EP578 Computing for Physicists

Topic 9

ROOT: Fitting

Department of
Engineering Physics

University of Gaziantep

Course web page
www.gantep.edu.tr/~bingul/ep578

Nov 2011

Curve Fitting

- Data is often given for discrete values along a continuum.
- You may require estimates at points between discrete values.
- In this section we will consider how to obtain values between the given experimental points using **curve fitting method**.

x	y
---	---
x_1	y_1
x_2	y_2
.	.
.	.
.	.
x_n	y_n

TGraph Fitting

1. Define **`TGraph`** pointer:

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

x	y
---	---
x_1	y_1
x_2	y_2
.	.
.	.
.	.
x_n	y_n

2. Fit using predefined function:

```
gr->Fit("pol1");
gr->Fit("gaus");
```

or using user defined function

```
TF1 *f1 = new TF1("fun", "[0]*x+[1]", 0, 80);
f1->SetParameter(0, 1);
f1->SetParameter(1, 1);
gr->Fit(f1);
```

Linear function: ax+b

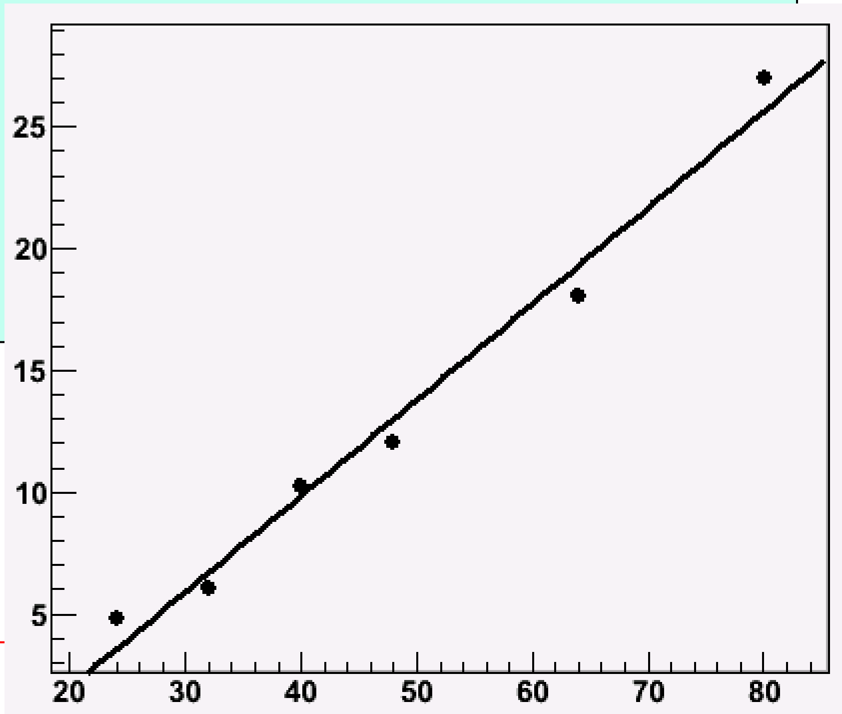
Example 1: *Linear Fit*

The distance required to stop an automobile is a function of its speed. The following data is collected to get this relationship:

v(km/h)	d(m)
24	4.8
32	6.0
40	10.2
48	12.0
64	18.0
80	27.0

Fit the data to a linear function and compute the goodness of the fit.

```
void gfit1()
{
  // values
  const int n = 6;
  double x[n] = {24, 32, 40, 48, 64, 80};
  double y[n] = {4.8, 6.0, 10.2, 12.0, 18.0, 27.0};

  TGraph *gr = new TGraph(n, x, y);
  gr->SetMarkerStyle(20);
  gr->SetMarkerSize(1);
  gr->Draw("AP");

  // fit with polynomials
  gr->Fit("pol1");
  //gr->Fit("pol2");
}
```



```
void gfit2(){
{
  // values
  const int n = 6;
  double x[n] = {24, 32, 40, 48, 64, 80};
  double y[n] = {4.8, 6.0, 10.2, 12.0, 18.0, 27.0};

  TGraph *gr = new TGraph(n, x, y);
  gr->SetMarkerStyle(20);
  gr->SetMarkerSize(1);
  gr->Draw("AP");

  // user defined fit function
  TF1 *f1 = new TF1("fitfun", "[0]*x+[1]", 0, 80);
  f1->SetParameter(0, 1);
  f1->SetParameter(1, 1);

  gr->Fit(f1);
}
```

