Computational Physics Lab

Data Analysis Frameworks

03/19/2009

Outline

1 Homework Assignment

2 ROOT
Object Oriented Concepts
ROOT's Services and Utilities
ROOT Organization

3 Getting Started

4 Project 9
Part 1
Part 2
Part 3
Part 4

Homework Assignment

1 Read Chapter 10, 12 (7 pages), and 15 (5 pages)

  10 *"Arrays and matrices"*

  12 *"Numerical error analysis - derivatives"*

  15 *"Differential equations"*

2 Assignments of Section 10.10: (1) - (9)

• Due next Tuesday, March 24

➔ Hand in a paper copy (preferred!) or a piece of
paper stating that you have posted the homework
to your `comphy` web site!

Computational
Physics Lab

Homework
Assignment

ROOT
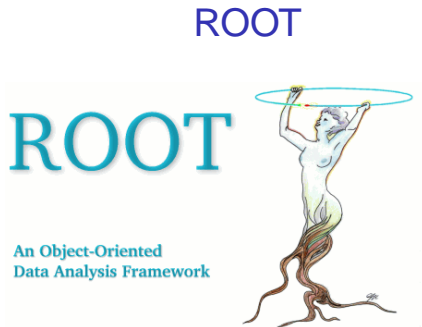
Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

ROOT

`http://root.cern.ch`

ROOT is an object-oriented framework which is aimed at solving the data analysis challenges of todays high-demand computing in physics.

Mission Statement:

We believe that ROOT is an ideal environment to introduce physicists quickly to the new world of Objects and C++.

Computational
Physics Lab

Homework
Assignment

ROOT

Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

Object Oriented Concepts

Members:
a *has a* relationship
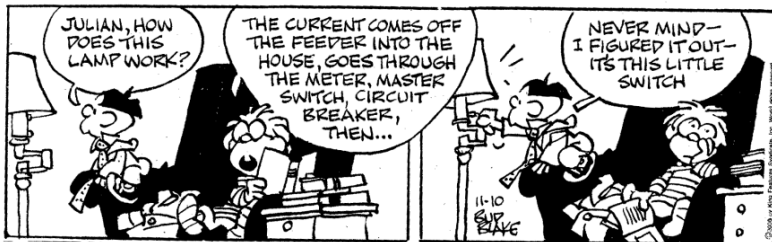to the class

Inheritance:
an *is a* relationship
to the class
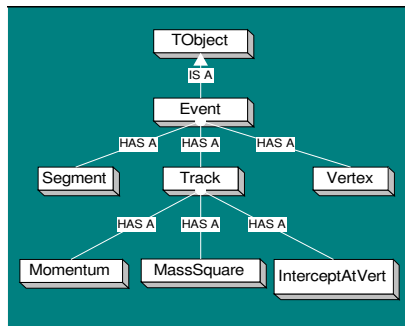
- Class: The description of a *thing* in the system
- Object: Instance of a class
- Methods: Functions for a class

Framework
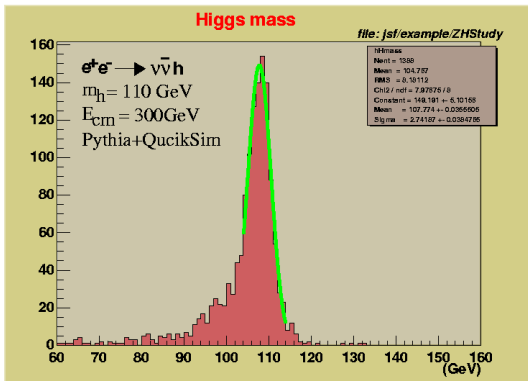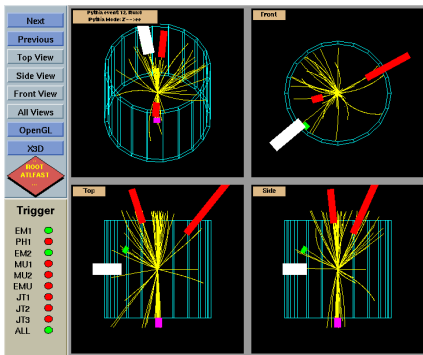
ROOT provides utilities and services.

