

Libreria grafica libplot

Uso

- Serve a visualizzare interattivamente i risultati dei programmi;
- per compilare usare
`gcc -o programma programma.c -lplot;`
- includere sempre `<plot.h>`;

Struttura del programma

- definire almeno una variabile puntatore a `pIPlotter` e una di tipo puntatore a `pIPlotterParams`;
- chiamare nel giusto ordine le funzioni per inizializzare ;
- aggiungere sempre il nome del plot nelle chiamate alle funzioni grafiche ;
- chiamare le funzioni per terminare;
- per la documentazione completa: `info plotutils`

Funzioni per inizializzare e terminare

- $PI_par = pl_newplparams ()$;
- $PI = pl_newpl_r ("X", stdin, stdout, stderr, PI_par)$
"X" può essere sostituito da "ps";
- $pl_openpl_r (PI)$
per aprire il plot;
- $pl_closepl_r (PI)$
per chiudere il plot;
- $pl_deletepl_r (PI)$
per cancellare il plot;

Funzioni per disegnare

- *pl_erase_r (PI)*
per cancellare il contenuto del plot;
- *pl_fspace_r (PI, xini, yini, xfin, yfin)*
per definire le coordinate utente;
- *pl_fmove_r (PI, x, y)*
per spostare il cursore grafico in x,y;
- *pl_pencolorname_r (PI, "red")*
per stabilire con che colore si disegna;
- *pl_flinewidth_r (PI, 0.25)*
stabilisce la larghezza della linea;
- *pl_fcircle_r (PI,x, y, raggio)*
disegna una circonferenza;
- *pl_bgcolorname_r (PI,"blue")*
stabilisce il colore dello sfondo;
- *pl_fline_r (PI, x1,y1,x2,y2)*
disegna una linea;

Cannibalismo informatico

Per usare un tipo nuovo di programma nono è di solito necessario capirlo profondamente: si può cominciare facendo taglia e incolla sui programmi scritti da altri, come questo preso dalle pagine info

```
#include <stdio.h>
#include <plot.h>

int main ()
{
    plPlotter *plotter;
    plPlotterParams *plotter_params;
    int i = 0, j;

    /* set Plotter parameters */
    plotter_params = pl_newplparams ();
    pl_setplparam (plotter_params,
        "BITMAPSIZE", "300x150");
    pl_setplparam (plotter_params,
        "VANISH_ON_DELETE", "yes");
    pl_setplparam (plotter_params,
        "USE_DOUBLE_BUFFERING", "yes");

    /* create an X Plotter with the specified parameters */
    if ((plotter = pl_newpl_r ("X", stdin, stdout, stderr,
        plotter_params)) == NULL)
    {
        fprintf (stderr, "Couldn't create Plotter\n");
        return 1;
    }

    /* open Plotter */
    if (pl_openpl_r (plotter) < 0)
```

```

{
    fprintf (stderr, "Couldn't open Plotter\n");
    return 1;
}

/* set user coor system */
pl_fspace_r (plotter, -0.5, -0.5, 299.5, 149.5);
/* set line thickness */
pl_linewidth_r (plotter, 8);
/* objects will be filled */
pl_filltype_r (plotter, 1);
/* set background color */
pl_bgcolorname_r (plotter, "saddle brown");
for (j = 0; j < 300; j++)
{
    /* erase window */
    pl_erase_r (plotter);
    /* use red pen */
    pl_pencolorname_r (plotter, "red");
    /* use cyan filling */
    pl_fillcolorname_r (plotter, "cyan");
    /* draw an ellipse */
    pl_ellipse_r (plotter, i, 75, 35, 50, i);
    /* use black pen and filling */
    pl_colorname_r (plotter, "black");
    /* draw a circle [the pupil] */
    pl_circle_r (plotter, i, 75, 12);
    /* shift rightwards */
    i = (i + 2) % 300;
}
/* close Plotter */
if (pl_closepl_r (plotter) < 0)
{
    fprintf (stderr, "Couldn't close Plotter\n");
    return 1;
}

```

Analisi del programma

A una prima occhiata si vede subito che le istruzioni si possono raggruppare in tre tipi: all'inizio (e alla fine) quelle per inizializzare le capacità grafiche del sistema, subito dopo quelle che definiscono lo spazio in cui si lavora (dimensione della finestra, sistema di coordinate) e infine vengono le funzioni che disegnano quello che ci interessa. Possiamo quindi mantenere quasi inalterate quelle appartenenti ai primi due tipi, cambiando a nostro vantaggio quelle del terzo tipo. Prima di far questo si compila ed esegue il programma, e questo ci chiarirà le idee su come procedere per il caso che ci interessa.

Problema del biliardo con due sfere

Nel biliardo mostrato nell'esercizio, due sfere rimbalsano sulle pareti. Come si fa a tener conto dei loro urti? Prima di tutto consideriamo che le sfere si urtano quando la loro distanza è $< 2R$. La componente della velocità tangente alla congiungente $\vec{\Delta} = \vec{x}_2 - \vec{x}_1$ i due centri resta inalterata, mentre le altre due componenti si scambiano.

$$\vec{v}'_1 = \vec{v}_1 + \vec{v}_{2\parallel} - \vec{v}_{1\parallel}$$

$$\vec{v}'_2 = \vec{v}_2 + \vec{v}_{1\parallel} - \vec{v}_{2\parallel}$$

Se

$$\vec{u} = \frac{(\vec{v}_2 - \vec{v}_1) \cdot \vec{\Delta}}{\Delta^2} \vec{\Delta}$$

abbiamo

$$\vec{v}'_1 = \vec{v}_1 + \vec{u} \quad \vec{v}'_2 = \vec{v}_2 - \vec{u}$$

Attenzione alle condizioni iniziali!

