



Funzioni Grafiche

Informatica, AA 2021/2022

Francesco Trovò

12 Novembre 2021

<https://trovo.faculty.polimi.it/>

francesco1.trovo@polimi.it



Funzioni Grafiche

Funzione	Scopo
figure(figNumber)	apre una figura identificata dall'handle figNumber . Se non presente definisce l'handle in maniera incrementale
hold	+ on / off definisce se tenere o cancellare il grafico attualmente presente nella figura alla prossima operazione di visualizzazione sulla figura.
plot(x,y)	disegna in un riferimento cartesiano 2D le coppie di punti identificati da (x(1),y(1))... (x(end) , y(end)). x ed y devono avere la stessa lunghezza
plot3(x,y,z)	disegna in un riferimento cartesiano 3D le coppie di punti identificati da (x(1),y(1),z(1))... (x(end) , y(end), z(end)). x,y e z devono avere la stessa lunghezza
plot(x,y, frmStr)	frmStr specifica il marker ed il colore usato nella visualizzazione dei punti
imagesc(A)	Visualizza la matrice A come un'immagine in colormap di default. Ogni pixel viene ridimensionato per migliorare la visualizzazione
imshow(A)	visualizza un'immagine A in scale di grigio (se A è di dimensione 2) o a colori nello spazio RGB (se A è di dimensione 3)
legend(titles)	Visualizza la legenda, usando le stringhe in titles



Plot



Diagrammi a due dimensioni

La funzione `plot(x, y)` disegna il **diagramma** cartesiano dei punti che hanno valori delle ascisse nel vettore `x`, delle ordinate nel vettore `y`

Il diagramma è l'insieme di **coppie di punti** `[x(1), y(1)], ..., [x(end), y(end)]` rappresentanti le coordinate dei punti del piano cartesiano

La funzione `plot` congiunge i punti con una linea, per dare continuità al grafico

In `plot(x, y)`, `x` e `y` devono essere **due vettori aventi le stesse dimensioni**

E' possibile specificare diversi elementi grafici (`help plot` per una lista delle opzioni)

Le funzioni `xlabel` visualizzano una stringa come nome asse ascisse, `ylabel` per ordinate, `title` per il titolo



Esercizio

Scrivere una funzione che prende in ingresso due coefficienti m, q ed un vettore di punti xx e restituisce il vettore yy dei punti che stanno sulla retta $y = mx + q$ in corrispondenza a xx

Si invochi la funzione e si disegni la retta nell'intervallo xx



Esercizio

Scrivere una funzione che prende in ingresso due coefficienti m, q ed un vettore di punti xx e restituisce il vettore yy dei punti che stanno sulla retta $y = mx + q$ in corrispondenza a xx

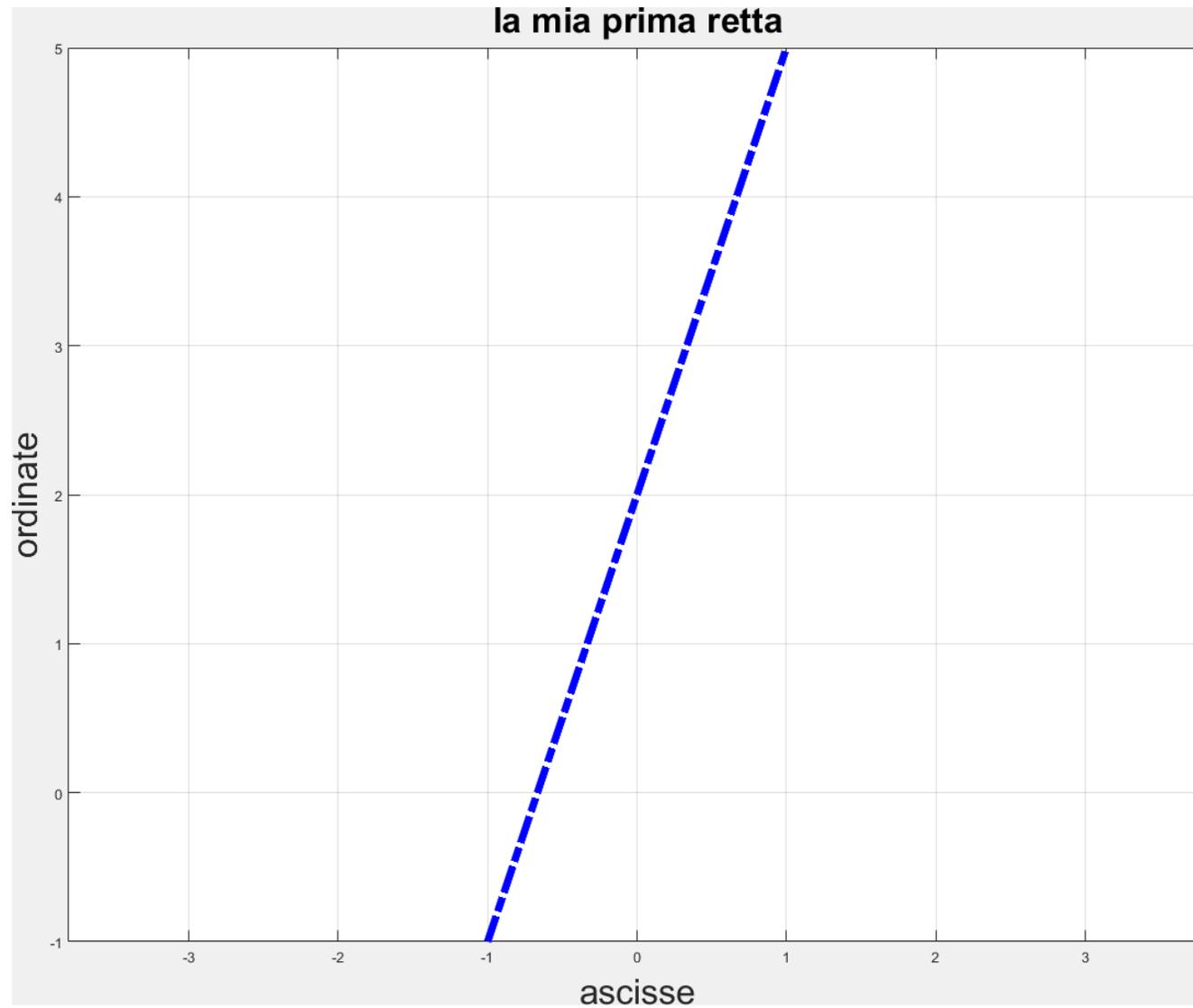
```
function [yy] = retta(m, q, xx)
    yy = m * xx + q;

% for ii = 1 : length(xx)
%     yy(ii) = m * xx(ii) + q;
% end
```



Esempi di script per invocare la funzione

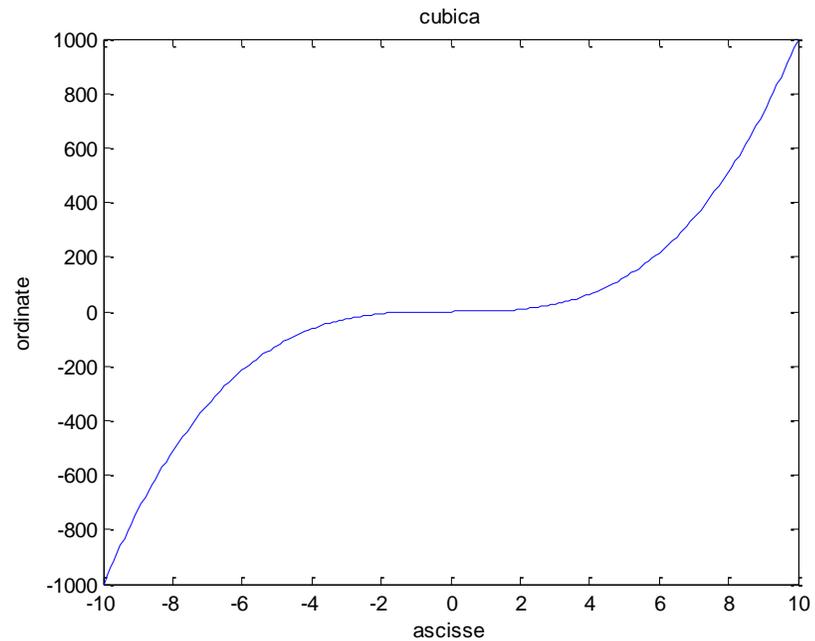
```
x = [-1 : 0.1 : 1];  
% invoco la funzione per plottare  $y = 3x + 2$   
y = retta(3,2,x)  
figure  
plot(x,y, 'b*') % disegno con le stelline  
axis equal % assi della stessa dimensione  
plot(x,y, 'b-'), %disegno con una retta  
grid on % aggingo aggiungo la griglia  
plot(x,y, 'b-', 'LineWidth', 3), axis equal, grid on  
plot(x,y, 'b--', 'LineWidth', 5), axis equal, grid on  
plot(x,y, 'b-.', 'LineWidth', 5), axis equal, grid on  
title('la mia prima retta', 'FontSize', 24)  
xlabel('ascisse', 'FontSize', 24)  
ylabel('ordinate', 'FontSize', 24)
```



