

GNU PLOT - GUIDA OPERATIVA

A CURA DI ROBERTO RAINERI

**Liceo Erasmo da Rotterdam
Anno scolastico 2003/2004
CLASSE VA**

GENERALITÀ

Copyright (C) 1986 – 1993 Thomas Williams, Colin Kelley

L'autorizzazione a utilizzare, copiare e distribuire questo software e la sua documentazione per qualsiasi scopo è attraverso questo documento concessa, a patto che codesto avviso di diritti d'autore appaia non solo in tutte le copie ma anche negli avvisi di copyright. Questo avviso di permesso appare nella documentazione inclusa. L'autorizzazione a modificare il software è concessa, ma non la sua distribuzione con il codice modificato. Le modifiche devono essere distribuite come patches. Questo software è fornito "così come è" senza garanzia esplicita o implicita.

AUTORI

Software originale:

Thomas Williams, Colin Kelley.

Modifiche e miglioramenti di Gnuplot versione 2.0:

Russell Lang, Dave Kotz, John Campbell.

Modifiche e miglioramenti di Gnuplot versione 3.0:

Gershon Elber e altri.

Introduzione

GNU PLOT è un programma interattivo che permette di tracciare il grafico di funzioni matematiche attraverso comandi guidati. I comandi e i nomi delle funzioni scritti in minuscolo non equivalgono agli stessi scritti in maiuscolo, bisogna pertanto porre attenzione a scrivere i comandi nel modo corretto.

Tutti i nomi di comando possono essere abbreviati a patto che l'abbreviazione non sia ambigua. Qualsiasi numero di comandi può apparire sulla stessa linea, separato dal punto e virgola (;). I nomi di file sono indicati tra virgolette. Le virgolette possono essere singole o doppie.

Esempio

```
load "nome del file"
```

```
cd 'dir'
```

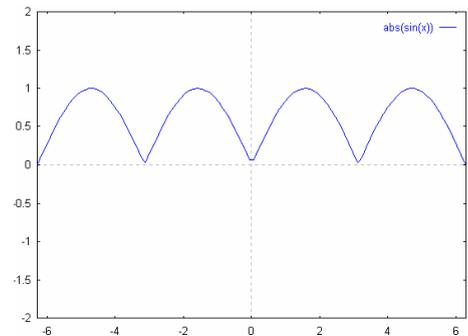
FUNZIONI

In generale, qualsiasi funzione o espressione matematica accettata da C, FORTRAN, Pascal o BASIC è valida. Lo spazio bianco (spazi e tabulazioni) è ignorato all'interno delle espressioni.

Abs

La funzione `abs` fornisce il valore assoluto del suo argomento.

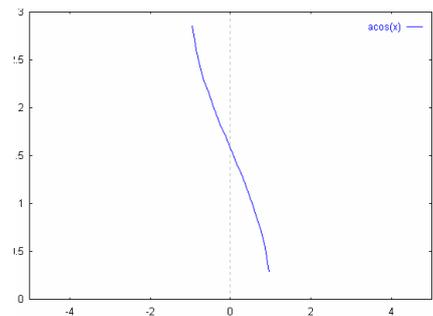
```
gnuplot>plot abs(sin(x))
```



Acos

La funzione `acos` calcola l'arcoseno (l'inverso del coseno) del suo argomento; `acos` restituisce il risultato in radianti.

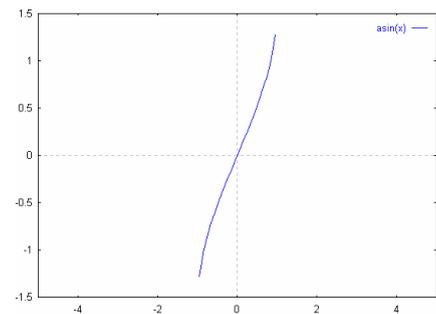
```
gnuplot>plot acos(x)
```



Asin

La funzione `asin` calcola l'arcoseno (l'inverso del seno) del suo argomento; `asin` restituisce il risultato in radianti.

