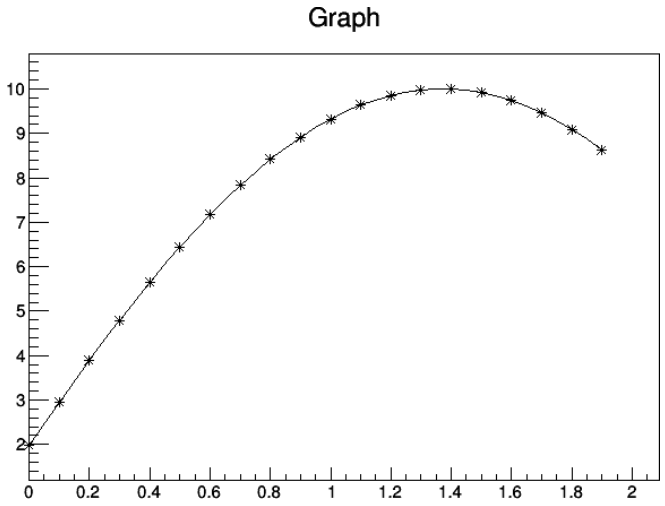
Corso di Laboratorio di Meccanica e Termodinamica
Modulo ROOT
Lezione III

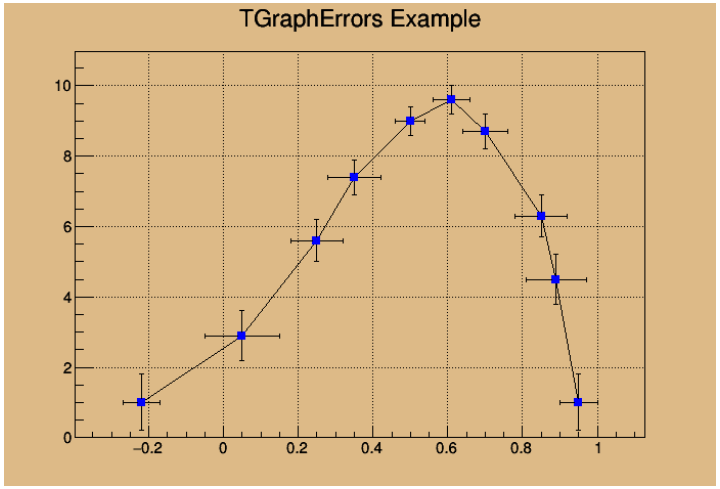
Silvia Arcelli

19 Aprile 2024

Grafici con ROOT

- **Due classi per creare grafici:**
  - **TGraph: rappresenta una serie di N coppie di due quantità X, Y (per esempio due variabili fisiche)**
  - **TGraphErrors (classe derivata di TGraph, consente di includere anche le incertezze EX, EY sulle quantità X,Y)**





Grafici con ROOT

Classe TGraph, due tipi di Costruttori principali:

1) TGraph (Int_t n, const Double_t *x, const Double_t *y)

- **n è il numero di coppie (x,y)**
- **x e y sono gli array di punti x e y**

2) TGraph (const char *filename, const char *format="%lg %lg", Option_t *option="")

- **Il file di input deve contenere almeno due colonne di numeri (separati da blank delimiter)**
- **Il formato della stringa è di default "%lg %lg" (2 double)**
- **Opzioni per skippare colonne ("%lg %*lg %lg") e interpretare delimitatori differenti dal blank**

Grafici con ROOT

Classe TGraphErrors, due tipi di Costruttori principali:

1) TGraphErrors (Int_t n, const Double_t *x, const Double_t *y, const Double_t *ex=0, const Double_t *ey=0)

