
Lezione 2

Argomenti della lezione:

Grafici

Creare un grafico (Classe TGraph)

Sovrapporre grafici

Grafici con barre di errore (Classe TGraphErrors)

Grafici con barre di errore asimmetriche (Classe TGraphAsymmErrors)

Grafici multipli (Classe TMultiGraph)

Disegnare grafici con assi personalizzati

Grafici

Creare un grafico (Classe TGraph)

- La classe TGraph consente di disegnare grafici per punti. Occorre dichiarare due vettori di n elementi di tipo float o double.

```

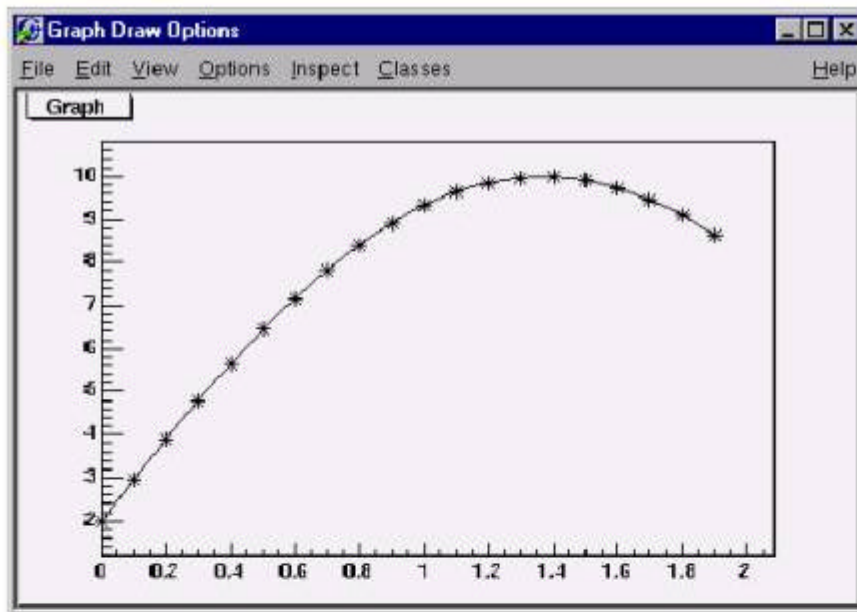
Int_t n = 20;
Double_t x[n], y[n];

for (Int_t i=0;i<n;i++) {
    x[i] = i*0.1;
    y[i] = 10*sin(x[i]+0.2);
}

TGraph * gr1 = new TGraph (n, x, y);
```

- Per disegnare si usa il metodo Draw(option), dove alcune opzioni possibili sono:
 - ✗ "L" disegna una linea per unire i vari punti
 - ✗ "A" disegna gli assi
 - ✗ "*" disegna un * per ogni punto
 - ✗ "B" disegna un rettangolo per ogni punto
 - ✗ "P" il marker corrente è disegnato per ogni punto
 - ✗ "C" una curva è disegnata per congiungere i punti

Le opzioni possono essere usate sia come lettere minuscole o maiuscole e possono essere usate anche più di una alla volta.



```

{
  Int_t n = 20;
  Double_t x[n], y[n];

  for (Int_t i=0;i<n;i++) {
    x[i] = i*0.1;
    y[i] = 10*sin(x[i]+0.2);
  }

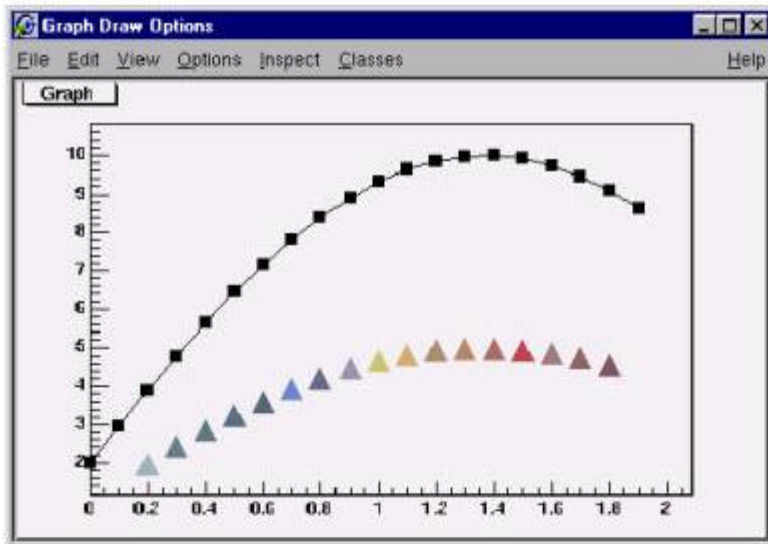
  // create graph
  TGraph *gr = new TGraph(n,x,y);

  TCanvas *c1 = new TCanvas ("c1","Graph Draw Options",
200, 10, 600, 400);

  // draw the graph with axis,contineous line, and
  // put a * at each point
  gr->Draw("AC*");
}

