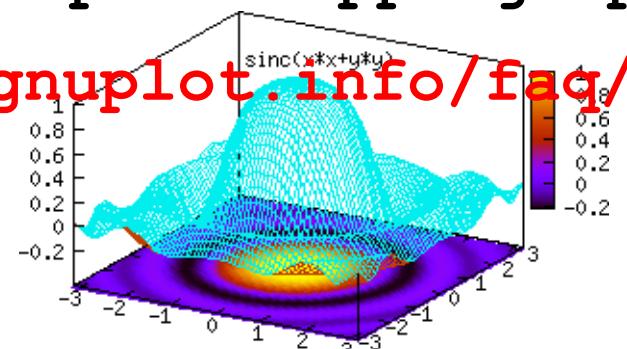
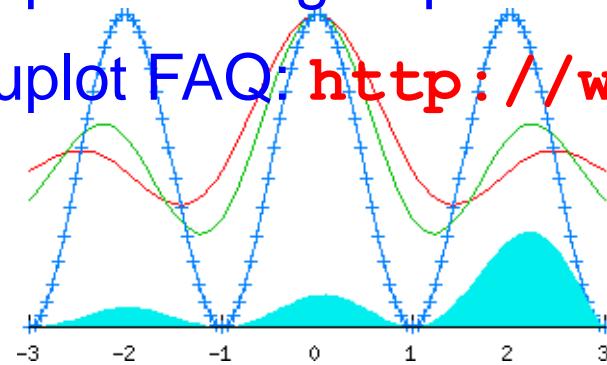


Sperimentazioni di Fisica 1 mod. A – Laboratorio 7

Tutorial di gnuplot

GNUPLOT

- Gnuplot è un programma a linea di comando per realizzare grafici con dati sperimentali o funzioni matematiche.
- Gnuplot funziona su diversi sistemi operativi:
 - ◆ il suo funzionamento è quasi indipendente dal sistema usato.
- Informazione reperibile su:
 - ◆ Gnuplot homepage: <http://www.gnuplot.info>
 - ◆ Gnuplot newsgroups: <comp.graphics.apps.gnuplot>
 - ◆ Gnuplot FAQ: <http://www.gnuplot.info/faq/>



GNUPLOT - credits

- Il breve tutorial riportato è disponibile, per intero, sul sito:

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



- not so Frequently Asked Questions -

Basic use

% **gnuplot**

G N U P L O T

Version 4.0 patchlevel 0
last modified Thu Apr 15 14:44:22 CEST 2004
System: Linux 2.4.27

Copyright (C) 1986 - 1993, 1998, 2004
Thomas Williams, Colin Kelley and many others

This is gnuplot version 4.0. Please refer to the documentation
for command syntax changes. The old syntax will be accepted
throughout the 4.0 series, but all save files use the new syntax.

Type `help` to access the on-line reference manual.
The gnuplot FAQ is available from

<http://www.gnuplot.info/faq/>

Terminal type set to 'x11'

gnuplot>

Gnuplot commands

■ I comandi di gnuplot permettono di:

- ◆ uscire, leggere un file di comandi, salvare parametri;
- ◆ realizzare grafici;
- ◆ modificare i parametri;
- ◆ eseguire comandi di shell;
- ◆ definire una funzione, sostituire variabili, e eseguire calcoli.

■ Per ottenere documentazione online digitare:

```
gnuplot> help
[...]
Help topics available:
batch/interactive   bugs           commands        comments
coordinates         copyright       environment    expressions
glossary            graphical      help-desk      introduction
line-editing        mouse          new-features  old_bugs
plotting            set             show           startup
substitution        syntax         time/date
```

Comandi per uscire, leggere un file e salvare parametri

- Per terminare gnuplot:

```
gnuplot> exit oppure quit
```

(i settings della sessione sono persi).

- Per salvare i parametri della sessione in corso:

```
gnuplot> save "filename"
```

stringhe delimitate da:
"" oppure da ''.

- Per caricare parametri da un file:

```
gnuplot> load "filename"
```

si rimane in gnuplot

◆ oppure dalla linea di comando della shell:

```
$ gnuplot "filename"
```

il controllo ritorna alla
shell

Comandi per realizzare plots

- Esistono due comandi:

◆ **plot**

grafici 2D

◆ **splot**

grafici 3D

- Permettono di piazzare:

- funzioni
- dati

- Esempio:

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

◆ crea una nuova finestra con

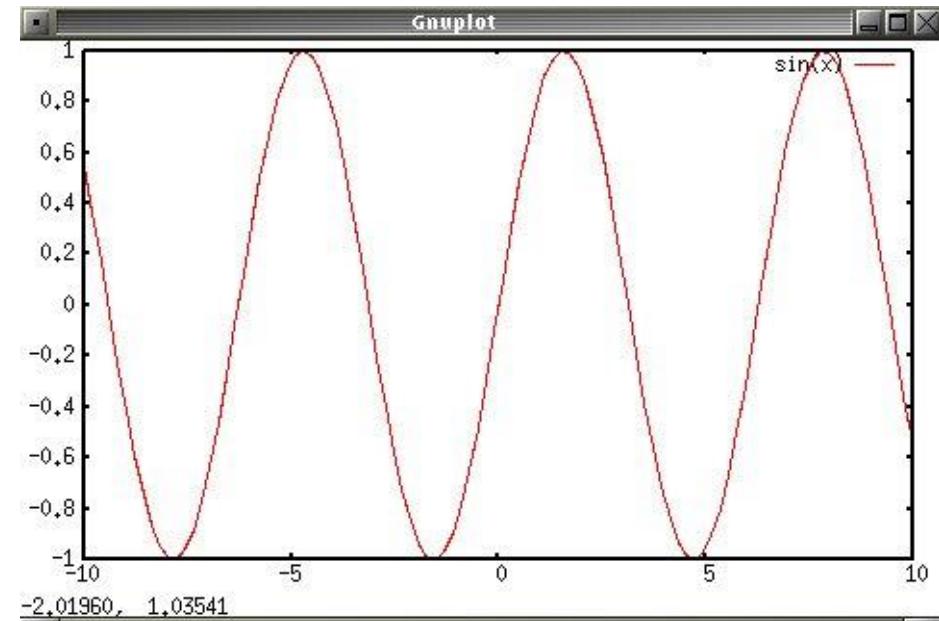


Grafico 2-dim

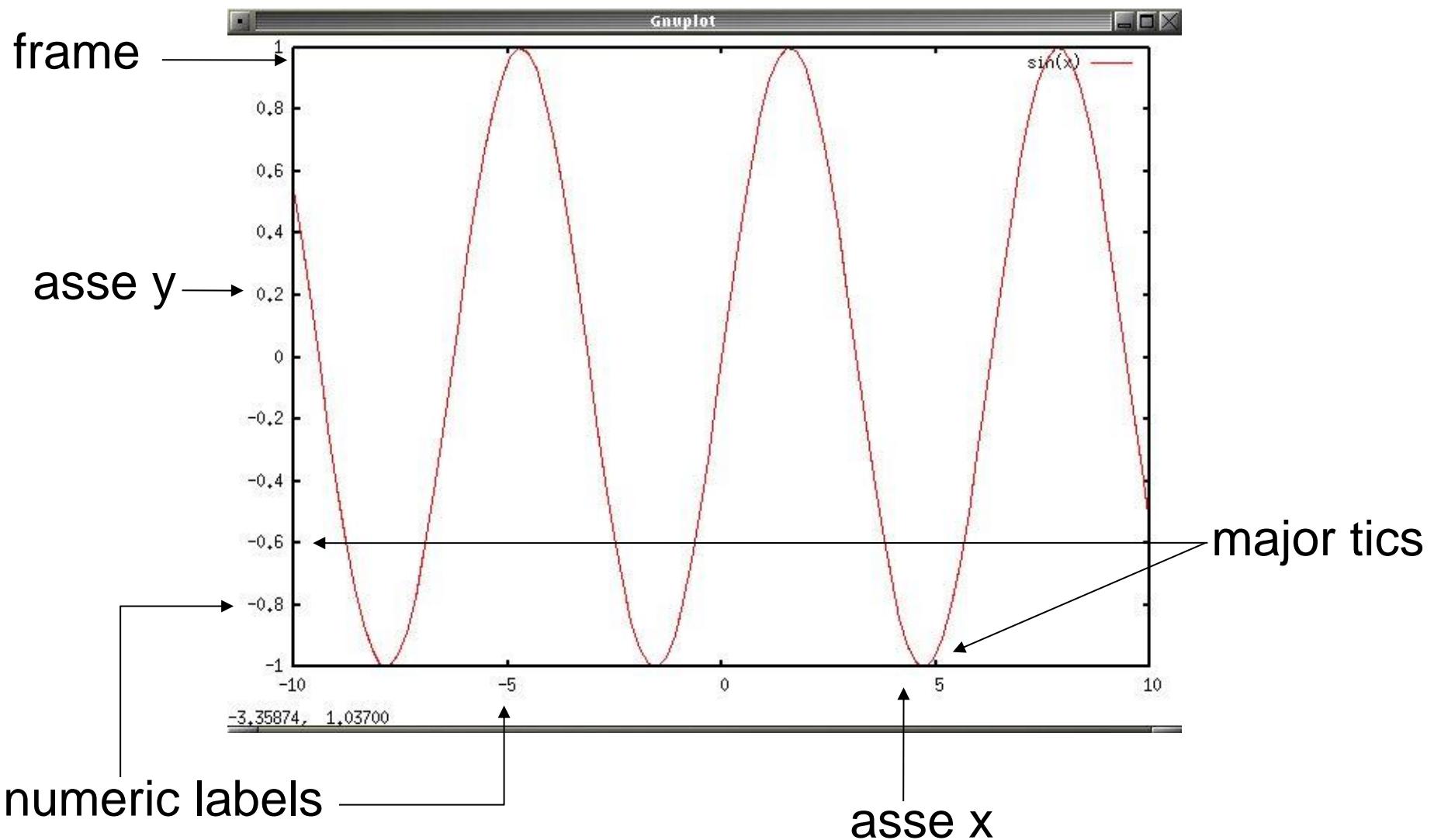


Grafico 2-dim

- Gnuplot determina i valori appropriati per gli assi x e y, se non sono stati specificati.
- Grafichiamo $\sin(x)$ tra 0 e 2π

```
gnuplot> plot [0:2*pi] sin(x)
```

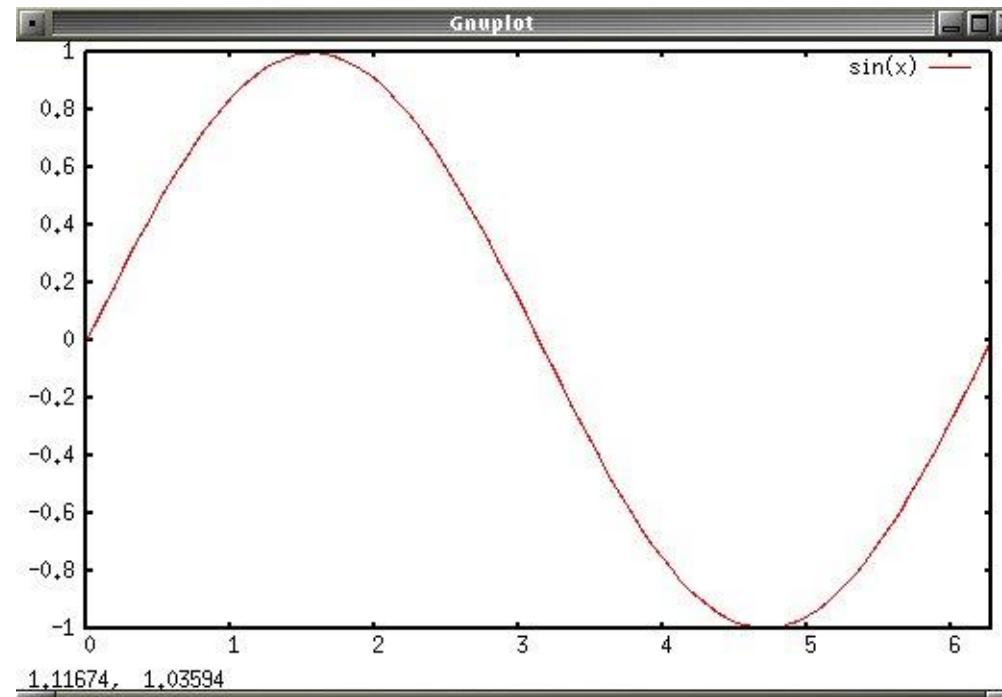
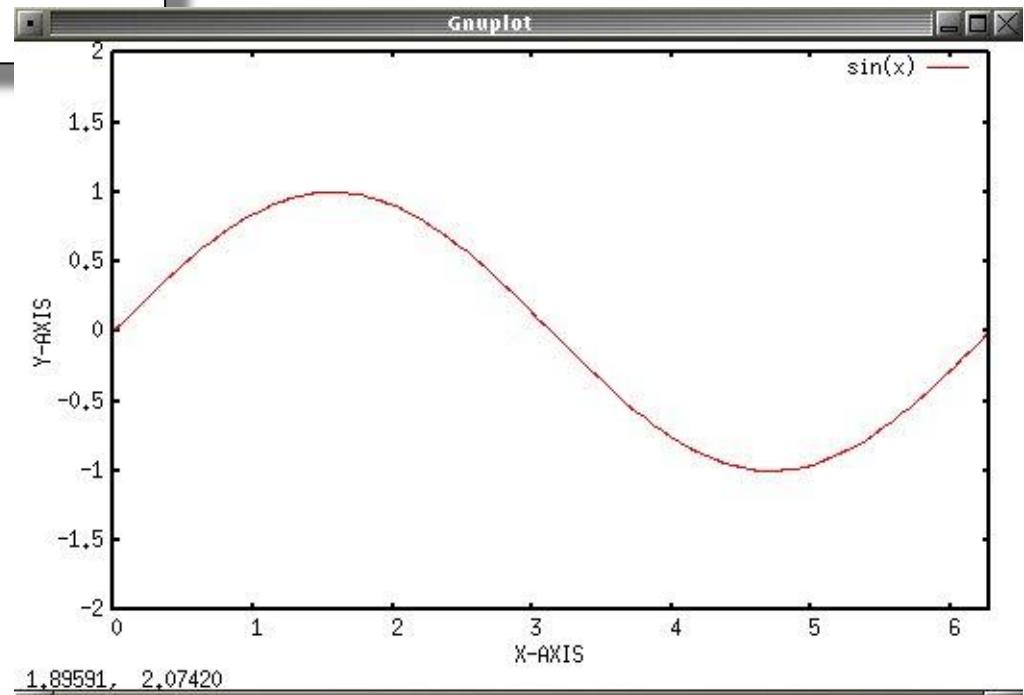


Grafico 2-dim – modifica di parametri

- Il comando **set** permette di modificare i parametri di un grafico.

- Grafichiamo $\sin(x)$ tra 0 e 2π

```
gnuplot> set xlabel "X-AXIS"  
gnuplot> set ylabel "Y-AXIS"  
gnuplot> set xrange [0:2*pi]  
gnuplot> set yrange [-2:2]  
gnuplot> plot sin(x)
```



online help con:
gnuplot> help set

Grafico 2-dim – set terminal

- Se non viene specificato un terminale, gnuplot crea i grafici sullo schermo.
- **set terminal** modifica il formato dell'output dei grafici, permettendo di creare un file
 - ◆ di tipo postscript;
 - ◆ per una stampante, etc.

■ Esempio:

```
gnuplot> set terminal postscript  
gnuplot> set output "prova.ps"  
gnuplot> plot sin(x)
```

online help con:

```
gnuplot> help set terminal
```

N.B.:

Senza ridirigere l'output,
il file postscript
apparirebbe direttamente
nella finestra usata per i
comandi di gnuplot.

Grafico 2-dim – comandi di shell

- È possibile sospendere temporaneamente gnuplot, passando alla shell, con il comando **shell**
 - ◆ **exit** permette di rientrare nella sessione di gnuplot.
- È possibile eseguire un comando di shell, senza lasciare gnuplot, facendolo iniziare con il carattere “!>:

```
gnuplot>!ls -l
total 60
-rw-r--r-- 1 alberto zeus 4540 Oct 24 17:57 figure.plt
-rw-r--r-- 1 alberto zeus 23504 Oct 24 17:57 figure.ps
...
-rw-r--r-- 1 alberto zeus 2150 Oct 24 17:45 pexp.out
gnuplot>
```

gnuplot supporta i comandi '**cd**' e '**pwd**'.

