Otto-von-Guericke-Universität Magdeburg Sommersemester 2020
Max Planck Institute for Dynamics of Complex Technical Systems
Prof. Dr. Peter Benner, Dr. Sara Grundel, Jennifer Przybilla

Numerische Lineare Algebra 2 – Schur-Parlett Algorithm for $f(A)$.

Let $A \in \mathbb{C}^{n \times n}$ and $f(z)$ be sufficiently smooth (e.g., analytic in convex set Ω with $\Lambda(A) \subset \Omega$). The following algorithm computes $f(A)$ using the Schur form of A and the block-Parlett recurrence.

Algorithm 1 Schur-Parlett Algorithm

Input: $A \in \mathbb{C}^{n \times n}$, $f(z)$
Output: $F = f(A)$.

1: Compute, partition, and order Schur form of A:

$$Q^*AQ = R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1q} \\ 0 & R_{22} & \cdots & R_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{qq} \end{bmatrix} \quad \text{such that} \quad \begin{cases} \bullet\ \Lambda(R_{ii}) \cap \Lambda(R_{jj}) = \emptyset \\ \quad \forall i \neq j \in \{1, \dots, q\} \\ \bullet\ |\lambda_i - \lambda_k| < \delta, \\ \quad \lambda_{i,k} \in \Lambda(R_{jj}), \\ \quad j \in \{1, \dots, q\} \end{cases}$$

2: Evaluate f at the diagonal blocks: $F_{ii} = f(R_{ii})$.
3: **for** $i = 1, \dots q$ **do**
4: **for** $j = i-1, i-2, \dots, 1$ **do**
5: Get upper triangular blocks F_{ij} of $F := f(R)$ by solving the Sylvester equations

$$R_{ii}F_{ij} - F_{ij}R_{jj} = F_{ii}R_{ij} - R_{ij}F_{jj} + \sum_{k=i+1}^{j-1} \left(F_{ik}R_{kj} - R_{ik}F_{kj}\right).$$

6: **end for**
7: **end for**
8: Backtransformation: $F \leftarrow QFQ^*$.

Remarks:

- Incorporation of real Schur form for $A \in \mathbb{R}^{n \times n}$ possible, but eigenvalue placement s.t. $|\lambda_i - \lambda_k| < \delta$ difficult.
- Ordering of the Schurform to the required form done by additional unitary transformations. Usually $\delta = 0.1$.
- Functions of *atomic* block $f(R_{ii})$, e.g., by Taylor series.
- Cost approximately $28n^3$. Can be up to $n^4/3$ in worst cases when very large diagonal blocks appear.
- Best general $f(A)$ algorithm. Price to pay: derivatives of f needed if diagonal blocks of size > 1 appear.
- The algorithm is the basis of the MATLAB® routine `funm`.

References

[1] Davies. P, Higham, N.: *A Schur-Parlett Algorithm for Computing Matrix Functions.* SIAM J. Matrix Anal. & Appl., 25(2), 464–485, 2003.

[2] Higham, N.: *Functions of Matrices: Theory and Computation (Chapter 4.6 & 9).* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2008)