

# Open Source & Open Science

Francesco Corona

GULCh



LinuxDay 2003  
Cagliari

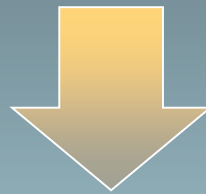


# Motivazioni

## Metodo Scientifico

Processo di Scoperta

Processo di Dimostrazione



Replicabilità

Verificabilità

Il processo che ha portato ad una scoperta deve essere reso pubblico

Diffusione delle Informazioni

Condivisione delle Idee



# Motivazioni

Continua

Verificabili

Ripetibili

Open Source

Condivisi



LinuxDay 2003  
Cagliari



# Agenda

## High Level Languages (HLL)

Fortran

## Scientific Plotting

gnuplot

## Very High Level Languages (VHLL)

Octave

Scilab



LinuxDay 2003  
Cagliari

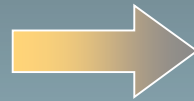


# High Level Languages

Fortran

Consente di scrivere programmi più simili al linguaggio umano che non al linguaggio macchina

I linguaggi di programmazione ad alto livello, **HLL**, risalgono agli inizi degli anni '50



Ada, Algol, BASIC, COBOL, C, C++, FORTRAN, LISP, Pascal e Prolog

Il FORTRAN, **FOR**mula **TRAN**slator, è il più vecchio linguaggio di programmazione ad alto livello

**E' facile da imparare ed è stato progettato per applicazioni scientifiche e ingegneristiche**

It's crunch time!



LinuxDay 2003  
Cagliari



# Fortran

g77

GNU Fortran **g77** è l'ambiente di sviluppo Fortran del Progetto GNU



[www.gnu.org](http://www.gnu.org)

Sistema completo che include un compilatore, librerie run-time e il supporto per il debugger

Supporto completo per l'ANSI Fortran77, per il l'ANSI/ISO Fortran90 e per alcuni dialetti

It's **FREE** crunch time!



LinuxDay 2003  
Cagliari



# Fortran

g77

```
program bill4tux
  real Acqua, Vino
  real Primo, Secondo, Contorno
  real Dolce, Frutta, Caffè
  real Ordine1, Ordine2, Conto

! MENU' del GIORNO

  Acqua      = 0.50
  Vino       = 1.00
  Primo      = 2.00
  Secondo    = 2.50
  Contorno   = 1.50
  Dolce      = 1.25
  Frutta     = 1.25
  Caffè      = 0.25

  call waitress (Acqua,Vino,Primo,Ordine1)
  write (*,*) 'Il primo ordine costa ', Ordine1
  call waitress (Secondo,Frutta,Caffè,Ordine2)
  write (*,*) 'Il secondo ordine costa', Ordine2

  Conto = Ordine1 + Ordine2
  write(*,*) 'Il Conto ', Conto

stop
end

subroutine waitress (Item1,Item2,Item3,Ordine)
  real Item1, Item2, Item3
  real Totale, Mancia, Ordine

  Totale = Item1 + Item2 + Item3
  Mancia = 0.15 * Totale
  Ordine = Totale + Mancia

return
end
```

Il **g77** è fornito con il **gcc**

```
$ g77 -o bill4tux bill4tux.f
$ ./bill4tux
```

```
$ g77 -o bill4tux bill4tux_m.f bill4tux_s.f
$ ./bill4tux
```

Possibilità di crearsi librerie  
personali di routines

```
$ g77 -c waitress.f
$ ls
bill4tux      bill4tux.f
bill4tux_m.f  bill4tux_s.f
waitress.f    waitress.o
```

```
$ g77 -o bill4tux bill4tux.f waitress.o
$ ./bill4tux
```



# Fortran

Librerie

La vera potenza del Fortran è rappresentata dalle sue librerie

Netlib è una miniera di software matematico disponibile  
per la comunità scientifica

**SLATEC**

Matematica

Algebra Lineare

**LAPACK**

**ODEPACK**

Eq. Diff. Ordinarie

Trasformate di Fourier

**FFTPACK**



[www.netlib.org](http://www.netlib.org)



