

**FLUSSO**

**MATLAB**

**R**  
**I**  
**X**

**O**  
**R**  
**A**  
**T**  
**O**  
**R**  
**Y**

**LEZIONE 3**

- Esercizio flusso
- Matrici
- Algebra matriciale

www.stoianov.it      matlab@stoianov.it

## Esercizio: Flusso, IF / CASE

**Compito:** Calcolare la superficie totale di N elementi grafici, di tipo disco o quadrato, i grandezza ed il tipo di quali sono richiesti dal utente.

- 1) Implementare una funzione che calcola la superficie secondo il tipo di elemento grafico

```

if type=='d'
...
else
...
end
switch <selector>
case 'd', ...;
case 's', ...;
otherwise, ...;
end

```

- 2) Scrivere un m-script che:

- (a) chiede quanti elementi (N) ci sono
- (b) fa un ciclo da 1 a N (1:N)
- (c) chiede info per ciascun elemento
- (d) calcola sa superficie di questo elemento
- (e) calcola la somma totale

```

for i=1:N
...
end

```

## Esercizio: Flusso, IF / CASE

Compito: Calcolare la superficie totale di N elementi grafici, di tipo disco o quadrato, i grandezza ed il tipo di quali sono richiesti dal utente.

- 1) Implementare una funzione che calcola la superficie secondo il tipo di elemento grafico

```
function s = calcsurf(type, r)
```

```
% Calculate the surface of a (d)isk or (s)quare
```

```
switch type
```

```
case 'd', s = pi * r^2;
```

```
case 's', s = (2*r)^2;
```

```
otherwise, s=0; disp('Wrong type');
```

```
end
```

```
if type=='d'           switch <selector>
...                   case 'd', ...;
else                  case 's', ...;
...                   otherwise, ...;
end                   end
```

File  
calcsurf2.m

CONSOLE

```
>> calcsurf('d',1)
```

```
>> calcsurf('s',1)
```

matlab@stoianov.it

## Esercizio: Flusso (2), FOR

Compito: Calcolare la superficie totale

- 2) Scrivere un m-script che:

(a) chiede quanti elementi (N) ci sono

(b) fa un ciclo da 1 a N (1:N)

(c) chiede info per ciascun elemento

(d) calcola sa supecie di questo elemento

(e) calcola la somma totale

```
N = input('How many elements?');
```

```
S = 0;
```

```
for i=1:N
```

```
    T=input('"d"isk or "d"quare ?');
```

```
    R = input('Radius ?');
```

```
    S = S + calcsurf(T,R);
```

```
end
```

```
fprintf('Total surface: %.1f \n',S);
```

matlab@stoianov.it

File  
elements.m

33

## Esercizi per casa

- Realizzare una funzione **fact**(n) che calcola il **fattoriale**:  $n!$  ( $=1*2*3* \dots n$ )
- Realizzare un programma che chiede l'utente di **indovinare** un numero da 1 a 10 con 5 tentativi
- **Calcolatrice**: realizzare la funzione **calc**(x,oper,y) che prende due numeri (x, y) e una lettera (oper) che indica un'operazione algebrica ('+', '-', '\*', '/') ed applica l'operazione sui numeri, e riporta il risultato.  
Ad esempio: `calc(2,'+',5)` fa la somma di 2 e 5.

matlab@stoianov.it

34

## MATrix LABoratory

La ricerca naturalistica offre una enorme quantità di dati descritti in miglior modo utilizzando matrici

MATLAB è ideato con lo scopo di **elaborare matrici** ed è molto efficace in tale tipo di calcoli.

A differenza dei linguaggi di programmazione "procedurali", in MATLAB **le iterazioni servono per controllo del flusso ad alto livello** ma **non per implementare elaborazioni su dati strutturati**.

**Tutti i dati in MATLAB sono matrici**, di dimensioni: 0 (scalari), 1 (vettori), 2 o più (matrici).

matlab@stoianov.it

35

# Matrici

- 0-d scalari 5.13
- 1-d vettori (riga: 1xN, colonna: Nx1)
- 2-d matrici MxN (ad es., un'immagine)
- 3-d matrici MxNxK (as es., un volume)
- 4-d matrici MxNxKxT (es: un volume nel tempo)
- ....
- N-d ...