```

- E' possibile cambiare il marker usando la classe TMarker



```

{
  Int_t n = 20;
  Double_t x[n], y[n];

  // build the arrays with the coordinate of points
  for (Int_t i=0;i<n;i++) {
    x[i] = i*0.1;
    y[i] = 10*sin(x[i]+0.2);
  }

  // create graphs
  TGraph *gr3 = new TGraph(n,x,y);

  TCanvas *c1 = new TCanvas ("c1","Graph Draw Options",
200,10, 600, 400);

  // draw the graph with the axis,contineous line, and put
  // a marker using the graph's marker style at each point
  gr3->SetMarkerStyle(21);
  c1->cd(4);
  gr3->Draw("APL");

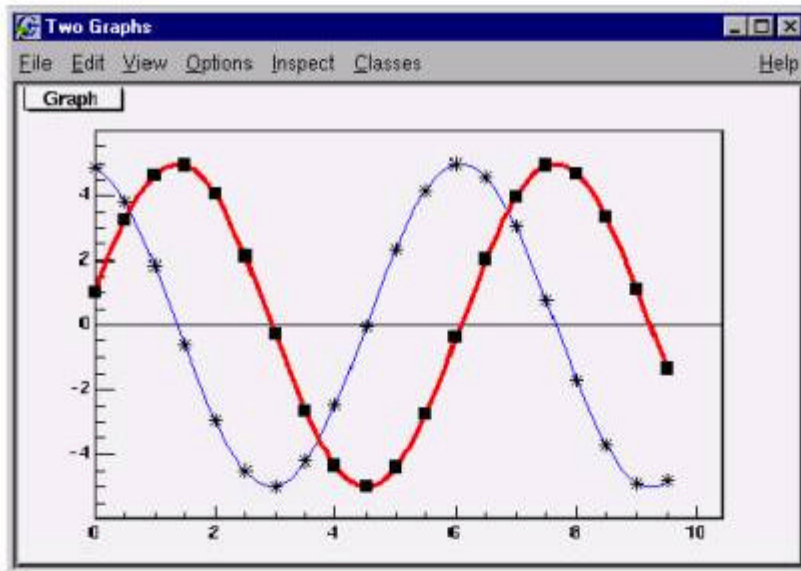
  // get the points in the graph and put them into an array
  Double_t *nx = gr3->GetX();
  Double_t *ny = gr3->GetY();

  // create markers of different colors
  for (Int_t j=2;j<n-1;j++) {
    TMarker *m = new TMarker(nx[j], 0.5*ny[j],22);
    m->SetMarkerSize(2);
    m->SetMarkerColor(31+j);
    m->Draw();
  }
}