LinuxDay 2003  
Cagliari

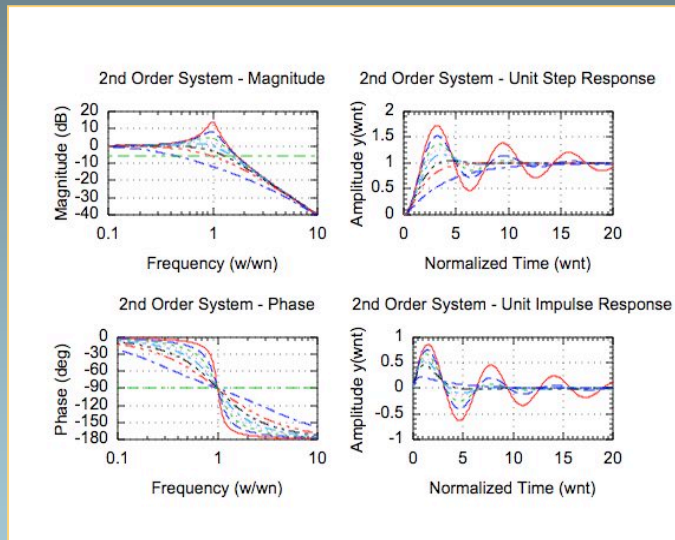




# Scientific Plotting

gnuplot

Programma Interattivo a Linea di Comando  
Per Diagrammare Funzioni e Dati



## Funzionalità

Diagrammi 2D di curve e dati

Diagrammi 3D di dati e superfici

Calcoli aritmetici

Definizione di funzioni personalizzate

[www.gnuplot.info](http://www.gnuplot.info)

Possibilità di compilare il codice sorgente in qualsiasi ambiente standard  
(ANSI/ISO C, POSIX)



LinuxDay 2003  
Cagliari



# gnuplot

2D

L'istruzione per diagrammare grafici bidimensionali è intuitiva

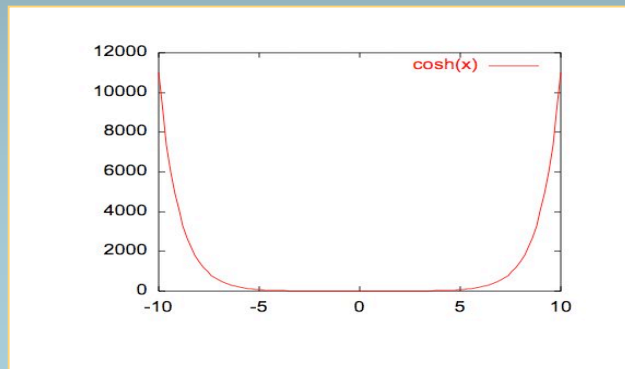
**plot**

e la sintassi più semplice per utilizzarla è:

**plot <function>**

Ad esempio, per diagrammare la funzione  $y = \cosh(x)$

```
gnuplot> plot cosh(x)
```

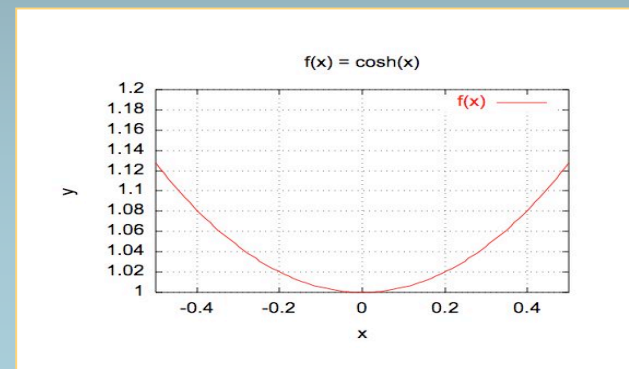


```
gnuplot> set title "f(x) = cosh(x)"
gnuplot> set xlabel "x"
gnuplot> set ylabel "y"
gnuplot> set grid
gnuplot> f(x) = cosh(x)
```

...e nell'intorno dello 0?