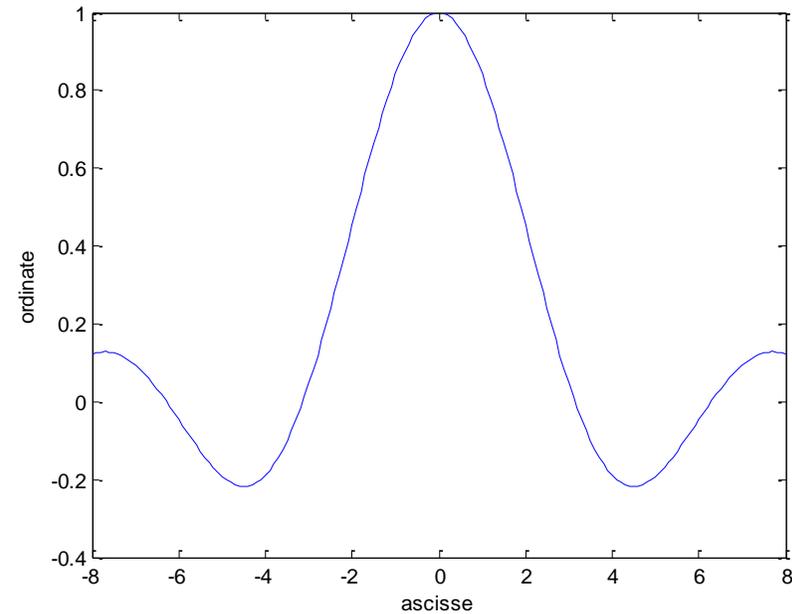


Diagrammi a due dimensioni: esempi

```
>> x = -10:0.1:10;  
>> y=x.^3;  
>> plot(x,y);  
>> xlabel('ascisse');  
>> ylabel('ordinate');  
>> title('cubica');
```



```
>> x=[-8:0.1:8];  
>> y= sin (x) ./ x;  
>> plot(x, y);  
>> xlabel('ascisse');  
>> ylabel('ordinate');
```





Esempi

Definire una funzione *samplePolynomial* che prende in ingresso

- un vettore di coefficienti C
- un vettore che definisce un intervallo $[a,b]$

e restituisce un due vettori di 100 punti xx ed yy contenente i punti della del polinomio

$$y = C(1)x^{n-1} + C(2)x^{n-2} + \dots + C(n-1)x^1 + C(n)$$

Utilizzare *samplePolynomial* per calcolare i punti delle seguenti curve (in un intervallo $[-10, 10]$) e visualizzarlo:

$$y = x - 1;$$

$$y = 2x^2 + x - 12;$$

$$y = -0.1x^3 + 2x^2 - 10x - 12$$

visualizzare, per ogni valore di x , la curva maggiore



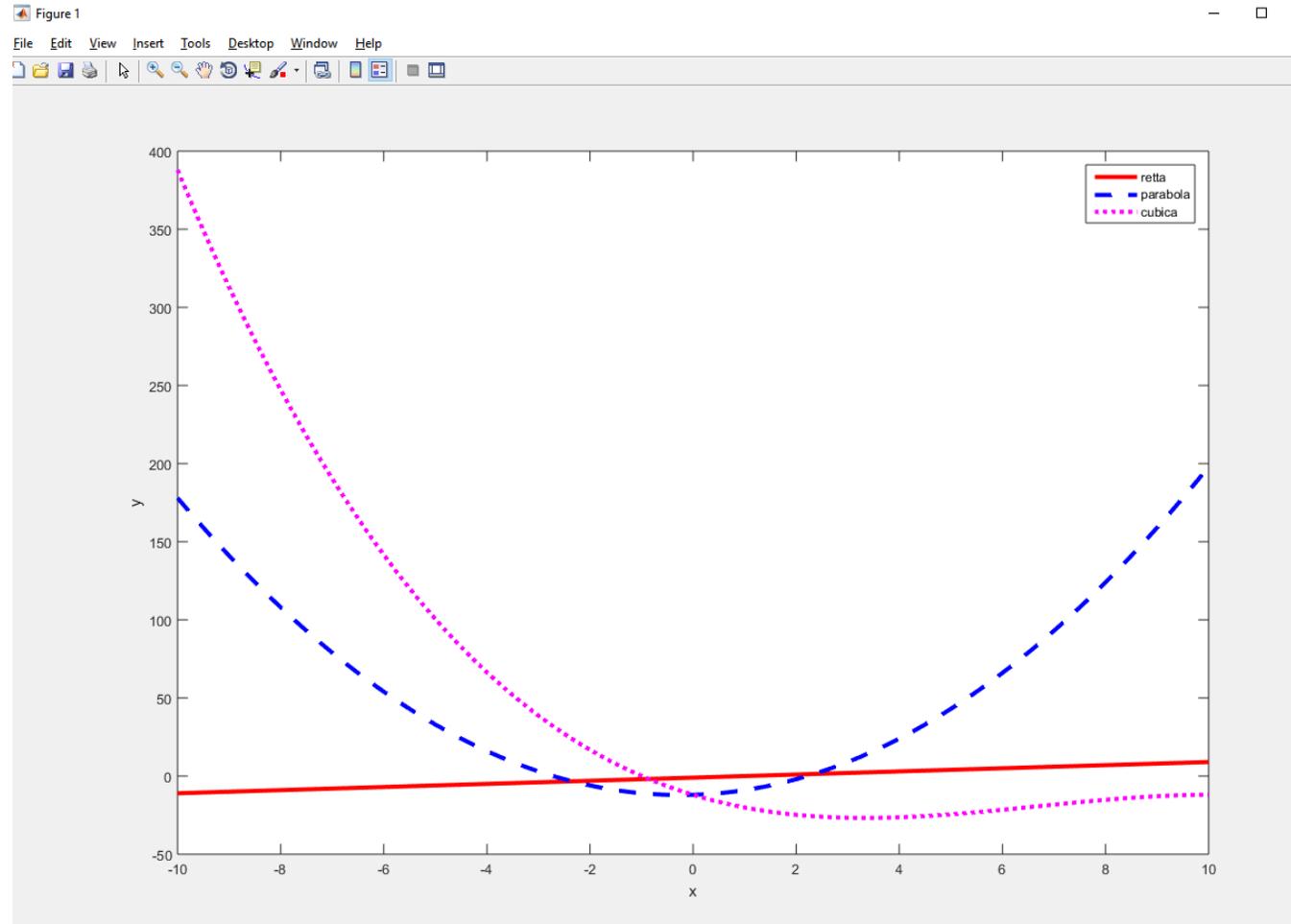
Invocazione della Funzione e Plot

```
interval = [-10 , 10];
rettaCoeffs = [1 , -1];
parabolaCoeffs = [ 2 , 1 , -12] ;
cubicaCoeffs = [-0.1 , 2 , -10 , -12];
% calcola i valori dei polinomi
[rx,ry] = samplePolynomial(rettaCoeffs , interval);
[px,py] = samplePolynomial(parabolaCoeffs , interval);
[cx,cy] = samplePolynomial(cubicaCoeffs, interval);

figure(1), plot(rx, ry, 'r-', 'LineWidth', 3)
hold on
plot(px, py, 'b--', 'LineWidth', 3)
plot(cx, cy, 'm:', 'LineWidth', 3)
hold off
legend('retta', 'parabola', 'cubica')
xlabel('x')
ylabel('y')
```

