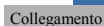


<http://www.fis.uniroma3.it/~meneghini>

Gnuplot è un programma di **grafici di dati e funzioni** portatile, interattivo multi-piattaforma (**UNIX**, **IBM OS/2**, **MS Windows**, **DOS**, **Apple Macintosh**, **VMS**, **Atari**, etc...) pilotato da "linea di comando".
E' un software **freeware e open-source**.
Può essere utilizzato in modo interattivo o come interfaccia grafica per altre applicazioni.
Gnuplot supporta diversi tipi di grafici in **2D**, e in **3D**.
Consente l'uso di diversi **terminali**: schermo interattivo, plotters, stampanti, output su file in diversi formati (ps, eps, LaTeX, metafont, pdf, svg, bitmap, etc...).

Include routines per fit non lineare di dati.

- <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>



Nota: i comandi sono CASE SENSITIVE,
cioè **P**rint è diverso da **p**rint

Calcolo di espressioni

Nota: i comandi possono esser dati in forma abbreviata

```
gnuplot> print 2+2
4
```

```
gnuplot> pr 3.14159*2*3
18.84954
```

Potenza	**	2**3
prodotto	*	3*2
divisione	/	3/2
somma	+	3+2
sottrazione	-	3-2
parentesi	()	

```
gnuplot> pr 3/2
```

1

Divisione tra interi

Ok

```
gnuplot> pr 3./2
```

1.5

Nota: Almeno uno dei due numeri deve essere un numero "reale"

Problema: calcolare il valore della forza di Coulomb tra due cariche $q_1=2$ C, $q_2=1.5$ C poste ad una distanza di 15 cm nel vuoto

$$F_c = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r^2}$$

Costanti

$\pi = 3.14159$

$\epsilon_0 = 8.85 \cdot 10^{-12}$

Dati

$q_1 = 2.0$ C

$q_2 = 1.5$ C

$r = 0.015$ m

Potenza	**
prodotto	*
divisione	/
somma	+
sottrazione	-
parentesi	()

```
gnuplot> pr 1/4 * 3.14159 * 8.85e-12 *2.0 * 1.5 / 0.015**2
```

?

```
0 - 1/4 * 3.14159 * 8.85e-12 *2.0 * 1.5 / 0.015**2
```

```
1 - 1/4 * 3.14159 * 8.85e-12 *2.0 * 1.5 / 0.000225
```

```
2 - 0.25 * 3.14159 * 8.85e-12 *2.0 * 1.5 / 0.000225
```

```
4 - 2.085e-11 / 0.000225
```

```
5 - 9.268e-8
```

$$F_c = \frac{1}{4} \frac{\pi \epsilon_0 q_1 q_2}{r^2}$$

!

Problema: calcolare il valore della forza di Coulomb tra due cariche $q_1=2$ C, $q_2=1.5$ C poste ad una distanza di 15 cm nel vuoto

$$F_c = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r^2}$$

Costanti

$\pi = 3.14159$

$\epsilon_0 = 8.85 \cdot 10^{-12}$

Dati

$q_1 = 2.0$

$q_2 = 1.5$

$r = 0.015$ m

Potenza	**	2**3
prodotto	*	3*2
divisione	/	3/2
somma	+	3+2
sottrazione	-	3-2
parentesi	()	

```
gnuplot> pr 1/(4 * 3.14159 * 8.85e-12) *2.0 * 1.5/ 0.015**2
```

```
0 - 1/(4 * 3.14159 * 8.85e-12) *2.0 * 1.5/ 0.015**2
```

```
1 - 1/(4*3.14159 * 8.85e-12) *2.0 * 1.5 / 0.000225
```

```
2 - 1 / 1.112e-10 * 2.0 * 1.5 / 0.000225
```

```
4 - 8.992e+9 * 2.0 * 1.5 / 0.000225
```

```
5 - 2.6975e10 / 0.000225
```

```
6 - 11.989e13
```

```
gnuplot> pr (1/(4 * 3.14159 * 8.85e-12))*(2.0*1.5/0.015**2)
```

Definizione e uso di variabili

Problema: calcolare il valore della forza di Coulomb tra due cariche $q_1=2$ C, $q_2=1.5$ C poste ad una distanza di 15 cm nel vuoto

$$F_c = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r^2}$$

Costanti

$\pi = 3.14159$

$\epsilon_0 = 8.85 \cdot 10^{-12}$

Dati

$q_1 = 2.0$

$q_2 = 1.5$

$r = 0.015$ m

```
gnuplot> q1=2.
```

```
gnuplot> q2=1.5
```

```
gnuplot> r=0.015 ; eo=8.85e-12
```

```
gnuplot> pr 1/(4 * pi * eo) * q1 * q2/ r**2
```

```
gnuplot> show variables
```

