

Αντικειμενοστραφείς Γλώσσες Προγραμματισμού C++ / ROOT

Ιωάννης Παπαδόπουλος

Τμήμα Φυσικής, Πανεπιστήμιο Ιωαννίνων

Δεκέμβριος 2025

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolars

Περιεχόμενα

1 Ιστογράμματα στο ROOT Framework

- Ιστογράμματα 1D
- Χρώματα, γραμμές και γεμίσματα
- Ιστογράμματα 2D
- Ιστογράμματα 3D

2 Διαχείριση αρχείων στο ROOT Framework

- Αρχεία απλού κειμένου
- Αρχεία τύπου *.root

3 Παράδειγμα χρήσης TH1 και TRandom: Προσομοίωση της σκέδασης Rutherford

4 Γραφήματα στο ROOT Framework

- Γραφήματα της κλάσης TGraph
- Γραφήματα της κλάσης TGraphErrors
- Γραφήματα της κλάσης TGraphPolar

Ιστογράμματα στο ROOT Framework

Ιστογράμματα στο ROOT Framework

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

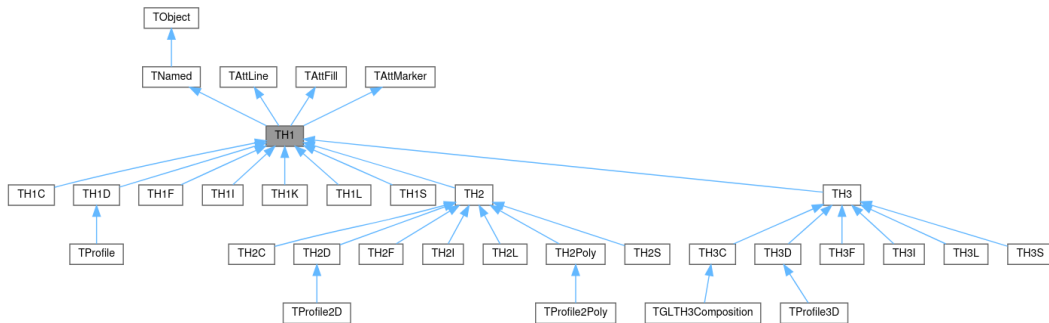
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

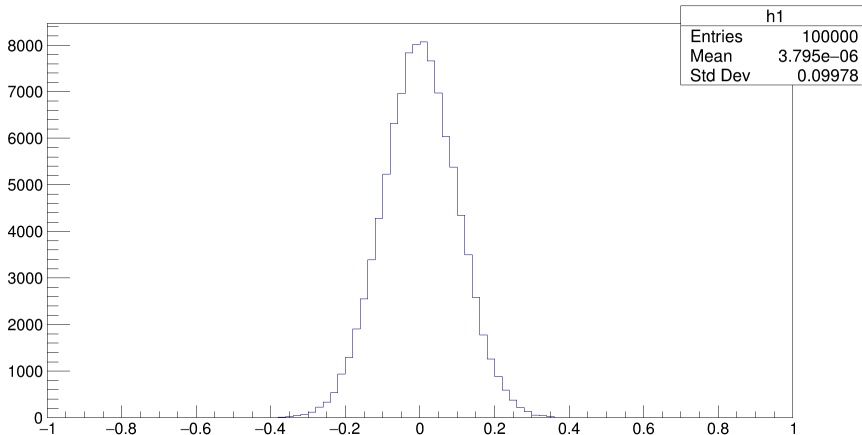
- Το λογισμικό ROOT υποστηρίζει ιστογράμματα μίας (TH1), δύο (TH2) και τριών (TH3) διαστάσεων. Κάθε ιστόγραμμα αποτελεί στιγμιότυπο μίας σχετικής κλάσης. Π.χ.
 - TH1F, TH2F, TH3F (με δεδομένα τύπου float)
 - TH1D, TH2D, TH3D (με δεδομένα τύπου double)
- Ιεραρχία κληρονομικότητας των κλάσεων ιστογραμμάτων:



Ιστογράμματα στο ROOT Framework

- Ένα ιστόγραμμα αποτελείται από ένα σύνολο μερών (bins) σε κάθε διάσταση.
 - 1D -> x-bins
 - 2D -> x-bins και y-bins
 - 3D -> x-bins, y-bins και z-bins

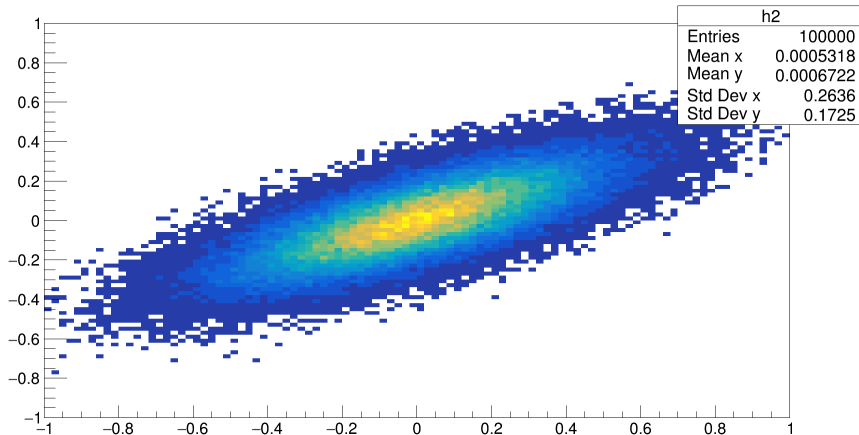
a 1D histo



Ιστογράμματα στο ROOT Framework

- Ένα ιστόγραμμα αποτελείται από ένα σύνολο μερών (bins) σε κάθε διάσταση.
 - 1D -> x-bins
 - 2D -> x-bins και y-bins
 - 3D -> x-bins, y-bins και z-bins

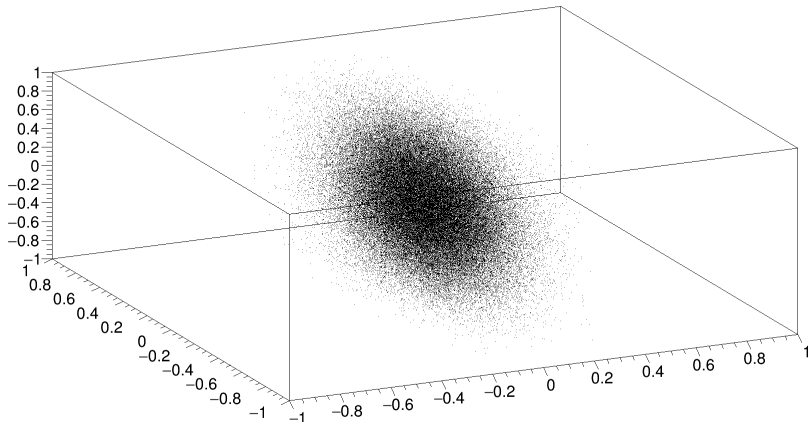
a 2D histo



Ιστογράμματα στο ROOT Framework

- Ένα ιστόγραμμα αποτελείται από ένα σύνολο μερών (bins) σε κάθε διάσταση.
 - 1D -> x-bins
 - 2D -> x-bins και y-bins
 - 3D -> x-bins, y-bins και z-bins

a 3D histo



Ιστογράμματα στο ROOT Framework

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

- Δημιουργία ιστογραμμάτων:
για κάθε διάσταση πρέπει να οριστεί ο αριθμός bins, το ελάχιστο και το μέγιστο Π.χ.

```
TH1F *ptr1 = new TH1F( "h1" , "my 1D histo", 100, -1.5, 1.5);
```

```
TH2F *ptr2 = new TH2F( "h2" , "my 2D histo", 40, -1.5, 1.5, 50, 0, 1.5);
```

```
TH3F *ptr3 = new TH3F( "h3" , "my 3D histo", 50, -1, 1, 30, -2, 1, 40, 0, 2);
```

- Γέμισμα ιστογραμμάτων:

```
for (int i=0; i<100000; i++) {
```

```
    double x=gRandom->Gaus(), y=gRandom->Gaus(), z=gRandom->Gaus();
```

```
    h1->Fill(x); h2->Fill(x, y); h3->Fill(x, y, z);
```

```
}
```

- Σχεδίαση ιστογραμμάτων:

```
auto c = new TCanvas("cv", "title", 600, 200);
```

```
c->Divide(3,1);
```