Services and Utilities

- Histogramming and Fitting

Invariant Mass: $m^2 = E^2 - p^2$

Services and Utilities

- Histogramming and Fitting

- Graphics (2D, 3D)

Computational
Physics Lab

Homework
Assignment

ROOT
Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

Services and Utilities

- Histogramming and Fitting

- Graphics (2D, 3D)

- I/O to file or socket
  → Specialized for histograms, Ntuples (Tree)

- Collection classes and run-time type identification

- User Interface

  1. GUI: Browsers, Panels, Tree Viewer

  2. Command line interface: C++ interpreter CINT

  3. Script Processor (C++ compiled or C++ interpreted)

Three User Interfaces

- GUI:
  Windows, Buttons, Menus

- ROOT Command Line:
  CINT (C++ Interpreter)

- Macros, applications,
  and libraries
  - ➜ C++ compiler and
    C++ interpreter

ROOT Organization

$ROOTSYS

bin | lib | tutorials | test | include

cint
makecint
new
proofd
proofserv
rmkdepend
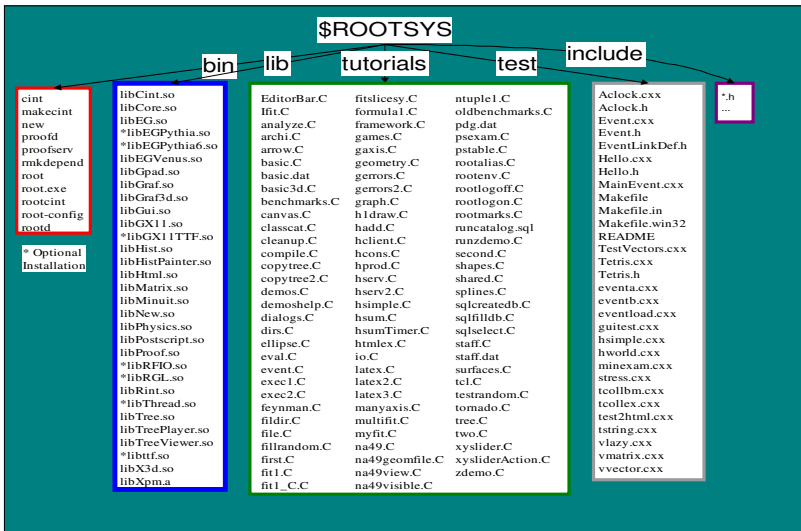root
root.exe
rootcint
root-config
rootd

* Optional
Installation

libCint.so
libCore.so
libEG.so
*libEGPythia.so
*libEGPythia6.so
libEGVenus.so
libGpad.so
libGraf.so
libGraf3d.so
libGui.so
libGX11.so
*libGX11TTF.so
libHist.so
libHistPainter.so
libHtml.so
libMatrix.so
libMinuit.so
libNew.so
libPhysics.so
libPostscript.so
libProof.so
*libRFIO.so
*libRGL.so
libRint.so
*libThread.so
libTree.so
libTreePlayer.so
libTreeViewer.so
*libttf.so
libX3d.so
libXpm.a

EditorBar.C
Ifit.C
analyze.C
archi.C
arrow.C
basic.C
basic.dat
basic3d.C
benchmarks.C
canvas.C
classcat.C
cleanup.C
compile.C
copytree.C
copytree2.C
demos.C
demoshelp.C
dialogs.C
dirs.C
ellipse.C
eval.C
event.C
exec1.C
exec2.C
feynman.C
fildir.C
file.C
fillrandom.C
first.C
fit1.C
fit1_C.C

fitslicesy.C
formula1.C
framework.C
games.C
gaxis.C
geometry.C
gerrors.C
gerrors2.C
graph.C
h1draw.C
hadd.C
hclient.C
hcons.C
hprod.C
hserv.C
hserv2.C
hsimple.C
hsum.C
hsumTimer.C
htmlex.C
io.C
latex.C
latex2.C
latex3.C
manyaxis.C
multifit.C
myfit.C
na49.C
na49geomfile.C
na49view.C
na49visible.C

ntuple1.C
oldbenchmarks.C
pdg.dat
psexam.C
pstable.C
rootalias.C
rootenv.C
rootlogoff.C
rootlogon.C
rootmarks.C
runcatalog.sql
runzdemo.C
second.C
shapes.C
shared.C
splines.C
sqlcreatedb.C
sqlfilldb.C
sqlselect.C
staff.C
staff.dat
surfaces.C
tcl.C
testrandom.C
tornado.C
tree.C
two.C
xyslider.C
xysliderAction.C
zdemo.C

