

Riccardo Fazio

Dipartimento di Matematica, Informatica,
Fisica e Scienze della Terra
Università degli Studi di Messina

`mat521.unime.it/fazio`
`rfazio@unime.it`

Introduzione a

OCTAVE ad uso

degli studenti e non solo

Capitolo 1

Introduzione ad OCTAVE

L'utilizzo di OCTAVE in molti settori delle Scienze Applicate é ai giorni routine quotidiana. Non sorprende, quindi la sua applicazione a molti problemi di economia o della finanza.

OCTAVE é un software gratuito, clone del piú famoso MATLAB, e come questi é un linguaggio evoluto interpretato. Linguaggio interpretato significa che OCTAVE legge i comandi linea per linea e li interpreta al fine di eseguirli. Per la veritá dovremmo dire GNU OCTAVE ma in questo libro per semplicitá useremo solo OCTAVE. Per la comoditá dell'utente, OCTAVE dispone di un'interfaccia grafica che si compone di tre parti: la finestra dei comandi, un editor dedicato e un ambiente di aiuto. Dalla linea dei comandi, OCTAVE si puó usare come una calcolatrice scientifica, noi tralascieremo di descrivere questa parte perché troppo elementare e diremo invece come l'elemento base di OCTAVE, cosí come del MATLAB, é la matrice. Inoltre, OCTAVE, proprio come il MATLAB, esegue operazioni in precisione doppia, per avere idea delle questioni che coinvolgono la precisione di macchina si

veda Fazio [8, pp. 3-15]. Qui basterá ricordare che la precisione di macchina é una misura della precisione dell'aritmetica di macchina, in quanto é una maggiorazione dell'errore relativo che si commette nel sostituire un qualsiasi numero reale con il corrispondente numero di macchina. Quindi, piú piccolo é il numero di macchina migliore é l'aritmetica di macchina.

1.1 O-files di tipo script

Dal nostro punto di vista, l'editor é la parte piú importante ed é quella che adopereremo molto spesso.

1.2 O-files

Le istruzioni di OCTAVE che permettono di risolvere un dato problema possono essere scritte sulla riga di comando, ma possono piú convenientemente essere salvate in un file, uno script file, un cosiddetto O-file, che per la compatibilitá con il MATLAB avrá estensione `.m`. In un O-file si possono inserire dei commenti che devono essere preceduti dal simbolo `%`.

Esempio: Supponiamo di voler disegnare la funzione seno nell'intervallo $[0, 2\pi]$. A tal fine, potremmo utilizzare il semplice O-file:

```
% script-file seno.m
% disegna il sin(x) nell'intervallo [0, 2*pi]
x = linspace(0,2*pi);
y = sin(x);
plot(x,y,'ro')
```

```
grid
xlabel('x')
ylabel('sin(x)')
```

Qualche commento sembra opportuno. Per prima cosa l'istruzione OCTAVE `linspace(a, b)` definisce un vettore di 101 componenti, equispaziati fra a e b . Se si volesse un numero diverso di componenti si potrebbe usare un valore numerico come terzo argomento di `linspace`. In OCTAVE, e in MATLAB, la costante `pi` ha il valore di π . OCTAVE opera in modo vettoriale, ecco perché per definire le ordinate basta una sola istruzione, la `y = sin(x)`; Il punto e virgola alla fine di una istruzione evita l'eco a video e quindi permette di velocizzare l'interpretazione dell'O-file da parte di OCTAVE. Il comando `plot` permette di ottenere un grafico bidimensionale, torneremo sulle istruzioni di grafica di OCTAVE in seguito. il terzo argomento di `plot` specifica che il plot sarà dato da pallini rossi, se volessimo anche una linea continua che unisca i pallini, allora potremmo aggiungere un trattino scrivendo `'ro-'`. L'istruzione `grid` fa sì che OCTAVE disegni un reticolo bidimensionale nel dominio che verrà rappresentato. Le due istruzioni `xlabel` e `ylabel` permettono di apporre le etichette indicate ai due assi.

1.3 O-files di tipo function

Gli O-files di tipo function vengono utilizzati per definire delle funzioni. Questi files hanno la struttura come nell'esempio seguente:

```
function [f] = Keplero(u,t,e)
% Function-file Keplero.m
```

```
% di seguito viene definita la funzione di Keplero  
f = u - t - e*sin(u);
```

nel quale viene definita la funzione di Keplero, vedi Fazio [8, pp. 36-37].

1.3.1 Handle

Un handle é un riferimento ad una funzione che può essere trattata come una variabile, cioè un valore di tipo funzione può essere assegnato ad una variabile. Per la definizione di un handle si utilizza il simbolo @.

