MA398 MATRIX ANALYSIS AND ALGORITHMS: ASSIGNMENT 1

Please submit your solutions to this assignment via Moodle by **noon on Monday October 24$^{\textbf{nd}}$**. Make sure that your submission is clearly marked with your name, university number, course and year of study. Note that some of the questions are practical questions that are concerned with real-world problems and require coding in Python.

- The written part of the solutions may preferably be typed in LaTeX, or written on paper and subsequently scanned/photographed provided the images are clearly legible. You are required to deliver a single document entitled *MA398_Assignment1_FirstnameLastname.pdf*, outlining your solutions and explaining your interpretation and arguments to the questions .
- The Matlab or Python code scripts relevant to each question should be submitted as *MA398_ Assignment1_ ExerciseN.m/.ipynb* where you need to make sure that you document the environment in which I should be able to run your code.

To avoid losing marks unnecessarily, please make sure that,

- you show all intermediate steps of your calculations.
- you include all the coding that you have to do in order to obtain your answers and according to the instruction given above.
- include any plots you generate and label them appropriately.
- when you provide an answer to a question make sure that you justify your answer and provide details of any mathematical calculations that are required.

⚠Only in an emergency or if the Moodle submission is unavailable because of a general outage, the assignment should in the respective case be submitted by email to Randa.Herzallah@warwick.ac.uk, and olayinka.ajayi@warwick.ac.uk.

---

1. (Vector and Matrix Norms).

   (a) (Frobenius norm of a matrix.) The Frobenius norm of a matrix $A \in \mathbb{C}^{m\times n}$ is defined by,

   $$\|A\|_F := \sqrt{\operatorname{trace}(A^* A)} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{i,j}|^2\right)^{1/2}$$

   i. Assuming $m = n$, show that the Frobenius norm is a matrix norm.
   Solution: For all $n \times n$ matrices A and B and all real numbers α we have,

   -
   $$\|A\|_F = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{i,j}|^2\right)^{1/2} \geq 0$$

     which is true with $\|A\|_F = 0$ if and only if A is the zero matrix.

- $$\begin{aligned}\|\alpha A\|_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}|\alpha a_{i,j}|^2\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}|\alpha|^2|a_{i,j}|^2\\ &= |\alpha|^2\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^2\\ &= |\alpha|^2\|A\|_F^2\\ &\implies \|\alpha A\|_F = |\alpha|^2\|A\|_F^2\end{aligned}$$

- $$\begin{aligned}\|A+B\|_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}+b_{i,j}|^2\\ &\le \sum_{i=1}^{n}\sum_{j=1}^{n}(|a_{i,j}|+|b_{i,j}|)^2\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(|a_{i,j}|^2+2|a_{i,j}||b_{i,j}|+|b_{i,j}|^2)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^2+\sum_{i=1}^{n}\sum_{j=1}^{n}2|a_{i,j}||b_{i,j}|+\sum_{i=1}^{n}\sum_{j=1}^{n}|b_{i,j}|^2\end{aligned}$$

  Using the Cauchy-Schwarz inequality: $\sum_{i=1}^{n}x_iy_i \le (\sum_{i=1}^{n}x_i^2)^{\frac{1}{2}}(\sum_{i=1}^{n}y_i^2)^{\frac{1}{2}}$, yields,

  $$\begin{aligned}&\le \sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^2+2\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^2\Big)^{\frac{1}{2}}\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}|b_{i,j}|^2\Big)^{\frac{1}{2}}+\sum_{i=1}^{n}\sum_{j=1}^{n}|b_{i,j}|^2\\ &= \Big(\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^2\Big)^{\frac{1}{2}}+\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}|b_{i,j}|^2\Big)^{\frac{1}{2}}\Big)^2\\ &= \Big(\|A\|_F+\|B\|_F\Big)^2\\ &\implies \|A+B\|_F \le \|A\|_F+\|B\|_F\end{aligned}$$

ii. Show that if U and V are unitary matrices, then $\|UA\|_F = \|AV\|_F = \|A\|_F$. Thus the Frobenius norm is not changed by a pre- or post- orthogonal transformation.
Solution:
- To show that $\|UA\|_F = \|A\|_F$, we evaluate,

  $$\|UA\|_F^2 = \mathrm{trace}(UA)^T(UA) = \mathrm{trace}A^TU^TUA = \mathrm{trace}A^TA = \|A\|_F^2$$

