

Αντικειμενοστραφείς Γλώσσες Προγραμματισμού C++ / ROOT

Ιωάννης Παπαδόπουλος

Τμήμα Φυσικής, Πανεπιστήμιο Ιωαννίνων

Δεκέμβριος 2024

Περιεχόμενα

- 1 ROOT Framework
- 2 Συναρτήσεις μίας μεταβλητής (TF1)
- 3 Γεννήτριες τυχαίων αριθμών (TF1), αρχεία macro

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

ROOT Framework

Το λογισμικό ROOT που αναπτύσσεται στο CERN:

- Είναι ένα από τα πιο διαδεδομένα εργαλεία στη φυσική υψηλών ενεργειών για την ανάλυση και την οπτικοποίηση δεδομένων.
- Σήμερα περιλαμβάνει πάνω από **1.4 εκατομμύρια γραμμές κώδικα**, κυρίως γραμμένες σε C++ (>95% του κώδικα).
 - Χρησιμοποιούνται επίσης άλλες γλώσσες, όπως η Python (κυρίως για bindings και διεπαφές χρηστών), καθώς και JavaScript και R σε μικρότερη έκταση.
 - Η έμφαση στη C++ οφείλεται στην αποδοτικότητά της και την ευελιξία που προσφέρει για την αντιμετώπιση των εκτεταμένων υπολογιστικών απαιτήσεων της ανάλυσης δεδομένων φυσικής.
- ROOT website: <https://root.cern>
- Οδηγίες εγκατάστασης: <https://root.cern/install>

Το ROOT, αν και αναπτύχθηκε αρχικά για τις ανάγκες της φυσικής υψηλών ενεργειών, χρησιμοποιείται και από άλλους κλάδους της φυσικής και επιστημονικούς τομείς γενικότερα.

Παραδείγματα εφαρμογών περιλαμβάνουν:

- **Αστροφυσική:** Το ROOT χρησιμοποιείται στην ανάλυση δεδομένων από τηλεσκόπια και κοσμολογικά πειράματα, όπως αυτά που αφορούν την ακτινοβολία υποβάθρου μικροκυμάτων ή τις ανιχνεύσεις βαρυτικών κυμάτων.
- **Ιατρική Φυσική:** Εφαρμόζεται στη διαχείριση και ανάλυση δεδομένων από απεικονιστικές μεθόδους (π.χ. PET, MRI) και στις στατιστικές αναλύσεις για την έρευνα στον καρκίνο.
- **Περιβαλλοντικές Επιστήμες:** Το ROOT χρησιμοποιείται για την επεξεργασία δεδομένων που σχετίζονται με την κλιματική αλλαγή ή την ανίχνευση ρύπων.
- **Χρηματοοικονομικά και Ανάλυση Μεγάλων Δεδομένων:** Εξαιτίας της ικανότητάς του να διαχειρίζεται μεγάλα σύνολα δεδομένων, έχει υιοθετηθεί και από τομείς που σχετίζονται με την οικονομία και την ανάλυση χρηματοοικονομικών δεδομένων.

Η ευελιξία και η προσαρμοστικότητα του ROOT, σε συνδυασμό με την υποστήριξη για πολυδιάστατη ανάλυση και γραφικές παραστάσεις, το καθιστούν πολύτιμο εργαλείο και σε επιστήμες πέραν της φυσικής υψηλών ενεργειών.