 : χωρισμός του καμβά σε τρία μέρη (pads)

```
c->cd(1); h1->Draw();
```

 : επιλογή του pad 1, εμφάνιση του h1

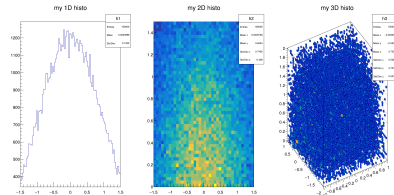
```
c->cd(2); h2->Draw();
```

 : επιλογή του pad 2, εμφάνιση του h2

```
c->cd(3); h3->Draw();
```

 : επιλογή του pad 3, εμφάνιση του h3

χρήση όπως και με τις συναρτήσεις TF1 -> πολυμορφισμός



Ιστογράμματα στο ROOT Framework

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

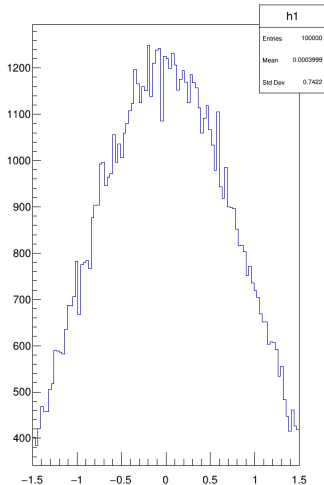
Γραφήματα

TGraph

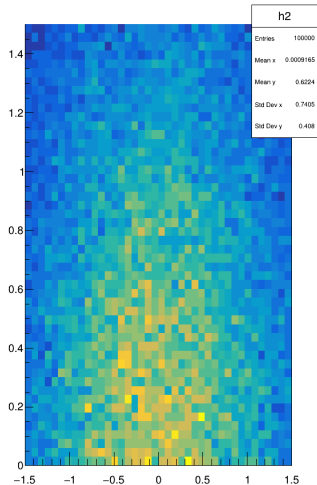
TGraphErrors

TGraphPolars

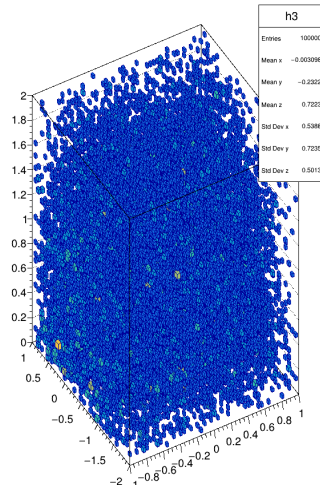
my 1D histo



my 2D histo



my 3D histo



Κατασκευή ενός καμβά, ενός ιστογράμματος 1D και μίας γεννήτριας ψευδοτυχαίων αριθμών:

```
auto c1 = new TCanvas( "c", "My Canvas", 1200, 600);  
auto h1 = new TH1D( "h1", "a 1D histo", 50, -1, 1);  
auto r = new TRandom( time(NULL) );  
for (int i=0; i<5000;i++) {  
    h1->Fill( r->Gaus( 0, 0.2 ) );  
}
```

- Ο καμβάς έχει διαστάσεις 1200x660 (σε πίξελ)
- Ο καμβάς μπορεί να αναφερθεί και ως “c”, έχοντας κληρονομήσει την κλάση TNamed
- Το ιστόγραμμα έχει 50 bins, με $x_{min} = -1$ και $x_{max} = 1$
- Η γεννήτρια ψευδοτυχαίων αριθμών έχει ως σπόρο την τρέχουσα ώρα (Unix Epoch Time)
- Το ιστόγραμμα γεμίζει πέντε χιλιάδες φορές:
κάθε φορά προστίθεται και μία είσοδος (entry) σε ένα τυχαίο bin,
το οποίο προκύπτει από μία κανονική κατανομή (Gaussian) με κέντρο το 0 και $\sigma=0.2$
- Επόμενο βήμα: σχεδίαση του ιστογράμματος

Ιστογράμματα 1D

C++ / ROOT

```
h1->Draw();
```

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

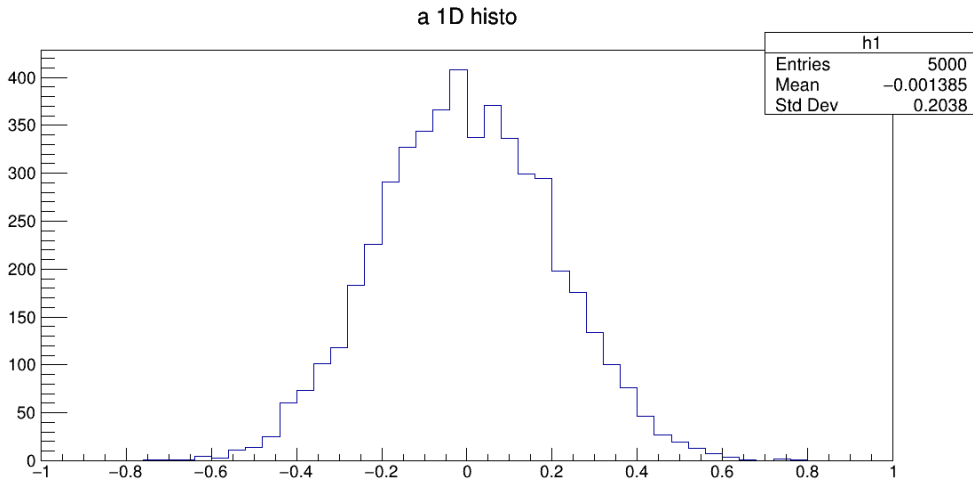
Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolars



Ιστογράμματα 1D

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

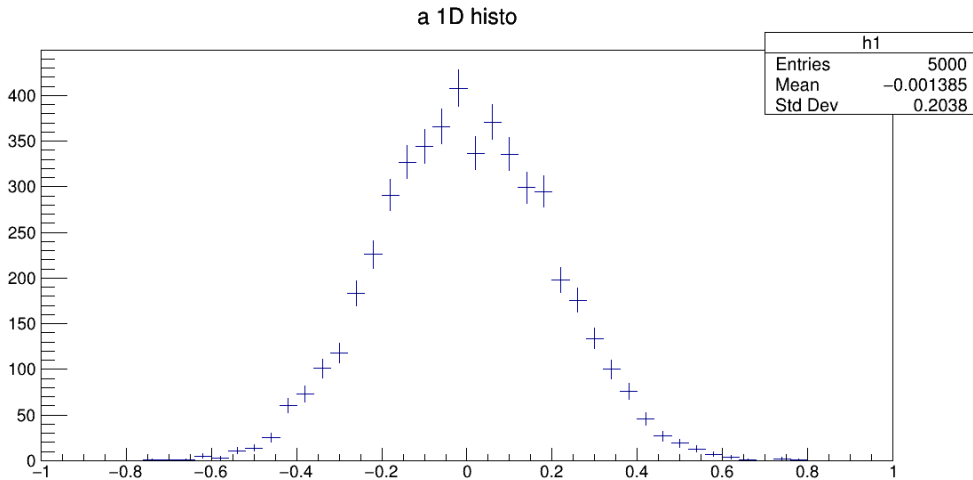
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
h1->Draw("e");
```



Ιστογράμματα 1D

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

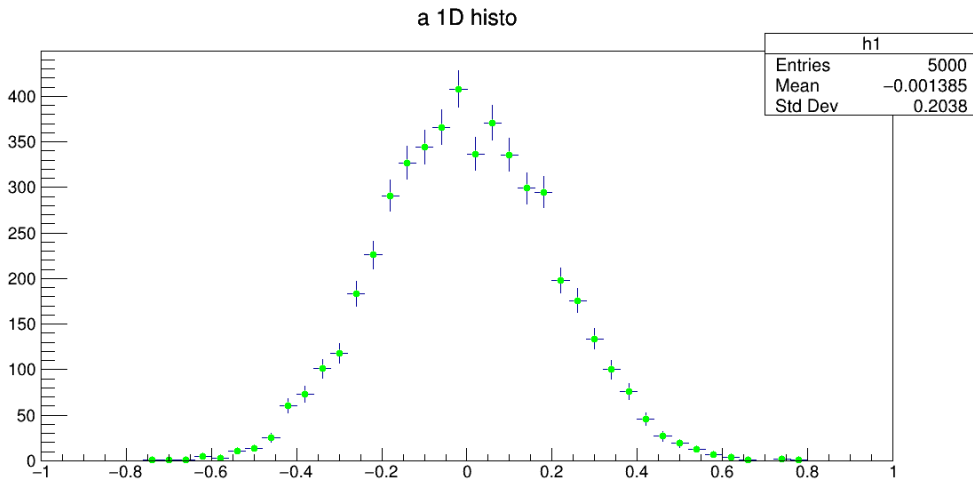
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
h1->SetMarkerColor(3); h1->SetMarkerStyle(8); h1->SetMarkerSize(1); h1->Draw("e");
```



