Otto-von-Guericke-Universität Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems
Prof. Dr. Peter Benner, Dr. Patrick Kürschner, Carolin Penke

summer term 2018

Advanced Topics in Numerical Linear Algebra – 6. Exercise

These problems will be discussed in the exercise course of July 5, 2018. Please prepare their solutions before the class, you will be asked to perform one of the questions on the board. You may send your MATLAB® implementations via email to *penke@mpi-magdeburg.mpg.de*.

Problem 1 (Schur-Parlett for Matrix Square Roots)

Let $A \in \mathbb{C}^{n \times n}$ be nonsingular with $\Lambda(A) \cap \mathbb{R}_- = \emptyset$. We look for square roots $F = \sqrt{A}$ of A, i.e., $F^2 = A$. Based on this relation, modify the basic Schur-Parlett algorithm (sheet 5, problem 4) for computing the principal square root of A. Discuss how this can be used to get all primary square roots of A.

Problem 2 (Padé Approximants and Scaling & Squaring for e^A)

Recall the rational approximations $r_m(z) = \frac{p_m(z)}{q_m(z)}$ with

$$p_m(z) = \sum_{j=0}^{m} \frac{(2m-j)!m!}{(2m)!(m-j)!j!} z^j, \quad q_m(z) = p_m(-z)$$

that we used for approximating e^z.

a) Consider the evaluation of $r_m(A) = p_m(A)q_m(A)^{-1}$ at A. Recall from the lecture that we only need $d+1$ matrix-matrix multiplications to evaluate $p_m(A)$, $q_m(A)$ for even degrees $m = 2d$, $d \in \mathbb{N}$.

- **i)** Derive a similar way to evaluate p_m, q_m for odd degrees $m = 2d + 1$ that still requires only $d + 1$ matrix-matrix multiplications (making even degrees essentially redundant).
- **ii)** Find an even cheaper way for degrees $m \geq 12$ (i.e., requiring $\leq d + 1$ matrix-matrix multiplications).

b) Implement a basic scaling & squaring method using a fixed Padé degree $m = 9$ and scaling(squaring) parameter $s \in \mathbb{N}$ s.t. $\|\frac{A}{2^s}\|_1 \leq 1$.

Problem 3 (Arnoldi for $f(A)b$)

Implement an Arnoldi method that approximates $f(A)b$ for large, sparse $A \in \mathbb{C}^{n \times n}$ and $b \in \mathbb{C}^n$. Test it with the data provided on the webpage for $f = \mathrm{e}^z$, $\sin(z)$, $z^{\frac{1}{2}}$, $\log(z)$. Use the MATLAB routine `funm` to evaluate the small, dense matrix functions.

Problem 4 (Randomized Low-rank Matrix Approximations)

Let $A \in \mathbb{R}^{n \times m}$ with $m \leq n$ and $\text{rank}(A) = m$. The goal is to compute an approximation of A which is of low-rank $k \leq m$. Recall how this can be done using the (thin) singular value decomposition (SVD) of A and implement this approach in MATLAB. Now consider and implement the following randomized approach that gives an approximation of rank $2k$:

1. Generate random Gaussian matrix $\Omega \in \mathbb{R}^{m \times 2k}$ (i.e. entries obey standard Gaussian distribution)
2. $Y := AA^T A\Omega$.
3. Compute Q with orthonormal columns s.t. $\text{range}(Q) = \text{range}(Y)$.
4. Compute SVD of $Q^T A$: $USV^T = Q^T A$.
5. Generate low-rank approximation by $\hat{A}_\Omega := (QU)SV^T$.

Run and compare both algorithms with respect to the overall computation time and the approximation quality of the generated results in the spectral norm. Use $A = \texttt{rand}(5000, 500)$ and $k = 10, 50, 100$ as test case.