Example 2: *Weighted Linear Fit*

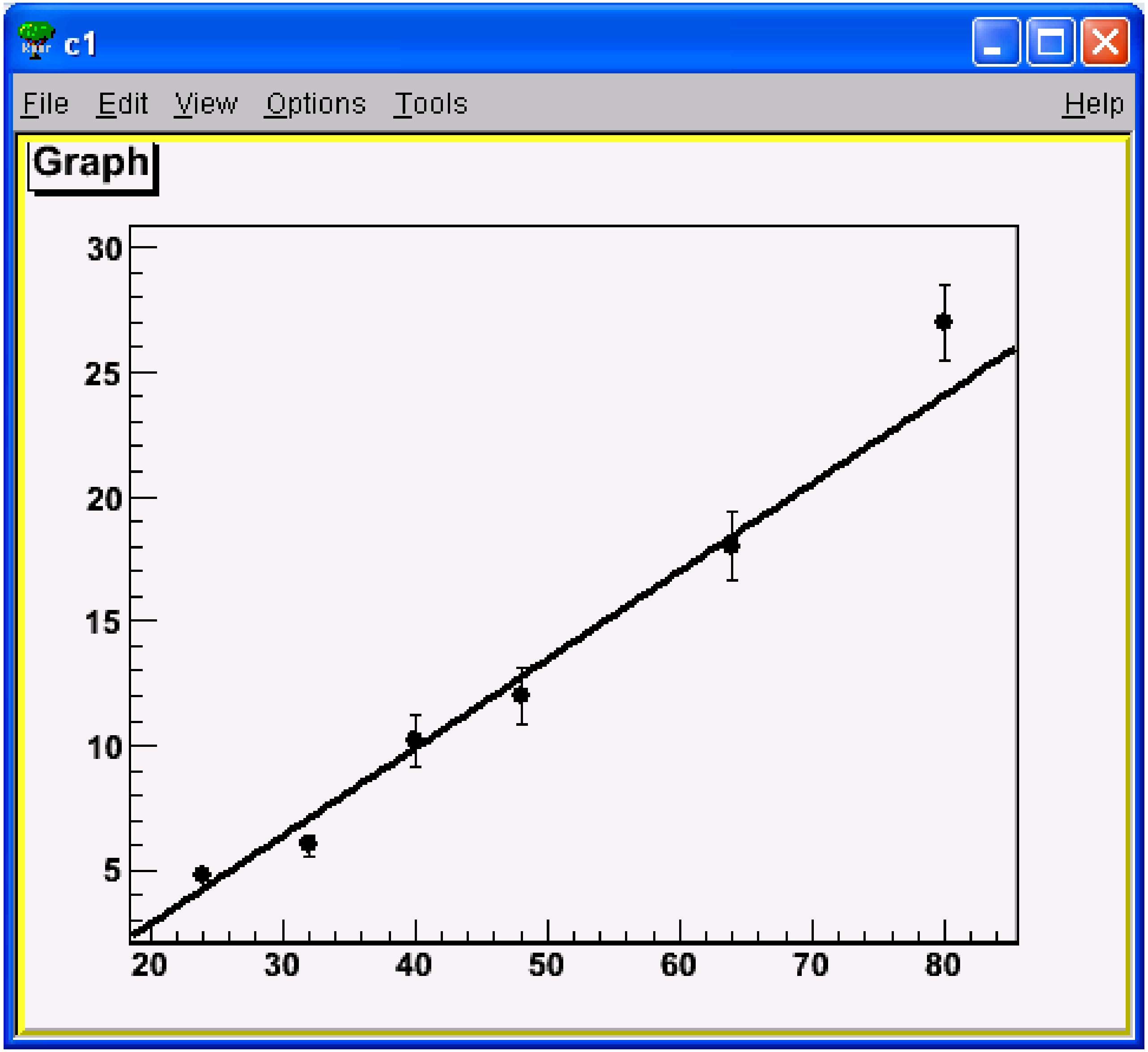
The distance required to stop an automobile is a function of its speed. The following data is collected to get this relationship:

v(km/h)	d(m)
24	4.8 +- 0.3
32	6.0 +- 0.4
40	10.2 +- 1.0
48	12.0 +- 1.1
64	18.0 +- 1.4
80	27.0 +- 1.5

The **+- value** represents the measurement error (one standard deviation).

Fit the data to a linear function and compute the goodness of the fit.

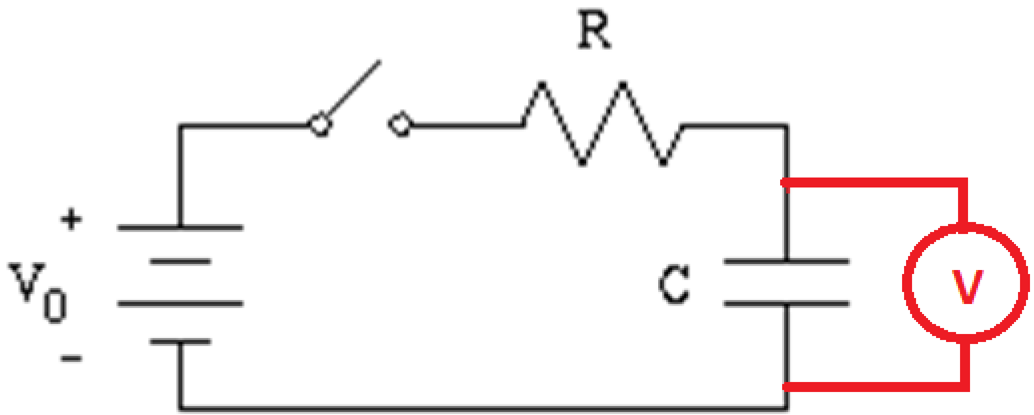
```
void gfit3(){
  // values
  const int n  = 6;
  double  x[n] = {24, 32, 40, 48, 64, 80};
  double  y[n] = {4.8, 6.0, 10.2, 12.0, 18.0, 27.0};
  double ex[n] = {0.0};
  double ey[n] = {0.3, 0.4, 1.0, 1.1, 1.4, 1.5};

  TGraphErrors *gr = new TGraphErrors(n, x, y, ex, ey);
  gr->SetMarkerStyle(20);
  gr->SetMarkerSize(1);
  gr->Draw("AP");

  // user defined fit function
  TF1 *f1 = new TF1("fitfun", "[0]*x+[1]", 0, 80);
  f1->SetParameter(0, 1);
  f1->SetParameter(1, 1);

  gr->Fit(f1);
}
```


