



Plot e Riepilogo Matlab

Loris Giulivi – Nicolò Folloni

Il comando plot

La function *plot* permette di disegnare un grafico date le coordinate di ascisse ed ordinate.

```
>> help plot
plot Linear plot.
plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
then the vector is plotted versus the rows or columns of the matrix,
whichever line up. If X is a scalar and Y is a vector, disconnected
line objects are created and plotted as discrete points vertically at
X.

plot(Y) plots the columns of Y versus their index.
If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)).
In all other uses of plot, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with
plot(X,Y,S) where S is a character string made from one element
from any or all the following 3 columns:

      b    blue      .    point      -    solid
      g    green     o    circle     :    dotted
      r    red       x    x-mark    -.   dashdot
      c    cyan      +    plus      --   dashed
      m    magenta   *    star      (none) no line
      y    yellow    s    square
      k    black     d    diamond
      w    white     v    triangle (down)
                        ^    triangle (up)
                        <    triangle (left)
                        >    triangle (right)
                        p    pentagram
                        h    hexagram

For example, plot(X,Y,'c+:') plots a cyan dotted line with a plus
at each data point; plot(X,Y,'bd') plots blue diamond at each data
point but does not draw any line.
```

Il comando plot

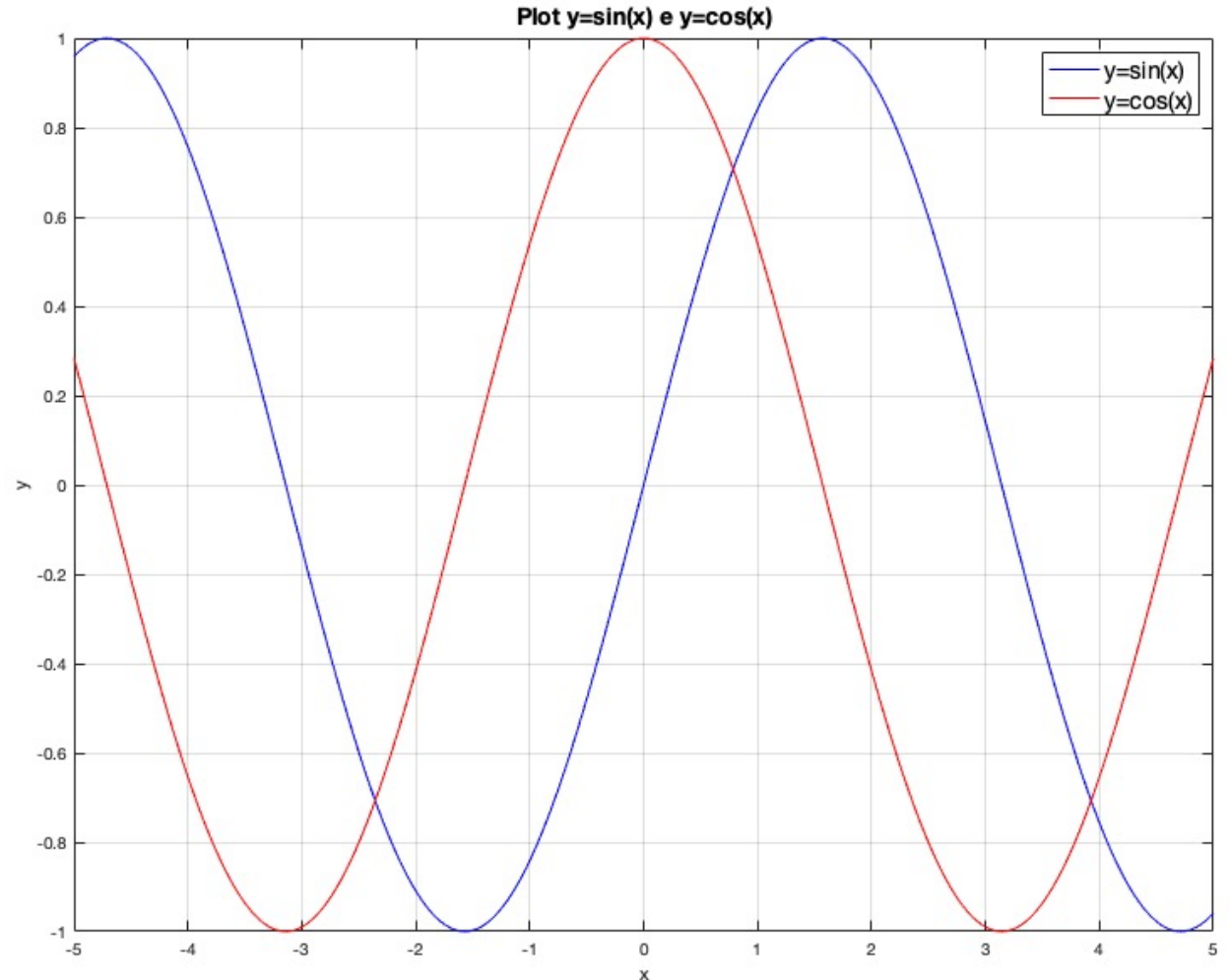
- $xx = \text{linspace}(a, b, N)$ restituisce un vettore xx di N punti ottenuti dividendo in modo uniforme l'intervallo $[a,b]$
- figure permette di aprire una nuova figura. Se si vogliono plottare più grafici, anteporre all' n -esimo plot il comando $\text{figure}(n)$
- hold on permette di sovrapporre due plot differenti
- hold off usato dopo hold on ne interrompe l'effetto
- $\text{title}(\text{char})$ per aggiungere un titolo al grafico
- $\text{xlabel}(\text{char})$ definisce il nome della variabile sulle ascisse
- $\text{ylabel}(\text{char})$ definisce il nome della variabile sulle ordinate
- $\text{legend}(\text{char}_1, \dots, \text{char}_n)$ nel caso di plot multipli, assegna un'etichetta a ciascuna funzione plottata
- $\text{xlim}([a, b])$ stabilisce il range delle ascisse
- $\text{ylim}([a, b])$ stabilisce il range delle ordinate
- grid on aggiunge una griglia di sfondo al plot

