

Alcuni appunti su Gnuplot

Vedere la Web page <http://www.gnuplot.info> per scaricare liberamente il programma e per ulteriori dettagli.

Questi appunti servono per dare i più semplici comandi Gnuplot in modalità di linea, per poter:

- Plottare i vostri dati (se sono nel formato opportuno)
- Salvare un plot dei vostri dati in formato PS o EPS
- Fittare i vostri dati con una funzione

Supponiamo che abbiate un file di dati **dati.dat** del tipo:

```
#prima riga di commento
#seconda riga di commento
0.00000 1.00000
0.0100000 0.904837
0.0200000 0.818731
0.0300000 0.740818
0.0400000 0.670320
0.0500000 0.606531
0.0600000 0.548812
0.0700000 0.496585
...
```

Al prompt scrivete:

```
host$ gnuplot
```

Vi apparirà sullo schermo

```
GNU PLOT
... version ....
.....
Terminal type set to ...
gnuplot
```

Il primo semplice comando è

```
gnuplot> plot 'dati.dat'
```

Le righe che iniziano con # (vedi esempio) sono interpretate come commenti. Per default la prima colonna di dati è presa come x e la seconda come y. Se avete più colonne di dati, specificate QUALI volete usare come x e come y. Se non specificate niente, è sempre sottinteso:

```
gnuplot> plot 'dati.dat' u 1:2
```

Se però avete e volete usare ad es. la colonna 4 come x e la 6 come y, scrivete:

```
gnuplot> plot 'dati.dat' u 4:6
```

Potete fare più curve sullo stesso grafico anche a partire dallo stesso file di dati, ad esempio:

```
gnuplot> plot 'dati.dat' u 1:2,'dati.dat' u 1:3,'dati.dat' u 1:4
```

Se non specificate la colonna usata come x, gnuplot prende di default l'indice del dato (quindi 1,2,3,...). Potete dire di fare un plot "con punti" (**with points**, abbreviato anche con **w p**; **ps** e **pt** specificano eventualmente size e type diversi) che è il default; oppure "con linee" (**with lines**, abbreviato con **w l**):

```
gnuplot> plot 'dati.dat' u 4:6 w l
```

Potete dire di fare un plot raggruppando i dati in blocchi separati da due linee vuote; sono indicizzati 0, 1, 2,...; se non specificate nulla, vengono usati gli stessi simboli (linee/colori/punti) ma ogni blocco viene plottato separatamente (quindi ad es. l'ultimo punto del blocco 0 NON e' unito al primo punto del blocco 1).

Si possono plottare anche vettori; il comando seguente disegna un vettore da (x,y) a (x+dx,y+dy), dando le colonne

di x, y, dx, dy:

```
gnuplot> plot 'file.dat' using 1:2:3:4 with vectors head filled lt 2
```

Altri dettagli li troverete dall'help-on-line:

```
gnuplot> help plot using
```

Supponiamo volete fittare i vostri dati con un esponenziale, allora dovete dare 1) la forma funzionale e 2) i parametri:

```
gnuplot> f(x) = a * exp (-x*b)
```

Poi richiamiamo la forma funzionale, i parametri che vogliamo determinare, e i dati su cui vogliamo fare il fit così:

```
gnuplot> fit f(x) 'dati.dat' via a,b
```

Dopo aver dato questo comando, otterrete sullo schermo qualcosa come:

```
Final set of parameters Asymptotic Standard Error
=====
a = 1 +/- 8.276e-08 (8.276e-06%)
b = 10 +/- 1.23e-06 (1.23e-05%)

correlation matrix of the fit parameters:

a b
a 1.000
b 0.671 1.000
```

Dopo aver fatto il fit, potete fare un plot contemporaneamente dei dati e della funzione che li fitta:

```
gnuplot> plot f(x), 'dati.dat'
```

Quantità plottate insieme devono essere separate da una virgola

Maggiori informazioni le trovate scrivendo **help** al prompt di gnuplot:

```
gnuplot> help
```

Dopo **plot**, il comando pi&u importante è forse **set**. Provate a scrivere:

```
gnuplot> help set
```

Per plottare i vostri dati in un file postscript (namely encapsulated postscript) in modo da includerlo eventualmente in un file TeX, dovete usare il comando **set**. Set cosa? Ad esempio **TERM**. Facendo partire gnuplot, automaticamente il term è settato come finestra **x11**. Per cambiarlo e forzarlo ad essere encapsulated postscript, scrivete:

```
gnuplot> set term post eps
```

oppure, se volete postscript e basta, solo

```
gnuplot> set term post
```

La risposta da gnuplot è:

```
Options are 'eps noenhanced monochrome dashed defaultplex "Helvetica" 14'
```

Adesso settiamo il formato dell' output. Dobbiamo dare un nome al file. Ad esempio "dati.eps", allora scriviamo:

```
gnuplot> set output 'dati.eps'
```

Attenzione a usare la virgoletta semplice in alto: '<filename>' per identificare tutti i nomi di files! A questo punto si può riplottare dati e funzione:.

```
gnuplot> plot f(x), 'dati.dat'
```

Se volete continuare a fare altri plot su una finestra x11, dovete resettare il TERM come **x11**, riscrivendo il comando:

```
gnuplot> set term x11
```

Supponiamo che volete fare un plot bidimensionale delle curve di livello di una funzione $z=f(x,y)$ che varia supponiamo da -1 a 1. Dovete scrivere i dati in un file in modo opportuno, cioè i valori di z in ordine su una griglia (x,y): tenendo fisso y e variando x, poi cambiando y e rifacendo variare x, etc. etc.; ogni volta che cambiate riga di questa griglia (cioè variate y) dovete lasciare una linea bianca nel file di dati, che quindi si presenta così:

```
#prima riga di commento  
1.00000  
0.904837  
....  
....  
0.818731  
0.670320  
-0.765234  
-0.097823  
....  
....  
-1.000000  
-0.876325  
....  
....  
....  
....  
0.876324  
0.876342  
....  
....  
0.972312  
0.765234
```

I comandi sono:

```
gnuplot> set view 0,0  
gnuplot> set nosurface  
gnuplot> set contour  
gnuplot> set cntrparam levels auto 5  
che vi fa automaticamente 5 livelli tra il max. e il min., oppure:  
gnuplot> set cntrparam levels discrete -1,-0.8,0.4,1  
se volete che vi faccia le curve solo corrispondenti ai livelli elencati: -1,-0.8,0.4,1; infine fate:  
gnuplot> splot "dati.dat"
```

Altre informazioni dal solito help-on-line, ad esempio:

```
gnuplot> help set cntrparam
```

BUON LAVORO!