$\widehat{X}\widehat{D}\widehat{X}^T = (I+F)XDX^T(I+F)^T,$$

with $\|F\|_2 \leq (2\delta + \delta^2)\,\kappa(X)$.

Theorem (Eisenstat, Ipsen (1995))

Let $A = A^T \in \mathbb{R}^{n\times n}$ and $\widetilde{A} = (I+F)A(I+F)^T \in \mathbb{R}^{n\times n}$. Let $\lambda_1 \geq \cdots \geq \lambda_n$ and $\widetilde{\lambda}_1 \geq \cdots \geq \widetilde{\lambda}_n$ be, respectively, the eigenvalues of A and $\widetilde{A}$. Then

$$|\widetilde{\lambda}_i - \lambda_i| \leq (2\,\|F\|_2 + \|F\|_2^2)\,|\lambda_i|, \quad \textit{for } i = 1,\ldots,n$$

A symmetric RRD determines accurately its eigenvalues: Proof and multiplicative perturbation theory

Write

$$\widehat{X}\widehat{D}\widehat{X}^T = (I+F)XDX^T(I+F)^T,$$

with $\|F\|_2 \leq (2\delta + \delta^2)\,\kappa(X)$.

Theorem (Eisenstat, Ipsen (1995))

Let $A = A^T \in \mathbb{R}^{n\times n}$ and $\widetilde{A} = (I+F)A(I+F)^T \in \mathbb{R}^{n\times n}$. Let $\lambda_1 \geq \cdots \geq \lambda_n$ and $\widetilde{\lambda}_1 \geq \cdots \geq \widetilde{\lambda}_n$ be, respectively, the eigenvalues of A and $\widetilde{A}$. Then

$$|\widetilde{\lambda}_i - \lambda_i| \leq (2\,\|F\|_2 + \|F\|_2^2)\;|\lambda_i|, \quad \text{for } i = 1, \ldots, n$$

Outline

The accuracy that we need

- The computed factors $\widehat{X}$ and $\widehat{D}$ of an RRD $A = XDX^T$ of $A = A^T$ have to satisfy the **forward error bounds**

$$|D_{ii} - \widehat{D}_{ii}| = O(\epsilon)|D_{ii}|, \quad \text{for all } i$$
$$\|X - \widehat{X}\|_2 = O(\epsilon)\|X\|_2,$$

to guarantee that the **relative errors** between the eigenvalues of $A = XDX^T$ and $\widehat{X}\widehat{D}\widehat{X}^T$ are $O(\epsilon\kappa(X))$.

- **This accuracy** can be obtained **only** for special types of matrices through highly structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP. Each class of matrices needs a different implementation.

The accuracy that we need

- The computed factors $\widehat{X}$ and $\widehat{D}$ of an RRD $A = XDX^T$ of $A = A^T$ have to satisfy the **forward error bounds**

$$|D_{ii} - \widehat{D}_{ii}| = O(\epsilon)|D_{ii}|, \quad \text{for all } i$$
$$\|X - \widehat{X}\|_2 = O(\epsilon)\|X\|_2,$$

  to guarantee that the **relative errors** between the eigenvalues of $A = XDX^T$ and $\widehat{X}\widehat{D}\widehat{X}^T$ are $O(\epsilon\kappa(X))$.

- **This accuracy** can be obtained **only** for special types of matrices through highly structured implementations of Gaussian elimination with complete pivoting (**GECP**), or variations of GECP. Each class of matrices needs a different implementation.

Classes of symmetric matrices with accurate RRDs

1. Well Scaled Symmetric Positive Definite (Demmel and Veselić).
2. Scaled diagonally dominant (Barlow and Demmel)
3. Symmetric Cauchy and Scaled-Cauchy (D and Koev).
4. Symmetric Vandermonde (D and Koev).
5. Symmetric Totally nonnegative (D and Koev).
6. Symmetric Graded Matrices (D and Molera).
7. Symmetric DSTU and TSC (Peláez and Moro).
8. Symmetric diagonally dominant M-matrices (Demmel and Koev), (Peña).
9. Symmetric diagonally dominant (Ye)....

An example: Symmetric Cauchy matrices (I)

$$a_{ij} = \frac{1}{x_i + x_j}, \qquad 1 \leq i, j \leq n$$

Algorithm for accurate RRD (D and Koev (2006))

- Compute accurate Schur Complements (Gohberg, Kailath, Olshevsky) and (Demmel).

$$S_{rs}^{(m)} = S_{rs}^{(m-1)} \frac{(x_r - x_m)(x_s - x_m)}{(x_m + x_s)(x_r + x_m)} \quad \text{for} \quad m+1 \leq r, s \leq n,$$

- Use Diagonal Pivoting Method with the Bunch-Parlett complete pivoting strategy on the Schur Complements to get

$$PAP^T = L\bar{D}L^T,$$

with L block lower triangular, $\bar{D}$ block diagonal matrix with blocks 1×1 or 2×2.

An example: Symmetric Cauchy matrices (I)

$$a_{ij} = \frac{1}{x_i + x_j}, \qquad 1 \leq i, j \leq n$$

Algorithm for accurate RRD (D and Koev (2006))

- Compute accurate Schur Complements (Gohberg, Kailath, Olshevsky) and (Demmel).

$$S_{rs}^{(m)} = S_{rs}^{(m-1)} \frac{(x_r - x_m)(x_s - x_m)}{(x_m + x_s)(x_r + x_m)} \quad \text{for} \quad m+1 \leq r, s \leq n,$$

- Use Diagonal Pivoting Method with the Bunch-Parlett complete pivoting strategy on the Schur Complements to get

$$PAP^T = L\bar{D}L^T,$$

with L block lower triangular, $\bar{D}$ block diagonal matrix with blocks 1×1 or 2×2.

An example: Symmetric Cauchy matrices (I)

$$a_{ij} = \frac{1}{x_i + x_j}, \qquad 1 \le i, j \le n$$

Algorithm for accurate RRD (D and Koev (2006))

- Compute accurate Schur Complements (Gohberg, Kailath, Olshevsky) and (Demmel).

$$S_{rs}^{(m)} = S_{rs}^{(m-1)} \frac{(x_r - x_m)(x_s - x_m)}{(x_m + x_s)(x_r + x_m)} \quad \text{for} \quad m+1 \le r, s \le n,$$

- Use Diagonal Pivoting Method with the Bunch-Parlett complete pivoting strategy on the Schur Complements to get

$$PAP^T = L\bar{D}L^T,$$

with L block lower triangular, $\bar{D}$ block diagonal matrix with blocks 1×1 or 2×2.

An example: Symmetric Cauchy matrices (II)

- Orthogonal diagonalization of the 2 x 2 pivots in $\bar{D} = (U D U^T)$

$$P A P^T = L \bar{D} L^T = L (U D U^T) L^T,$$

-

$$\begin{aligned} A &= (P^T L U)\, D\, (P^T L U)^T \\ &\equiv X\, D\, X^T \end{aligned}$$

Remark

A long and detailed error analysis is needed to prove that the computed RRD is accurate.