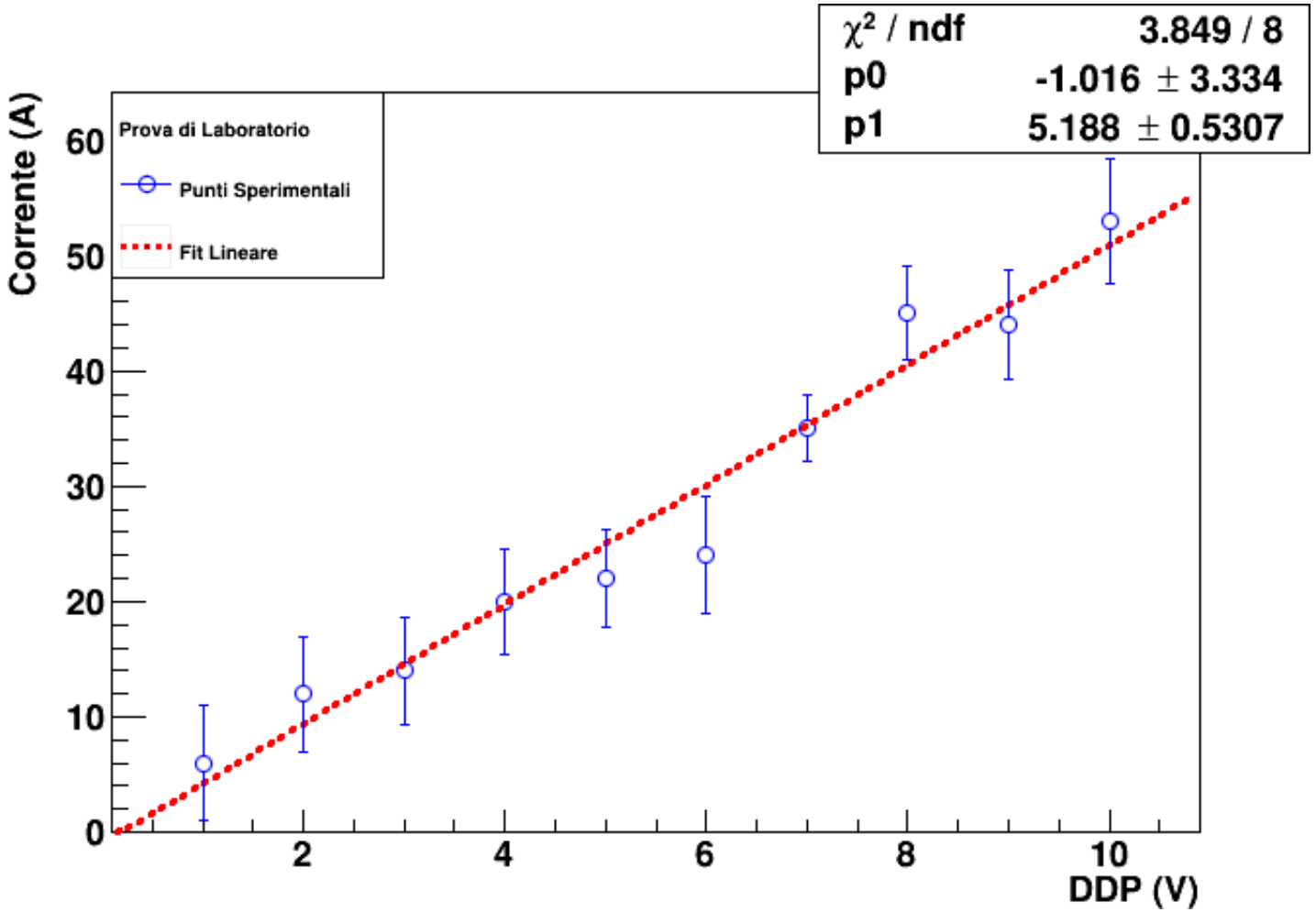
- **n è il numero di punti**
- **x , y ex, ey sono i nomi degli array di punti x e y e i rispettivi errori**

2) TGraphErrors (const char *filename, const char *format="%lg %lg %lg %lg", Option_t *option="")

- **Il file di input deve contenere almeno tre colonne di numeri**
- **Di norma, sarebbero 4. Se ne contiene solo 3 allora la terza colonna è interpretata come errore in y.**

Attenzione! Leggendo da file dei float o double, la virgola non è un carattere accettato, occorre sostituirla con il punto decimale.