Παραδείγματα διαγραμμάτων και εικόνων που έχουν δημιουργηθεί με το ROOT:

<https://root.cern/gallery>

Ξεκινώντας το ROOT

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

- Ας θεωρήσουμε πως το λογισμικό ROOT έχει εγκατασταθεί υπό τον φάκελο `/opt/root/`
- Έστω επίσης ότι χρησιμοποιούμε το BASH ως κέλυφος στα τερματικά μας
- Εντολή ρύθμισης μεταβλητών περιβάλλοντος (π.χ. PATH, LD_LIBRARY_PATH, ROOTSYS κ.ά.):
`source /opt/root/bin/thisroot.sh`
- Εντολή εκκίνησης του ROOT:
`root`
- Ξεκινώντας, εμφανίζονται πληροφορίες της έκδοσης του ROOT και εμφανίζεται η γραμμή εντολής του:

```
-----  
| Welcome to ROOT 6.30/02                                     | https://root.cern |  
| (c) 1995-2023, The ROOT Team; conception: R. Brun, F. Rademakers |  
| Built for linuxx8664gcc on Nov 27 2023, 20:01:18          |  
| From tags/v6-30-02@                                       |  
| With c++ (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0           |  
| Try '.help'/'?', '.demo', '.license', '.credits', '.quit'/'q'|  
-----  
root [0]
```

- Η γραμμή εντολής του ROOT υποστηρίζει "TAB completion"

Απλές πράξεις και μικροί κώδικες

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

```
root [0] double sinc(double x) { return x==0?1:sin(x)/x; }
```

```
root [1] sinc(M_PI)
(double) 3.8981718e-17
```

```
root [2] sinc(0)
(double) 1.0000000
```

```
root [3] sinc(0.1)
(double) 0.99833417
```

```
root [4] for (int i=-7; i<=7; i++) {
root (cont'ed, cancel with .@) [5]   cout << sinc(i/2.) << endl;
root (cont'ed, cancel with .@) [6]   }
-0.100224
0.04704
0.239389
0.454649
0.664997
0.841471
0.958851
1
0.958851
0.841471
0.664997
0.454649
0.239389
0.04704
-0.100224
```


Απλές πράξεις και μικροί κώδικες

Το λογισμικό ROOT

```
root [28] TMath::Sqrt(-4)
(double) -nan
```

```
root [29] TComplex::Sqrt( -4 )
(TComplex) (1.22465e-16,2i)
```

```
root [30] TComplex::Power( TComplex(2,-1) , 2)
(TComplex) (3,-4i)
```

```
root [31] for (int i=0; i<5; i++) { sleep(3); cout << time(NULL) << endl; }
1701789316
1701789319
1701789322
1701789325
1701789328
```

Θα πρέπει να μας έχουν πληροφορήσει για τις διαθέσιμες κλάσεις, π.χ.

- Class TMath: <https://root.cern/doc/master/namespaceTMath.html>
- Class TComplex: <https://root.cern/doc/master/classTComplex.html>

Απλές πράξεις και μικροί κώδικες

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

```
root [39] auto z = new TComplex(2,1)
(TComplex *) 0x55e0824725c0
```

```
root [40] *z
(TComplex &) (2,1i)
```

```
root [41] TComplex::Sqrt( *z )
(TComplex) (1.45535,0.343561i)
```

```
root [42] TComplex::Power( *z , 2 )
(TComplex) (3,4i)
```

```
root [43] z->Im
Im
ImplFileLine
ImplFileName
root [43] z->Im()
(double) 1.0000000
```

```
root [44] z->R
Range
Re
Rho
Rho2
root [44] z->Re()
(double) 2.0000000
```

Απλές πράξεις και μικροί κώδικες

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

```
root [46] gRandom->G
Gaus
GetDrawOption
GetDtorOnly
GetIconName
GetName
GetObjectInfo
GetObjectStat
GetOption
GetSeed
GetTitle
GetUniqueID
root [46] gRandom->Gaus(
Double_t Gaus(Double_t mean = 0, Double_t sigma = 1)
root [46] gRandom->Gaus(11.1, 2.5)
(double) 13.597332
```

```
root [47] gRandom->Gaus(11.1, 2.5)
(double) 10.013089
```

```
root [48] gRandom->Gaus(11.1, 2.5)
(double) 13.054491
```

