



MATLAB

Informatica A - 26/11/2024

Recap



Matlab è un software di programmazione scritto in C sviluppato per il calcolo numerico e l'analisi statistica. Matlab permette di eseguire script scritti in un linguaggio di programmazione proprio: è un linguaggio interpretato e non compilato

- | | |
|-----------------------|---|
| Current Folder | → Per accedere ai file dell'utente |
| Command Window | → Per inserire i comandi della riga di comando |
| Workspace | → Per visualizzare i dati creati o importati |
| Script | → Per salvare funzioni e programmi più complicati |

Recap



Matlab può essere utilizzato come calcolatrice. Se non specificato il risultato è inserito automaticamente nella variabile **ans**. Se la riga finisce con un ';' (punto e virgola), l'operazione viene eseguita ma il risultato non è mostrato.

Non occorre dichiarare le variabili come nel C, basta assegnare un valore per dichiarare automaticamente la variabile. **NON** si possono utilizzare variabili che non siano mai state assegnate.

Esempi: somma(5+3, a=5+3, a=5+3;), potenza(5^3), radice(sqrt(144)),divisione destra(5/3), divisione sinistra(5\3→3diviso5), valore assoluto (abs(-3)), pi(pi)

Recap - Vettori e Matrici

I **vettori** sono definiti con parentesi quadre. In un vettore **riga** gli elementi sono separati da **virgole** o **spazi** e in un vettore **colonna** gli elementi sono separati da **;** (o **andando a capo**).

Attenzione: Gli indici partono da 1!

Le **matrici** sono definite affiancando vettori di dimensioni compatibili, usando gli operatori **,** (o **spazio**) e **;** (o **vai a capo**)

L'operatore **'** (apostrofo) di trasposizione inverte le righe e le colonne di vettori e matrici.

Esempi: $A = [1 \ 2 \ 3 \ 4 \ 5]$ $A = [1, 2, 3, 4, 5]$ $A = [1; 2; 3; 4; 5]$ A' (trasposto) $A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$
 $A(2,3)$ (per accedere) $A(2:end)$ (su vettore) $A(2:end, 2:end)$ su matrice $A(2:)A(:3)$, $B = [A; A.^2; A.^3]$ $\text{zeros}(5)$ (5x5 di zeri) $\text{zeros}(2,3)$ (2x3 di zeri), $\text{ones}(3,2)$ (3x2 di uni), $5 * \text{ones}(3)$,
 $\text{diag}([1,2,3,4])$, $\text{diag}(A)$, $\text{diag}(\text{diag}(A))$ $C = [A \ B]$ $C = \text{horzcat}(A, B)$ $C = \text{vertcat}(A, B)$ (attenzione a un mismatch di dimensioni) $\det(A)$ $A = 3:13$ (vettore riga da 3 a 13) $A = 3:4:13$ (vettore riga con skip di 4) $A = 11.5:(14.2-11.5)/6:14.2$ $A = \text{linspace}(11.5, 14.2, 7)$ $+++ \text{ Somma/prodotto/Divisione}$

Recap - Altre funzioni con le matrici



- `rand(n)` o `rand(n,m)`. `rand` genera una matrice `nxn` o `nxm` con numeri casuali tra 0 e 1.
- `randi([min max],[n m])` genera una matrice `nxm` con numeri interi casuali da `min` a `max`.
- `size(A)` restituisce le dimensioni della matrice
- `length(A)` restituisce la lunghezza del vettore `A`
- `mean(A)` restituisce la media delle colonne di `A`
- `sum(A)` restituisce la somma calcolata sulle colonne di `A`

Esempi: `rand(5)` `rand(2,3)` `randi(8)` `randi([3 10])` `randi([3 10] [2 3])` `size(A)` `size(A, 1)` `size(A,2)`
`length(A)` `mean(A)=mean(A,1)` `mean(A,2)`, `mean(mean(A))` (per quella di tutta la matrice),
`sum(A)=sum(A,1)` `sum(A,2)` `sum(sum(A))` +++ Altre funzioni `mod/rem/find`

Recap - Condizioni e Cicli

Come in C, i principali costrutti in MATLAB sono **if/else**, cicli **while** e cicli **for**.

Le parentesi {} vengono sostituite da **end** alla fine di ogni costrutto

```
if I == J
    A(I, J) = 2;
elseif abs(I-J) == 1
    A(I, J) = -1;
else
    A(I, J) = 0;
end
```

```
while a > 0
    %Istruzioni da ripetere
end

while a > 0
    %Istruzioni da ripetere
end
```

Recap - Funzioni



Matlab permette di creare delle funzioni. Queste vanno create in un file .m che abbia lo stesso nome della funzione.

