# High Performance Computing
# USN-HCMV, Paris 13
# Joint Master 2022
# ———-
# Worksheet 2

Let A be the a) matrix defined in worksheet 1, b) The 1D Finite Difference Matrix c) Matrices defined in Matrix Market

## Exercise 1

In the restarting gmres.m you received , add the missing lines that transform the Hessenberg matrix into an upper matrix by applying Givens rotation and which also provides the error at iteration j+1
Check that your restarting gmres.m is correct using manufactured solutions

- First , use a restart parameter that is greater than the matrix dimension. What does this imply ?
- Analyse in terms of computing and memory storage , the effect of the restarting parameter

Comment the efficiency of gmres compared to the previous iterative methods worksheet 1)

## Exercise 2

Left preconditioner : Solve $MAx = Mb$
Right preconditioner : Solve $AMy = b$ , and then $x = My$
Using the above gmres.m, modify it to precgmres.m so as to be able to use a right preconditioner $M$ that can be changed at each restart of gmres.
Validate your precgmres.m defining $M = diag(A)$

## Exercise 3

Look into the matlab function ilu that performs an ILU(0) approximation of A. [L0, U0] = ILU(A) Use this ILU(0) as a right preconditioner in precgmres.m to solve Ax = b as in TP1

## Exercise 4

Look into the matlab function schur that performs a Schur decomposition of a matrix. (help schur)

# Exercise 5

Following the algorithm below, modify precgmres.m to obtain a function that performs a Deflation Preconditioned Gmres

**Algorithm** DEFLGMRES(m,l)

Choose $x_0$ ;
$M = I_n$ ;
U= ;
1. $r_0 = b - Ax_0$ , $\beta = \|r_0\|$, $v_1 := r_0/\beta$ ;
2. Generate the Arnoldi basis applied to $AM^{-1}$ and the associated Hessenberg matrix $\tilde{H_m}$ starting with $v_1$;
3. Compute $y_m$ which minimises $\|\beta e_1 - \tilde{H_m} y\|$ and $x_m = x_0 + M^{-1} V_m y_m$ ;
4. If convergence Stop, else set ;
 $x_0 = x_m$ ;
 Compute l Schur vectors of $H_m$ noted $S_l$ ;
 Compute the approximation of $|\lambda_n|$ ;
 Orthogonalize $V_m S_l$ against $U$ ;
 Increase $U$ with $V_m S_l$ ;
 $T = U^T A U$ ;
 $M^{-1} = I_n + U(|\lambda_n| T^{-1} - I_r) U^T$ ;
 Go To 1 ;