Ιστογράμματα 1D

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

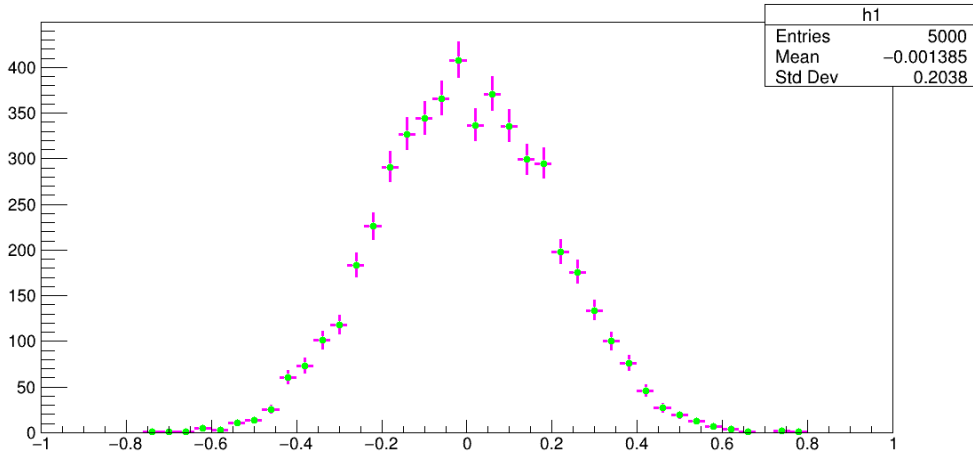
TGraph

TGraphErrors

TGraphPolars

```
h1->SetLineColor(6); h1->SetLineWidth(3); h1->Draw("e");
```

a 1D histo



Ιστογράμματα 1D

C++ / ROOT

```
h1->SetFillColor(kYellow-10); h1->Draw("ebar");
```

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

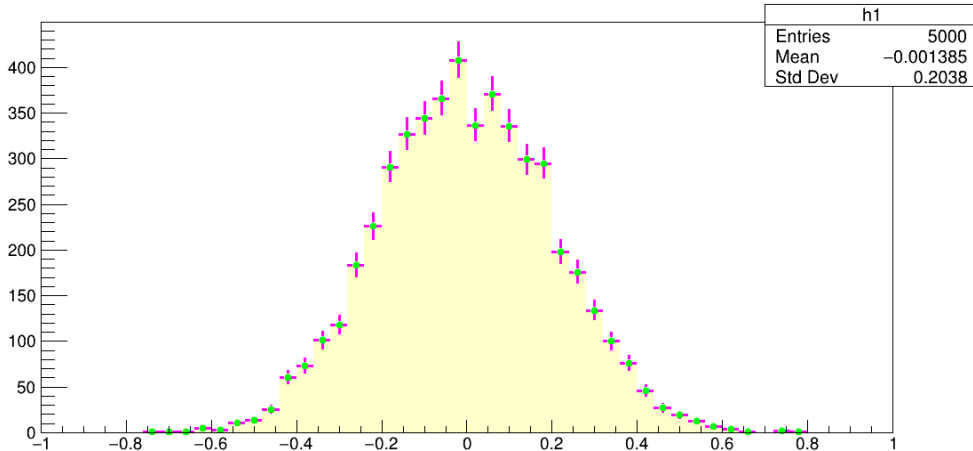
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

a 1D histo



Ιστογράμματα 1D

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

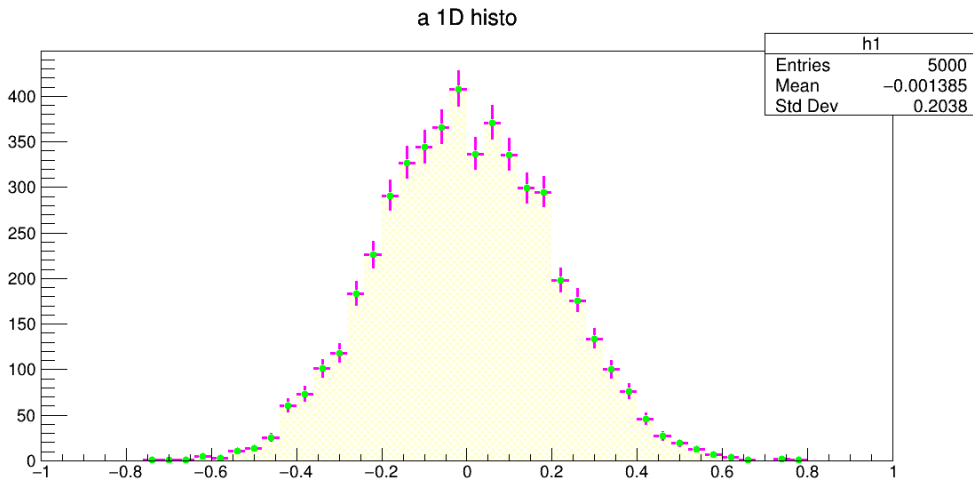
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
h1->SetFillStyle(3008); h1->Draw("ebar");
```



Ιστογράμματα 1D το ένα δίπλα στο άλλο

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
auto c1 = new TCanvas( "c", "My Canvas", 1200, 600);
auto h1 = new TH1D( "h1", "a normal distribution", 50, 0, 10);
auto h2 = new TH1D( "h2", "a Landau distribution", 50, 0, 10);
auto r = new TRandom( time(NULL) );
for (int i=0; i<10000;i++) {
    h1->Fill( r->Gaus( 5, 1.5 ) );    // κανονική κατανομή
    h2->Fill( r->Landau( 3, 0.75 ) ); // κατανομή Landau
}
c1->Divide( 2, 1 );
h1->SetMarkerColor(3);
h1->SetMarkerStyle(8);
h1->SetMarkerSize(1);
h1->SetLineColor(6);
h1->SetLineWidth(3);
c1->cd(1);
h1->Draw("e");
h2->SetMarkerColor(3);
h2->SetMarkerStyle(8);
h2->SetMarkerSize(1);
h2->SetLineColor(6);
h2->SetLineWidth(3);
c1->cd(2);
h2->Draw("e");
```

Ιστογράμματα 1D το ένα δίπλα στο άλλο

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

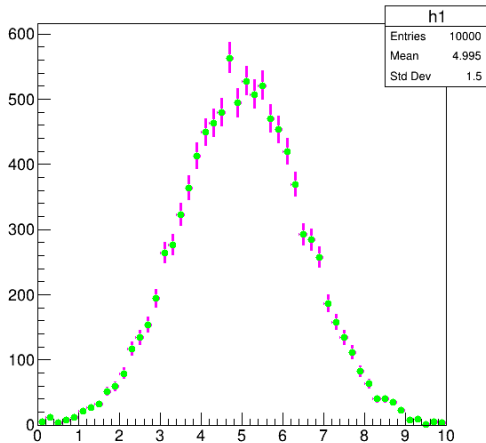
Γραφήματα

TGraph

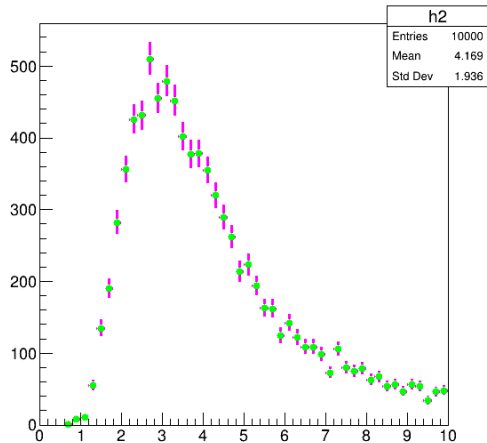
TGraphErrors

TGraphPolar

a normal distribution



a Landau distribution



Ιστογράμματα 1D το ένα δίπλα στο άλλο

Εναλλακτικός κώδικας:

```
auto c1 = new TCanvas( "c", "My Canvas", 1200, 600);
auto h1 = new TH1D( "h1", "a normal distribution", 50, 0, 10);
auto h2 = new TH1D( "h2", "a Landau distribution", 50, 0, 10);
auto r = new TRandom( time(NULL) );
for (int i=0; i<10000;i++) {
    h1->Fill( r->Gaus( 5, 1.5 ) ); // κανονική κατανομή
    h2->Fill( r->Landau( 3, 0.75 ) ); // κατανομή Landau
}
c1->Divide( 2, 1 );
for ( int i : { 1, 2 } ) {
    char s[10]; sprintf( s, "h%d", i ); // κατασκευή ονόματος
    // αναζήτηση αντικειμένου βάσει ονόματος:
    TH1D * h = (TH1D *) gROOT->FindObject(s);
    h->SetMarkerColor(3);
    h->SetMarkerStyle(8);
    h->SetMarkerSize(1);
    h->SetLineColor(6);
    h->SetLineWidth(3);
    c1->cd(i);
    h->Draw("e");
}
```