L'istruzione **plot** consente di definire l'intervallo di visualizzazione

```
gnuplot> plot [-0.5:0.5] [1.0:1.2] f(x)
```



# gnuplot

3D

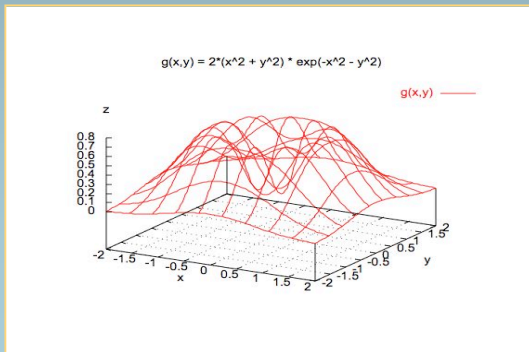
L'istruzione per diagrammare  
funzioni tridimensionali  
è altrettanto intuitiva

**splot**

e la semplicità di utilizzo della sintassi  
è la medesima:

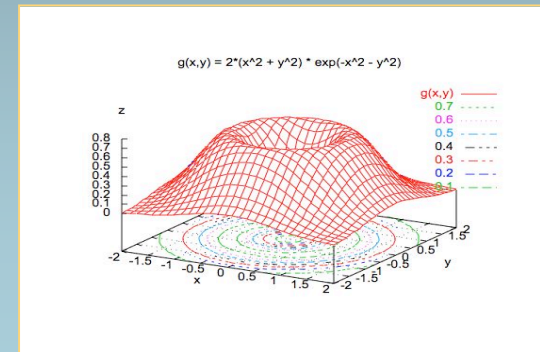
**splot <function>**

```
gnuplot> f(x,y) = x**2 + y**2
gnuplot> g(x,y) = 2*(f(x,y))*exp(-f(x,y))
gnuplot> splot [-2:2] [-2:2] g(x,y)
```



Anche in questo caso può essere  
necessario migliorare la qualità

```
gnuplot> set isosamples 30,30
gnuplot> set contour base
gnuplot> set hidden3d
gnuplot> replot
```



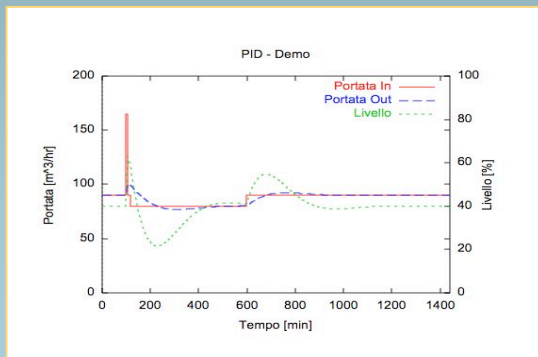
# gnuplot

## Data

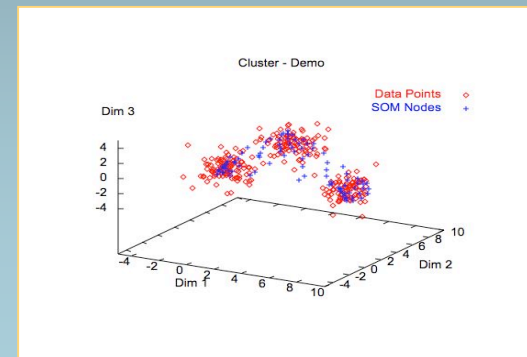
```
gnuplot> !pwd  
/home/fcorona/LD2K3/gnuplot/data  
gnuplot> !ls  
PID.r codebooks.r data.d
```

Diagrammare con **plot** e **splot** uno o più file di dati mono- e bi- e tridimensionali

```
gnuplot> plot [1:1440]  
"PID.r" using 1 smooth unique title "Portata In",  
"PID.r" using 2 smooth unique title "Portata Out",  
"PID.r" using 3 smooth unique title "Livello"
```



```
gnuplot> splot "data.d" title "Data Points",  
"codebooks.r" title "SOM Nodes"
```



