

Visualizzazione grafica

dell'output di un programma con GNUplot

GNUplot è un programma interattivo per la visualizzazione di grafici di funzione; non è una "libreria" da includere in un programma C, ma un programma a se stante, che può essere richiamato in modalità *interattiva* o in modalità *batch*; in questa seconda modalità è possibile richiamare in modo trasparente il programma GNUplot da un programma scritto in C, utilizzando la funzione `system()`, senza richiedere l'intervento dell'utente per la visualizzazione del grafico di una funzione o di un set di coordinate cartesiane.

In questo modo GNUplot è uno strumento molto comodo per visualizzare un grafico con l'output dell'elaborazione numerica effettuata da un programma scritto in linguaggio C. Se sulla macchina è installato GNUplot, la seguente istruzione inserita in un programma C

```
...  
system("start gnuplot-dir\\wgnuplot.exe");  
  
(sotto linux) system("gnuplot.exe");  
...
```

viene lanciato il programma GNUplot.

Per prima cosa, dunque, è necessario installare sulla macchina una versione di GNUplot compatibile con il sistema operativo utilizzato.

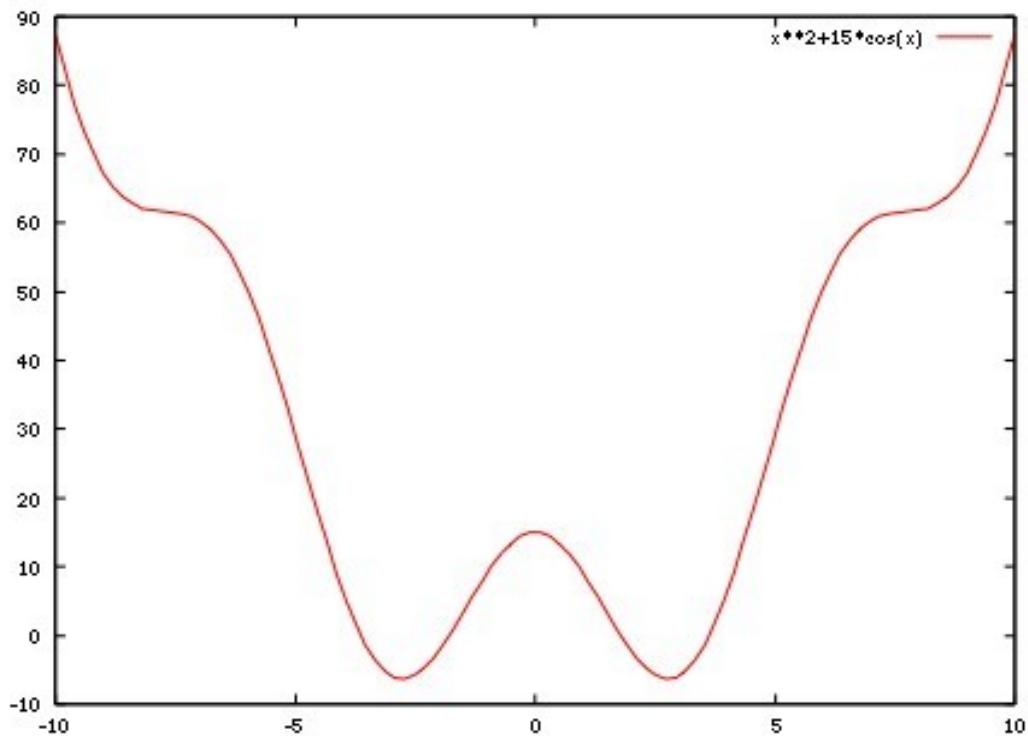
Una volta scaricato dal sito dipartimentale il file compresso che contiene GNU PLOT per l'ambiente windows (<http://www.mat.uniroma1.it/> => servizi => laboratorio e centro di calcolo => Anonymous FTP (dalla colonna di sinistra) => SOFTWARE => WIN => gnuplot => gnuplot-420win32.zip), basta aprire il file wgnuplot contenuto nella directory bin della cartella opportunamente decompressa.