c1
File Edit View Options Tools Help
Graph
30
25
20
15
10
5
20 30 40 50 60 70 80

Example 3: *Non-Linear Fit*

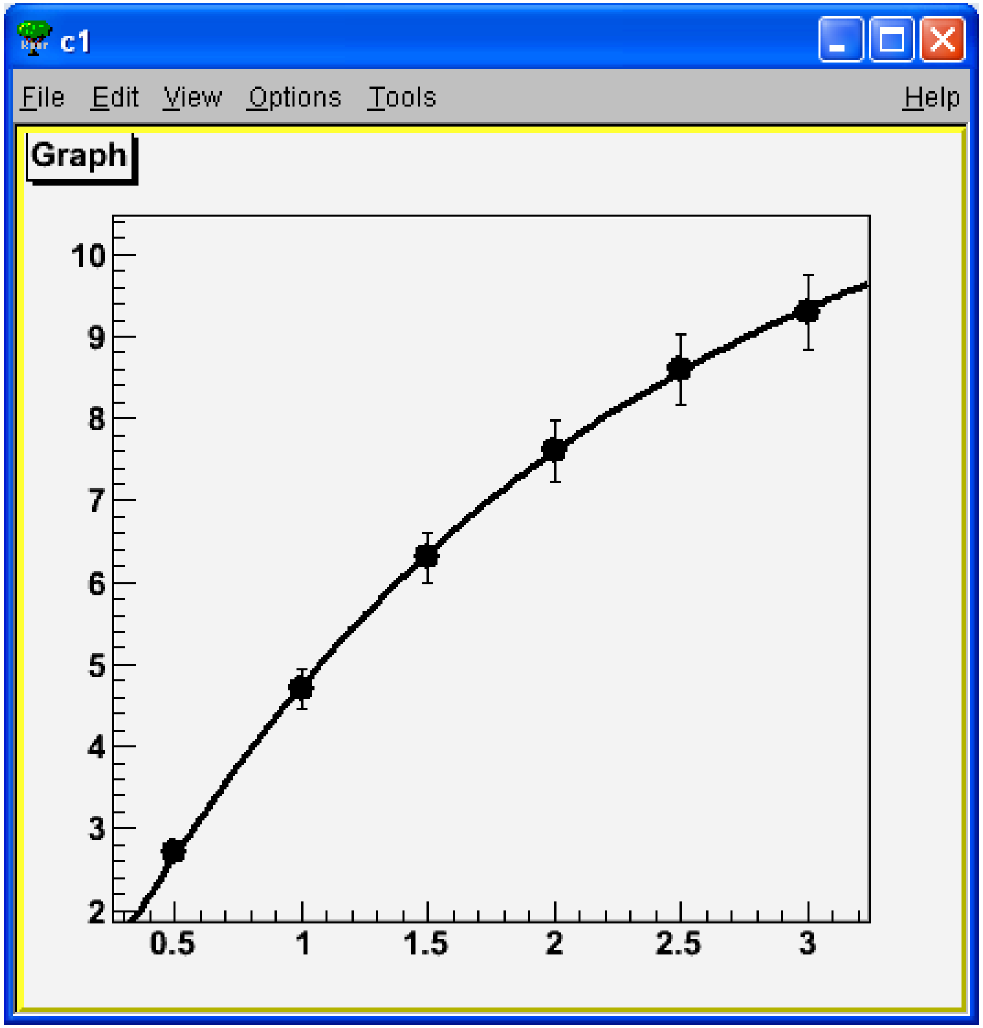
Consider a charging RC circuit containing a resistance (R) and an initially uncharged capacitor (C). The switch is closed at $t = 0$ and using a voltmeter the following experimental data is obtained. Assume that each voltage measurement has %5 error.

t(sec)	Vc(Volts)
0.5	2.7
1.0	4.7
1.5	6.3
2.0	7.6
2.5	8.6
3.0	9.3



where t is time and Vc is potential difference the across the capacitor. Using least square fitting method, determine the time constant and the emf (V_0) of the circuit.

```
void gfit4(){
  // values
  const int n = 6;
  double  x[n]= {0.5, 1.0, 1.5, 2.0, 2.5, 3.0};
  double  y[n]= {2.7, 4.7, 6.3, 7.6, 8.6, 9.3};
  double ex[n]= {0.0};
  double ey[n]= {0.0};

  for(int i=0; i<n; i++) ey[i] = 0.05*y[i];

  TGraphErrors *gr = new TGraphErrors(n, x, y, ex, ey);
  gr->SetMarkerStyle(20);
  gr->SetMarkerSize(1.5);
  gr->Draw("AP");

  // user defined fit function
  TF1 *f1 = new TF1("ff", "[0]*(1-exp(-x/[1]))",0.5,3);
  f1->SetParameter(0, 6.0);
  f1->SetParameter(1, 0.8);

  gr->Fit(f1);
}
```




```
  EXT PARAMETER                                     STEP         FIRST
  NO.   NAME      VALUE            ERROR            SIZE      DERIVATIVE
   1  p0          1.18583e+001   1.01870e+000   1.24531e-004   2.72520e-005
   2  p1          1.95738e+000   2.57916e-001   3.15288e-005  -1.05960e-004
root [6]
```