Per specificare una directory, è necessario racchiuderla tra “”:

```
gnuplot> cd "../other/dir"
```

Gnuplot come calcolatore

- Si possono utilizzare delle variabili, usando il formato:

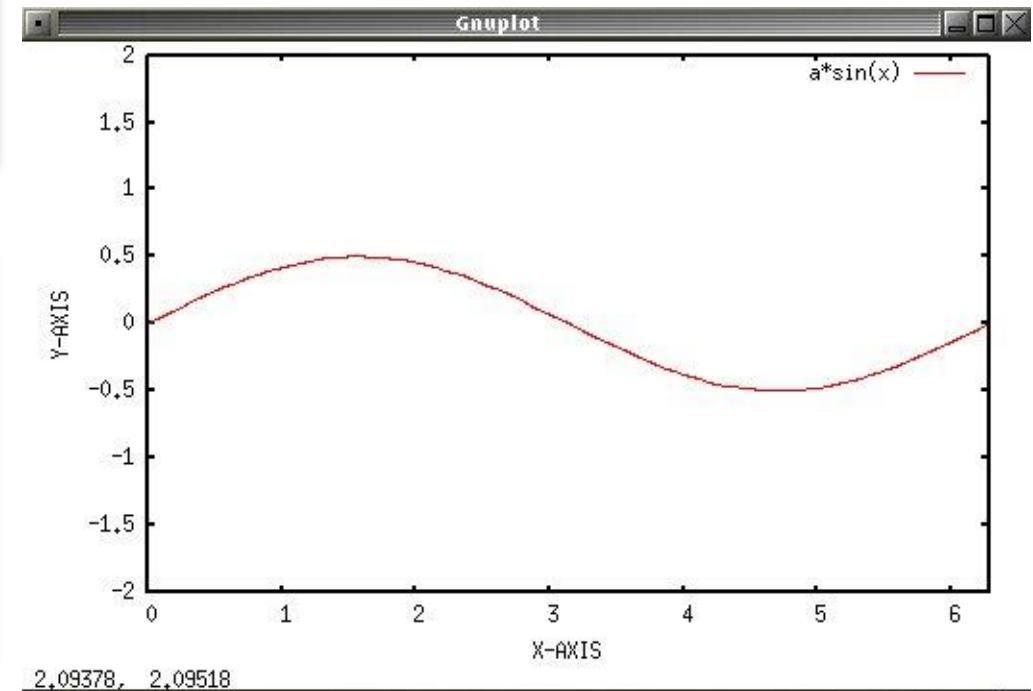
`gnuplot> variable = valore (oppure espressione).`

- Esempio:

```
gnuplot> a = 10  
gnuplot> b = 1+sqrt(a)  
gnuplot> print log(b)  
1.42606243890537
```

- Esempio:

```
gnuplot> a = 0.5  
gnuplot> set terminal x11  
gnuplot> f(x)=a*sin(x)  
gnuplot> plot f(x)
```



Gnuplot – plot di risultati numerici

- Supponiamo di aver bisogno di graficare dei risultati ottenuti con calcoli numerici.
- I risultati, prodotti da un programma che abbiamo scritto, sono salvati in un file.
- Esempio:

◆ abbiamo calcolato l'approssimazione di $y=\exp(-x)$ con il metodo di Pade:

```
gnuplot> !cat /home/agarfa/pexp.out
```

0.10	9.0484e-01	9.0484e-01	9.0484e-01
------	------------	------------	------------

...

4.90	7.4466e-03	-7.6597e-02	6.5886e-01
------	------------	-------------	------------

5.00	6.7379e-03	-7.8431e-02	6.8750e-01
------	------------	-------------	------------

↑ ↑ ↑ ↑
coordinata x $\exp(-x)$ Pade approx. a ordini differenti

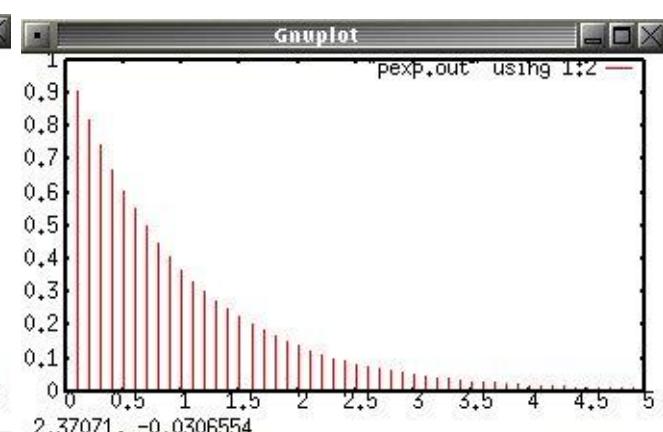
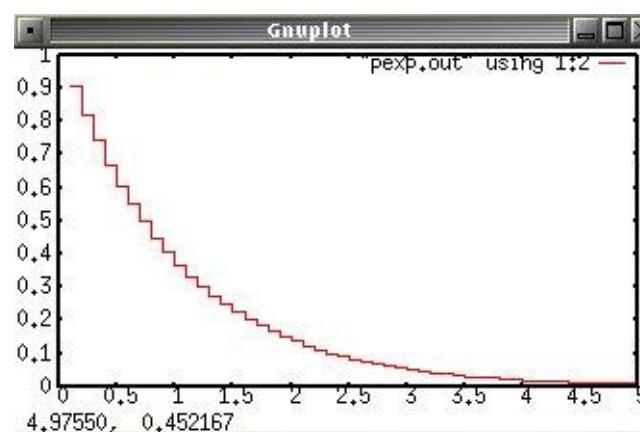
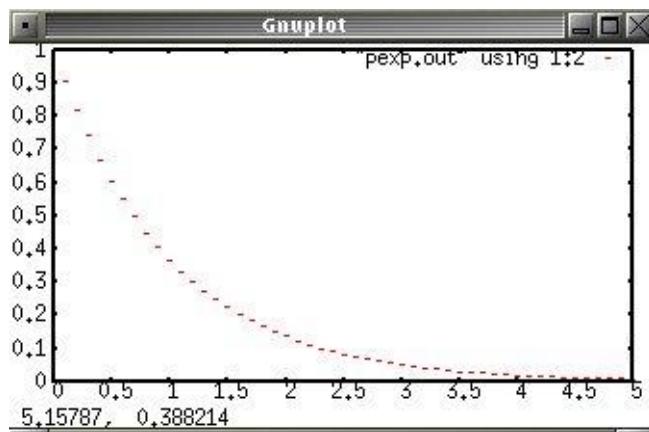
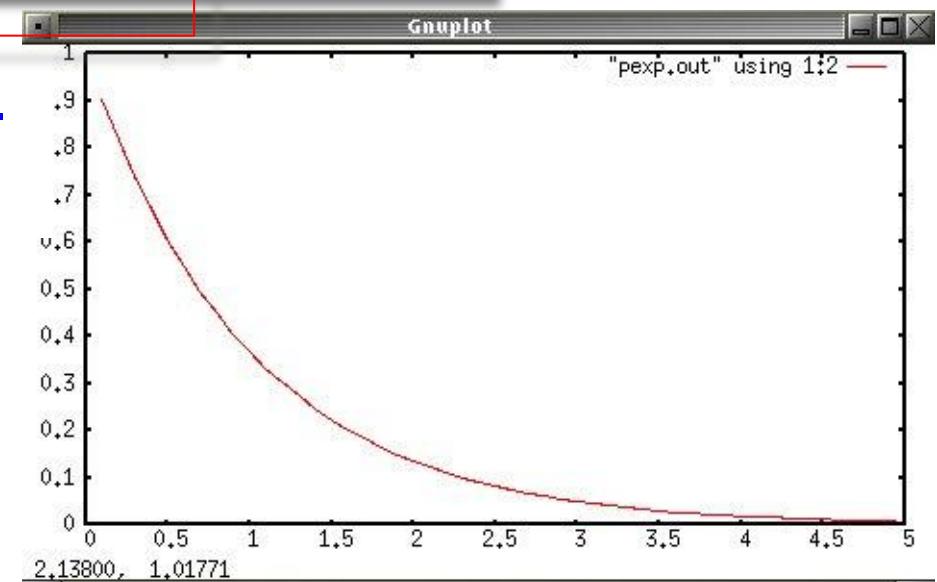
Gnuplot - plot di risultati numerici

```
gnuplot> plot "pexp.out" using 1:2 with lines
```

- **with** specifica lo stile del grafico.

◆ possibilità:

- **with points;**
- **with steps;**
- **with impulses;**

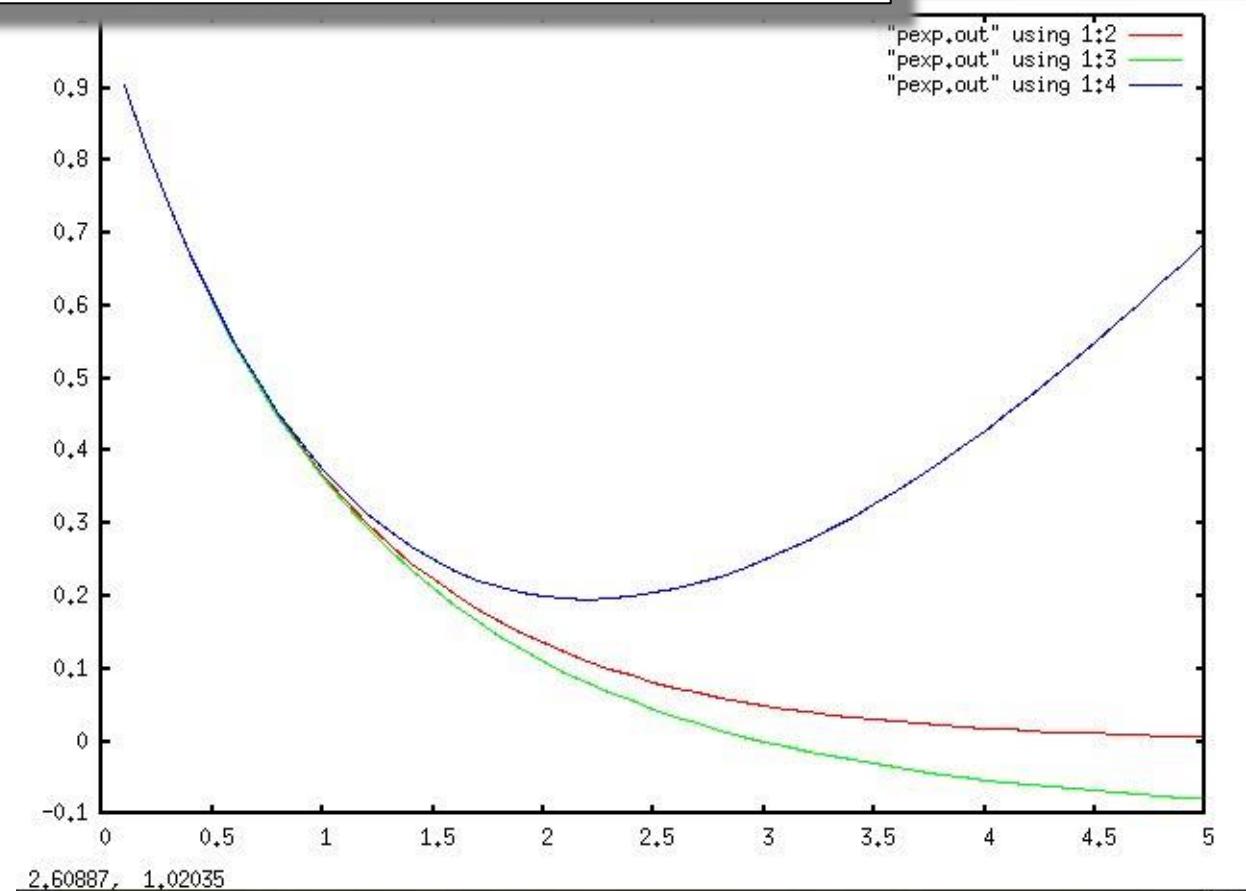


Gnuplot - plot di risultati numerici

- È possibile graficare più dati nello stesso plot:

```
gnuplot> plot "pexp.out" using 1:2 with lines, \
"pexp.out" using 1:3 with lines, \
"pexp.out" using 1:4 with lines
```

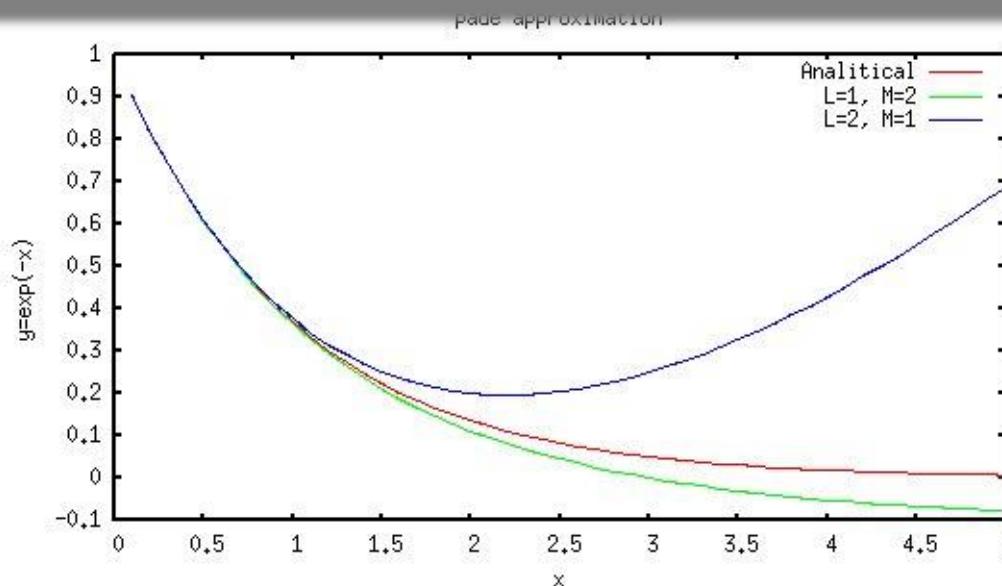
Il carattere backslash
\\ permette di
continuare il comando
alla linea seguente



Gnuplot - plot di risultati numerici

- Vogliamo aggiungere una “label” per le varie curve, e specificare i titoli per gli assi.

```
> set xlabel "x"
> set ylabel "y=exp(-x)"
> set title "Pade approximation"
> plot "pexp.out" using 1:2 title "Analytical" with lines,
> "pexp.out" using 1:3 title "L=1, M=2" with lines,
> "pexp.out" using 1:4 title "L=2, M=1" with lines
```

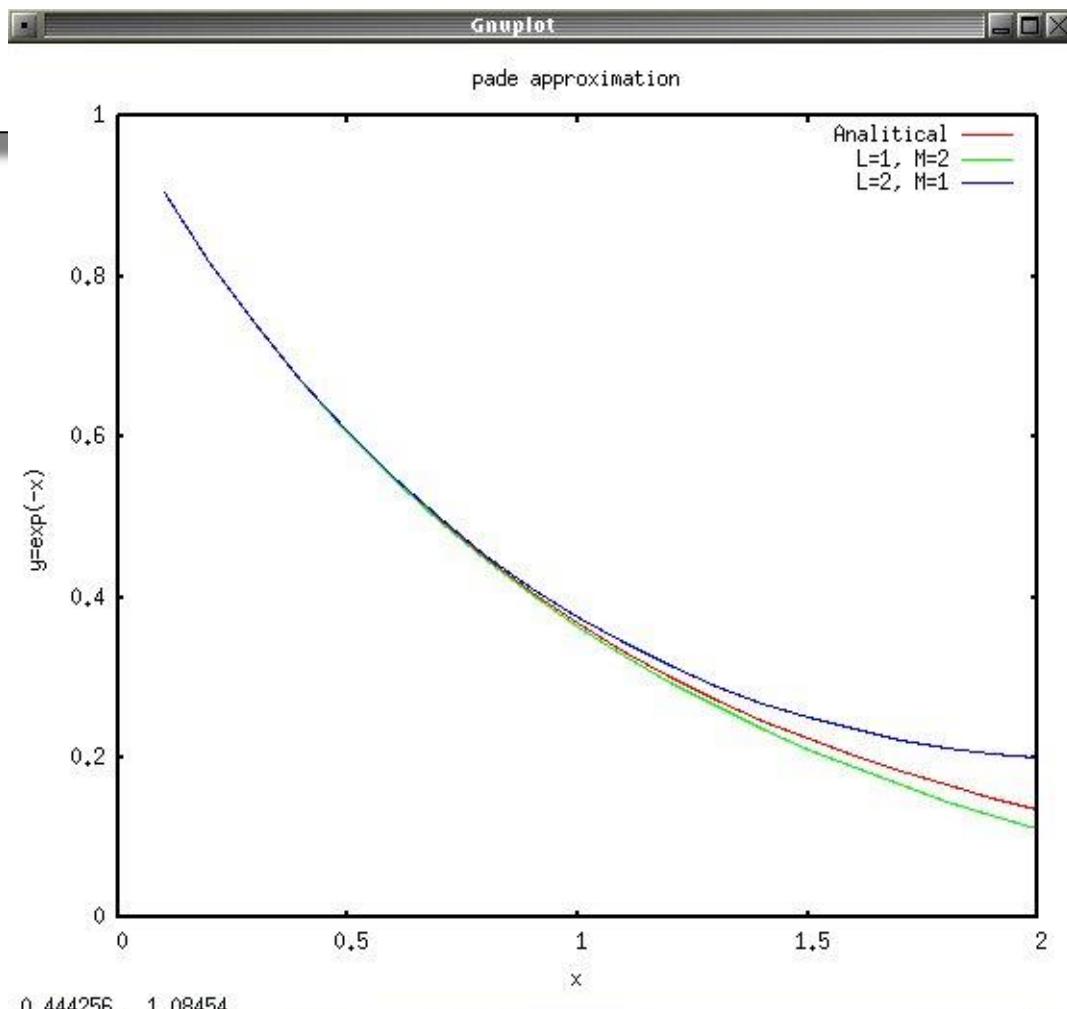


Gnuplot - plot di risultati numerici

- Desideriamo modificare soltanto i domini degli assi e riplottare

```
> set xrange [0:2]
> set yrange [0:1]
> replot
```

Si ricorda dei valori
specificati
precedentemente!