Ιστογράμματα 1D αλληλεπικαλυπτόμενα

C++ / ROOT

Ι. Παπαδόπουλος

1/2

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
auto c1 = new TCanvas( "c", "My Canvas", 1200, 600);
auto h1 = new TH1D( "h1", "normal distribution A", 100, 0, 10);
auto h2 = new TH1D( "h2", "normal distribution B", 100, 0, 10);
auto h3 = new TH1D( "h3", "normal distribution C", 100, 0, 10);
auto h4 = new TH1D( "h4", "Landau distribution A", 100, 0, 10);
auto h5 = new TH1D( "h5", "Landau distribution B", 100, 0, 10);
auto h6 = new TH1D( "h6", "Landau distribution C", 100, 0, 10);
auto r = new TRandom( time(NULL) );
for (int i=0; i<10000;i++) {
    h1->Fill( r->Gaus( 2, 0.5 ) ); // κανονική κατανομή A
    h2->Fill( r->Gaus( 5, 1.5 ) ); // κανονική κατανομή B
    h3->Fill( r->Gaus( 7.5, 0.35 ) ); // κανονική κατανομή C
    h4->Fill( r->Landau( 3, 0.75 ) ); // κατανομή Landau A
    h5->Fill( r->Landau( 4, 0.5 ) ); // κατανομή Landau B
    h6->Fill( r->Landau( 5, 0.2 ) ); // κατανομή Landau C
}
```

Ιστογράμματα 1D αλληλεπικαλυπτόμενα

C++ / ROOT

2/2

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
double max=0;
for ( int i : { 1, 2, 3, 4, 5, 6 } ) {
    char s[10];    sprintf( s, "h%d", i );
    TH1D * h = (TH1D *) gROOT->FindObject(s);
    h->SetMarkerColor(i);
    h->SetMarkerStyle(8);
    h->SetMarkerSize(1);
    h->SetLineColor(6);
    h->SetLineWidth(3);
    if (h->GetMaximum()>max) max=h->GetMaximum();
    if (i==1)
        h->Draw("e");
    else
        h->Draw("esame");
}
h1->SetMaximum( max*1.05 );
h1->SetTitle("six distributions;x;events");
gStyle->SetOptStat(0);
c1->Draw();
```

Εμφανίζουμε το πρώτο ιστόγραμμα, και έπειτα τα υπόλοιπα από πάνω του χρησιμοποιώντας την παράμετρο “same”.

Ιστογράμματα 1D αλληλεπικαλυπτόμενα

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

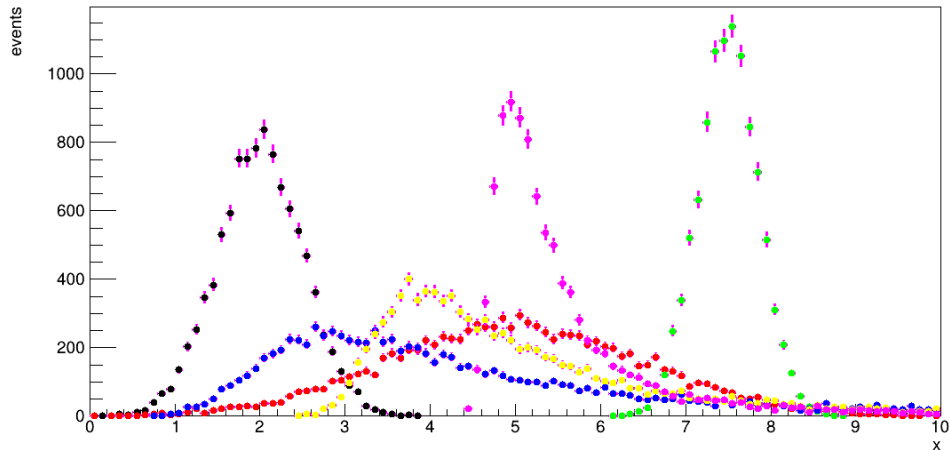
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

six distributions



Χρώματα, γραμμές και γεμίσματα

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

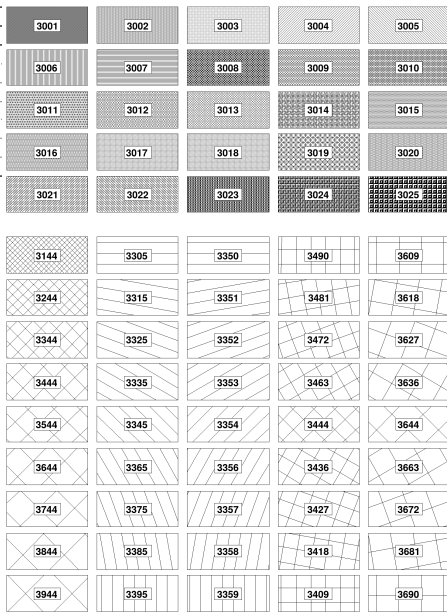
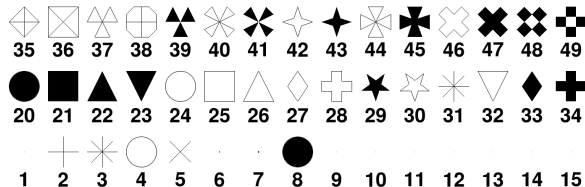
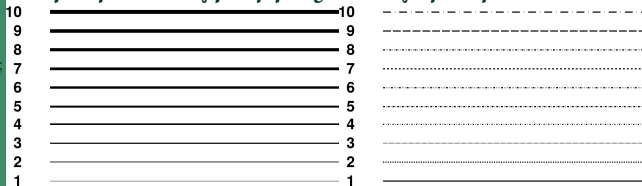
Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar



Χρώματα, γραμμές και γεμίσματα

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

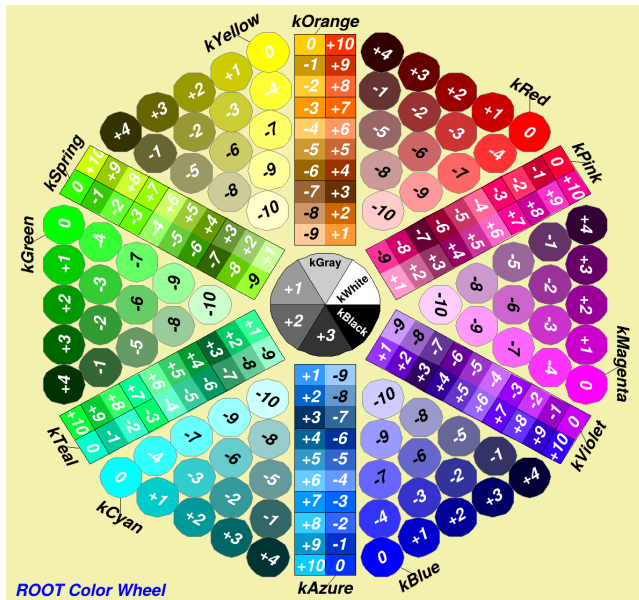
Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

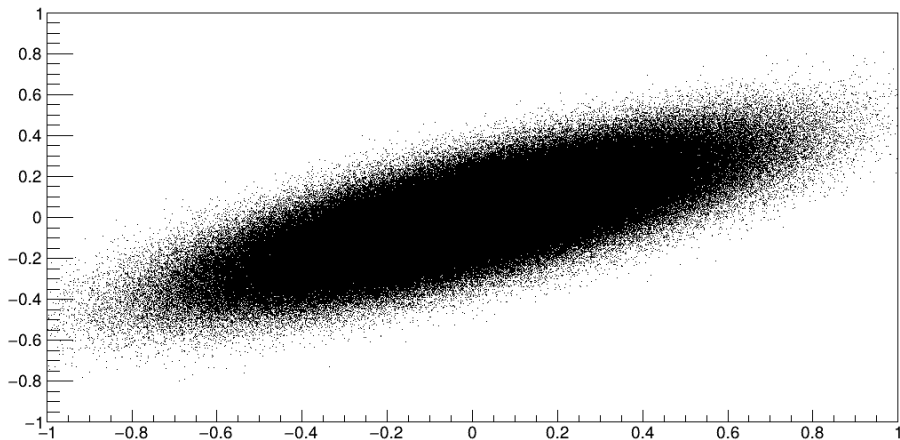


Δημιουργία ενός ιστογράμματος 2D με μία δι-διάστατη κατανομή Gauss που έχει περιστραφεί κατά 30 μοίρες.

```
TCanvas * c1 = new TCanvas("c1","",1200,600);
TH2D * h2 = new TH2D("h2","2D rotated gaussian distribution",100,-1,1,100,-1,1);
for (int i=0; i<1000000;i++) {
    double x, y, u, v, theta=30*M_PI/180;
    x=gRandom->Gaus(0,0.3);
    y=gRandom->Gaus(0,0.1);
    u=x*cos(theta)-y*sin(theta);
    v=x*sin(theta)+y*cos(theta);
    h2->Fill( u , v );
}
gStyle->SetOptStat(false);
h2->SetContour(50);
h2->SetLineWidth(1);
h2->Draw("scat"); // scatter plot
h2->Draw("box"); // box plot
h2->Draw("cont4"); // colour contours
h2->Draw("cont4z"); // colour contours with z-colour scale
h2->Draw("lego"); // 3D LEGO plot
h2->Draw("surf3"); // 3D surface with colour contours on top
h2->Draw("surf3z"); // same, with z-colour scale
```

