

# Introduzione ad Gnuplot

## versione 1

Roberto Boggiani

7 febbraio 2003

## 1 Introduzione

### 1.1 Presentazione

Gnuplot è un programma per produrre grafici a due e tre dimensioni. Esso permette di formattare tali grafici in una varietà di modi e di esportarli in formati facilmente importabili in documenti  $\text{\LaTeX}$ .

### 1.2 Come ottenere un aiuto su un comando

Il programma permette, conoscendo il nome di un comando, di ottenere un aiuto sul comando stesso relativo sia al suo utilizzo sia al risultato che da esso si ottiene. Per richiamare un aiuto su un qualunque comando si usa la seguente sintassi:

```
help comando
```

in cui `comando` è il nome del comando di cui vogliamo un aiuto.

Come vedremo il programma permette di modificare i valori predefiniti attraverso il comando:

```
set nomeopzione
```

in cui `nomeopzione` rappresenta l'opzione che vogliamo modificare. In questo caso per ottenere l'help sull'opzione dovremmo digitare:

```
help set nomeopzione.
```

### 1.3 Salvare un grafico

Una volta ottenuto un grafico, magari dopo aver dato molte opzioni, potremmo avere la necessità di memorizzarlo per poterlo successivamente modificare. Ciò può essere eseguito con il comando:

```
save nomefile.plt
```

il questo modo chiuso il programma potremmo successivamente richiamarlo utilizzando il comando:

```
load nomefile.plt
```

ed operare ulteriori modifiche su di esso. Si faccia attenzione che i file devono essere accessibili da gnuplot, in caso contrario dovremmo anche inserire la directory in cui esso è memorizzato.

## 2 Specificazione delle funzioni

### 2.1 Introduzione

Per tracciare il grafico di una funzione a una o due variabili sarà necessario prima di tutto specificare tramite una formula la funzione da tracciare. In questo paragrafo vedremo come effettuare tale operazione.

### 2.2 Come specificare una funzione

La specificazione di una funzione è una operazione molto semplice in quanto per la specificazione basterà rispettare la normale sintassi delle espressioni algebriche. In ogni caso dobbiamo fare le seguenti precisazioni:

- la moltiplicazione è denotata con `*`
- la divisione è denotata con `/`
- l'elevamento a potenza è denotato con `**`
- $\pi$  è denotato con `pi`

Inoltre per quanto riguarda l'uso delle variabili indipendenti:

- per tracciare grafici in due dimensioni la variabile indipendente sarà sempre la `x`
- per tracciare grafici in tre dimensioni le variabili indipendenti saranno sempre `x,y`

Alcuni esempi di specificazione di funzioni sono i seguenti:

- $f(x)=1/x$
- $f(x,y)=x^{**2}+y^{**2}$
- $g(x,y)=\sin(x)-\cos(y)$

### 2.3 Funzioni predefinite

Il programma permette nella specificazione di una funzione, la possibilità di servirsi di una serie di funzioni già predefinite. L'elenco completo di tali funzioni si ottiene con il comando:

```
help functions
```

e successivamente digitando il nome di una delle funzioni visualizzate si accede all'help corrispondente.

### 2.4 Definire le costanti

Può accadere di dover utilizzare nello specificare una funzione delle costanti prestabilite. Possiamo definire una costante tramite la seguente sintassi:

```
nomecostante=valorecostante
```

successivamente il `nomecostante` potrà essere inserito nelle funzioni create al posto di inserire il `valorecostante`. Un uso più interessante delle costanti è invece dato dal seguente esempio. Si supponga di definire la funzione:

```
f(x,mu,sigma)=1/(sqrt(2*pi)*sigma)*exp(-(x-mu)**2/(2*sigma**2))
```

Così facendo possiamo ottenere funzioni diverse al variare dei parametri in essa inserita come ad esempio

```
f(x,1,6)
```

```
f(x,1,1)
```

e questo fatto ci permette di ottenere grafici in modo più veloce.

### 2.5 Specificare una funzione pice-wise

Molte volte accade che la funzione di cui vogliamo ottenere il grafico si componga da formule che sono diverse al variare del valore della variabile indipendente  $x$ . In gnuplot è possibile inserire tali tipi di funzioni utilizzando la sintassi seguente:

```
nome(x)=(condizione su x)?:primaformula:secondaformula
```

Ad esempio se dobbiamo specificare la seguente funzione:

$$f(x) = \begin{cases} x(1-x) & x < 1 \\ x-1 & -1 \leq x \leq 4 \\ \sqrt{x}+1 & x > 4 \end{cases}$$

possiamo utilizzare i seguenti comandi:

```
f1(x)=(x<-1) ? x*(1-x) : x-1
```

```
f2(x)=(x<=4) ? f1(x) : sqrt(x)+1
```

Quando tratteremo il grafico di questa funzione vedremo che il programma cercherà di unire con una linea i vari tronconi della funzione. Se vogliamo invece che ci sia uno stacco tra i vari pezzi conviene definire le funzioni separatamente nel seguente modo:

```
f1(x)=(x<-1) ? x*(1-x) : sqrt(-1)
```

```
f2(x)=(x>=-1 && x<=4) ? x-1 :sqrt(-1)
```

```
f3(x)=(x>4) ? sqrt(x)+1 : sqrt(-1)
```

### 2.6 Gli operatori unari

Nel definire le funzioni pice-wise può essere necessario utilizzare anche operatori unari. Esistono:

- - operatore meno: -a
- + operatore più: +a
- ~ operatore complemento: ~a
- ! operatore negazione: !a
- ! operatore fattoriale a!