In alternativa: <http://www.gnuplot.info/>.

In modalità interattiva sostanzialmente GNUplot richiede che l'utente digiti dei comandi che producono un output grafico in una finestra. GNUplot è un programma molto potente e sofisticato, ma in estrema sintesi possiamo dire che il comando principale per la visualizzazione di grafici è `plot` (`splot` per le superfici), seguito da un'espressione matematica nella variabile `x`, oppure dal nome di un file contenente un le coordinate dei punti da visualizzare nel piano cartesiano. Ad esempio:

```
gnuplot> plot sin(x)

gnuplot> plot "dati.txt"
gnuplot> plot "dati.txt" with lines
gnuplot> plot x**2+15*cos(x)
```



Il comando `help` visualizza una guida interattiva di aiuto sull'uso dei comandi di GNUplot.

Se i comandi che vogliamo far eseguire a GNUplot sono riportati in un file di testo, uno sotto all'altro, ciascuno su una riga del file, è possibile lanciare GNUplot in modalità *batch* passandogli il nome del file sulla riga di comando; in questo modo, invece di operare in modalità interattiva, richiedendo all'utente di digitare i comandi, il programma li legge dal file e li esegue senza la necessità di intervento da parte dell'utente. Ad esempio il seguente comando lancia il programma

GNUplot e gli fa visualizzare i grafici utilizzando i comandi contenuti nel file "comandi.txt":

```
$ gnuplot comandi.txt
```

Spesso può tornare utile produrre una visualizzazione grafica di un set di coordinate prodotte come output da un programma di elaborazione numerica, ottenendo, ad esempio, il grafico di una funzione per punti. Per far questo il nostro programma C deve salvare i dati (coppie di punti, coordinata x e coordinata y) su un file, una coppia di numeri separati tra loro da uno spazio su ogni riga del file; ad esempio, utilizzando un editor di testo, possiamo salvare sul file "parabola.txt" le seguenti righe:

```
0.0 0.0
0.5 0.25
1.0 1.0
1.5 2.25
2.0 4.0
2.5 6.25
```

È possibile quindi visualizzare il grafico per punti con il seguente comando di GNUplot:

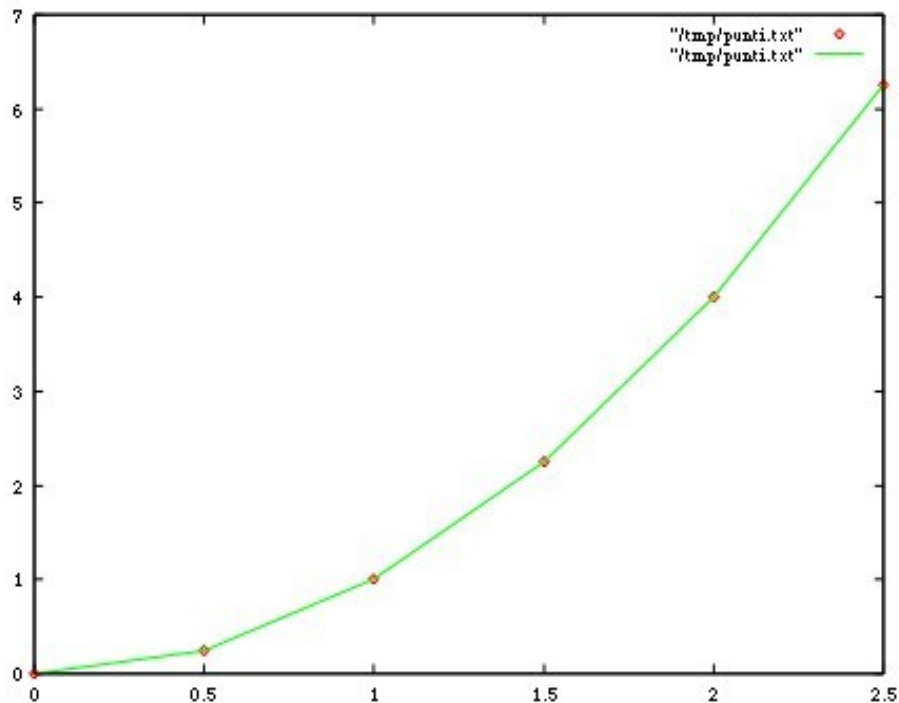
```
gnuplot> plot "parabola.txt"
```

oppure, unendo i punti con dei segmenti, con il seguente comando:

```
gnuplot> plot "parabola.txt" with lines
```

oppure ancora, tracciando il grafico sia con dei punti che con delle linee:

```
gnuplot> plot "parabola.txt","parabola.txt" with lines
```



Quindi il nostro programma di calcolo numerico, dovrà aprire un file in scrittura con la funzione `fopen`, scrivere le coppie di numeri nel file con la funzione `fprintf`, chiudere il file con la funzione `fclose` e alla fine lanciare `GNUplot` con la funzione `system` per visualizzare i punti contenuti nel file.

Di seguito riporto un breve programmino con un esempio per il calcolo di una funzione per punti, la registrazione delle coordinate su un file, e l'esecuzione di `GNUplot` per tracciare sul piano cartesiano una spezzata che collega i punti in cui è stata calcolata la funzione. Oltre al file "dati.txt" con le coordinate dei punti della funzione, calcolate nell'intervallo $[x_0, x_1]$ in $x_0, x_0 + \Delta x, x_0 + 2\Delta x, \dots, x_0 + n\Delta x = x_1$ (dove $\Delta x = (x_1 - x_0)/n$), viene anche creato il file "comando.txt" contenente solo una riga con il comando di `GNUplot` per la visualizzazione del grafico.

```

/*

** questo programma si chiama grafico.c
**
** Calcola per punti la funzione  $y=f(x)$ , salva sul file
** "dati.txt" le coordinate dei punti e visualizza il
** grafico utilizzando GNUplot.
**
** LPC- 2009-10
*/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

/*
 * Si definisce una funzione  $f(x)$  qualsiasi
 */

float f(float x) {
    return(sin(x)*15 + x*x);
}

/*
 * Funzione principale.
 */

int main(void) {
    FILE *file;
    float x, x0, x1, deltax, y;
    int n;

    printf("Inserisci gli estremi dell'intervallo: ");
    scanf("%f %f", &x0, &x1);
    printf("Inserisci il numero di passi: ");
    scanf("%d", &n);
    deltax = (x1-x0)/n;
    x = x0;
    /*
     * apro il file "dati.txt" in scrittura per registrare le
     * coordinate dei punti della funzione
     */
    file = fopen("dati.txt", "wt");

    while (x <= x1) {
        /*
         * calcolo il valore della funzione nel punto x
         */
        y = f(x);

        /*
         * scrivo sul file le coordinate del punto (x,y)
         */
    }
}

```

```

    fprintf(file, "%f %f\n", x, y);

    /*
     * incremento la variabile x
     */
    x = x + deltax;

}
/*
 * chiudo il file con le coordinate dei punti da
 * visualizzare
 */
fclose(file);

/*
 * apro in scrittura il file "comando.txt" per registrarci
 * il comando che dovra` essere eseguito da GNUplot
 */
file = fopen("comando.txt", "wt");

/*
 * scrivo sul file il comando da eseguire
 */
fprintf(file, "plot \"dati.txt\" with lines\n");

/*
 * chiudo il file su cui ho scritto il comando da eseguire
 */
fclose(file);

/*
 * eseguo il programma GNUplot passandogli il nome del
 * file che contiene il comando da eseguire
 */
system("start C:\\programmi\\gnuplot\\wgnuplot.exe comando.txt");

return(0);
}

```