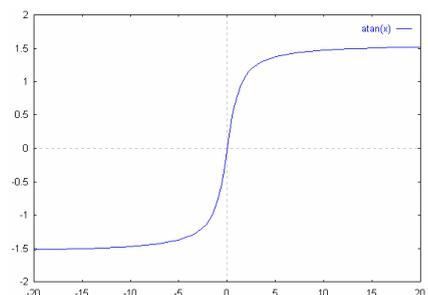
```
gnuplot>plot asin(x)
```



Atan

La funzione `atan` calcola l'arcotangente (inverso tangente) del suo argomento. `Atan` restituisce il risultato in radianti.

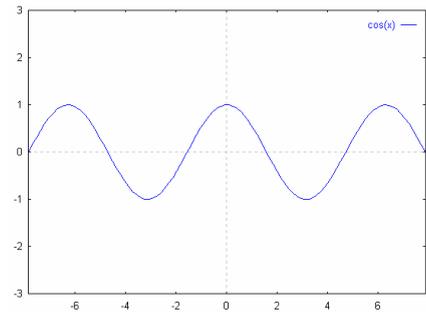
```
gnuplot>plot atan(x)
```



Cos

La funzione cos calcola il coseno del suo argomento; cos esige che il suo argomento sia in radianti (a meno di non aver predisposto Gnuplot per lavorare con gli angoli espressi in gradi).

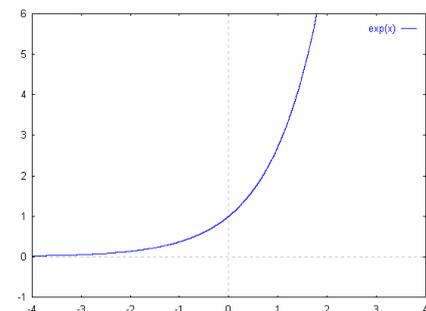
gnuplot>plot cos(x)



Exp

La funzione exp calcola l'esponenziale i base e (numero di Nepero) del suo argomento.

gnuplot>plot exp(x)



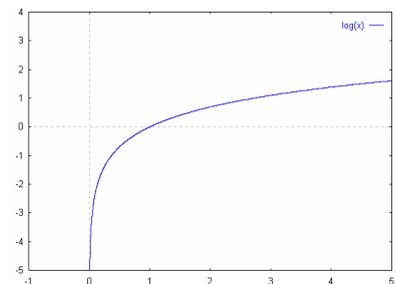
Int

La funzione int restituisce la parte intera del suo argomento approssimata per difetto.

Log

La funzione logaritmo calcola il logaritmo naturale (in base e) del suo argomento.

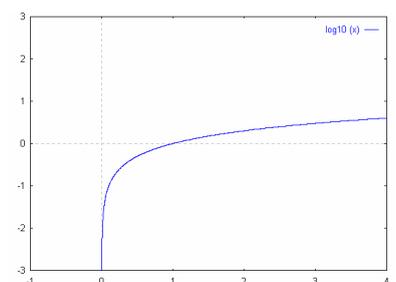
gnuplot>plot log(x)



Log10

La funzione log10 calcola il logaritmo in base 10 del suo argomento.

gnuplot>plot log10(x)



Rand

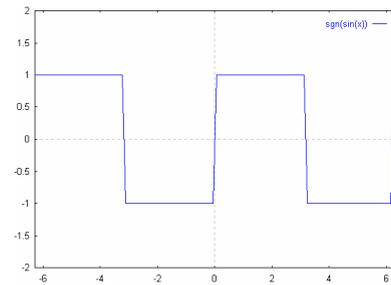
La funzione rand restituisce un numero casuale nell'intervallo [0:1]; necessita come argomento un numero reale positivo qualsiasi.

Sgn (funzione segno)

La funzione segno restituisce 1 se il suo argomento è positivo, -1 se il suo argomento è negativo, e 0 se il suo argomento è zero.

E' possibile tramite questa funzione studiare il segno di una qualsiasi altra funzione matematica.

