

# Software di supporto all'insegnamento della Matematica

(autore: Vittorio Albertoni)

## Premessa

Si dice che Einstein non abbia mai imparato le tabelline e abbia vissuto con le tabelline scritte su lavagnette consultabili all'occorrenza. Al contrario di Gauss che pare invece fosse un abilissimo realizzatore di calcoli mentali. Ciò dimostra che si può essere grandi matematici indipendentemente dalla propria abilità nel fare calcoli.

Pertanto il pericolo che l'uso di strumenti automatici di calcolo porti lo studente a disimparare a calcolare con carta e penna non è affatto un pericolo, anzi...

Il calcolo in sé infatti, si tratti di calcolare un prodotto tra due interi o si tratti di calcolare una derivata, altro non è che la pedissequa esecuzione di regole e algoritmi mandati a memoria e, nei casi di maggiore complicazione, può comportare un assurdo e distraente spreco di risorse che, soprattutto in fase di apprendimento, andrebbero più concentrate sul perché si deve fare un certo calcolo e non tanto sul come lo si deve fare.

L'insistenza nel far imparare l'algoritmo di Ruffini per la fattorizzazione di un polinomio e nel far esercitare ripetutamente lo studente su questo rischia di far dimenticare come e perché la fattorizzazione del polinomio porti a trovare le radici di questo. Invece l'importante è capire questo concetto, il calcolo poi lo faccia pure una macchina.

Lo stesso vale per derivate e integrali: esercizi su esercizi per calcolarne anche in situazioni sempre più complicate serve solo ad allontanare lo studente dai sublimi segreti del calcolo differenziale, che sono l'unica cosa che conta veramente sul piano culturale.

Che dire poi dei sistemi di equazioni lineari. Con Cramer arriviamo a risolverne a tre incognite. Proviamo con carta e penna a risolvere un sistema con sette o otto incognite.

Se solo pensiamo a quanto tempo si possa perdere con esercitazioni e prove scritte pretendendo di insegnare a fare tutte queste cose con carta e penna ed a quanto se ne risparmierebbe delegandole a una macchina per concentrarsi su cose più concettuali ed importanti...

Stesso discorso per la geometria se unicamente approcciata con riga, squadra e compasso quando la grafica sviluppabile con un computer può consentire un approccio più vivo e interattivo.

Ben venga allora la pretesa che l'insegnante di materie scientifiche sia lui stesso portatore e divulgatore nella scuola dei moderni strumenti che l'informatica ci mette a disposizione ormai in grande abbondanza.

E' proprio la grande abbondanza e l'esistenza di software vecchi e obsoleti accanto a software più recenti ed efficienti che rende opportuna una piccola guida di orientamento alla scelta ed è ciò che modestamente cercherò di fare in questo manualetto.

# Indice

<b>1</b>	<b>Geometria dinamica</b>	<b>3</b>
1.1	Geogebra . . . . .	3
1.2	Cabri . . . . .	4
1.3	Cinderella . . . . .	4
1.4	Pygraph . . . . .	4
<b>2</b>	<b>Calcolo simbolico</b>	<b>5</b>
2.1	Maxima . . . . .	6
2.2	Maple . . . . .	7
2.3	MATLAB . . . . .	7
2.4	Geogebra . . . . .	8
2.5	Sympy . . . . .	8
<b>3</b>	<b>Sistemi lineari</b>	<b>8</b>
3.1	Foglio di calcolo . . . . .	9
3.2	Maxima . . . . .	9
3.3	MATLAB . . . . .	9
3.4	Numpy . . . . .	10
<b>4</b>	<b>Rappresentazione grafica di funzioni</b>	<b>10</b>
4.1	Gnuplot . . . . .	10
4.2	MAFA Plotter . . . . .	11
4.3	Google . . . . .	11

# 1 Geometria dinamica

Un software di geometria dinamica, o, come altrimenti si dice, di geometria interattiva, si basa sull'interazione tra la costruzione di una figura geometrica sullo schermo del computer attraverso il governo dell'accensione o dello spegnimento di pixel di vario colore e le relazioni algebriche esistenti tra gli elementi che caratterizzano la figura, in modo che la modifica della posizione di uno di questi si rifletta immediatamente sulla figura stessa. Il tutto utilizzando la tastiera del computer e lo strumento di puntamento sullo schermo (touchpad o mouse).

