TU Bergakademie Freiberg
Institut für Numerische Mathematik und Optimierung
Prof. Dr. O. Rheinbach
Dipl.-Math. Matthias Brändel

Wintersemester 2021/22
25.10.2021

Lineare Algebra, Datenanalyse und maschinelles Lernen 2

1. Exercise Sheet

Exercise 38
Use `python` and `numpy` to do the following:

(a) Create two vectors a, b of size 10×1. Fill them with values. Print them.

(b) Compute the scalar product of $a^T b$. Use `numpy` and write a function that uses a `for` loop to solve the problem. Compare both results. What happens if you compute ab^T?

(c) Create two vectors c, d of size $n \times 1$. Fill them with random numbers. Use both the `numpy` function `np.random.rand` for uniformly distributed values and the `python` function `random.gauss` for normally distributed values. Use your scalar product function to compute $c^T d$.

(d) Use `matplotlib.pyplot` to plot the histogram of $[c; d]$. Do you need to use `np.vstack` or `np.hstack`? Adjust the number of bins such that a comprehensive output is shown.

Exercise 39
LU decomposition and solution of a linear system.

(1) Create a tridiagonal matrix A of size $n \times n$ with 2's on the main and -1's on the off-diagonals. Use two approaches: one using `for` loops and one where you utilize the `diags` function of the `scipy` package (`from scipy.sparse import diags`). Compare both results.

(2) Compute the LU decomposition of A inside a function using loops. Compare your result with the LU decomposition you obtain using `scipy.linalg.lu`.

(3) Modify A such that $A[0, :] = A[:, 0] = 0$, $A[0, 0] = 1$ and $A[n, :] = A[:, n] = 0]$, $A[n, n] = 1$. Create a vector $b = 1/(n-1)^2 \cdot [4, 4, \ldots, 4]^T$. Solve the linear system $Ax = b$ using `numpy`, `scipy` and a function of your own, where you utilize the computed LU decomposition and solve the system directly by forward and backward substitution. Compare the results.

(4) Use `matplotlib.pyplot` to plot x.