Per i parametri di ingresso non occorre specificare il tipo di variabile. Anche per i parametri di uscita non occorre specificarne il tipo e, a differenza del C, si possono avere anche più parametri. Non esiste il passaggio per puntatore. Se si vogliono modificare gli stessi parametri di ingresso, o si sovrascrivono passandoli sia in ingresso che in uscita, o si usano variabili globali.

```
function [out1 , out2 , ...] = nomeFunzione(in1 , in2 ...)  
    %Contenuto della funzione  
end
```

Esercizio 1



1. Creare una matrice 4X5 contenente solo 3
 2. Inserire valori da 1 a 5 nella riga 2
 3. Sostituire le colonne dispari con soli numeri 10
 4. Sostituire i numeri dispari con 0
 5. Quanto vale la radice quadrata della somma di A?
 6. Eliminare le righe dispari
1. `A = 3*ones(4,5)`
 2. `A(2, :) = 1:5`
 3. `A(:, 1:2:end)=10*ones(size(A(:, 1:2:end)))`
 4. `A(mod(A,2)~=0)=0`
 5. `sqrt(sum(sum(A)))`
 6. `A(1:2:end, :)=[]`

Esercizio 2



1. Creare una matrice con numeri casuali da -10 a 10
 2. Eliminare le colonne con media > 0
 3. Eliminare la colonna 3
 4. Inserire dopo la seconda colonna una colonna contenente valori casuali da -11 a 37
 5. Chiedere un numero all'utente e stampare solamente i numeri di cui il numero inserito è un divisore
 6. Creare una matrice B che contiene i valori di A meno la media dei soli valori negativi
1. `A = randi([-10 10], [4 5])`
 2. `A(:, mean(A)>0) = []`
 3. `A(:, 3) = []` oppure `A = [A(:,1:2) A(:, 4:end)]`
 4. `A = [A(:,1:2) randi([-11 37], [4,1]) A(:, 3:end)]`
 5. `n = input('Inserisci un numero: ');
A(rem(A,n)==0)`
 6. `B = A - mean(A(A<0))`

Esercizio 3



Scrivere una funzione matlab che calcoli la somma dei valori interi da 1 fino a n con n inserito dall'utente. Stampare a video solo il valore della somma

```
function s = somma(n)
    if (n==0)
        s = 0;
    else
        s = n + somma(n-1);
    end
end
```

Esercizio 4



Scrivere una funzione ricorsiva per il calcolo nella serie di fibonacci

```
function n = fibonacci(x)
    if (x==1)
        n=0;
    else if (x==2)
        n=1;
    else
        n = fibonacci(x-1)+fibonacci(x-2);
    end
end
```

Esercizio 5



1. Creare la matrice M con dimensione di righe e colonne casuali (diverse) da 5 a 10 contenente solo valori pari a 1
2. Inserire nelle righe dispari valori casuali da 1 a 10
3. Calcolare la media delle celle con valore maggiore di 3

1. `M = ones(randi([5 10], [1 2]));`
.
.
2. `M(1:2:end, :) = randi([1, 10], size(M(1:2:end, :)))`
.
3. `mean(M(M>3))`

Esercizio 6

Scrivere una funzione che presa in ingresso la matrice M, crei una nuova matrice “specchio” N con la prima colonna in ultima posizione la seconda in penultima e così via.

```
function [N] = funzione(M)
    for i=1:1:size(M,2)
        N(:, size(M,2)-i+1) = M(:, i);
    end
end
```

Esercizio 7



1. Creare un vettore colonna A e un vettore colonna B contenente rispettivamente 6 e 5 valori casuali differenti tra loro con valori da 1 a 10
2. A partire dai vettori A e B, creare una matrice C di 6 righe e cinque colonne
3. Cancellare tutte le righe che hanno come primo valore un numero inferiore a 10
4. Calcolare la somma di tutte le colonne dispari

1. `A = randi([1 10], [6 1]);`
`B = randi ([1 10], [1 5]);`
oppure `A = randi([1 10], [6 1]);`
`B = randi ([1 10], [5 1]);`
2. `C = A*B, C=A.*B`
`C = A*B' C=A.B'`
3. `C(C(:,1)<10,:) = []`
.
4. `sum(sum(C(:, 1:2:end)))`

Esercizio 8



Scrivere una funzione che presa in ingresso la matrice C crei una nuova matrice cornice ponendo cioè a zero tutti gli elementi non appartenenti alla prima e ultima riga/colonna

```
function [out] = funzione(in)
    out = in;
    for i=2:1:size(in,1)-1
        for j=2:1:size(in,2)-1
            out(i,j) = 0;
        end
    end
end
```

Esercizio 9



Scrivere una funzione che presi in ingresso il numero di righe n e colonne m crea la matrice M $n \times m$ contenente i primi $N = (n \times m)$ numeri dispari

```
function out = luglio5_19(row, column)
    c = 1;
    for (i=1:row)
        for (j=1:column)
            out(i, j) = c;
            c = c+2;
        end
    end
end
```


Esercizio 9



1. Sostituire in una colonna a caso tutti valori pari a 1
 2. Eliminare la riga con la media più bassa
 3. Moltiplicare per -1 tutte le celle che contengono numeri divisibili per 7
1. `A(:, randi(size(A,2))) = ones(size(A,1),1)`
.
 2. `M(mean(M,2)==min(mean(M, 2)), :)=[]`
 3. `M(mod(M, 7) == 0) = M(mod(M,7)==0)*-1`