Outline

Accurate e-values from X and D: Positive definite case

Algorithm (Demmel, Veselić (1992))

Given RRD $A = XDX^T$ **positive definite**:

1. Compute SVD of
$$X\sqrt{D} = U\Sigma V^T$$
with **one-sided Jacobi on the left**.
2. The spectral decomposition is
$$A = X\sqrt{D}(X\sqrt{D})^T = U\Sigma^2 U^T.$$

Note on one-sided Jacobi

One sided Jacobi on $(X\sqrt{D})$ consists simply in computing the usual Jacobi rotations corresponding to $(X\sqrt{D})(X\sqrt{D})^T$, and apply them only on $(X\sqrt{D}) \longrightarrow R^T\,(X\sqrt{D})$.

Accurate e-values from X and D: Positive definite case

Algorithm (Demmel, Veselić (1992))

Given RRD $A = XDX^T$ **positive definite**:

1. Compute SVD of
$$X\sqrt{D} = U\Sigma V^T$$
with **one-sided Jacobi on the left.**
2. The spectral decomposition is
$$A = X\sqrt{D}(X\sqrt{D})^T = U\Sigma^2 U^T.$$

Note on one-sided Jacobi

One sided Jacobi on $(X\sqrt{D})$ consists simply in computing the usual Jacobi rotations corresponding to $(X\sqrt{D})(X\sqrt{D})^T$, and apply them only on $(X\sqrt{D}) \longrightarrow R^T\,(X\sqrt{D})$.

Accurate RRD computation and One-sided Jacobi in action

100×100 Hilbert Matrix:

$$h_{ij} = \frac{1}{i+j-1}, \quad 1 \leq i,j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0.$
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.77970086283480\,2 \cdot 10^{-151}$
RRD+Jacobi	$5.77970086283481\,3 \cdot 10^{-151}$
MATLAB (eig)	$-1.21607266026676\,0 \cdot 10^{-19}$
Jacobi	$-2.48894364564948\,8 \cdot 10^{-17}$

Accurate RRD computation and One-sided Jacobi in action

100×100 Hilbert Matrix:

$$h_{ij} = \frac{1}{i+j-1}, \quad 1 \le i,j \le 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.77970086283480\mathbf{2} \cdot 10^{-151}$
RRD+Jacobi	$5.779700862834813 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

Accurate RRD computation and One-sided Jacobi in action

100×100 Hilbert Matrix:

$$h_{ij} = \frac{1}{i+j-1}, \quad 1 \leq i, j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
RRD+Jacobi	$5.779700862834813 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

Accurate RRD computation and One-sided Jacobi in action

100×100 Hilbert Matrix:

$$h_{ij} = \frac{1}{i+j-1}, \quad 1 \leq i, j \leq 100$$

- $\lambda_1 > \lambda_2 > \ldots > \lambda_{100} > 0$.
- $\kappa(H) \approx 3.8 \cdot 10^{150}$

	λ_{100}
EXACT	$5.779700862834802 \cdot 10^{-151}$
RRD+Jacobi	$5.779700862834813 \cdot 10^{-151}$
MATLAB (eig)	$-1.216072660266760 \cdot 10^{-19}$
Jacobi	$-2.488943645649488 \cdot 10^{-17}$

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Accurate e-values from X and D: General case

The solution of the **indefinite case** has been much more difficult. A satisfactory algorithm has been found only very recently.

Essentially **two Jacobi** type algorithms were proposed in **the past** for the **indefinite** case. They work well in practice, but **they both have shortcomings**:

- **One-sided Hyperbolic Jacobi** (Slapničar, Veselić (1992,2003)).
  1. It uses hyperbolic transformations (symmetric matrices are diagonalizable by orthogonal similarity).
  2. The error bounds implied by the use of hyperbolic rotations are not rigorously bounded.
- **Signed-SVD** (D., Molera, Moro (2003), D., Molera (2008)),
  1. It does guarantee hra error bounds.
  2. It does not respect the symmetry of the problem.

Our Goal

To present a new and simple **implicit Jacobi algorithm** (D, Koev, Molera 2008) on a given RRD

$$A = XDX^T$$

possibly indefinite that has the following three properties:

1. it computes the eigenvalues and eigenvectors of A to high relative accuracy,
2. it preserves the symmetry of the problem, and
3. it uses only orthogonal transformations.

Our Goal

To present a new and simple **implicit Jacobi algorithm** (D, Koev, Molera 2008) on a given RRD

$$A = XDX^T$$

possibly indefinite that has the following three properties:

1. it computes the eigenvalues and eigenvectors of A to high relative accuracy,
2. it preserves the symmetry of the problem, and
3. it uses only orthogonal transformations.

Our Goal

To present a new and simple **implicit Jacobi algorithm** (D, Koev, Molera 2008) on a given RRD

$$A = XDX^T$$

possibly indefinite that has the following three properties:

1. it computes the eigenvalues and eigenvectors of A to high relative accuracy,
2. it preserves the symmetry of the problem, and
3. it uses only orthogonal transformations.

Our Goal

To present a new and simple **implicit Jacobi algorithm** (D, Koev, Molera 2008) on a given RRD

$$A = XDX^T$$

possibly indefinite that has the following three properties:

1. it computes the eigenvalues and eigenvectors of A to high relative accuracy,
2. it preserves the symmetry of the problem, and
3. it uses only orthogonal transformations.

Outline

Basic Description (1)

- **INPUT:** Factors X and D of a decomposition $A = XDX^T$ of a symmetric matrix, where X is well-conditioned and D is diagonal, perhaps indefinite.
- We run the standard Jacobi algorithm to compute eigenvalues and eigenvectors but applying the rotations only on X.
- **BASIC STEP:** Compute a plane Jacobi rotation R such that $(R^T AR)_{ij} = 0$, for some $i \neq j$, then

$$XDX^T \longrightarrow (R^T X)D(R^T X)^T.$$

- From a decomposition of A we obtain a decomposition of $R^T AR$. The matrix A is never formed.

Basic Description (1)

- **INPUT:** Factors X and D of a decomposition $A = XDX^T$ of a symmetric matrix, where X is well-conditioned and D is diagonal, perhaps indefinite.
- We run the standard Jacobi algorithm to compute eigenvalues and eigenvectors but applying the rotations only on X.
- **BASIC STEP:** Compute a plane Jacobi rotation R such that $(R^T AR)_{ij} = 0$, for some $i \neq j$, then

$$XDX^T \longrightarrow (R^T X)D(R^T X)^T.$$

- From a decomposition of A we obtain a decomposition of $R^T AR$. The matrix A is never formed.