### 2.7 Gli operatori binari

Nel definire le funzioni pice-wise può essere necessario utilizzare anche operatori binari. Esistono:

- \% operatore modulo: a%b
- == operatore di uguaglianza a==b
- != operatore di disuguaglianza a!=b

- < operatore di disuguaglianza  $a < b$
- <= operatore di disuguaglianza  $a \leq b$
- > operatore di disuguaglianza  $a > b$
- >= operatore di disuguaglianza  $a \geq b$
- && operatore AND
- || operatore OR

### 3 I grafici delle funzioni a due dimensioni

#### 3.1 Introduzione

Tramite gnuplot è possibile tracciare molte tipologie di grafici relativi alle funzioni a due dimensioni. Sarà infatti possibile ottenere:

- grafici cartesiani
- grafici parametrici
- grafici polari

#### 3.2 I grafici cartesiani

Definita come visto nel paragrafo 2 la funzione  $f(x)$ , per ottenere il suo grafico cartesiano utilizziamo il comando `plot` nel seguente modo:

```
plot f(x)
```

Se vogliamo invece ottenere il grafico cartesiano simultaneo delle funzioni  $f_1(x), f_2(x), \dots$  utilizziamo la seguente sintassi:

```
plot f1(x), f2(x), ...
```

Notiamo per inciso che per tracciare il grafico di una funzione, gnuplot ci consente di utilizzare una sintassi più semplificata, è infatti possibile evitare di definire preventivamente una funzione e di scriverne la formula dopo il comando `plot`. Ad esempio con:

```
plot sin(x), cos(x)
```

viene tracciato il grafico simultaneo delle funzioni seno e coseno.

#### 3.3 Specificare il dominio e il codominio

Il comando `plot` traccia automaticamente il grafico della funzione per  $x \in [-10; 10]$  e adattando l'asse delle  $y$  ai valori così ottenuti. Molte volte abbiamo però la necessità di tracciare il grafico per valori di  $x$  diversi da quelli di default. Se  $x_1, x_2, y_1, y_2$  sono numeri reali un primo modo per ottenere il risultato voluto è quello di utilizzare `plot` con la seguente sintassi:

```
plot [x1:x2] [y1:y2] f(x) viene specificato sia dominio che codominio
```

```
plot [x1:x2] f(x) viene specificato solo il dominio
```

```
plot [] [y1:y2] f(x) viene specificato solo il codominio
```

In questo modo i valori specificati per il dominio e codominio valgono solo per il grafico che stiamo tracciando. Se vogliamo mantenere questi valori validi anche per i grafici successivi dobbiamo prima di tracciare il grafico fissare gli intervalli voluti con i seguenti comandi:

`set xrange[x1:x2]` per fissare l'intervallo del dominio  
`set yrange[y1:y2]` per fissare l'intervallo del codominio

Si noti che se fissiamo l'intervallo del dominio sarà opportuno utilizzare anche l'opzione:

`set autoscale y`

Questi valori resteranno fissi per ogni grafico a due dimensioni che tratteremo successivamente.

#### 3.4 I grafici parametrici bidimensionali

In gnuplot è possibile tracciare grafici parametrici bidimensionali utilizzando il comando:

`set parametric`

per ripassare alle coordinate cartesiane sarà sufficiente utilizzare il comando:

`set noparametric`

Se vogliamo per esempio tracciare il grafico parametrico formato dalle equazioni:

$$\begin{cases} x = 5 \cos(t) \\ y = 2 \sin(t) \end{cases}$$

dobbiamo dare i seguenti comandi:

`set parametric`

`set xrange[-6:6]`

`set yrange[-6:6]`

`set trange[0:10]`

`set isosamples 60`

`plot 5*cos(t),2sin(t)`

Si noti come sia possibile specificare separatamente il range dell'asse delle x, dell'asse delle y e della variabile t.

#### 3.5 I grafici polari bidimensionali

In gnuplot è possibile tracciare grafici polari bidimensionali utilizzando il comando:

`set polar`

per ripassare alle coordinate cartesiane sarà sufficiente utilizzare il comando:

`set nopolar`

Se vogliamo per esempio tracciare il grafico polare di:

$$3 \cos(2t)$$

dobbiamo dare i seguenti comandi:

`set polar`

`set xrange[-6:6]`

`set yrange[-6:6]`

`set trange[0:6]`

`plot 3*cos(2*t)`

Si noti come sia possibile specificare separatamente il range dell'asse delle x, dell'asse delle y e della variabile t. Si noti che per default gli angoli sono interpretati in radianti. Per cambiarne la misurazione in gradi si usa l'opzione:

`set angles degrees`

per ripassare quindi alla misurazione in radianti si usa l'opzione:

`set angles radians`

## 4 I grafici delle funzioni a tre dimensioni

### 4.1 Introduzione

Tramite gnuplot è possibile tracciare molte tipologie di grafici relativi alle funzioni a due dimensioni. Sarà infatti possibile ottenere:

- grafici cartesiani
- grafici parametrici

### 4.2 I grafici cartesiani

Definita come visto nel paragrafo 2 la funzione  $f(x,y)$ , per ottenere il suo grafico cartesiano utilizziamo il comando `splot` nel seguente modo:

```
splot f(x,y)
```

Se vogliamo invece ottenere il grafico cartesiano simultaneo delle funzioni  $f_1(x,y), f_2(x,y), \dots$  utilizziamo la seguente sintassi:

```
splot f1(x,y), f2(x,y), ...
```

Notiamo per inciso che per tracciare il grafico di una funzione, gnuplot ci consente di utilizzare una sintassi più semplificata, è infatti possibile evitare di definire preventivamente una funzione e di scriverne la formula dopo il comando `splot`. Ad esempio con:

```
splot sin(x+y), cos(x-y)
```

viene tracciato il grafico simultaneo delle funzioni seno e coseno.