Esempio di handle. Si vuole creare un handle della funzione di Keplero da utilizzare in diversi O-files. Ad esempio, si potrà disegnare la funzione di Keplero nell'intervallo $[-10, 10]$ scrivendo la linea di comando `fplot(@Keplero, [-10, 10])` che ha come argomenti l'handle e l'intervallo sull'asse delle ascisse. Oppure, se si vuole utilizzare il comando `plot` che abbiamo incontrato in precedenza si dovrà specificare un reticolo sull'asse delle ascisse e poi valutare la funzione di Keplero nei punti del reticolo per mezzo dell'istruzione `feval`.

```
% This is PlotKeplero.m  
% plots Keplero's function in the range [-10,10]  
% at time t = 20 and eccentricity e = 0.9 u = [-10:0.25:10];  
f = feval(@Keplero,u,20,.9);  
plot(x,f,'bs-', 'LineWidth',2, 'MarkerSize',10)  
grid  
xlabel('u')  
ylabel('f')
```

```
print -depsc2 Keplero
```

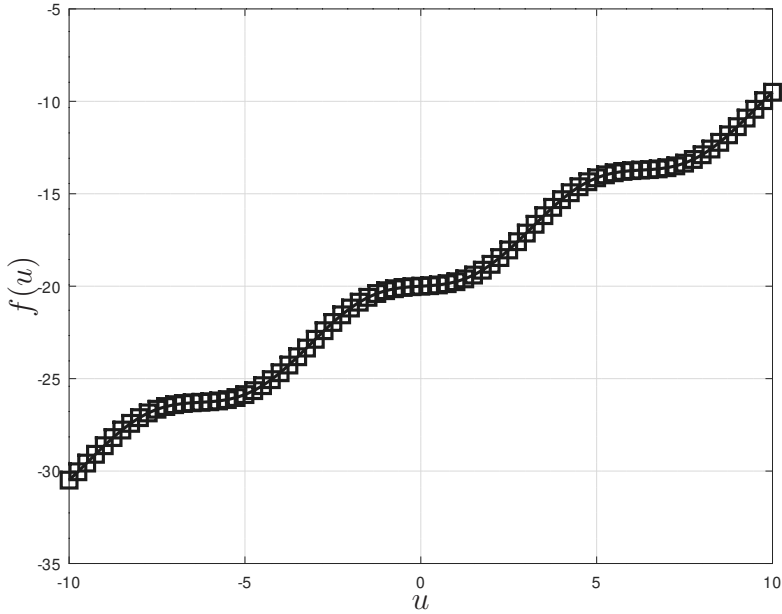


Figura 1.1: Funzione di Keplero nell'intervallo $[-10, 10]$.

Si noti che a causa delle opzioni del comando `plot`, avremo una linea tratteggiata con dei quadrati in corrispondenza ai nodi del reticolo per `'rs-'` la linea di dimensione 2 a causa dell'opzione `'LineWidth', 2` e i marker di dimensione 10 per via dell'opzione `'MarkerSize', 10`. Il grafico risultate si può vedere nella figura 1.1

Capitolo 2

Grafici e figure con OCTAVE

OCTAVE, al pari di MATLAB, ha delle potenti istruzioni per generare e gestire la grafica.

In questo capitolo ci occuperemo di elencare le istruzioni per la grafica di OCTAVE e riportare delle figure per mostrare il loro effetto.

2.1 Grafici bidimensionali

Iniziamo con il considerare le istruzioni per la grafica bidimensionale. Il grafico di una funzione reale di variabile reale $y = f(x)$ é l'insieme dei punti del piano \mathbb{R}^2 individuati dalla coordinate $(x, f(x))$. Come abbiamo già visto, il comando di OCTAVE per la grafica in due dimensioni é `plot`. Si noti come, entro certi limiti, piú fine é il reticolo e migliore risulterà il grafico della funzione. Come abbiamo precedentemente visto con degli esempi, il grafico puó essere reso piú esplicativo mediante le numerose opzioni del comando `plot`, facilmente richiamabili attraverso `help` di OCTAVE. I colori utilizzabili dal comando

Tabella 2.1: Colori e stili di linea principali predefiniti in OCTAVE.

Colore	Simbolo	Stile di linea	
y	giallo	*	asterischi
m	magenta	s	quadrati
c	turchese	p	pentagoni
r	rosso	h	esagoni
g	verde	o	cerchi
b	blu	x	x
w	bianco	+	croci
k	nero	-	linea continua
		—	tratto con —
		:	tratto punteggiato
		-.	tratto punti
		.	punti

`plot`, così come i principali stili di linea, sono riportati nella tabella 2.1.

