Department of Mathematics, University of Utah

Analysis of Numerical Methods I

MATH 6610 – Section 001 – Fall 2019

Homework 1

Basic linear algebra and the SVD

Due Monday, September 9, 2019 by 11:59pm MT

Submission instructions:
Create a private repository on `github.com` named `math6610-homework-1`. Add your LaTeX source files and your Matlab/Python code and push to Github. To submit: grant me (username `akilnarayan`) write access to your repository.
You may grant me write access before you complete the assignment. I will not look at your submission until the due date+time specified above. If you choose this route, I will only grade the assignment associated with the last commit before the due date.
All commits timestamped after the due date+time will be ignored.

Problem assignment:
Trefethen & Bau III, Lecture 1: # 1.3
Trefethen & Bau III, Lecture 2: # 2.1, 2.2, 2.3, 2.5, 2.6
Trefethen & Bau III, Lecture 3: # 3.1, 3.3, 3.5
Trefethen & Bau III, Lecture 4: # 4.1 (a-c), 4.4
Trefethen & Bau III, Lecture 5: # 5.1, 5.2, 5.3, 5.4

Programming assignment:

1. (*Computing the SVD*) In either Matlab or Python, explore computational times for computing the SVD for large matrices. Let A be an $m \times n$ matrix, and compile running times for computing the SVD for *both* m and n ranging from small numbers (say, 5) up to large numbers (as large as your computer can handle in a reasonable amount of time). Plot the computational time to compute the SVD of A as a function of m and n. The formal complexity of the SVD is $\mathcal{O}\left(\min(m^2 n, mn^2)\right)$. Do you observe this complexity in your plots?

For both Matlab and Python, you may find the `mesh`, `surf`, and/or `pcolor` visualization tools helpful for this exercise. For Python, these functions are accessible in the `matplotlib.pyplot` module.