Variables:

pi = 3.14159265358979

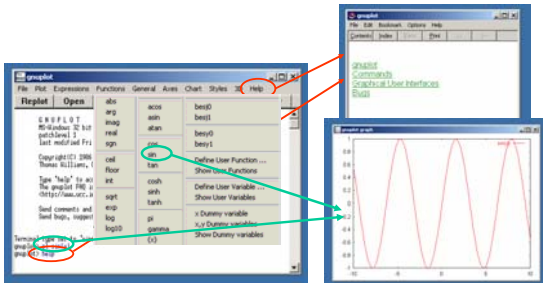
q1 = 2.0

q2 = 1.5

r = 0.015

eo = 8.85e-012

Comandi

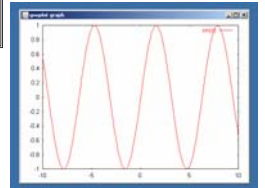


Possono esser dati sulla linea di comando o usando i pulsanti

Calcolo/Grafico di Funzioni

```
gnuplot> print sin(0.5)
0.479425538604203
```

```
gnuplot> plot sin(0.5)
```



```
gnuplot> help functions
```

abs	cos	log	tm_min	atan2	ibeta	sinh
acos	cosh	log10	tm_mon	atanh	inverf	sqrt
acosh	erf	norm	tm_sec	besj0	igamma	tan
arg	erfc	rand	tm_wday	besj1	imag	tanh
asin	exp	real	tm_yday	besy0	invtanh	column
asinh	floor	sgn	tm_year	besy1	int	tm_hour
atan	gamma	sin	valid	ceil	lgamma	tm_mday

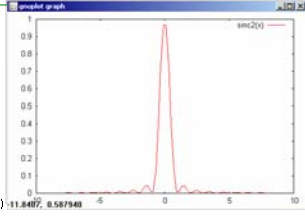
Definire le proprie funzioni

```
gnuplot> f(x) = a*x**2 + b*x + c
gnuplot> a=1.5 ; b=-2 ; c = 1
gnuplot> pl f(x)
```

```
gnuplot> sinc(x) = sin(pi*x)/(pi*x)
gnuplot> pl sinc(x)
```

```
gnuplot> sinc2(x) = (sin(pi*x)/(pi*x))**2
gnuplot> pl sinc2(x)
```

```
w = 2
q = floor(tan(pi/2 - 0.1))
f(x) = sin(w*x)
sinc(x) = sin(pi*x)/(pi*x)
delta(t) = (t == 0)
ramp(t) = (t > 0) ? t : 0
min(a,b) = (a < b) ? a : b
comb(n,k) = n!/(k!*(n-k)!)
len3d(x,y,z) = sqrt(x*x+y*y+z*z)
```



```
gnuplot> f(x) = sin(a*x)
```

```
gnuplot> a = 0.2
```

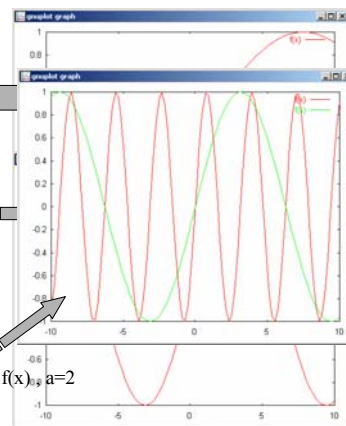
```
gnuplot> plot f(x)
```

```
gnuplot> f(x) = sin(a*x)
```

```
gnuplot> plot f(x), a=0.5
```

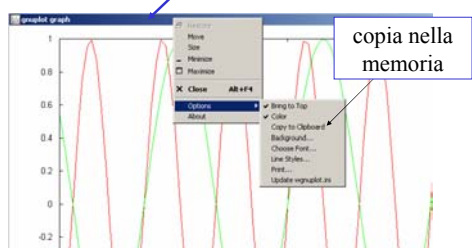
```
gnuplot> f(x) = sin(a*x)
```

```
gnuplot> plot f(x), a=0.5, f(x), a=2
```