Basic Description (1)

- **INPUT:** Factors X and D of a decomposition $A = XDX^T$ of a symmetric matrix, where X is well-conditioned and D is diagonal, perhaps indefinite.
- We run the standard Jacobi algorithm to compute eigenvalues and eigenvectors but applying the rotations only on X.
- **BASIC STEP:** Compute a plane Jacobi rotation R such that $(R^T AR)_{ij} = 0$, for some $i \neq j$, then

$$XDX^T \longrightarrow (R^T X)D(R^T X)^T.$$

- From a decomposition of A we obtain a decomposition of $R^T AR$. The matrix A is never formed.

Basic Description (1)

- **INPUT:** Factors X and D of a decomposition $A = XDX^T$ of a symmetric matrix, where X is well-conditioned and D is diagonal, perhaps indefinite.
- We run the standard Jacobi algorithm to compute eigenvalues and eigenvectors but applying the rotations only on X.
- **BASIC STEP:** Compute a plane Jacobi rotation R such that $(R^T AR)_{ij} = 0$, for some $i \neq j$, then

$$XDX^T \longrightarrow (R^T X)D(R^T X)^T.$$

- From a decomposition of A we obtain a decomposition of $R^T AR$. The matrix A is never formed.

Basic Description (2)

- Algorithm stops when the off diagonal part of $A_f = X_f D X_f^T$ is small enough.
- **OUTPUT:**
  1. The eigenvalues of A are the computed diagonal entries of $X_f D X_f^T$.
  2. Eigenvectors are the columns of $R_1 R_2 \cdots R_f$
- Let ϵ be the unit roundoff. The **errors** in computed eigenvalues are

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon \kappa(X)) \qquad \text{for all} \quad i,$$

for any condition number of A, i.e., of D. ($\kappa(X) = \|X\|_2 \|X^{-1}\|_2$)

Basic Description (2)

- Algorithm stops when the off diagonal part of $A_f = X_f D X_f^T$ is small enough.
- **OUTPUT:**
  1. The eigenvalues of A are the computed diagonal entries of $X_f D X_f^T$.
  2. Eigenvectors are the columns of $R_1 R_2 \cdots R_f$
- Let ϵ be the unit roundoff. The **errors** in computed eigenvalues are

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon \kappa(X)) \qquad \text{for all} \quad i,$$

for any condition number of A, i.e., of D. ($\kappa(X) = \|X\|_2 \|X^{-1}\|_2$)

Basic Description (2)

- Algorithm stops when the off diagonal part of $A_f = X_f D X_f^T$ is small enough.
- **OUTPUT:**
  1. The eigenvalues of A are the computed diagonal entries of $X_f D X_f^T$.
  2. Eigenvectors are the columns of $R_1 R_2 \cdots R_f$
- Let ϵ be the unit roundoff. The **errors** in computed eigenvalues are

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon \kappa(X)) \qquad \text{for all} \quad i,$$

for any condition number of A, i.e., of D. ($\kappa(X) = \|X\|_2 \|X^{-1}\|_2$)

Basic Description (2)

- Algorithm stops when the off diagonal part of $A_f = X_f D X_f^T$ is small enough.
- **OUTPUT:**
  1. The eigenvalues of A are the computed diagonal entries of $X_f D X_f^T$.
  2. Eigenvectors are the columns of $R_1 R_2 \cdots R_f$
- Let ϵ be the unit roundoff. The **errors** in computed eigenvalues are

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon \kappa(X)) \qquad \text{for all} \quad i,$$

for any condition number of A, i.e., of D. ($\kappa(X) = \|X\|_2 \|X^{-1}\|_2$)

Basic Description (2)

- Algorithm stops when the off diagonal part of $A_f = X_f D X_f^T$ is small enough.
- **OUTPUT:**
  1. The eigenvalues of A are the computed diagonal entries of $X_f D X_f^T$.
  2. Eigenvectors are the columns of $R_1 R_2 \cdots R_f$
- Let ϵ be the unit roundoff. The **errors** in computed eigenvalues are

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon \kappa(X)) \qquad \text{for all} \quad i,$$

for any condition number of A, i.e., of D. ($\kappa(X) = \|X\|_2 \|X^{-1}\|_2$)

Implicit Jacobi for square factors

INPUT: $X \in \mathbb{R}^{n\times n}$ nonsingular and $D \in \mathbb{R}^{n\times n}$ diag. and nonsingular
OUTPUT: e-values, λ_i, and matrix of e-vectors, U, of $A = XDX^T$

$U = I_n$
repeat
  for $i < j$
    compute a_{ii}, a_{ij}, a_{jj} of $A = XDX^T$ and $T = \left[\begin{smallmatrix} c & -s \\ s & c \end{smallmatrix}\right]$, such that

$$T^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} T = \begin{bmatrix} \mu_1 & \\ & \mu_2 \end{bmatrix}$$

$X = R(i,j,c,s)^T X$
$U = U\, R(i,j,c,s)$
  endfor
until convergence $\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \text{tol} = O(\epsilon) \quad \text{for all } i > j \right)$
compute $\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n$.

Outline

Jacobi rotations on X preserve accurate e-values

Lemma (Small multiplicative backward errors of Jacobi rotations)

Let R_i be **exact** *Jacobi rotations and $\widehat{R}_i$ their floating point approximations. Then*

1. $$\widehat{X}_N \equiv \mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = (I+F) R_N^T \cdots R_1^T X,$$

   where $\|F\|_2 = O(N \epsilon \kappa(X))$, and

2. $$\widehat{X}_N D \widehat{X}_N^T = (I+F)(R_1 \cdots R_N)^T X D X^T (R_1 \cdots R_N)(I+F)^T$$

Jacobi rotations on X preserve accurate e-values

Lemma (Small multiplicative backward errors of Jacobi rotations)

Let R_i be **exact** *Jacobi rotations and $\widehat{R}_i$ their floating point approximations. Then*

1. $$\widehat{X}_N \equiv \mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = (I+F) R_N^T \cdots R_1^T X,$$

   where $\|F\|_2 = O(N\,\epsilon\,\kappa(X))$, *and*

2. $$\widehat{X}_N D \widehat{X}_N^T = (I+F)(R_1 \cdots R_N)^T X D X^T (R_1 \cdots R_N)(I+F)^T$$

Jacobi rotations on X preserve accurate e-values

Lemma (Small multiplicative backward errors of Jacobi rotations)

Let R_i be **exact** *Jacobi rotations and $\widehat{R}_i$ their floating point approximations. Then*

1. $$\widehat{X}_N \equiv \mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = (I+F) R_N^T \cdots R_1^T X,$$
   where $\|F\|_2 = O(N\,\epsilon\,\kappa(X))$, and
2. $$\widehat{X}_N D \widehat{X}_N^T = (I+F)(R_1 \cdots R_N)^T X D X^T (R_1 \cdots R_N)(I+F)^T$$

Proof of Rounding Errors in Jacobi rotations

Proof.