2.2 Grafici bidimensionali multipli

Vediamo adesso come é possibile disegnare piú grafici bidimensionali nella stessa figura.

Esempio. Supponiamo di voler disegnare le due funzioni $f_1(n) = n^3$ e $f_2(n) = n!$, al fine di comprendere la crescita delle due funzioni e quale fra queste cresce piú velocemente. Possiamo allora utilizzare i comandi seguenti:

```
% This is Grafico2Dm.m
```



```
clear all
nmax = 25;
n = [0:1:nmax];
f1 = n.*n.*n;
f2(1) = 1;
for j = 2:nmax+1
    f2(j) = f2(j-1)*j;
end
plot(n,f1,'bs-', 'MarkerSize',10,n,f2,'ro-', 'MarkerSize',10)
grid; xlabel('n'); ylabel('f')
axis([0 nmax 0 10000])
```

Il comando `clear all` ripulisce il Workspace di OCTAVE, cioè le allocazioni di memoria che OCTAVE conserva per ulteriori utilizzi dopo il calcolo. Si noti il comando `.*` che viene utilizzato da OCTAVE per eseguire la moltiplicazione fra due vettori componente per componente. Il comando OCTAVE `axis` definisce l'ampiezza del grafico che verrà visualizzato. Il risultato dei nostri calcoli è visibile nella figura 2.1.

Un secondo modo di ottenere lo stesso risultato, è utilizzare il comando OCTAVE `hold on`, e in tal modo si possono agevolmente disegnare diversi grafici sulla stessa figura.

2.3 Sotto-figure con OCTAVE

In questa sezione, vedremo di chiarire come la grafica di OCTAVE permetta di ottenere delle finestre grafiche più piccole nella singola figura. Come vedremo, i comandi `hold` e `grid` funzionano al solito

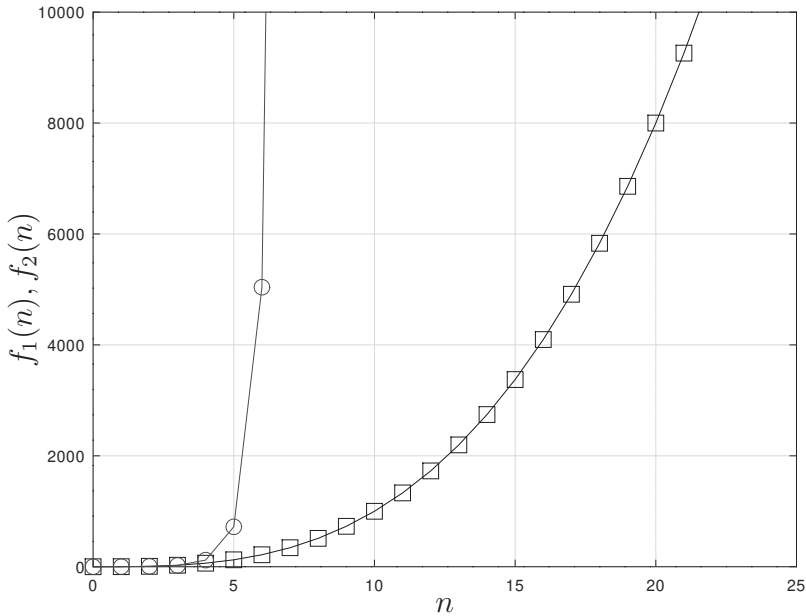


Figura 2.1: Le due funzioni $f_1(n) = n^3$ e $f_2(n) = n!$ disegnate su un unico grafico.

anche nelle sotto-finestre. Per ottenere delle sotto-finestre, OCTAVE dispone del comando `subplot(i, j, k)`, dove i e j sono gli indici di riga e colonna di una ipotetica matrice e corrispondono alla posizione della (i, j) -esima sotto-finestra, mentre k va da uno a $i \times j$ e individua la k -esima sotto-finestra a partire dalla finestra in alto a sinistra. Chiariamo con un esempio, quanto abbiamo cercato di spiegare a parole fin qui. Secondo un detto cinese vale piú una figura che non mille parole.