Esportare i grafici

right click



Definire le proprie funzioni

Problema: scrivere le funzioni per il calcolo dell'area e il volume di figure e solidi geometrici (es. quadrato, cubo e sfera).

$$Aq = L^2 \quad Vc = L^3$$

```
gnuplot> L = -.3
gnuplot> Aq = L ** 2; Vc = L ** 3
gnuplot> pr Aq,Vc
```

$$Ss = 4\pi r^2 \quad Vs = \frac{4}{3}\pi r^3$$

```
gnuplot> r=1.
gnuplot> Ss = 4. * pi * r**2
gnuplot> Vs = (4./3.)*pi * r ** 3
gnuplot> pr Vs, Ss
gnuplot> r=1.5
gnuplot> pr Vs, Ss
```

```
gnuplot> r=1.
gnuplot> Ss = 4. * pi * r**2
gnuplot> Vs = (4./3.)*pi * r ** 3
gnuplot> pr Vs, Ss
```

$R \neq r$

$4/3 = 1$

$Pi \neq pi$

Problema: Le funzioni predefinite in Gnuplot consentono il calcolo del logaritmo naturale ($\log(x) = \log_e x$) e del logaritmo decimale ($\log_{10}(x) = \log_{10} x$). Scrivere un'espressione per calcolare il logaritmo di un numero A in una base b qualunque.

Proprietà dei logaritmi:

$$\log_b A = \frac{\log_c A}{\log_c b}$$

$$\log_b(b, x) = \log(x) / \log(b)$$

es.: $\log_2 16$

```
gnuplot> logb(b,x) = log(x)/log(b)
gnuplot> pr logb(2,16)
4.0
```

```
gnuplot> f(x) = a*x**3 + b*x**2 + c*x + d
gnuplot> show functions
```

```
User-Defined Functions:
f(x) = a*x**3 + b*x**2 + c*x + d
```

Esercizio

Problema: calcolare il valore della distribuzione di Gauss:

$$G(x) = \frac{A}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_0)^2}{2\sigma^2}}$$

per $A=12.5$, $\sigma = 0.08$, $x_0 = 1.0$, $x=1.5$

Costanti

$\pi = 3.14159$

Dati

$A = 12.5$

$\sigma = 0.08$

$x_0 = 1.0$

$x = 1.5$

Numeri Complessi

Parte reale
Parte Immaginaria
 $C = \text{Re} + i \text{Im}$

$C = 3 + i4$

```
gnuplot> C = {3.,4.}
gnuplot> abs(C)
5.0
```

$$|C| = \sqrt{\text{Re}^2 + \text{Im}^2}$$

```
gggnuplot> abs(C)
5.0
```

Esercizi

dati
 $C1 = \{3., 4.\}$
 $C2 = \{3.2, -3.32\}$

Calcolare:
 $C1 + C2$; $C1 - C2$
 $\sin(a*b)$; $\abs(a/b)$
...

Operatori

The following is a list of all the unary operators and their usages:

Symbol	Example	Explanation
~	~a	unary minus
+	+a	unary plus (no-operation)
-	-a	* cast to complement
!	!a	* logical negation
*	a*	* factorial
()	(c)	* call, eval/colon during 'using' manipulation

(*) Starred explanations indicate that the operator requires an integer argument.

Operator precedence is the same as in Fortran and C. As in those languages, parentheses may be used to change the order of operation. Thus $2**2 = 4$, but $(2)**2 = 4$.

Symbol	Example	Explanation
**	a**b	exponentiation
*	a/b	division
/	a/b	division
+	a+b	* modulo
+	a+b	addition
-	a-b	subtraction
==	a==b	equality
!=	a!=b	inequality
<	a<b	less than
<=	a<=b	less than or equal to
>	a>b	greater than
>=	a>=b	greater than or equal to
&	a&b	* bitwise AND
^	a^b	* bitwise exclusive OR
	a b	* bitwise inclusive OR
&&	a&&b	* logical AND
	a b	* logical OR

(*) Starred explanations indicate that the operator requires integer arguments.

Logical AND (&&) and OR (||) short circuit the way they do in C. That is, the second && operand is not evaluated if the first is false, the second || operand is not evaluated if the first is true.

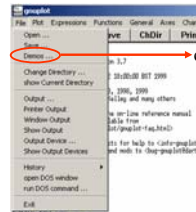
Symbol	Example	Explanation
?:	a?:b:c	ternary operation

The ternary operator behaves as it does in C. The first argument (a), which must be an integer, is evaluated. If it is true (non-zero), the second argument (b) is evaluated and returned; otherwise the third argument (c) is evaluated and returned.

