MAT 423: Homework 5 (09/26)

Ex 1. [3pts]

- $|||A|||_1 = 2$, $|||A|||_\infty = 2$. Characteristic polynomial: $P_A(\lambda) = -(\lambda+1)(\lambda^2-\lambda-2)$ thus the eigenvalues are $\lambda = -1, 2$. Therefore, $\rho(A) = 2$. 1.5pt
- $|||B|||_1 = 4$, $|||B|||_\infty = 4$. Characteristic polynomial: $P_B(\lambda) = (3-\lambda)(\lambda - 4\lambda + 3)$ thus the eigenvalues are $\lambda = 3,\ 2 \pm 1$. Therefore, $\rho(B) = 3$. 1.5pt

Ex 2.

Take $A = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$. Then: $|||A|||_\infty = 2 \ < \ 3 = |||A|||_1$.

For B, we can take $B = A^T = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$. Then: $|||B|||_1 = 2 \ < \ 3 = |||B|||_\infty$.

Ex 3. [4pts]

a) Consider $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. We have $\varphi(A) = |det(A)| = 0$ but $A \neq 0$. So φ is not a norm. 1.5pt

b) We prove the three conditions:

i) $\varphi(A) \geq 0$. Moreover if $\varphi(A) = 0$ then $a_{ij} = 0$ for all i, j, thus $A = 0$. .5pt

ii) For any matrix A and scalar α: .5pt

$$\begin{aligned} \varphi(\alpha A) &= \max_{1\leq i,j\leq n} |\alpha \cdot a_{ij}| \\ &= |\alpha| \max_{1\leq i,j\leq n} |a_{ij}| = |\alpha|\varphi(A). \end{aligned}$$

iii) For any nÖn matrices A and B: .5pt

$$\begin{aligned} \varphi(A+B) &= \max_{1\leq i,j\leq n} |a_{ij} + b_{ij}| \\ &\leq \max_{1\leq i,j\leq n} |a_{ij}| + \max_{1\leq i,j\leq n} |b_{ij}| = \varphi(A) + \varphi(B) \end{aligned}$$

Therefore, φ is a norm. However, it is not a matrix norm as it does not satisfy $\varphi(AB) \leq \varphi(A)\varphi(B)$. Indeed, take: $A = B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, we have $\varphi(A) = \varphi(B) = 1$ and $\varphi(AB) = 2$. 1pt

Ex 4. [3pts] + [1pt] (Gauss-Seidel)

a) See script below. 1pt

b) We plot in figure 1 the evolution of the errors (using $\|.\|_2$) for both Jacobi and Gauss-Seidel methods. We use logarithmic scale in y to visualize the geometric decay of the error.
We find after 400 iterations:

$$x_G \approx (5, 9, 12, 14, 15, 15, 14, 12, 9, 5)^T. \quad \text{1pt}$$

c^*) To estimate the convergence rate of each method, we perform a linear regression:

$$\log \text{error}(n) \approx \log C + n \cdot \log k$$

We find numerically the rate $k = .9595$ for the Jacobi-method, whereas $k = .9207$ for the Gauss-Seidel method. To confirm this decay rate, we plot (black dash-line) on the figure 1 the decay with this exact rates. 1pt 1pt

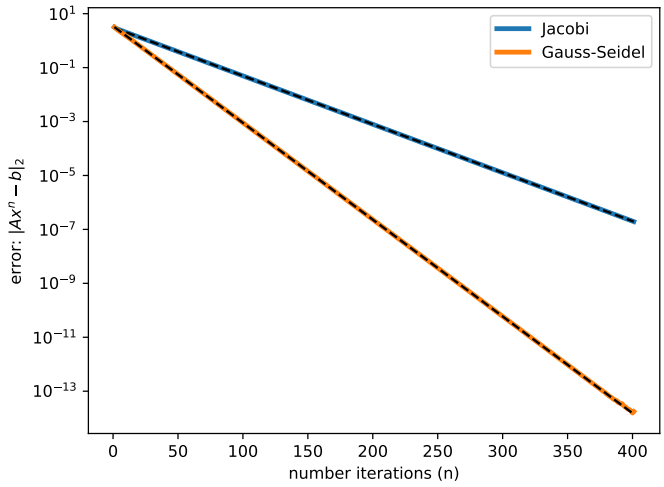

Figure 1: Evolution of the errors $\|Ax^n - b\|_2$ for both Jacobi and Gauss-Seidel methods. Both methods converge and the Gauss-Seidel converges faster.

Remark. We can also estimate analytically the expected decay rate of both methods by looking at the spectral radius of the matrices: $J_j = D^{-1}(L+U)$ and $J_g = (D-L)^{-1}U$. We find that $\rho(J_j) = .9595$ and $\rho(J_g) = .9206$ which are consistent with the estimations found previously.

```
#=
  Comparison between Jacobi and Gauss-Seidel method
=#
using LinearAlgebra
using PyPlot

function Jacobi_method(A,x0,b,nbIter)
    # init
    error = zeros(nbIter+1)
    error[1] = norm(A*x0-b)
    x = copy(x0)
    # matrices
    D = diagm(0=>diag(A))
    R = D-A
    invD = diagm( 0=> 1 ./diag(A))
    # loop
    for k=1:nbIter
        x .= invD*(R*x+b)
        error[k+1] = norm(A*x-b)
    end
    #println(error)
    return x,error

end

function Gauss_Seidel_method(A,x0,b,nbIter)
    # init
    error = zeros(nbIter+1)
    error[1] = norm(A*x0-b)
    x = copy(x0)
    # matrices
    DmL = LowerTriangular(A)
    R = DmL-A
    invDmL = inv(DmL)
    # loop
    for k=1:nbIter
        x .= invDmL*(R*x+b)
        error[k+1] = norm(A*x-b)
    end
    #println(error)
    return x,error

end
```

```
# no need in Python or Matlab/Octave
function polyfit(x, y, n)
    A = [ float(x[i]).^p for i = 1:length(x), p = 0:n ]
    return A \ y
end

#-----------------------------------------#
#-------            Testing          -------#
#-----------------------------------------#
# A) Construction matrix
N = 10
D = diagm(0=>2*ones(N))
U = diagm(1=>ones(N-1))
L = diagm(-1=>ones(N-1))
A = D-L-U
b = ones(N)
x0  = zeros(N)
nbIter = 400
# B) Jacobi method
x_J,error_J = Jacobi_method(A,x0,b,nbIter)
# error(n) = C k^n => log(error(n)) = log(C) + n log(k)
coeff_J = polyfit(1:(nbIter+1),log.(error_J),1)
# C) Gauss-Seidel method
x_GS,error_GS = Gauss_Seidel_method(A,x0,b,nbIter)
coeff_GS = polyfit(1:(nbIter+1),log.(error_GS),1)
# B) plot
figure(1)
clf()
semilogy(1:(nbIter+1),error_J,linewidth=3,label="Jacobi")
plot(1:(nbIter+1), exp(coeff_J[1])*exp(coeff_J[2]).^(1:(nbIter+1)),"k--")
plot(1:(nbIter+1),error_GS,linewidth=3,label="Gauss-Seidel")
plot(1:(nbIter+1), exp(coeff_GS[1])*exp(coeff_GS[2]).^(1:(nbIter+1)),"k--")
legend()
xlabel("number iterations (n)")
ylabel(L"error: $\|Ax^n-b\|_2$")
savefig("Jacobi_VS_GaussSeidel.pdf")
```