Esempio di Grafico con ROOT
(Legge di Ohm)



Esempio di Grafico con ROOT

Macro che fa un grafico e un fit lineare:

- **Per prima cosa, Il nome della funzione nella macro:**

```
void grafico(){
```

- **I dati di input, con errori (10 coppie di misure, errore in x trascurabile)**

```
 const int n_points=10;
double x_vals[n_points]= {1,2,3,4,5,6,7,8,9,10};
double y_vals[n_points]= {6,12,14,20,22,24,35,45,44,53};
double y_errs[n_points]= {5,5,4.7,4.5,4.2,5.1,2.9,4.1,4.8,5.4};
```

- **Potete utilizzare anche dei vector/array stl (dopo facciamo un esempio)**

Esempio di Grafico con ROOT

- Creazione del grafico. Lo 0 sta ad indicare che gli errori in x non sono considerati.

```
TGraphErrors *graph=new TGraphErrors(n_points,x_vals,y_vals,0,y_errs);
```

- Titolo del grafico e titoli degli assi X,Y

```
graph->SetTitle("Test della legge di Ohm;DDP (V); Corrente (A)");
```

- Cosmetica del grafico:

```
graph->SetMarkerStyle(kOpenCircle);
graph->SetMarkerColor(kBlue);
graph->SetLineColor(kBlue);
```

- kOpenCircle,kBlue codici predefiniti in ROOT

Esempio di Grafico con ROOT

- **Creazione della Canvas**

TCanvas * mycanvas = new TCanvas();

- **Disegno il grafico graph->Draw("APE");**

- **Opzioni:**
  - ***A*** **: disegna gli assi.**
  - ***P*** **: disegna il marker dei punti.**
  - ***E*** **: disegna le barre di errore.**

- **Coefficiente di correlazione e covarianza fra i punti:**

  graph->GetCorrelationFactor ()

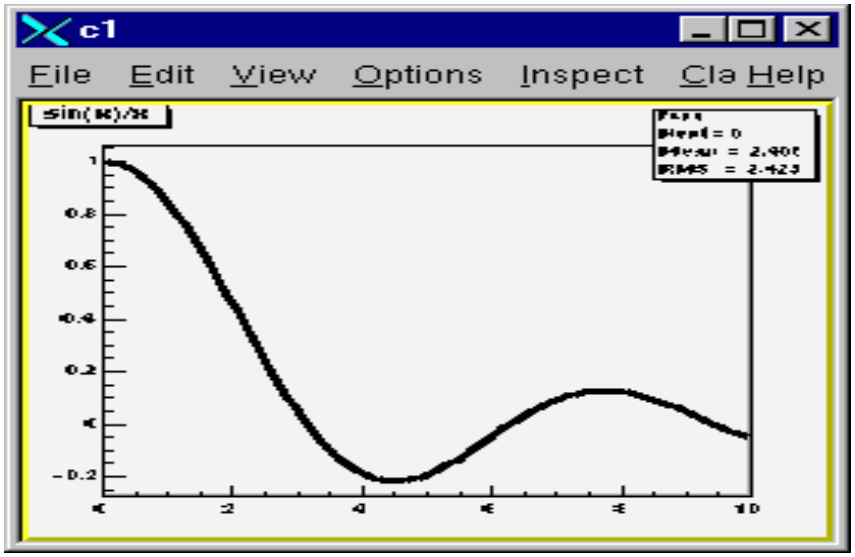
  graph->GetCovariance ()

Funzioni in ROOT

- **Classe per funzioni di una variabile: TF1**
  - **Consente di rappresentare una espressione C++ -like in cui la variabile è x**
  - **Funzioni definite dall'utente**
  - **Sono disponibili anche dei "built in" function objects (gaus, expo, poln,…), vedere in: http://root.cern.ch/root/html/TFormula.html**
  - **Corrispondenti in 2 e 3 dimensioni: TF2, TF3**