Molto di cio' che Gnuplot puo' fare

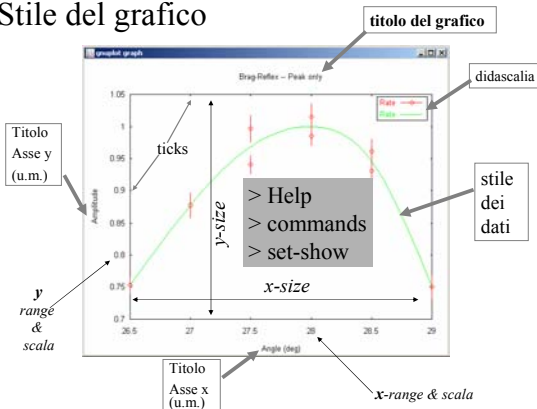
<http://gnuplot.sourceforge.net/demo/>

oppure:



c:/programmi/gnuplot/demos/*.dem

Stile del grafico



Comandi

`gnuplot > set title 'titolo'`

Definisce il titolo del Grafico

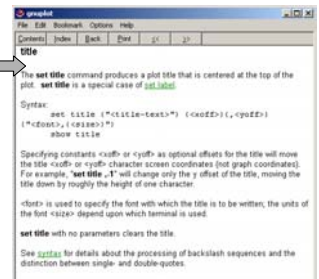
`gnuplot > show title`

mostra il titolo del Grafico

`gnuplot > set title`

titolo vuoto

`[gnuplot] help set title`

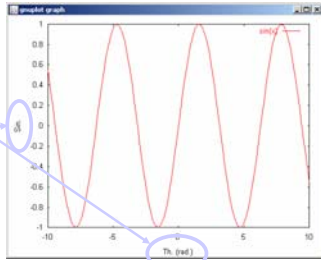


```
gnuplot > set xlabel 'titolo'
gnuplot > show xlabel
gnuplot > set xlabel
gnuplot > help set xlabel
```

Definisce l'asse delle x
mostra l'asse delle x
label (etichetta!) vuoto
help

lo stesso per l'asse delle y

```
[gnuplot] set xlabel 'Th (rad)'
[gnuplot] set ylabel 'sin'
```

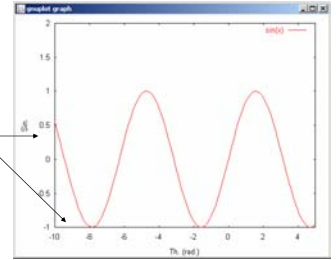


```
gnuplot > set xrange [min:max]
gnuplot > show xrange
gnuplot > set auto
gnuplot > help set xrange
```

Range per l'asse delle x
mostra il range ed i x
help

lo stesso per l'asse delle y

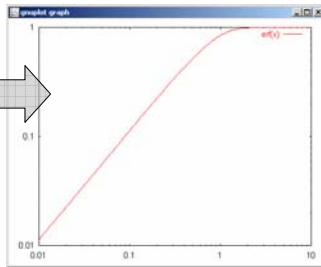
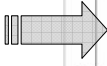
```
[gnuplot] set xr [-10:5]
[gnuplot] set yr [-1:2]
[gnuplot] pl sin(x)
```



```
gnuplot > set logscale x(y)
gnuplot > set nologscale x(y)
```

scala logaritmica sull'asse x (y)
scala lineare sull'asse x (y)

```
[gnuplot] set xr [0.01:10]
[gnuplot] set yr [0.01:1]
[gnuplot] set log x
[gnuplot] set log y
[gnuplot] pl exp(x)
```

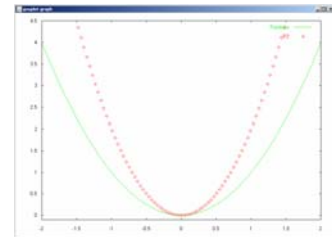


Il comando Plot (funzioni)

```
gnuplot> plot [x_min-x_max][y_min-y_max] f(x) title 'caption' with line l
```

