Hamiltonian Jacobi methods: sweeping away convergence problems

Christian Mehl
School of Mathematics
University of Birmingham
United Kingdom

GAMM Workshop Applied and Numerical Linear Algebra
TU Hamburg-Harburg, September 11-12, 2008

Jacobi algorithms

$$\begin{bmatrix} * & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & * & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

Jacobi's algorithm for symmetric matrices:

- select a pivot element in the lower triangular part;

Jacobi algorithms

$$\begin{bmatrix} \ast & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \ast & \cdot & \cdot & \circ & \cdot \\ \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ast & \cdot & \cdot \\ \cdot & \circ & \cdot & \cdot & \ast & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \ast \end{bmatrix}$$

Jacobi's algorithm for symmetric matrices:

- select a pivot element in the lower triangular part;
- diagonalize a corresponding 2×2-problem with a Jacobi-rotation;

Jacobi algorithms

$$\begin{bmatrix} \ast & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \ast & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ast & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \ast & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \ast \end{bmatrix}$$

Jacobi's algorithm for symmetric matrices:

- select a pivot element in the lower triangular part;
- diagonalize a corresponding 2×2-problem with a Jacobi-rotation;
- cyclic version: do repeatedly $\frac{n(n-1)}{2}$ steps, where each off-diagonal element is annihilated at least once (sweep)

Jacobi algorithms

$$\begin{bmatrix} * & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix} \qquad \mathrm{off}(A) := \sqrt{\sum_{i \neq j} |a_{ij}|^2}$$

Jacobi's algorithm for symmetric matrices / **properties**:

- $\mathrm{off}(A)$ decreases monotonically;
- globally convergent (...);
- asymptotically quadratically convergent (...)

Nonsymmetric Jacobi algorithms

There are Jacobi-like algorithms for other types of eigenvalue problems:

- computing the Schur form for complex matrices (Greenstadt, 1955, Eberlein, 1962/87, Stewart 1985)
- diagonalization of normal matrices (Goldstein, Hurwitz, 1959)
- Hamiltonian matrices (Byers, 1986, Bunse-Gerstner Faßbender, 1997)
- Generalized Schur form of regular pencils (Charlier, Van Dooren, 1989)
- doubly structured matrices (Faßbender, Mackey, Mackey, 2001)
- ... many more ... Drmač, Hari, Veselić, ...

Nonsymmetric Jacobi algorithms

$$\begin{bmatrix} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & * & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

Nonsymmetric Jacobi algorithm for general complex matrices:

- select a pivot element in the lower triangular part;

Nonsymmetric Jacobi algorithms

$$\begin{bmatrix} \ast & \ast & \ast & \ast & \ast & \ast \\ \cdot & \ast & \ast & \ast & \ast & \ast \\ \cdot & \cdot & \ast & \ast & \ast & \ast \\ \cdot & \cdot & \cdot & \ast & \ast & \ast \\ \cdot & \circ & \cdot & \cdot & \ast & \ast \\ \cdot & \cdot & \cdot & \cdot & \cdot & \ast \end{bmatrix}$$

Nonsymmetric Jacobi algorithm for general matrices:

- select a pivot element in the lower triangular part;
- triangularize a corresponding 2×2-problem with a Jacobi-rotation;

Nonsymmetric Jacobi algorithms

$$\begin{bmatrix} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & \cdot & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

Nonsymmetric Jacobi algorithm for general matrices:

- select a pivot element in the lower triangular part;
- triangularize a corresponding 2×2-problem with a Jacobi-rotation;
- use a cyclic version: do repeatedly $\frac{n(n-1)}{2}$ steps, where each element in the lower triangular part is annihilated at least once (sweep)

Nonsymmetric Jacobi algorithm: properties

$$\begin{bmatrix} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & \cdot & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix} \qquad \text{off}(A) := \sqrt{\sum_{i>j} |a_{ij}|^2}$$

Nonsymmetric Jacobi algorithm for general matrices / **properties**:

- off(A) does NOT decrease monotonically;
- global convergence in experiments (...) – **proof?**;
- asymptotical quadratic convergence in experiments (...) – **but...**;

Hamiltonian Jacobi algorithms

Reminder: the Hamiltonian eigenvalue problem

A matrix $H \in \mathbb{R}^{2n \times 2n}$ is called **Hamiltonian** if

$$H^T J + JH = 0, \qquad \text{where } J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

or, equivalently, if

$$H = \begin{bmatrix} A & C \\ D & -A^T \end{bmatrix},$$

where A, C, D are $n \times n$ and $C = C^T$ and $D = D^T$.

For Jacobi, consider the corresponding complex eigenvalue problem and replace $(\cdot)^T$ with $(\cdot)^*$.

Hamiltonian Jacobi algorithms

Reminder: the Hamiltonian eigenvalue problem

A matrix $S \in \mathbb{R}^{2n\times 2n}$ is called **symplectic** if

$$S^T J S = J, \qquad \text{where } J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}.$$

Similarity transformation with symplectic matrices preserve the Hamiltonian structure.

In the complex case replace $(\cdot)^T$ with $(\cdot)^*$.

Hamiltonian Jacobi algorithms

Reminder: the Hamiltonian eigenvalue problem

Hamiltonian Schur form:

$$H = \begin{bmatrix} R & C \\ 0 & -R^* \end{bmatrix},$$

where $C = C^*$ and R is upper triangular.