Esempio di grafica con sotto-finestre. Supponiamo di voler confrontare le quattro funzioni elementari: $f_1(x) = x$, $f_2(x) = x^2$, $f_3(x) =$

$1/x$ e $f_4(x) = \sin(x)$ nell'intervallo $x \in [-10, 10]$. A tal fine, potremmo utilizzare la grafica di OCTAVE disegnando una sola figura come segue.

```
% This is Grafica2Df.m
x = [-10:0.5:10];
subplot(2,2,1)
plot(x,x,'k*-')
grid; xlabel('x'); ylabel('f_1(x) = x')
subplot(2,2,2)
plot(x,x.*x,'bo-')
grid; xlabel('x'); ylabel('f_2(x) = x^2')
subplot(2,2,3)
plot(x,1./x,'rs-')
grid; xlabel('x'); ylabel('f_3(x) = 1/x')
subplot(2,2,4)
plot(x,sin(x),'mh-')
grid; xlabel('x'); ylabel('f_4(x) = sin(x)')
```

Il risultato dei nostri calcoli é visibile nella figura 2.2.

Naturalmente, OCTAVE ha altre istruzioni per la grafica, ad esempio `semilogx` e `semilogy` per i grafici semi-logaritmici, `loglog` per i grafici logaritmici e altre ancora. Il lettore interessato, può a tal proposito consultare l'help di OCTAVE.

2.4 Istogrammi

In varie applicazioni risulta utile rappresentare i dati per mezzo di istogrammi. Come farlo con OCTAVE é l'argomento di questa sezione.

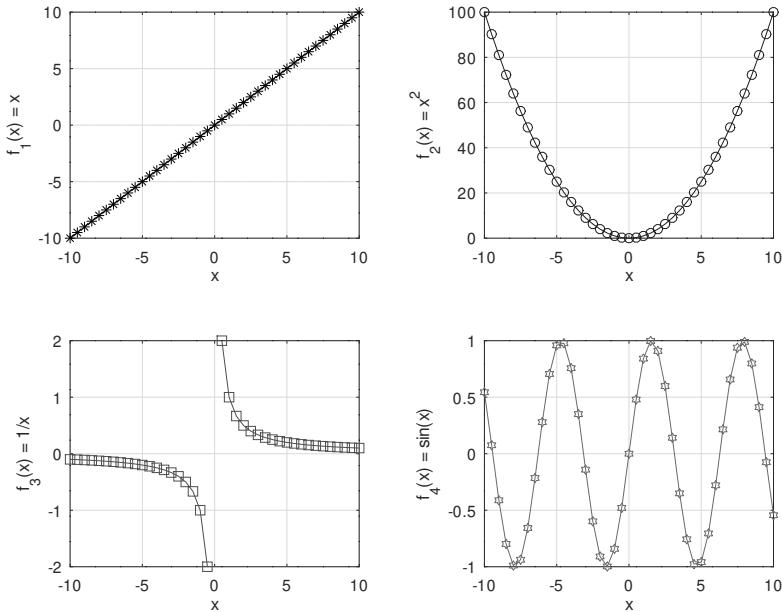


Figura 2.2: Le quattro funzioni $f_1(x) = x$, $f_2(x) = x^2$, $f_3(x) = 1/x$ e $f_4(x) = \sin(x)$ disegnate su quattro sotto-figure.

Il comando `bar(x, y)` disegna barre centrate in x , che deve essere un vettore, di altezze definite nel vettore o matrice, y . Se y é una matrice, allora ogni sua colonna é disegnata con delle barre nello stesso grafico. Di default le barre vengono disegnate fianco a fianco, ma questo si può eventualmente, cambiare agendo sulle opzioni del comando.

Esempio di istogramma. Supponiamo di voler disegnare l'istogramma degli incrementi o decrementi percentuali delle quotazioni di alcune azioni in diversi anni come calcolate alla fine dell'anno di riferimento.

```
% This is Bar1.m
clear all
```

```
x = [2015 2016 2017 2018 2019];  
y = [3 2 4 -1;  
     0 2 3 -2;  
     1 0 -2 3;  
     4 -1 1 1;  
     2 5 11 -4];  
bar(x,y)  
xlabel('Anni')  
ylabel('Incremento quotazioni')
```