Funzioni in ROOT-TF1

1. **Primo modo: espressione della funzione specificata come stringa di caratteri nel costruttore. Sintassi C++:**




```
TF1 *f1 = new TF1("f1", "sin(x)/x",0,10);
f1->Draw(); // disegna la funzione
TF1 *f2 = new TF1("f2", "f1 * 2",0,10); // potete usare
funzioni precedentemente definite)
```

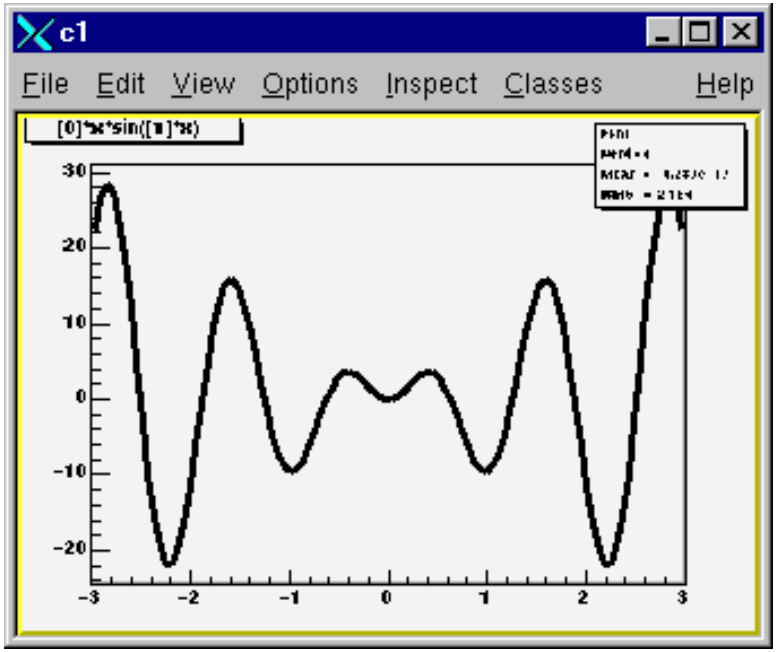
Funzioni in ROOT-TF1

2. **Secondo modo: come prima, ma con parametri:**

```
TF1 *f1 = new TF1("f1","[0]*x*sin([1]*x)",-3,3);
```

Per inizializzare i parametri:

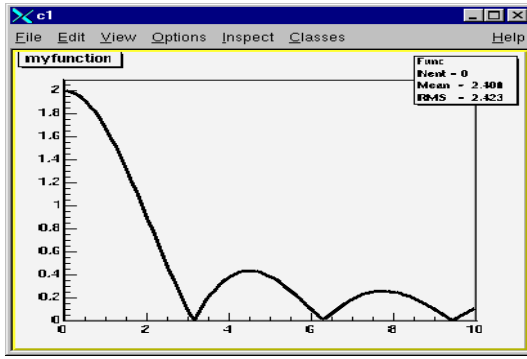
```
f1->SetParameter(0,10);
f1->SetParameter(1,5);

f1->Draw(); // disegna f1
```



Funzioni in ROOT-TF1

3. **Modo più generale: funzione definita dall'utente in una macro (qui sempre 1-D):**

```
Double_t MyFunction(Double_t *x, Double_t *par){
 Float_t  xx     =  x[0];
 Double_t val    =
    TMath::Abs(par[0]*sin(par[1]*xx)/xx);
 return val;
}
```



N.B. Importante rispettare questa segnatura!