If $H \in \mathbb{C}^{2n\times 2n}$ is Hamiltonian, then there exists a unitary symplectic matrix $U \in \mathbb{C}^{2n\times 2n}$ such that U^*HU is in Hamiltonian Schur form if H has no eigenvalues on the imaginary axis.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
consider 4×4 **subproblems** instead of 2×2 subproblems;

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \bullet & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

If a **pivot element** is selected...

Hamiltonian Jacobi algorithms

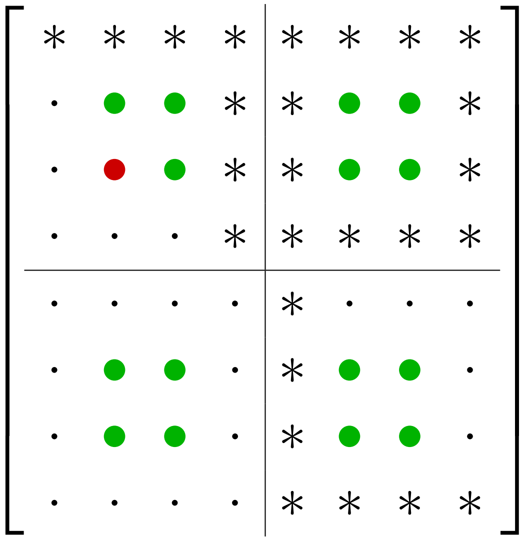
Hamiltonian Jacobi algorithm:
consider 4×4 **subproblems** instead of 2×2 subproblems;

$$\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & \bullet & \bullet & * & * & * & * & * \\
\cdot & \bullet & \bullet & * & * & * & * & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

... then the **corresponding** 2×2 **subproblem** is NOT Hamiltonian if the **pivot element** is off-diagonal.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
consider 4×4 **subproblems** instead of 2×2 subproblems;



$$\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & {\color{green}\bullet} & {\color{green}\bullet} & * & * & {\color{green}\bullet} & {\color{green}\bullet} & * \\
\cdot & {\color{red}\bullet} & {\color{green}\bullet} & * & * & {\color{green}\bullet} & {\color{green}\bullet} & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & {\color{green}\bullet} & {\color{green}\bullet} & \cdot & * & {\color{green}\bullet} & {\color{green}\bullet} & \cdot \\
\cdot & {\color{green}\bullet} & {\color{green}\bullet} & \cdot & * & {\color{green}\bullet} & {\color{green}\bullet} & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

The **corresponding** 4×4 **subproblem** is the smallest Hamiltonian subproblem containing the off-diagonal **pivot element**.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
consider 4×4 **subproblems** instead of 2×2 subproblems;

$$\left[\begin{array}{cccc|cccc} * & * & * & * & * & * & * & * \\ \cdot & {\color{green}\bullet} & {\color{green}\bullet} & * & * & {\color{green}\bullet} & {\color{green}\bullet} & * \\ \cdot & {\color{red}\circ} & {\color{green}\bullet} & * & * & {\color{green}\bullet} & {\color{green}\bullet} & * \\ \cdot & \cdot & \cdot & * & * & * & * & * \\ \hline \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & {\color{green}\circ} & {\color{green}\circ} & \cdot & * & {\color{green}\bullet} & {\color{green}\circ} & \cdot \\ \cdot & {\color{green}\circ} & {\color{green}\circ} & \cdot & * & {\color{green}\bullet} & {\color{green}\bullet} & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & * \end{array}\right]$$

Compute the Hamiltonian Schur form of the 4×4 **subproblem** and transform the matrix accordingly. $\rightsquigarrow$ The **pivot element** is annihilated.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
\bullet & \bullet & * & * & \bullet & \bullet & * & * \\
\circ & \bullet & * & * & \bullet & \bullet & * & * \\
\cdot & \cdot & * & * & * & * & * & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\circ & \circ & \cdot & \cdot & \bullet & \circ & \cdot & \cdot \\
\circ & \circ & \cdot & \cdot & \bullet & \bullet & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
\bullet & * & \bullet & * & \bullet & * & \bullet & * \\
\cdot & * & * & * & * & * & * & * \\
\circ & * & \bullet & * & \bullet & * & \bullet & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\circ & \cdot & \circ & \cdot & \bullet & \cdot & \circ & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
\circ & \cdot & \circ & \cdot & \bullet & \cdot & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
\bullet & * & * & \bullet & \bullet & * & * & \bullet \\
\cdot & * & * & * & * & * & * & * \\
\cdot & \cdot & * & * & * & * & * & * \\
\circ & \cdot & \cdot & \bullet & \bullet & * & * & \bullet \\
\hline
\circ & \cdot & \cdot & \circ & \bullet & \cdot & \cdot & \circ \\
\cdot & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & \cdot \\
\circ & \cdot & \cdot & \circ & \bullet & \cdot & \cdot & \bullet
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & \bullet & \bullet & * & * & \bullet & \bullet & * \\
\cdot & \circ & \bullet & * & * & \bullet & \bullet & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & \circ & \circ & \cdot & * & \bullet & \circ & \cdot \\
\cdot & \circ & \circ & \cdot & * & \bullet & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \bullet & \ast & \bullet & \ast & \bullet & \ast & \bullet \\
\cdot & \cdot & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \circ & \cdot & \bullet & \ast & \bullet & \ast & \bullet \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \circ & \cdot & \circ & \ast & \bullet & \cdot & \circ \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
\cdot & \circ & \cdot & \circ & \ast & \bullet & \cdot & \bullet
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
sweep: annihilate each pivot element at least once;