### 4.3 Specificare il dominio e il codominio

Il comando `splot` traccia automaticamente il grafico della funzione per  $x \in [-10; 10]$  e per  $y \in [-10; 10]$  e adattando l'asse delle  $z$  ai valori così ottenuti. Molte volte abbiamo però la necessità di tracciare il grafico per valori di  $x$  e di  $y$  diversi da quelli di default. Se  $x_1, x_2, y_1, y_2, z_1, z_2$  sono numeri reali un primo modo per ottenere il risultato voluto è quello di utilizzare `splot` con la seguente sintassi:

```
splot [x1:x2] [y1:y2] [z1:z2] f(x,y) viene specificato sia dominio che codominio
```

```
splot [x1:x2] [y1:y2] f(x) viene specificato solo il dominio
```

```
splot [] [] [z1:z2] f(x) viene specificato solo il codominio
```

In questo modo i valori specificati per il dominio e codominio valgono solo per il grafico che stiamo tracciando. Se vogliamo mantenere questi valori validi anche per i grafici successivi dobbiamo prima di tracciare il grafico fissare gli intervalli voluti con i seguenti comandi:

```
set xrange [x1:x2] per fissare l'intervallo del dominio
```

```
set yrange [y1:y2] per fissare l'intervallo del dominio
```

```
set zrange [z1:z2] per fissare l'intervallo del codominio
```

Si noti che se fissiamo l'intervallo del dominio sarà opportuno utilizzare anche l'opzione:

```
set autoscale z
```

Questi valori resteranno fissi per ogni grafico a due dimensioni che tratteremo successivamente.

### 4.4 L'opzione `hidden3d`

Tracciato il grafico di una funzione a due variabili può essere conveniente usare l'opzione `hidden3d`. Tale opzione serve per eliminare la trasparenza del grafico ottenuto. La sua sintassi è la seguente:

```
set hidden3d
replot
per annullare l'effetto possiamo utilizzare l'opzione:
set nohidden3d
replot
```

#### 4.5 L'opzione isosamples

Tracciato il grafico di una funzione a due variabili può essere conveniente usare l'opzione `isosamples`. Tale opzione serve per aumentare la precisione del grafico ottenuto. La sua sintassi è la seguente:

```
set isosamples x_rate, y_rate
replot
```

in cui `x_rate`, `y_rate` sono numeri interi che rappresentano la precisione voluta.

#### 4.6 L'opzione contour

Tracciato il grafico di una funzione a due variabili può essere conveniente usare l'opzione `contour`. Tale opzione serve per generare delle linee alla base o sulla superficie del grafico ottenuto. La sua sintassi è la seguente:

```
set contour base traccia delle linee alla base del grafico
set contour surface traccia delle linee lungo la superficie del grafico
set contour both traccia delle linee sia alla base che sulla superficie del grafico
replot
```

si noti che `set contour surface` non è disponibile se è attiva l'opzione `set hidden3d`. Per annullare l'effetto possiamo utilizzare l'opzione:

```
set nosurface
replot
```

#### 4.7 L'opzione cntrparam

Tracciato il grafico di una funzione a due variabili può essere conveniente usare l'opzione `cntrparam`. Tale opzione serve per generare delle linee alla base o sulla superficie del grafico ottenuto. La differenza rispetto alla funzione `contour` sta nel fatto di avere a disposizione molti più parametri da variare e quindi può offrire grafici più interessanti. Prima di usare tale opzione dovremo aver usato una delle seguenti opzioni `set contour base`

```
set contour surface
set contour both
```

La sua sintassi è la seguente:

```
set cntrparam { {linear | cubicspline | bspline}
                { points <n> } { order <n> }
                { levels auto {<n>} | <n>
                  | discrete <z1> {,<z2>{,<z3>...}}
                  | incremental <start>, <incr> {,<end>}
                }
            }
```

Ad esempio con la seguente sintassi:

```
set contour base
```

```
set cntrparam levels incremental 0,10,200
replot
```

vengono tracciate 20 linee di livello alla base della funzione ai livelli 0,10,...,200. Se avessimo voluto tali linee sulla superficie avremmo dovuto inserire come prima linea:

```
set contour surface
```

### 4.8 Cambiare la visualizzazione di un grafico

Per cambiare la visualizzazione di un grafico possiamo utilizzare l'opzione:

```
set view angolo_orrizzontale,angolo_verticale,zoom.
```

Si notino le seguenti particolarità:

```
set view angolo_orrizzontale
```

 cambia solo l'angolo orrizzontale

```
set view ,angolo_verticale
```

 cambia solo l'angolo verticale

```
set view ,,zoom
```

 cambia solo lo zoom

### 4.9 I grafici parametrici tridimensionali

In gnuplot è possibile tracciare grafici parametrici tridimensionali utilizzando il comando:

```
set parametric
```

per ripassare alle coordinate cartesiane sarà sufficiente utilizzare il comando:

```
set noparametric
```