TF1 Constructor:

```
TF1 *f1 = new TF1("f1",MyFunction,0,10,2);
```

(NOTA: il 2 indica il numero di parametri in MyFunction)

Inizializzazione dei parametri:

```
f1->SetParameters(2,1); // inizializza i due parametri a
2 e 1
```

Fitting in ROOT

- **Per fare un fit di un grafico/istogramma con la funzione scelta:**

  Definire la funzione:

```
TF1 *fn1 = new TF1("f1","[0] *x*sin([1]*x)",-3,3);
fn1->SetParameters(10,5); // inizializzo i due parametri
a 10, 5 (operazione non sempre necessaria)
```

  Invocare il metodo Fit():

```
aGraph->Fit("f1"); //con il nome della funzione
aGraph->Fit(fn1); // o con il puntatore
```

  Totalmente analogo nel caso di un fit ad un istogramma

```
aHistogram->Fit("f1"); //con il nome della funzione
aHistogram->Fit(fn1); // o con il puntatore
```

Fitting in ROOT

- **Risultati del fit:**

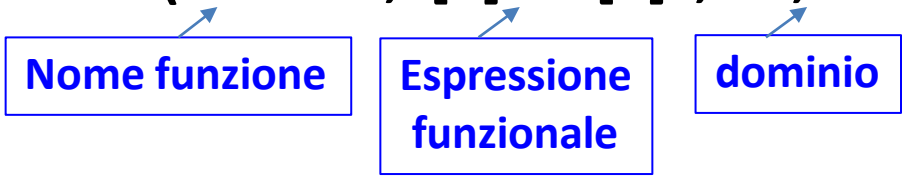
  Valore dei parametri con i loro errori, Chiquadrato e gradi di libertà:

```
fn1->GetParameter(i) //valore del parametro i-esimo
fn1->GetParError(i) //errore sul parametro i-esimo
fn1->GetChisquare() // Chiquadrato
fn1->GetNDF() //Numero di Gradi di libertà
```

Esempio di Grafico con ROOT

- **Adattamento a una retta: definizione di una funzione lineare con due parametri (sono quelli indicati con [0] e [1]).**

```
TF1 * f =new TF1("linear","[0]+x*[1]",0.5,10.5);
```



- **Cosmetica della funzione di fit**

```
f->SetLineColor(kRed);  //linea di colore rosso
f->SetLineStyle(2); //linea tratteggiata
```

- **Fit del grafico con la funzione f**

```
graph->Fit(f);
```

Esempio di Grafico con ROOT

- **La legenda (riporta cosa indicano i punti e la linea):**

```
TLegend *leg = new TLegend(.1,.7,.3,.9,“Prova di Laboratorio ");
leg->AddEntry(graph,"Punti sperimentali");
leg->AddEntry(f,“Fit Lineare");
leg->Draw("Same");
```

- **Salvo il grafico come immagine:**

```
myCanvas->Print(“LeggeDiOhm.gif");
```

- **Chiudo lo scope della funzione grafico**

```
}
```

Esempio di Grafico con ROOT

- **Inserire anche gli errori sui valori in x (che vengono convertiti in incertezze in y secondo il metodo della σ_y equivalente e sommati in quadratura con gli errori in y)**

- **Utilizzare un vector STL**

- **Inserire i valori nel grafico attraverso lettura da file (utilizzate grafico.dat su Virtuale come file di input):**
  - **Fare il grafico inserendo solo gli errori in y**
  - **Fare il grafico inserendo sia gli errori in x che in y**

Grafici - Altri Metodi

Alcuni metodi utili per Esperimenti Finali (grafici):

- **Recupera il contenuto dell'iesimo punto del grafico**

  graph->GetPoint (*i*, *x*, *y*)
- **Recuperano le misure in x, y (ritorna un puntatore a un array) e il numero di punti:**

  graph->GetX ( ) , graph-> GetY ( ) , graph-> GetN ()
- **Integrale dell'area racchiusa dai punti del grafico:**

  graph->Integral ( )
- **Inserisce un punto extra nel grafico:**

  graph->AddPoint (x,y)
- **Sovrascrive il valore del'i-esimo punto del grafico:**

  graph->SetPoint (i,x,y)

Funzioni- Altri Metodi

Alcuni metodi utili per Esperimenti Finali (funzioni):

- **Valutare la funzione in un punto:**
  f->Eval (x)

- **integrale della funzione nel range [a,b]:**
  f->Integral (a, b)