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \cdot & \bullet & \bullet & \ast & \ast & \bullet & \bullet \\
\cdot & \cdot & \circ & \bullet & \ast & \ast & \bullet & \bullet \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \cdot & \cdot \\
\cdot & \cdot & \circ & \circ & \ast & \ast & \bullet & \circ \\
\cdot & \cdot & \circ & \circ & \ast & \ast & \bullet & \bullet
\end{array}\right]$$

typical row-by-column sweep

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
potential problem: even if the Hamiltonian Schur form exists for the large matrix, it need not exist for the 4×4 subproblem;

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \bullet & \bullet & \ast & \ast & \bullet & \bullet & \ast \\
\cdot & \bullet & \bullet & \ast & \ast & \bullet & \bullet & \ast \\
\cdot & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \bullet & \bullet & \cdot & \ast & \bullet & \bullet & \cdot \\
\cdot & \bullet & \bullet & \cdot & \ast & \bullet & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

Problem: the subproblem may have purely imaginary eigenvalues even if the large matrix does not have.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
potential problem: even if the Hamiltonian Schur form exists for the large matrix, it need not exist for the 4×4 subproblem;

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & \bullet & \bullet & \ast & \ast & \bullet & \bullet & \ast \\
\cdot & \circ & \bullet & \ast & \ast & \bullet & \bullet & \ast \\
\cdot & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \circ & \circ & \cdot & \ast & \bullet & \circ & \cdot \\
\cdot & \circ & \bullet & \cdot & \ast & \bullet & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

way out: if the subproblem has only two purely imaginary eigenvalues, compute a partial Hamiltonian Schur form.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm:
potential problem: even if the Hamiltonian Schur form exists for the large matrix, it need not exist for the 4×4 subproblem;

$$\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & \bullet & \bullet & * & * & \bullet & \bullet & * \\
\cdot & \bullet & \bullet & * & * & \bullet & \bullet & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & \bullet & \bullet & \cdot & * & \bullet & \bullet & \cdot \\
\cdot & \bullet & \bullet & \cdot & * & \bullet & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

way out: if the subproblem has four purely imaginary eigenvalues, proceed without doing anything and hope this does not affect convergence.

Hamiltonian Jacobi algorithms

Hamiltonian Jacobi algorithm: observation in numerical experiments:

- the algorithm appears to be globally convergent for Hamiltonian matrices with no purely imaginary eigenvalues;
- in the beginning (sorting phase) a few subproblems with purely imaginary eigenvalues may occur, but this does not affect convergence of the algorithm;
- in the end (convergence phase), when the matrix is already close to Hamiltonian Schur form, no subproblems with purely imaginary eigenvalues occur;
- presence of subproblems with purely imaginary eigenvalues does not affect the asymptotic convergence behaviour of the algorithm;
- the asymptotic convergence rate is **only linear**!!

Hamiltonian Jacobi algorithms

Why???

For an explanation, let's revisit the nonsymmetric Jacobi algorithm for general complex matrices.

Surprise: sweep $\neq$ sweep

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \circ & \bullet & \bullet & \bullet & \bullet \\ \circ & \circ & \circ & \bullet & \bullet & \bullet \\ \circ & \circ & \circ & \circ & \bullet & \bullet \\ \circ & \circ & \circ & \circ & \circ & \bullet \end{bmatrix}$$

Consider two “cyclic-by-column” sweeps:

- i) “top-to-bottom”: $(2,1),\ldots,(n,1),(3,2),\ldots,(n,2),\ldots,(n,n-1)$
- ii) “bottom-to-top”: $(n,1),\ldots,(2,1),(n,2),\ldots,(3,2),\ldots,(n,n-1)$

symmetric case:
quadratic asymptotic convergence for both sweep selections;

Surprise: sweep $\neq$ sweep

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \circ & \bullet & \bullet & \bullet & \bullet \\ \circ & \circ & \circ & \bullet & \bullet & \bullet \\ \circ & \circ & \circ & \circ & \bullet & \bullet \\ \circ & \circ & \circ & \circ & \circ & \bullet \end{bmatrix}$$

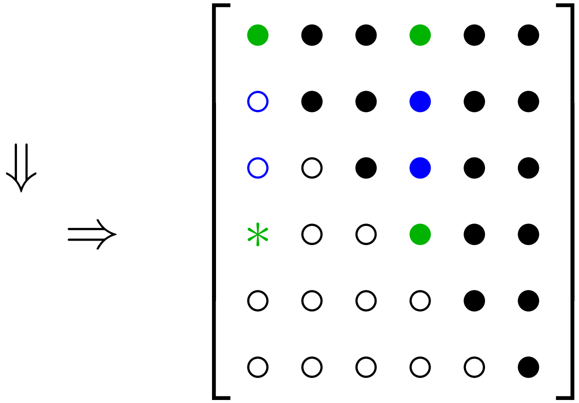
Consider two “cyclic-by-column” sweeps:

i) “top-to-bottom”: $(2,1),\ldots,(n,1),(3,2),\ldots,(n,2),\ldots,(n,n-1)$

ii) “bottom-to-top”: $(n,1),\ldots,(2,1),(n,2),\ldots,(3,2),\ldots,(n,n-1)$

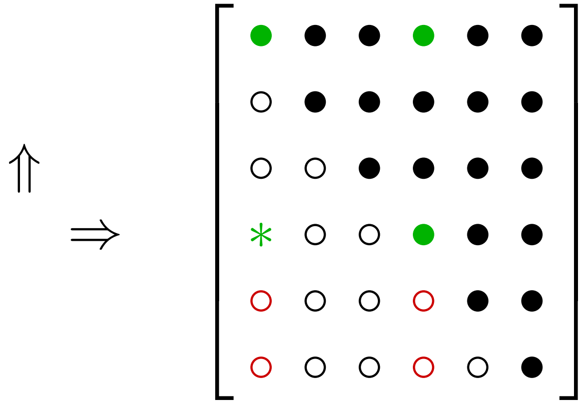
nonsymmetric case:
quadratic asymptotic convergence for “bottom-to-top”-sweeps,
but only linear asymptotic convergence for “top-to-bottom”-sweeps;