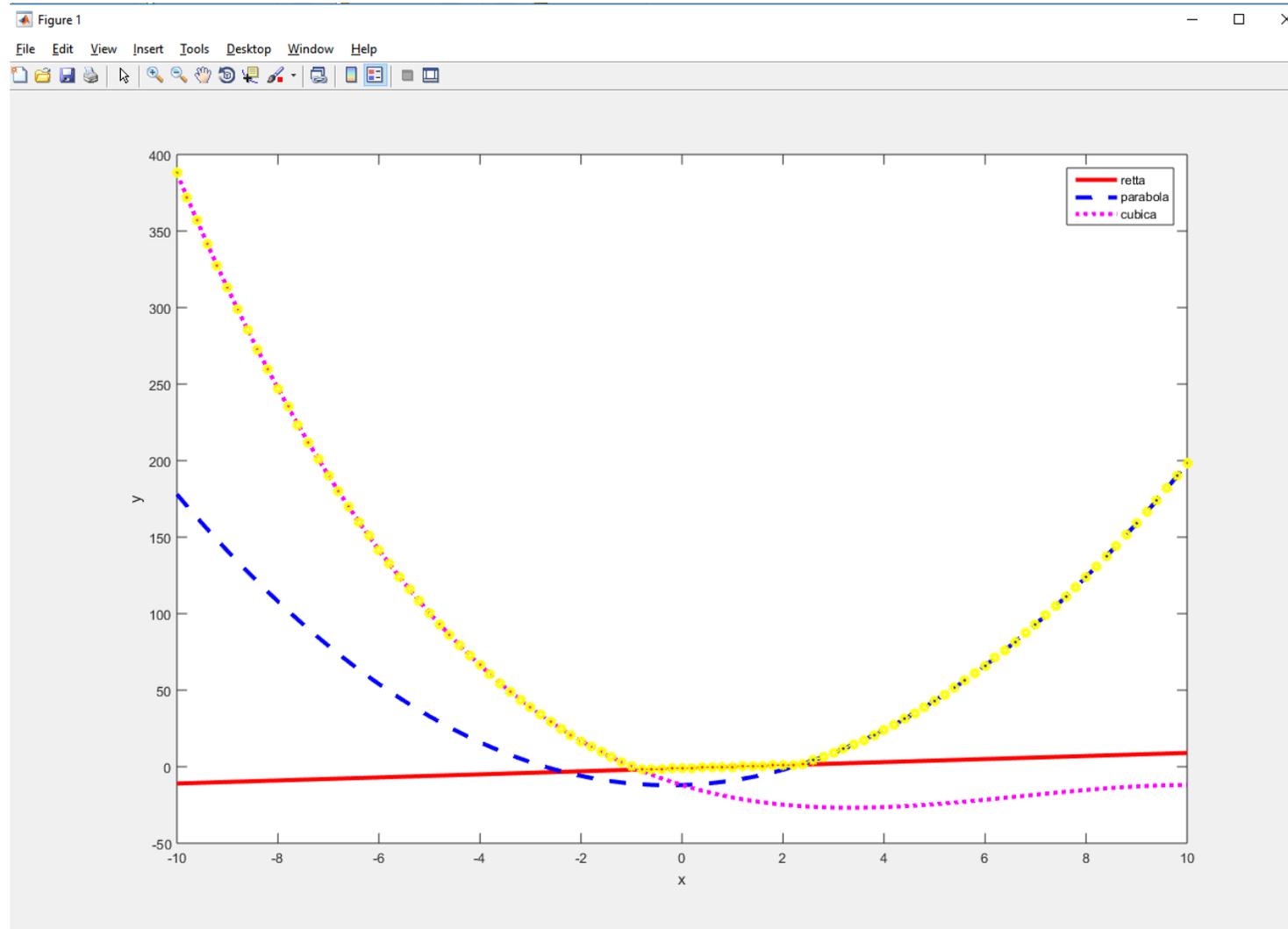


Risultato sul Grafico





Nuovo Risultato





Diagrammi a due dimensioni: altri esempi

Un diagramma è semplicemente una sequenza ordinata di punti, di coppie di coordinate cartesiane

In `plot(x, y)` non necessariamente `x` contiene valori equispaziati e `y` non è necessariamente funzione di `x`. Sia `x` che `y` possono essere, ad esempio, funzioni di qualche altro parametro.

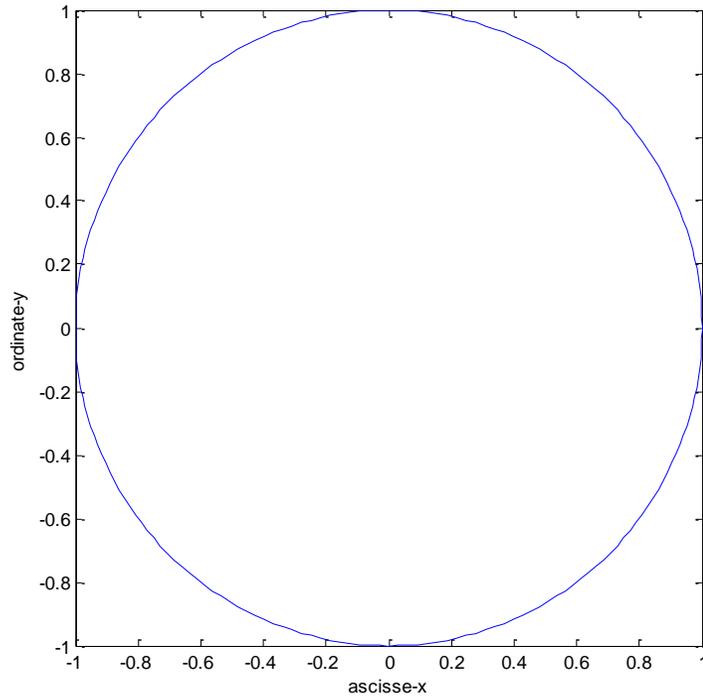
Che diagrammi disegnano i seguenti esempi?

```
>> t=[0:pi/100:2*pi];  
>> x=cos(t);  
>> y=sin(t);  
>> plot(x,y);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');
```

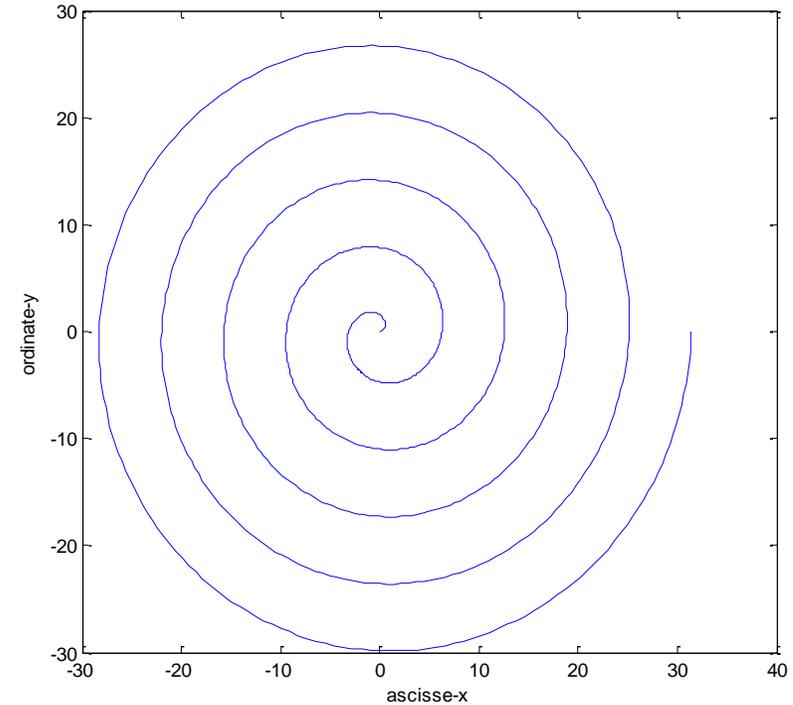
```
>> t=[0:pi/100:10*pi];  
>> x=t .* cos(t);  
>> y=t .* sin(t);  
>> plot(x,y);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');
```