Gnuplot - plot di risultati numerici

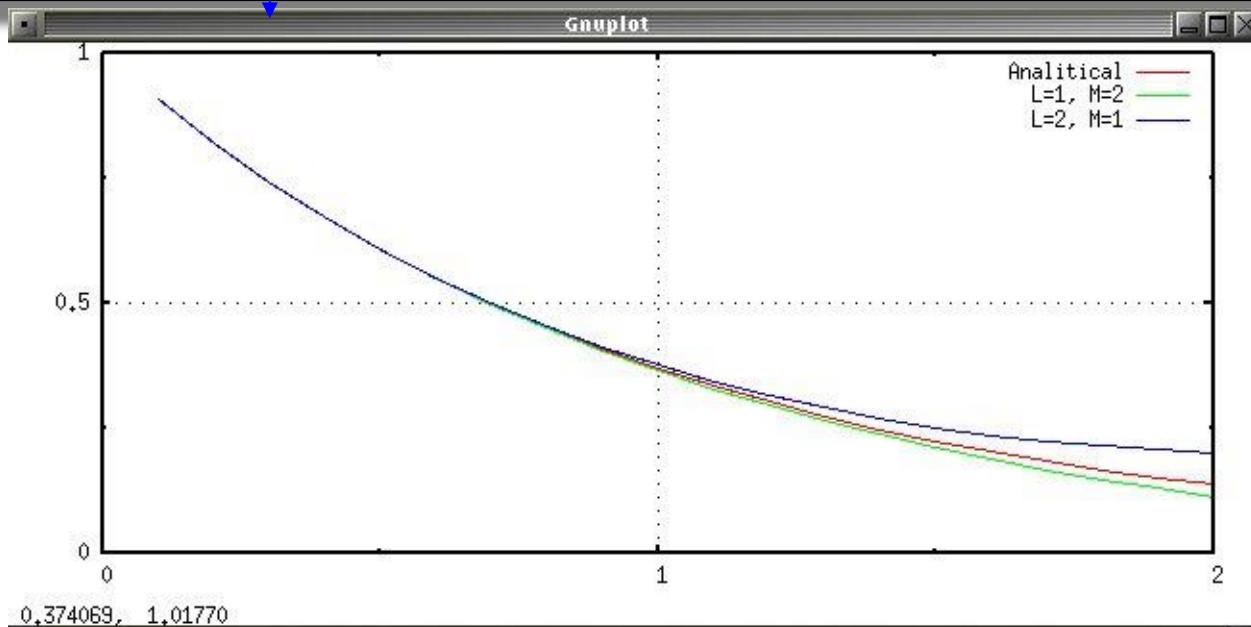
- Modifichiamo ora i “marker” degli assi:

- ◆ asse x tra 0 e 2 con etichette ad incremento di 1

```
> set xtics 1  
> set mxtics 2  
> set ytics 0.5  
> set mytics 2  
> set grid  
> replot
```

incremento

divisioni



Gnuplot - plot di risultati numerici

- Salviamo ora i risultati in un file postscript.

```
> set term postscript
```

```
Terminal type set to 'postscript'  
Options are 'landscape noenhanced monochrome blacktext \  
dashed dashlength 1.0 linewidth 1.0 defaultplex \  
palfuncparam 2000,0.003 \  
butt "Helvetica" 14'
```

```
> set output "pexp.ps"
```

```
> replot
```

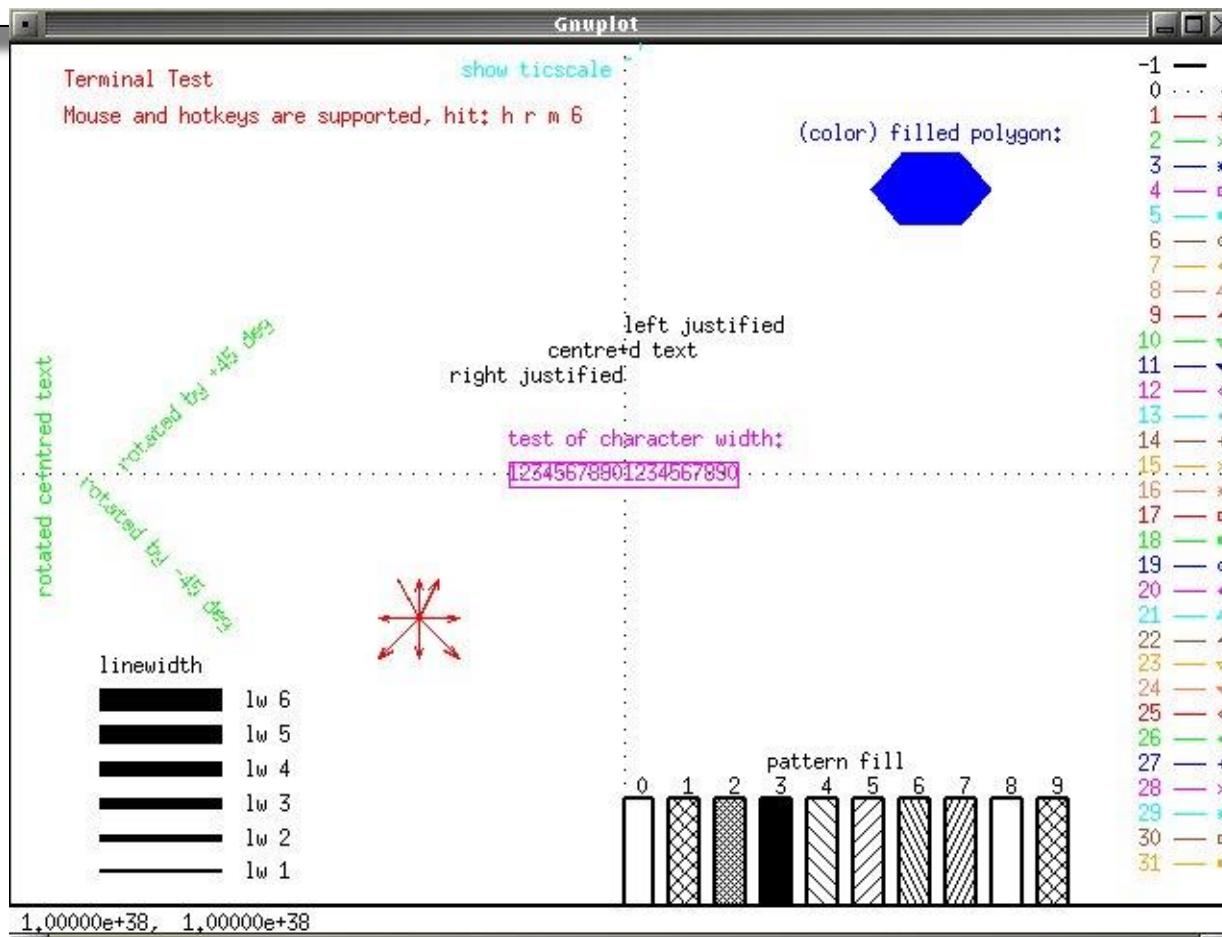
```
> save "pexp.plt"
```

```
> quit
```

Gnuplot - plot di risultati numerici

- Per verificare i parametri del sistema:

```
> set term x11  
> test
```



Gnuplot – dati sperimentali

- Supponiamo di avere in un file i risultati di una misura:

```
gnuplot> !cat plotexp.dat
# Data No. 1
2.1500E-02 1.3060E+00 5.3098E-02
...
1.1330E+00 1.2670E+00 4.2289E-02

# Data No. 2
2.4000E-02 1.2970E+00 3.1387E-02
...
7.4000E+00 1.1860E+00 2.5618E-02

# Data No. 3
2.2500E-02 1.3310E+00 3.4606E-02
...
2.1296E+01 1.0310E+00 3.1961E-02
```