Il risultato dei nostri calcoli si può vedere nella figura 2.3. Se si vo-

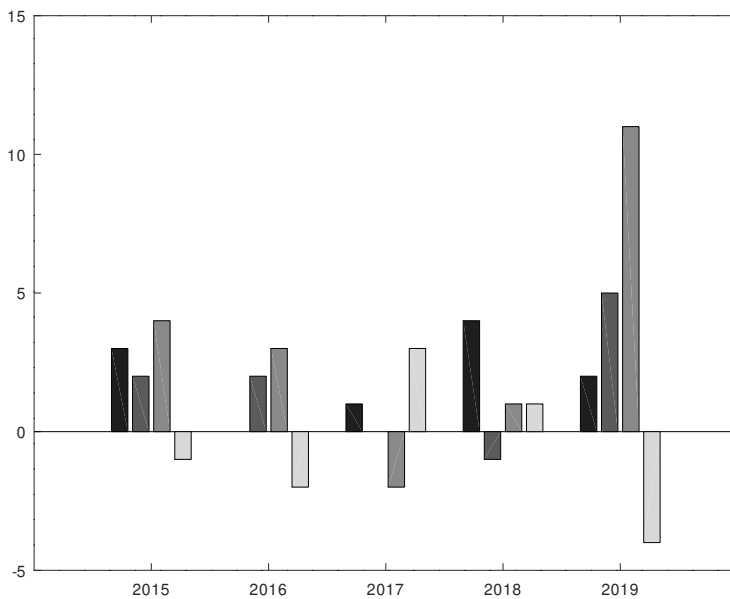


Figura 2.3: Esempio di istogramma.

lesse un istogramma con delle barre orizzontali, allora si potrebbe

usare il comando `barh`. Quindi, sostituendo il comando `bar(x,y)` con `barh(x,y)` si ottiene il grafico della figura 2.4.

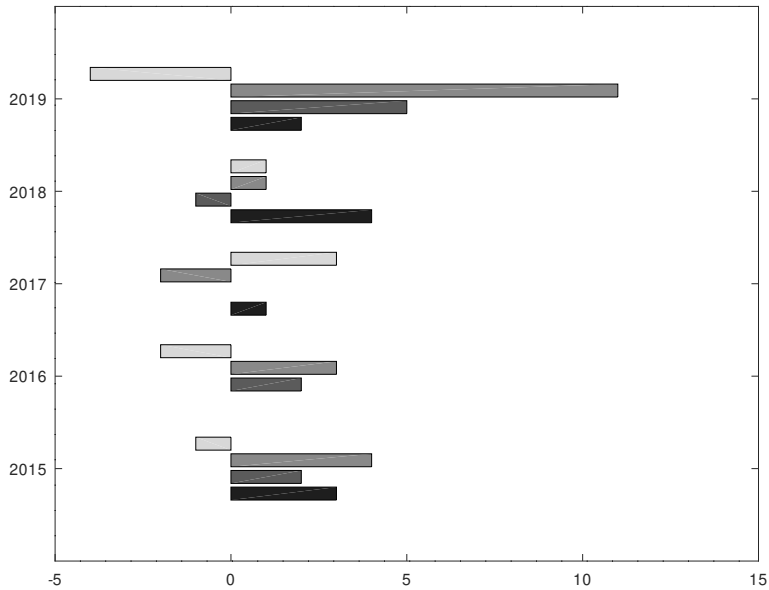


Figura 2.4: Esempio di istogramma con barre orizzontali.

Un altro comando che permette di disegnare degli istogrammi é il comando `ist`. Le istruzioni seguenti permettono di disegnare l'istogramma in figura 2.5.

```
% This is Ist.m
clear all
x = randn(100,1);
n = 10;
hist(x, n)
```

Il comando `randn` genera dei numeri casuali con distribuzione normale che hanno media zero e varianza uno. In rapporto alla generazione di numeri casuali, si vedano anche i comandi `rand`, `rande`, `randg`, `randp`. Nella tabella 2.2 sono riportati i comandi di OCTAVE per la grafica

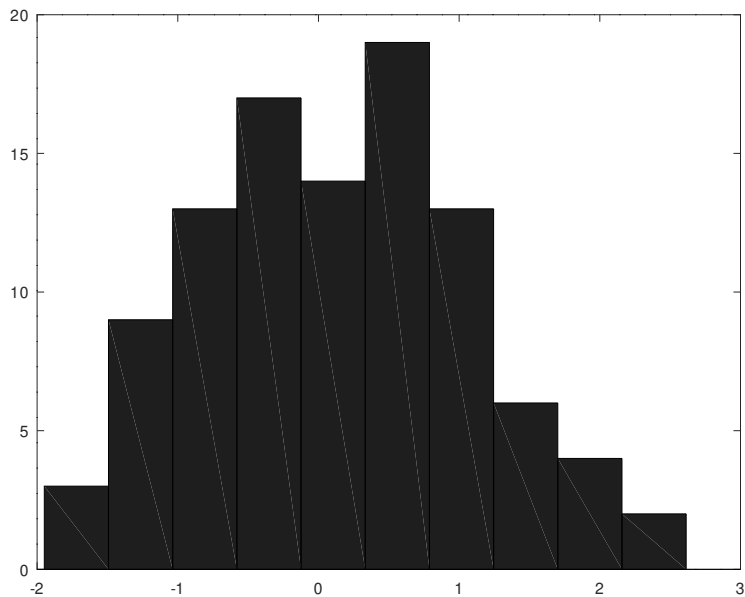


Figura 2.5: Esempio di istogramma generato dal comando `ist`.

bidimensionale.