Range funzione caption Stile

```
gnuplot> plot [-2:2][-1:4.5] f(x) title 'parabola' with line 2\
> 2, f(x)*2 title 'P2' with points 1
```



```
gnuplot> pl [-2:2][-1:4.5] f(x) t 'parabola' w l 2, f(x)*2 t 'P2' w p 1
```

```
gnuplot> set
valid set options: {} = choose one, {} means optional
'angles', '(no)arrow', '(no)autoscale', 'bars', '(no)border',
'boxwidth', '(no)clip', '(no)clip', 'contour', '(no)contour',
'data style', '(no)grid', 'dummy', 'encoding', 'format',
'function style', '(no)grid', '(no)hidden', 'isosamples',
'(no)key', '(no)label', '(no)linestyle', 'local', '(no)logscale',
'(no)margin', 'missing', 'missing', '(no)multiple', 'offsets',
'origin', 'output', '(no)parametric', 'pointsize', '(no)polar',
'rtvrange', 'samples', 'size', 'no surface', 'terminal',
'tics', 'ticscale', 'ticlevel', '(no)timestamp', 'timestep',
'title', 'view', 'xaxis1(2)label', 'xaxis1(2)label', 'xaxis1(2)tics',
'(no)logaxis1(2)tics', 'logaxis1(2)tics', '(no)logaxis1(2)zeroaxis',
'zero'
gnuplot> show
valid set options: {} = choose one, {} means optional
'all', 'angles', 'arrow', 'autoscale', 'bar', 'border', 'boxwidth',
'clip', 'contour', 'contour', 'data', 'dgrid3d', 'dummy',
'encoding', 'function', 'grid', 'hidden', 'isosamples',
'key', 'label', 'linestyle', 'local', 'logscale', 'margin', 'bargin',
'missing', 'offsets', 'origin', 'output', 'plot', 'parametric',
'pointsize', 'polar', 'rtvrange', 'samples', 'size', 'terminal',
'tics', 'timestep', 'timestep', 'title', 'variables', 'version',
'view', 'xaxis1(2)label', 'xaxis1(2)label', '(no)logaxis1(2)tics',
'logaxis1(2)tics', 'xaxis1(2)zeroaxis', 'xaxis1(2)zeroaxis',
'zeroaxis'
```

I dati

Devono essere file ASCII

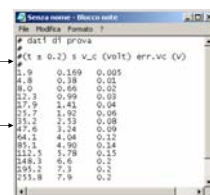
I dati devono essere ordinati per colonne,

Le righe che iniziano per # sono considerate commenti



Commenti

Colonne di
dati

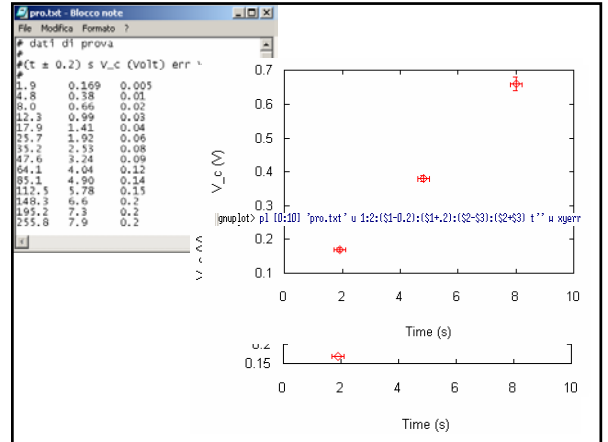


Barre d'errore

Per graficare un file di dati **gnuplot** richiede 2, 3, 4 o 6 colonne (o set di dati):

colonne	defaults	style	
2 colonne:	u 1:2	x,y	1 : colonna delle x, 2 colonna delle y
3 colonne:	u 1:2:3	w yerr	errore su y (simmetrico) in colonna 3
4 colonne:	u 1:2:3:4	w xerr	errore su x (simmetrico) in colonna 3
5 colonne:	u 1:2:3:4	w yerr	errore su x (simmetrico) in colonna 3
6 colonne:	u 1:2:3:4:5:6	w xerr	errore su y (simmetrico) in colonna 3

>pl 'nomefile' using 1:2:3 with xerrorbars considera la terza colonna come errore su x
>pl 'nomefile' using 1:2:3 with yerrorbars considera la terza colonna come errore su y
>pl 'nomefile' using 1:2:(S2+S3):(S2-S4) with errorbars considera la terza e quarta colonna limite superiore e inferiore per l'errore su x
>pl 'nomefile' u 1:2:(\$1+\$3):(\$1-\$3):(\$2-\$4):(\$2-\$5) w xerrorbars considera la colonna 3 (\$3) come errore su x, la colonna 4 e 5 sono, rispettivamente, il limite e inferiore per l'errore su y
 ovviamente si possono usare funzioni e combinazioni relative:
> pl 'nomefile' u 1:2:(\$2*0.01) w ye rappresenta i dati con un errore del 10% sull'ordinata
> plo 'datafile.dat' u 1:2:(\$1-\$4):(\$1+\$3):(\$2-\$5):(\$2-\$5) w xye