Let $U^T = R_N^T \cdots R_1^T$.

- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = R_N^T \cdots R_1^T (X + E)$ with $\|E\|_2 = O(N\epsilon \|X\|_2)$.
- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = U^T(I + EX^{-1})X = (I + U^T E X^{-1} U) U^T X$.
- $\|U^T E X^{-1} U\|_2 = \|E X^{-1}\|_2 = O(N\epsilon\kappa(X))$.

$\square$

Proof of Rounding Errors in Jacobi rotations

Proof.

Let $U^T = R_N^T \cdots R_1^T$.

- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = R_N^T \cdots R_1^T (X + E)$ with $\|E\|_2 = O(N\epsilon \|X\|_2)$.
- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = U^T (I + EX^{-1}) X = (I + U^T E X^{-1} U) U^T X$.
- $\|U^T E X^{-1} U\|_2 = \|EX^{-1}\|_2 = O(N\epsilon\kappa(X))$.

$\square$

Proof of Rounding Errors in Jacobi rotations

Proof.

Let $U^T = R_N^T \cdots R_1^T$.

- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = R_N^T \cdots R_1^T (X + E)$ with $\|E\|_2 = O(N\epsilon \|X\|_2)$.
- $\mathtt{fl}(\widehat{R}_N^T \cdots \widehat{R}_1^T X) = U^T(I + EX^{-1})X = (I + U^T E X^{-1} U) U^T X$.
- $\|U^T E X^{-1} U\|_2 = \|EX^{-1}\|_2 = O(N\epsilon\kappa(X))$.

$\square$

Implicit Jacobi for square factors

INPUT: $X \in \mathbb{R}^{n\times n}$ nonsingular and $D \in \mathbb{R}^{n\times n}$ diag. and nonsingular
OUTPUT: e-values, λ_i, and matrix of e-vectors, U, of $A = XDX^T$

$U = I_n$
repeat
  for $i < j$
    compute a_{ii}, a_{ij}, a_{jj} of $A = XDX^T$ and $T = \left[\begin{smallmatrix} c & -s \\ s & c \end{smallmatrix}\right]$, such that

$$T^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} T = \begin{bmatrix} \mu_1 & \\ & \mu_2 \end{bmatrix}$$

    $X = R(i,j,c,s)^T X$
    $U = U\, R(i,j,c,s)$
  endfor
until convergence $\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \text{tol} = O(\epsilon) \quad \text{for all } i > j \right)$
compute $\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n.$ $\longrightarrow$ IS THIS ACCURATE???

Implicit Jacobi for square factors

INPUT: $X \in \mathbb{R}^{n\times n}$ nonsingular and $D \in \mathbb{R}^{n\times n}$ diag. and nonsingular
OUTPUT: e-values, λ_i, and matrix of e-vectors, U, of $A = XDX^T$

$U = I_n$
repeat
  for $i < j$
    compute a_{ii}, a_{ij}, a_{jj} of $A = XDX^T$ and $T = \left[\begin{smallmatrix} c & -s \\ s & c \end{smallmatrix}\right]$, such that

$$T^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} T = \begin{bmatrix} \mu_1 & \\ & \mu_2 \end{bmatrix}$$

    $X = R(i,j,c,s)^T X$
    $U = U\, R(i,j,c,s)$
  endfor
until convergence $\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \text{tol} = O(\epsilon) \quad \text{for all } i > j \right)$
compute $\lambda_k = a_{kk}$ for $k = 1, 2, \ldots, n$. $\longrightarrow$ IS THIS ACCURATE???

Errors on diagonal entries of almost diagonal RRDs (I)

Given $X \in \mathbb{R}^{n\times n}$ nonsingular and $D = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ diagonal and nonsingular:

- Assume that $A = XDX^T$ satisfies $\dfrac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} = O(\epsilon)$ for all $i > j$.
- $a_{ii} = \displaystyle\sum_{k=1}^{n} x_{ik}^2 d_k$
- $$\left| \frac{\mathtt{fl}(a_{ii}) - a_{ii}}{a_{ii}} \right| \leq \frac{(n+1)\epsilon}{1-(n+1)\epsilon} \, \frac{\displaystyle\sum_{k=1}^{n} x_{ik}^2 |d_k|}{\left|\displaystyle\sum_{k=1}^{n} x_{ik}^2 d_k\right|}$$

Errors on diagonal entries of almost diagonal RRDs (I)

Given $X \in \mathbb{R}^{n\times n}$ nonsingular and $D = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ diagonal and nonsingular:

- Assume that $A = XDX^T$ satisfies $\dfrac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} = O(\epsilon)$ for all $i > j$.
- $a_{ii} = \displaystyle\sum_{k=1}^{n} x_{ik}^2 d_k$
-

$$\left| \frac{\mathtt{fl}(a_{ii}) - a_{ii}}{a_{ii}} \right| \leq \frac{(n+1)\epsilon}{1-(n+1)\epsilon} \frac{\displaystyle\sum_{k=1}^{n} x_{ik}^2 |d_k|}{\left|\displaystyle\sum_{k=1}^{n} x_{ik}^2 d_k\right|}$$

Errors on diagonal entries of almost diagonal RRDs (I)

Given $X \in \mathbb{R}^{n\times n}$ nonsingular and $D = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ diagonal and nonsingular:

- Assume that $A = XDX^T$ satisfies $\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} = O(\epsilon)$ for all $i > j$.
- $a_{ii} = \sum_{k=1}^{n} x_{ik}^2 d_k$
-

$$\left|\frac{\mathtt{fl}(a_{ii}) - a_{ii}}{a_{ii}}\right| \leq \frac{(n+1)\epsilon}{1-(n+1)\epsilon} \, \frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{\left|\sum_{k=1}^{n} x_{ik}^2 d_k\right|}$$

Errors on diagonal entries of almost diagonal RRDs (I)

Given $X \in \mathbb{R}^{n\times n}$ nonsingular and $D = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ diagonal and nonsingular:

- Assume that $A = XDX^T$ satisfies $\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} = O(\epsilon)$ for all $i > j$.
- $a_{ii} = \sum_{k=1}^{n} x_{ik}^2 d_k$
- $$\left| \frac{\mathtt{fl}(a_{ii}) - a_{ii}}{a_{ii}} \right| \leq \frac{(n+1)\epsilon}{1-(n+1)\epsilon} \, \frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{\left| \sum_{k=1}^{n} x_{ik}^2 d_k \right|}$$

