

粒子物理与核物理实验中的数据 分析

王喆 杨振伟
清华大学

第三讲：ROOT在数据分析中的应用(1)

之前几讲的要点

■ C++基本概念

类的定义与实现，指针

■ Linux下用g++编译C++程序

`g++ -o hello.exe -I<include> ./src/*.cc`

当前目录下输出
可执行文件hello.exe

指定include目录
如-I./include

源文件

■ 用makefile进行C++编译

`make` 进行编译

`make clean` 清除编译结果

本讲要点

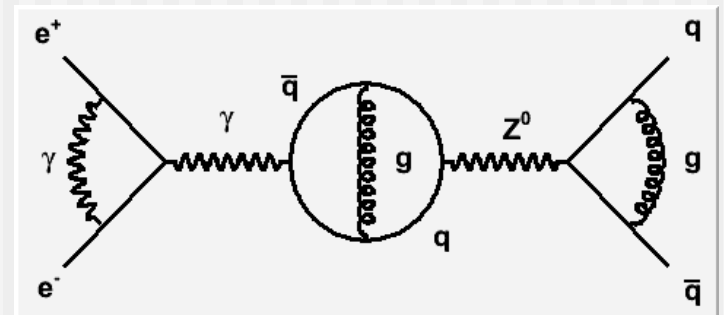
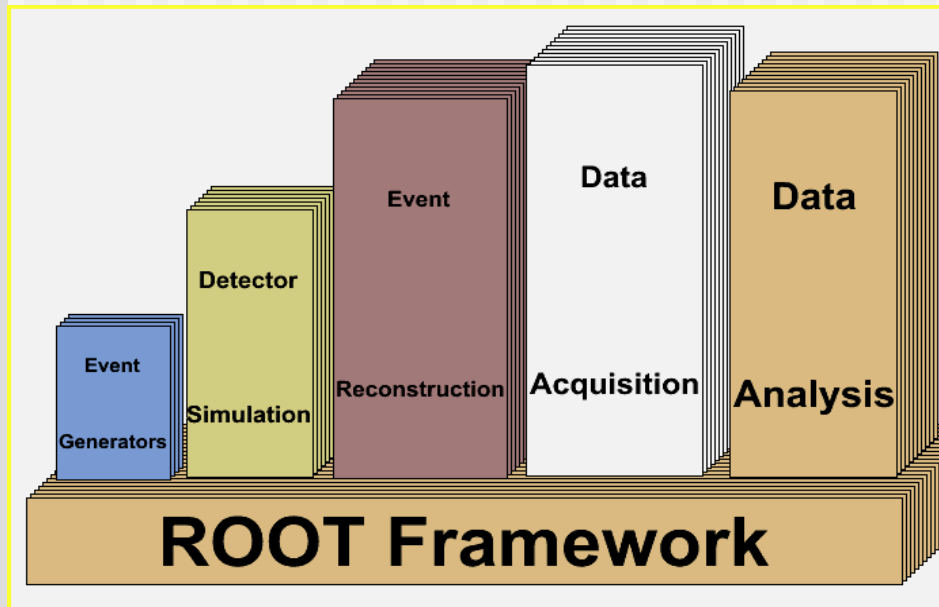
- 什么是ROOT?
- ROOT体验中心
- 安装登录ROOT环境
- Tutorial 目录
- ROOT的常用指令
- ROOT的结构、语法简介
- ROOT的随机数
- ROOT的函数
- ROOT直方图
- ROOT的文件操作

什么是 ROOT ?

ROOT: Executive Summary

... provides a set of **OO frameworks** with **all the functionality** needed to handle and analyse **large amounts of data** in a **very efficient** way....