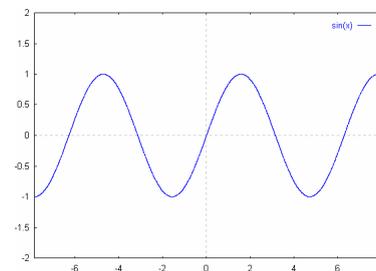
```
gnuplot> plot sgn (sin(x))
```



Sin

La funzione seno calcola il seno del suo argomento; sin esige che il suo argomento sia in radianti (a meno di non aver predisposto Gnuplot per lavorare con gli angoli espressi in gradi).

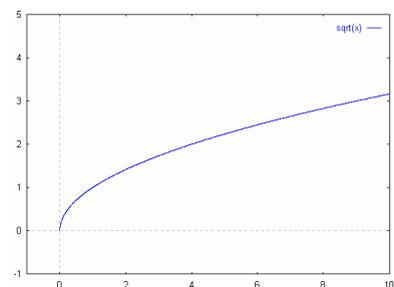
```
gnuplot>plot sin(x)
```



Sqrt

La funzione sqrt calcola la radice quadrata del suo argomento.

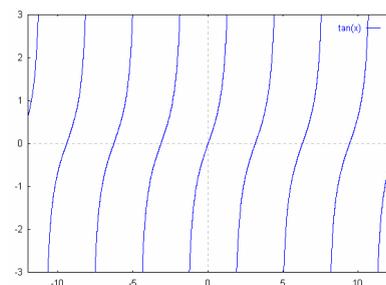
```
gnuplot>plot sqrt(x)
```



Tan

La funzione tan restituisce la tangente del suo argomento; tan esige che il suo argomento sia in radianti (a meno di non aver predisposto Gnuplot per lavorare con gli angoli espressi in gradi).

```
gnuplot>plot tan(x)
```



OPERATORI

Gli operatori in GNUPLOT sono gli stessi che nel linguaggio di programmazione C, salvo che in GNUPLOT gli operatori accettano argomenti interi, reali e complessi.

Operatori unari di interesse

Fattoriale di un numero

Il comando per ottenere il fattoriale di un numero è "a!". Esempio: per ottenere il fattoriale di 5:

```
print 5!
```

Operatori Binari: operatori a due argomenti, ad esempio la moltiplicazione.

La seguente è una tabella che mostra come ottenere operazioni binarie tipiche:

Simbolo	Esempio	Significato
**	a**b	elevamento a potenza
*	a*b	moltiplicazione
/	a/b	divisione
%	a%b	modulo: dà il resto della divisione tra due numeri*
+	a+b	addizione
-	a-b	sottrazione
=	a=b	uguaglianza
!=	a!=b	disuguaglianza (equivalente al simbolo ≠)

(*) l'operatore richiede argomento intero.

COMANDI PRINCIPALI

Cd

Il comando cd cambia la directory in cui si sta lavorando.

Sintassi:

```
cd "<nome-directory>"
```

Il nome della directory deve essere tra virgolette.

Esempi:

```
cd 'subdir'          cd ".."
```

Exit e quit

Con questi due comandi si esce da gnuplot.

Help

Il comando di help mostra l'help in linea. Per avere informazioni su un particolare argomento usare la sintassi

```
help {<argomento>}
```

Ad esempio digitando help title si ottiene l'help del comando title.

Plot e splot

Plot e splot sono i comandi principali del programma. Essi tracciano le funzioni e i dati in moltissimi modi. il comando plot è utilizzato per tracciare funzioni e grafici in 2 dimensioni, mentre splot traccia superfici e grafici in 3 dimensioni.