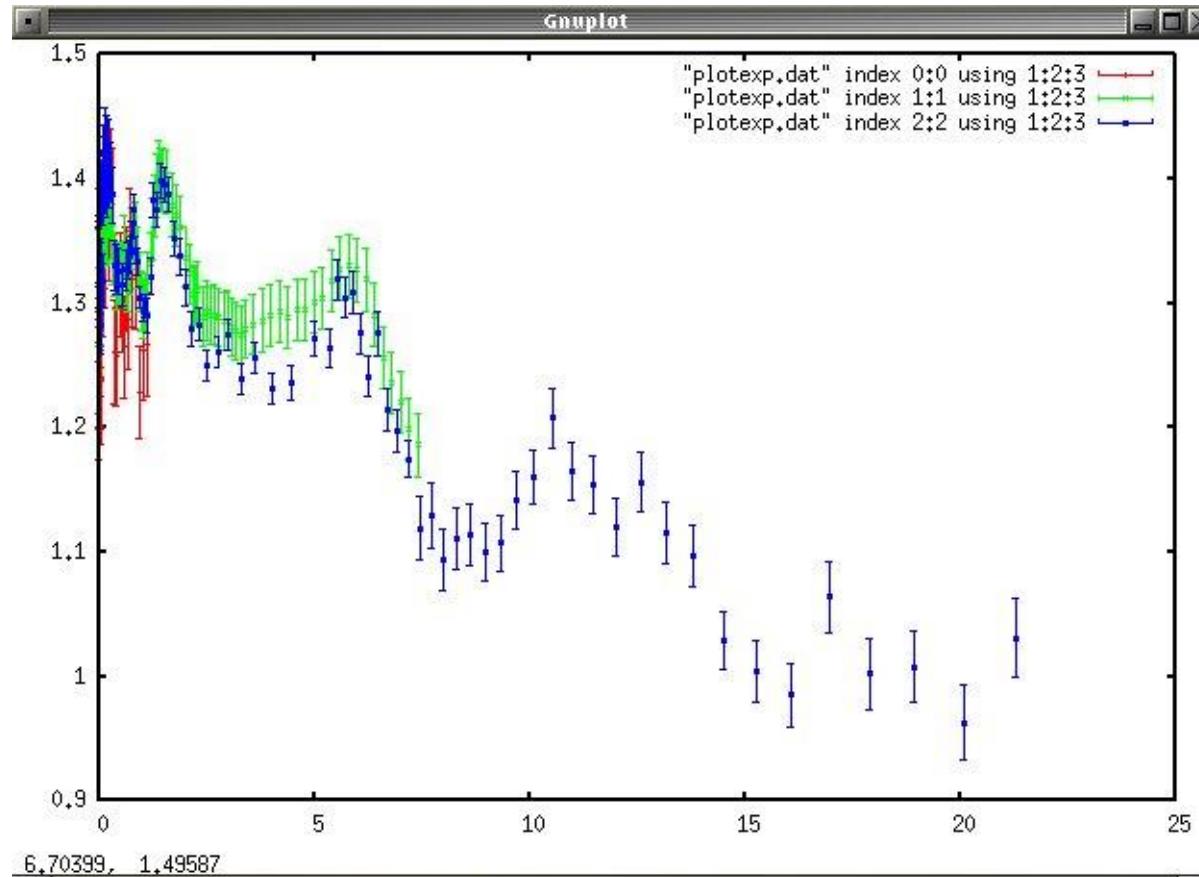
Tre set di dati.
Separati da due righe bianche

→ linea di commento

Energia Sezione d'urto Errore

Gnuplot – dati sperimentali

```
gnuplot> plot "plotexp.dat" index 0:0 using 1:2:3 with yerrorbars,\  
> "plotexp.dat" index 1:1 using 1:2:3 with yerrorbars,\  
> "plotexp.dat" index 2:2 using 1:2:3 with yerrorbars
```



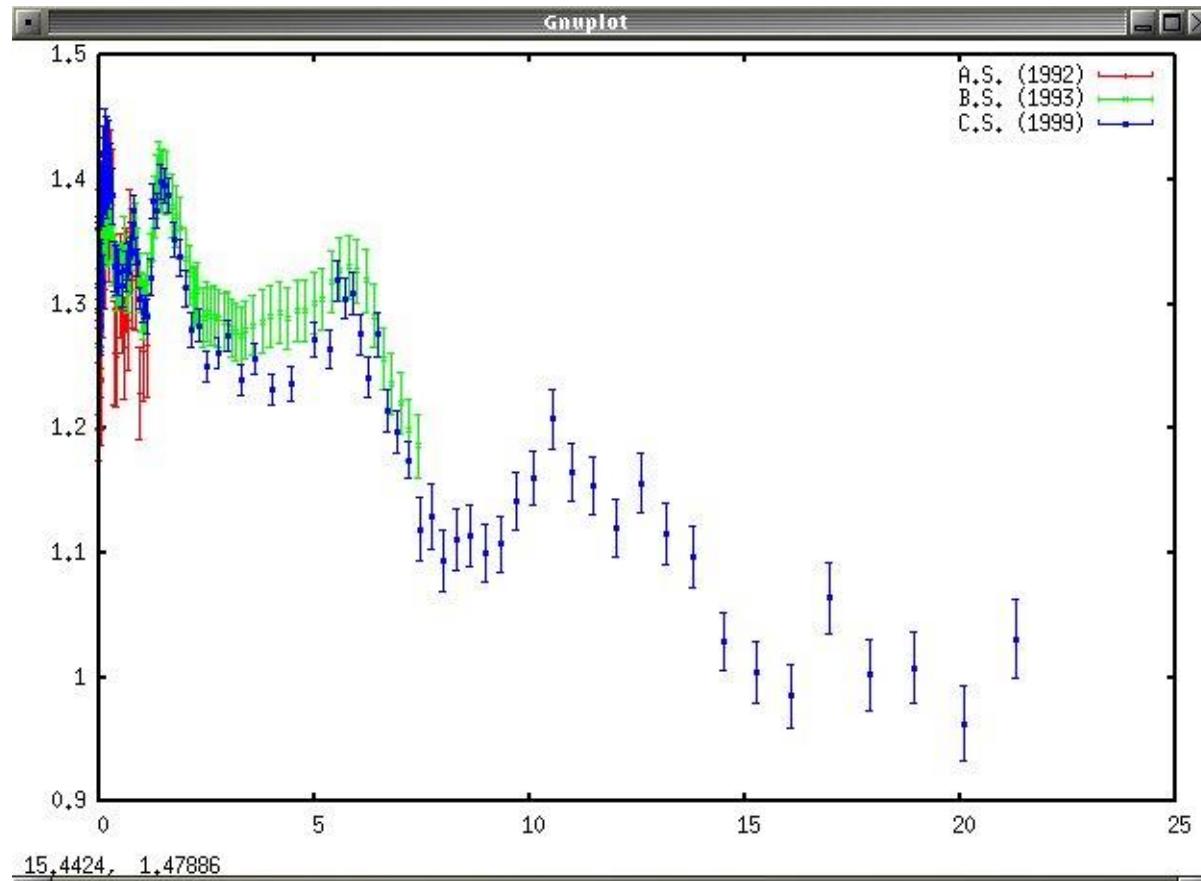
index

permette di
selezionare il
blocco di dati

with yerrorbars
specifica di graficare i
dati con l'errore.

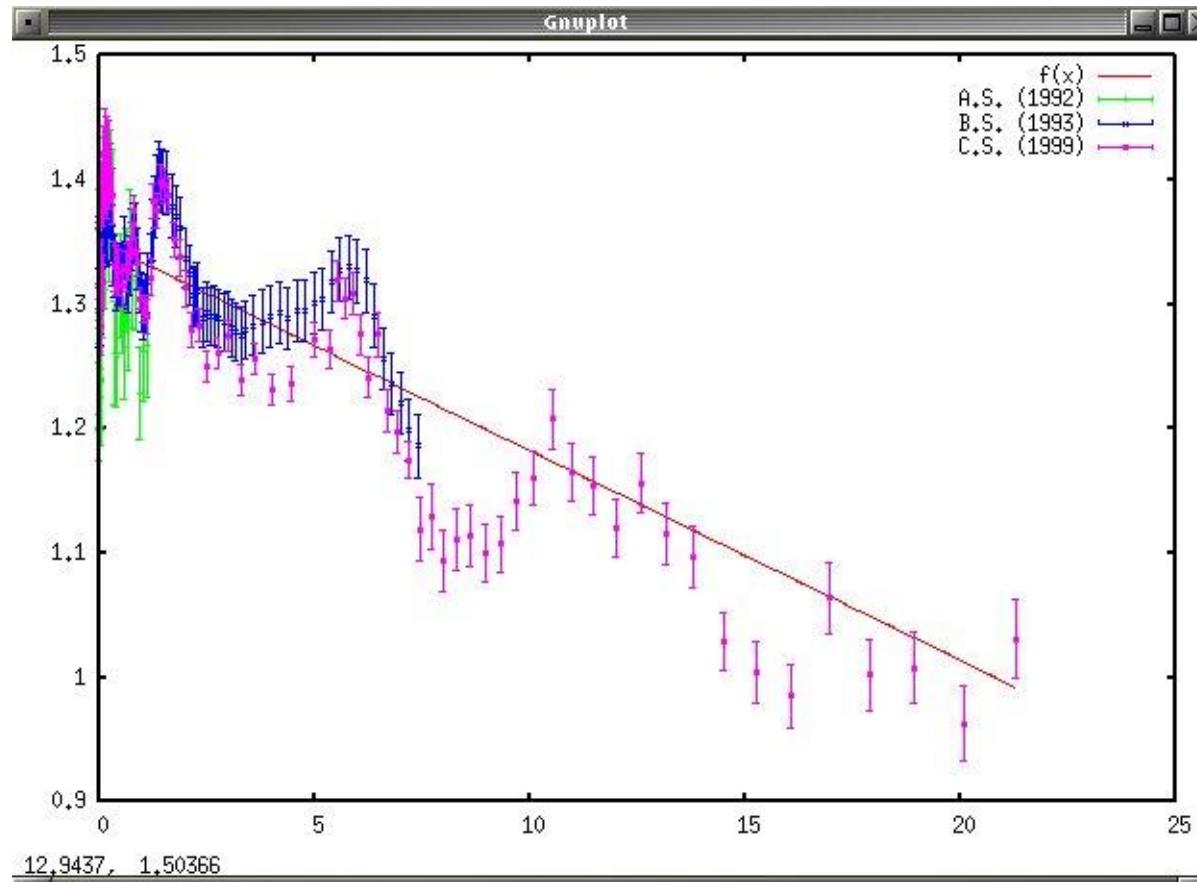
Gnuplot – dati sperimentali

```
gnuplot> plot "plotexp.dat" ind 0:0 usi 1:2:3 ti "A.S. (1992)" w yerr,\> "plotexp.dat" ind 1:1 usi 1:2:3 ti "B.S. (1993)" w yerr,\> "plotexp.dat" ind 2:2 usi 1:2:3 ti "C.S. (1999)" w yerr
```