  Similarly to show that $\|AV\|_F = \|A\|_F$, we evaluate,

  $$\|AV\|_F^2 = \mathrm{trace}(AV)^T(AV) = \mathrm{trace}(AV)(AV)^T = \mathrm{trace}AVV^TA^T = \mathrm{trace}AA^T = \mathrm{trace}A^TA = \|A\|_F^2$$

  where we have used, $\mathrm{trace}MN = \mathrm{trace}NM$.

iii. The singular value decomposition of the matrix A can be obtained from a factorisation of the form $A = U\Sigma V^*$ where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices, and $\Sigma \in \mathbb{C}^{m\times n}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal, $\sigma_1,\dots,\sigma_r$ known as the singular values of A. Show that,

$$\|A\|_F = \sqrt{\sigma_1^2+\cdots+\sigma_r^2}.$$

Solution:

$$\|A\|_F = \|U\Sigma V^*\|_F = \|\Sigma V^*\|_F = \|\Sigma\|_F = \sqrt{\sigma_1^2+\cdots+\sigma_r^2}$$

(b) (Vector norm.) For all $x \in \mathbb{C}^n$ show that $\|x\|_1 \geq \|x\|_2$.
Solution: Let $x = \left(x_1, x_2, \dots, x_n\right)^T$, and observing that,

$$\left(\sum_{i=1}^{n} |x_i|\right)^2 \geq \sum_{i=1}^{n} x_i^2$$
$$(|x_1| + |x_2| + \cdots + |x_n|)^2 \geq x_1^2 + x_2^2 + \cdots + x_3^2$$

Then taking the square root of both sides of both sides of the equation yields,

$$\sum_{i=1}^{n} |x_i| \geq \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$$
$$\implies \|x\|_1 \geq \|x\|_2$$

(c) (Induced matrix norms.) Consider the following matrix,

$$A = \begin{pmatrix} 4 & -2 & 0 \\ -3 & 6 & -2 \\ 0 & -3 & 5 \end{pmatrix}$$

- Compute the matrix norm induced by the infinity norm.
  Solution:
  The matrix norm induced by the infinity norm is defined as,

  $$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{i,j}|, \quad A \in \mathbb{C}^{n\times n}$$

  Hence,

  $$\sum_{j=1}^{n} |a_{1,j} = |a_{1,1}| + |a_{1,2}| + |a_{1,3}| = 4 + 2 + 0 = 6$$
  $$\sum_{j=1}^{n} |a_{2,j} = |a_{2,1}| + |a_{2,2}| + |a_{2,3}| = 3 + 6 + 2 = 11$$
  $$\sum_{j=1}^{n} |a_{1,j} = |a_{3,1}| + |a_{3,2}| + |a_{3,3}| = 0 + 3 + 5 = 8$$

  Thus, we have $\|A\|_\infty = \max\{6, 11, 8\} = 11$
- Compute the 1-norm. The matrix norm induced by the 1-norm is defined as,

  $$\|A\|_1 = \|A^*\|_\infty = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}|. \quad A \in \mathbb{C}^{n\times n}.$$

  Hence,

  $$\sum_{i=1}^{n} |a_{i,1} = |a_{1,1}| + |a_{2,1}| + |a_{3,1}| = 4 + 3 + 0 = 7$$
  $$\sum_{i=1}^{n} |a_{i,2} = |a_{1,2}| + |a_{2,2}| + |a_{3,2}| = 2 + 6 + 3 = 11$$
  $$\sum_{i=1}^{n} |a_{i,3} = |a_{1,3}| + |a_{2,3}| + |a_{3,3}| = 0 + 2 + 5 = 8$$

