Prof. Dr. U. Rüde
Benjamin Mann

Winter Term
2024/2025

Algorithms of Numerical Linear Algebra
Assignment 6

Exercise 1 (*Tridiagonal Matrices*) **4P.**

(a) Let $A \in \mathbb{C}^{m\times m}$ be tridiagonal and Hermitian, with all its sub- and superdiagonal entries nonzero. Prove that A has m distinct eigenvalues.
Hint: Show that $A - \lambda I$ has at least rank $m-1$ and think about the relation between the geometric and the algebraic multiplicity of the eigenvalues.

(b) On the other hand, let A be upper-Hessenberg, with all its subdiagonal as well as strict upper triangular entries nonzero, i.e., $a_{ij} \neq 0$ if $j = i-1$ or $j > i$. Use a counterexample to show that the eigenvalues of A are not necessarily distinct.

Exercise 2 (*Krylov Space*) **4P.**

Let $A \in \mathbb{C}^{m\times m}$ be nonsingular and $b \in \mathbb{C}^m \backslash \{0\}$. Consider the corresponding Krylov subspace $\mathcal{K}_n = \operatorname{span}\{b, Ab, ..., A^{n-1}b\}$ with $n \in \{1, ..., m\}$.

(a) Show that $\dim(\mathcal{K}_n)$ is not necessarily equal to n.

(b) Now consider the system $Ax = b$. Show that $x \in \mathcal{K}_n$ if $\dim(\mathcal{K}_n) < n$.

Exercise 3 (*Eigenvalue Algorithms*) **12P.**

Make sure to follow the requirements for programming tasks stated on the information sheet!

(a) Write a Python3 function `(λ_min, λ_max) = gershgorin(A)` which computes an upper and lower bound for the eigenvalues of a symmetric matrix $A \in \mathbb{R}^{m\times m}$ by applying *Gershgorin's theorem.* The return values of your function should be the minimum and maximum point in the union of all Gershgorin disks, i.e., $[\lambda_{min}, \lambda_{max}] = \operatorname{conv}(\bigcup_i \mathcal{G}_i \cap \mathbb{R})$ where $\mathcal{G}_k$ denotes the k-th Gershgorin disk. Test your implementation, e.g., by applying it to the matrix given in the corresponding task on the previous assignment sheet.

(b) Write a Python3 function `(v,λ,[err]) = power(A, v0)`, which implements a power iteration. The input values are a symmetric matrix $A \in \mathbb{R}^{m\times m}$ and an initial eigenvector guess $v_0 \in \mathbb{R}^m$. The return values λ and v should be approximations to the dominant eigenvalue and corresponding eigenvector, respectively, as detailed in Theorem 27.1 in [1]. Additionally, the function should return a list containing the error $\|Av - \lambda v\|_\infty$ in each iteration. The iteration should terminate when this error is sufficiently small, i.e., $\|Av - \lambda v\|_\infty \leq 10^{-13}$.

(c) Write a Python3 function `(v,λ,[err]) = inverse(A, v0, μ)`, which implements the inverse iteration. The input and return values, as well as the stopping criterion should be the same as in (b). The only difference is the additional eigenvalue estimate μ and that the return values should behave as described in Theorem 27.2 in [1].

(d) Write a Python3 function `(v,`λ`, [err]) = rayleigh(A, v0)`, which implements a Rayleigh Quotient Iteration. Once again, stopping criterion, input and return values are the same as in (b) with the only difference that the return values should behave as in Theorem 27.3 in [1].

(e) Make yourself familiar with the given script `convergence_comparison.py` and run it. Include the resulting plots as well as a detailed discussion in your pdf submission. What conclusions can you draw from the plots? Do they look as expected?

(f) Test your implementations for different input values, e.g. using the function `randomInput(m)`, and discuss your results. Do **not** simply write down the return values!

References

[1] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.