Le Macro (o script)

gnuplot> save 'file.plt'

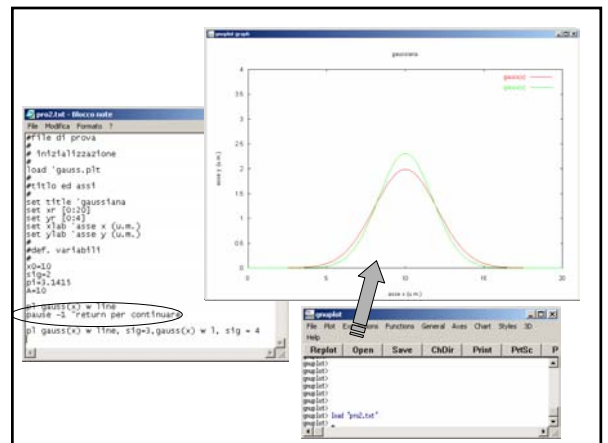
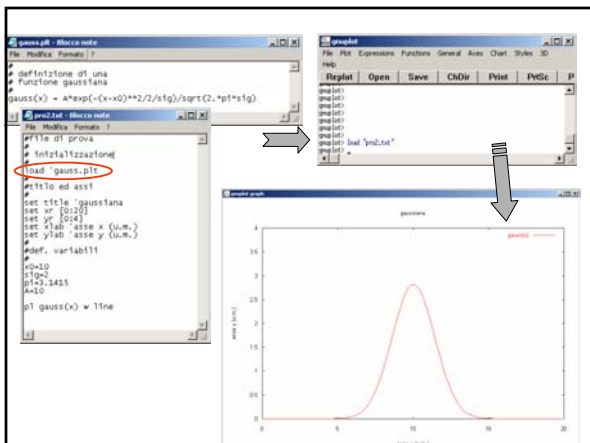
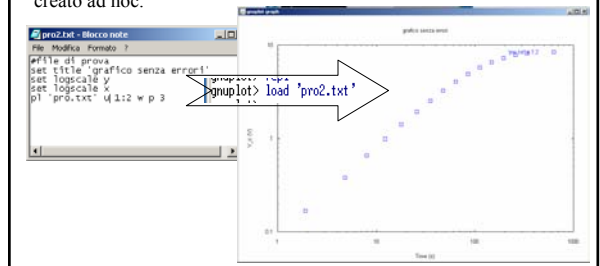
```
#
#                               'file.plt'
#
# GNUPLOT
# MS-Windows 32 bit version 3.7
# patchlevel 1
# last modified Fri Oct 22 18:00:00 BST 1999
#
# Send comments and requests for help to <info-gnuplot@datamath.edu>
# Send bugs, suggestions and patches to <bug-gnuplot@datamath.edu>
#
# set terminal windows
# set output
# set multiplot points
# set function style lines
#
# set xlabel "Time (s)" 0.000000,0.000000 ""
# set ylabel "V_c (V)" 0.000000,0.000000 ""
# set xrange [*:*] noreverse nowatchback
#
#
# f(x)=sin(x/180*3.14)
# pl [0:10] 'pro.txt' u 1:2:($1-0.2):($1+2):($2-$3):($2+$3) t'' w xerr
# EOF
```

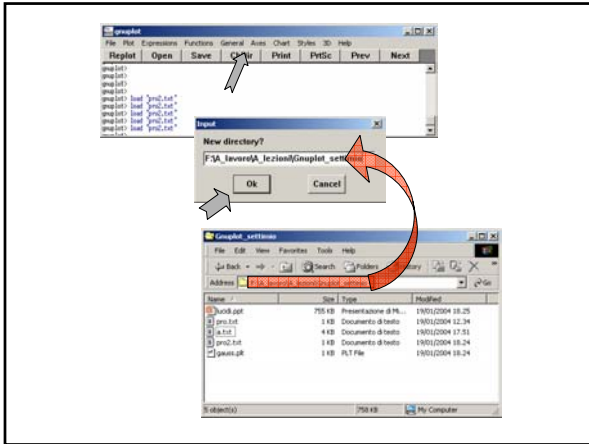
gnuplot> load 'file.plt' → **plot!!!**

Il file macro deve essere ASCII (blocco note o edito equivalente)

Il file macro puo' contenere uno o piu' comandi "Load", quindi leggere altre macro.