  Thus, we have $\|A\|_\infty = \max\{7, 11, 8\} = 11$

2. (Economic systems model). The topic of this question is to calculate the required output by each sector in an economy consisting of n sectors where the outputs are measured by $x = (x_1, x_2, \ldots, x_n)$. A sector i will need a number of units from sector j as well as from itself to produce a unit of its good which is referred to as a_{ij}. In addition there is a consumer demand, $D = (D_1, D_2, \ldots, D_n)$. The following equation describes this economic model,

$$x = Ax + D$$

Suppose that the economy consists of 4 sectors: electricity, water, agriculture, and manufacturing. Each of these sectors consumes a certain proportion of the total output the other sectors according to the following table.

	proportion produced by electricity	proportion produced by water	proportion produced by agriculture	proportion produced by manufacturing
proportion used by electricity	0.2	0.25	0.1	0.1
proportion used by water	0.15	0.1	0.005	0.005
proportion used by agriculture	0.3	0.6	0.1	0.3
proportion used by manufacturing	0.6	0.5	0.1	0.1

The consumer demand for each sector is given in the following vector,

$$D = \begin{pmatrix} 100 \\ 150 \\ 120 \\ 70 \end{pmatrix}$$

(a) Use Gaussian elimination method to calculate by hand the required output by each sector to meet the demand given by the matrix D.

Answer:

$$x = Ax + D$$
$$x = \begin{pmatrix} 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.1 & 0.005 & 0.005 \\ 0.3 & 0.6 & 0.1 & 0.3 \\ 0.6 & 0.5 & 0.1 & 0.1 \end{pmatrix} x + \begin{pmatrix} 100 \\ 150 \\ 120 \\ 70 \end{pmatrix}$$
$$\Longrightarrow \underbrace{\begin{pmatrix} 0.8 & -0.25 & -0.1 & -0.1 \\ -0.15 & 0.9 & -0.005 & -0.005 \\ -0.3 & -0.6 & 0.9 & -0.3 \\ -0.6 & -0.5 & -0.1 & 0.9 \end{pmatrix}}_{M=I-A} x = \underbrace{\begin{pmatrix} 100 \\ 150 \\ 120 \\ 70 \end{pmatrix}}_{D}$$

Now compute the upper triangular matrix of M,

- $$[M|D] = \left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ -0.15 & 0.9 & -0.005 & -0.005 & 150 \\ -0.3 & -0.6 & 0.9 & -0.3 & 120 \\ -0.6 & -0.5 & -0.1 & 0.9 & 70 \end{array}\right) \begin{array}{l} \\ (0.15/0.8)*R_1+R_2 \\ (0.3/0.8)*R_1++R_3 \\ (0.6/0.8)*R_1++R_4 \end{array}$$

  $$= \left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ 0 & 0.853125 & -0.02375 & -0.02375 & 168.75 \\ 0 & -0.69375 & 0.8625 & -0.3375 & 157.5 \\ 0 & -0.6875 & -0.175 & 0.825 & 145 \end{array}\right)$$

- $$\left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ 0 & 0.853125 & -0.02375 & -0.02375 & 168.75 \\ 0 & -0.69375 & 0.8625 & -0.3375 & 157.5 \\ 0 & -0.6875 & -0.175 & 0.825 & 145 \end{array}\right) \begin{array}{l} \\ \\ (0.69375/0.853125)*R_2++R_3 \\ (0.6875/0.853125)*R_2++R_4 \end{array}$$

  $$= \left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ 0 & 0.853125 & -0.02375 & -0.02375 & 168.75 \\ 0 & 0 & 0.8432 & -0.3568 & 294.7 \\ 0 & 0 & -0.19414 & 0.80586 & 280.98 \end{array}\right)$$

- $$\left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ 0 & 0.853125 & -0.02375 & -0.02375 & 168.75 \\ 0 & 0 & 0.8432 & -0.3568 & 294.7 \\ 0 & 0 & -0.19414 & 0.80586 & 280.98 \end{array}\right) \begin{array}{l} \\ \\ \\ (0.19414/0.8432)*R_3++R_4 \end{array}$$

  $$= \left(\begin{array}{cccc|c} 0.8 & -0.25 & -0.1 & -0.1 & 100 \\ 0 & 0.853125 & -0.02375 & -0.02375 & 168.75 \\ 0 & 0 & 0.8432 & -0.3568 & 294.7 \\ 0 & 0 & 0 & 0.7237 & 348.84.98 \end{array}\right)$$