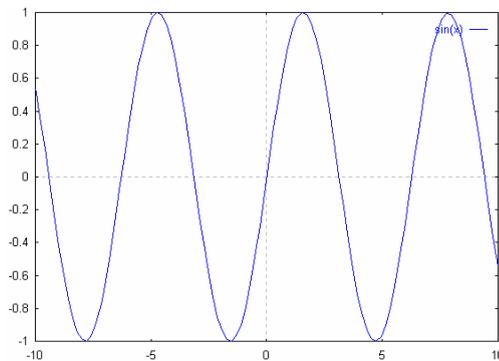
Sintassi:

```
plot {[estremi del grafico]} <funzione> {titolo "nome del titolo"}
```

```
splot {[estremi del grafico]} <funzione> {titolo "nome del titolo"}
```

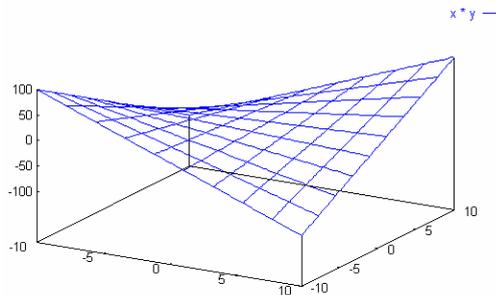
I comandi plot e splot possono essere semplici come

```
plot sin(x)
```



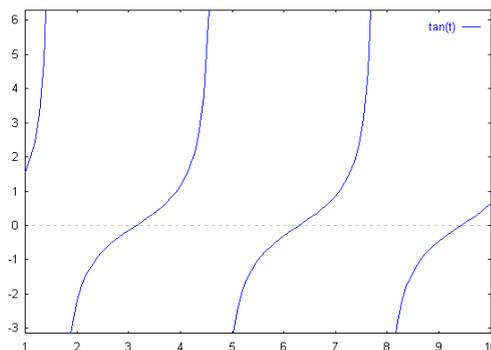
o

```
splot x * y
```



o complessi come

```
plot [t=1:10] [-pi:pi*2] tan(t)
```



Title

Il titolo di ogni comando plot appare nel grafico tracciato. Di solito tale titolo coincide con il nome della funzione. Il titolo può essere tuttavia cambiato utilizzando il comando title.

Sintassi:

```
title "<nome del titolo>"
```

dove <nome del titolo> è il nuovo titolo del grafico e deve essere racchiuso tra virgolette.

esempi:

Il comando seguente disegna il grafico della funzione $y=x$ con titolo 'x':

```
plot x
```

il comando seguente disegna il grafico della funzione $y = e^x$ con titolo “crescita esponenziale”

```
Plot exp(x) title “crescita esponenziale”
```

il titolo può essere omissso con l’opzione "notitle" sia per plot che per splot. Questo può essere utile quando molte curve vengono disegnate nello stesso piano cartesiano

Print

Il comando stampa a video il valore della <espressione>.

Sintassi:

```
print <espressione>
```

Esempio se si vuole il valore di π si digita:

```
print pi
```

Replot

Il comando replot senza argomenti ridisegna l'ultimo grafico tracciato.

Save

Il comando save permette di salvare funzioni, variabili, opzioni definite dall'utente, Sintassi:

```
save {<opzioni>} "<nome del file>"
```

Se nessuna opzione è indicata GNUPLOT salva le funzioni, le variabili, le opzioni di default e l'ultimo comando plot (splot).

I files salvati sono scritti in formato testo e possono essere letti dal comando load.

I nomi dei files devono essere racchiusi tra virgolette

Esempi:

```
save "work.gnu"  
save functions 'func.dat'  
save var 'var.dat'  
save set "options.dat"
```

Angles

Normalmente GNUPLOT si aspetta che gli angoli vengano espressi in radianti.

Sintassi:

```
set angles { gradi | radianti }  
show angles
```

Autoscale

La scala automatica del grafico può essere fissata per l'asse x, y o z o globalmente per tutti gli assi.

Sintassi:

```
set autoscale <asse>      fissa l'autoscala per l'asse specificato  
set noautoscale <asse>   elimina l'autoscala per l'asse specificato  
show autoscale           mostra quali assi sono in autoscala
```

dove <asse> è anche x,y,z o x,y. Se <asse> non è specificato il comando vale per tutti gli assi.