关键字：面向对象的框架、所有功能、海量数据、非常有效



结论：很不谦虚！

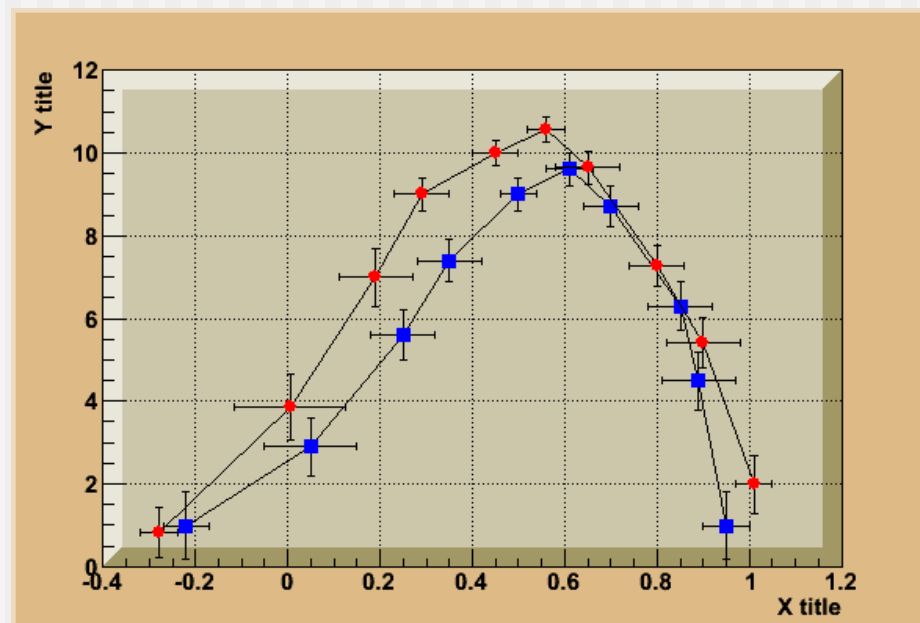
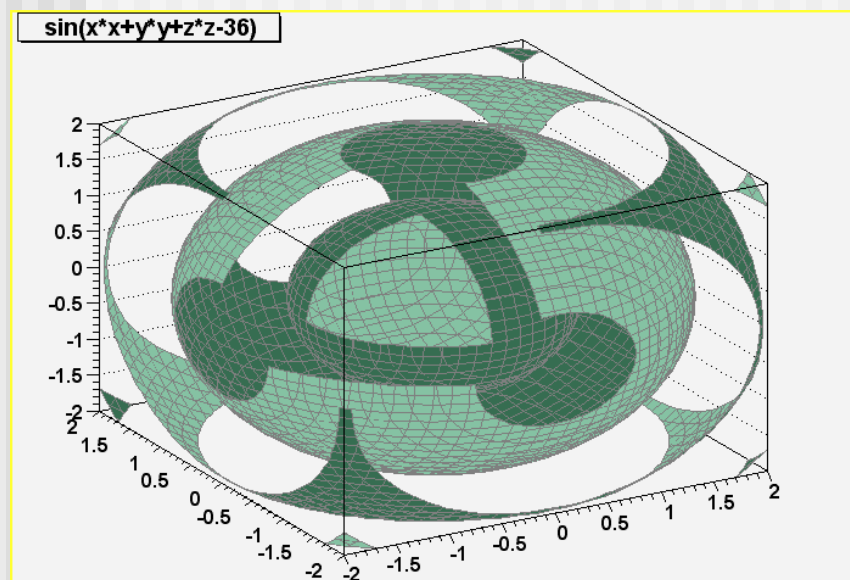
ROOT体验中心

还可以在ROOT网站上看到一些ROOT图片:

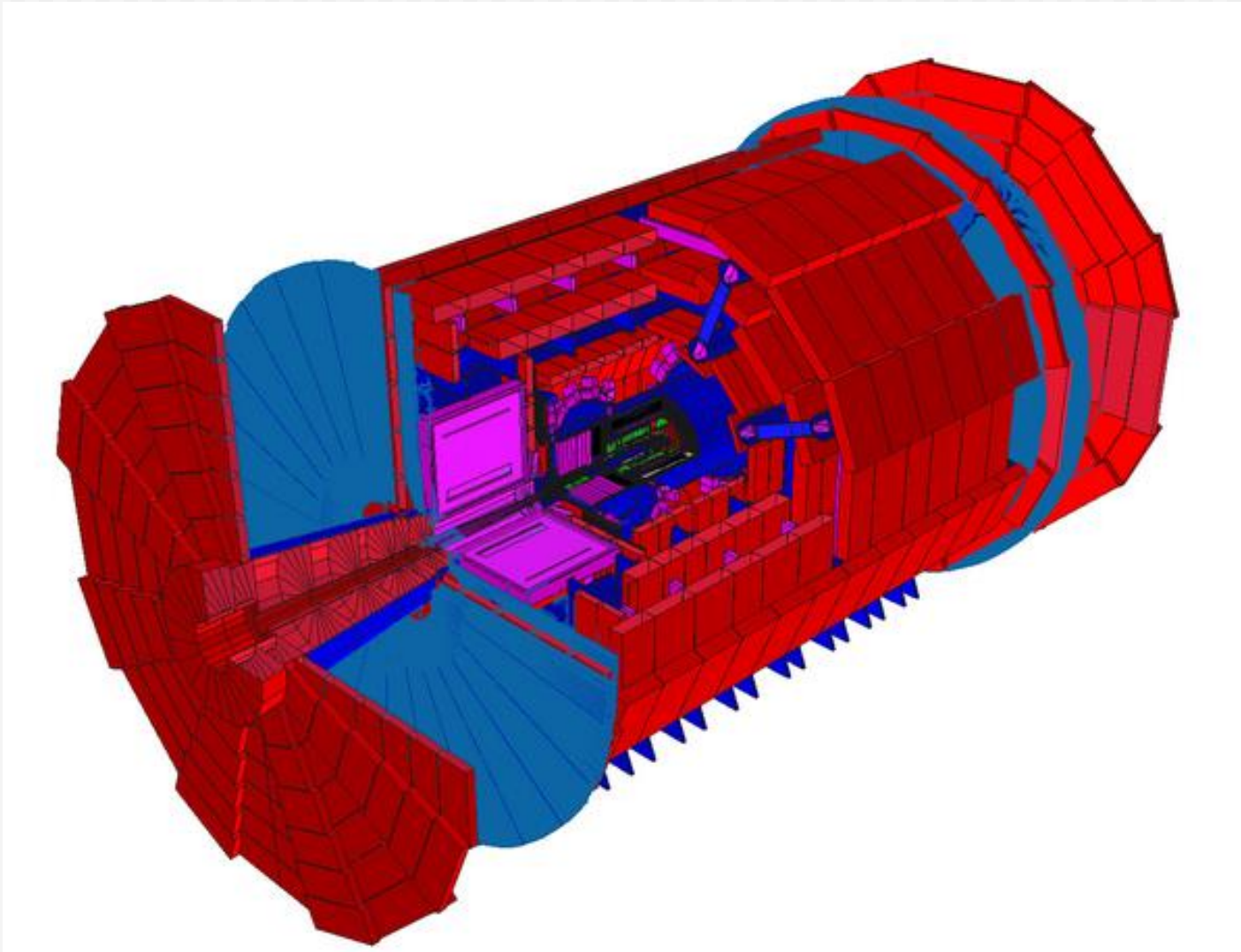
<https://root.cern.ch/gallery>

当然, **ROOT**的功能不只是做图, 它不是一个作图工具。
跟数据分析有关的东西, 基本都是**ROOT**的擅长;
跟物理有关的很多东西, **ROOT**基本都可以做得很好:

事例产生、探测器模拟、事例重建、数据采集、**数据分析**

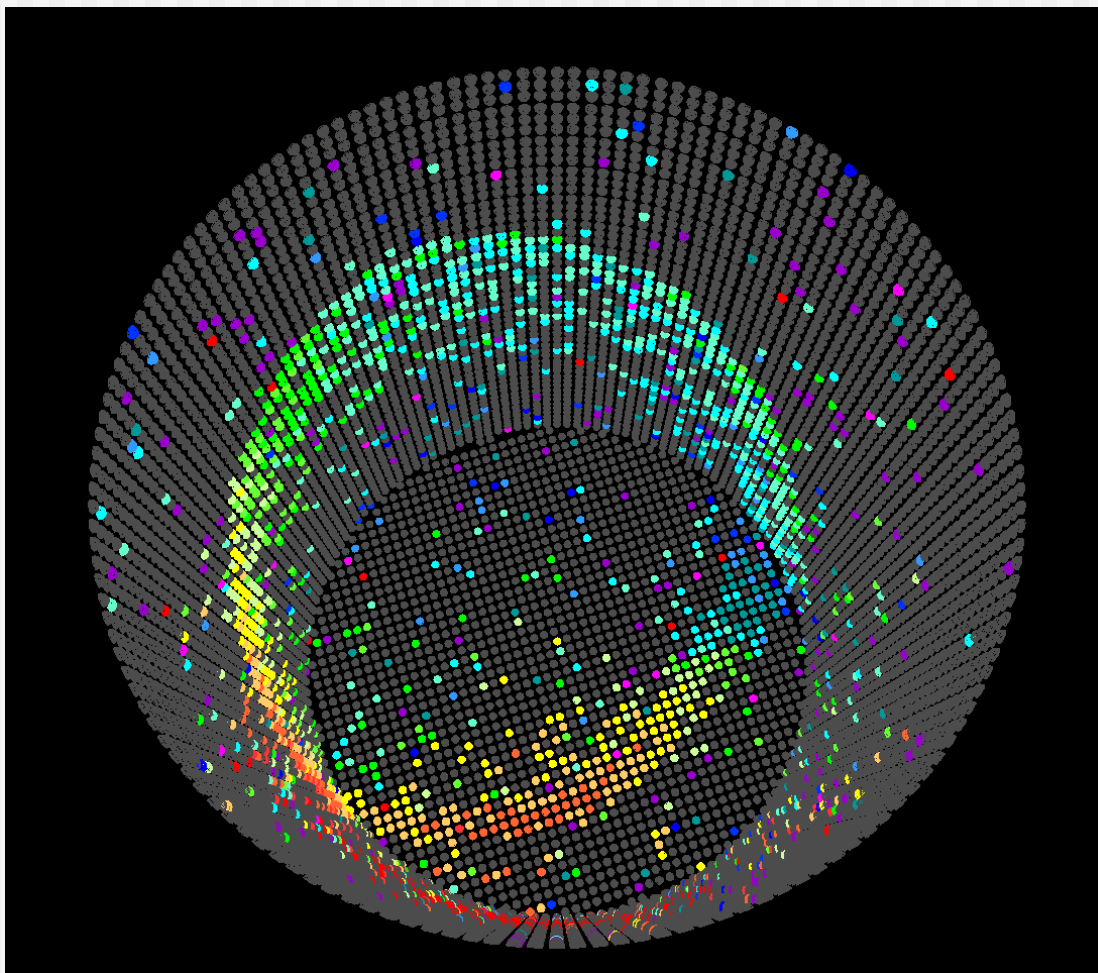


ATLAS

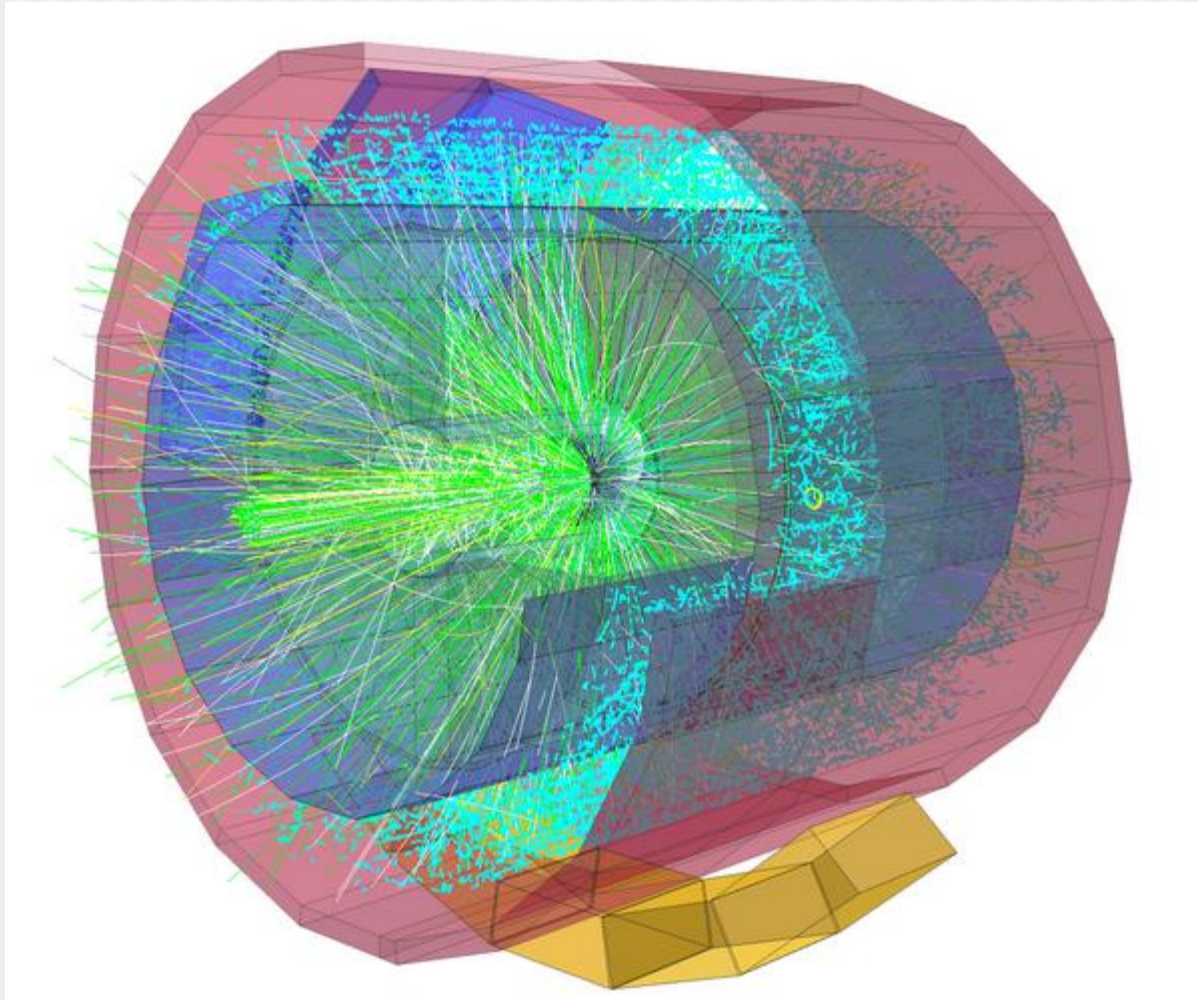


超级神冈

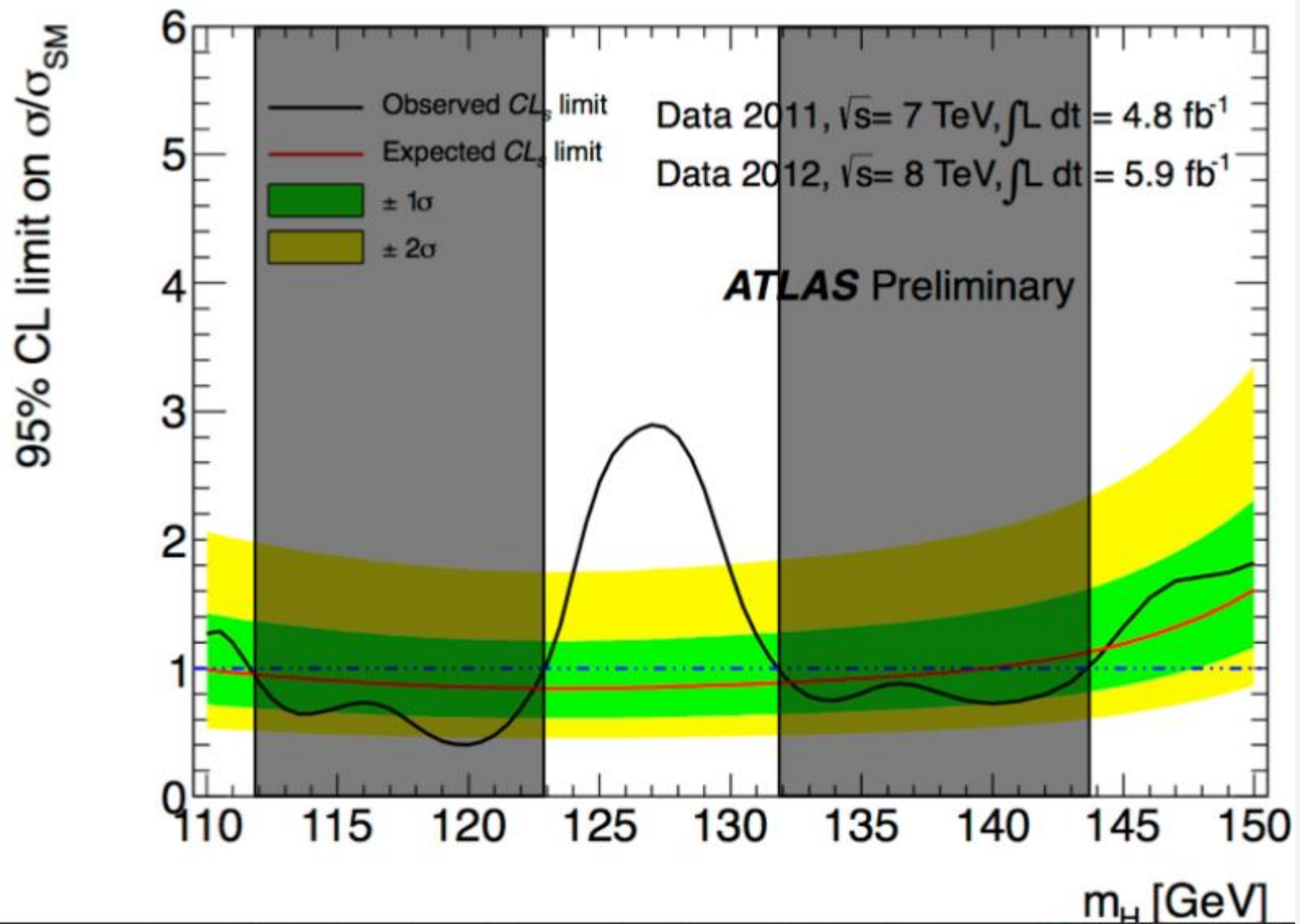
日本超级神冈中微子实验事例显示
超大的水池，内外装满了光电倍增管，1万多个



ALICE (simulated Pb-Pb collision)



Higgs search



安装ROOT(1)

到ROOT主页下载需要的版本到指定目录。

(<http://root.cern.ch/drupal/content/downloading-root>)

下载源代码:

<https://root.cern.ch/downloading-root>

由源代码安装: (基本过程, 订制安装可能需要一些参数)