Esempio

Plottiamo la funzione $y = \sin(x)$ e la funzione $y = \cos(x)$

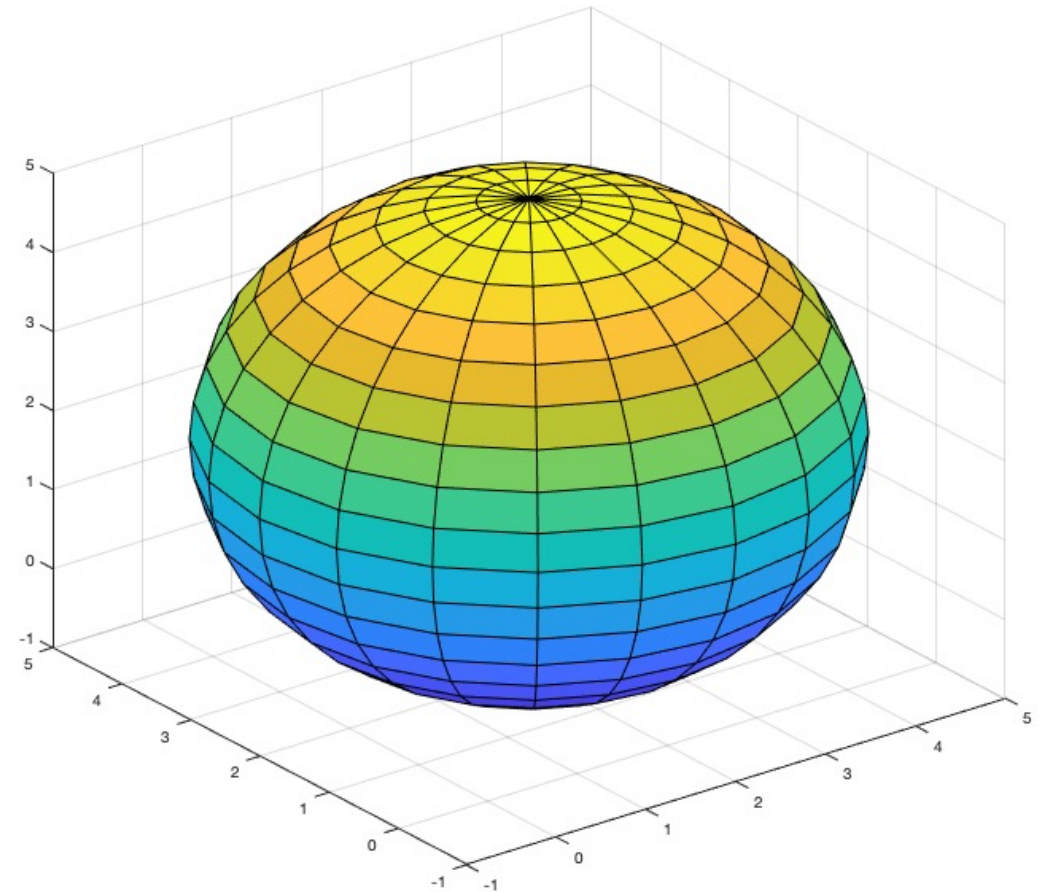
```
close all
clear all
clc

xx = linspace(-5,5,200);
y1 = sin(xx);
y2 = cos(xx);
plot(xx, y1, 'b');
hold on
plot(xx, y2, 'r');
title('Plot y=sin(x) e y=cos(x)')
xlabel('x')
ylabel('y')
legend('y=cos(x)', 'y=sin(x)')
grid on
```