Il file macro puo' essere generato usando il comando "save", puo' essere una modifica del file generato dal comando "save" o essere creato ad hoc.





rebin.exe

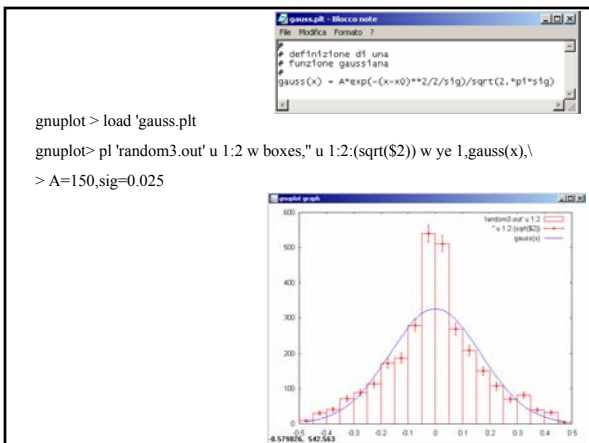
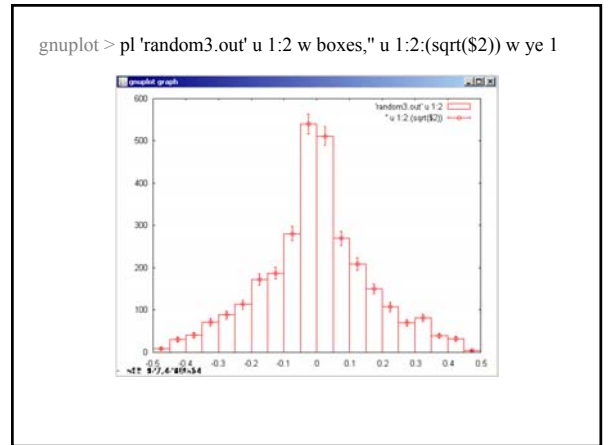
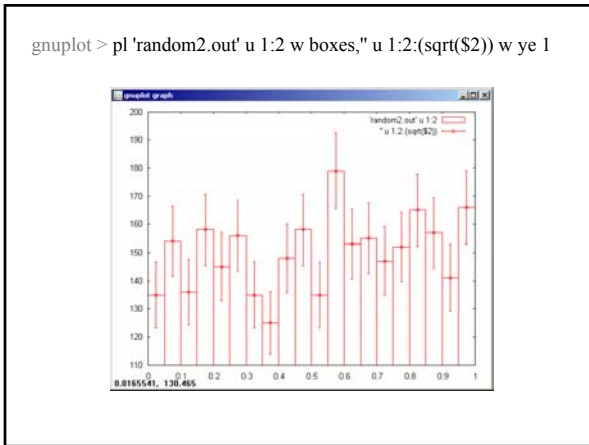
```

V:\A lavoro\A lezione\gnuplot_set\rebin.exe
=====
primo il binario di dati su una mesh data
File di dati: REBIN 2 colonne??
rebin2.dat
colonna dei dati
letti 20000 dati
=====
media: -0.41879641877777777777
varianza: 2.79145770287544E-002
std dev.: 0.1670801625558
valore minimo per gli intervalli: -0.541481500000000
=====
larghezza dell'intervallo? es.: 9.70000000000000E-002
valore massimo? es.: 0.534351500000000
=====
N. di intervalli: 20
es. per esito
=====

```

Il file random.dat:

- colonna 1: numero progressivo,
- colonna 2: numero casuale uniformemente distribuito nell'intervallo 0-1
- colonna 3: distribuzione gaussiana

