EE5609 – Matrix Theory
Programming Assignment 1

Due on Oct 10, 11.59pm

Team Details

Please register your team (team size can be either 1 or 2) here on or before Sep 28:

`https://forms.gle/PRHt48ExPb7fGXj17`

Only one registration per team. I will inform you your team ID once all the teams register.

I will send you another Google Form link for the assignment submission.

Problem Description

You must write a Julia code to perform row reduction, compute the rank of a matrix $\boldsymbol{A}$ and to determine if $\boldsymbol{Ax} = \boldsymbol{b}$ has a solution. The input to the program is the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$, and the output are the echelon matrix $\boldsymbol{U}$ of $\boldsymbol{A}$, the rank r of $\boldsymbol{A}$ and a boolean variable `consistent`. The variable `consistent=true` if $\boldsymbol{Ax} = \boldsymbol{b}$ has at least one solution, and is equal to `false` otherwise. The size of $\boldsymbol{A}$ can be arbitrary (wide, tall or square); you must make sure that your code works for matrices of size up to 500×500.

The programming assignment has the following requirements.

- The matrices ans vectors we are interested in are NOT defined over $\mathbb{R}$ or $\mathbb{C}$, but over a field containing four elements $\mathbb{F}_4 = \{0, 1, 2, 3\}$ (this is an example of a *finite field* or *Galois field*). That is, each entry of $\boldsymbol{A}$ and $\boldsymbol{U}$ will be $0, 1, 2$ or 3. The rules for arithmetic in $\mathbb{F}_4$ are given below.

  Addition

  $$\begin{aligned} &0+0=0, \quad 0+1=1, \quad 0+2=2, \quad 0+3=3, \\ &1+0=1, \quad 1+1=0, \quad 1+2=3, \quad 1+3=2, \\ &2+0=2, \quad 2+1=3, \quad 2+2=0, \quad 2+3=1, \\ &3+0=3, \quad 3+1=2, \quad 3+2=1, \quad 3+3=0 \end{aligned}$$

  Multiplication

  $$\begin{aligned} &0 \cdot a = a \cdot 0 = 0 \text{ for all } a \in \mathbb{F}_4 \\ &1 \cdot a = a \cdot 1 = a \text{ for all } a \in \mathbb{F}_4 \\ &2 \cdot 2 = 3, \quad 2 \cdot 3 = 3 \cdot 2 = 1, \quad 3 \cdot 3 = 2 \end{aligned}$$

  Negation and Subtraction

  $$\begin{aligned} -a &= a \text{ for all } a \in \mathbb{F}_4 \\ a - b &= a + b \text{ for all } a, b \in \mathbb{F}_4 \end{aligned}$$

  Multiplicative Inverse

  $$1^{-1} = 1, \quad 2^{-1} = 3, \quad 3^{-1} = 2$$

- In Julia, the entries of $\boldsymbol{A}, \boldsymbol{U}$ should be restricted to the values $0, 1, 2, 3$ only. However, the Julia data type of these matrices must be `Matrix{Int64}`. The type of r must be `Int64`, and the type of `consistent` must be `Bool`.
- You must use partial pivoting to determine the row interchanges. To pick the 'largest' available pivot entry at each stage, use the following ordering among the elements of $\mathbb{F}_4$

  $$0 < 1 < 2 < 3.$$

  If there are two rows that yield the same largest entry, you must pick the row with the smaller row index for performing the row swap.

  You must make sure that the above rule is coded into your program. It will be impossible for me to verify the correctness of your program if this pivoting rule is not implemented.
- Each team will submit a single Julia file `EE5609TeamIDxx.jl` through a Google Forms link (to be shared with you later), where the last 2 digits xx denote your Team ID. This file must contain the definition of your function `rankconsistencyTeamIDxx` which must be callable as follows:

```
include("EE5609TeamID07.jl")  # including the file from Team number 07
m = 100 # must work for any m <= 500
n = 50 # must work for any n <= 500
A = rand(0:3,(m,n)) # this is a Matrix{Int64} array with entries 0,1,2,3
b = rand(0:3,m) # b::Vector{Int64}
U, r, consistent = rankconsistencyTeamID07(A,b)
# U::Matrix{Int64}, r::Int64, consistent::Bool
```

- You must not use any Julia packages to implement your code. The implementation must be completely written by you.

Evaluation

The maximum marks for this assignment is 25, plus a 5 marks bonus over and above the maximum marks of 25.

Each submission will be evaluated for correctness using inputs generated randomly. Marks will be awarded based on the number of correct responses generated by your code.

The computation time of your code will also be measured. The teams with the smallest computation time will be awarded an additional 5 marks bonus. Among the teams whose program yields correct responses for all the randomly generated inputs, the top 10% teams will receive the bonus marks.