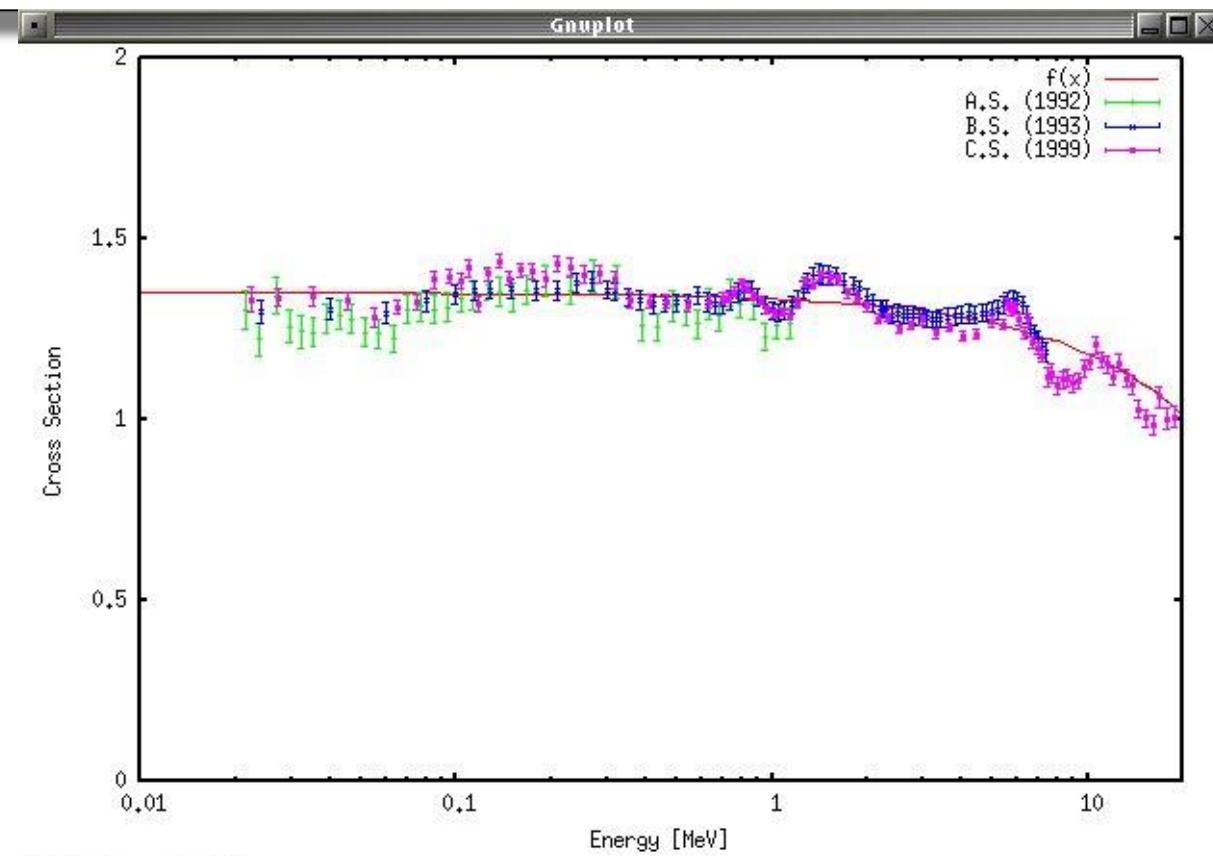
Gnuplot – dati sperimentali

```
gnuplot> f(x)= -0.01687*x + 1.3512
gnuplot> plot f(x) w lines \
> "plotexp.dat" ind 0:0 usi 1:2:3 ti "A.S. (1992)" w yerr,\ 
> "plotexp.dat" ind 1:1 usi 1:2:3 ti "B.S. (1993)" w yerr,\ 
> "plotexp.dat" ind 2:2 usi 1:2:3 ti "C.S. (1999)" w yerr
```



Gnuplot – dati sperimentali

```
gnuplot> set xrange [0.01:20]
gnuplot> set yrange [0:2]
gnuplot> set logscale x
gnuplot> set xlabel "Energy [MeV]"
gnuplot> set ylabel "Cross Section"
gnuplot> replot
```



Gnuplot - functions

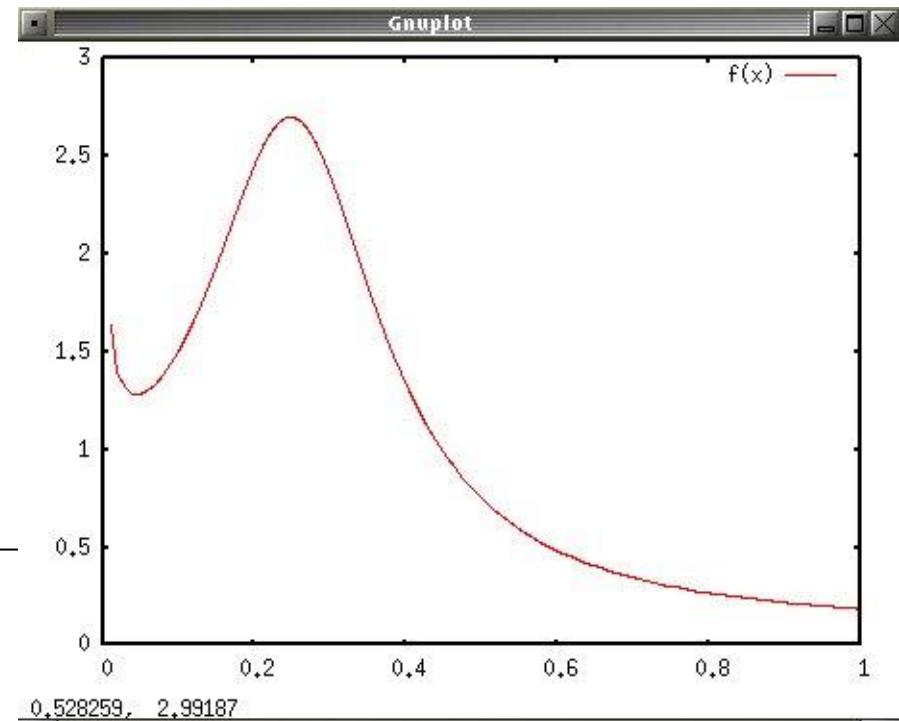
- Supponiamo di voler graficare delle misure e di sovrapporre una funzione, stimando i parametri che meglio descrivano i dati.
- ◆ Consideriamo una Lorenziana alla quale aggiungiamo una funzione $1/\sqrt{x}$:

$$f(x) = \frac{c}{(x - a)^2 + b} + \frac{d}{\sqrt{x}}$$

```

gnuplot> a = 0.25
gnuplot> b = 0.02
gnuplot> c = 0.05
gnuplot> d = 0.1
gnuplot> f(x)=c/((x-a)*(x-a)+b)+d/sqrt(x)
gnuplot> plot f(x)

```



Gnuplot - functions

- È possibile calcolare i valori di una funzione e visualizzarli con il comando **print**.

```
gnuplot>print f(0.25)
2.7
gnuplot> print f(0.4)
1.33458447124371
gnuplot> a = 0.4
gnuplot> print f(0.4)
2.65811388300842
```