Se vogliamo per esempio tracciare il grafico parametrico formato dalle equazioni:

$$\begin{cases} x = 2u \\ y = u^2 + v \\ z = v^2 \end{cases}$$

dobbiamo dare i seguenti comandi:

```
set parametric
```

```
set xrange[0:30]
```

```
set yrange[0:30]
```

```
set zrange[0:80]
```

```
set vrange[-10:10]
```

```
set urange[-7:7]
```

```
splot 2*u,u**2+v,v**2
```

Si noti come sia possibile specificare separatamente il range dell'asse delle x, dell'asse delle y e z e delle variabili u e v. Inoltre in tale tipo di grafico sarà possibile usare nel modo visto in precedenza anche le opzioni:

- hidden3d
- isosamples
- contour
- cntrparam

## 4.10 I grafici tridimensionali in coordinate sferiche e cilindriche

Per ottenere tali tipi di grafici dobbiamo innanzitutto selezionare l'opzione

```
set polar
```

quindi digitare l'opzione:

```
set mapping spherical o set mapping cylindrical
```

e quindi avere a disposizione un file `dati.dat` a tre colonne con le coordinate sferiche o cilindriche della funzione da disegnare. Il grafico avverrà nel modo precisato per paragrafo 5.3. Si veda anche l'help in linea di `set mapping`.

## 5 Grafici di dati

### 5.1 Introduzione

Accade molte volte di avere a disposizione dei dati derivanti ad esempio da una analisi statistica o da un esperimento e di dover predisporre un grafico di questi dati. Supporremo che tali dati siano memorizzati nel file `dati.dat` raggiungibile da `gnuplot`, in caso contrario nel richiamare il file dovremmo specificare la directory in cui esso è memorizzato. Si noti che il file `dati.dat` può essere composto da più colonne e il separatore di colonna deve essere `tab`.

### 5.2 Grafico di dati a due dimensioni

Se il file `dati.dat` è composto solo da due colonne con per ottenere il grafico per punti dei dati desiderati sarà sufficiente digitare il comando:

```
plot "dati.dat"
```

la prima colonna individua la variabile indipendente mentre la seconda quella dipendente. Se nel file ci sono più colonne sarà necessario specificare quali colonne usare ed indicare anche la variabile dipendente. Ciò è fatto con il comando:

```
plot "dati.dat" using a:b
```

in cui `a` indica il numero della colonna della variabili indipendente mentre `b` indica la colonna della variabile dipendente.

Il grafico predefinito che si ottiene è uno scatterplot bidimensionale anche se è possibile come vedremo successivamente precisare stili diversi.

### 5.3 Grafico di dati a tre dimensioni

Se il file `dati.dat` è composto solo da tre colonne con per ottenere il grafico per punti dei dati desiderati sarà sufficiente digitare il comando:

```
splot "dati.dat"
```

la prima e la seconda colonna individuano le variabili indipendenti mentre la terza quella dipendente. Se nel file ci sono più colonne sarà necessario specificare quali colonne usare ed indicare anche le variabili dipendenti. Ciò è fatto con il comando:

```
plot "dati.dat" using a:b:c
```

in cui `a` e `b` indicano il numero delle colonne delle variabili indipendenti mentre `c` indica la colonna della variabile dipendente.

Il grafico predefinito che si ottiene è uno scatterplot tridimensionale anche se è possibile come vedremo successivamente precisare stili diversi.

## 6 Aspetti comuni

### 6.1 Introduzione

In questo paragrafo vedremo una serie di comandi che potranno essere applicati a qualunque tipo di grafico che vogliamo tracciare sia esso bidimensionale o tridimensionale. Si tratta di una semplice elencazione di comandi accompagnata dalle principali opzioni circa il loro uso. Per un aiuto completo di veda sempre l'help in linea richiamabile con il comando `help set nomecomando`.

### 6.2 Unità di misura degli angoli

La misura predefinita per gli angoli è il radiante. Se si desidera cambiare unità di misura in gradi dobbiamo dare il seguente comando:

```
set angles degrees
```

per ritornare successivamente alla misura in radianti daremo il seguente comando:

```
set angles radians
```

per vedere l'unità di misura impostata si usa il comando:

```
show angles.
```

### 6.3 Usare la scala logaritmica

Per impostare la scala logaritmica si usa il comando:

```
set logscale asse base
```

in cui asse è `x,y,z` o qualunque combinazione due a due di questi valori, e base è la base logaritmica prescelta.

Per eliminare la scala logaritmica si usa il comando:

```
set nologscale asse
```

in cui asse è `x,y,z` o qualunque combinazione due a due di questi valori.

Per eliminare tutte le scale logaritmiche impostate si usa il comando:

```
set nologscale
```

mentre per vedere la situazione delle scale logaritmiche impostate si usa il comando:

```
show logscale.
```

Per ulteriori precisazioni si veda l'help in linea richiamabile con `help set logscale`

### 6.4 Inserire la griglia nel grafico

Per inserire la griglia nel grafico si usa il comando:

```
set grid
```

per eliminarla il comando

```
set nogrid
```

 L'ampiezza delle maglie della griglia verrà discussa nel paragrafo relativo ai tics.