Histogram Fitting

1. Define **`TH1F`** pointer:

```
TH1F *h = new TH1F("histo","title", n,min, max);
```

2. Fit using predefined function:

```
h->Fit("pol1");
h->Fit("gaus");
```

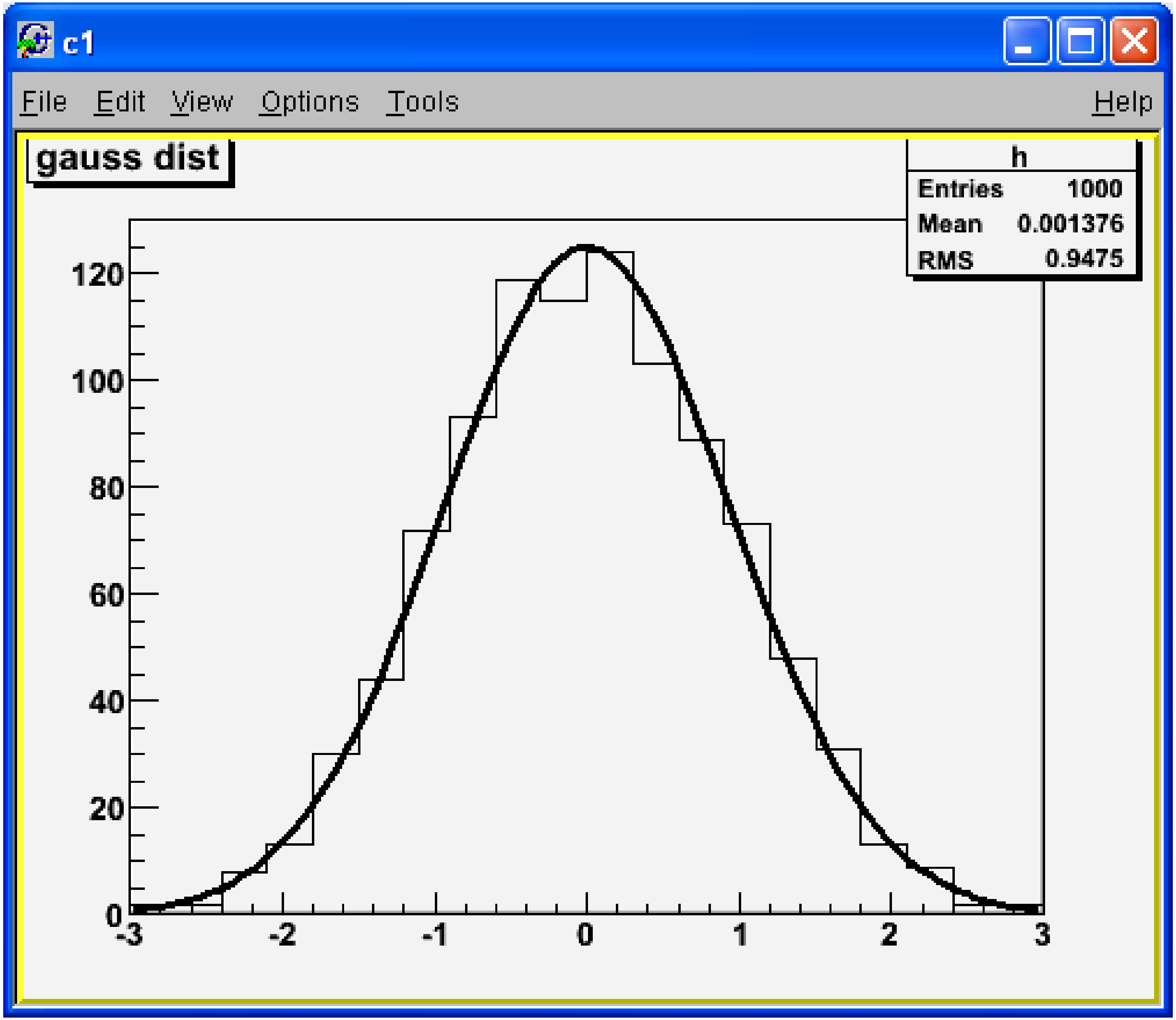
or using user defined function:

```
TF1 *f1 = new TF1("fun", "[0]*x+[1]", min, max);
f1->SetParameter(0, 10);
f1->SetParameter(1, 100);
h->Fit(f1);
```

