problem sheet 13

discussion: week of Monday, 24.1.2022

13.1. (Gerschgorin theorem) Let $\mathbf{A} \in \mathbb{C}^{n\times n}$. Show: for every eigenvalue λ of $\mathbf{A}$ there is an i such that $|\lambda - \mathbf{A}_{ii}| \leq \sum_{j\neq i} |\mathbf{A}_{ij}|$. Put differently: the spectrum of $\mathbf{A}$ is contained in the closed circles with centers $\mathbf{A}_{ii}$ and radii $r_i = \sum_{j\neq i} |\mathbf{A}_{ij}|$. *Hint:* Let $\mathbf{x}$ be eigenvector for λ. Consider i such that $|\mathbf{x}_i| \geq |\mathbf{x}_j|$.

13.2. The computationally expensive part of the QR-algorithm is the QR-factorization in each step (cost: $O(n^3)$). It can be shown (cf. Example 4.53 of the class notes) that for Hessenberg matrices, the cost can be reduced to $O(n^2)$ and in fact that the subsequent multiplication RQ leads to a) a Hessenberg matrix and b) is achieved with cost $O(n^2)$.

Check numerically the cost $O(n^2)$ of the MATLAB (or Python) realization of the QR-factorization. To that end, create for $n = 2^j$, $j = 3, \ldots, 12$, random matrices $A \in \mathbb{R}^{n\times n}$, extract the upper triangular part (in MATLAB, see `trilu`) and add a random subdiagonal to create a random Hessenberg matrix. Check the timing of the MATLAB/Python realization of the QR-factorization. Do you observe $O(n^2)$?

13.3. (to be uploaded in TUWEL) Realize 3 different versions of the QR-algorithm in parts a)-c).

a) Program the basic QR-method (i.e., without shift). Input is the matrix $\mathbf{A}$ and the number ℓ_{max} of QR-steps to be done. Output are the ℓ_{max} diagonals of the matrices of the QR method (i.e., the approximations to the eigenvalues).

b) Program the QR-method with Rayleigh shift and deflation as described in Algorithm 7.39 of the class notes. Input is the matrix $\mathbf{A}$, Output is the list of (approximate) eigenvalues as well as the number of QR steps needed. The shift is defined as the matrix entry $\mathbf{A}(n, n)$. The criterion to determine whether deflation is possible is $|\mathbf{A}(n-1, n)| \leq \varepsilon\,[|\mathbf{A}(n-1, n)| + |\mathbf{A}(n, n)|]$ with $\varepsilon = 10^{-14}$. If deflation is possible, then the algorithm is called recursively with the submatrix $\mathbf{A}([1 : n-1], [1 : n-1])$.
You may use `qr` (to compute the QR factorization) and `hess` to compute the Hessenberg form of a matrix.

c) Program the QR-method with the so-called Wilkinson shift and deflation. Input is the matrix $\mathbf{A}$. Output is the list of (approximate) eigenvalues as well as the number of QR steps needed. This is the same algorithm as in part b) except that the shift parameter, the so-called Wilkinson shift μ is determined as follows: Let λ_1, λ_2 be the 2 eigenvalues of the 2×2 matrix $\mathbf{A}([n-1; n], [n-1 : n])$ and $\mu \in \{\lambda_1, \lambda_2\}$ is the eigenvalue that is closest to $\mathbf{A}(n, n)$. To compute the eigenvalues of the 2×2 matrix, you may use in `matlab` the routine `eig`.

13.4. (continuation of 13.3) Define the tridiagonal matrices

$$\mathbf{A}^{(n)} = a = \texttt{diag}(2 * \texttt{ones}(n, 1), 0) - \texttt{diag}(\texttt{ones}(n-1, 1), -1) - \texttt{diag}(\texttt{ones}(n-1, 1), 1);$$

with $n = 2^j$, $j = 2, \ldots, 10$.

a) Consider the QR-method without shift (i.e., your algorithm of 13.3a)) Plot the maximal error of the eigenvalues versus the iteration number ℓ for $n = 2^3$ and $n = 2^5$. What do you observe? You may use `eig` to compute the exact eigenvalues for the purpose of computing the error.

b) For the above matrices $\mathbf{A}^{(n)}$ with $n = 2^j$, $j \in \{3, 5, 7\}$, plot (use `loglog`) the number of QR-steps of the QR method without shift and the QR method with Wilkinson shift needed versus the problem size n. For the QR method without shift, use 10^{-5} as the error tolerance. (Again, you may use `eig` to compute the exact eigenvalues so as to compute the actually error.) What do you observe?

c) For $n = 2$ (i.e., $\mathbf{A}^{(2)}$ is a 2×2 matrix) apply your Algorithm of 13.3b). Does the algorithm terminate? Explain the behavior by checking by hand one step of the QR algorithm.