matlab@stoianov.it

36

# Creazione

**[ ]** delimitatore di matrici  
" " == " ," concatena valori in un vettore-riga  
 $v=[4, 1, 6] \Rightarrow 4 \ 1 \ 6$

**;** concatena valori in un vettore-colonna  
 $v=[1; 5; 7] \Rightarrow 1$   
5  
7

**," & " ;** usarne insieme per costruire matrici 2D  
 $A=[1 \ 2 \ 3; 4 \ 5 \ 6] \Rightarrow 1 \ 2 \ 3$   
4 5 6

**inizio:passo:fine** definisce una progressione (vettore-riga)  
 $l= 1:10 \Rightarrow 1 \ 2 \ 3 \ 4 \ \dots \ 10$   
 $l= 10:-1:1 \Rightarrow 10 \ 9 \ 8 \ 7 \ \dots \ 1$

matlab@stoianov.it

37

## Riferimenti

Se  $M$  è una matrice 2-D,

- $M(i, j)$  il valore posizionato alla riga  $i$  e colonna  $j$
- $M(i, :)$  la riga  $i$
- $M(:, i)$  la colonna  $j$
- $M(:, :)$  l'intera matrice  $M$
  
- $M(i, j:end)$  la sotto-riga  $i$  che inizia a posizione  $j$
- $M(i, j:end-1)$  --/-- ... e finisce appena prima del fine
- $M(i, end+1)=x$  (a) aggiunge una colonna;  
(b) assegna il valore  $x$  alla cella  $(i, end+1)$
- $M(:, j:end)$  tutte le colonne di  $M$ , partendo da  $j^{th}$
- $M([i_1:i_2 1], j_1:j_2)$  una certa sottomatrice

matlab@stoianov.it

38

## Operazioni di base

'  
trasposizione  
 $[1, 5, 7]'$  ==  $\begin{matrix} 1 \\ 5 \\ 7 \end{matrix}$

Si nota:  $A'' = A$

$A(:)$

**size**(x) dimensioni della matrice  $x$   
**size**(x,i) la grandezza della dimensione  $i$  nella matrice  $x$   
**length**(x) la grandezza della dimensione più numerosa

**[A B]** concatenare le colonne di due matrici con lo stesso numero di righe  
**[A;B]** concatenare le righe di due matrici con lo stesso numero di colonne

matlab@stoianov.it

39

## Matrici speciali

<code>zeros(n)</code>	M(1:n,1:n) tutti i valori 0
<code>zeros(m,n)</code>	M(1:m,1:n)
<code>ones(n)</code>	M(1:n,1:n) tutti i valori 1
<code>ones(m,n)</code>	M(1:m,1:n)
<code>eye(n)</code>	M(1:n,1:n) i valori sul diagonale = 1, il resto = 0
<code>rand(n)</code>	R(1:n,1:n) con valori casuali, tra 0 e 1
<code>rand(m,n)</code>	R(1:m,1:n)

matlab@stoianov.it

40

## Matrici triangolari

- `triu(X,k)` matrice **superiore** triangolare, estratta da X, partendo da ( il diagonale principale + k )

```
>> triu(ones(4),1)
```

```
0 1 1 1
0 0 1 1
0 0 0 1
0 0 0 0
```

- `tril(X,k)` matrice **inferiore** triangolare, estratta da X, partendo da ( il diagonale principale + k )

matlab@stoianov.it

41

## Calcoli matriciali

**Matrice-A** {+, -, \*, /} **costante:** (ad es.,  $A5 = A*5$ )

- l'operazione si applica su tutti i valori nella matrice A

**Matrice-A** {+, -, \*} **Matrice-B** (ad es.,  $C = A*B$ )

- operazioni **matriciali algebrici**

- il risultato ha dimensioni secondo il tipo dell'operazione

**Matrice-A** {\*, /} **Matrice-B** (ad es.,  $A = A.*B$ )

- operazioni **"element-wise"**

- le dimensioni delle due matrici devono essere uguali

- il risultato ha dimensioni uguali a quelle delle matrici

matlab@stoianov.it

42

## Matrice-Scalare

```
A=rand(m,n);
c=rand;
R=A*c
R=zeros(m,n);
for i = 1:m
    for j = 1:n
        R(i,j) = A(i,j)*c;
    end
end
```

```
A=rand(m,n);
c=rand;
R=A+c
R=zeros(m,n);
for i = 1:m
    for j = 1:n
        R(i,j) = A(i,j)+c;
    end
end
```

matlab@stoianov.it

43

# Algebra Matriciale

```
A=rand(m,k);
B=rand(k,n);
R=zeros(m,n);
R=A*B
for i = 1:m
    for j = 1:n
        S=0;
        for l=1:k,
            S = S + A(i,l)*B(l,j);
        end
        R(i,j)=S;
    end
end
```

```
A=rand(m,n);
B=rand(m,n);
R=zeros(m,n);
R=A+B
for i = 1:m
    for j = 1:n
        R(i,j) = A(i,j)+B(i,j);
    end
end
```