Esempi



```
>> t=[0:pi/100:2*pi];  
>> x=cos(t);  
>> y=sin(t);  
>> plot(x,y);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');
```



```
>> t=[0:pi/100:10*pi];  
>> x=t .* cos(t);  
>> y=t .* sin(t);  
>> plot(x,y);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');
```



Plot 3D



Diagrammi lineari a tre dimensioni

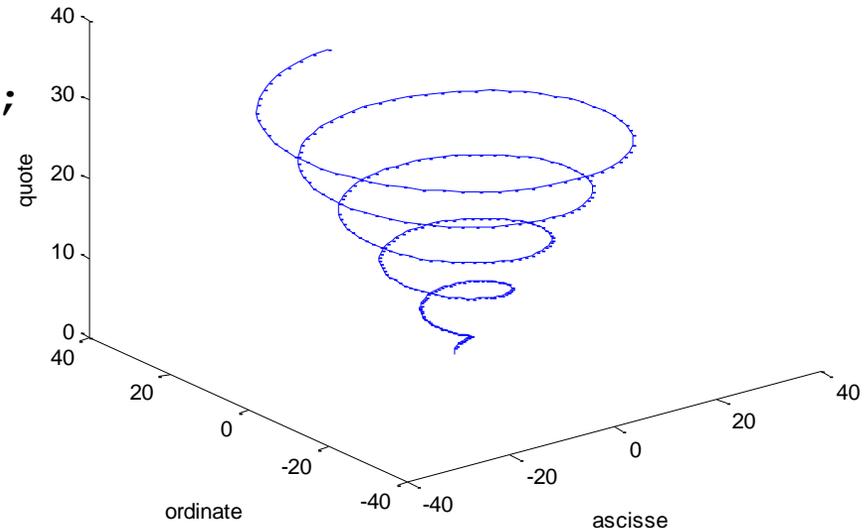
Generalizzazione del diagramma a due dimensioni: insieme di terne di coordinate

`plot3(x, y, z)` disegna un diagramma cartesiano con `x` come ascisse, `y` come ordinate e `z` come quote

funzioni `xlabel`, `ylabel`, `zlabel`, `title`

Esempio

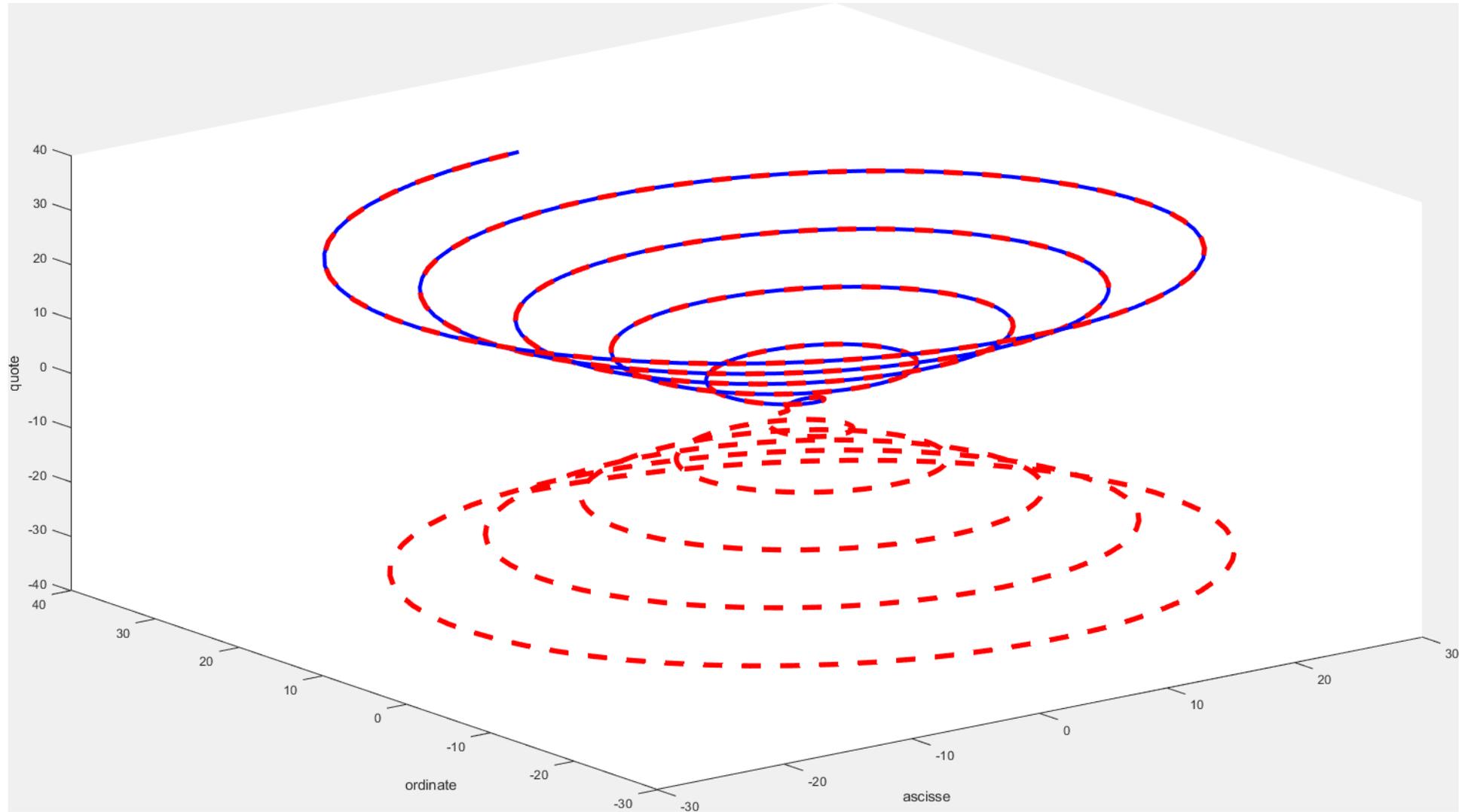
```
>> t = 0:0.1:10*pi;  
>> plot3 (t.*sin(t), t.*cos(t), t);  
>> xlabel('ascisse');  
>> ylabel('ordinate');  
>> zlabel('quote');
```





Cosa fa?

```
figure(2),  
t = 0: 0.1 : 10*pi;  
plot3(abs(t).*sin(t), abs(t).*cos(t), t, 'b-', 'LineWidth',  
3);  
hold on  
t = [-t(end : -1 : 1), t];  
plot3(abs(t).*sin(t), abs(t).*cos(t), t, 'r--', 'LineWidth',  
4);  
xlabel('ascisse');  
ylabel('ordinate');  
zlabel('quote');  
hold off
```





La funzione linspace

`linspace(a, b, n)`:

crea un vettore di n punti equispaziati tra a e b

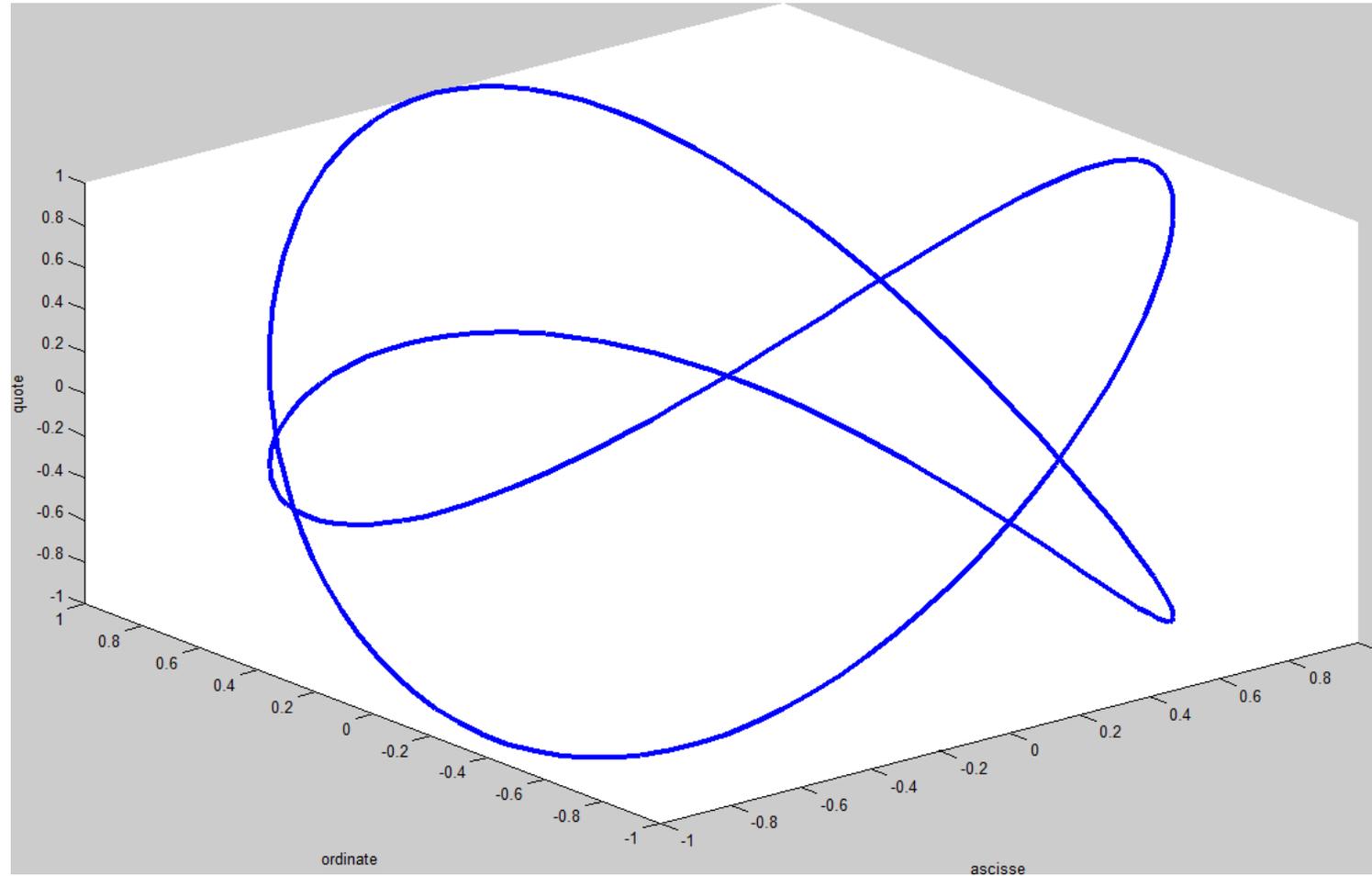
`plot` restituisce un handle, una variabile di riferimento per poter accedere nuovamente all'insieme di punti disegnato

`Set(plot_handle, 'Property Name', PropertyValue)`
permette di modificare

```
t = linspace(0,4*pi ,200);  
plot_hnd = plot3(sin(t),cos(t),cos(3/2 *t))  
set(plot_hnd, 'LineWidth', 3)  
xlabel('ascisse');  
ylabel('ordinate');  
zlabel('quote');
```



Risultato



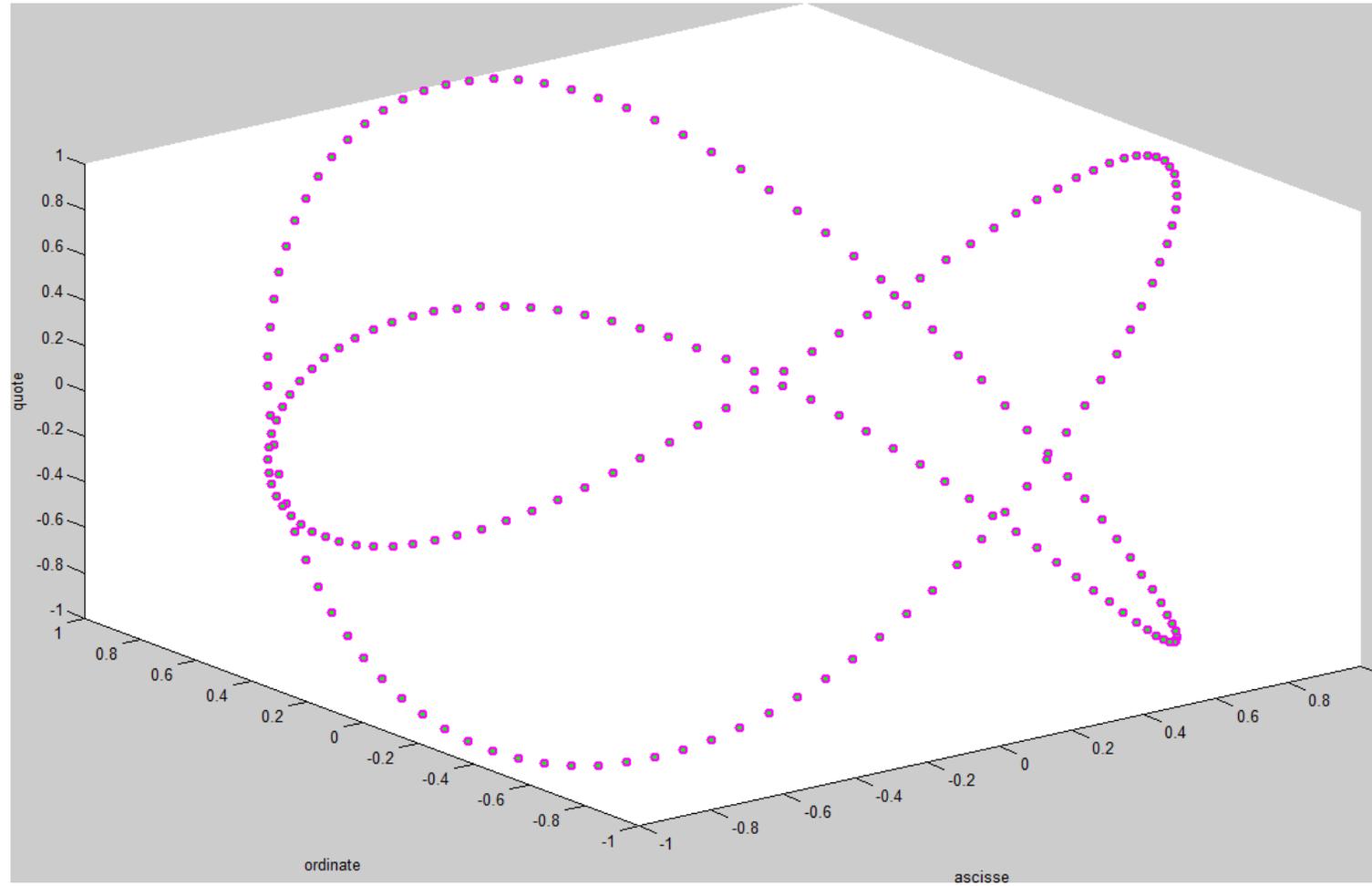


Cosa Cambia?

```
t = linspace(0,4*pi ,200);  
plot_hnd = plot3(sin(t),cos(t),cos(3/2 *t))  
set(plot_hnd, 'LineWidth', 3)  
xlabel('ascisse');  
ylabel('ordinate');  
zlabel('quote');  
  
set(plot_hnd , 'LineStyle','none','Marker','o',  
'MarkerFaceColor', [0 1 0],...  
'MarkerEdgeColor',[1 0 1],...  
'MarkerSize',5,'LineWidth' ,1.5)
```



Risultato





Superfici



Superfici

Come si disegna una superficie che rappresenta una funzione a due variabili $z = f(x, y)$?

Occorre definire il dominio che non è più un intervallo in una retta ma una porzione del piano



Superfici

Come si disegna una superficie che rappresenta una funzione a due variabili $z = f(x, y)$?