Aclock.cxx
Aclock.h
Event.cxx
Event.h
EventLinkDef.h
Hello.cxx
Hello.h
MainEvent.cxx
Makefile
Makefile.in
Makefile.win32
README
TestVectors.cxx
Tetris.cxx
Tetris.h
eventa.cxx
eventb.cxx
eventload.cxx
guitest.cxx
hsimple.cxx
hworld.cxx
minexam.cxx
stress.cxx
tcollbm.cxx
tcollex.cxx
test2html.cxx
tstring.cxx
vlazy.cxx
vmatrix.cxx
vvector.cxx

*.h
...

Online Reference Guide

http://root.cern.ch/root/Reference.html

Index of PHYSICS classes

TFeldmanCousins ....
TGenPhaseSpace ....
TLorentzRotation ....
TLorentzVector ....
TRotation ....
TVector2 ....
TVector3 ....

TLorentzVector

class description - source file - inheritance tree

class **TLorentzVector** : public **TObject**

        public:

        TLorentzVector TLorentzVector(Double_t x = 0.0,
Double_t y = 0.0, Double_t z = 0.0, Double_t t = 0.0)
        TLorentzVector TLorentzVector(const Double_t* carray)
        TLorentzVector TLorentzVector(const Float_t* carray)
        TLorentzVector TLorentzVector(const TVector3
& vector3, Double_t t)
        TLorentzVector TLorentzVector(const TLorentzVector& l
        TLorentzVector operator+(const TLorentzVector& q) con
        TLorentzVector operator-(const TLorentzVector& q) con
        TLorentzVector operator-() const
        TLorentzVector operator*(Double_t a)
           virtual void ~TLorentzVector()
                Double_t Angle(const TVector3& v) const
                Double_t Beta() const
                    void Boost(Double_t, Double_t, Double_t)
                    void Boost(const TVector3& b)
                 TVector3 BoostVector() const
        static TClass* Class()
                Double_t CosTheta() const

Outline

1 **Homework Assignment**

2 **ROOT**

Object Oriented Concepts
ROOT's Services and Utilities
ROOT Organization

3 **Getting Started**

4 **Project 9**

Part 1
Part 2
Part 3
Part 4

Getting Started

Environment

➜ Edit your `./.cshrc` file

Add these lines at the end of the file:

---

setenv ROOTSYS /export/home/skpark/root

setenv PATH ${PATH}:$ROOTSYS/bin

setenv LD_LIBRARY_PATH $ROOTSYS/lib

Check environment

Open new terminal window and type "root"
(to exit ROOT use: root[0] .q)

Outline

1 Homework Assignment

2 ROOT

Object Oriented Concepts
ROOT's Services and Utilities
ROOT Organization

3 Getting Started

4 Project 9

Part 1
Part 2
Part 3
Part 4

Project 9

1 Part 1
- Creating a ROOT macro
- Executing a macro with ROOT
  (Fitting a distribution with Breit-Wigner and Polynomial)
- Modifying a macro and improving graph formatting
- Interacting formatting and fitting
- Save image of graph

2 Part 2
- Repeat Part 1 using a 2-Gaussian fit function

3 Part 3
- Convert macro to stand-alone program

4 Part 4
- Document project on your Computational Physics web site
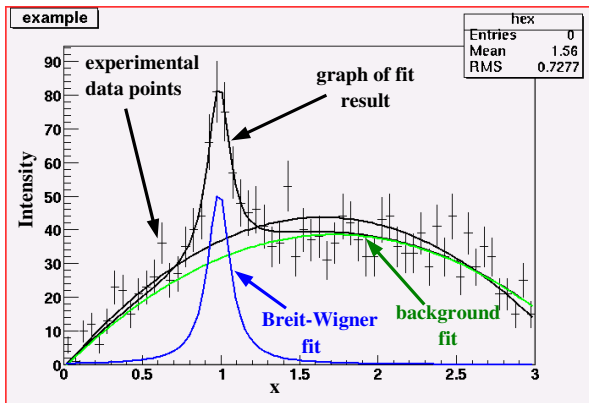
Histograms

Histograms are graphs representing statistical distributions

- Projections from large volumes of data
- Heights of the bars represent observed frequencies