Surprise: sweep $\neq$ sweep



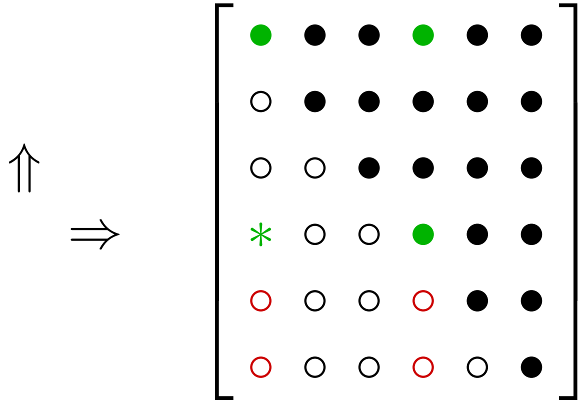
Heuristic explanation: in "top-to-bottom"-sweeps **elements** that have already been annihilated in the current sweep are recombined with potentially large elements from the upper triangular part.

Surprise: sweep $\neq$ sweep



Heuristic explanation: in “bottom-to-top”-sweeps such **elements** are recombined with usually small elements from the lower triangular part.

Surprise: sweep $\neq$ sweep

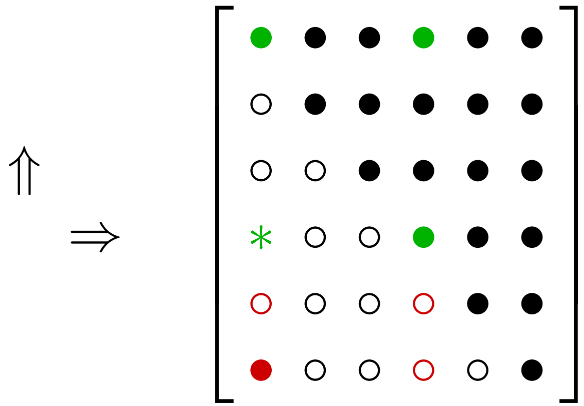


Theorem (M., 2008): The nonsymmetric Jacobi algorithm is asympotically quadratically convergent **if northeast directed sweeps are used**.

Northeast directed sweep: the sequence of indices $((i_1, j_1), ..., (i_N, j_N))$, $N = n(n-1)/2$ in which order the elements are annihilated satisfies

$$\nu < \mu \quad \Rightarrow \quad (i_\nu > i_\mu \text{ or } j_\nu < j_\mu)$$

Surprise: sweep $\neq$ sweep



Examples for northeast directed sweeps:

- “bottom-to-top”: $(n,1),\ldots,(2,1),(n,2),\ldots,(3,2),\ldots,(n,n-1)$;
- “east-and-up”: $(n,1),\ldots,(n,n{-}1),(n{-}1,1),\ldots,(n{-}1,n{-}2),\ldots,(2,1)$;
- “out-to-in”: $(n,1),(n-1,1),(n,2),(n-2,1),(n,3),\ldots,(2,1),(n,n-1),(n-1,2),\ldots$;

You always have to start with the **(n,1)-element**!

Revisiting the Hamiltonian Jacobi algorithm

Observation:

$$H = \begin{bmatrix} R & B \\ 0 & -R^* \end{bmatrix}$$

is in Hamiltonian Schur form if and only if

$$\begin{bmatrix} I_n & 0 \\ 0 & F_n \end{bmatrix} H \begin{bmatrix} I_n & 0 \\ 0 & F_n \end{bmatrix} = \begin{bmatrix} R & BF_n \\ 0 & -F_n R^* F_n \end{bmatrix}$$

is in Schur form. Here

$$F_n = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix}$$

denotes the $n \times n$ ‘flip’ matrix or reverse identity.

Revisiting the Hamiltonian Jacobi algorithm

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
7 & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
6 & \cdot & \ast & \ast & \ast & \ast & \ast & \ast \\
5 & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
1 & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \ast & \ast & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
4 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

If the elements of the first column are annihilated in the depicted order, this corresponds to the first-column-annihilation of the “top-to-bottom” sweep for general complex matrices.

Let’s check the typical Hamiltonian Jacobi sweep!

Revisiting the Hamiltonian Jacobi algorithm

$$\left[\begin{array}{cccc|cccc}
\bullet & \bullet & * & * & \bullet & \bullet & * & * \\
7 & \bullet & * & * & \bullet & \bullet & * & * \\
6 & \cdot & * & * & * & * & * & * \\
5 & \cdot & \cdot & * & * & * & * & * \\
\hline
1 & \circ & \cdot & \cdot & \bullet & \circ & \cdot & \cdot \\
2 & \circ & \cdot & \cdot & \bullet & \bullet & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & * & * & * & \cdot \\
4 & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

Alert: the element “7” is annihilated too early and ...