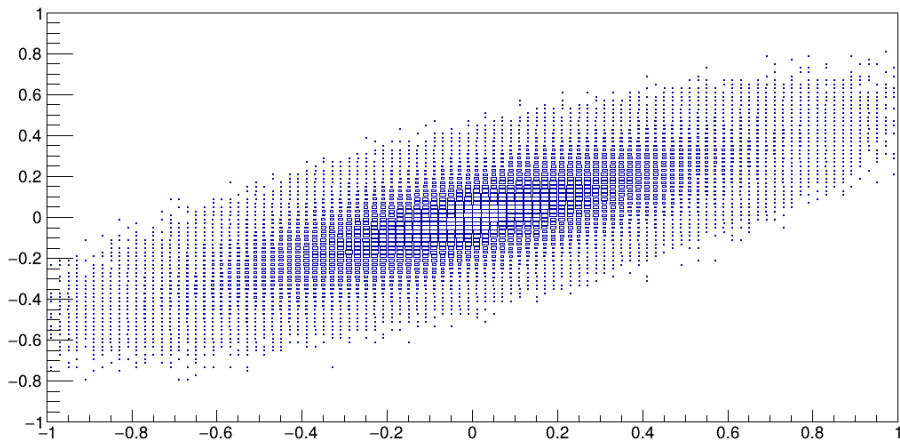
```
h2->Draw("scat"); // scatter plot
```

2D rotated gaussian distribution



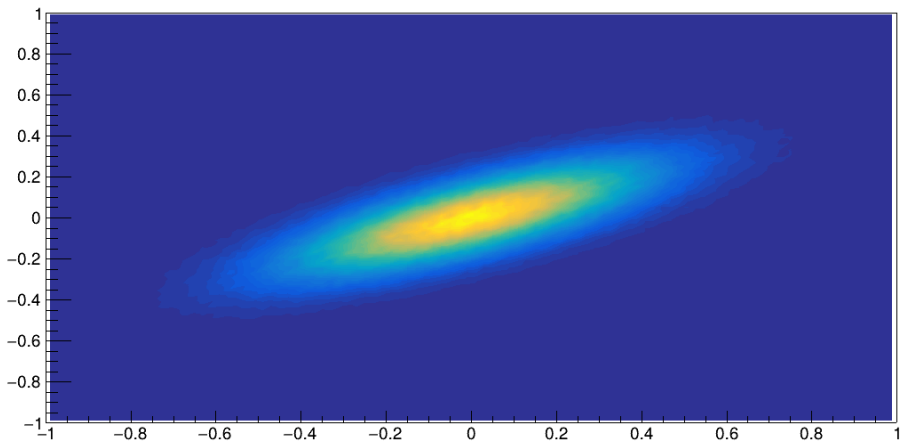
```
h2->Draw("box"); // box plot
```

2D rotated gaussian distribution



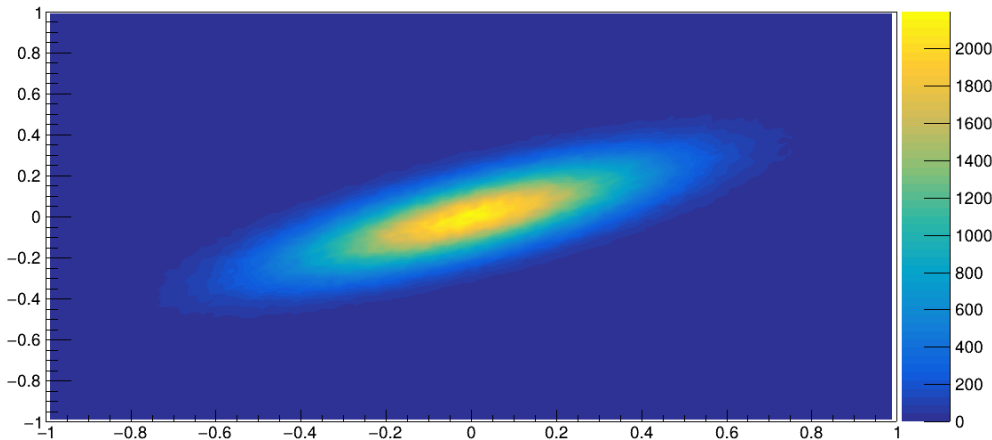
```
h2->Draw("cont4"); // colour contours
```

2D rotated gaussian distribution



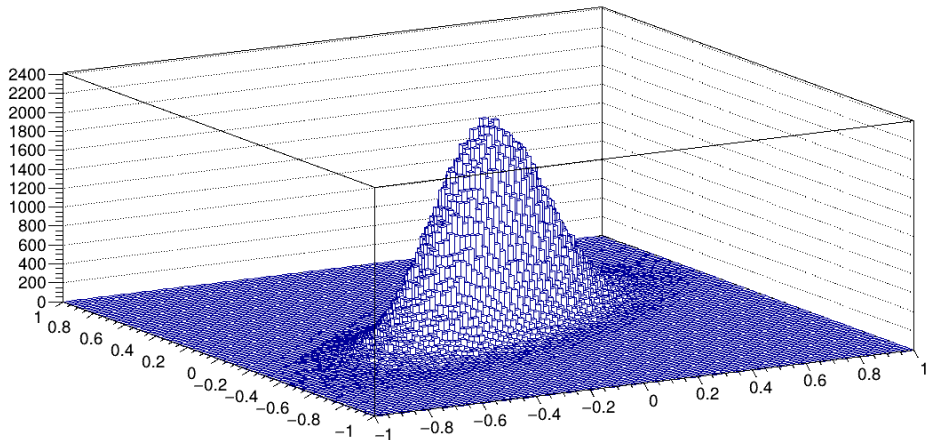
```
h2->Draw("cont4z"); // colour contours with z-colour scale
```

2D rotated gaussian distribution



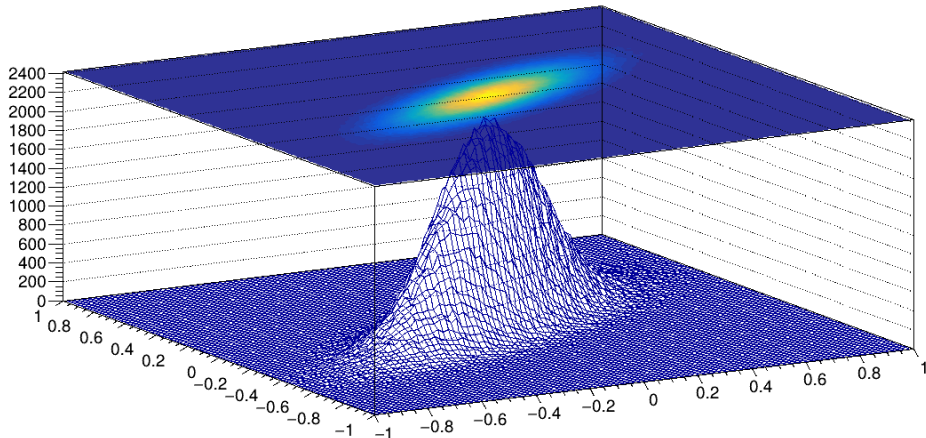
```
h2->Draw("lego"); // 3D LEGO plot
```

2D rotated gaussian distribution



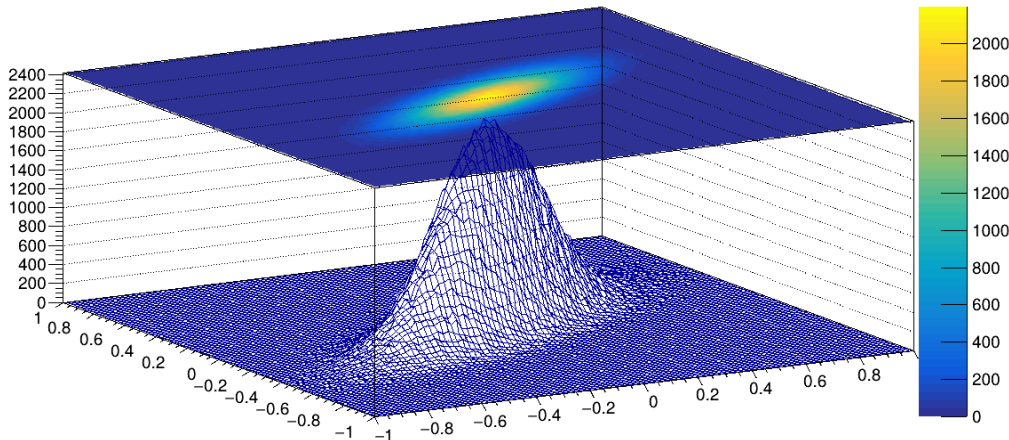
```
h2->Draw("surf3"); // 3D surface with colour contours on top
```

2D rotated gaussian distribution



```
h2->Draw("surf3z"); // same, with z-colour scale
```

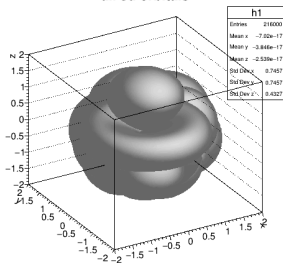
2D rotated gaussian distribution



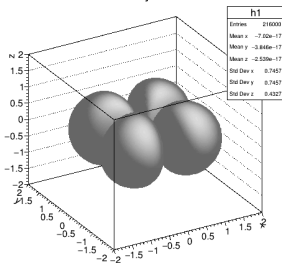
Ιστογράμματα 3D

Δημιουργία ιστογραμάτων 3D για τον υπολογισμό και απεικόνιση ατομικών τροχιακών.

