

Informatica B

Esercitazione 11 (Soluzioni)

13 dicembre 2021

Strutture

11.1 Scrivere uno **script MATLAB** che gestisca una lista di studenti e i rispettivi voti. Lo script permette all'utente di inserire un numero di studenti. Ciascun studente ha un nome, una lista di 5 voti (compresi tra 18 e 30) e la classe corrispettiva (Classe A o B). Il programma deve ordinare i dati dei studenti rispetto al voto medio (ordine decrescente) e stampare i dati dei studenti in ordine. Inoltre, salvare i dati dei studenti in ordine, separato per classe, in un file nominato 'dati_studenti.mat' in due variabili, studenti_a e studenti_b.

```
1 clear all;
2 close all;
3 clc;
4 n = input("Quanti studenti volete inserire");
5 studenti = [];
6 for ii=1:n
7     fprintf("Inserimento studente %d\n", ii);
8     nome = input("Inserire nome:", 's');
9     studenti(ii).nome = nome;
10    for jj=1:5
11        voto = 0;
12        while voto <18 || voto > 30
13            fprintf("Inserire Voto %d", jj)
14            voto = input(":");
15        end
16        studenti(ii).voti(jj) = voto;
17    end
18    classe = input('Classe A o B? (a/b)\n', 's');
19    switch classe
20        case 'a'
21            studenti(ii).classe = 'a';
22        case 'b'
23            studenti(ii).classe = 'b';
```

```

24         otherwise
25             studenti(ii).classe = 'a';
26     end
27 end
28
29 voti = [studenti.voti]; % voti e un array di lunghezza n
      *5
30 voti = reshape(voti,n,5)'; % voti e una matrice di
      dimensini (n,5)
31 voti_medi = mean(voti,2); % media delle righe della
      matrice voti
32 [voti_medi_ordinati, idx] = sort(voti_medi, 'descend');
33 studenti = studenti(idx);
34
35 fprintf("Studenti Ordinati:");
36 for ii = 1:n
37     disp([studenti(ii).nome, ' ( Classe ', studenti(ii).
      classe, ...
38         ') ha voto medio' , num2str(voti_medi_ordinati(ii
      ))])
39 end
40 % importante estrarre la classe dopo avere ordinato per
      voto medio
41 % cosi i studenti sono ordinati anche dentro la singola
      classe
42 classi = [studenti.classe];
43 classe_a_idx = find(classi == 'a');
44 classe_b_idx = find(classi == 'b');
45
46 studenti_a = studenti(classe_a_idx);
47 studenti_b = studenti(classe_b_idx);
48
49 save('dati_studenti', 'studenti_a', 'studenti_b');

```

11.2 Scrivere uno **script MATLAB** che gestisca un sistema di raccomandazione di ristoranti. Lo script permette all'utente di inserire un grande numero di ristoranti (fino a 1000). Ciascun ristorante ha un nome, un indice di qualità (da 1 a 5) e un flag che specifica se il ristorante è economico. Ad ogni nuovo inserimento, il sistema deve stampare l'elenco dei nomi dei ristoranti economici che hanno un indice di qualità maggiore o uguale della media di tutti i ristoranti memorizzati.

```

1 % Es. 11.2
2 % Ristoranti
3
4 close all
5 clear
6 clc
7
8 max_dati = 1000;
9 %Inizializzazione generica (non necessaria)
10 ristoranti = [];
11
12 for ii = 1:max_dati
13     %Inserimento ristorante
14     fprintf('\nInserimento nuovo ristorante\n');
15     %Inserisci nome
16     nome = input('Inserisci nome\n', 's');
17     ristoranti(ii).nome = nome;
18     %Inserisci voto
19     voto = 0;
20     while(voto < 1 || voto > 5)
21         voto = input('Inserisci voto (1-5)\n');
22     end
23     ristoranti(ii).voto = voto;
24     %Inserisci economico
25     eco = input('Economico? (s/n)\n', 's');
26     switch eco
27         case 's'
28             ristoranti(ii).economico = true;
29         case 'n'
30             ristoranti(ii).economico = false;
31         otherwise
32             ristoranti(ii).economico = false;
33     end
34
35     %Consiglia qualita/prezzo elevata
36     %Calcola media
37     media = mean([ristoranti.voto]);
38     %Estrai indici

```

```

39     idx = find([ristoranti.voto] >= media & [ristoranti.
              economico]);
40     fprintf('\n');
41     %Stampa nomi
42     for jj = idx
43         fprintf('Consiglio "%s"!\n', ristoranti(jj).nome)
              ;
44     end
45     if isempty(idx)
46         fprintf('Non ho consigli!\n');
47     end
48 end
49
50 fprintf('Memoria esaurita\n');

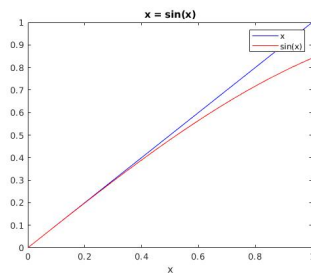
```

Grafici 2D

11.3 Visualizzare i grafici $y = x$ e $y = \sin(x)$ nel piano cartesiano. In particolare, il grafico deve mostrare (vedi Figura 1):

- La funzione $y = x$;
- La funzione $y = \sin(x)$ (sullo stesso grafico);
- *Bonus*: mostrare graficamente che per angoli piccoli le funzioni sono simili.