Occorre definire il dominio che non è più un intervallo in una retta ma una porzione del piano

La funzione `mesh (xx, yy, zz)` genera superficie, a partire da tre argomenti

- `xx` contiene le ascisse
- `yy` contiene le ordinate
- `zz` contiene le quote

`xx` e `yy` sono matrici che identificano una griglia in corrispondenza del quale per `zz` rappresenta il valore della funzione



Funzione meshgrid

Le due matrici, xx , e yy , si possono costruire, mediante la funzione `meshgrid(x, y)`

```
[xx, yy] = meshgrid(x, y)
```

- x e y sono due vettori
- xx e yy sono due matrici entrambe di `length(y)` righe e `length(x)` colonne
- la prima, xx , contiene, ripetuti in ogni riga, i valori di x
- la seconda, yy , contiene, ripetuti in ogni colonna, i valori di y trasposto



```
[xx, yy] = meshgrid([-3 : 3], [-4 : 4]);
```

xx =

```
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3
```

yy =

```
-4 -4 -4 -4 -4 -4 -4
-3 -3 -3 -3 -3 -3 -3
-2 -2 -2 -2 -2 -2 -2
-1 -1 -1 -1 -1 -1 -1
0 0 0 0 0 0 0
1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4 4
```

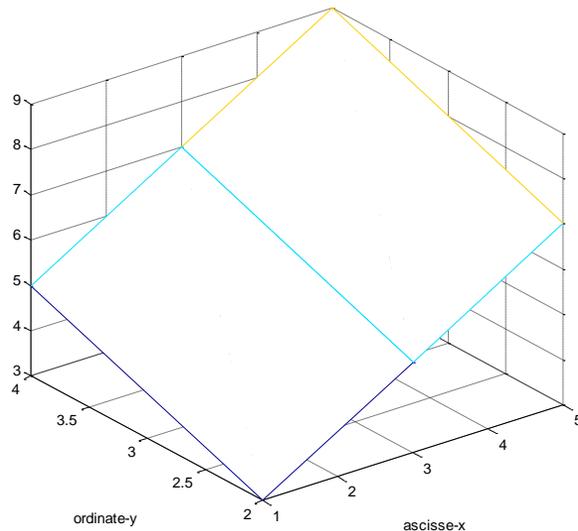
È possibile quindi valutare una funzione di queste due matrici, e.g., $zz = xx + yy$, e disegnarla mediante mesh



Superfici: esempi

Disegniamo $z=x+y$

```
>> x=[1, 3, 5];  
>> y=[2, 4];  
>> [xx,yy]=meshgrid(x,y);  
>> zz=xx+yy;  
>> mesh(xx,yy,zz);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');
```



```
>> xx  
xx =  
 1  3  5  
 1  3  5
```

```
>> yy  
yy =  
 2  2  2  
 4  4  4
```

Punti di coordinate (x,y)...

```
(1,2) (3,2) (5,2)  
(1,4) (3,4) (5,4)
```

```
>> zz  
zz =  
 3  5  7  
 5  7  9
```

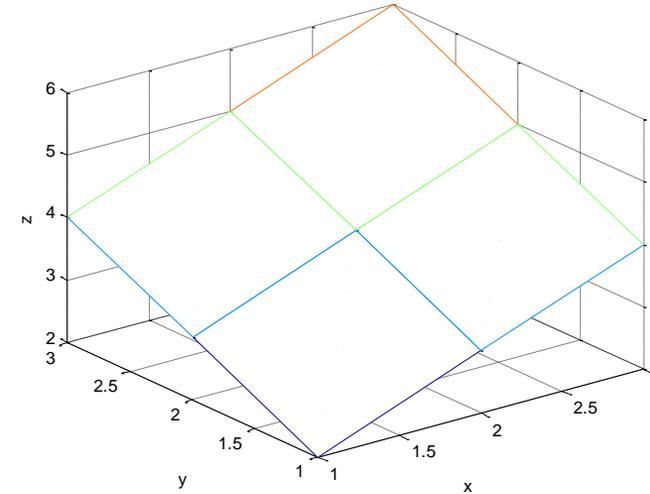
...hanno coordinate (x,y,z)

```
(1,2,3) (3,2,5) (5,2,7)  
(1,4,5) (3,4,7) (5,4,9)  
(NB: z=x+y)
```

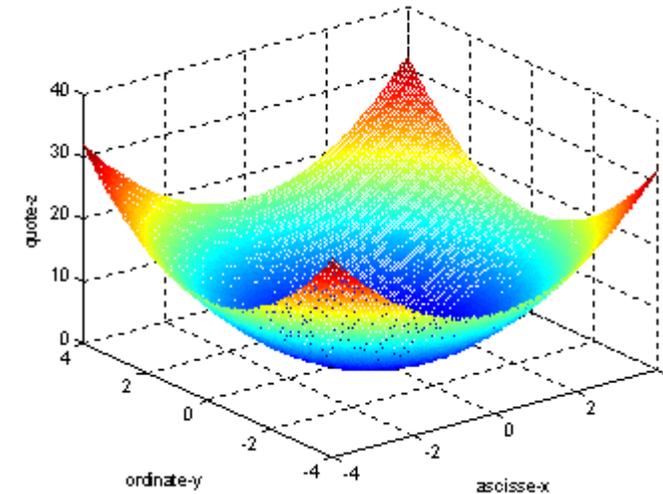


Superfici: esempi (2)

```
>> x=[1:1:3];  
>> y=x;  
>> [xx,yy]=meshgrid(x,y);  
>> zz=xx+yy;  
>> mesh(xx,yy,zz);  
>> xlabel('x');  
>> ylabel('y');  
>> zlabel('z');
```



```
>> x=[-4:0.05:4];  
>> y=x;  
>> [xx,yy]=meshgrid(x,y);  
>> zz=xx.^2 + yy.^2;  
>> mesh(xx,yy,zz);  
>> xlabel('ascisse-x');  
>> ylabel('ordinate-y');  
>> zlabel('quote-z');
```





Hold on

Le superfici vengono visualizzate su un grafico 3D.

È quindi possibile aggiungere degli elementi in sovrapposizione utilizzando la funzione

- plot3(), mesh(), altre funzioni grafiche quali surf() etc..
- Per sovrascrivere ad un grafico usare la funzione hold **on** e hold **off** quando si ha terminato

Esempio, disegnare in sovrapposizione al paraboloido precedente la curva $\begin{cases} z = x^2 \\ y = 0 \end{cases}$

```
dove x = [ -4 : 0.05 : 4 ] ;  
figure, mesh(xx, yy, zz)  
hold on  
% aggiunge una linea rossa con uno spessore di 5  
plot3(x, zeros(size(x)), x.^2, 'r-', 'LineWidth', 5);  
hold off
```



Mesh

```
[xx, yy] = meshgrid([-3 : 0.1 :3], [-4 : 0.1 :4]);
```

```
f = @(x, y)(1 - x.^2 - y.^2);
```