Revisiting the Hamiltonian Jacobi algorithm

$$\left[\begin{array}{cccc|cccc}
\bullet & * & \bullet & * & \bullet & * & \bullet & * \\
7 & * & * & * & * & * & * & * \\
6 & * & \bullet & * & \bullet & * & \bullet & * \\
5 & \cdot & \cdot & * & * & * & * & * \\
\hline
1 & \cdot & \circ & \cdot & \bullet & \cdot & \circ & \cdot \\
2 & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
3 & \cdot & \circ & \cdot & \bullet & \cdot & \bullet & \cdot \\
4 & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

... and will be messed up in the next step through recombination with **potentially large elements**.

That's where we lose asymptotic quadratic convergence.

sweeping away convergence problems

There are two ways out:

- extend the sweeps of the "4×4 Jacobi method" and annihilate elements twice if necessary;
- invent a "2×2 Jacobi method".

sweeping away convergence problems

Remedy 1: extend the sweeps of the “4×4 Jacobi method”

$$\left[\begin{array}{cccc|cccc}
\bullet & \bullet & \ast & \ast & \bullet & \bullet & \ast & \ast \\
7 & \bullet & \ast & \ast & \bullet & \bullet & \ast & \ast \\
6 & \cdot & \ast & \ast & \ast & \ast & \ast & \ast \\
5 & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
1 & \circ & \cdot & \cdot & \bullet & \circ & \cdot & \cdot \\
2 & \circ & \cdot & \cdot & \bullet & \bullet & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
4 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

Start the sweep as usual ...

1,2 ✓

sweeping away convergence problems

Remedy 1: extend the sweeps of the “4×4 Jacobi method”

$$\left[\begin{array}{cccc|cccc}
\bullet & * & \bullet & * & \bullet & * & \bullet & * \\
7 & * & * & * & * & * & * & * \\
6 & * & \bullet & * & \bullet & * & \bullet & * \\
5 & \cdot & \cdot & * & * & * & * & * \\
\hline
1 & \cdot & \circ & \cdot & \bullet & \cdot & \circ & \cdot \\
2 & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
3 & \cdot & \circ & \cdot & \bullet & \cdot & \bullet & \cdot \\
4 & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

Start the sweep as usual …

3 ✓

sweeping away convergence problems

Remedy 1: extend the sweeps of the "4×4 Jacobi method"

$$\left[\begin{array}{cccc|cccc}
\bullet & * & * & \bullet & \bullet & * & * & \bullet \\
7 & * & * & * & * & * & * & * \\
6 & \cdot & * & * & * & * & * & * \\
5 & \cdot & \cdot & \bullet & \bullet & * & * & \bullet \\
\hline
1 & \cdot & \cdot & \circ & \bullet & \cdot & \cdot & \circ \\
2 & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & * & * & * & \cdot \\
4 & \cdot & \cdot & \circ & \bullet & \cdot & \cdot & \bullet
\end{array}\right]$$

Start the sweep as usual ...

4,5 ✓

sweeping away convergence problems

Remedy 1: extend the sweeps of the "4×4 Jacobi method"

$$\left[\begin{array}{cccc|cccc}
\bullet & \ast & \bullet & \ast & \bullet & \ast & \bullet & \ast \\
7 & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
6 & \ast & \bullet & \ast & \bullet & \ast & \bullet & \ast \\
5 & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
1 & \cdot & \circ & \cdot & \bullet & \cdot & \circ & \cdot \\
2 & \cdot & \cdot & \cdot & \ast & \ast & \cdot & \cdot \\
3 & \cdot & \circ & \cdot & \bullet & \cdot & \bullet & \cdot \\
4 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

... then redo some annihilation to maintain the "top-to-bottom" order.

6 ✓

sweeping away convergence problems

Remedy 1: extend the sweeps of the “4×4 Jacobi method”

$$\left[\begin{array}{cccc|cccc}
\bullet & \bullet & \ast & \ast & \bullet & \bullet & \ast & \ast \\
7 & \bullet & \ast & \ast & \bullet & \bullet & \ast & \ast \\
6 & \cdot & \ast & \ast & \ast & \ast & \ast & \ast \\
5 & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
1 & \circ & \cdot & \cdot & \bullet & \circ & \cdot & \cdot \\
2 & \circ & \cdot & \cdot & \bullet & \bullet & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
4 & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

... then redo some annihilation to maintain the “top-to-bottom” order.

7 ✓ ... and so on ...

sweeping away convergence problems

Remedy 1: extend the sweeps of the "4×4 Jacobi method"

Properties of the extended 4×4 Jacobi method:

– sweep almost twice as expensive as sweep for the original 4×4 Jacobi

+ asymptotic quadratic convergence!!!

sweeping away convergence problems

Remedy 2: invent a "2×2 Jacobi method"

$$\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & {\color{green}\bullet} & * & * & * & {\color{green}\bullet} & * & * \\
\cdot & \cdot & * & * & * & * & * & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & {\color{red}\bullet} & \cdot & \cdot & * & {\color{green}\bullet} & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]$$

If the **pivot element** is on the diagonal of the $(2, 1)$-block, solve a **Hamiltonian 2×2 subproblem** to annihilate it. (If it has purely imaginary eigenvalues, skip).