Errors on diagonal entries of almost diagonal RRDs (II): EXAMPLE

INPUT: $\kappa(X) = 7.21$

$$XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

RUNNING IMPLICIT JACOBI UNTIL CONVERGENCE

$$\begin{aligned} X_f D X_f^T &= \begin{bmatrix} 4.79 \cdot 10^{-48} & 5.35 \cdot 10^{-1} & 2.04 \cdot 10^{-47} \\ 3.8 \cdot 10^{-1} & 4.03 \cdot 10^{-2} & 1.64 \\ 2.42 & 1.65 & 5.67 \cdot 10^{-1} \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X_f^T \\ &= \begin{bmatrix} 2.86 \cdot 10^{-1} & -3.16 \cdot 10^{3} & 2.39 \cdot 10^{-3} \\ -3.16 \cdot 10^{3} & -2.53 \cdot 10^{50} & 1.04 \cdot 10^{34} \\ 2.39 \cdot 10^{-3} & 2.08 \cdot 10^{34} & 5.53 \cdot 10^{50} \end{bmatrix} \end{aligned}$$

THERE IS NO CANCELLATION

$$\begin{aligned} 2.86 \cdot 10^{-1} &= (4.79 \cdot 10^{-48})^2 \times 10^{50} + (5.35 \cdot 10^{-1})^2 \times 1 + (2.04 \cdot 10^{-47})^2 \times (-10^{50}) \\ &= 2.29 \cdot 10^{-45} + 2.86 \cdot 10^{-1} - 4.18 \cdot 10^{-44} \end{aligned}$$

Errors on diagonal entries of almost diagonal RRDs (II): EXAMPLE

INPUT: $\kappa(X) = 7.21$

$$XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

RUNNING IMPLICIT JACOBI UNTIL CONVERGENCE

$$\begin{aligned} X_f D X_f^T &= \begin{bmatrix} 4.79 \cdot 10^{-48} & 5.35 \cdot 10^{-1} & 2.04 \cdot 10^{-47} \\ 3.8 \cdot 10^{-1} & 4.03 \cdot 10^{-2} & 1.64 \\ 2.42 & 1.65 & 5.67 \cdot 10^{-1} \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X_f^T \\ &= \begin{bmatrix} 2.86 \cdot 10^{-1} & -3.16 \cdot 10^{3} & 2.39 \cdot 10^{-3} \\ -3.16 \cdot 10^{3} & -2.53 \cdot 10^{50} & 1.04 \cdot 10^{34} \\ 2.39 \cdot 10^{-3} & 2.08 \cdot 10^{34} & 5.53 \cdot 10^{50} \end{bmatrix} \end{aligned}$$

THERE IS NO CANCELLATION

$$\begin{aligned} 2.86 \cdot 10^{-1} &= (4.79 \cdot 10^{-48})^2 \times 10^{50} + (5.35 \cdot 10^{-1})^2 \times 1 + (2.04 \cdot 10^{-47})^2 \times (-10^{50}) \\ &= 2.29 \cdot 10^{-45} + 2.86 \cdot 10^{-1} - 4.18 \cdot 10^{-44} \end{aligned}$$

Errors on diagonal entries of almost diagonal RRDs (II): EXAMPLE

INPUT: $\kappa(X) = 7.21$

$$XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

RUNNING IMPLICIT JACOBI UNTIL CONVERGENCE

$$\begin{aligned} X_f D X_f^T &= \begin{bmatrix} 4.79 \cdot 10^{-48} & 5.35 \cdot 10^{-1} & 2.04 \cdot 10^{-47} \\ 3.8 \cdot 10^{-1} & 4.03 \cdot 10^{-2} & 1.64 \\ 2.42 & 1.65 & 5.67 \cdot 10^{-1} \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X_f^T \\ &= \begin{bmatrix} 2.86 \cdot 10^{-1} & -3.16 \cdot 10^{3} & 2.39 \cdot 10^{-3} \\ -3.16 \cdot 10^{3} & -2.53 \cdot 10^{50} & 1.04 \cdot 10^{34} \\ 2.39 \cdot 10^{-3} & 2.08 \cdot 10^{34} & 5.53 \cdot 10^{50} \end{bmatrix} \end{aligned}$$

THERE IS NO CANCELLATION

$$\begin{aligned} 2.86 \cdot 10^{-1} &= (4.79 \cdot 10^{-48})^2 \times 10^{50} + (5.35 \cdot 10^{-1})^2 \times 1 + (2.04 \cdot 10^{-47})^2 \times (-10^{50}) \\ &= 2.29 \cdot 10^{-45} + 2.86 \cdot 10^{-1} - 4.18 \cdot 10^{-44} \end{aligned}$$

Errors on diagonal entries of almost diagonal RRDs (II): EXAMPLE

INPUT: $\kappa(X) = 7.21$

$$XDX^T = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X^T$$

RUNNING IMPLICIT JACOBI UNTIL CONVERGENCE

$$\begin{aligned} X_f D X_f^T &= \begin{bmatrix} 4.79 \cdot 10^{-48} & 5.35 \cdot 10^{-1} & 2.04 \cdot 10^{-47} \\ 3.8 \cdot 10^{-1} & 4.03 \cdot 10^{-2} & 1.64 \\ 2.42 & 1.65 & 5.67 \cdot 10^{-1} \end{bmatrix} \begin{bmatrix} 10^{50} & & \\ & 1 & \\ & & -10^{50} \end{bmatrix} X_f^T \\ &= \begin{bmatrix} 2.86 \cdot 10^{-1} & -3.16 \cdot 10^{3} & 2.39 \cdot 10^{-3} \\ -3.16 \cdot 10^{3} & -2.53 \cdot 10^{50} & 1.04 \cdot 10^{34} \\ 2.39 \cdot 10^{-3} & 2.08 \cdot 10^{34} & 5.53 \cdot 10^{50} \end{bmatrix} \end{aligned}$$

THERE IS NO CANCELLATION

$$\begin{aligned} 2.86 \cdot 10^{-1} &= (4.79 \cdot 10^{-48})^2 \times 10^{50} + (5.35 \cdot 10^{-1})^2 \times 1 + (2.04 \cdot 10^{-47})^2 \times (-10^{50}) \\ &= 2.29 \cdot 10^{-45} + 2.86 \cdot 10^{-1} - 4.18 \cdot 10^{-44} \end{aligned}$$

Errors on diagonal entries of almost diagonal RRDs (III): THE MAIN THEOREM

Theorem

Let $X, D \in \mathbb{R}^{n\times n}$ *be nonsingular and* $D = \mathrm{diag}(d_1, \ldots, d_n)$ *be diagonal. If the matrix* $A \equiv XDX^T$ *satisfies* $a_{ii} = \sum_{k=1}^n x_{ik}^2 d_k \neq 0$ *for all* i, *and*

$$\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \delta, \quad \text{for all } \ i \neq j, \quad \text{where } \delta \leq \frac{1}{5n}, \text{ then}$$

$$\frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \frac{\kappa(X)}{1-2n\delta}\left(1 + \frac{2\,n^{5/2}\delta}{1-n\delta} + n^2\left(\frac{n\delta}{1-n\delta}\right)^2\right), \qquad i = 1, \ldots, n.$$

$$\frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \kappa(X)\left(1 + O(n^{5/2}\delta)\right), \qquad i = 1, \ldots, n.$$