Tabella 2.2: Comandi di OCTAVE per la grafica bidimensionale.

comando	effetto
<code>plot</code>	semplice grafico 2D
<code>loglog</code>	grafico con assi logaritmici
<code>semilogx</code>	grafico con asse x logaritmico
<code>semilogy</code>	grafico con asse y logaritmico
<code>polar</code>	grafico in coordinate polari
<code>fplot</code>	funzione grafica automatica
<code>ezplot</code>	semplice versione di <code>fplot</code>
<code>ezpolar</code>	semplice versione di <code>polar</code>
<code>fill</code>	riempimento di poligoni
<code>area</code>	riempimento di aree
<code>bar</code>	grafici con istogrammi
<code>barh</code>	istogrammi con barre orizzontali
<code>hist</code>	semplice istogramma
<code>pie</code>	grafico a torta
<code>comet</code>	grafico animato 2D
<code>errorbar</code>	grafico a barre dell'errore
<code>quiver</code>	ampiezza di vettori velocità

2.5 Grafica tridimensionale

Il comando `plot3` é l'analogo per la grafica in tre dimensioni del semplice comando `plot`. Il caso piú semplice da considerare é quello di una superficie parametrica definita dai punti dello spazio individuati dalle coordinate $(x_1(t), x_2(t), x_3(t))$, dove t é il parametro scalare che

varia nell'intervallo t_{\min}, t_{\max} . Consideriamo un semplice esempio.

Esempio di grafico ottenuto con plot3. Supponiamo di voler disegnare la superficie parametrica data dal parametro definito da $t = -10 : 0.01 : 10$ con $(x_1 = t, x_2 = (1 + t.^2) * \sin(25 * t), x_3 = (1 + t.^2) * \cos(10 * t))$. Potremmo allora utilizzare le istruzioni di OCTAVE:

```
t = -10:0.01:10;
x_1 = t;
x_2 = (1+t.^2).*sin(25*t);
x_3 = (1+t.^2).*cos(10*t);
plot3d(x_1, x_2, x_3)
grid
xlabel('x_1(t)')
ylabel('x_2(t)')
zlabel('x_3(t)')
title('Esempio di grafico con plot3')
```

Il risultato si può vedere nella figura 2.6.

Il grafico di una funzione reale di due variabili reali $f : D \subset \mathbb{R}^2 \leftarrow \mathbb{R}$ é l'insieme dei punti dello spazio tridimensionali individuati dalle coordinate $(x_1, x_2, x_3 = f(x_1, x_2))$. In questo caso, per disegnare una superficie in tre dimensioni, si deve generare un reticolo di punti nel piano $x - y$. A tal fine potremo usare il comando OCTAVE `meshgrid`. Il comando `[X,Y] = meshgrid(x,y)` riscrive le coordinate dei nodi del reticolo (x, y) nelle due matrici X e Y . Ad esempio il comando

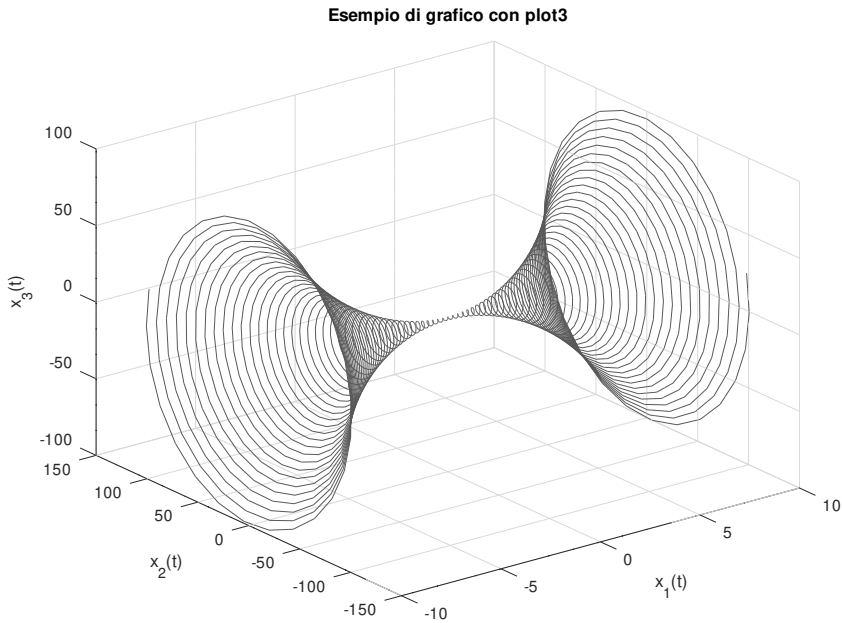


Figura 2.6: Esempio di utilizzo di semplice grafica tridimensionale.

