

Jacobi-Algorithmen für symmetrische Eigenwertprobleme

Algorithmus III.1 sym.schur2

Input: $\alpha, \beta, \gamma \in \mathbb{R}$

Output: $c = \cos \Theta$, $s = \sin \Theta$, so dass $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \tilde{\alpha} & 0 \\ 0 & \tilde{\gamma} \end{bmatrix}$.

- 1: $\tau = \frac{\alpha - \gamma}{2\beta}$
 - 2: $t = \frac{\text{sign}(\tau)}{|\tau| + \sqrt{1 + \tau^2}}$
 - 3: $c = \frac{1}{\sqrt{1 + t^2}}$, $s = ct$
-

Algorithmus III.2 Klassisches Jacobi-Verfahren

Input: $A = A^T \in \mathbb{R}^{n \times n}$, Toleranz ε .

Output: $A = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) + E$ mit $\|E\|_F \leq \varepsilon$ und $\Lambda(A) \approx \{\tilde{\lambda}_1, \dots, \tilde{\lambda}_n\}$,

$Q \in \mathbb{R}^{n \times n}$ orthogonal mit $Q^T A Q = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) + E$.¹

- 1: $Q \leftarrow I_n$.
 - 2: **while** $\text{off}(A) > \varepsilon \|A\|_F$ **do**
 - 3: Wähle j, k , so dass $|a_{jk}| = \max_{r \neq s} |a_{rs}|$.
 - 4: $[c, s] = \text{sym.schur2}(a_{jj}, a_{jk}, a_{kk})$.
 - 5: $A \leftarrow J(j, k, \Theta)^T A J(j, k, \Theta)$, $Q \leftarrow Q J(j, k, \Theta)$.
 - 6: **end while**
-

Algorithmus III.3 Zyklisches Jacobi-Verfahren

Input: $A = A^T \in \mathbb{R}^{n \times n}$, Toleranz ε .

Output: $A = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) + E$ mit $\|E\|_F \leq \varepsilon$ und $\Lambda(A) \approx \{\tilde{\lambda}_1, \dots, \tilde{\lambda}_n\}$,

$Q \in \mathbb{R}^{n \times n}$ orthogonal mit $Q^T A Q = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) + E$.

- 1: $Q \leftarrow I_n$.
 - 2: **while** $\text{off}(A) > \varepsilon \|A\|_F$ **do**
 - 3: {ein "Sweep"}
 - 4: **for** $j = 1 : n - 1$ **do**
 - 5: **for** $k = j + 1 : n$ **do**
 - 6: $[c, s] = \text{sym.schur2}(a_{jj}, a_{jk}, a_{kk})$.
 - 7: $A \leftarrow J(j, k, \Theta)^T A J(j, k, \Theta)$, $Q \leftarrow Q J(j, k, \Theta)$.
 - 8: **end for**
 - 9: **end for**
 - 10: **end while**
-

¹Teile in blau sind optional.