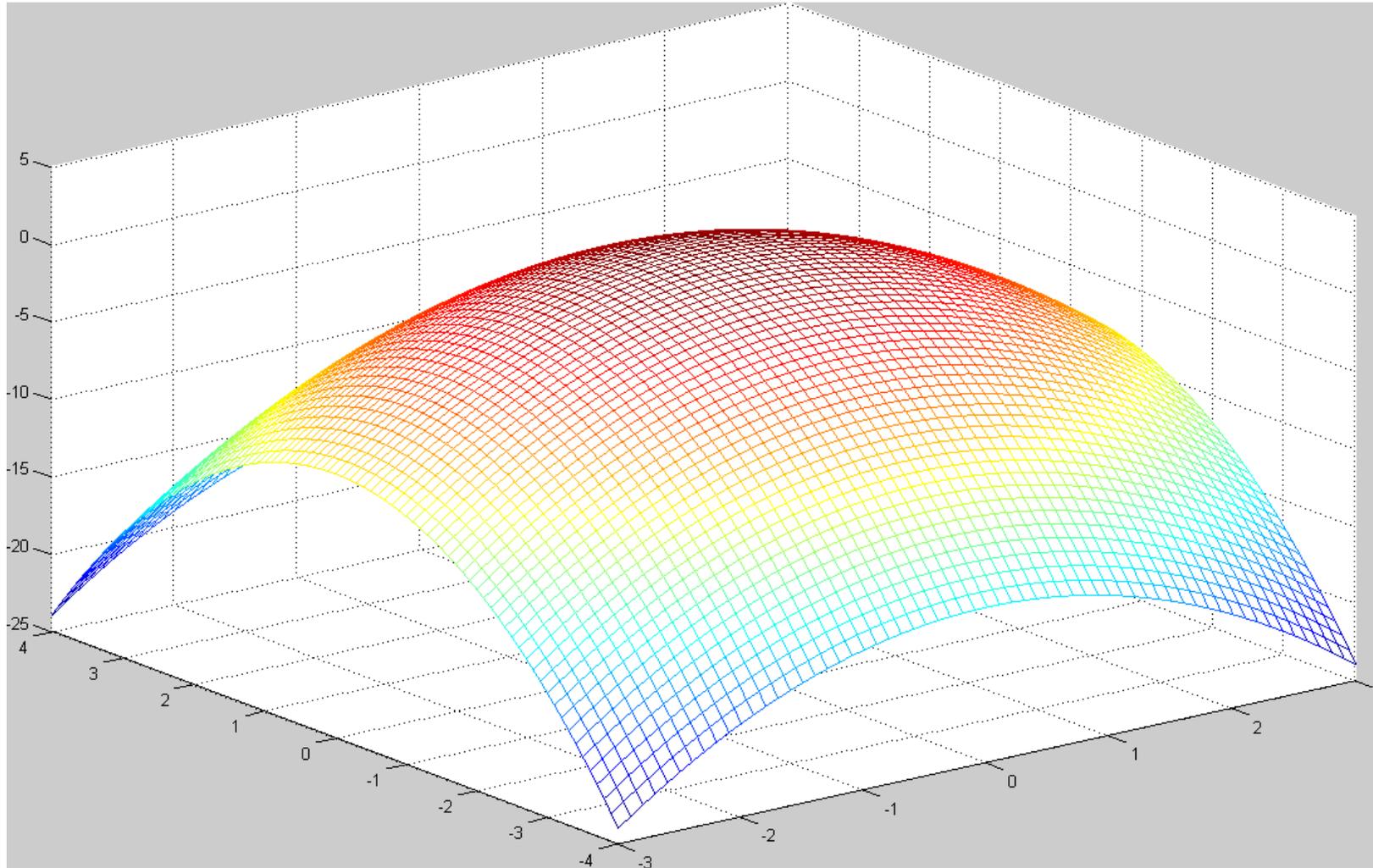
```
figure,
```

```
aa = mesh(xx, yy, f(xx, yy))
```

Mesh unisce i punti con delle linee colorate. Di default il colore indica il valore della quota



Risultato





Riempire la superficie

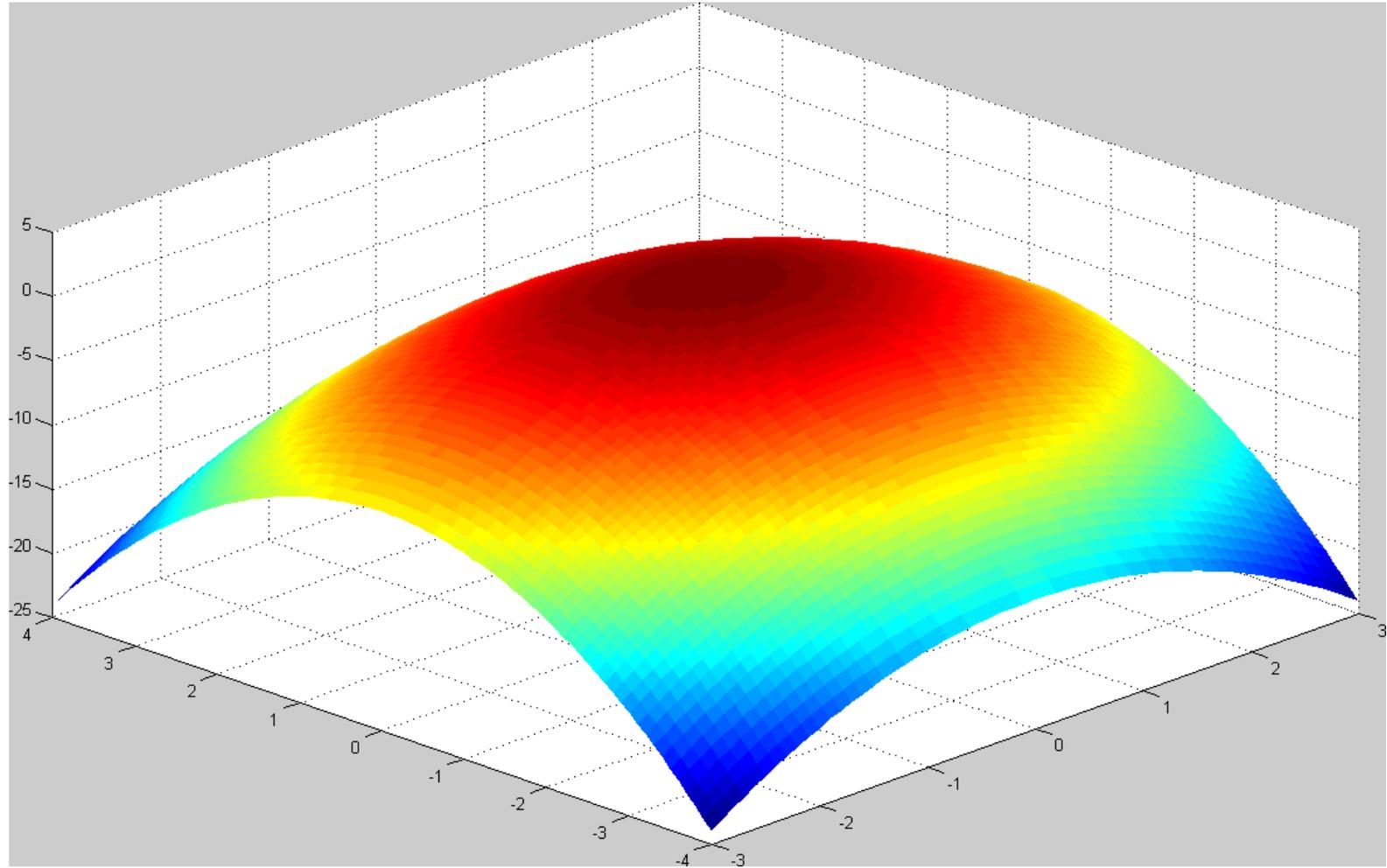
```
[xx, yy] = meshgrid([-3 : 0.1 :3], [-4 : 0.1 :4]);
```

```
f = @(x, y)(1 - x.^2 - y.^2);
```

```
figure,
```

```
aa = surf(xx, yy, f(xx, yy))
```

```
set(aa, 'EdgeColor', 'none')
```





Risultato

```
[xx, yy] = meshgrid([-3 : 0.1 :3], [-4 : 0.1 :4]);
```

```
f = @(x, y)(1 - x.^2 - y.^2);
```