Errors on diagonal entries of almost diagonal RRDs (III): THE MAIN THEOREM

Theorem

Let $X, D \in \mathbb{R}^{n\times n}$ *be nonsingular and* $D = \mathrm{diag}(d_1, \ldots, d_n)$ *be diagonal. If the matrix* $A \equiv XDX^T$ *satisfies* $a_{ii} = \sum_{k=1}^{n} x_{ik}^2 d_k \neq 0$ *for all* i*, and*

$$\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \delta, \quad \text{for all } i \neq j, \quad \text{where } \delta \leq \tfrac{1}{5n}, \text{ then}$$

$$\frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \frac{\kappa(X)}{1-2n\delta}\left(1 + \frac{2\,n^{5/2}\delta}{1-n\delta} + n^2\left(\frac{n\delta}{1-n\delta}\right)^2\right), \qquad i = 1, \ldots, n.$$

$$\frac{\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \kappa(X)\left(1 + O(n^{5/2}\delta)\right), \qquad i = 1, \ldots, n.$$

Errors on diagonal entries of almost diagonal RRDs (III): THE MAIN THEOREM

Theorem

Let $X, D \in \mathbb{R}^{n\times n}$ *be nonsingular and* $D = \mathrm{diag}(d_1, \ldots, d_n)$ *be diagonal. If the matrix* $A \equiv XDX^T$ *satisfies* $a_{ii} = \sum_{k=1}^{n} x_{ik}^2 d_k \neq 0$ *for all* i, *and*

$$\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq \delta, \quad \textit{for all } \; i \neq j, \quad \textit{where } \delta \leq \tfrac{1}{5n}, \textit{ then}$$

$$\frac{\displaystyle\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \frac{\kappa(X)}{1-2n\delta} \left(1 + \frac{2\, n^{5/2}\delta}{1-n\delta} + n^2 \left(\frac{n\delta}{1-n\delta} \right)^2 \right), \qquad i = 1, \ldots, n.$$

$$\frac{\displaystyle\sum_{k=1}^{n} x_{ik}^2 |d_k|}{|a_{ii}|} \leq \kappa(X) \left(1 + O(n^{5/2}\delta) \right), \qquad i = 1, \ldots, n.$$

Errors on diagonal entries of almost diagonal RRDs (IV): Corollary

Corollary

If $A = XDX^T$ *satisfies the stopping criterion then*

$$\left| \frac{\texttt{fl}(a_{ii}) - a_{ii}}{a_{ii}} \right| \leq (n+1)\,\epsilon\,\kappa(X) + O(\kappa(X)\,\epsilon^2)$$

Key idea in the proof of THE MAIN THEOREM

Proof by contradiction

- $A = XDX^T$ is close to diagonal, then its diagonal entries are close to its eigenvalues.
- Assume
$$\frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|a_{ii}|} = \frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|\sum_{k=1}^n x_{ik}^2 d_k|} >> \kappa(X)$$
- Then there are perturbations $\widetilde{d}_k = d_k(1+\delta_k)$, $|\delta_k| < \beta << 1$ such that $(X\widetilde{D}X^T)_{ii} = \sum_{k=1}^n x_{ik}^2 \widetilde{d}_k$, satisfy
$$\frac{|a_{ii} - (X\widetilde{D}X^T)_{ii}|}{|a_{ii}|} >> \beta\kappa(X).$$
- This is in contradiction with an RRD determining accurately its eigenvalues.

Key idea in the proof of THE MAIN THEOREM

Proof by contradiction

- $A = XDX^T$ is close to diagonal, then its diagonal entries are close to its eigenvalues.
- Assume
$$\frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|a_{ii}|} = \frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|\sum_{k=1}^n x_{ik}^2 d_k|} >> \kappa(X)$$
- Then there are perturbations $\widetilde{d}_k = d_k(1+\delta_k)$, $|\delta_k| < \beta << 1$ such that $(X\widetilde{D}X^T)_{ii} = \sum_{k=1}^n x_{ik}^2 \widetilde{d}_k$, satisfy
$$\frac{|a_{ii} - (X\widetilde{D}X^T)_{ii}|}{|a_{ii}|} >> \beta\kappa(X).$$
- This is in contradiction with an RRD determining accurately its eigenvalues.

Key idea in the proof of THE MAIN THEOREM

Proof by contradiction

- $A = XDX^T$ is close to diagonal, then its diagonal entries are close to its eigenvalues.
- Assume
$$\frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|a_{ii}|} = \frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|\sum_{k=1}^n x_{ik}^2 d_k|} >> \kappa(X)$$
- Then there are perturbations $\widetilde{d}_k = d_k(1+\delta_k)$, $|\delta_k| < \beta << 1$ such that $(X\widetilde{D}X^T)_{ii} = \sum_{k=1}^n x_{ik}^2 \widetilde{d}_k$, satisfy
$$\frac{|a_{ii} - (X\widetilde{D}X^T)_{ii}|}{|a_{ii}|} >> \beta\kappa(X).$$
- This is in contradiction with an RRD determining accurately its eigenvalues.

Key idea in the proof of THE MAIN THEOREM

Proof by contradiction

- $A = XDX^T$ is close to diagonal, then its diagonal entries are close to its eigenvalues.
- Assume
$$\frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|a_{ii}|} = \frac{\sum_{k=1}^n x_{ik}^2 |d_k|}{|\sum_{k=1}^n x_{ik}^2 d_k|} >> \kappa(X)$$
- Then there are perturbations $\widetilde{d}_k = d_k(1+\delta_k)$, $|\delta_k| < \beta << 1$ such that $(X\widetilde{D}X^T)_{ii} = \sum_{k=1}^n x_{ik}^2 \widetilde{d}_k$, satisfy
$$\frac{|a_{ii} - (X\widetilde{D}X^T)_{ii}|}{|a_{ii}|} >> \beta\kappa(X).$$
- This is in contradiction with an RRD determining accurately its eigenvalues.