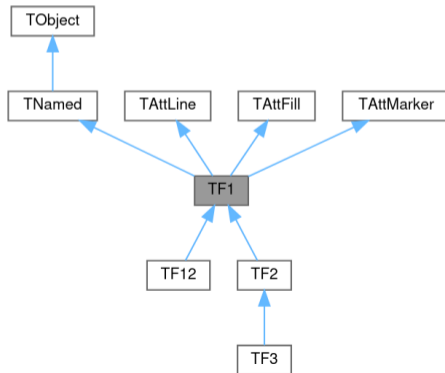
```
root [49] gRandom->Gaus(11.1, 2.5)
(double) 11.024868
```

Συναρτήσεις μίας μεταβλητής (TF1)

Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

- TF1 class reference: <https://root.cern/doc/master/classTF1.html>
- Χρησιμοποιείται για τον ορισμό και χειρισμό συναρτήσεων μίας μεταβλητής.
- Παρέχει πολλές δυνατότητες όπως π.χ.:
 - Σχεδίαση συνάρτησης.
 - Προσαρμογή δεδομένων (fitting).
 - Αριθμητική ολοκλήρωση.
 - Παραγωγή τυχαίων αριθμών βάσει συνάρτησης.

Inheritance diagram for TF1:



Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

Μέθοδοι κατασκευής της TF1 (constructors):

```
TF1 TF1()  
TF1 TF1(const TF1& f1)  
TF1 TF1(const char* name, Double_t xmin, Double_t xmax,  
        Int_t npar, Int_t ndim=1, TF1::EAddToList addToGlobList=EAddToList::kDefault)  
// συνήθως χρησιμοποιούμε αυτή τη μέθοδο κατασκευής:  
TF1 TF1(const char* name, Double_t(*) (Double_t*, Double_t*) fcn, Double_t xmin=0, Double_t xmax=1,  
        Int_t npar=0, Int_t ndim=1, TF1::EAddToList addToGlobList=EAddToList::kDefault)  
// όπως επίσης και αυτή:  
TF1 TF1(const char* name, Double_t(*) (const Double_t*, const Double_t*) fcn, Double_t xmin=0,  
        Double_t xmax=1, Int_t npar=0, Int_t ndim=1, TF1::EAddToList addToGlobList=EAddToList::kDefault)  
TF1 TF1(const char* name, ROOT::Math::ParamFunc f, Double_t xmin=0, Double_t xmax=1,  
        Int_t npar=0, Int_t ndim=1, TF1::EAddToList addToGlobList=EAddToList::kDefault)  
TF1 TF1(const char* name, const char* formula, Double_t xmin=0, Double_t xmax=1,  
        TF1::EAddToList addToGlobList=EAddToList::kDefault, bool vectorize=false)  
TF1 TF1(const char* name, const char* formula, Double_t xmin, Double_t xmax,  
        Option_t* option) // same as above but using a string for option
```

Στο ROOT χρησιμοποιούνται προσαρμοσμένοι τύποι δεδομένων

Ισχύουν οι αντιστοιχίσεις με τύπους της C++ :

- Double_t \Leftrightarrow double
- Int_t \Leftrightarrow int

Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

Δημιουργία αντικειμένων TF1 μέσω φόρμουλας:

```
root [16] TF1 *f = new TF1( "fun" , "sin(x)/x" , -10, 10 )
(TF1 *) 0x5cc1fe39a3c0
root [17] f->Dump()
==> Dumping object at: 0x00005cc1fe39a3c0, name=fun, class=TF1
fXmin          -10          Lower bounds for the range
fXmax          10          Upper bounds for the range
fNpar          0           Number of parameters
fNdim          1           Function dimension
fNpx          100         Number of points used for the graphical representation
fType          0
fNpfits        0           Number of points used in the fit
fNDF           0           Number of degrees of freedom in the fit
fChisquare     0           Function fit chisquare
fMinimum       -1111      Minimum value for plotting
fMaximum       -1111      Maximum value for plotting
fParErrors     ->5cc1fe39a470 Array of errors of the fNpar parameters
fParMin        ->5cc1fe39a488 Array of lower limits of the fNpar parameters
fParMax        ->5cc1fe39a4a0 Array of upper limits of the fNpar parameters
...
```