- The above matrix is an upper triangular matrix from which the solution can be obtained using the BS algorithm,

  $$\begin{aligned} x_4 &= 482 \\ x_3 &= 553.46 \\ x_2 &= 226.623 \\ x_1 &= 325.25 \end{aligned}$$

(b) Suppose that $M \in \mathbb{C}^{n\times n}$ admits an LU factorization $M = LU$ and that n is an integer power of 2. Consider a block decomposition of M, L and U into $\frac{n}{2} \times \frac{n}{2}$ blocks as follows:

$$\underbrace{\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}}_{M} = \underbrace{\begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}}_{U}. \tag{1}$$

Write down a recursive method to compute LU factorizations of $n \times n$ matrices using block decompositions of the form (1).

Answer:

If $n = 1$, take $L = 1$, $U = M$.

Else write M using the block decomposition above.

- Compute the LU factorization of M_{11}. This gives U_{11} and L_{11}.
- Compute $U_{12} = L_{11}^{-1} M_{12}$. (The inverse of L_{11} can be written explicitely).
- Compute $L_{21} = M_{21} U_{11}^{-1}$. (The inverse can be computed by aubstitution).
- Compute the LU factorization of $M_{22} - L_{21} U_{12}$. This gives L_{22} and U_{22}.

(c) Use your method to compute by hand the LU factorization of the economic example described above in this question.

Answer:

First step: $B = M_{11} = \begin{pmatrix} 0.8 & -0.25 \\ -0.15 & 0.9 \end{pmatrix}$ The LU factorization of B can be done as explained in the recursive method above

- $B_{11} = 0.8 = U_{B_{11}}$, i.e. $L_{B_{11}} = 1$.
- $U_{B_{12}} = L_{B_{11}}^{-1} B_{12} = 1 \times (-0.25) = -0.25$
- $L_{B_{21}} = B_{21} U_{B_{11}}^{-1} = -\frac{0.15}{0.8} = -0.1875$
- $B_{22} - L_{B_{21}} U_{B_{12}} = 0.9 - 0.1875 \times 0.25 = 0.85325$.

This gives $L_{11} = \begin{pmatrix} 1 & 0 \\ -0.1875 & 1 \end{pmatrix}$, and $U_{11} = \begin{pmatrix} 0.8 & -0.25 \\ 0 & 0.85325 \end{pmatrix}$.

Second step: $U_{12} = L_{11}^{-1} M_{12}$, hence $U_{12} = \begin{pmatrix} -0.1 & -0.1 \\ -0.0238 & -0.0238 \end{pmatrix}$.

Third step: $L_{21} = M_{21} U_{11}^{-1} = \begin{pmatrix} -0.375 & -0.8132 \\ -0.75 & -0.8059 \end{pmatrix}$.

Fourth step: $M_{22} - L_{21} U_{12} = \begin{pmatrix} 0.8432 & -0.3568 \\ -0.1941 & 0.8059 \end{pmatrix}$. We can take $L_{22} = \begin{pmatrix} 1 & 0 \\ -0.2302 & 1 \end{pmatrix}$, and $U_{22} = \begin{pmatrix} 0.8432 & -0.3568 \\ 0 & 0.7238 \end{pmatrix}$. This gives at the end

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -0.1875 & 1 & 0 & 0 \\ -0.375 & -0.8132 & 1 & 0 \\ -0.75 & -0.8059 & -0.2302 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 0.8 & -0.25 & -0.1 & -0.1 \\ 0 & 0.8531 & -0.0238 & -0.0238 \\ 0 & 0 & 0.8432 & -0.3568 \\ 0 & 0 & 0 & 0.7238 \end{pmatrix}.$$

.

(d) Implement your method as a Python function `L,U = recursive_lu(A)`. You may assume that A is factorisable without pivoting. The SciPy Linear Algebra Library may come in handy.

(e) Generate a random square matrix of size $2^k \times 2^k$ and apply your Python code to find its LU decomposition. Do this several times for several values of k, and plot the runtimes. Use your plot to formulate a conjecture about the computational cost of your method as a function of k.

Hint: Consider `time.time()` in Python as a simple way to consider timing.