Dummy

Normalmente GNUPLOT suppone che la variabile indipendente per il comando plot sia x e le variabili indipendenti per il comando splot siano x e y. Essi sono chiamati le variabili fittizie perché sono solo un modo per indicare le variabili indipendenti. Il comando dummy permette di cambiare tali nomi; per esempio può essere più conveniente chiamare la variabile fittizia “t” quando si fa il grafico di funzioni del tempo

```
set dummy t
```

```
plot sin(t), cos(t)
```

Sintassi:

```
set dummy <dummy-var>{,<dummy-var>}  
show dummy
```

Esempi:

```
set dummy u, v  
set dummy, s
```

Il primo comando rinomina le variabili fittizie da x e y a u e v.

Il secondo comando rinomina solo il nome della seconda variabile da v a s.

Set grid

Opzione per il disegno di una griglia nel piano cartesiano.

Sintassi:

```
set grid  
set nogrid  
show grid
```

Samples (campionatura)

Il numero dei punti usati per disegnare un grafico è normalmente 100 punti. Può essere modificato a piacere con il comando samples. Un maggior numero di punti produce grafici migliori, ma allunga il tempo per il loro tracciamento.

Sintassi:

```
set samples <campionamento_1> {,<campionamento_2>}  
show samples
```

graph-menu

Premendo il tasto destro del mouse sul grafico prodotto da Gnuplot si accede al menù delle opzioni grafiche:

Bring to top: porta il grafico in primo piano.

Color: colora il grafico; se l'opzione è tolta il grafico verrà mostrato in bianco e nero.

Copy to Clipboard: copia l'immagine come un file bitmap.

Background: seleziona il colore dello sfondo.

Choose Font: modifica il carattere delle scritte nel grafico

Line Styles: permette la personalizzazione di tutte le linee del grafico.

Print: permette la stampa immediata del grafico.

Update wgnuplot.ini salva le opzioni grafiche precedenti definite dall'utente nel file di inizializzazione wgnuplot.ini.

X e Y range

Xrange fissa l'insieme di valori dell'asse x che verrà utilizzato nel disegno delle funzioni.

Sintassi:

```
set xrange [{<xmin> : <xmax>}]
```

dove i termini <xmin> e <xmax> sono costanti o espressioni.

Entrambi <xmin> and <xmax> sono specificabili a discrezione dall'utente.

Esempio:

```
set xrange [:10]
```

cambia il valore xmax a 10, ma lascia xmin al valore impostato precedentemente.

Yrange fissa l'insieme di valori dell'asse y che verrà utilizzato nel disegno delle funzioni.

Sintassi:

```
set yrange [{<ymin> : <ymax>}]
```

dove i termini <ymin> e <xmax> sono costanti o espressioni.

Entrambi <ymin> and <ymax> sono specificabili a discrezione dall'utente.

Esempio:

```
set yrange [:10]
```

cambia il valore ymax a 10, ma lascia ymin al valore impostato precedentemente.

Funzioni definite dall'utente

Oltre alle funzioni offerte dal programma, è possibile definirne di nuove a proprio piacimento utilizzando la sintassi seguente:

sintassi:

$$\langle \text{nome della funzione} \rangle (\langle \text{variabile1} \rangle \{ \langle \text{variabile2} \rangle \{ , \dots \} \}) = \langle \text{espressione} \rangle$$

dove $\langle \text{espressione} \rangle$ definisce una espressione funzione delle variabili $\langle \text{variabile1} \rangle$, $\langle \text{variabile2} \rangle$...

Esempi:

$$w = 2$$

$$f(x) = \sin(w*x)$$

Proviamo ad esempio a costruire la funzione logaritmo in base generica a.
Dalla teoria sappiamo che:

$$\log_c b = \log_a b \cdot \log_c a \quad \text{ossia} \quad \log_a b = \frac{\log_c b}{\log_c a}$$

$$\text{infatti ponendo:} \quad a^x = b \quad e \quad c^y = a \quad [1.1]$$

$$\text{segue} \quad x = \log_a b \quad e \quad y = \log_c a \quad [1.2]$$

ed elevando la seconda delle [1.1] a potenza con esponente x si ottiene:

$$c^{xy} = b \quad \text{o anche} \quad xy = \log_c b$$

da cui, sostituendo i valori ricavati dalle [1.2] si ottiene:

$$\log_c b = \log_a b \cdot \log_c a$$

da cui:

$$\boxed{\log_a b = \frac{\log_c b}{\log_c a}} \quad [1.3]$$

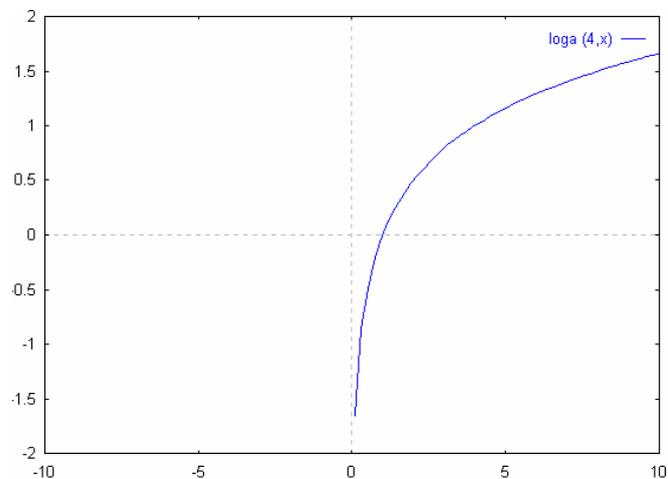
Questa proprietà rappresenta l'importante regola del cambiamento di base nei logaritmi.

La formula ottenuta [1.3] consente di effettuare il cambio di base in un logaritmo e pertanto può essere utilizzata con successo nel definire la funzione logaritmo in base generica “a” che chiameremo per comodità loga. Loga sarà pertanto una funzione a due argomenti: base del logaritmo e argomento del logaritmo e possiamo ottenerla definendola come segue

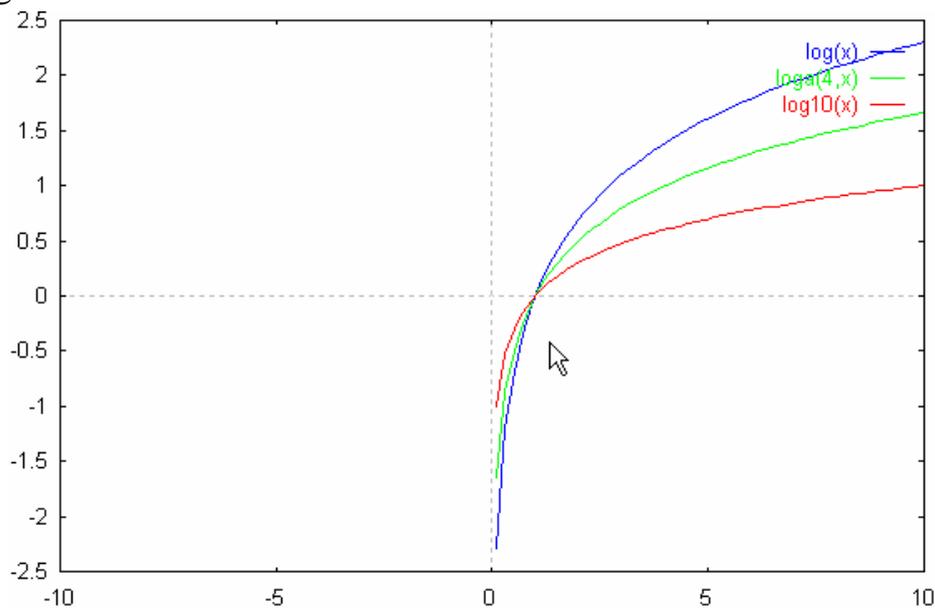
$$\text{loga}(a, x) = \log(x) / \log(a)$$

Ad esempio dopo aver digitato il comando precedente, volendo disegnare il logaritmo in base 4 di x si scriverà

plot loga (4,x)



Come si può vedere sotto il suo grafico è compreso tra quello del logaritmo in base 10 e del logaritmo in base “e”.



Una volta definita una funzione è opportuno salvarla con il comando save in modo da averla a disposizione ogni volta che si usa il programma chiamandola tramite il comando open.