Ο δείκτης `f` μπορεί να χρησιμοποιηθεί για την κλήση μεθόδων της κλάσης `TF1`. Τό όνομα “`fun`” που δόθηκε, μετατρέπεται επίσης σε δείκτη, και μπορεί να χρησιμοποιηθεί όπως ο δείκτης `f`.

```
root [19] new TF1( "fun2" , "sin(x)*cos(x*1.01)" , -10, 10 )
(TF1 *) 0x5cc1fe218b90
root [20] fun2->ls()
OBJ: TF1 fun2 sin(x)/x : 0 at: 0x5cc1fe218b90
```

Εδώ δημιουργήθηκε ένα αντικείμενο `TF1` χωρίς να αποδοθεί σε κάποιον δείκτη. Μπορούμε όμως να χρησιμοποιούμε το όνομά που του δώσαμε (`fun2`).

Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

Μέθοδος Eval (υπολογισμός τιμής):

```
root [51] new TF1( "f2" , "sin(x)*cos(x*1.1)" , -10, 10 )
(TF1 *) 0x5cc1ff250ba0
root [52] f2->Eval(1.5) // υπολογισμός τιμής
(double) -0.078922690
root [53] f2->Eval(M_PI) // υπολογισμός τιμής
(double) -1.1647083e-16
```

```
root [64] new TF1( "f2" , "sin(x)/x" , -10, 10 )
(TF1 *) 0x5cc1ff1e9c00
root [65] f2->Eval(0) // υπολογισμός τιμής
(double) -nan
```

```
root [55] new TF1( "f2" , "(x!=0)?sin(x)/x:1" , -10, 10 )
(TF1 *) 0x5cc1ff23b000
root [56] f2->Eval(M_PI) // υπολογισμός τιμής
(double) 3.8981718e-17
root [57] f2->Eval(1.5) // υπολογισμός τιμής
(double) 0.66499666
root [58] f2->Eval(0) // υπολογισμός τιμής
(double) 1.0000000
```

Μέθοδος Integral (υπολογισμός ολοκληρώματος):

```
root [66] TF1( "f2" , "sin(x)*cos(x*1.1)" , -10, 10 )
(TF1) Name: f2 Title: sin(x)*cos(x*1.1)
root [67] f2->Integral(
Double_t Integral(Double_t a, Double_t b, Double_t epsrel = 9.999999999999998E-13)
root [67] f2->Integral(0,10)
(double) -1.9299815
```


Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

Μέθοδοι πρόσβασης (getters):

GetChisquare GetCurrent GetDrawOption GetDtorOnly GetExpFormula GetFillColor GetFillStyle
GetFormula GetHistogram GetIconName GetLineColor GetLineStyle GetLineWidth GetLinearPart
GetMarkerColor GetMarkerLineWidth GetMarkerSize GetMarkerStyle GetMarkerStyleBase
GetMaximum GetMaximumStored GetMaximumX GetMethodCall GetMinimum GetMinimumStored
GetMinimumX GetNDF GetName GetNdim GetNpar GetNpx GetNumber GetNumberFitPoints
GetNumberFreeParameters GetObjectInfo GetObjectStat GetOption GetParError GetParErrors
GetParLimits GetParName GetParNumber GetParameter GetParameters GetParent GetProb
GetQuantiles GetRandom GetRange GetSave GetTitle GetUniqueID GetVariable GetX GetXaxis
GetXmax GetXmin GetYaxis GetZaxis

Μέθοδοι πρόσβασης (setters):

SetBit SetChisquare SetCurrent SetDrawOption SetDtorOnly SetFillAttributes SetFillColor
SetFillColorAlpha SetFillStyle SetFitResult SetLineAttributes SetLineColor SetLineColorAlpha
SetLineStyle SetLineWidth SetMarkerAttributes SetMarkerColor SetMarkerColorAlpha SetMarkerSize
SetMarkerStyle SetMaximum SetMinimum SetNDF SetName SetNameTitle SetNormalized SetNpx
SetNumberFitPoints SetObjectStat SetParError SetParErrors SetParLimits SetParName SetParNames
SetParameter SetParameters SetParent SetRange SetSavedPoint SetTitle SetUniqueID SetVectorized

Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

Δημιουργία αντικειμένου TF1 μέσω εξωτερικής συνάρτησης:

Constructor:

```
TF1 TF1(const char* name, Double_t(*) (Double_t*, Double_t*) fcn,  
        Double_t xmin = 0, Double_t xmax = 1, Int_t npar = 0,  
        Int_t ndim = 1, TF1::EAddToList addToGlobList = EAddToList::kDefault)
```

`Double_t(*) (Double_t*, Double_t*)` : συνάρτηση που δέχεται έναν δείκτη σε `Double_t` για τις συντεταγμένες (εν προκειμένω το `x`) και έναν δείκτη σε `Double_t` για τις τυχόν παραμέτρους, και επιστρέφει `Double_t`

```
root [6] Double_t myFunc(Double_t *x, Double_t *p) { return sin(*x)*cos(*x*1.1); }  
root [7] new TF1( "myF", myFunc, -20, 20)  
root [8] myF->Eval(0)  
(double) 0.0000000  
root [9] myF->Eval(M_PI)  
(double) -1.1647083e-16  
root [10] myF->Eval(1.5)  
(double) -0.078922690  
root [11] myF->Integral(0, M_PI)  
(double) -0.23306421
```

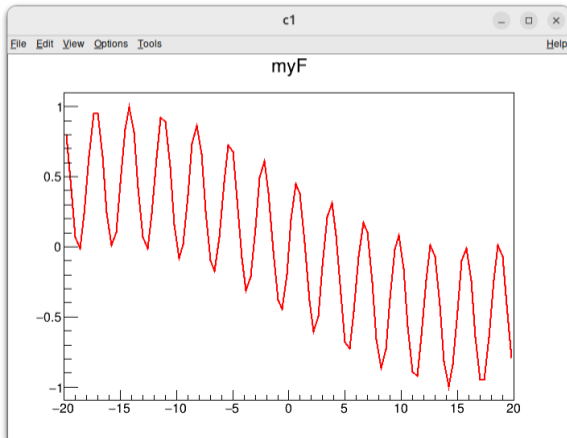
Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

Σχεδίαση αντικειμένου TF1:

```
root [12] myF->Draw()  
Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
```

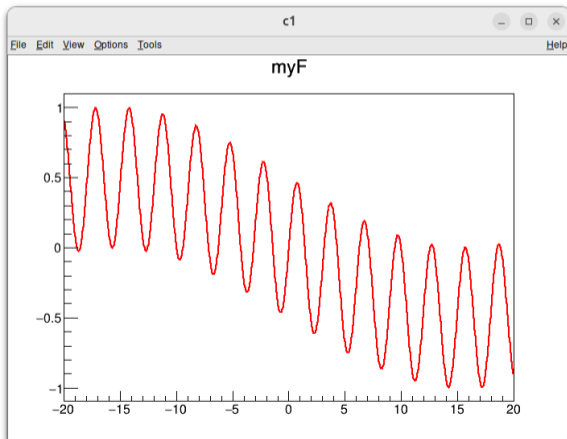
Η μέθοδος Draw σχεδιάζει τη γραφική παράσταση μιας συνάρτησης μέσα σε έναν καμβά (αντικείμενο κλάσης `TCanvas`).

Αν δεν υπάρχει ήδη ανοιχτός καμβάς, δημιουργείται ένας αυτομάτως (εδώ ο `c1`).



