

Programación: normas matriciales 1 e ∞

Objetivos. Programar las funciones que calculen las normas matriciales asociadas a las normas vectoriales $\|\cdot\|_1$ y $\|\cdot\|_\infty$.

Requisitos. Normas matriciales, programación de funciones, ciclos `for`, entradas de matrices.

1. Fórmulas. Dada una matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, sus normas matriciales $\|\cdot\|_{\text{matr},1}$ y $\|\cdot\|_{\text{matr},\infty}$ se pueden calcular mediante las siguiente fórmulas:

$$\|A\|_{\text{matr},1} = \max_{1 \leq k \leq n} \sum_{j=1}^m |A_{j,k}|, \quad \|A\|_{\text{matr},\infty} = \max_{1 \leq j \leq m} \sum_{k=1}^n |A_{j,k}|.$$

2. Problema: la norma matricial 1. En algún lenguaje de programación escribir una función que calcule la norma $\|A\|_{\text{matr},1}$ de la matriz dada A . Una posible solución en el lenguaje de MATLAB:

```
function result = matrixnorm1(A),
    [m, n] = size(A);
    result = 0;
    for k = 1 : n,
        s = 0;
        for j = 1 : m,
            s = s + ???;
        end
        result = max(result, s);
    end
end
```

3. Problema: calcular la norma matricial 1 usando las funciones `sum` y `max` de manera eficiente. Estudie la sintaxis de la función `sum` del lenguaje MATLAB. ¿Qué pasa al aplicar esta función a una matriz? Ejecute los siguientes comandos:

```
A = [3 4 5; 11 12 13; 51 52 53];
sum(A)
sum(A, 2)
```

Usando las funciones `sum` y `max` escriba la función `matrixnorm1` de manera más breve y más eficiente, evitando ciclos.

4. Problema: la norma matricial ∞ . En algún lenguaje de programación escribir una función que calcule la norma $\|A\|_{\text{matr},\infty}$ de la matriz dada A .

```
function result = matrixnorminf(A),  
    [m, n] = size(A);  
    ...  
end
```

5. Pruebas con matrices pequeñas. Solución en el lenguaje de MATLAB (guardar en el archivo `testmatrixnormsmall.m`):

```
function [] = testmatrixnormsmall(),  
    A = [3 -4; -5 1];  
    display([matrixnorm1(A), norm(A, 1)]);  
    display([matrixnorminf(A), norm(A, Inf)]);  
end
```

6. Pruebas con matrices grandes. Solución en el lenguaje de MATLAB (guardar en el archivo `testmatrixnormlarge.m`):

```
function [] = testmatrixnormlarge(),  
    for n = [256, 512, 1024, 2048, 4096],  
        display(n);  
        A = 2 * rand(n, n) - ones(n, n);  
        t = cputime();  
        r = matrixnorm1(A);  
        t1 = cputime() - t;  
        t = cputime();  
        r = matrixnorminf(A);  
        tinf = cputime() - t;  
        display([t1 tinf]);  
    end  
end
```

Después de ejecutar esta función se recomienda corregirla de la siguiente manera: en vez de $n = 256, \dots, 4096$ poner tales valores de n (por ejemplo, algunas potencias de 2) para los cuales el tiempo de ejecución es de 10^{-2} segundos a 10^2 segundos.

¿Cómo se cambia el tiempo de ejecución al multiplicar n por 2?