Computational
Physics Lab

Homework
Assignment

ROOT
Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4
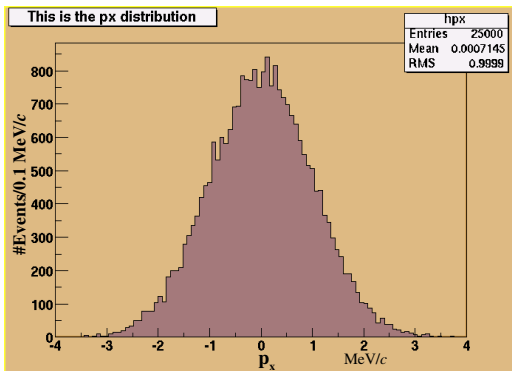
Fitting Histogram Data

Breit-Wigner Resonance plus a polynomial background

$$BreitWigner(x) = \frac{1}{2\pi} \frac{I_o \Gamma}{(x - M_0)^2 + 0.24 \Gamma_o^2}$$

$$Background(x) = a_o + b_o x + c_o x^2$$

parameters of the fit: a_o, b_o, c_o, I_o, M_o, Γ_o

Computational
Physics Lab

Homework
Assignment

ROOT
Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

Part 1: ROOT Macro Creation

Create a text file called "histoFit.C" with editor

Add the following user-defined functions:

```
double background(double *x, double *par) {
  return par[0] + par[1]*x[0] + par[2]*x[0]*x[0];
}

double breitWigner(double *x, double *par) {
  return ( 0.5*par[0]*par[1] / TMath::Pi() ) /
    TMath::Max( 1e-10, (x[0] − par[2]) * (x[0] − par[2]) +
    0.24*par[1]*par[1] ); }

double fitFunction(double *x, double *par) {
  return background(x, par) + breitWigner(x, &par[3]);
}
```

Computational
Physics Lab

Homework
Assignment

ROOT

Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9

Part 1
Part 2
Part 3
Part 4

Macro: Add a "doFit()" Function

```
void doFit() {    // Get the text for doFit() at ~/crede/root/doFit.txt
  const int nBins = 60;
  double data[60] = { 6, 1, 10, 12, 6, 13, 23, 22, 15, 21, 23, 26, 36,
          25, 27, 35, 40, 44, 66, 81, 75, 57, 48, 45, 46, 41, 35, 36,
          53, 32, 40, 37, 38, 31, 36, 44, 42, 37, 32, 32, 43, 44, 35,
          33, 33, 39, 29, 41, 32, 44, 26, 39, 29, 35, 32, 21, 21, 15,
          25, 15 };

  // Look up TH1F at http://root.cern.ch/root/Reference.html
  TH1F *histogram = new TH1F("histo","title", 60, 0, 3);

  for (int i = 0; i < nBins; i++) {
    histogram->SetBinContent(i+1, data[i]);
    histogram->SetBinError(i+1, TMath::Sqrt(data[i]));
  }

  // Look up TF1 at ROOT Refrence page
  TF1 *fitFcn = new TF1("fitFcn", fitFunction, 0, 3, 6);

  // Try fitting wo/ starting values for the parameters
  histogram->Fit("fitFcn");
}
```

Execute Macro w/ CINT/ROOT

gluon 25% root

```
************************************
*
*      W E L C O M E  to  R O O T
*
*   Version   5.14/00       14 Dec 2006
*
*   You are welcome to visit our Web site
*         http://root.cern.ch
*
************************************
```
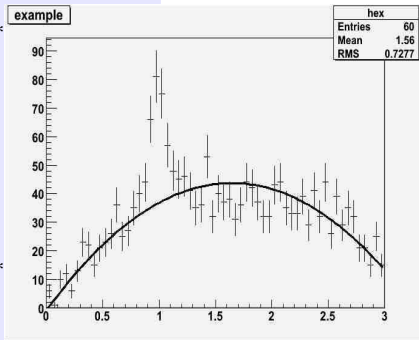