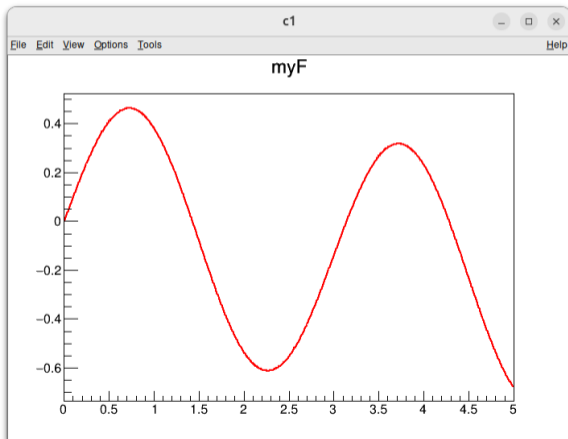
Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

```
root [13] myF->GetNpx() // αριθμός σημείων για τη σχεδίαση εξ ορισμού = 100  
root [14] myF->SetNpx(1000) // μπορεί να αλλάξει για καλύτερη σχεδίαση  
root [15] myF->Draw() // σχεδίαση εκ νέου
```



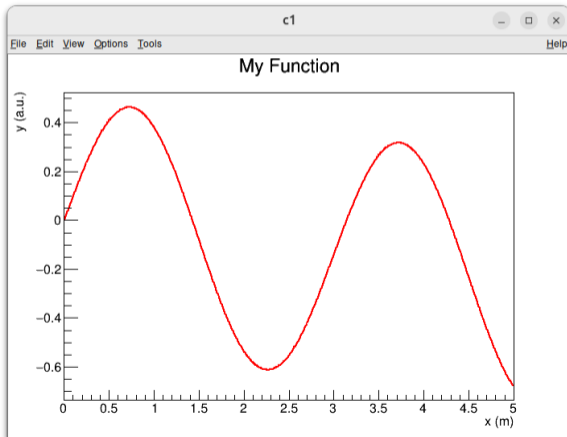
Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

```
root [16] myF->SetRange(0,5) // επιλογή ορίων άξονα x  
root [17] myF->Draw()
```



Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

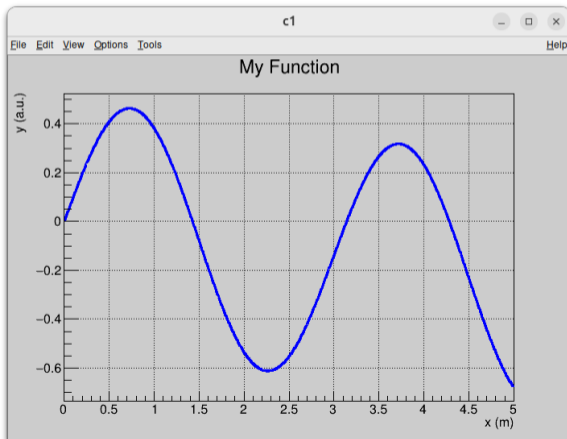
```
root [22] myF->SetTitle("My Function") // τίτλος του γραφήματος  
root [23] myF->GetXaxis()->SetTitle("x (m)") // τίτλος του άξονα x  
root [24] myF->GetYaxis()->SetTitle("y (a.u.)") // τίτλος του άξονα y  
root [25] myF->Draw()
```



Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

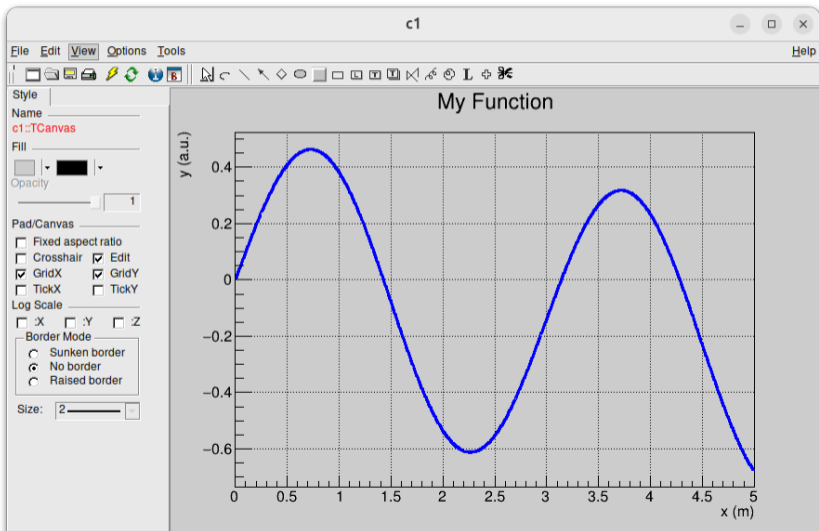
Μπορεί να αλλάξει η μορφή των επιμέρους αντικειμένων. Π.χ.

```
root [26] myF->SetLineColor(4)
root [27] myF->SetLineWidth(3)
root [28] gPad->SetFillColor(kGray)
root [29] gPad->SetGrid()
root [30] myF->Draw()
```