Se, per esempio, individuiamo con il mouse due punti sullo schermo, il software ne memorizza la posizione attraverso le relative coordinate e calcola la distanza tra di essi e, se vogliamo, la scrive.

Se poi utilizziamo la funzione che ci offre il software per disegnare un cerchio che abbia il centro in uno dei due punti e la circonferenza che passi per l'altro, il software traccia la circonferenza e immediatamente definisce la distanza tra i due punti come raggio di un cerchio e determina area del cerchio e lunghezza della circonferenza e, se vogliamo, scrive queste grandezze.

Infine se spostiamo con il mouse il punto sulla circonferenza la circonferenza stessa si ridisegna e vengono ricalcolati raggio, area e lunghezza della circonferenza del nuovo cerchio.

Tutto ciò apre la strada a interpretazioni algebriche delle figure, a traslazioni e trasformazioni geometriche e ci porta ben oltre la geometria osservabile con riga, squadra e compasso: la possibilità di manipolare liberamente le figure conservandone la natura (i cerchi rimangono cerchi, le parallele rimangono parallele, i triangoli rimangono triangoli) dà modo agli studenti di scoprire, in modo divertente e interattivo, le proprietà geometriche delle figure stesse e gli insegnanti possono presentare, in modo particolarmente efficace, le loro lezioni.

In ordine di preferenza mia personale i migliori software di questa categoria sono i seguenti.

## 1.1 Geogebra

Con il suo nome che combina Geometria e Algebra fa un po' di tutto.

Si trova all'indirizzo <https://www.geogebra.org/?lang=it>.

Come è fatto bene il software è fatto bene il sito su cui lo troviamo.

Abbiamo infatti qui un'ampia documentazione in lingua italiana e la possibilità di provare on-line le varie applicazioni che compongono il software in modo da poter scegliere quale scaricare.

Se scarichiamo la versione GeoGebra Classico le avremo tutte in uno, comprese molte cose che esulano dal campo della geometria interattiva ma che sono utili per altre esigenze.

Geogebra è uno dei migliori esempi di software libero: iniziato da Markus Hohenwarter nel 2001 presso l'Università di Salisburgo, vi hanno poi collaborato schiere di sviluppatori open-source e traduttori sparsi in tutto il mondo.

Sono disponibili versioni del software per i sistemi operativi Linux, Windows e Mac OS X per quanto riguarda i computer, oltre che per i sistemi Android e iOS per quanto riguarda smartphone e tablet.

Qualche eccezione alla libertà del software si trova per il suo utilizzo commerciale sui sistemi Windows e Mac OS X, ma si tratta di cose che non interessano docenti, discenti e amatori della matematica, che possono liberamente utilizzarlo e senza alcuna spesa.

Vista la dovizia di informazioni e di dimostrazioni che si possono trovare all'indirizzo che ho prima indicato, mi fermo qui. Sottolineo soltanto, e lo si può capire visitando il sito, che si tratta di un software concepito e nato proprio per essere di ausilio all'insegnamento, inizialmente della geometria e poi anche di molto altro.

La modularità, la potenza e la versatilità di Geogebra, che funziona su qualsiasi apparecchiatura fissa, portatile e mobile, hanno ormai oscurato i numerosi altri software esistenti per la geometria interattiva.

Per rispetto alla loro anzianità ed alla loro tuttora apprezzabile qualità devo citarne un altro paio.

## 1.2 Cabri

Si trova all'indirizzo <https://cabri.com/>, dove possiamo fruire di presentazioni per studenti, insegnanti e imprese in lingua inglese, francese o spagnola.