`tar xvfz *.tar.gz`, 假定解压后的文件在root目录中,

`> cd root`

`> ./configure [--prefix=/home/wangzhe/root (安装目录)]`

`> make`

(自行安装的时候有很多麻烦, 主要是缺少一些依赖的软件, 你的linux系统总包含一些软件库、管理器之类, 很容易得到解决。)

使用:

`source bin/thisroot.sh (或者 .csh)`

(这个还可以加在你的登录环境中.bashrc或.cshrc, 自动加载)

`root`

(进入root)

安装ROOT(2)

如果是其它发行版的Linux，首先查看是否ROOT网站上是否有预编译好的程序包，一般情况下，官方提供SLC4和SLC5在各种不同CPU以及不同gcc版本下的二进制包，

ROOT官网也提供包括Solaris以及Mac OS X以及Windows下的预编译包。

Windows用户在官网下载相应的.msi文件直接安装即可。

（但是，由于windows缺省没有编译器，所以root的大部分功能都受到限制。有用，但很多局限，完成不了我们的作业）

一些linux操作系统近来把root当作了必包含的软件包，例如在ubuntu (synaptic), redhat (yum)的软件选择列表中就可以找到，省去了自行安装过程中要找到所依赖软件的麻烦。

登录ROOT环境

- 运行 `>root`
- 退出 `root[0] .q`
- 键入 `help` 指令, 如
`root[0]?`
`root[1].ls`
`root[2].!ls`

```
*****
*
*      W E L C O M E
*
*      Version  5.30/02 21
*
*      You are welcome to visit our web site
*      http://root.cern.ch
*
*****

ROOT 5.30/02 (tags/v5-30-02@40973, Sep 22 2011, 10:55:04 on win32)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```



Root的常用命令

其他常用命令

```
> root plot.C  
> root plot.C+  
> root plot.C++  
> root -l plot.C  
> root momentum.root
```

```
root[0] .x plot.C  
root[0] .x plot.C(1)  
root[0] .x plot.C+  
root[0] .x plot.C++
```

```
root[0] .L plot.C  
root[0] .L plot.C+  
root[0] .L plot.C++  
root[0] plot()
```

其他:

- `.ls` 显示ROOT当前环境的所有信息
- `!.ls` 显示Linux系统当前目录的所有信息
- 注: ROOT环境中, **ROOT**指令都以“.”开头, 系统指令都以“!. ”开头

```
1. > root myFunc.C
```

```
2. > root
```

```
    root [0] .x myFunc.C
```

```
3. > root
```

```
    root [0] .L myFunc.C
```

```
    root [1] myFunc()
```

```
4. > root
```

```
    root [0] .L myFunc.C+
```

```
    root [1] myFunc()
```

解释执行

编译执行

编译执行会执行最严格的语法检查，
程序运行最安全，推荐使用

Tutorial 目录

课上演示

在\$ROOTSYS/tutorials目录下，有五花八门的例子。
以后会经常与这个目录打交道。先尝试一下吧。

尝试方法：

```
>cd $HOME
```

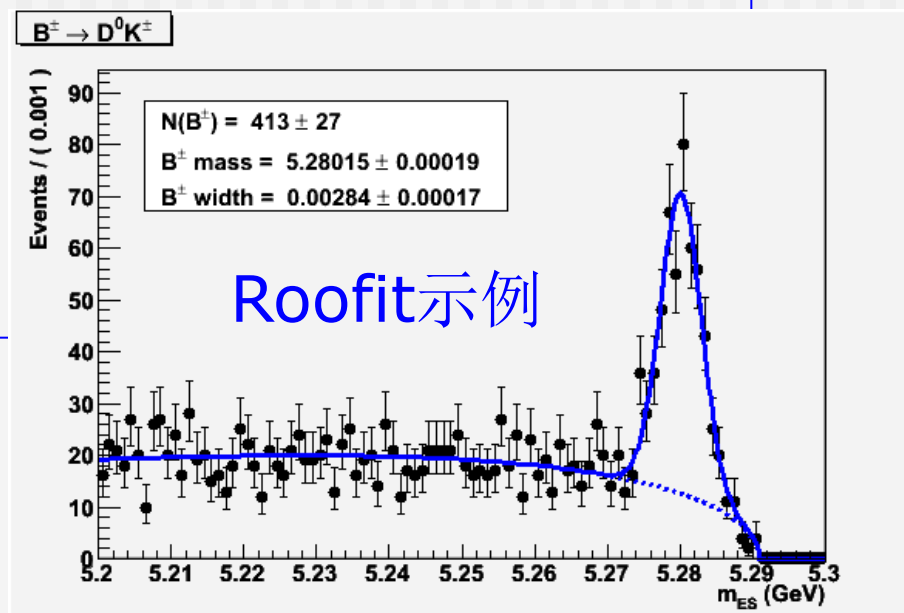
```
>cp -r $ROOTSYS/tutorials . (注意不要把这个"."漏掉了)
```