Κλάση TF1 (συναρτήσεις μίας μεταβλητής)

Το ROOT υποστηρίζει διαδραστική αλληλεπίδραση με τα διάφορα αντικείμενα. Πολύ χρήσιμος στην περίπτωση αυτή είναι ο Editor και το Toolbar (μπορούν να ανοίξουν από το μενού View):



Γεννήτριες τυχαίων αριθμών (TF1), αρχεία macro

Κλάση TRandom (τυχαίοι αριθμοί)

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

ROOT
framework

Κλάση TF1

TRandom,
macros

```
root [0] TRandom *r = new TRandom(12345) // νέα γεννήτρια τυχαίων με SEED=12345
(TRandom *) 0x57411829a270
root [1] r->Gaus(
Double_t Gaus(Double_t mean = 0, Double_t sigma = 1)
root [2] for (int i=0; i<2000; i++) cout << r->Gaus(100, 10) << endl
101.888
113.367 // κατανομή Gauss
115.878
107.967
91.3687
120.498
83.8306
87.5466
107.743
root [3] for (int i=0; i<2000; i++) cout << r->Uniform() << endl
0.0391979
0.241429 // επίπεδη κατανομή στο διάστημα [0,1]
0.639249
0.157915
0.909093
0.438855
0.862305
0.498328
0.163414
0.780893
root [4] for (int i=0; i<20; i++) cout << r->Uniform(100,110) << endl
102.097
106.723 // επίπεδη κατανομή στο διάστημα [100,110]
106
108.805
102.146
101.335
105.579
109.876
104.315
```

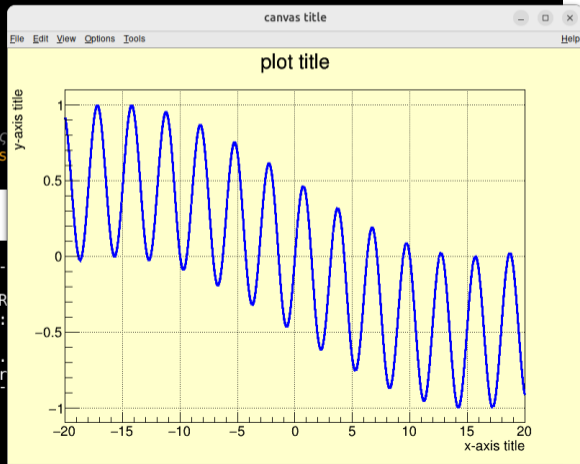
Χρησιμοποιώντας αρχεία πηγαίου κώδικα (macros)

Αρχείο (ROOT macro): mymacro1.C

```
{ // mymacro1.C
  Double_t myFunc(Double_t *x, Double_t *p) {
    return sin(*x)*cos(*x*1.1);
  }
  TF1 *f = new TF1( "myF", myFunc, -20, 20);
  TCanvas *cv = new TCanvas("cname" , "canvas title", 800, 600);
  cv->cd(); // change directory
  myF->SetNpx(10000);
  myF->SetLineColor(4);
  myF->SetLineWidth(3);
  gPad->SetFillColor(kYellow-10);
  gPad->SetGrid();
  myF->Draw();
  // θέσε όλους τους τίτλους μεριάς, παρέχοντας
  myF->SetTitle("plot title;x-axis title;y-axis title");
}
```