Implicit Jacobi is multiplicative backward stable

Theorem

Let N be the **number of rotations** *applied by implicit Jacobi on $A = XDX^T$ until convergence, and $\widehat{\Lambda}$ and $\widehat{U}$ be the computed matrices of eigenvalues and eigenvectors. Then there exists an* **exact orthogonal** *matrix $U \in \mathbb{R}^{n\times n}$ such that*

$$U\widehat{\Lambda}U^T = (I+E)\, XDX^T\, (I+E)^T,$$

with

$$\|E\|_F = O(\epsilon\, N\, \kappa(X)) \quad \text{and} \quad \|\widehat{U} - U\|_F = O(N\,\epsilon).$$

Corollary (Forward errors in e-values)

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon\, N\, \kappa(X)) \qquad \text{for all} \quad i,$$

Implicit Jacobi is multiplicative backward stable

Theorem

Let N be the **number of rotations** *applied by implicit Jacobi on $A = XDX^T$ until convergence, and $\widehat{\Lambda}$ and $\widehat{U}$ be the computed matrices of eigenvalues and eigenvectors. Then there exists an* **exact orthogonal** *matrix $U \in \mathbb{R}^{n\times n}$ such that*

$$U\widehat{\Lambda}U^T = (I+E)\,XDX^T\,(I+E)^T,$$

with

$$\|E\|_F = O(\epsilon\, N\, \kappa(X)) \quad \textit{and} \quad \|\widehat{U} - U\|_F = O(N\,\epsilon).$$

Corollary (Forward errors in e-values)

$$\frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} \leq O(\epsilon\, N\, \kappa(X)) \qquad \textit{for all} \quad i,$$

Technical comments

To establish the backward error result, we need to prove that

- The stopping criterion in finite arithmetic on $A = X_f D X_f^T$ gives *exact* information, i.e.,

$$\mathtt{fl}\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}}\right) \leq \epsilon\,\kappa(X) \Longrightarrow \frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq n\,\epsilon\,\kappa(X) + O(\epsilon^2)$$

  for all $i \neq j$, which is the case if there is no cancellation in $\mathtt{fl}(a_{ii})$.

- The stopping criterion introduces small multiplicative backward errors, i.e.,

$$\mathrm{diag}(\mathtt{fl}(a_{11}), \ldots, \mathtt{fl}(a_{nn})) = (I+F) X_f D X_f^T (I+F)^T,$$

  where $\|F\|_F = O(n^2\,\epsilon\,\kappa(X))$.

Technical comments

To establish the backward error result, we need to prove that

- The stopping criterion in finite arithmetic on $A = X_f D X_f^T$ gives *exact* information, i.e.,

$$\mathtt{fl}\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}}\right) \leq \epsilon\,\kappa(X) \Longrightarrow \frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq n\,\epsilon\,\kappa(X) + O(\epsilon^2)$$

  for all $i \neq j$, which is the case if there is no cancellation in $\mathtt{fl}(a_{ii})$.

- The stopping criterion introduces small multiplicative backward errors, i.e.,

$$\operatorname{diag}(\mathtt{fl}(a_{11}), \ldots, \mathtt{fl}(a_{nn})) = (I+F) X_f D X_f^T (I+F)^T,$$

  where $\|F\|_F = O(n^2\,\epsilon\,\kappa(X))$.

Technical comments

To establish the backward error result, we need to prove that

- The stopping criterion in finite arithmetic on $A = X_f D X_f^T$ gives *exact* information, i.e.,

$$\mathtt{fl}\left(\frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}}\right) \leq \epsilon\,\kappa(X) \Longrightarrow \frac{|a_{ij}|}{\sqrt{|a_{ii}a_{jj}|}} \leq n\,\epsilon\,\kappa(X) + O(\epsilon^2)$$

  for all $i \neq j$, which is the case if there is no cancellation in $\mathtt{fl}(a_{ii})$.

- The stopping criterion introduces small multiplicative backward errors, i.e.,

$$\mathrm{diag}(\mathtt{fl}(a_{11}), \ldots, \mathtt{fl}(a_{nn})) = (I+F) X_f D X_f^T (I+F)^T,$$

  where $\|F\|_F = O(n^2\,\epsilon\,\kappa(X))$.

Outline

Rectangular RRDs

- So far we have considered $A = XDX^T$ with **square and nonsingular** X and D, which excludes singular matrices A.
- If we insist on X being nonsingular, then A is singular if and only if D is singular.
- The zero eigenvalues of A are revealed by the zero diagonal entries of D
- Discarding these entries we get

$$A = XDX^T \in \mathbb{R}^{n \times n} \quad \text{where} \quad X \in \mathbb{R}^{n \times r} \quad D \in \mathbb{R}^{r \times r},$$

with $n > r$, X with full rank, and D nonsingular.

- Implicit Jacobi converges to an $n \times n$ diagonal matrix with zero entries and cancellation is unavoidable.

Rectangular RRDs

- So far we have considered $A = XDX^T$ with **square and nonsingular** X and D, which excludes singular matrices A.
- If we insist on X being nonsingular, then A is singular if and only if D is singular.
- The zero eigenvalues of A are revealed by the zero diagonal entries of D
- Discarding these entries we get

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{where} \quad X \in \mathbb{R}^{n\times r} \quad D \in \mathbb{R}^{r\times r},$$

  with $n > r$, X with full rank, and D nonsingular.
- Implicit Jacobi converges to an $n \times n$ diagonal matrix with zero entries and cancellation is unavoidable.

Rectangular RRDs

- So far we have considered $A = XDX^T$ with **square and nonsingular** X and D, which excludes singular matrices A.
- If we insist on X being nonsingular, then A is singular if and only if D is singular.
- The zero eigenvalues of A are revealed by the zero diagonal entries of D
- Discarding these entries we get

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{where} \quad X \in \mathbb{R}^{n\times r} \quad D \in \mathbb{R}^{r\times r},$$

  with $n > r$, X with full rank, and D nonsingular.
- Implicit Jacobi converges to an $n \times n$ diagonal matrix with zero entries and cancellation is unavoidable.

Rectangular RRDs

- So far we have considered $A = XDX^T$ with **square and nonsingular** X and D, which excludes singular matrices A.
- If we insist on X being nonsingular, then A is singular if and only if D is singular.
- The zero eigenvalues of A are revealed by the zero diagonal entries of D
- Discarding these entries we get

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{where} \quad X \in \mathbb{R}^{n\times r} \quad D \in \mathbb{R}^{r\times r},$$

  with $n > r$, X with full rank, and D nonsingular.
- Implicit Jacobi converges to an $n \times n$ diagonal matrix with zero entries and cancellation is unavoidable.