```

Sovrapporre grafici

Per sovrapporre due o più grafici basta semplicemente non usare l'opzione "A" nel metodo Draw nei grafici successivi al primo visualizzato.



```

{
  gROOT->Reset();
  Int_t n = 20;
  Double_t x[n], y[n], x1[n], y1[n];

  // create the blue graph with a cos function
  for (Int_t i=0;i<n;i++) {
    x[i] = i*0.5;
    y[i] = 5*cos(x[i]+0.2);
    x1[i] = i*0.5;
    y1[i] = 5*sin(x[i]+0.2);
  }

  TGraph *gr1 = new TGraph(n,x,y);
  TGraph *gr2 = new TGraph(n,x1,y1);

  TCanvas *c1 = new TCanvas ("c1","Two Graphs" , 200,
    10, 600, 400);

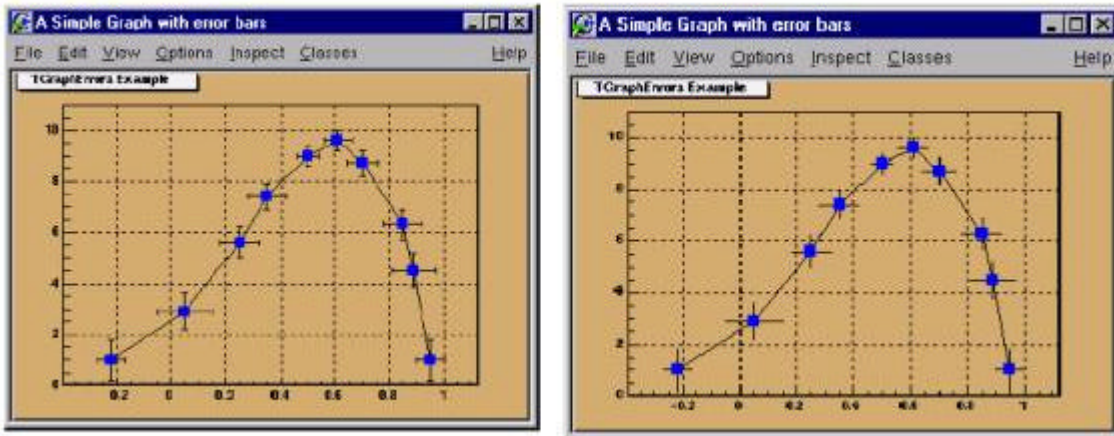
  // draw the graph with axis,contineous line, and
  // put a * at each point
  gr1->SetLineColor(4);
  gr1->Draw("AC*");

  // superimpose the second graph by leaving out
  // the axis option "A"
  gr2->SetLineWidth(3);
  gr2->SetMarkerStyle(21);
  gr2->SetLineColor(2);
  gr2->Draw("CP");
}

```

Grafici con barre di errore (Classe TGraphErrors)

La classe TGraphErrors deriva dalla classe TGraph. La si può pensare come una TGraph con le barre di errore. Valgono le stesse opzioni del metodo Draw della classe TGraph, con l'aggiunta dell'opzione "Z" che permette di non disegnare le linee alla fine delle barre di errore.



Il costruttore della `TGraphErrors` è simile a quello della `TGraph` con l'aggiunta di ulteriori due parametri che rappresentano gli errori X e Y.

```
{
  gROOT->Reset();

  c1 = new TCanvas("c1", "A Simple Graph with error
bars", 200, 10, 700, 500);

  c1->SetFillColor(42);
  c1->SetGrid();
  c1->GetFrame()->SetFillColor(21);
  c1->GetFrame()->SetBorderSize(12);

  // create the coordinate arrays
  Int_t n = 10;
  Float_t x[n] = {-.22, .05, .25, .35, .5, .61, .7, .85, .89, .95};
  Float_t y[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};

  // create the error arrays
  Float_t ex[n] = {.05, .1, .07, .07, .04, .05, .06, .07, .08, .05};
  Float_t ey[n] = {.8, .7, .6, .5, .4, .4, .5, .6, .7, .8};

  // create the TGraphErrors and draw it
  gr = new TGraphErrors(n, x, y, ex, ey);
  gr->SetTitle("TGraphErrors Example");
  gr->SetMarkerColor(4);
  gr->SetMarkerStyle(21);
  gr->Draw("ALP");

  c1->Update();
}
```

Grafici con barre di errore asimmetriche (Classe `TGraphAsymmErrors`)

Una `TGraphAsymmErrors` è una `TGraph` con barre di errore asimmetriche. Le opzioni del metodo `Draw` sono le stesse.