Si tratta di un software proprietario, nato all'Università di Grenoble nel lontano 1988 ed è il decano dei software di geometria interattiva.

Viene offerto in varie edizioni, per la scuola primaria (poco più che un giocattolo), per la scuola secondaria di 1° grado (geometria piana, incluse le trasformazioni geometriche) e per la scuola secondaria di 2° grado (geometria sintetica, geometria analitica, rappresentazione di funzioni e di luoghi geometrici).

Il tutto a pagamento, con l'eccezione di una versione, Cabri Express, offerta gratuitamente ma di buon contenuto (geometria euclidea, rappresentazione di funzioni).

Tutto il software Cabri funziona solo su Windows e Mac OS X.

## 1.3 Cinderella

Si trova all'indirizzo <https://www.cinderella.de/>.

Software proprietario che attualmente è in manutenzione presso l'Università di Monaco di Baviera. Dopo poco tempo dalla sua prima stesura è stato riscritto in linguaggio Java nel 1996 da Jürgen Richter-Gebert e Ulrich Kortenkamp ed attualmente ne è distribuita la versione Cinderella.2, utilizzabile su qualsiasi sistema operativo su cui sia presente la macchina virtuale Java (JRE).

Dalla pagina del download sul sito possiamo scaricare gli installer per Windows e Mac OS X, che contengono la macchina virtuale Java. Per Linux si scarica un eseguibile che richiede la presenza della macchina virtuale Java, eventualmente da installare a parte. A differenza di quanto avveniva fino a qualche tempo fa il software è gratuito.

Come si può vedere dalla documentazione fornita in lingua inglese nella relativa pagina sul sito, questo software è solo in parte dedicato alla geometria interattiva ed in questo è meno ricco dei due visti prima.

Molto interessante la parte dedicata a simulazioni di Fisica e altrettanto interessante la parte dedicata allo scripting: utilizzando un semplice linguaggio è possibile creare disegni, figure, animazioni e perfino suoni.

## 1.4 Pygraph

Chi conosce il linguaggio di programmazione Python può divertirsi a creare personalmente programmi di geometria interattiva utilizzando un modulo grafico, che si chiama Pygraph.

Non si tratta del modulo Pygraph della biblioteca ufficiale di Python, che si occupa di grafi. Questo Pygraph è un prodotto che si occupa di grafica, ha finalità didattiche e ce lo regala l'ottimo insegnante Daniele Zambelli.

Il manuale, scritto in italiano, è scaricabile all'indirizzo <https://media.readthedocs.org/pdf/pygraph/latest/pygraph.pdf>.

Esso contiene tutte le istruzioni per lavorare con Pygraph ma anche passi la cui lettura è molto istruttiva per chi voglia apprendere o consolidare la tecnica di programmazione a oggetti con Python.

Il luogo da cui possiamo scaricare il software è

<https://bitbucket.org/zambu/pygraph>.

Aperta la scheda del download (ultimo dei link elencati sulla destra della home page) scarichiamo il file `pygraph33_RC01.zip`, che contiene la versione 3.3 del 18 gennaio 2018, nella nostra directory dei file scaricati.

Estratto l'archivio, ciò che maggiormente interessa sono il file `pygraph.pth` e la directory `pygraph` col relativo contenuto: questo è il vero e proprio pacchetto. Trasferiamo il file e la directory nel posto giusto, cioè:

. se usiamo Linux in `/usr/local/lib/pythonX.X/dist-packages` ,

. se usiamo Windows in `C:\PythonXX\Lib\site-packages` ,

. se usiamo Mac in `/Library/Frameworks/Python.framework/Versions/X.X/lib/pythonX.X/site-packages/` ,

dove X.X sta per i numeri che identificano la versione Python installata sul computer (esempio 3.5 o 3.7, ecc.).

Tutto il resto è documentazione e lo possiamo mettere dove più ci fa comodo.

Il modulo contiene la libreria `Pyig`, che è quella che ci dà modo di sviluppare geometria interattiva.

