Assignment 1

Numerical Analysis II
MATH 494 / MAST 661 / MAST 881 II
Concordia University, Winter 2023

Instructor: Simone Brugiapaglia

Due date: Friday, February 3, 2023, at 6:00 pm

Instructions

- This assignment should be completed in teams:
  - Teams should be composed by 2 to 3 students.
  - As soon as you have formed your team, please send an email to the instructor with the list of students (with ID numbers) composing the team.
  - If you do not have a team, consider (i) posting on the Moodle Discussion board or (ii) write to your instructor and he will make sure to find a team for you.
  - Only one of the team members shall upload the solution to Moodle. The names and student ID of the other team members should be written in the first page.
  - **Communications aimed at sharing solutions between different teams is not allowed** and it is considered a violation of academic integrity. Each team should only communicate internally to solve the proposed problems.
- Each team shall upload their solutions on Moodle as a two separate files:
  1. The first file should be a PDF report containing solutions to problems in Part A and B, in the assigned order (A.1, A.2, ... B.1, B.2, ...). Solutions should be typewritten (preferably, using LaTeX or Word). Also very clear and well-written scanned handwritten solutions are acceptable, although typewritten solutions are preferable.
  2. The second file should contain a .zip file with the **complete source code used to generate solutions** to computational problems (Part B). Make sure to give clear names to files (e.g., `solution_B1.m`), describe your code using comments, and identify what problem is solved by each section of your code. Readability of code will be taken into account upon grading. If you use Python or R, you could also illustrate solutions using notebooks (e.g., Jupyter notebook or R markdown files).

Problems

Acknoweldgement: The proposed problems might contain variations of exercises from the textbooks [Burden & Faires, 2016], [Demmel, 1997], [Golub & Van Loan, 2013], [Quarteroni, 2009], [Quarteroni, Saleri, Gervasio, 2006] and [Trefethen & Bau, 1997] (see the course outline). The copyright of these problems belongs to the authors.

Part A (Theoretical problems)

Problem A.1

After reading Section 1.3 of [Golub & Van Loan, 2006] on block matrices, rigorously prove the following block matrix identity:

$$\begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}^* = \begin{bmatrix} A_{11}^* & \cdots & A_{m1}^* \\ \vdots & \ddots & \vdots \\ A_{1n}^* & \cdots & A_{mn}^* \end{bmatrix} \tag{1}$$

where $A_{ij} \in \mathbb{C}^{p \times q}$.

Problem A.2

Let $v \in \mathbb{C}^m$, $w \in \mathbb{C}^n$ and $A = vw^*$. Show that $\|A\|_F = \|A\|_2 = \|v\|_2 \|w\|_2$ and $\|A\|_\infty \leq \|v\|_\infty \|w\|_1$

Problem A.3

Two matrices $A, B \in \mathbb{C}^{m \times m}$ are *unitarily equivalent* if $A = QBQ^*$ for some unitary $Q \in \mathbb{C}^{m \times m}$. Is it true of false that A and B are unitarily equivalent if and only if they have the same singular values?

Problem A.4

Consider the 2×2 matrix

$$A = \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix}$$

with $z \in \mathbb{C}$. What is the best rank-1 approximation of A with respect to the Frobenius norm?

Problem A.5 (Graduate students only)

Suppose $A \in \mathbb{C}^{n \times n}$. Using the SVD, solve the following problem:

$$\min_{\det(B) = |\det(A)|} \|A - B\|_F.$$

Problem A.6 (Graduate students only)

Suppose the $m \times n$ matrix A has the block form

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where A_1 is a nonsingular matrix of dimension $n \times n$ and A_2 is an arbitrary matrix of dimension $(m-n) \times n$. Prove that $\|A^\dagger\|_2 \leq \|A_1^{-1}\|_2$, where $A^\dagger$ denotes the Moore-Penrose pseudo inverse of A.

Part B (Computational problems)

Problem B.1

1. Load the image `brickwall.jpeg` in Matlab/Octave (or other computing environments), extract only the "R" (Red) part from the (R,G,B) image format and store it into a matrix $A \in \mathbb{R}^{m \times n}$.[1]

2. Compute the SVD of $A = U\Sigma V^*$ and plot U, Σ and V using suitable visualization strategies.

3. Compute and visualize the best rank-q approximation A_q of A with respect to the Frobenius norm for $q = 1, 2, 4, 8, 16, 32$ and compute the corresponding relative approximation errors. Comment on the results.

4. Plot the relative approximation error as a function of q for $q = 1, 2, \ldots, \min(m, n)$ through a suitable visualization strategy. What do you observe?

5. Repeat steps 1-4 using the matrix induced 2-norm. Any difference with the Frobenius case?

6. Repeat steps 1-5 using two images of your choice (be careful with their dimensions, as computing the SVD could be too expensive for very large images). Try to find an example where the low rank approximation is effective, and another example where it isn't. Justify your choices.

Problem B.2

Read Lecture 9 of [Trefethen, Bau, 1997]. Reproduce the results of Experiments 2 and 3. Note that you need to implement your own version of the functions `clgs` (Classical Gram-Schmidt) and `mgs` (modified Gram-Schmidt).

[1] In Matlab/Octave, you can use the `imread` command. Read its documentation.

Problem B.3

1. Write a Matlab/Octave (or other language) function `[W,R] = house(A)` that computes an implicit representation of a full QR factorization $A = QR$ of an $m \times n$ matrix A with $m \geq n$ using Householder reflections. The output variables are a lower-triangular matrix $W \in \mathbb{C}^{m\times n}$ whose columns are the vectors v_k defining the successive Householder reflections, and a triangular matrix $R \in \mathbb{C}^{n\times n}$.

2. Write a Matlab/Octave (or other language) function `Q = formQ(W)` that takes the matrix W produced by `house` as input and generates a corresponding $m \times m$ orthogonal matrix Q.

3. Let Z be the matrix

$$Z = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{bmatrix}$$

Compute three reduced QR factorizations of Z: by the Gram-Schmidt routine `mgs` of Problem B.2, by the Householder routines `house` and `formQ`, and by Matlab/Octave built-in command `[Q,R] = qr(Z,0)` (if you are not using Matlab/Octave, you should replace `qr` with a suitable built-in function, e.g. from a numerical linear algebra library). Compare these three and comment on any differences you see.