Compiled for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.16, Nov 2004
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .L histoFit.C
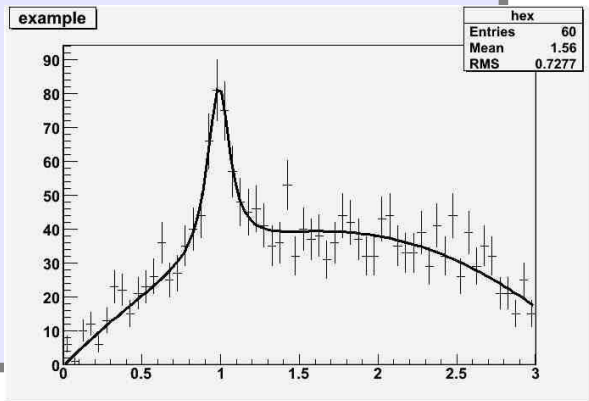root [1] doFit()

Macro: Improving the Fitting

```
// Add these commands to the doFit() function
// now set start values
fitFcn->SetParameter(4,0.2); //width
fitFcn->SetParameter(5,1);  // peak

histogram->Fit("fitFcn","V+");
```
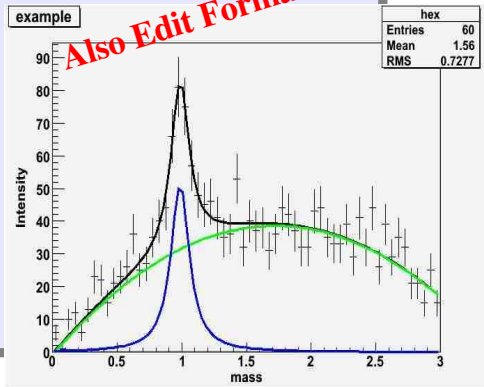
Macro: Improving the Graph

```
// Add these commands to the end of the doFit() function
// improving the figure
TF1* backFcn = new TF1("backFcn",
                       background,0,3,3);
backFcn->SetLineColor(3);
TF1* bwFcn = new TF1("bwFcn", brietWigner,0,3,3);
bwFcn->SetLineColor(4);

double par[6];
fitFcn->GetParameters(par);

backFcn->SetParameters(par);
backFcn->Draw("same");

bwFcn->SetParameters(&par[3]);
bwFcn->Draw("same");
```
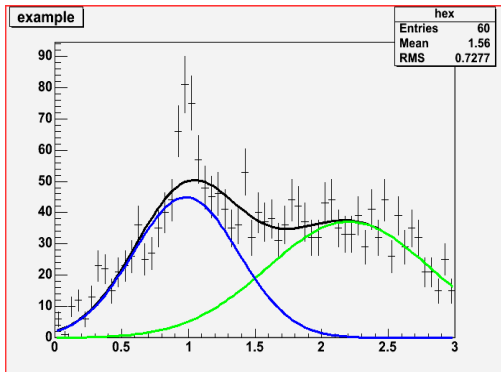

Also Edit Format Interactively!

Computational
Physics Lab

Homework
Assignment

ROOT

Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

Part 2: Fit Histogram to Gaussian plus Gaussian

$$G(A, \sigma, \overline{x}) = \frac{A}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\overline{x})^2}{2\sigma^2}}$$

Computational
Physics Lab

Homework
Assignment

ROOT
Object Oriented
Concepts
ROOT's Services
and Utilities
ROOT Organization

Getting
Started

Project 9
Part 1
Part 2
Part 3
Part 4

Part 3: Making a Stand-Alone Application

Get the Makefile

➜ Type: cp /export/home/crede/root/Makefile .

Create fit.cc source

1. Type: cp histoFit.C fit.cc
2. Add includes & main() function
3. Modify source code to adhere to coding conventions

Add at Beginning:	Add at End:

#include "TApplication.h"
#include "TH1.h"
#include "TF1.h"
#include "TMath.h"
#include <iostream>
 	
using namespace std;
	int main(int argc, char **argv) {
	  TApplication theApp("App", &argc, argv);
	  doFit();
	  theApp.Run();
	}

Part 4: Document Project 9

Post Project 9 to your computational physics web site. Create a `html` page for this project. Create a link from your main project web page to this `html` page. It should include the following heading information: project title, project number, your name, & today's date.

The main content of this page should include the following:

- A short description of the project
- A link to the macro code for part 2
- A text region which contains the actual macro code for part 2
- An image for a formatted plot for part 3
- A link to the stand-alone code for part 3
- A text region which contains the course code for part 3

For text regions use the `html` object tag; example:

<object width="600" height="400" type="text/plain"
                data="yourProgram.cc" border="0"></object>