Il costruttore ha 6 vettori come parametri: le posizione dei punti X e Y, i valori corrispondenti agli errori sinistro e destro di X e quelli corrispettivi di Y.

```

{
  gROOT->Reset();
  c1 = new TCanvas ("c1", "A Simple Graph with error bars",
                  200,10,700,500);
  c1->SetFillColor(42);
  c1->SetGrid();
  c1->GetFrame()->SetFillColor(21);
  c1->GetFrame()->SetBorderSize(12);

  // create the arrays for the points
  Int_t n = 10;
  Double_t x[n] = {-.22,.05,.25,.35,.5, .61,.7,.85,.89,.95};
  Double_t y[n] = {1,2.9,5.6,7.4,9,9.6,8.7,6.3,4.5,1};

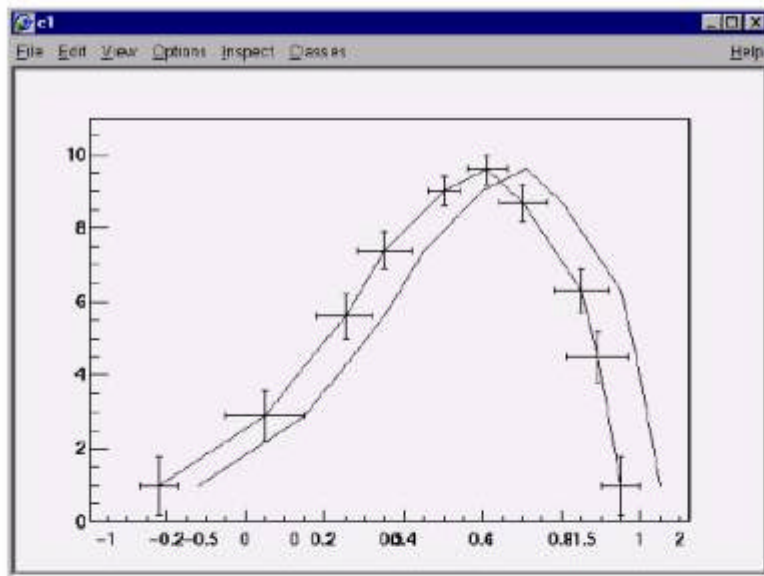
  // create the arrays with high and low errors
  Double_t exl[n] = {.05,.1,.07,.07,.04,.05,.06,.07,.08,.05};
  Double_t eyl[n] = {.8,.7,.6,.5,.4,.4,.5,.6,.7,.8};
  Double_t exh[n] = {.02,.08,.05,.05,.03,.03,.04,.05,.06,.03};
  Double_t eyh[n] = {.6,.5,.4,.3,.2,.2,.3,.4,.5,.6};

  // create TGraphAsymmErrors with the arrays
  gr = new TGraphAsymmErrors(n,x,y,exl,exh,eyl,eyh);
  gr->SetTitle("TGraphAsymmErrors Example");
  gr->SetMarkerColor(4);
  gr->SetMarkerStyle(21);
  gr->Draw("ALP");
}

```

Grafici multipli (Classe TMultiGraph)

Un TMultiGraph è un insieme di oggetti TGraph. Per aggiungere nuovi grafici si usa il metodo Add. Le opzioni del metodo Draw sono le stesse della TGraph.



```

{
  // create the points
  Int_t n = 10;
  Double_t x[n] = {-0.22, .05, .25, .35, .5, .61, .7, .85, .89, .95};
  Double_t y[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};

  Double_t x2[n] = {-0.12, .15, .35, .45, .6, .71, .8, .95, .99, 1.05};
  Double_t y2[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};

  // create the width of errors in x and y direction
  Double_t ex[n] = {.05, .1, .07, .07, .04, .05, .06, .07, .08, .05};
  Double_t ey[n] = {.8, .7, .6, .5, .4, .4, .5, .6, .7, .8};

  // create two graphs
  TGraph *gr1 = new TGraph(n, x2, y2);
  TGraphErrors *gr2 = new TGraphErrors(n, x, y, ex, ey);

  // create a multigraph and draw it
  TMultiGraph *mg = new TMultiGraph();
  mg->Add(gr1);
  mg->Add(gr2);
  mg->Draw("ALP");
}

```

Disegnare grafici con assi personalizzati

Per disegnare un grafico in un certo intervallo X e Y bisogna prima disegnare gli assi in quest'intervallo e successivamente disegnare il grafico senza l'opzione "A". Per disegnare gli assi ci serviamo di un istogramma bidimensionale (classe TH2F).

```
{
gROOT->Reset();
c1 = new TCanvas("c1", "A Zoomed Graph", 200, 10, 700, 500);

// create a histogram for the axis range
hpx = new TH2F
    ("hpx", "Zoomed Graph Example", 10, 0, 0.5, 10, 1.0, 8.0);
// no statistics
hpx->SetStats(kFALSE);
hpx->Draw();

// create a graph
Int_t n = 10;
Double_t x[n] = {-.22, .05, .25, .35, .5, .61, .7, .85, .89, .95};
Double_t y[n] = {1, 2.9, 5.6, 7.4, 9, 9.6, 8.7, 6.3, 4.5, 1};
gr = new TGraph(n, x, y);
gr->SetMarkerColor(4);
gr->SetMarkerStyle(20);
// and draw it without an axis
gr->Draw("LP");
}
```

Alfio Luzzaro, 4/12/2002