### 6.5 Ottenere più grafici sullo stesso diagramma

Abbiamo notato che utilizzando i comandi:

```
plot f(x),g(x)
```

```
splot f(x,y),g(x,y)
```

vengono inseriti nello stesso diagramma più grafici di funzioni. Un modo alternativo per ottenere lo stesso risultato e che consente di tracciare anche più grafici parametrici in uno stesso diagramma è il seguente:

```
plot f(x)
replot g(x)
```

in questo modo al primo grafico tracciato viene aggiunto anche quello che compare dopo il comando replot.

## 6.6 Modificare lo stile dei grafici a due dimensioni

Lo stile predefinito per il tracciamento dei grafici di gnuplot può essere modificato in modo diverso a seconda si stia tracciando il grafico di funzioni o il grafico di dati. I comandi di modifica dello stile sono i seguenti:

- `set function style stilelinea` per i grafici di funzioni
- `set data style stilelinea` per i grafici di dati

Per vedere gli stili impostati si usa invece il comando:

- `show function style` per i grafici di funzioni
- `show data style` per i grafici di dati

Il tipo di stile di linea che possiamo utilizzare è il seguente:

- `lines`
- `points`
- `linespoints`
- `impulses`
- `dots`
- `steps`
- `errorbars` (utilizzabile solamente per grafici di dati vedere l'help in linea)
- `boxes`

che sono di facile interpretazione.

## 6.7 Modificare lo stile dei grafici a tre dimensioni

Per modificare lo stile dei grafici a tre dimensioni vale quanto detto parlando della modifica dello stile dei grafici a due dimensioni con la sola variante che gli stili di linea utilizzabili sono solo i seguenti:

- `lines`
- `points`
- `linespoints`
- `impulses`

che sono di facile interpretazione.

## 6.8 Modifica dello stile di linea funzione per funzione

Nell'ipotesi di dover tracciare più grafici di funzione in uno stesso diagramma, potrebbe accadere che gli stili richiesti siano diversi. Per tracciare quindi i grafici con stili valevoli solo per la funzione indicata si usano i seguenti comandi:

```
plot f(x) with stilelinea
plot f(x) with stilelinea, g(x) with stilelinea
plot "data.dat" with stilelinea
plot "data.dat" with stilelinea, "dati2.dat" with stilelinea
plot "data.dat" using a:b with stilelinea
```

in cui stilelinea è una delle opzione viste precedentemente.

## 6.9 Valutare espressioni numeriche

In gnuplot è anche possibile effettuare una valutazione di espressioni numeriche utilizzando il comando `print`. Un esempio dell'utilizzo di tale comando è il seguente:

```
print sin(2)
```

che restituisce il valore del seno del numero due.

## 6.10 Cambiare il nome delle variabili

Come abbiamo visto precedentemente, le variabili usate per default da gnuplot sono le seguenti:

- `x,y,z` per i grafici cartesiani
- `t` per grafici parametrici bidimensionali
- `u,v` per grafici parametrici tridimensionali

Può essere necessario in alcuni casi variare il nome predefinito di tali variabili e ciò è fatto tramite la funzione `dummy`. Il suo utilizzo è il seguente:

1. `set dummy h` modifica la variabile `x` in `h`
2. `set dummy h,k` modifica le variabili `x` ed `y` in `h` e `k`

in modo analogo possiamo modificare i nomi di default per le altre variabili.

## 6.11 Modificare le etichette degli assi

Per modificare le etichette degli assi utilizziamo il seguente comando:

```
set xlabel "etichetta" a,b scrive etichetta sull'asse delle x spostata di a carattere e di b righe
set ylabel "etichetta" a,b scrive etichetta sull'asse delle y spostata di a carattere e di b righe
set zlabel "etichetta" a,b scrive etichetta sull'asse delle z spostata di a carattere e di b righe
```

Si noti che se

- `a` è positivo lo spostamento avviene verso destra se negativo verso sinistra
- `b` è positivo lo spostamento avviene verso l'alto se negativo verso il basso

Analogo a tale comando è anche il comando `x2label` e `y2label` i quali operano nello stesso modo di `xlabel` e `ylabel` ma in questo caso le etichette sono scritte rispettivamente sopra e a destra del grafico.

Per togliere le etichette precedentemente assegnate si usa:

```
set xlabel
set ylabel
set zlabel
```

## 6.12 Visualizzazione degli assi cartesiani

Quando tracciamo un grafico, gli assi cartesiani non sono visualizzati. Per permettere al programma di tracciare gli assi possiamo dare i seguenti comandi:

```
set zeroaxis permette il tracciamento di tutti gli assi cartesiani possibili
set xzeroaxis permette il tracciamento dell'asse delle x
set yzeroaxis permette il tracciamento dell'asse delle y
```

Per eliminare il tracciamento degli assi cartesiani non voluti possiamo utilizzare i comandi:

```
set nozeroaxis impedisce il tracciamento di tutti gli assi cartesiani possibili
set noxzeroaxis impedisce il tracciamento dell'asse delle x
set noyzeroaxis impedisce il tracciamento dell'asse delle y
```

## 6.13 Eliminare la visualizzazione del bordo

Per impedire la visualizzazione del bordo del grafico utilizziamo il comando:

```
set noborder
```

per reinserire la visualizzazione digitiamo:

```
set border.
```

## 6.14 Visualizzazione della legenda

Quando tracciamo uno o più grafici, il programma visualizza una legenda delle funzioni tracciate. Per impedire tale visualizzazione possiamo utilizzare il comando:

```
set nokey
```

per reinserirla digiteremo:

```
set key
```

E' anche possibile ottenere una legenda con i nomi personalizzati, si veda a proposito l'help in linea.

