

Control Matlab

Escribe un programa en Matlab que:

- Lea unha matriz cadrada **a** dende o arquivo `matriz.dat` e determine a súa orde n .
- Lea por teclado un vector **v**, que tamén debe ser de orde n (o programa non debe verificalo).
- Cree unha matriz **b** cadrada de orde n e calcule b_{ij} da seguinte forma:

$$b_{ij} = \frac{1}{i+j} \left(v_i^2 + \sum_{k=1}^i v_k a_{kj} \right), \quad i, j = 1, \dots, n \quad (1)$$

- Engadir ao arquivo `matriz.dat` a matriz **b** (unha fila en cada liña do arquivo, escribindo os números decimais con dous díxitos) e o vector **v** na derradeira fila.

Proba co arquivo `matriz.dat` seguinte:

```
1 2 3 4
5 6 7 8
9 8 7 6
5 4 3 2
```

Usa o vector $\mathbf{v} = (1, 2, 3, 4)$. O contido final do arquivo `matriz.dat` debe se-lo seguinte:

```
1 2 3 4
5 6 7 8
9 8 7 6
5 4 3 2
1.00 1.00 1.00 1.00
5.00 4.50 4.20 4.00
11.75 9.40 7.83 6.71
14.80 11.67 9.43 7.75
1 2 3 4
```

SOLUCIÓN:

```
clear all
a = load('matriz.dat');
n = size(a, 1); b = zeros(n);
v = input('v (orde n) [ ]? ');
for i = 1:n
    for j = 1:n
        % SOLUCION SEN VECTORIZAR
        s = 0;
        for k = 1:i
            s = s + v(k)*a(k, j);
        end
        b(i, j) = (v(i)^2 + s)/(i + j);
        % SOLUCION VECTORIZADA ALTERNATIVA
        b(i, j) = (v(i)^2 + sum(v(1:i).*a(1:i,j)))/(i + j);
    end
end
```

```
        end
    end
    fid = fopen('matriz.dat', 'a');
    if -1 == fid
        fprintf('erro abrindo matriz.dat\n');
        break
    end
    for i = 1:n
        fprintf(fid, '%.2f ', b(i, :));
        fprintf(fid, '\n');
    end
    fprintf(fid, '%i ', v);
    fclose(fid);
```