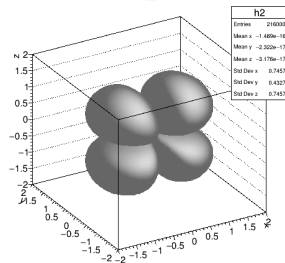
all 3d orbitals



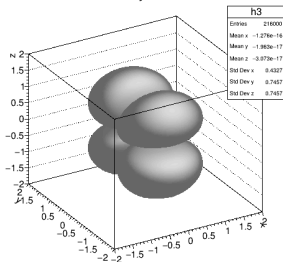
xy



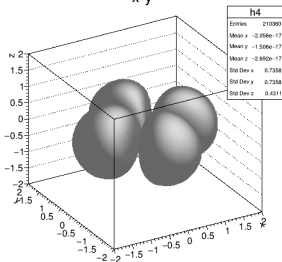
xz



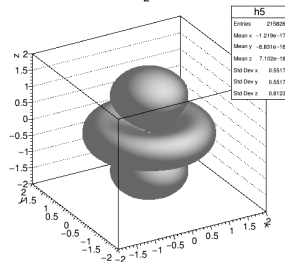
yz



x^2y^2



z^2



Διαχείριση αρχείων στο ROOT Framework

Macro 1: Δημιουργία του αρχείου MyData.txt
Χρήση της κλάσης ofstream της C++.

```
ofstream outfile("MyData.txt");  
auto r = new TRandom( time(NULL) );  
for (int i=0; i<100000;i++) {  
    outfile << r->Gaus( 2, 0.5 ) << endl; // κανονική κατανομή A  
    outfile << r->Gaus( 5, 0.6 ) << endl; // κανονική κατανομή B  
    outfile << r->Gaus( 8, 0.4 ) << endl; // κανονική κατανομή C  
    outfile << r->Landau( 3, 0.7 ) << endl; // κατανομή Landau A  
}
```

Αρχεία απλού κειμένου

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

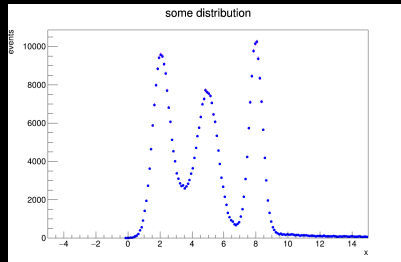
TGraph

TGraphErrors

TGraphPolar

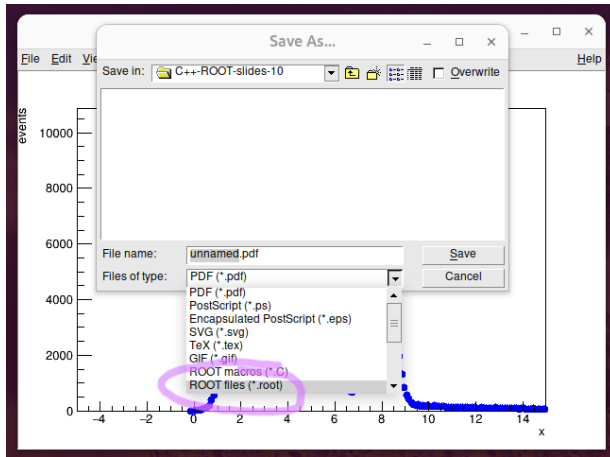
Macro 2: Ανάγνωση του αρχείου MyData.txt και επεξεργασία των δεδομένων
Χρήση της κλάσης ifstream της C++.

```
ifstream infile("MyData.txt");  
if ( ! infile.is_open() ) {  
    cout << "Σφάλμα κατά το άνοιγμα του αρχείου MyData.txt" << endl;  
    gROOT->ProcessLine(".q"); // quit ROOT  
}  
auto c1 = new TCanvas( "c", "My Canvas", 1200, 600);  
auto h1 = new TH1D( "h1", "some distribution;x;events", 300, -5, 25);  
gStyle->SetOptStat(0); // απόκρυψη στατιστικών  
double x;  
int cnt=0;  
while ( infile >> x ) {  
    h1->Fill(x);  
    cnt++;  
}  
cout << "Διάβασα " << cnt << " αριθμούς." << endl;  
h1->SetMarkerColor(4);  
h1->SetMarkerStyle(8);  
h1->SetMarkerSize(1);  
h1->SetLineColor(6);  
h1->SetLineWidth(3);  
h1->Draw("e");
```



Αρχεία δεδομένων τύπου *.root

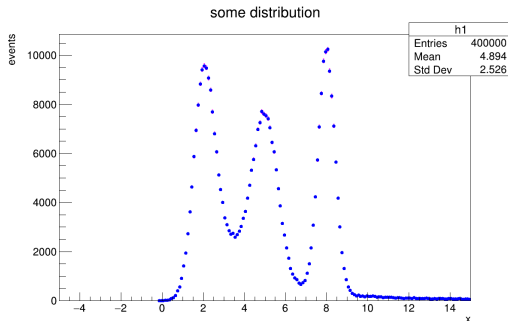
- Τα αρχεία τύπου **root** χρησιμοποιούνται για να αποθηκεύονται σε αυτά αντικείμενα των κλάσεων του ROOT Framework από τη μνήμη RAM.
- Π.χ. μπορεί κανείς αφού δημιουργήσει κάποια διαγράμματα σε έναν καμβά να χρησιμοποιήσει την επιλογή: “File -> Save As...”, οπότε στο αρχείο ROOT θα αποθηκευτεί ο,τιδήποτε υπάρχει στον καμβά αυτόν.



Αρχεία δεδομένων τύπου *.root

- Αργότερα, μέσα από το περιβάλλον του ROOT Framework, για το άνοιγμα αρχείων root χρησιμοποιείται η κλάση TFile.

```
root [0] auto f = new TFile("datafiles-in.root")
(TFile *) 0x59ead1ccaea0
root [1] f->ls()           // περιεχόμενα αρχείου
TFile**      datafiles-in.root
TFile*       datafiles-in.root
KEY: TCanvas  c;1      My Canvas
root [2] f->GetListOfKeys()->Print() // λίστα κλειδιών στο αρχείο
Collection name=THashList, class=THashList, size=1
TKey Name = c, Title = My Canvas, Cycle = 1
root [3] c->Draw()         // σχεδίαση του καμβά c που είχε αποθηκευτεί
```



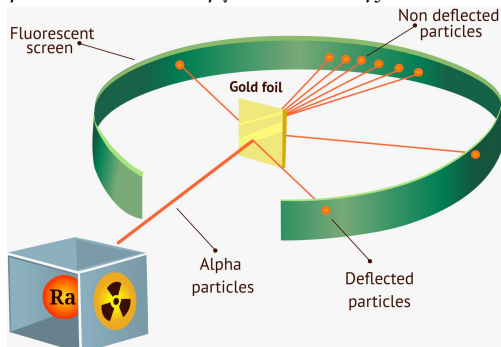
Παράδειγμα χρήσης TH1 και TRandom: Προσομοίωση της σκέδασης Rutherford

Προσομοίωση σκέδασης Rutherford

- Η σκέδαση Rutherford αφορά στο πείραμα κατά το οποίο θετικά φορτισμένα σωματίδια εκτρέπονται από την πορεία τους όταν προσκρούουν σε έναν στόχο, λόγω των απωστικών δυνάμεων Coulomb από τους ατομικούς πυρήνες.
- Η διαφορική ενεργός διατομή ανά μονάδα στερεάς γωνίας περιγράφει την πιθανότητα να σκεδαστεί ένα σωματίδιο σε συγκεκριμένη γωνία θ :

$$\frac{d\sigma}{d\Omega} = \left(\frac{Z_1 Z_2 e^2}{16\pi\epsilon_0 E} \right)^2 \frac{1}{\sin^4(\theta/2)}$$

όπου Z_1 ο ατομικός αριθμός του προσπίπτοντος σωματιδίου, Z_2 ο ατομικός αριθμός του στόχου, e το φορτίο του ηλεκτρονίου, E η κινητική ενέργεια του προσπίπτοντος σωματιδίου, ϵ_0 η διηλεκτρική σταθερά του κενού και θ η γωνία σκέδασης.