## 6.15 Numero dei tics del grafico

I tics sono i punti in cui sono visualizzati i numeri nei vari assi. La posizione di tali tics è stabilita dal programma di default ma può essere modificata a piacimento. Per eliminare la visualizzazione dei tics possiamo digitare i comandi:

```
set noxtics
set noytics
set noztics
```

Per reinserire i tics si potrà digitare il comando:

```
set xtics
set ytics
```

`set ztics`

è possibile ottenere dei tics non i nomi dei mesi o i giorni della settimana. Si veda a tal proposito l'help in linea.

### 6.16 Particolarità di `xtics` ed `ytics`

I comandi `xtics` e `ytics` presentano delle particolarità di notevole interesse le quali permettono di tracciare i tics in modo molto interattivo specificando anche se necessario uno a uno i numeri su cui tracciare tali tics. La sintassi completa di tali comandi è la seguente:

```
set xtics (ytics) {axis | border} {{no}mirror} {{no}rotate}
    { autofreq
      | <incr>
      | <start>, <incr> {,<end>}
      | ({"<label>"} <pos> {,{"<label>"} <pos>}...) }
```

Si noti che se si specifica l'opzione che permette la visualizzazione della griglia nel grafico, essa verrà tracciata in corrispondenza dei tics tracciati.

### 6.17 Aggiungere un titolo

Per aggiungere un titolo ad un grafico si usa il comando:

```
set title "titolodelgrafico"
```

é possibile anche specificare la posizione di tale titolo, si veda a proposito l'help in linea.

### 6.18 Aggiungere etichette al grafico

Una delle potenzialità più elevate di `gnuplot` è quella di permettere di aggiungere delle etichette in qualunque punto del grafico. Ciò può essere effettuato con il comando:

```
set label n "etichetta" at a,b,c
```

in cui:

- `n` rappresenta il numero dell'etichetta utile per il comando `nolabel`
- `a,b,c` sono le coordinate in cui vogliamo che l'etichetta sia inserita, se il grafico è bidimensionale non deve essere specificato il valore di `c`

Per levare una etichetta da un grafico si usa il comando:

```
set nolabel n
```

in cui `n` è il numero dell'etichetta stampata in precedenza.

### 6.19 Aggiungere una freccia

E' possibile inserire anche una freccia in un grafico tramite il comando `set arrows` si veda a tal proposito l'help in linea.

## 6.20 Elenco completo dei comandi

Diamo in conclusione di questa sezione un elenco completo dei comandi che possono essere dati con l'opzione `set`, esso è il seguente:

```
set
^
valid set options:  [] = choose one, {} means optional

'angles', '{no}arrow', '{no}autoscale', 'bars', '{no}border',
'boxwidth', '{no}clabel', '{no}clip', 'cntrparam', '{no}contour',
'data style', '{no}dgrid3d', 'dummy', 'encoding', 'format',
'function style', '{no}grid', '{no}hidden3d', 'isosamples',
'{no}key', '{no}label', '{no}linestyle', 'locale', '{no}logscale',
'[blrt]margin', 'mapping', 'missing', '{no}multiplot', 'offsets',
'origin', 'output', '{no}parametric', 'pointsize', '{no}polar',
'[rtuv]range', 'samples', 'size', '{no}surface', 'terminal',
'tics', 'ticscale', 'ticslevel', '{no}timestamp', 'timefmt',
'title', 'view', '[xyz]{2}data', '[xyz]{2}label', '[xyz]{2}range',
'{no}{m}[xyz]{2}tics', '[xyz]{2}[md]tics', '{no}{[xyz]{2}}zeroaxis',
'zero'
```

Per ogni opzione tramite `help set nomecomando` è possibile ottenere l'help in linea completo del comando.

## 7 Grafici multipli

### 7.1 Introduzione

In certi casi sarà necessario ottenere grafici multipli, ossia diagrammi con più grafici visualizzati separatamente. Ciò può essere effettuato con il comando `multiplot`.

### 7.2 Come ottenere grafici multipli

Per ottenere grafici multipli possiamo operare nel seguente modo:

```
set multiplot
set size 0.5,0.5
set origin 0.5,0.5
set title 'Funzione potenza con esponente pari'
plot x**2,x**4,x**8
set size 0.5,0.5
set origin 0.3,0.6
set title ' Funzione potenza con esponente dispari'
plot x**3,x**5,x**7,x**9
```

# tutte le opzioni specifiche per ogni grafico devono essere date dopo `set origin`  
#e prima di `plot`. Tutte le opzioni prima di `multiplot` valgono per tutti i grafici

```
#che stamperemo dopo.
#set size a,b          a individua la larghezza del grafico
#                      b individua la lunghezza
#set origin a,b        con l'aumento di a da 0 a 1 spostiamo il grafico verso destra
#                      con l'aumento di b da 0 a 1 spostiamo il grafico verso l'alto
#ci sono inoltre i seguente casi particolari
#set origin 0,0.5      per stampare il grafico nella prima riga prima colonna
#set origin 0.5,0.5    per stampare il grafico nella prima riga seconda colonna
#set origin 0,0        per stampare il grafico nella seconda riga prima colonna
#set origin 0.5,0      per stampare il grafico nella seconda riga seconda colonna
```

## 8 Output dei grafici

### 8.1 Introduzione

Quando si accede al programma, l'output predefinito è il video. Se vogliamo però stampare il grafico o salvarlo in qualche file per essere importato in altri documenti dobbiamo modificare l'output. In questo paragrafo faremo prevalentemente riferimento ai salvataggi di grafici in formato `postscript` e `latex`.