# gnuplot

Stampa

## Stampa su carta

Utile se avete bisogno di una copia cartacea del grafico

gnuplot fornisce numerosi strumenti per conservare il lavoro fatto

## Stampa su file PostScript

(.ps)

Utile se avete bisogno di numerose copie cartacee del vostro lavoro e non volete ri-diagrammare i vostri grafici ogni volta

(.eps)

Utile se avete necessità di includere i vostri grafici ad alta definizione in documenti di testo come relazioni

## Stampa su altro file grafico

Utile se volete pubblicare una pagina web o una presentazione con il risultato dei vostri studi

(.png)



# Very High Level Languages

Octave & Scilab

$$C(m,n) = A(m,p) \cdot B(p,n)$$

## Codice Fortran

```
do i = 1, m
  do j = 1, n
    do k = 1, p
      C(i,j) = C(i,j) + A(i,k) * B(k,j)
    end do
  end do
end do
```

## Codice C/C++

```
for( i=0 ; i<m ; i++)
  for( j=0 ; j<n ; j++)
    for( k=0 ; k<p ; k++)
      C[i][j] = A[j][k] * B[k][i]
    }
  }
}
```

Qualche volta vogliamo semplificarci la vita

## Codice VHLL

$$C = A * B$$

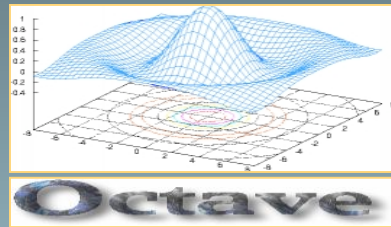


LinuxDay 2003  
Cagliari



# Octave

VHLL per l'analisi numerica  
di problemi lineari e non lineari



[www.octave.org](http://www.octave.org)

[octave.sourceforge.net](http://octave.sourceforge.net)

## Funzionalità

Problemi di Algebra Lineare

Soluzione di Equazioni Non Lineari

Integrazione di Funzioni

Funzioni Polinomiali

Equazioni Differenziali Ordinarie

Equazioni Differenziali-Algebriche

Librerie Fortran Standard: LAPACK, LINPACK, ODEPACK, BLAS etc.

Capacità Grafiche 2D e 3D

gnuplot



LinuxDay 2003  
Cagliari



# Octave

## Matrici

Inserire e manipolare matrici  
costituisce la forza di GNU Octave

```
octave:1> a = [ 1 2 ; 3 4 ]
a =
    1  2
    3  4
octave:2> b = [ 3 4 ; 1 2 ];
octave:3> c = a * b
c =
    5  8
   13 20
octave:4> d = a. * b
d =
    3  8
    3  8
octave:5> e = a + (2 * b)
e =
    7  10
    5  8
```

Oltre ai semplici operatori aritmetici  
esistono numerose funzioni  
per eseguire operazioni matriciali

```
octave:6> f = rand (3,2);
octave:7> g = f (1,1)
g = 0.44234
octave:8> h = e (:,1)
h =
    0.44234
    0.70343
    0.76667
```

La funzione **rand** genera numeri  
casuali utilizzando RANLIB

La funzione **eig** calcola  
gli autovalori e gli autovettori  
con LAPACK e LINPACK

```
octave:9> [eigval,eigvec] = eig(a)
eigval =
   -0.82456   -0.41597
    0.56577   -0.90938

eigvec =
   -0.37228    0.00000
    0.00000    5.37228
```





# Octave

## Polinomi

In Octave i polinomi sono rappresentati dal vettore dei coefficienti

$$x^3 + 2x^2 + 3x - 1$$



```
octave:1> pol = [ 1 2 3 -1]
pol =
     1     2     3    -1
octave:2> x = 2
octave:3> y = polyval (pol,x)
y = 21
```

La funzione **polyval** consente di valutare il polinomio per un dato valore della variabile