Rectangular RRDs

- So far we have considered $A = XDX^T$ with **square and nonsingular** X and D, which excludes singular matrices A.
- If we insist on X being nonsingular, then A is singular if and only if D is singular.
- The zero eigenvalues of A are revealed by the zero diagonal entries of D
- Discarding these entries we get

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{where} \quad X \in \mathbb{R}^{n\times r} \quad D \in \mathbb{R}^{r\times r},$$

  with $n > r$, X with full rank, and D nonsingular.
- Implicit Jacobi converges to an $n \times n$ diagonal matrix with zero entries and cancellation is unavoidable.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{with} \quad X \in \mathbb{R}^{n\times r}, \quad D \in \mathbb{R}^{r\times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n\times n}, \quad R \in \mathbb{R}^{r\times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \ldots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{with} \quad X \in \mathbb{R}^{n\times r}, \quad D \in \mathbb{R}^{r\times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n\times n}, \quad R \in \mathbb{R}^{r\times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \ldots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{with} \quad X \in \mathbb{R}^{n\times r}, \quad D \in \mathbb{R}^{r\times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n\times n}, \quad R \in \mathbb{R}^{r\times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \ldots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n \times n} \quad \text{with} \quad X \in \mathbb{R}^{n \times r}, \quad D \in \mathbb{R}^{r \times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n \times n}, \quad R \in \mathbb{R}^{r \times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \ldots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{with} \quad X \in \mathbb{R}^{n\times r}, \quad D \in \mathbb{R}^{r\times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n\times n}, \quad R \in \mathbb{R}^{r\times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \ldots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Algorithm for rectangular RRD $A = XDX^T$

$$A = XDX^T \in \mathbb{R}^{n\times n} \quad \text{with} \quad X \in \mathbb{R}^{n\times r}, \quad D \in \mathbb{R}^{r\times r},$$

1. Compute full QR factorization of X

$$Q \begin{bmatrix} R \\ 0 \end{bmatrix} = X \quad \text{where} \quad Q \in \mathbb{R}^{n\times n}, \;\; R \in \mathbb{R}^{r\times r}$$

2. Note that

$$A = Q \begin{bmatrix} RDR^T & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

3. Apply Implicit Jacobi on RDR^T (**with factors square and nonsingular**) to compute
   1. Nonzero eigenvalues of A: $\lambda_1, \dots, \lambda_r$.
   2. Eigenvector matrix of RDR^T: U_R
4. $[Q(:, 1:r)U_R \mid Q(:, r+1:n)]$ is the eigenvector matrix of A.

Outline

Numerical Experiments

- Thousands of numerical experiments confirm the high relative accuracy of Implicit Jacobi that we have rigorously proven.
- Traditional Jacobi is **slow**, then Implicit Jacobi is **slow**.
- The new Implicit Jacobi is the fastest algorithm with guaranteed errors bounds.
- We only offer one example.

Numerical Experiments

- Thousands of numerical experiments confirm the high relative accuracy of Implicit Jacobi that we have rigorously proven.
- Traditional Jacobi is **slow**, then Implicit Jacobi is **slow**.
- The new Implicit Jacobi is the fastest algorithm with guaranteed errors bounds.
- We only offer one example.

Numerical Experiments

- Thousands of numerical experiments confirm the high relative accuracy of Implicit Jacobi that we have rigorously proven.
- Traditional Jacobi is **slow**, then Implicit Jacobi is **slow**.
- The new Implicit Jacobi is the fastest algorithm with guaranteed errors bounds.
- We only offer one example.

Numerical Experiments

- Thousands of numerical experiments confirm the high relative accuracy of Implicit Jacobi that we have rigorously proven.
- Traditional Jacobi is **slow**, then Implicit Jacobi is **slow**.
- The new Implicit Jacobi is the fastest algorithm with guaranteed errors bounds.
- We only offer one example.

Accurate RRD and Implicit Jacobi in action

EXAMPLE: Symmetric INDEFINITE 100×100 Cauchy matrix A

$$a_{ij} = \frac{1}{x_i + x_j}, \quad \text{with} \quad \begin{cases} x_i = i - 0.5 \quad for \quad i = 1:99 \\ x_{100} = -99.5 \end{cases}$$

- $\kappa(A) = 3.5 \cdot 10^{147}$
- **Errors in RRD and Implicit Jacobi** compared to 200-decimal digits MATLAB's `eig` command

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.2 \cdot 10^{-13} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 5.7 \cdot 10^{-14}.$$

- **Errors in MATLAB's `eig` function**

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.84 \cdot 10^{132} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 1.41.$$

Accurate RRD and Implicit Jacobi in action

EXAMPLE: Symmetric INDEFINITE 100×100 Cauchy matrix A

$$a_{ij} = \frac{1}{x_i + x_j}, \quad \text{with} \quad \begin{cases} x_i = i - 0.5 \quad for \quad i = 1:99 \\ x_{100} = -99.5 \end{cases}$$

- $\kappa(A) = 3.5 \cdot 10^{147}$
- **Errors in RRD and Implicit Jacobi** compared to 200-decimal digits MATLAB's `eig` command

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.2 \cdot 10^{-13} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 5.7 \cdot 10^{-14}.$$

- **Errors in MATLAB's `eig` function**

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.84 \cdot 10^{132} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 1.41.$$

Accurate RRD and Implicit Jacobi in action

EXAMPLE: Symmetric INDEFINITE 100×100 Cauchy matrix A

$$a_{ij} = \frac{1}{x_i + x_j}, \quad \text{with} \quad \begin{cases} x_i = i - 0.5 \quad for \quad i = 1 : 99 \\ x_{100} = -99.5 \end{cases}$$

- $\kappa(A) = 3.5 \cdot 10^{147}$
- **Errors in RRD and Implicit Jacobi** compared to 200-decimal digits MATLAB's `eig` command

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.2 \cdot 10^{-13} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 5.7 \cdot 10^{-14}.$$

- **Errors in MATLAB's `eig` function**

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.84 \cdot 10^{132} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 1.41.$$

Accurate RRD and Implicit Jacobi in action

EXAMPLE: Symmetric INDEFINITE 100×100 Cauchy matrix A

$$a_{ij} = \frac{1}{x_i + x_j}, \quad \text{with} \quad \begin{cases} x_i = i - 0.5 \quad for \quad i = 1:99 \\ x_{100} = -99.5 \end{cases}$$

- $\kappa(A) = 3.5 \cdot 10^{147}$
- **Errors in RRD and Implicit Jacobi** compared to 200-decimal digits MATLAB's `eig` command

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.2 \cdot 10^{-13} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 5.7 \cdot 10^{-14}.$$

- **Errors in MATLAB's `eig` function**

$$\max_i \frac{|\hat{\lambda}_i - \lambda_i|}{|\lambda_i|} = 1.84 \cdot 10^{132} \quad \text{and} \quad \max_i \|\hat{v}_i - v_i\|_2 = 1.41.$$

Outline

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.

Conclusions

- RRDs together with **Implicit Jacobi algorithms** are the standard way to compute accurate eigenvalues of structured symmetric matrices.
- To compute an accurate rank revealing decomposition (RRD) is essential to get accurate eigenvalues. It is a nontrivial task.
- The new implicit Jacobi algorithm on symmetric RRDs $A = XDX^T$ is the first algorithm that:
  1. computes accurate e-values and e-vectors of A,
  2. preserves the symmetry, and uses only orthogonal transformations.
- The error bounds are the **best possible ones** from the sensitivity of the problem.
- The implicit Jacobi algorithm is a **very simple** extension of standard Jacobi.
- The implicit Jacobi algorithm is **backward stable** in a strong **multiplicative** sense.