### 8.2 I terminali

I terminali sono i device grafici utilizzati da `gnuplot`. Tramite i terminali riusciamo a visualizzare il grafico su video o possiamo stamparlo su di un file in un formato desiderato. Il terminale predefinito dal programam è `x11` ossia il video. Per modificare il terminale usiamo il comando: `set terminal nometerminale` in cui nome terminale è il nome del terminale da utilizzare. Un elenco dei terminali utilizzabili si ottiene con `help set terminal`.

### 8.3 Stampare un grafico

Per reindirizzare l'output dallo schermo alla stampante dovremmo dare il comando:

```
set output "|lp"
```

per ulteriori precisazioni si veda `help set output`

### 8.4 Salvataggio in un file ps

Se si usa `LATEX` per importare grafici nei documenti sarà necessario salvare il grafico in un file `ps`. Per operare questo tipo di salvataggio dobbiamo dare i seguenti comandi:

```
set output "filenamse.ps" per specificare il nome del file ove salvare il grafico
```

```
set terminal postscript per specificare il tipo di terminale da usare
```

```
replot per stampare su file il grafico ottenuto a video precedentemente
```

Il file così ottenuto sarà importato in `latex` nel consueto modo.

### 8.5 Salvataggio in un file eps

In `LATEX` possiamo usare anche file di tipo `eps`. Per permettere a `gnuplot` di salvare un grafico in questo formato possiamo dare i seguenti comandi:

`set output“filenamse.eps“` per specificare il nome del file ove salvare il grafico  
`set terminal postscript eps` per specificare il tipo di terminale da usare  
`replot` per stampare su file il grafico ottenuto a video precedentemente  
Il file così ottenuto sarà importato in latex nel consueto modo.

### 8.6 Salvataggio in un file tex

Per importare un grafico in un documento tex abbiamo un modo alternativo di procedere che consiste nel salvare il grafico utilizzando il terminal latex. Il vantaggio di operare in questo modo è quello di poter inserire in tutti gli aspetti del grafico che richiedono un input di caratteri, espressioni in latex racchiuse dai segni di dollaro. In questo modo potremmo importare il grafico in latex come se si trattasse di un qualsiasi documento. Per effettuare tale esportazione dobbiamo dare i seguenti comandi:

`set output“filenamse.tex“` per specificare il nome del file ove salvare il grafico  
`set terminal latex` per specificare il tipo di terminale da usare  
`replot` per stampare su file il grafico ottenuto a video precedentemente. Il file così ottenuto sarà importato in latex nel consueto modo.

Un modo alternativo di procedere è quello di usare i seguenti comandi:

`set output“filenamse.tex“` per specificare il nome del file ove salvare il grafico  
`set terminal pslatex` per specificare il tipo di terminale da usare  
`replot` per stampare su file il grafico ottenuto a video precedentemente. In questo modo vengono creati due file, uno ps e l'altro latex. Il vantaggio di questo modo di procedere è quello relativo all'individuazione delle modifiche da effettuare più facilmente individuabili nel file latex rispetto al metodo precedente.

### 8.7 Opzioni per il terminal postscript

Vogliamo rimarcare alcune opzioni molto importanti che possono essere utilizzate con il terminal postscript. Esse sono:

`set terminal postscript color` crea un file postscript a colori  
`set terminal postscript solid` per usare linee nella stampa.  
`set terminal postscript enhanced` per utilizzare formule scritte in latex nei grafici  
altre opzioni si possono ottenere con `set terminal postscript`.

### 8.8 Output in formato diverso

Anche se con l'opzione terminal otteniamo grafici in vari formati, possiamo utilizzare gnuplot assieme ad un altro programma grafico molto potente ossia Image Magick. In questo modo possiamo aumentare notevolmente la classe dei formati in cui possiamo salvare i grafici ottenuti. Dobbiamo operare nel seguente modo:

- creare un grafico con gnuplot
- avviare Image Magick con il comando `display` da console di linux
- cliccare sulla lavagna di Image Magick
- scegliere file, open, grab, grab
- cliccare sul grafico generato con gnuplot

- utilizzando il menu di Image Magick salvare il grafico nel formato desiderato

## 9 Gli script

### 9.1 Introduzione

Per creare dei grafici complessi dovremmo dare come vedremo una serie di opzioni relative agli assi, ai titoli, al modo di visualizzare le linee. Dovremmo anche inserire legende o simboli all'interno del grafico stesso. Queste operazioni se digitate da console di gnuplot restano perse e non possono essere recuperate. E' quindi necesario poter creare degli script, ossia una sequeza di istruzioni, che possano essere memorizzate in un file di testo ed essere eseguite da gnuplot tutte assieme con un semplice comando.