Per esemplificazioni sull'argomento rimando all'allegato «grafica\_interattiva\_python» al mio articolo «Ancora grafica con Python» dell'ottobre 2018 archiviato nella categoria Programmazione del mio blog all'indirizzo [www.vittal.it](http://www.vittal.it).

## 2 Calcolo simbolico

L'utilità del calcolo simbolico sta nella sua capacità di astrazione e di generalizzazione del calcolo numerico attraverso l'introduzione di simboli che sottendono un valore numerico qualsiasi in notazioni che chiamiamo espressioni matematiche.

Gli stessi simboli possono essere oggetto di calcolo fornendo sempre un risultato espresso in simboli in nuove espressioni matematiche.

Solo sostituendo ai simboli dei valori numerici si addiuviene ad un risultato numerico.

Tutto ciò richiama innanzi tutto il meccanismo delle formule matematiche. Il modo più sintetico di definire il doppio di un numero è  $2x$ . Per ottenere il doppio di 3, nell'espressione  $2x$  sostituiamo 3 a  $x$  e otteniamo 6.

Complicando un po' le cose: il modo più semplice per calcolare il valore della derivata prima della funzione  $f(x) = x^2$  per  $x = 0,5$  è quello di calcolare in simboli che la derivata prima della funzione è  $2x$  e poi sostituire 0,5 a  $x$  in quest'ultima espressione per arrivare al risultato numerico 1. Ma la formula  $2x$  può utilmente servire per calcolare il valore della derivata della funzione  $x^2$  in qualsiasi punto, compresa la possibilità, attraverso la soluzione dell'equazione  $2x = 0$ , di trovare il punto in cui la derivata si azzerava, ad indicare la possibile esistenza, in quel punto, di un massimo o di un minimo relativo della funzione  $x^2$ .

Come si vede, riconosciuto al calcolo numerico tutto il merito di quantificare soluzioni a problemi concreti nei più disparati campi, non si può negare che senza il calcolo

simbolico non esisterebbe l'analisi matematica e il linguaggio scientifico sarebbe veramente a corto di possibilità espressive. Peraltro lo stesso calcolo numerico molto spesso fornisce risultati utili in quanto alle sue spalle esiste un tracciato costruito in simboli che prepara la strada per raggiungere quei risultati.

I primi linguaggi di programmazione per computer ebbero l'obiettivo di automatizzare il calcolo numerico evitando il lavoro di carta, penna e prontuari vari per quando le cose si complicavano (tavole logaritmiche, tavole delle funzioni trigonometriche, ecc.) per risolvere problemi che potessero essere descritti in termini matematici e risolti mediante un'approssimazione numerica. Dopo pionieristiche applicazioni dei primi anni cinquanta del secolo scorso, molte dovute a Grace Hopper, nel 1955 John Backus ci ha regalato il linguaggio Fortran (FORmula TRANslator), ormai vecchio e superato ma ancora utilizzato da qualcuno.

I linguaggi di programmazione orientati al trattamento non soltanto di numeri ma anche di simboli sono nati appena dopo e sono associati ai primi progetti di intelligenza artificiale. Nel 1958 John McCarthy, appunto l'inventore del termine «intelligenza artificiale», ideò il linguaggio LISP (LISt Processor) il cui primo obiettivo fu quello di supportare lo studio della computabilità di funzioni ricorsive su espressioni simboliche. Questo linguaggio è tuttora utilizzato (basti pensare a software come AutoCAD e simili) ed è stato ampiamente impiegato per programmare software dedicato al calcolo simbolico.

La differenza tra i due approcci, del calcolo numerico e del calcolo simbolico, sta nel fatto che con il calcolo numerico arriviamo sempre ad un risultato numerico, magari approssimato, ma dobbiamo sempre partire da numeri mentre con il calcolo simbolico ciò che non è calcolabile rimane un simbolo (e possiamo forzarne il calcolo approssimato solo se lo vogliamo) e, se il calcolo implica anche simboli veri e propri (come  $a$ ,  $x$ ,  $\beta$ ,  $p$ ), essi rimangono tali e quali.