Linear function: ax+b

```
void hfit1(){
  TH1F *h = new TH1F("h","gauss dist",20,-3,3);

  for(int i=0;i<1000;i++){
    h->Fill(gRandom->Gaus());
  }
  // the Gaussian function
  char *fitfun="[0]*exp(-0.5*(x-[1])/[2]*(x-[1])/[2])";

  // user defined fit function
  TF1 *f1 = new TF1("f", fitfun);
  f1->SetParameter(0, 100.0);
  f1->SetParameter(1, 0.1);
  f1->SetParameter(2, 0.5);

  h->Fit(f1);
}
```

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$


c1
File Edit View Options Tools Help
gauss dist
h
Entries 1000
Mean 0.001376
RMS 0.9475
0
20
40
60
80
100
120
-3
-2
-1
0
1
2
3

Example 4:

Consider that a file contains the data given right.

These numbers are content of a 40-bin histogram.

Assume that histogram range is [0, 300].

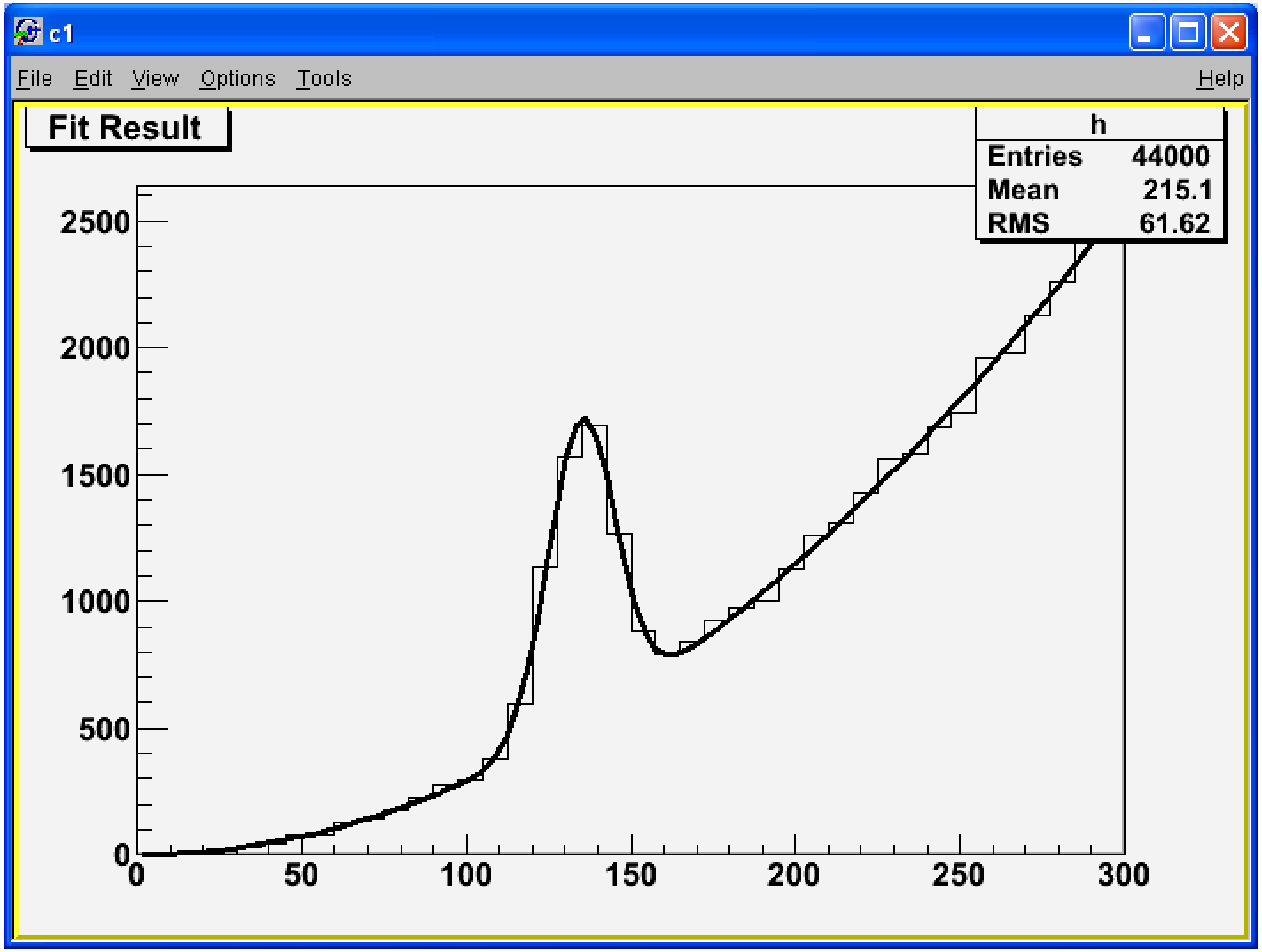
a) Read the file and draw the histogram.