### 9.2 Creare uno script

Per creare uno script sarà necessario creare un semplice file di testo ed in esso inserire in ogni riga una istrizione eseguibile da gnuplot. Ad esempio

```
#Grafico di un integrale
set xrange [-1:3]
set yrange [-1:4]
set zeroaxis
set xlabel 'x'
set ylabel 'f(x)'
set nokey
set label 1 'a' at 0.5,-0.1
set label 2 'b' at 2.5,-0.1
set noxtics
set noytics
f(x)=x<0.5 ? 1/0 : x>2.5 ? 1/0 : -(x-1)**2 +(x-1)**3+1
plot f(x) with lines , f(x) with impulse
set terminal postscript enhanced
set output "c:/integrale.ps"
replot
```

Si notino due cose importanti:

- si tratta di una serie di istruzioni che avremmo potuto inserire direttamente in gnuplot
- il # all'inizio di una riga è onsiderato commento e quindi non sarà eseguito

L'operazione successiva sarà quella di salvare questo script in un file ad esempio `nomescript.plt`.

### 9.3 Eseguire uno script

Per eseguire lo script ed ottenere quanto in esso digitato basterà eseguire il comando:

```
load "nomescript.plt"
```

Si noti che il file deve essere accessibile da gnuplot in caso contrario dovremmo inserire anche il percorso del file.

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Presentazione . . . . .	1
1.2	Come ottenere un aiuto su un comando . . . . .	1
1.3	Salvare un grafico . . . . .	1
<b>2</b>	<b>Specificazione delle funzioni</b>	<b>2</b>
2.1	Introduzione . . . . .	2
2.2	Come specificare una funzione . . . . .	2
2.3	Funzioni predefinite . . . . .	2
2.4	Definire le costanti . . . . .	2
2.5	Specificare una funzione pice-wise . . . . .	3
2.6	Gli operatori unari . . . . .	3
2.7	Gli operatori binari . . . . .	3
<b>3</b>	<b>I grafici delle funzioni a due dimensioni</b>	<b>4</b>
3.1	Introduzione . . . . .	4
3.2	I grafici cartesiani . . . . .	4
3.3	Specificare il dominio e il codominio . . . . .	4
3.4	I grafici parametrici bidimensionali . . . . .	5
3.5	I grafici polari bidimensionali . . . . .	5
<b>4</b>	<b>I grafici delle funzioni a tre dimensioni</b>	<b>6</b>
4.1	Introduzione . . . . .	6
4.2	I grafici cartesiani . . . . .	6
4.3	Specificare il dominio e il codominio . . . . .	6
4.4	L'opzione hidden3d . . . . .	6
4.5	L'opzione isosamples . . . . .	7
4.6	L'opzione contour . . . . .	7
4.7	L'opzione cntrparam . . . . .	7
4.8	Cambiare la visualizzazione di un grafico . . . . .	8
4.9	I grafici parametrici tridimensionali . . . . .	8
4.10	I grafici tridimensionali in coordinate sferiche e cilindriche . . . . .	9
<b>5</b>	<b>Grafici di dati</b>	<b>9</b>
5.1	Introduzione . . . . .	9
5.2	Grafico di dati a due dimensioni . . . . .	9
5.3	Grafico di dati a tre dimensioni . . . . .	9
<b>6</b>	<b>Aspetti comuni</b>	<b>10</b>
6.1	Introduzione . . . . .	10
6.2	Unità di misura degli angoli . . . . .	10
6.3	Usare la scala logaritmica . . . . .	10
6.4	Inserire la griglia nel grafico . . . . .	10
6.5	Ottenere più grafici sullo stesso diagramma . . . . .	10
6.6	Modificare lo stile dei grafici a due dimensioni . . . . .	11
6.7	Modificare lo stile dei grafici a due dimensioni . . . . .	11

6.8	Modifica dello stile di linea funzione per funzione . . . . .	12
6.9	Valutare espressioni numeriche . . . . .	12
6.10	Cambiare il nome delle variabili . . . . .	12
6.11	Modificare le etichette degli assi . . . . .	12
6.12	Visualizzazione degli assi cartesiani . . . . .	13
6.13	Eliminare la visualizzazione del bordo . . . . .	13
6.14	Visualizzazione della legenda . . . . .	13
6.15	Numero dei tics del grafico . . . . .	13
6.16	Particolarità di xtics ed ytics . . . . .	14
6.17	Aggiungere un titolo . . . . .	14
6.18	Aggiungere etichette al grafico . . . . .	14
6.19	Aggiungere una freccia . . . . .	14
6.20	Elenco completo dei comandi . . . . .	15
<b>7</b>	<b>Grafici multipli</b> . . . . .	<b>15</b>
7.1	Introduzione . . . . .	15
7.2	Come ottenere grafici multipli . . . . .	15
<b>8</b>	<b>Output dei grafici</b> . . . . .	<b>16</b>
8.1	Introduzione . . . . .	16
8.2	I terminali . . . . .	16
8.3	Stampare un grafico . . . . .	16
8.4	Salvataggio in un file ps . . . . .	16
8.5	Salvataggio in un file eps . . . . .	16
8.6	Salvataggio in un file tex . . . . .	17
8.7	Opzioni per il terminal postscript . . . . .	17
8.8	Output in formato diverso . . . . .	17
<b>9</b>	<b>Gli script</b> . . . . .	<b>18</b>
9.1	Introduzione . . . . .	18
9.2	Creare uno script . . . . .	18
9.3	Eseguire uno script . . . . .	18