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”

$$\left[\begin{array}{cccc|cccc}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\cdot & {\color{green}\bullet} & {\color{green}\bullet} & \ast & \ast & \ast & \ast & \ast \\
\cdot & {\color{red}\bullet} & {\color{green}\bullet} & \ast & \ast & \ast & \ast & \ast \\
\cdot & \cdot & \cdot & \ast & \ast & \ast & \ast & \ast \\
\hline
\cdot & \cdot & \cdot & \cdot & \ast & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \cdot \\
\cdot & \cdot & \cdot & \cdot & \ast & \ast & \ast & \ast
\end{array}\right]$$

Otherwise, solve an unstructured 2×2 **subproblem** to annihilate the **pivot element**.

sweeping away convergence problems

Remedy 2: invent a "2×2 Jacobi method"

$$\left[\begin{array}{cccc|cccc} 1 & & & & & & & \\ & \bar{u}_1 & \bar{u}_2 & & & & & \\ & -u_2 & u_1 & & & & & \\ & & & 1 & & & & \\ \hline & & & & 1 & & & \\ & & & & & u_1 & u_2 & \\ & & & & & -\bar{u}_2 & \bar{u}_1 & \\ & & & & & & & 1 \end{array}\right] \left[\begin{array}{cccc|cccc} * & * & * & * & * & * & * & * \\ \cdot & \bullet & \bullet & * & * & * & * & * \\ \cdot & \bullet & \bullet & * & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * & * & * \\ \hline \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & * \end{array}\right] \left[\begin{array}{cccc|cccc} 1 & & & & & & & \\ & u_1 & -\bar{u}_2 & & & & & \\ & u_2 & \bar{u}_1 & & & & & \\ & & & 1 & & & & \\ \hline & & & & 1 & & & \\ & & & & & \bar{u}_1 & -u_2 & \\ & & & & & \bar{u}_2 & u_1 & \\ & & & & & & & 1 \end{array}\right]$$

If unitary symplectic transformation matrices are used ...

sweeping away convergence problems

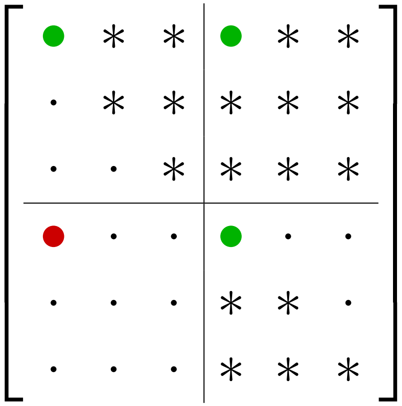
Remedy 2: invent a "2×2 Jacobi method"

$$\left[\begin{array}{cccc|cccc} 1 &&&&&&& \\ & \overline{u}_1 & \overline{u}_2 &&&&& \\ & -u_2 & u_1 &&&&& \\ &&& 1 &&&& \\ \hline &&&& 1 &&& \\ &&&&& u_1 & u_2 & \\ &&&&& -\overline{u}_2 & \overline{u}_1 & \\ &&&&&&& 1 \end{array}\right] \left[\begin{array}{cccc|cccc} * & * & * & * & * & * & * & * \\ \cdot & \color{green}{\bullet} & \color{green}{\bullet} & * & * & * & * & * \\ \cdot & \color{red}{\bullet} & \color{green}{\bullet} & * & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * & * & * \\ \hline \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & \color{green}{+} & \color{red}{\diamond} & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & \color{green}{+} & \color{green}{+} & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & * \end{array}\right] \left[\begin{array}{cccc|cccc} 1 &&&&&&& \\ & u_1 & -\overline{u}_2 &&&&& \\ & u_2 & \overline{u}_1 &&&&& \\ &&& 1 &&&& \\ \hline &&&& 1 &&& \\ &&&&& \overline{u}_1 & -u_2 & \\ &&&&& \overline{u}_2 & u_1 & \\ &&&&&&& 1 \end{array}\right]$$

... then automatically a related 2×2 subproblem will be solved.

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”



A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

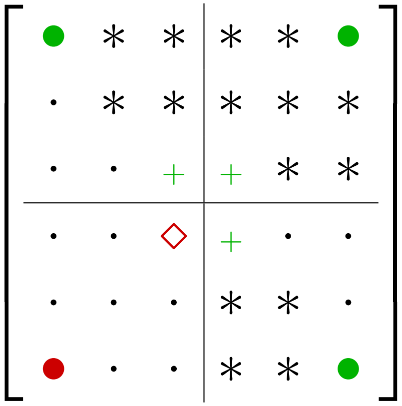
Remedy 2: invent a “2×2 Jacobi method”

$$\left[\begin{array}{ccc|ccc}
{\color{green}\bullet} & \ast & \ast & \ast & {\color{green}\bullet} & \ast \\
\cdot & {\color{green}+} & \ast & {\color{green}+} & \ast & \ast \\
\cdot & \cdot & \ast & \ast & \ast & \ast \\
\hline
\cdot & {\color{red}\diamond} & \cdot & {\color{green}+} & \cdot & \cdot \\
{\color{red}\bullet} & \cdot & \cdot & \ast & {\color{green}\bullet} & \cdot \\
\cdot & \cdot & \cdot & \ast & \ast & \ast
\end{array}\right]$$

A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”



A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

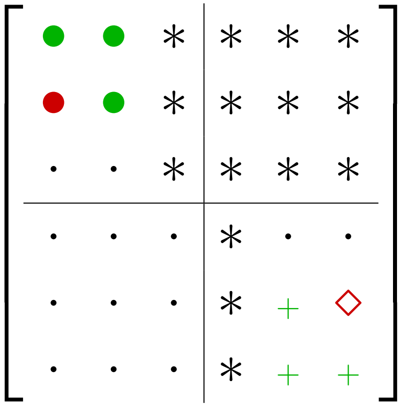
Remedy 2: invent a “2×2 Jacobi method”

$$\left[\begin{array}{ccc|ccc}
{\color{green}\bullet} & * & {\color{green}\bullet} & * & * & * \\
\cdot & * & * & * & * & * \\
{\color{red}\bullet} & \cdot & {\color{green}\bullet} & * & * & * \\
\hline
\cdot & \cdot & \cdot & {\color{green}+} & \cdot & {\color{red}\diamond} \\
\cdot & \cdot & \cdot & * & * & \cdot \\
\cdot & \cdot & \cdot & {\color{green}+} & * & {\color{green}+}
\end{array}\right]$$