Προσομοίωση σκέδασης Rutherford

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρόματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

- Probability Distribution Function (Συνάρτηση Κατανομής Πιθανότητας):

$$\text{PDF}(\theta) = \frac{1}{\sin^4(\theta/2)}$$

- Επιλογή γεγονότων:

- Επιλογή τυχαίας **γωνίας θ**
από **επίπεδη κατανομή** στο διάστημα $[\theta_{min}, 180^\circ]$
Εξαιρείται η γωνία 0, λόγω απειρισμού της PDF.
- Επιλογή τυχαίας **πιθανότητας p**
από **επίπεδη κατανομή** στο διάστημα $[0, \text{PDF}_{max}]$,
όπου $\text{PDF}_{max} = \text{PDF}(\theta_{min})$
- Αν η πιθανότητα p είναι αποδεκτή, δηλαδή στο διαστημα $[0, \text{PDF}(\theta)]$,
το γεγονός γίνεται αποδεκτό και καταγράφεται.

- Με τα γεγονότα που επιλέγονται γεμίζει ένα ιστόγραμμα με τη γωνία θ στον οριζόντιο άξονα.

Προσομοίωση σκέδασης Rutherford

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
1  double pdf(double th) { // συνάρτηση κατανομής πιθανότητας, th σε °
2      return 1.0 / pow(sin(th*TMath::DegToRad()/2.0), 4);
3  };
4
5  TRandom r( time(NULL) ); // γεννήτρια τυχαίων, αρχικοποίηση βάσει της ώρας
6  const int N = 20000000; // αριθμός γεγονότων που θα παραχθούν
7
8  string tit="Rutherford Scattering: #frac{d#sigma}{d#Omega}";
9  tit+="#propto #frac{1}{sin^4(#theta/2)}";
10 tit+=";#theta (#circ);Events";
11
12 TH1D *hTheta = new TH1D("hTheta", tit.c_str(), 180, 0, 180); // bins της 1°
13
14 const double theta_min = 1; // ° , αποφυγή του μηδενός που οδηγεί σε απειρισμό
15
16 double pdf_max = pdf(theta_min); // μέγιστη πιθανότητα στη μικρότερη γωνία
17
18 for (int i=0; i<N; i++) { // παραγωγή γεγονότων
19     double theta, p;
20
21     while (true) { // δειγματοληψία απόρριψης απίθανων γεγονότων
22         theta = r.Uniform(theta_min, 180);
23         p = r.Uniform(0, pdf_max);
24         if (p < pdf(theta)) break; // αποδεκτό γεγονός
25     }
26
27     hTheta->Fill(theta); // σε °
28 }
29
30 TCanvas * c = new TCanvas("c", "Rutherford Simulation", 800, 600);
31 c->SetLogy(); // λογαριθμική κλίμακα στον άξονα y
```

Προσομοίωση σκέδασης Rutherford

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

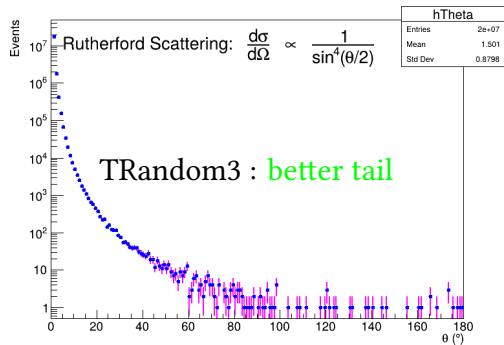
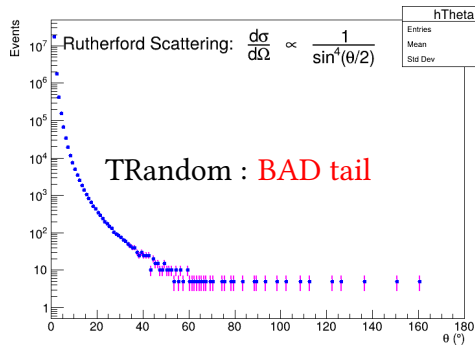
Γραφήματα

TGraph

TGraphErrors

TGraphPolar

```
32 hTheta->SetMarkerColor(4); hTheta->SetMarkerStyle(8); hTheta->SetMarkerSize(0.75);
33 hTheta->SetLineColor(6); hTheta->SetLineWidth(2);
34 hTheta->Draw("e");
35
36 c->Draw();
37 auto t = (TPaveText*) c->GetPrimitive("title");
38 t->SetTextSize(0.05);
39 t->SetX1NDC(0.15); t->SetY1NDC(0.75); t->SetX2NDC(0.70); t->SetY2NDC(0.90);
40
41 c->SaveAs("rutherford.root");
```



Γραφήματα στο ROOT Framework

- Ένα γράφημα αποτελείται από ένα σύνολο n σημείων.
- Έστω ότι οι συντεταγμένες των σημείων παρέχονται ως πίνακες αριθμών: $x[n]$ και $y[n]$
- Τότε το γράφημα μπορεί να οριστεί ως αντικείμενο της κλάσης TGraph ως:

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

- Μέθοδοι κατασκευής (Constructors):

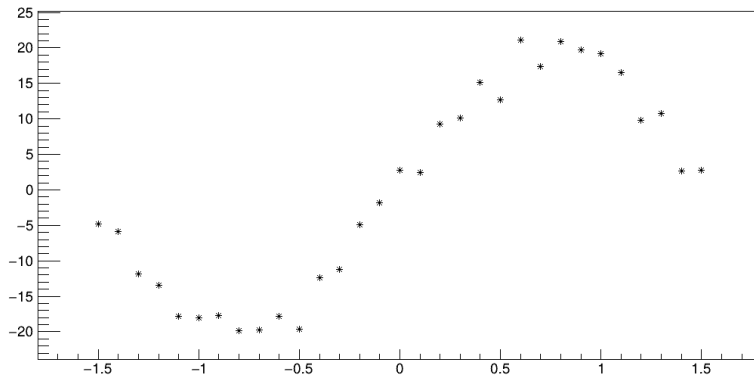
```
TGraph()  
TGraph(Int_t n)  
TGraph(Int_t n, const Double_t* x, const Double_t* y)  
TGraph(Int_t n, const Double_t* y, Double_t start = 0., Double_t step = 1.)  
TGraph(Int_t n, const Float_t* x, const Float_t* y)  
TGraph(Int_t n, const Int_t* x, const Int_t* y)  
TGraph(const TF1* f, Option_t* option = "")  
TGraph(const TGraph& gr)  
TGraph(const TH1* h)  
TGraph(const TVectorD& vx, const TVectorD& vy)  
TGraph(const TVectorF& vx, const TVectorF& vy)  
TGraph(const char* filename, const char* format = "%lg %lg", Option_t* option = "")
```

Κλάση TGraph

- Ακολουθεί ένα παράδειγμα που δημιουργεί και σχεδιάζει ένα γράφημα:

```
double x[31], y[31];  
for ( int i=0 ; i<31 ; i++ ) {  
    x[i] = (i-15)*0.1; // [-1.5,1.5]  
    y[i] = 20*sin(2*x[i]) - gRandom->Gaus(0,1);  
}  
TGraph * gr = new TGraph(31,x,y);  
gr->Draw("A*");
```

Graph



Παράμετροι σχεδίασης:

- "A" : Εκτυπώνονται οι άξονες x και y αυτόματα
- "L" : Συνδέει τα διαδοχικά σημεία με ευθύγραμμα τμήματα
- "C" : Συνδέει τα διαδοχικά σημεία με καμπύλες (splines)
- "*" : Σημειωδοτεί τα σημεία με αστερίσκους
- "P" : Χρησιμοποιεί τον τρέχοντα marker για τα σημεία
- "F" : Γεμίζει την περιοχή που ορίζουν τα σημεία
- "B" : Τυπώνει το γράφημα σε μορφή ιστογράμματος (bar chart)

Κλάση TGraph

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

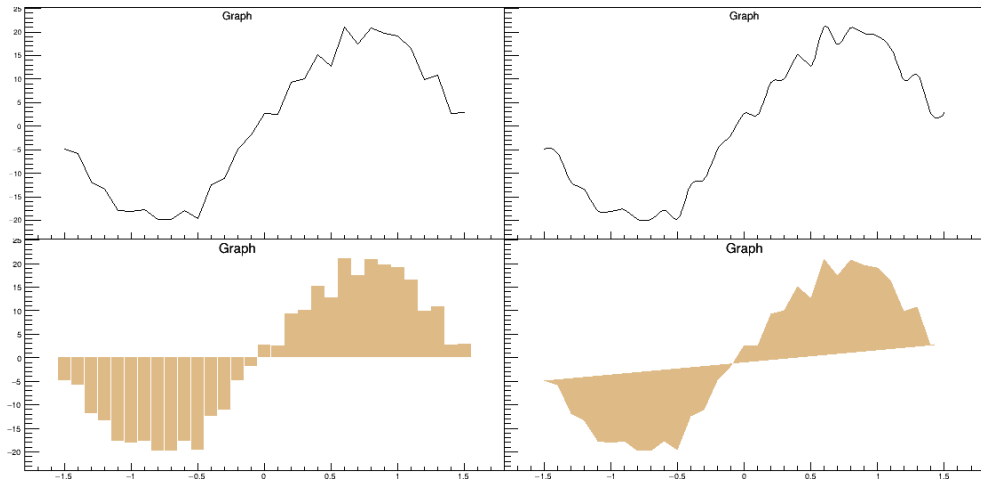
TGraphErrors

TGraphPolar

Draw options:

"AL" "AC"
"AB" "AF"