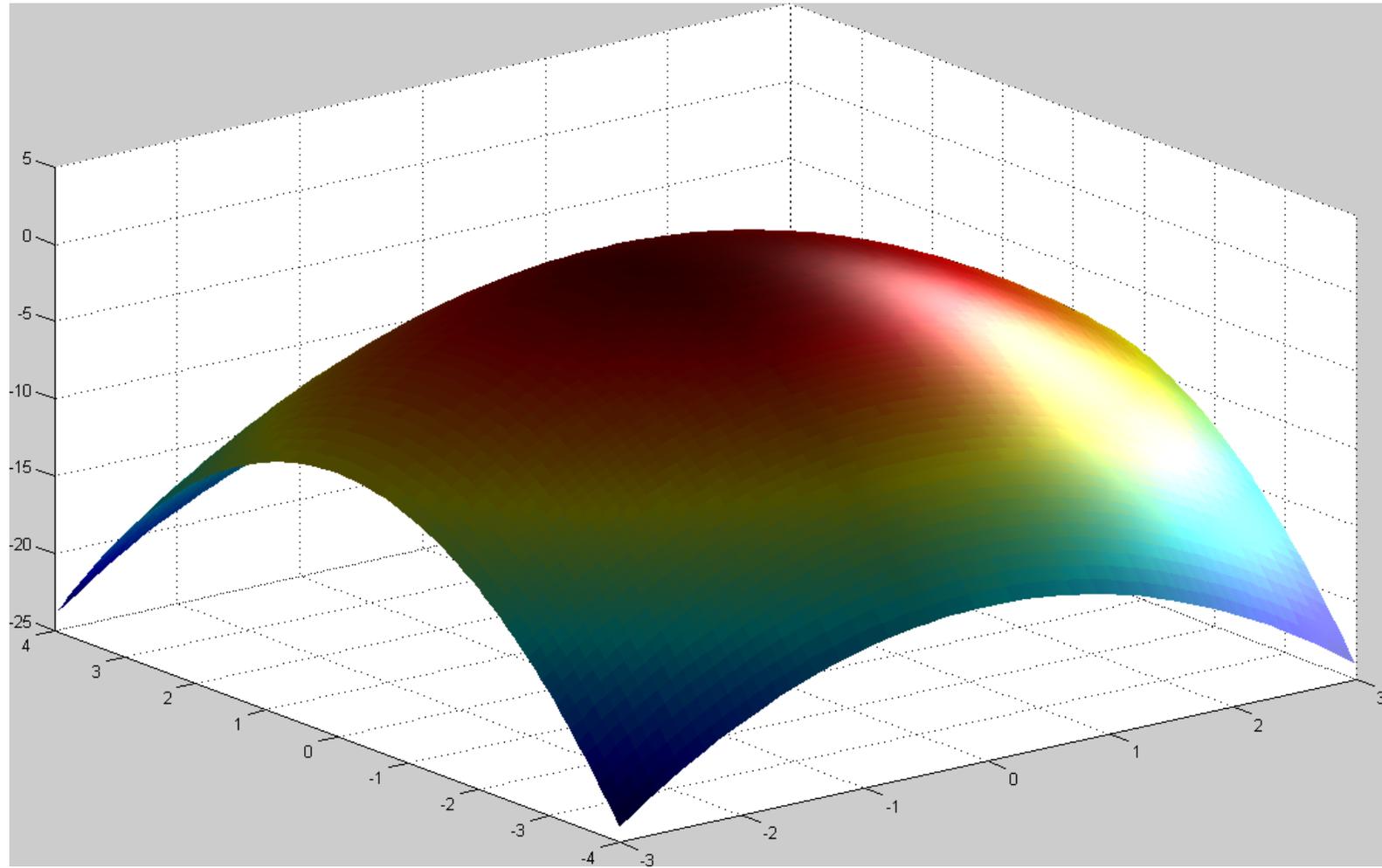
```
figure,
```

```
aa = surf(xx, yy, f(xx, yy))
```

```
set(aa, 'EdgeColor', 'none')
```

```
light % aggiunge una sorgente luminosa per il rendering
```

Surf riempie le regioni tra le linee con del colore che, di default, dipende dal valore della quota





Hold on

Le superfici vengono visualizzate su un grafico 3D

È quindi possibile aggiungere degli elementi in sovrapposizione utilizzando la funzione

- `plot3()`, `mesh()`, altre funzioni grafiche quali `surf()`
- Per sovrascrivere ad un grafico usare la funzione `hold on` e `hold off` quando si ha terminato



Esempio

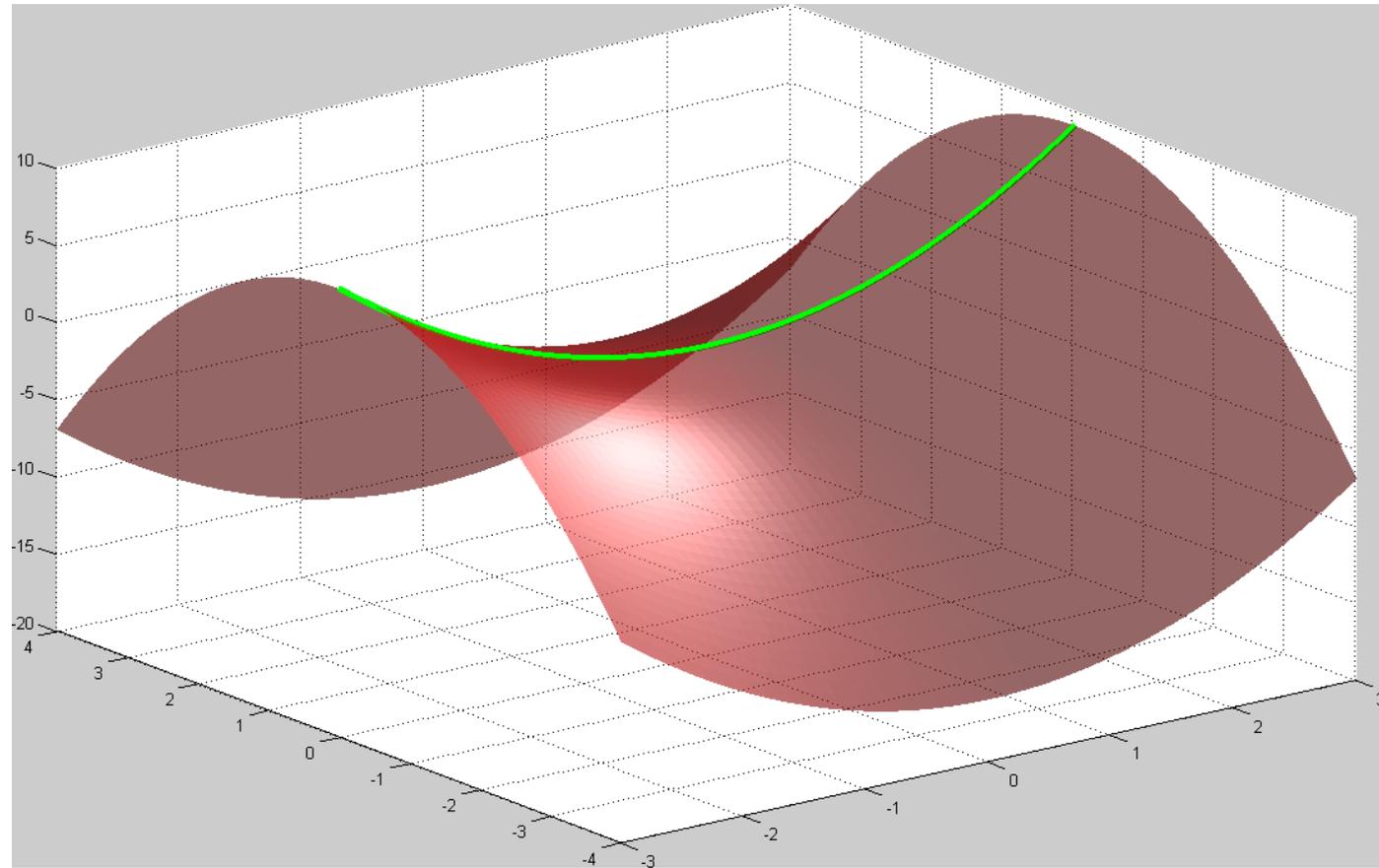
Disegnare in sovrapposizione alla quadrica

$$z = x^2 - y^2 \text{ la curva } \begin{cases} z = x^2 \\ y = 0 \end{cases}$$

```
[xx, yy] = meshgrid([-3 : 0.1 :3], [-4 : 0.1 :4]);  
f = @(x, y) (x.^2 - y.^2);  
figure(),  
aa = surf(xx, yy, f(xx, yy))  
hold on  
x = xx(1, :);  
y = zeros(size(x));  
bb = plot3(x, y, f(x,y), 'g-')  
set(aa, 'EdgeColor', 'none', 'FaceColor', 'red',  
'FaceAlpha', 0.6)  
set(bb, 'Linewidth', 3)  
light  
hold off
```



Risultato





Esempio

Disegnare la funzione

$$z = \frac{\sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}}$$

e una curva su questa funzione passante per l'origine

```
tx = [-8:0.1:8];  
ty = tx;  
[xx, yy] = meshgrid (tx, ty);  
f = @(x,y)(sin(sqrt(x.^2 + y.^2)) ./ sqrt(x.^2 + y.^2));  
figure,  
aa = surf(xx, yy, f(xx, yy));  
hold on  
bb = plot3(tx, tx, f(tx, tx), 'r-', 'LineWidth', 3)  
set(aa, 'EdgeColor', 'none')
```



Risultato