A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

Remedy 2: invent a "2×2 Jacobi method"



A typical sweep now corresponds to an "out-to-in" sweep for general complex matrices.

sweeping away convergence problems

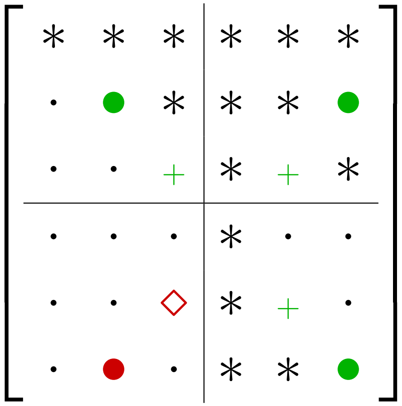
Remedy 2: invent a “2×2 Jacobi method”

$$\left[\begin{array}{ccc|ccc}
* & * & * & * & * & * \\
\cdot & {\color{green}\bullet} & * & * & {\color{green}\bullet} & * \\
\cdot & \cdot & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & * & \cdot & \cdot \\
\cdot & {\color{red}\bullet} & \cdot & * & {\color{green}\bullet} & \cdot \\
\cdot & \cdot & \cdot & * & * & *
\end{array}\right]$$

A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”



A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

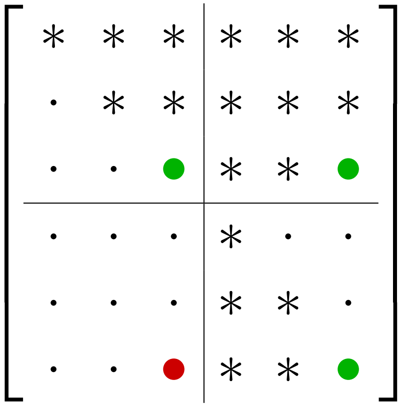
Remedy 2: invent a "2×2 Jacobi method"

$$\left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ \cdot & \textcolor{green}{\bullet} & \textcolor{green}{\bullet} & * & * & * \\ \cdot & \textcolor{red}{\bullet} & \textcolor{green}{\bullet} & * & * & * \\ \hline \cdot & \cdot & \cdot & \textcolor{green}{+} & \textcolor{red}{\diamond} & \cdot \\ \cdot & \cdot & \cdot & \textcolor{green}{+} & \textcolor{green}{+} & \cdot \\ \cdot & \cdot & \cdot & * & * & * \end{array}\right]$$

A typical sweep now corresponds to an "out-to-in" sweep for general complex matrices.

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”



$$\left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & {\color{green}\bullet} & * & * & {\color{green}\bullet} \\ \hline \cdot & \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & * & \cdot \\ \cdot & \cdot & {\color{red}\bullet} & * & * & {\color{green}\bullet} \end{array}\right]$$

A typical sweep now corresponds to an “out-to-in” sweep for general complex matrices.

sweeping away convergence problems

Remedy 2: invent a “2×2 Jacobi method”

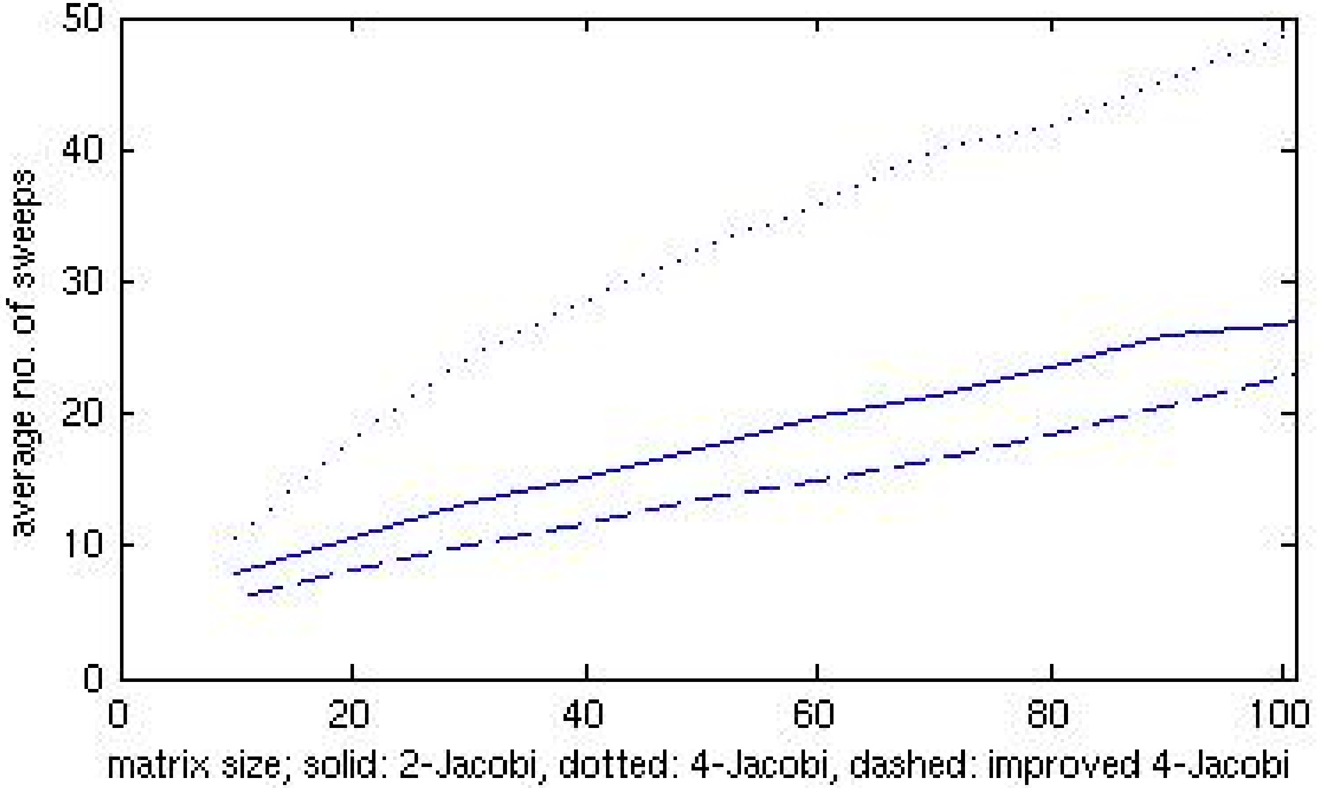
Properties of the 2×2 Jacobi method:

$+$ sweep approximately 80% of the cost of a sweep for the original 4×4 Jacobi (and thus about 40% of the cost of a sweep of the extended 4×4 Jacobi

$+$ asymptotic quadratic convergence (if convergent)!!!

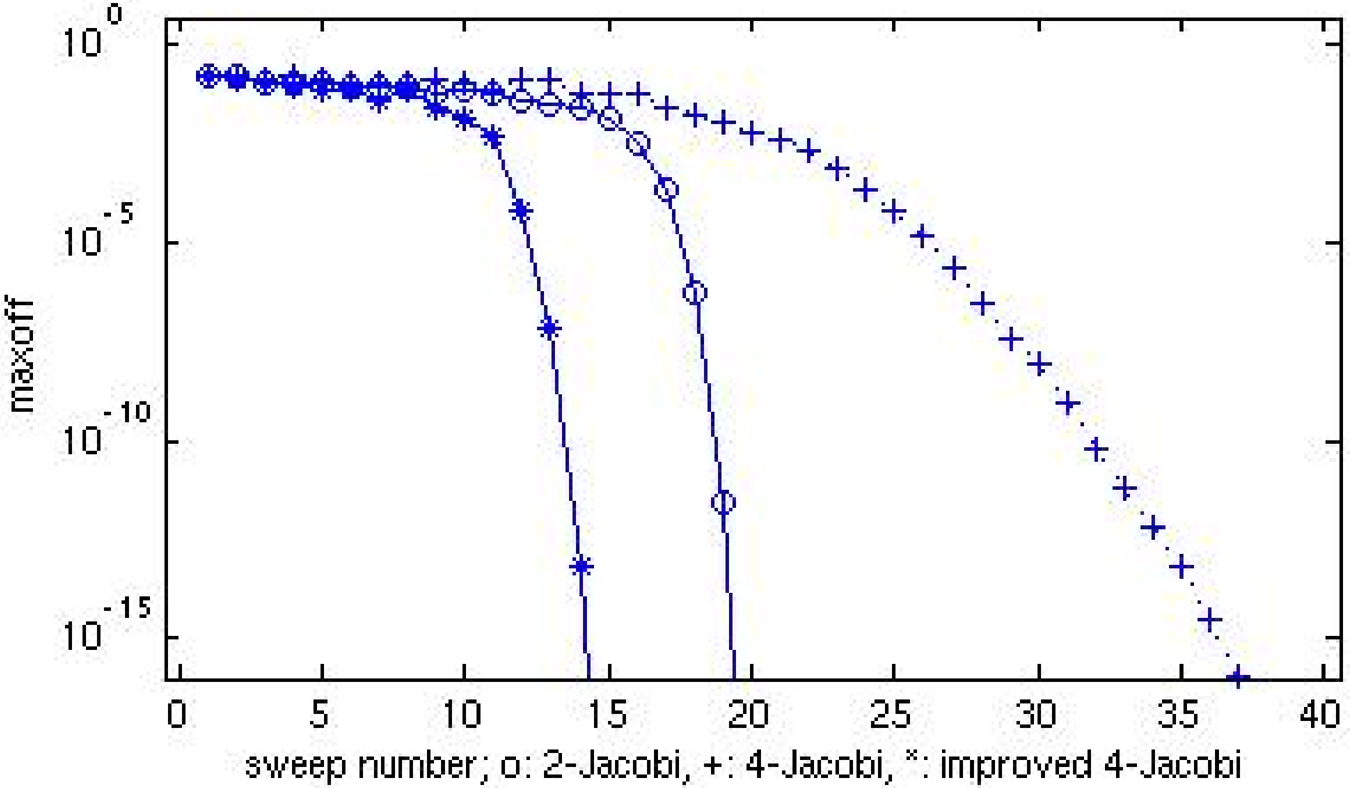
$-$ sometimes stagnation when some Hamiltonian 2×2 subproblems cannot be solved;
Remedy: solve a 4×4 subproblem in this case; this only has to be done a few times and does not affect the asymptotic convergence;

Numerical examples



Test for 100 random Hamiltonian matrices with no purely imaginary eigenvalues.

Numerical examples



Typical convergence behaviour for a 60×60 Hamiltonian
`maxoff`=largest modulus of elements in the part that is to be annihilated.

Thank you for your attention!