Se, per esempio, calcoliamo la radice quadrata di 8 con un programma di calcolo numerico otteniamo il risultato approssimato 2.8284271247461903 ma se la calcoliamo con un programma di calcolo simbolico il risultato finale che otteniamo è  $2\sqrt{2}$  oppure  $2^{\frac{3}{2}}$ : in ogni caso il numero irrazionale  $\sqrt{2}$  rimane in simbolo.

Altro esempio: siamo di fronte all'espressione matematica  $2x - 3y + 5$ . Con un software di calcolo numerico possiamo semplicemente calcolare il valore dell'espressione trattando la  $x$  e la  $y$  come variabili e, ogni volta che facciamo il calcolo, dobbiamo deciderne i valori. Così sapremo che, se  $x = 1$  e  $x = 2$ , il valore dell'espressione sarà 1, se  $x = 2$  e  $y = 0$ , il valore dell'espressione sarà 9, ecc. Con un software di calcolo simbolico possiamo invece entrare nell'essenza dell'espressione matematica, esplorando il legami che in essa si concretizzano tra la  $x$  e la  $y$ :  $x = \frac{3y-5}{2}$  e  $y = \frac{5+2x}{3}$  e da qui sviluppare tutta una serie di discorsi su come varia la  $x$  in funzione della  $y$  o viceversa, ecc. ecc.

Anche per il software di calcolo simbolico propongo un piccolo elenco, basato sulle mie conoscenze e ordinato secondo le mie preferenze personali.

## 2.1 Maxima

Dal 1998 è un progetto di software libero open source innestato su un vecchio progetto, che si chiamava Macsyma (MAC symbolic manipulation, dove MAC non c'entra con la Apple ma era un laboratorio del Massachusetts Institute of Technology, MIT) nato negli anni sessanta del secolo scorso.

Interamente sviluppato in linguaggio LISP, lo troviamo, insieme a ricca documentazione anche in buona lingua italiana, all'indirizzo <http://maxima.sourceforge.net/>, con

tutti link per procurarcelo: sono disponibili versioni per Linux, Windows, Mac OS X per quanto riguarda i computer e Android per quanto riguarda smartphone e tablet.

Nella sua versione di base è un software a riga di comando e, per utilizzarlo, occorre conoscere il linguaggio con cui impartire i vari comandi: in proposito il software è supportato da un esauriente help in lingua inglese.

Esiste tuttavia la versione wxMaxima, che troviamo all'indirizzo <https://wxmaxima-developers.github.io/wxmaxima/>

per computer con sistema operativo Linux, Windows e Mac OS X, ma non per Android, che ci si presenta con una ricca interfaccia grafica utente e chiarissimo menu che ci guida a fare le varie cose anche in lingua italiana. Sempre con la possibilità di accedere ad un bellissimo manuale, purtroppo disponibile solo in lingua inglese.

Gli installer per Windows e Mac OS X includono anche il software Maxima, mentre su Linux occorrono installazioni separate.

Questo software fa di tutto, calcolo numerico, calcolo simbolico, tracciamento di grafici di funzione.

Per quanto riguarda il calcolo simbolico vi troviamo la possibilità di risolvere equazioni di ogni tipo attraverso la ricerca di radici di polinomi, tutto ciò che riguarda il calcolo differenziale, la semplificazione e la fattorizzazione di polinomi e quant'altro si possa immaginare.

Nella buona consuetudine del software libero l'uso del software non ha limiti ed è completamente gratuito.

## 2.2 Maple

Deriva da un progetto nato negli anni ottanta del secolo scorso con l'obiettivo di creare un sistema algebrico simbolico accessibile ai ricercatori e agli studenti, alternativo al progetto Macsymba, cui si è accennato nel precedente paragrafo, e che impegnasse meno l'hardware. Penso che il nome derivi dal fatto che il progetto fu lanciato presso l'Università di Waterloo in Ontario, Canada, la cui bandiera porta la foglia di acero (maple leaf).