b) Fit the data to the function:

$$f(x) = a_0 \exp\left[-\frac{(x-a_1)^2}{2a_2^2}\right] + a_3 + a_4 x + a_5 x^2$$

where a_i are free parameters.

```
0
1
8
18
33
45
67
84
118
148
176
240
260
316
378
589
1043
1572
1639
1269
938
801
872
868
974
1028
1154
1260
1381
1323
1481
1598
1723
1874
1898
2057
2247
2281
2325
2505
```


c1
File Edit View Options Tools Help
Fit Result
h
Entries 44000
Mean 215.1
RMS 61.62
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

Homework

Solve the following problems. You have to prepare a pdf document and sent it to me until next lecture.

E-mail: bingul[at]gantep.edu.tr (*replace* [at] *with* @)

1. In example 3, assume that each voltage measurement has 5% uncertainty, resistance has the value R = 1 kΩ and battery has 50 Ω internal resistance. Modify the program to calculate capacitance C.

2.

The table shows the experimental results of the measured Coulomb Force, *F*, between two charges, (q_1 and q_2) corresponding to distance *r*.

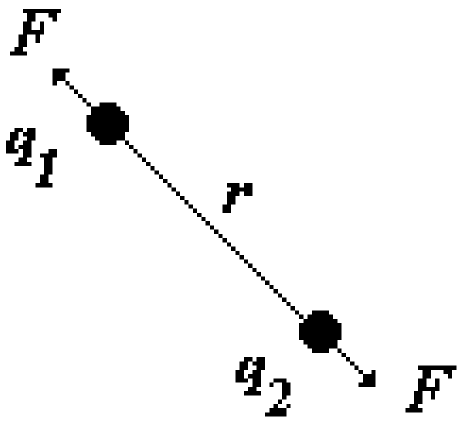
r(cm)	F (N)
40	2.5 +- 0.1
30	4.5 +- 0.1
20	10.1 +- 0.2
10	40.5 +- 0.4
5	162.0 +- 0.9

q1 = 5 µC
q2 = 9 µC

General form of the Coulomb Force is:

$$F(r) = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r^n}$$

Experimental setup



Determine value of ε_0 and *n* using the ROOT fitting tool.

3.

Table shows a data obtained from a radioactive substance. Here *t* is the time in seconds and *R* is the decay rate measured in Bq. Using ROOT fitting tool, determine the half life and identify the nucleus. Assume that each decay rate measurement (*R*) has an associated counting error of sqrt(*R*). **e.g**, for *R* = 300 Bq then measurement error is sqrt(300) = 17.3 Bq.

t(s)	R(Bq)
40	300
100	245
140	210
200	165
240	127
300	110
340	90
400	85
440	58
500	45