I diversi tipi di plot

- `plot(xx, yy)`: plot della funzione F definita da $yy = F(xx)$ nel piano 2-dimensionale
- `plot3(xx, yy, zz)`: plot della funzione F definita da $zz = F(xx, yy)$ nello spazio 3-dimensionale
- `scatter(xx, yy)`: plot dei punti (xx, yy) nel piano 2-dimensionale
- `scatter3(xx, yy, zz)`: plot dei punti (xx, yy, zz) nel piano 3-dimensionale
- `surf(xx, yy, zz)`: plot della superficie definita da $zz = F(xx, yy)$
- `rectangle('Position', [x, y, base, altezza])`: plot del rettangolo centrato in (x,y) con base e altezza date
- `[xx, yy, zz] = sphere`: genera 3 vettori da plottare con `surf` per ottenere una sfera unitaria centrata nell'origine



Esercizio 1

ES1: Rappresentare graficamente in un unico plot una parabola passante per l'origine e una retta avente coefficiente angolare $m = 2$. In un secondo plot, rappresentare le due funzioni considerate in valore assoluto. In seguito, rappresentare in un terzo plot le rispettive derivate. I plot devono contenere: titolo, legenda, griglia, nomi degli assi cartesiani

Esercizio 1 - Soluzione

```
xx = linspace(-10, 10);
yr = 2.*xx + 4;
yp = 2.*(xx.^2) + 3.*xx;

yr_abs = abs(yr);
yp_abs = abs(yp);

yr_der = ones(1,100)*2;
yp_der = 4.*xx + 3;

figure(1)
plot(xx, yr, xx, yp)
grid on
title('Funzioni')
legend('y = 2x + 4', 'y = 2x^2 + 3x')

figure(2)
plot(xx, yr_abs, xx, yp_abs)
grid on
title('Funzioni in valore assoluto')
legend('y = |2x + 4|', 'y = |2x^2 + 3x|')

figure(3)
plot(xx, yr_der, xx, yp_der)
grid on
title('Derivata di funzioni')
legend('y = 2', 'y = 4x + 3')
```

Esercizio 2

ES2: Considerare N punti nello spazio 2 dimensionale con coordinate comprese tra -1 e 1 .

- Scrivere una function che ricevuti in ingresso N punti restituisce un vettore N dimensionale contenete la norma di ciascun punto.
- Stabilire quanti punti cadono all'interno di un cerchio di raggio 0.5 centrato nell'origine
- Stabilire il raggio R minimo affinché un cerchio di raggio R centrato nell'origine contenga tutti i punti
- Determinare il baricentro geometrico dell'insieme di punti considerato
- Plottare i punti, il baricentro e i cerchi nel piano 2d
- Adattare lo script anche al caso 3 dimensionale

Esercizio 2 - Soluzione

```
N = input('Quanto vale N? ');
d = input('Quanto vale d? ');
Punti = rand(d,N) - rand(d,N);

norme = Norme(Punti);

R = 0.5;
inside = sum(norme<R);
outside = N - inside;

Rmin = max(norme);

bar = mean(Punti, 2);

switch(d)
    case 2
        scatter(Punti(1,:), Punti(2,:))
        hold on
        scatter(bar(1), bar(2))
        hold on
        t = linspace(0,2*pi);
        x = R*cos(t);
        y = R*sin(t);
        plot(x,y);

    case 3
        [x, y, z] = sphere;
        surf(x*R, y*R, z*R)
        alpha 0.1
        hold on
        scatter3(Punti(1,:), Punti(2,:), Punti(3,:), 'filled')
        hold on
        scatter3(bar(1), bar(2), bar(3), 'filled')
    otherwise
        disp('d > 3, non si può plottare');
end
```

```
function [n] = Norme(Punti)

n = sqrt(sum(Punti.^2,1));

end
```