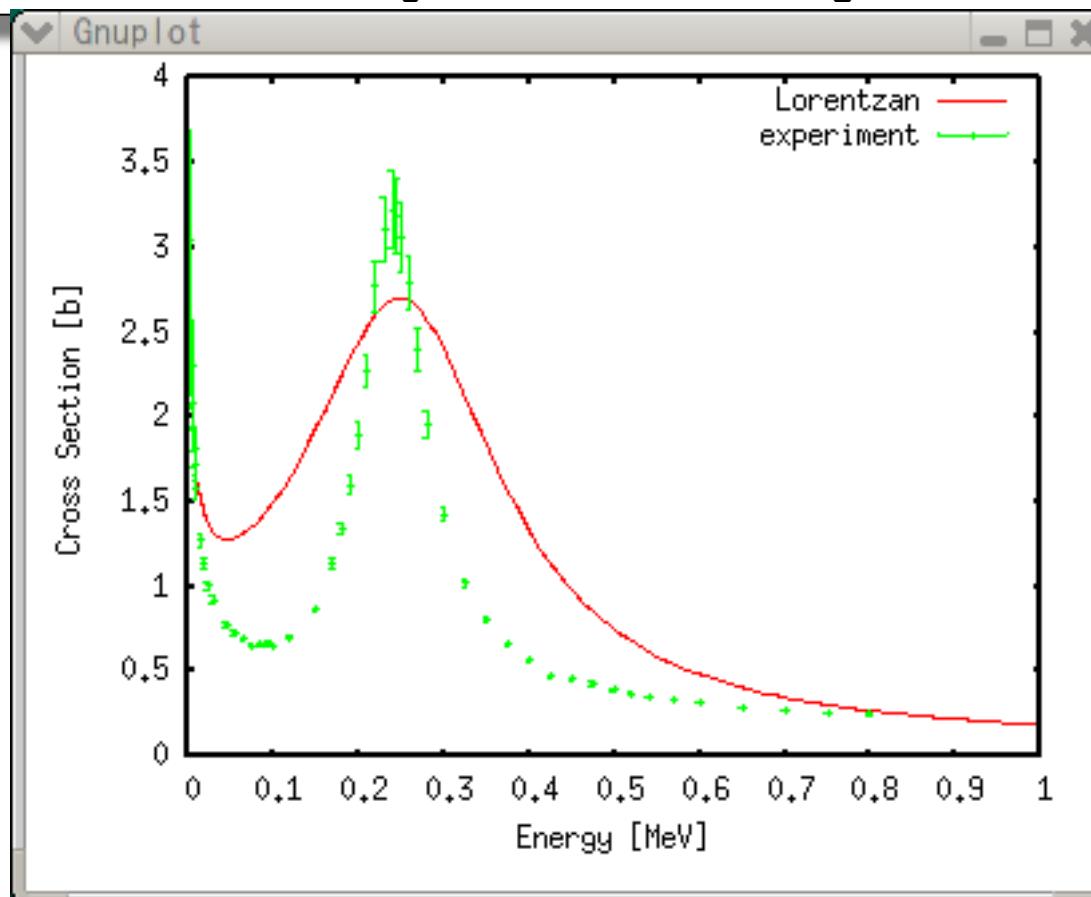
- È possibile ottenere i valori come tabella e salvarli in un file.

```
gnuplot> set term table
Terminal type set to 'table'
gnuplot> plot f(x)
#Curve 0, 100 points
#x y type
0 0 u
0.010101 1.63972 i
.....
1 0.185837 i
gnuplot> set output "calc.plt"
gnuplot> replot
```

Gnuplot - functions

- Adesso grafichiamo i dati salvati e la funzione.

```
gnuplot> set xlabel "Energy [MeV]"
gnuplot> set ylabel "Cross Section"
gnuplot> plot f(x) tit "Lorentzan" \
>"exp.dat" using 1:2:3 tit "experiment" with yerr
```



Gnuplot – fitting data

- Il comando fit permette di calcolare un fit ai minimi quadrati con

```
gnuplot> fit f(x) "exp.dat" using 1:2:3 via a,b,c,d
```

```
Iteration 0
WSSR      : 96618.1          delta(WSSR)/WSSR   : 0
delta(WSSR) : 0              limit for stopping : 1e-05
lambda    : 1150.73

initial set of free parameter values
...
After 17 iterations the fit converged.
final sum of squares of residuals : 3341.93
rel. change during last iteration : -5.29173e-06

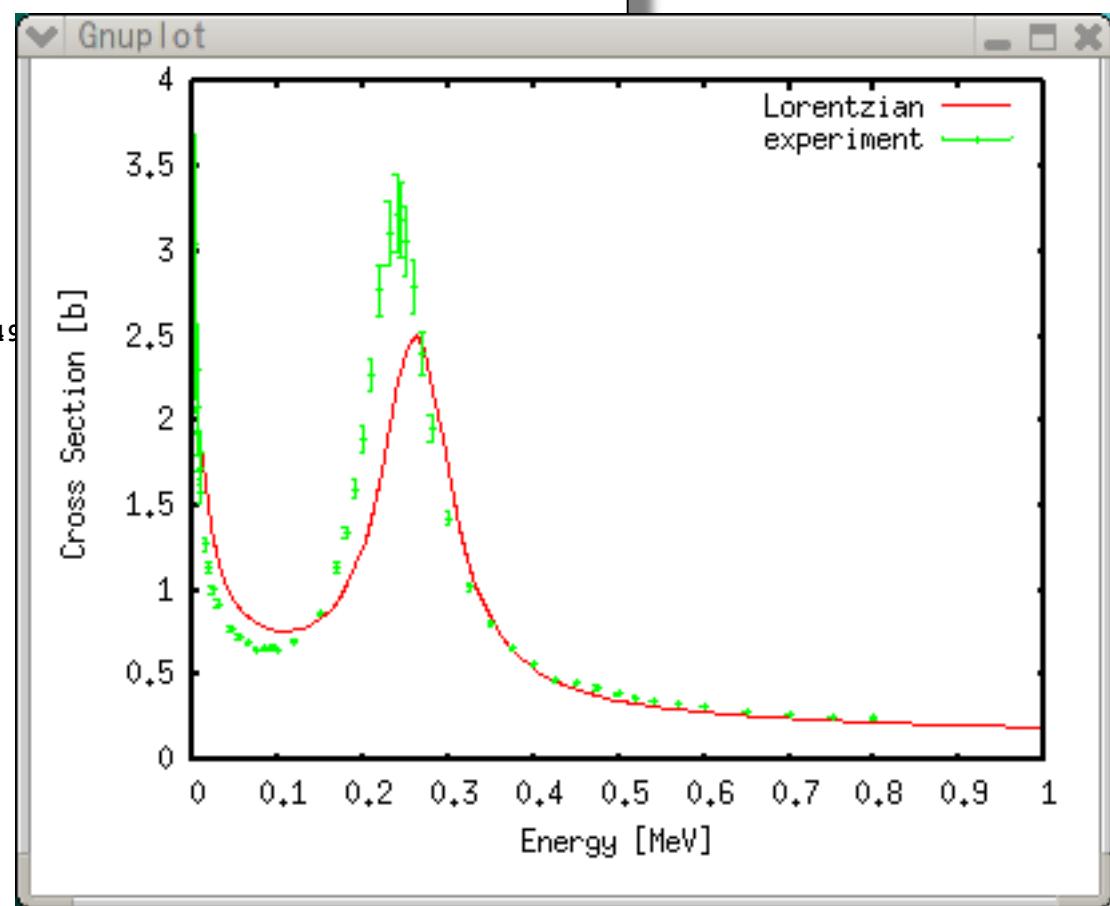
degrees of freedom (ndf) : 47
rms of residuals (stdfit) = sqrt(WSSR/ndf)   : 8.43237
variance of residuals (reduced chisquare) = WSSR/ndf : 71.1049

Final set of parameters      Asymptotic Standard Error
=====      =====
a = 0.26191      +/- 0.005759      (2.199%)
b = 0.00251445  +/- 0.0008358     (33.24%)
c = 0.00541346  +/- 0.0009206     (17.01%)
d = 0.182469    +/- 0.007329      (4.016%)

correlation matrix of the fit parameters:

      a      b      c      d
a  1.000
b  0.042  1.000
c -0.229  0.783  1.000
d  0.210 -0.538 -0.768  1.000

gnuplot> replot
```



Gnuplot – fitting data

■ Modifichiamo la funzione di background

```
gnuplot> e = -0.5  
gnuplot> f(x)=c/((x-a)*(x-a)+b)+d*x**e  
gnuplot> fit f(x) "exp.dat" using 1:2:3 via a,b,c,d,e
```

```
...  
Final set of parameters      Asymptotic Standard Err  
=====  =====  
a = 0.25029      +/- 0.002106      (0.8412%)  
b = 0.00197707    +/- 0.0002747     (13.89%)  
c = 0.00550098    +/- 0.0003662     (6.657%)  
d = 0.21537      +/- 0.003743      (1.738%)  
e = -0.358371     +/- 0.0115       (3.208%)
```

```
correlation matrix of the fit parameters:
```

	a	b	c	d	e
a	1.000				
b	0.021	1.000			
c	-0.078	0.788	1.000		
d	-0.110	-0.384	-0.500	1.000	
e	-0.304	0.198	0.335	0.381	1.000

```
gnuplot> replot
```