Figura 1: $x = \sin(x)$



```
1 % Es. 11.3
2 % sin(x)
3
4 close all %chiude tutte le figure
5 clear
6 clc
7
8 %Ascisse
9 x = linspace(0, 1, 300);
10 %Ordinate di sin(x)
11 y = sin(x);
12
13 figure; %apri figura
14 plot(x, x, 'b'); %il terzo argomento specifica il colore
15 hold on %conserva plot precedenti
16 plot(x, y, 'r');
17 %Decorazioni
18 xlabel('x');
19 legend('x', 'sin(x)');
20 title('x = sin(x)');
21 hold off;
22 figure(2); % crea nuova figura
23 %Trova gli indici di x per cui x e' simile a sin(x)
```

```
24 diff = abs(y - x);
25 idx = find(diff < 0.0001);
26 %disegna solo gli indici vicini
27 plot(x(idx), x(idx), 'b');
28 hold on;
29 plot(x(idx), y(idx), 'r');
30
31 %Decorazioni
32 xlabel('x');
33 legend('x', 'sin(x)');
34 title('Angoli piccoli');
```

11.4 Scrivere uno script matlab che disegna la funzione $f(x) = x^2 - \sin(x)$ nell'intervallo [0,2]. In piu, trovi approssimamente il valore di minimo di questa funzione e lo disegni come un punto rosso nello stesso grafico

```
1 % Es. 11.4
2 % Minimo
3
4 close all
5 clear
6 clc
7
8 % Parametro
9 x = linspace(0, 2, 1000);
10 % ordinate in funzione del parametro
11 y = x.^ 2 - sin(x);
12
13
14 figure;
15 plot(x, y);
16 xlabel('x');
17 ylabel('y');
18 title('Minimo');
19
20 % trova il valore minimo tra quelli campionati
21 [min_value, min_index] = min(y);
22 min_x = x(min_index);
23
24 hold on;
25 scatter([min_x], [min_value], 'red', 'filled');
26 legend('x^2 - sin(x)', '(x_{min}, y_{min})');
```

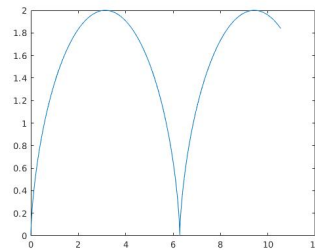
11.5 La cicloide (vedi Figura 2) è la curva tracciata nell'aria da un punto situato sul bordo di una ruota in movimento. Disegnare la cicloide generata da un disco di raggio $r = 1$.

Suggerimento: la curva è descritta dalle seguenti equazioni:

$$\begin{aligned}x &= r * (t - \sin(t)), \\y &= r * (1 - \cos(t)),\end{aligned}$$

dove t è un parametro che varia, ad esempio, da 0 a 10.

Figura 2: cicloide

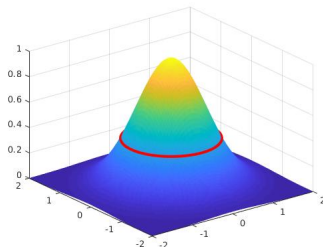


```
1 % Es. 11.5
2 % Cicloide
3
4 close all
5 clear
6 clc
7
8 % Parametro
9 t = linspace(0, 10, 1000);
10 r = 1;
11
12 %Ascisse e ordinate in funzione del parametro
13 x = r*(t - sin(t));
14 y = r*(1 - cos(t));
15
16 figure;
17 plot(x, y);
18 xlabel('x');
19 ylabel('y');
20 title('Cicloide');
```


Grafici 3D

11.6 Disegnare la superficie $f(x, y) = e^{-x^2-y^2}$. Disegnare anche un cerchio di raggio 1 situato interamente su questa superficie, come in Figura 3.

Figura 3: $f(x, y) = e^{-x^2-y^2}$



```
1 function z = bell(x, y)
2 %Campana (funzione di due variabili)
3     z = exp(-x.^2 - y.^2);
4 end

1 % Es. 11.6
2 % Campana
3
4 close all
5 clear
6 clc
7
8 %Ascisse
9 x = linspace(-2, 2, 100);
10 %Ordinate
11 y = linspace(-2, 2, 100);
12
13 %Griglia di coppie (x,y)
14 [xx, yy] = meshgrid(x, y);
15
16 %Quote, funzione di ascisse e ordinate
17 zz = bell(xx, yy);
18
19 %Superficie
20 grafico_1 = surf(xx, yy, zz);
21
22 hold on;
23 r = 1;
24 %Parametro (angolo)
```

```
25 t = linspace(0, 2*pi, 100);
26 %Curva nello spazio
27 grafico_2 = plot3(cos(t), sin(t), fun(cos(t), sin(t)), 'r
    ');
28
29 %Decorazioni
30 set(grafico_1, 'Edgecolor', 'none');
31 set(grafico_2, 'LineWidth', 3);
32 xlabel('x');
33 ylabel('y');
34 zlabel('z');
35 title('Campana');
```