Con **polyderiv** e **polyinteg** è possibile differenziare ed integrare i polinomi (Octave usa zero come costante di integrazione)

```
octave:4> pold = polyderiv (pol)
pold =
     3     4     3
octave:5> poli = polyinteg (pol)
poli =
     0.25000     0.66667     1.5000    -1.00000     0.00000
```

.... e **roots** calcola gli zeri!

```
octave:6> zeri = roots (pol);
octave:7> size (zeri)
ans =
     3     1
octave:8> zeri
zeri =
    -1.13784 + 1.52731i
    -1.13784 - 1.52731i
     0.27568 + 0.00000i
```



# Octave

Funzioni & Toolboxes

Con la funzione **function** è possibile definire routines personalizzate

```
function [uscite] = nome (ingressi)
...
...
endfunction
```



```
function [sum,diff] = sumdiff (x,y)
    sum = x + y
    diff = x - y
endfunction
```

GNU Octave è un linguaggio completo con funzioni per il controllo di flusso, cicli e input-output

Algebra Lineare e Matematica

Ottimizzazione e Statistica

Teoria del Controllo e Trattamento dei Segnali

... e tanti altri!



LinuxDay 2003  
Cagliari



# Scilab

Ambiente di sviluppo per applicazioni numeriche  
di tipo scientifico e ingegneristico



## Funzionalità

Grafici 2D e 3D e Animazioni  
Integrazione di Eq. Differenziali  
Ottimizzazione

Trattamento dei Segnali  
Interfaccia con altri Linguaggi  
Simulatore di Sistemi Dinamici

Librerie Fortran Standard; Netlib

[www.scilab.org](http://www.scilab.org)

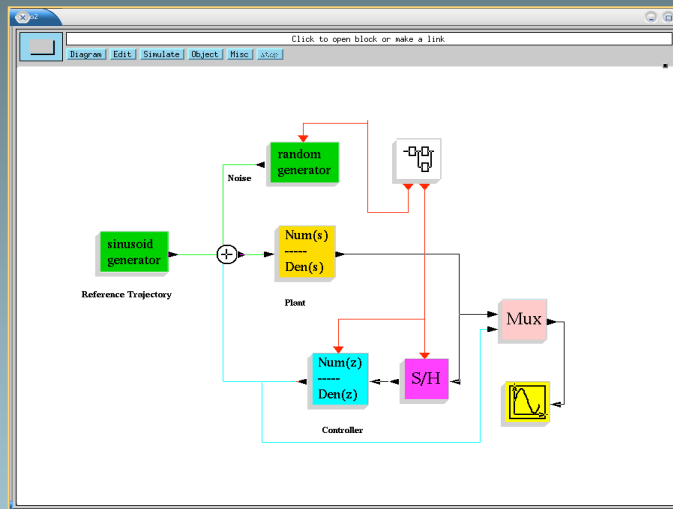
LinuxDay 2003  
Cagliari



# Scilab

Scicos

## SCIlab Connected Object Simulator



Programma per la Modellazione e  
la Simulazione Dinamica  
di Sistemi Continui e Discreti

**Editore Grafico e Simulatore**  
Costruzione di modelli mediante  
blocchi interconnessi che generano i  
segnali

I segnali di attivazione ai blocchi generano segnali di uscita in  
base a funzioni predefinite o personalizzabili

[www.scicos.org](http://www.scicos.org)

LinuxDay 2003  
Cagliari



# Conclusioni

”Convincete professori, post-dottorati, dottorandi, studenti, i vostri colleghi e voi stessi che la buona scienza si basa su risultati che sono verificabili e la reale verifica non è possibile senza codice sorgente pubblicamente disponibile”

Prof. J. D., Gezelter - University of Notre Dame

**The Open Science Project**

[www.openscience.org](http://www.openscience.org)



**Scientific Application for Linux**

[sal.kachinatech.com](http://sal.kachinatech.com)



LinuxDay 2003  
Cagliari





*Open Source Software, always better!*



LinuxDay 2003  
Cagliari