```
gr->SetFillColor(42);
```



Κλάση TGraphErrors

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

- Πρόκειται για επέκταση της κλάσης TGraph.
- Παρέχονται τα n σημεία και τα σφάλματά τους, π.χ. ως πίνακες αριθμών: $x[n]$, $y[n]$, $sx[n]$ και $sy[n]$.
- Τότε το γράφημα μπορεί να οριστεί ως αντικείμενο της κλάσης TGraphErrors ως:

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

- Μέθοδοι κατασκευής (Constructors):

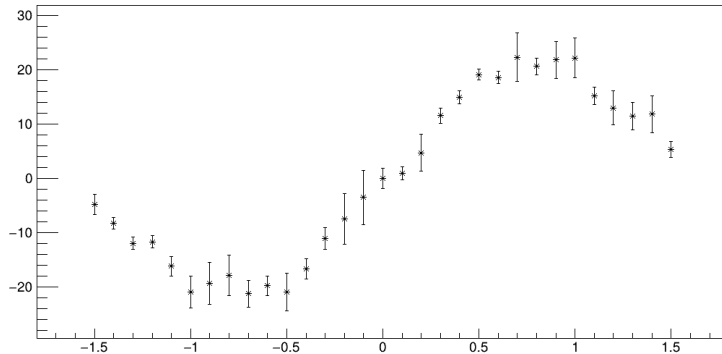
```
TGraphErrors()  
TGraphErrors(Int_t n)  
TGraphErrors(Int_t n, const Double_t* x, const Double_t* y,  
               const Double_t* ex = nullptr, const Double_t* ey = nullptr)  
TGraphErrors(Int_t n, const Float_t* x, const Float_t* y,  
               const Float_t* ex = nullptr, const Float_t* ey = nullptr)  
TGraphErrors(const TGraphErrors& gr)  
TGraphErrors(const TH1* h)  
TGraphErrors(const TVectorD& vx, const TVectorD& vy,  
               const TVectorD& vex, const TVectorD& vey)  
TGraphErrors(const TVectorF& vx, const TVectorF& vy,  
               const TVectorF& vex, const TVectorF& vey)  
TGraphErrors(const char* filename,  
               const char* format = "%lg %lg %lg %lg", Option_t* option = "")
```

Κλάση TGraphErrors

- Ακολουθεί ένα παράδειγμα που δημιουργεί και σχεδιάζει ένα γράφημα:

```
double x[31], y[31], sx[31]={0}, sy[31];  
for ( int i=0 ; i<31 ; i++ ) {  
    x[i] = (i-15)*0.1; // [-1.5,1.5]  
    y[i] = 20*sin(2*x[i]) - gRandom->Gaus(0,2);  
    sy[i] = 1 + fabs(gRandom->Gaus(0,2));  
}  
TGraph * gr = new TGraphErrors(31,x,y,sx,sy);  
gr->Draw("A*");
```

Graph



Κλάση TGraphErrors: προσομοίωση του νόμου του Ohm

- Ακολουθεί ένα παράδειγμα προσομοίωσης του νόμου του Ohm:

```
1 // 40 προσομοιωμένα πειραματικά σημεία τάσης και ρεύματος για R=4.7kΩ
2 double V[40], I[40], sV[40]={0}, sI[40];
3 for ( int i=0 ; i<40 ; i++) {
4     V[i]=(i+1)/4.0;
5     I[i]=V[i]/4.7 + gRandom->Gaus(0, 0.5/4.7);
6     sI[i]=0.5/4.7 + 0.1*I[i];
7 }
8
9 TGraphErrors *gr=new TGraphErrors(40, V, I, sV, sI);
10
11 auto c1 = new TCanvas("c1", "A TGraphErrors object...", 1200, 600);
12 c1->SetFillColor(41);
13 c1->SetGrid();
14
15 gr->SetTitle ("Ohm's Law");
16 gr->GetXaxis()->SetTitle("V [V]");
17 gr->GetYaxis()->SetTitle("I [mA]");
18
19 gr->SetMarkerStyle(21); // τετράγωνα
20 gr->SetMarkerSize(0.7);
21 gr->SetMarkerColor(9);
22 gr->SetLineColor(46);
23 gr->SetLineWidth(2);
24
25 gr->Draw ("AP"); // άξονες και σημεία
26 c1->Draw();
27 c1->GetFrame()->SetFillColor(17);
```

Κλάση TGraphErrors: προσομοίωση του νόμου του Ohm

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

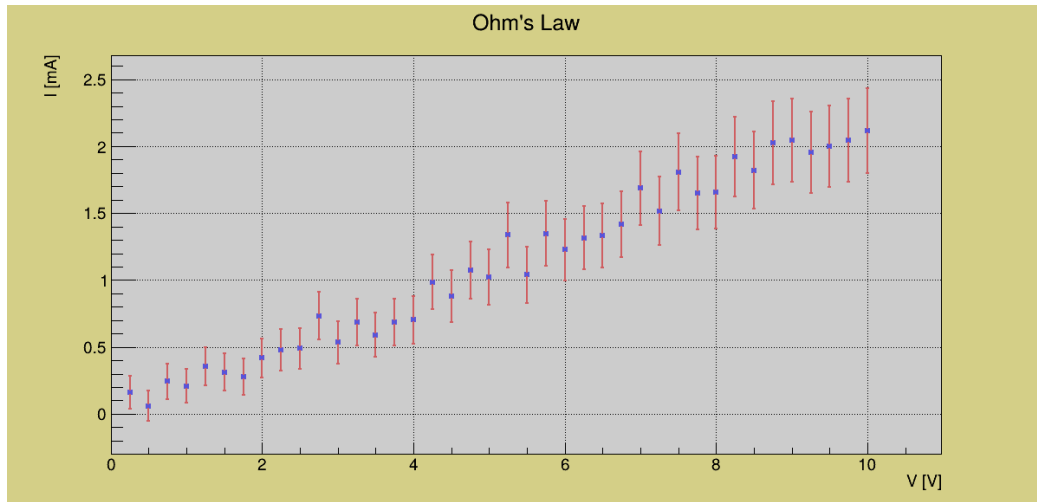
Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolat



Κλάση TGraphPolar

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

- Πρόκειται για επέκταση της κλάσης TGraph σε πολικές συντεταγμένες.
- Παρέχονται τα n σημεία, π.χ. ως πίνακες γωνιών και ακτίνων: $\phi[n]$ και $r[n]$.
- Τότε το γράφημα μπορεί να οριστεί ως αντικείμενο της κλάσης TGraphPolar ως:

```
TGraph * gr = new TGraphPolar( n, phi, r );
```

- Ένα αντικείμενο TGraphPolar υποστηρίζει και σφάλματα στη γωνία και την ακτίνα, κατ' αντιστοιχία της κλάσης TGraphErrors.

- Μέθοδοι κατασκευής (Constructors):

```
TGraphPolar()  
TGraphPolar(Int_t n, const Double_t* theta = nullptr, const Double_t* r = nullptr,  
              const Double_t* etheta = nullptr, const Double_t* er = nullptr)  
TGraphPolar(const TGraphPolar&)
```

Κλάση TGraphPolar

- Ακολουθεί ένα παράδειγμα που σχεδιάζει το $\cos(\varphi)$ σε πολικές συντεταγμένες:

```
1  double phimin=0;
2  double phimax=TMath::Pi();
3
4  double phi[1000];
5  double r[1000];
6
7  TF1 *fpl = new TF1("fpl", "cos(x)", phimin, phimax) ;
8  for(int i=0; i<1000; ++i) {
9      phi[i] = i*(phimax-phimin)/1000 + phimin; // γωνία
10     r[i]    = fpl->Eval(phi[i]);              // ακτίνα
11 }
12
13 TCanvas *c = new TCanvas("c", "TGraphPolar Example", 600,600);
14
15 TGraphPolar *gr = new TGraphPolar(1000, phi, r);
16 gr->SetTitle("cos(#varphi) in polar coordinates: a circle!");
17 gr->SetLineColor(kRed);
18 gr->SetLineWidth(3);
19 gr->Draw ("AL");
20
21 //-----
22 // Προσθέστε αυτές τις εντολές για σχεδίαση με κέντρο το 0
23 c->Draw();
24 gr->SetMinRadial(0);
25 gr->SetMaxRadial(1); // <-- βάλτε τη σωστή τιμή του ρ_max
26 gr->SetMinPolar(0);
27 gr->SetMaxPolar(2*TMath::Pi());
28 //-----
```

Κλάση TGraphPolar

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Ιστογράμματα

1D

Χρώματα κλπ

2D

3D

Αρχεία

απλού κειμένου

αρχεία *.root

Σκέδαση

Rutherford

Γραφήματα

TGraph

TGraphErrors

TGraphPolar

$\cos(\varphi)$ in polar coordinates: a circle!