`[X,Y] = meshgrid(0:1,0:5)` genera le matrici X data da

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

e Y , che invece, é data da

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{bmatrix} .$$

Esempio di grafico di una superficie. Supponiamo di voler disegnare il grafico della superficie di equazione $z = x + y^3$ nel dominio bidimensionale $(x, y) \in [-5, 5] \times [-5, 5]$. Le istruzioni seguenti permettono di disegnare la figura 2.7.

```
%This is Grafico3D.m
clear all
x = [-5:0.5:5];
y = x;
[X,Y] = meshgrid(x, y);
f = X+Y.^3;
surf(X,Y,f)
title('grafico della superficie z = x+y^3')
colormap
```

Il comando OCTAVE per disegnare la superficie é `surf(X,Y,f)` . Il comando `title` permette di aggiungere un titolo al grafico, mentre il comando `colormap` aggiunge una scala di colori, con i diversi livelli, alla figura.

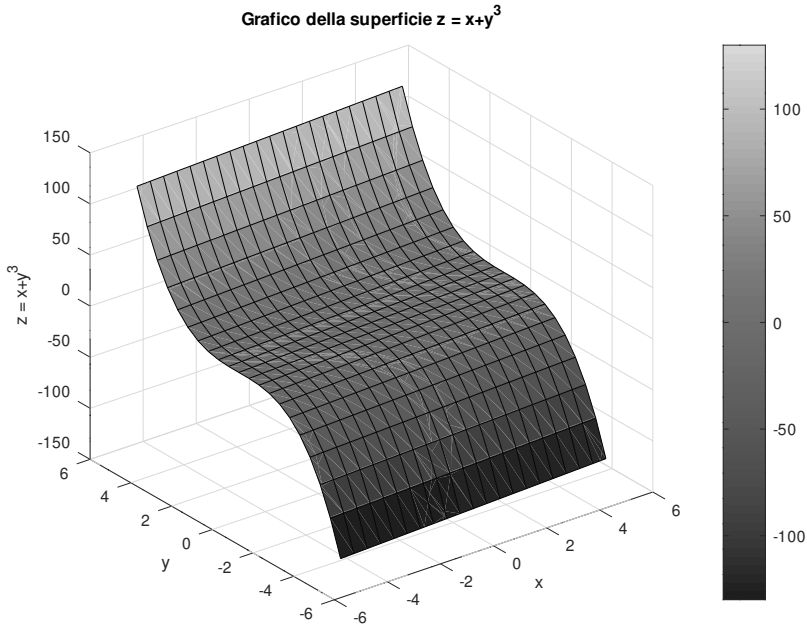


Figura 2.7

2.5.1 Restrizioni

A volte può essere utile, o viene richiesto espressamente, di disegnare la restrizione di una superficie ad un dato piano. Vediamo come questo sia possibile in OCTAVE con un esempio.

Esempio di un grafico con restrizione. In questo esempio vogliamo disegnare la restrizione della superficie $z = x + y^3$, $(x, y) \in [-5, 5] \times [-5, 5]$, al piano $y = 0$. A tal fine fra le istruzioni precedenti basterà sostituire la $y = x$; con $y = 0 \cdot \text{ones}(\text{size}(x))$; Il comando OCTAVE `ones` genera un vettore con tutte le componenti uguali a uno della dimensione dell'argomento. Il comando `size` ricava la dimensione del vettore in argomento. Il risultato che otteniamo si può