matlab@stoianov.it

44

# Element-wise

```
A=rand(m,n);
B=rand(m,n);
R=zeros(m,n);
R=A.*B
for i = 1:m
    for j = 1:n
        R(i,j) = A(i,j)*B(i,j);
    end
end
```

```
A=rand(m,n);
B=rand(m,n);
R=zeros(m,n);
R=A+B
for i = 1:m
    for j = 1:n
        R(i,j) = A(i,j)+B(i,j);
    end
end
```

matlab@stoianov.it

45



## Algebra matriciale - Esempi

### MULTIPLICAZIONE ALGEBRICA

- $(1:10) \cdot (10:-1:1) = [10 \ 18 \ 24 \ 28 \ 30 \ 28 \ 24 \ 18 \ 10]$
- $(1:10) \cdot (10:-1:1)' = 220$
- $(1:10)' \cdot (10:-1:1) = \begin{bmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 20 & 18 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 \\ \dots & & & & & & & & & \\ 100 & 90 & 80 & & & & & & & 10 \end{bmatrix}$

matlab@stoianov.it

46

## Matrici – Esempi (2)

AllTeens = `ones(10)*10`  
AllTeens\_again = `zeros(10)+10`  
  
RandomDiagonal = `eye(10).*rand(10)`  
ZeroDiagonal = `~eye(10)`

matlab@stoianov.it

47

## Statistica descrittiva

V: un vettore

- `mean(V)`
- `median(V)`
- `min(V)`
- `max(V)`
- `std(V)`
- `var(V)`

Su matrici 2D, l'operazione viene eseguita per **ciascuna colonna**

matlab@stoianov.it

48

## Indici

- **Indice**: un vettore utilizzato per riferire gli elementi di una matrice
- Un strumento importante nell'elaborazione di dati matriciali.
- Serve per elaborazione disomogenea di matrici
- $I = \text{find}(\text{operazione logica su una matrice})$   
riporta l'indice degli elementi per quali il risultato è vero.

```
V=rand(10,1);  
I=find(V>0.9); % Creare un indice  
V(I)=0         % Usare l'indice come riferimento
```

matlab@stoianov.it

49

## Indici - Esempi

```
M=rand(10,3);

Dist=abs(M(:,1)-0.5); %Vettore con distanze da 0.5

I = find(Dist>0.4); % Creare un indice
M(I,2:end)=0 % Utilizzare l'indice

I = find(Dist<0.4); % Un altro indice
RT=mean(M(I,1)); % Utilizzo
```

matlab@stoianov.it

50

## Indici – Applicazioni per Statistica Descrittiva

```
x=randn(30,1)*20+500; % (Creare) Osservazioni
m=mean(x); % media di tutti i dati
s=std(x); % dev.st di tutti i dati
I=abs(x-m)<=2.0*s; % Trovare non-outliers

m1=mean(x(I)); % media dei dati "puliti"
s1=std(x(I)); % dev.st dei tutti puliti
e1=s1/sqrt(length(x)); % st.err

fprintf('Mean=%.2f; Std=%.2f, mRT=%.2f\n',m,s,m1);
I1=abs(x-m)>1.5*s; % Trovare outliers
disp([find(I1) x(I1)]); % Visualizzare loro indici e valori
```

matlab@stoianov.it

51

## Matrici - Esercizi

### Compito:

- (i) produrre 100 numeri casuali (0...1)
- (ii) calcolare il numero dei valori maggiori di 0.52
- (ii) replicare (i/ii) 30 volte, mantenendo il risultato in un vettore **D**
- (iii) per le osservazioni in **D**, riportare la *media*, *std. err.*, e la probabilità che la media è diversa da 50.

Tool: `tcdf(t,n-1)` la probabilità che una osservazione nella distribuzione di Student  $(n-1) < t$

```
D=sum(rand(100,30)>0.52);  
% 1. matrice casuale con 100 righe e 30 colonne  
% 2. confronto element-wise con 0.5  
% 3. somma per colonne  
m=mean(D); % Media  
e=std(D)/sqrt(30); % Std. Error  
t=abs(m-50)/e; % Media standardizzata (T)  
p=2-2*tcdf(t,30-1); % p(m!=50)  
fprintf('rnd>0.52: %.2f(%.2f,t=%.2f,p=%.3f)',m,e,t,p);
```

matlab@stoianov.it

52

## Matrici – Esercizi (2)

**Compito:** Calcolare la superficie totale di 10 dischi, il radio di ciascuna di quale è dato in un vettore.

```
R=rand(10,1)*10;  
S=sum(calcsurf('d',R));  
fprintf('Surface: %.1f \n',S);
```

```
function s = calcsurf2(type, r)  
switch type  
case 'd', s = pi * r.^2;  
case 's', s = (2*r).^2;  
otherwise, s=0; disp('Wrong type');  
end
```

File  
calcsurf2.m

matlab@stoianov.it

53