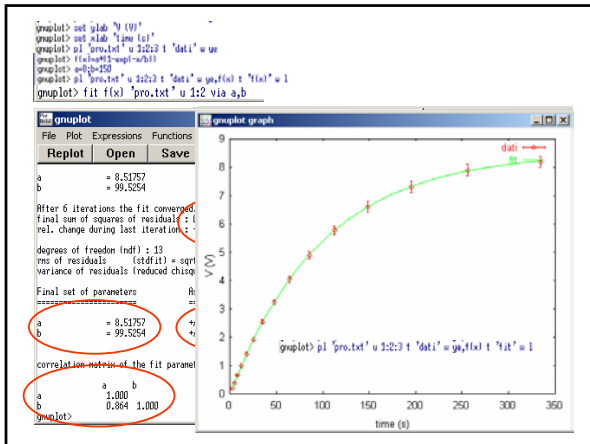


Fit di dati

Data una distribuzione di dati rappresentata da un insieme di punti sperimentali (x_i, y_i) e data una funzione $f(a_1, \dots, a_n; x)$ trovare i valori dei parametri a_j in modo tale che la $f(a_1, \dots, a_n; x)$ rappresenti la migliore approssimazione della distribuzione dei dati.

gnuplot> fit {{xrange}} {{yrange}} f(x) 'datafile' u 1:2 via *parameters*

gnuplot> fit {{xrange}} {{yrange}} f(x) 'datafile' u 1:2:3 via *parameters*

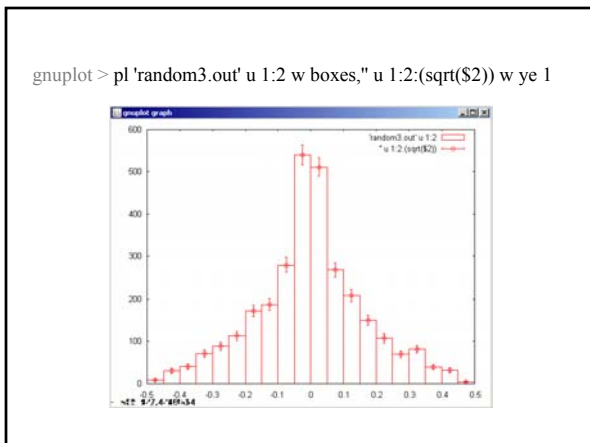
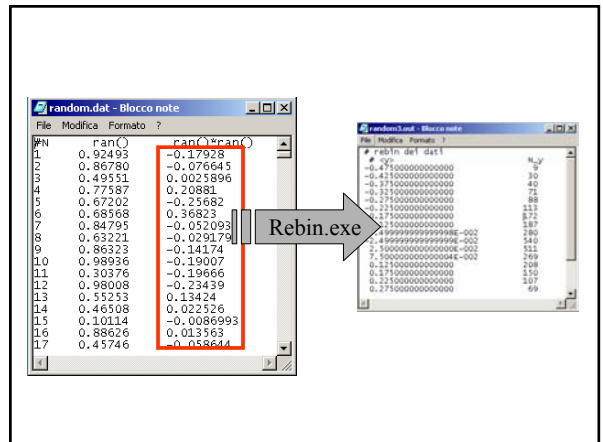
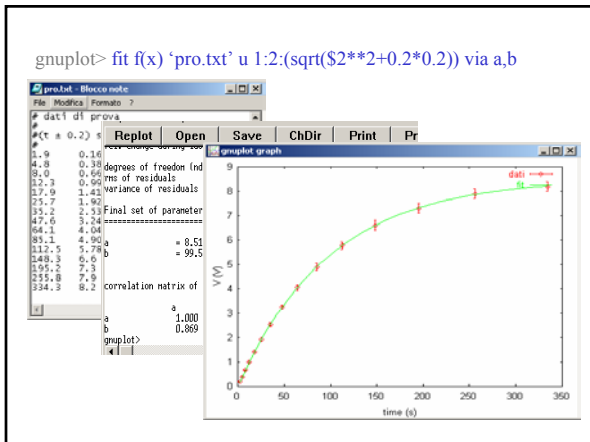


gnuplot> fit f(x) 'pro.txt' u 1:2:3 via a,b

il valore nella colonna 3 viene utilizzato per "pesare" il fit: i punti vengono pesati per w^{-2} dove w è il valore in colonna 3

gnuplot> fit f(x) 'pro.txt' u 1:2:(sqrt(\$2)) via a,b

Il peso può essere una funzione definita ad hoc, se in colonna 2 c'è un numero di conteggi con distribuzione normale, l'errore è proporzionale a $N^{1/2}$, quindi definisco il peso come $\sqrt{N}$



> gauss(x) = A*exp(-(x-x0)**2/(sig**2)/sqrt(2*pi*sig))

> a=100

> x0=0

> sig=0.1

> fit gauss(x) 'random3.out' u 1:2 via a, sig

> plot 'random3.out' u 1:2 w boxes, '' u 1:2:(sqrt(\$2)) w yerr, gauss(x)