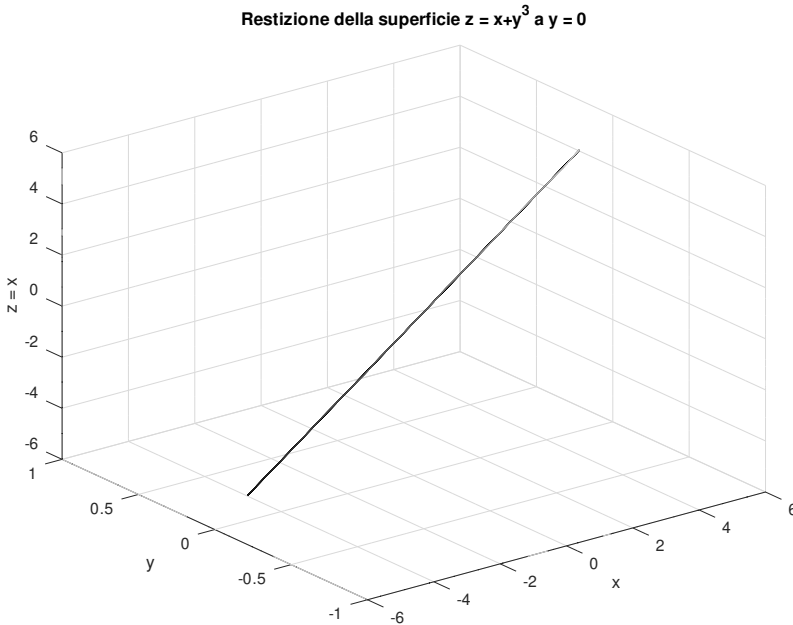


Figura 2.8: Restrizione di un grafico tridimensionale.

vedere nella figura 2.8.

2.5.2 Grafici di contorni

OCTAVE mette a disposizione dell'utente il comando `ezcontour` per ottenere dei semplici grafici di contorni. Inoltre, é anche disponibile il comando `contour` per grafici piú dettagliati. Proviamo a vedere il loro utilizzo con un esempio.

Esempio di utilizzo dei comandi `ezcontour` e `contour`. Supponiamo di voler disegnare dei grafici di contorni della funzione $\sin(y - 2 * x^2 + 2) + \cos(3 * y^2 - x - 10)$ nel dominio definito da $x \in [-33]$ e $y \in [-33]$. Potremo usare i seguenti comandi:

```
clear all
x = -3:0.2:3;
y = x;
[X, Y] = meshgrid(x, y);
Z = sin(Y-2*X.^2+2)+cos(3*Y.^2-X-10);
subplot(211)
ezcontour(X, Y, Z)
subplot(212)
contour(X, Y, Z)
title('Esempio dell'uso di contour.')
```

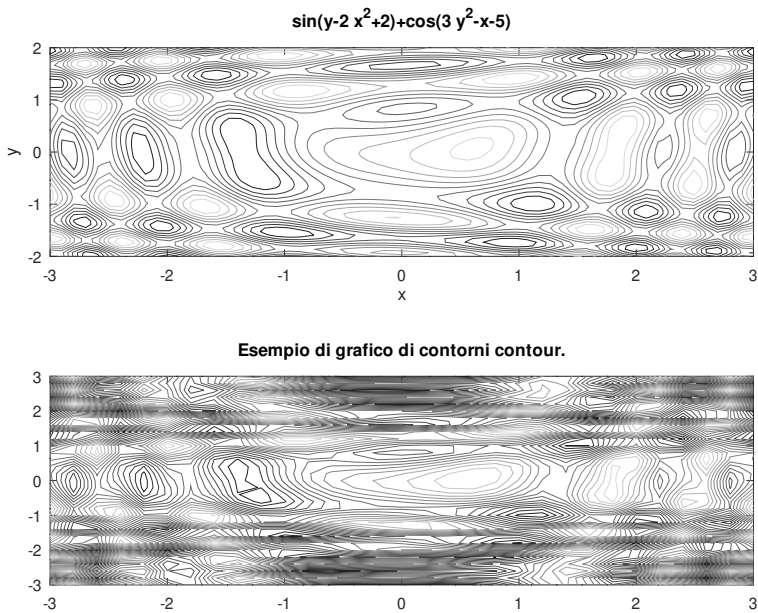


Figura 2.9: Esempio dell'uso dei due comandi per i grafici di contorni.

Il risultato che otteniamo si può vedere nella figura 2.9.

Nella tabella 2.3, infine, sono riportati i comandi di OCTAVE per la grafica tridimensionale.

Tabella 2.3: Comandi di OCTAVE per la grafica tridimensionale.

comando	effetto
<code>plot3</code>	semplice grafico 3D
<code>contour</code>	grafico dei contorni
<code>contourf</code>	grafico dei contorni con riempimenti
<code>contour3</code>	grafico dei contorni 3D
<code>mesh</code>	grafico 3D wireframe
<code>meshc</code>	grafico 3D wireframe con contorni
<code>surf</code>	grafico di superficie solida
<code>surfz</code>	grafico di superficie solida con contorni
<code>pie3</code>	grafico a barre 3D
<code>comet3</code>	grafico animato 3D