Εκτέλεση του ROOT macro:

```
[user20 @ pc198 15] root -x mymacro1.C
-----
| Welcome to ROOT 6.30/02
| (c) 1995-2023, The ROOT Team; conception: R
| Built for linuxx8664gcc on Nov 27 2023, 19:
| From tags/v6-30-02@
| With c++ (Ubuntu 11.4.0-lubuntu1~22.04) 11.
| Try '.help'/'?', '.demo', '.license', '.cr
-----
root [0]
Processing mymacro1.C...
root [1] .q
[user20 @ pc198 16]
```



Χρησιμοποιώντας αρχεία πηγαίου κώδικα (macros)

Αρχείο (ROOT macro): mymacro2.C

```
{ // mymacro2.C
  R = new TRandom( time(NULL) );
  // το ROOT προσθέτει αυτομάτως τον τύπο του R
  for ( int i=0; i<15; i++ )
    cout << R->Gaus( 24.5, 2.3 ) << endl;
}
```

Εκτέλεση του ROOT macro:

```
[user20 @ pc198 18] root -l mymacro2.C
root [0]
Processing root1.C...
26.146
24.3251
22.5565
22.1559
23.7621
25.0886
23.7708
29.221
24.7098
25.2809
24.8183
25.3132
25.7973
25.9213
25.501
root [1] .q
[user20 @ pc198 19]
```

Χρησιμοποιώντας αρχεία πηγαίου κώδικα (macros)

Αρχείο (ROOT macro): mymacro3.C

```
{ // mymacro3.C
  Double_t myFunc(Double_t *x, Double_t *p) {
    return sin(x[0])*cos(x[0]*p[0]);
  }
  TCanvas *cv = new TCanvas("cname" , "canvas title", 800, 600);
  cv->cd(); // change directory
  R = new TRandom( time(NULL) );
  TF1 *f = new TF1( "myF", myFunc, -20, 20, 1);
  myF->SetTitle("overlapped plots;x-axis title;y-axis title");
  myF->SetNpx(10000);
  myF->SetLineWidth(1);
  myF->SetLineColor(4);
  gPad->SetFillColor(kYellow-10);
  gPad->SetGrid();
  string opt="";
  for ( int i=0; i<20; i++ ) {
    myF->SetParameter(0,R->Gaus( 1.0, 0.01 ));
    myF->DrawClone( opt.c_str() );
    opt="same";
  }
}
```

Εκτέλεση του ROOT macro:

```
[user20 @ pc198 23] root -l mymacro3.C
root [0]
Processing mymacro3.C...
```

Χρησιμοποιώντας αρχεία πηγαίου κώδικα (macros)

Αρχείο (ROOT macro): mymacro3.C

```
{ // mymacro3.C
  Double_t myFunc(Double_t *x, Double_t *p) {
    return sin(x[0])*cos(x[0])*p[0];
  }
  TCanvas *cv = new TCanvas("cn", "plot title");
  cv->cd(); // change directory
  TRandom *R = new TRandom( time(NULL) );
  TF1 *f = new TF1( "myF", myFunc, -20, 20, 1 );
  myF->SetTitle("overlapped plot");
  myF->SetNpx(10000);
  myF->SetLineWidth(1);
  myF->SetLineColor(4);
  gPad->SetFillColor(kYellow-10);
  gPad->SetGrid();
  string opt="";
  for ( int i=0; i<20; i++ ) {
    myF->SetParameter(0, R->Gaus(0, 1, 1));
    myF->DrawClone( opt.c_str() );
    opt="same";
  }
}
```

Εκτέλεση του ROOT macro:

```
[user20 @ pc198 23] root -l mymacro3.C
root [0]
Processing mymacro3.C...
```