Nel tempo è diventato un ambiente completo di elaborazione matematica dotato di un proprio linguaggio di programmazione. Funziona a riga di comando ed occorre conoscere il linguaggio per i comandi.

Per il calcolo simbolico non ha nulla in più di ciò che troviamo negli altri software.

Il suo sito si trova all'indirizzo <https://www.maplesoft.com/products/maple/> e il software, a parte versioni dimostrative funzionanti gratuitamente per una quindicina di giorni, è parecchio costoso, anche se per gli studenti sono previsti sconti.

Funziona su Linux, Windows e Mac OS X.

Ne esiste una versione gratuita, con contenuto di tutto rispetto, per i sistemi mobile iOS e Android, scaricabile rispettivamente da App Store e da Google Play.

## 2.3 MATLAB

Per capire cosa è diventato MATLAB andiamo sul sito <https://www.mathworks.com/>.

Nato come software di calcolo numerico (MATrix LABoratory) alla fine degli anni settanta del secolo scorso si è arricchito nel tempo fino a contenere tools per il machine learning e per la data science. Per un certo periodo fornì supporto anche per il calcolo simbolico grazie ad una licenza acquisita da Maplesoft per l'utilizzo del motore Maple di cui al paragrafo precedente. Dopo che la Mathworks ha assorbito la softwarehouse che distribuiva MuPAD, quest'altro glorioso software di calcolo simbolico è entrato a



dove  $A$  è la matrice dei coefficienti,  $x$  è il vettore delle incognite e  $b$  è il vettore dei termini noti.

Il sistema si risolve ponendo

$$A^{-1}b = x$$

cioè, moltiplicando la matrice inversa della matrice dei coefficienti per il vettore dei termini noti, otteniamo il vettore delle incognite espresse numericamente, come soluzione del sistema di equazioni.

Ciò ammesso che esista la matrice inversa, cioè che esista una matrice tale per cui il prodotto tra la matrice e la sua inversa sia uguale a 1:

$$AA^{-1} = A^{-1}A = 1$$

In questo caso, infatti, se moltiplichiamo entrambi i membri della nostra equazione sintetica per  $A^{-1}$  otteniamo  $A^{-1}Ax = A^{-1}b$ , cioè  $x = A^{-1}b$  che è la formula risolutiva.

Capito il meccanismo rimangono da fare i conti e il conto più laborioso è quello della matrice inversa.

Con l'algoritmo di Cramer evitiamo questo calcolo ma ci troviamo a dover calcolare i determinanti. Se le equazioni sono due facciamo presto, se sono tre facciamo altrettanto presto, magari aiutati dalla regola di Sarrus. Se sono di più può diventare un bel problema.

Anche con l'algoritmo della riduzione di Gauss non andiamo lontano e dobbiamo lavorare parecchio.

Allora ci vuole proprio il computer, che ci mette a disposizione tutta una serie di strumenti per risolvere il problema.

### 3.1 Foglio di calcolo

Utilizzando le formule di matrice, con Excel di Microsoft Office o Calc di Libre Office possiamo agevolmente trovare la soluzione di sistemi lineari secondo la relazione  $A^{-1}b = x$ .

Basta scrivere in una zona di celle la matrice dei coefficienti, invertirla applicando la relativa formula preconfezionata, e moltiplicare l'inversa per il vettore dei termini noti, scritto in verticale in una zona di celle, sempre utilizzando la funzione preconfezionata.

### 3.2 Maxima

Con i comandi adeguati si fa la stessa cosa.

Se usiamo wxMaxima troviamo nel menu EQUAZIONI ▷ RISOLVI SISTEMA LINEARE... la via per risolvere il problema semplicemente inserendo, una via l'altra, le equazioni che compongono il sistema e indicando le incognite, senza nemmeno scrivere matrici e quant'altro.

### 3.3 MATLAB

Come ho detto nel precedente capitolo, MATLAB è nato per il calcolo matriciale, per cui, con i comandi appropriati possiamo sviluppare la nostra relazione  $A^{-1}b = x$ .

### 3.4 Numpy

Chi conosce il linguaggio Python può divertirsi con il modulo Numpy, concepito per il calcolo numerico e adatto al calcolo matriciale.

## 4 Rappresentazione grafica di funzioni

Di quanto visto finora, quello di tracciare il grafico di una funzione è il compito più facile che possa capitare ad un computer.

Data una relazione funzionale che associa il valore di una variabile a quello di un'altra, il computer compone idealmente la tabella descrittiva di una serie di queste associazioni di valori e se ne serve da coordinate per accendere un pixel sul proprio schermo: pixel dopo pixel si viene a creare una linea che rappresenta il grafico della relazione funzionale. Più sono tra loro ravvicinati gli intervalli di determinazione delle associazioni più è continua la linea.

Con i moderni computer le velocità di calcolo consentono di calcolare in pochissimo tempo tantissime e fittissime associazioni di valore e le definizioni grafiche sempre più accurate arrivano a tracciare grafici di funzione perfetti.

Tutti i software visti nei capitoli precedenti (Geogebra, Cabri, Cinderella, Maxima, Maple e MATLAB) hanno la capacità di produrre grafici di funzione, anche a tre dimensioni.

Con il linguaggio Python, per grafici a due dimensioni, possiamo utilizzare la libreria `Pyplot` del modulo `Pygraph` visto nel paragrafo 1.4.

Esistono anche software specifici tra i quali, sempre con il criterio della mia preferenza personale, cito i seguenti.

### 4.1 Gnuplot

Probabilmente è il più anziano dei software dedicati al tracciamento grafico ma è tuttora vivo e vegeto: l'ultimo rilascio è del dicembre 2019.

È software libero open source, ma le lettere Gnu che appartengono al suo nome non hanno nulla a che vedere con il progetto di software libero GNU: sono semplicemente un modo per riscrivere il nome Newplot che gli si voleva assegnare per evitare omonimia con un altro progetto.

Lo troviamo all'indirizzo [www.gnuplot.info](http://www.gnuplot.info), dove possiamo accedere a documentazione e tutorial.

Nella pagina del download troviamo i link per poter scaricare gli eseguibili per i sistemi Linux, Windows e Mac OS X.

Su Google Play troviamo una versione per Android.

Gnuplot è un software a riga di comando.

Con questo software possiamo rappresentare in grafici molto elaborati qualsiasi serie di dati ed occorre conoscere il linguaggio per impartire le varie istruzioni.

Per quanto riguarda il tracciamento grafico di funzioni matematiche i comandi sono molto semplici:

`plot f(x)` traccia il grafico della funzione  $y = f(x)$

`splot f(x,y)` traccia il grafico tridimensionale della funzione  $z = f(x,y)$

Prima di indicare la funzione possiamo indicare i range entro cui tracciare i grafici indicando, tra parentesi quadre e divisi dal simbolo `:`, i limiti inferiore e superiore delle variabili, nell'ordine,  $x$ ,  $y$  e, nel caso,  $z$ .

Ad esempio:

```
plot [-5:5] [-10:10] sin(x)
```

disegna il grafico della funzione  $y = \sin(x)$  per  $x$  tra -5 e 5 e  $y$  tra -10 e 10;

```
splot [-10:10] [-20:20] [-30:30] sin(x)/y
```

disegna il grafico della funzione  $z = \frac{\sin(x)}{y}$  per  $x$  tra -10 e 10,  $y$  tra -20 e 20 e  $z$  tra -30 e 30.

## 4.2 MAFA Plotter

Se andiamo all'indirizzo [www.mathe-fa.de](http://www.mathe-fa.de) possiamo fruire on-line di un simpatico programma per tracciare grafici di funzione, completamente guidato da interfaccia grafica.

## 4.3 Google

Forse non tutti sanno che se scriviamo una funzione matematica nella barra di ricerca di Google, miracolosamente ce ne viene mostrato il grafico.