```
>cd tutorials
```

然后找个感兴趣的目录/文件，
执行ROOT脚本，比如

```
>cd roofit
```

```
>root -l RoofitDemo.C
```



小技巧提示：

根据关键字"xxxx"从tutorials的例子中寻找线索

```
grep -sirn "xxxx" $ROOTSYS/tutorials
```

比如找随机数用法:

```
grep -sirn "random" $ROOTSYS/tutorials
```


ROOT语法—基本信息

- ROOT使用C++语法

一段C++程序可以直接在ROOT环境运行

- 数据类型重定义

int → Int_t

float → Float_t

double → Double_t

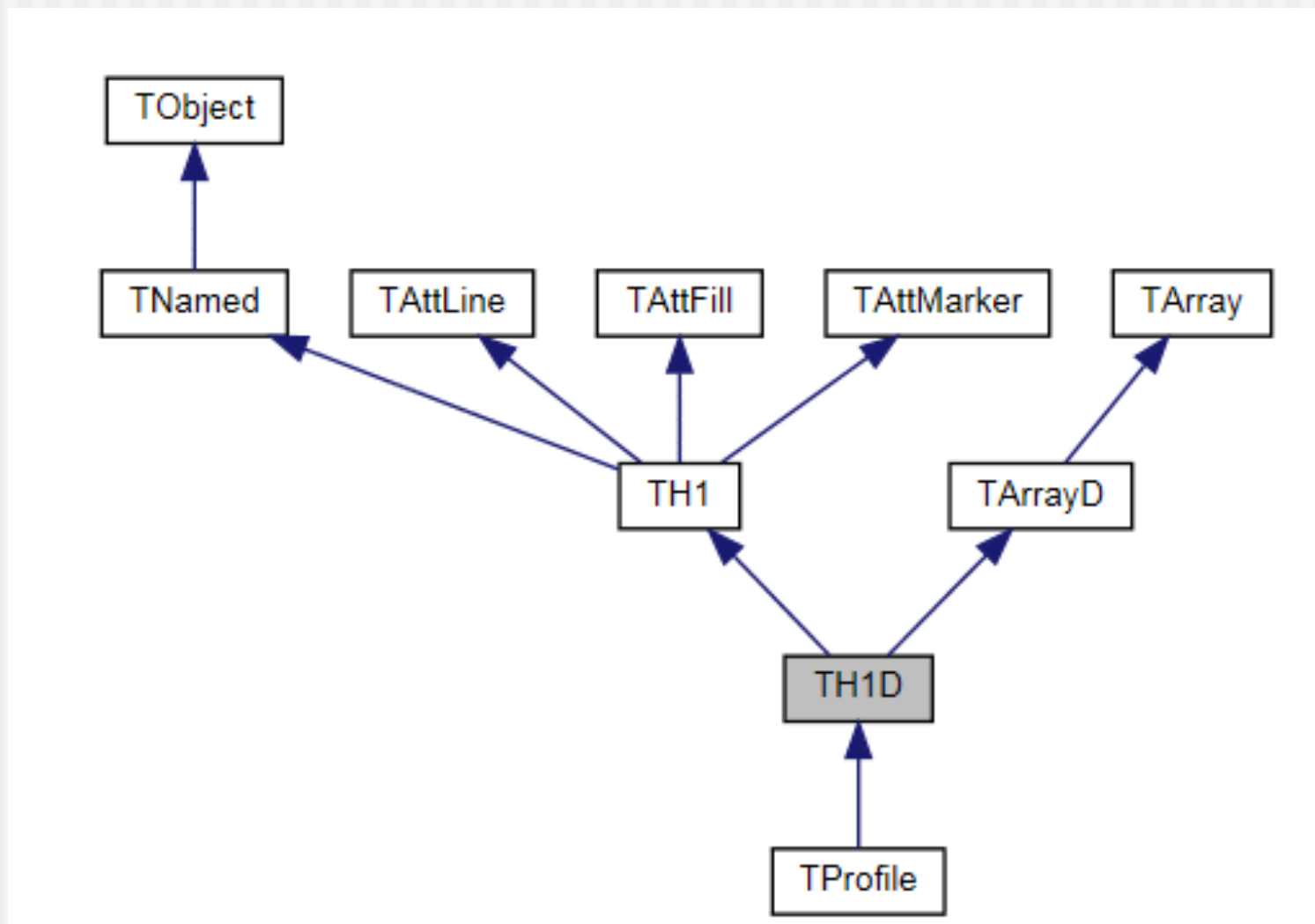
.....

- ROOT的类都以T开头

如TFile, TH1F, TTree, ...

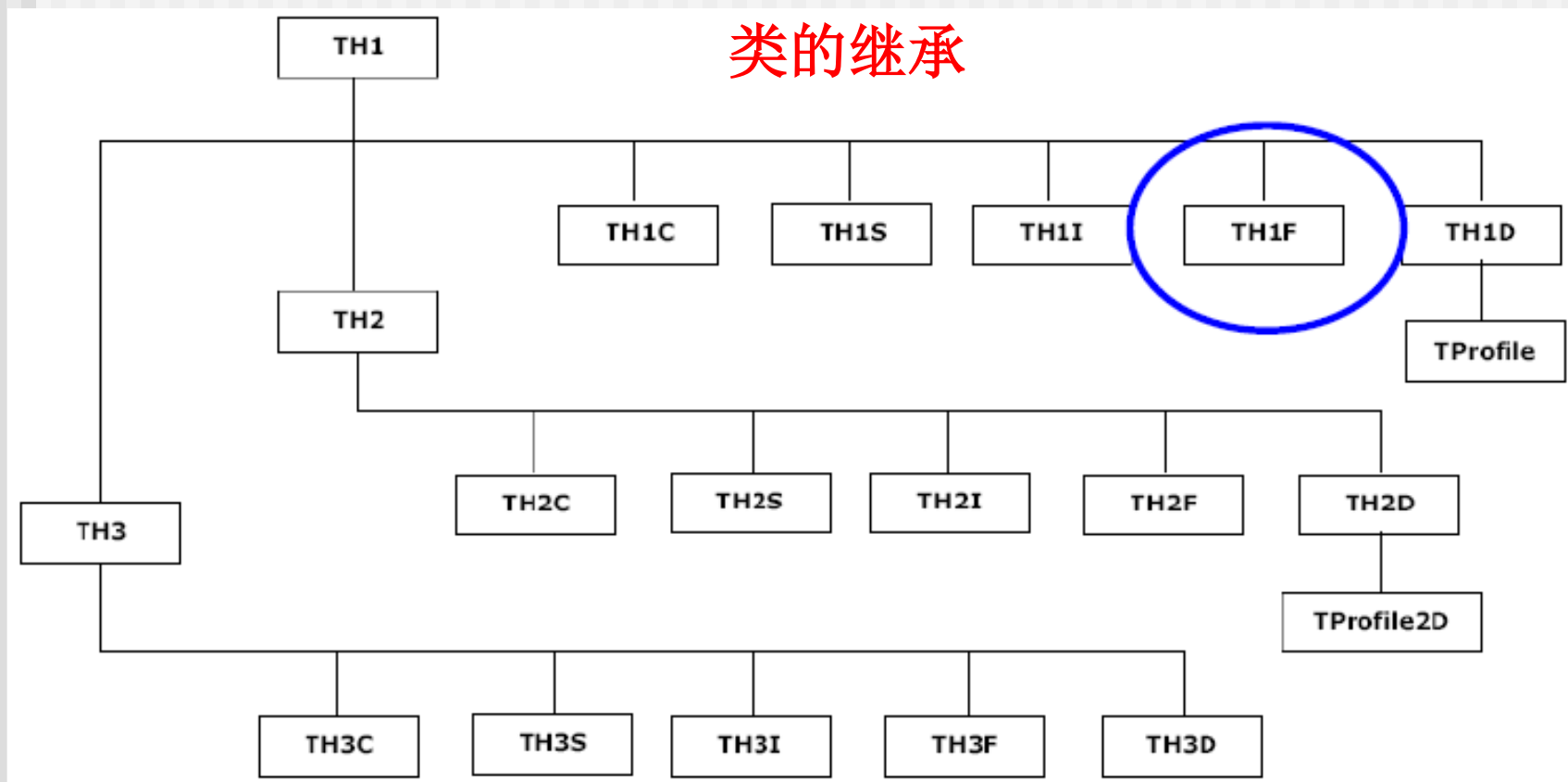
- 可以直接在ROOT环境中运行macro文件(自动调用cint编译器), 也可以在makefile中设置好相关参数用g++编译得到可执行文件运行。

Root的关于继承的例子



ROOT结构一类的的使用

ROOT中有众多已经定义好的类可供使用，
比如直方图家族



ROOT的一些重要的类

其它常用类

基础类: TObject

数学函数: TF1, TF2, TF3...

图 形: TGraph, TGraphErrors, TGraph2D,...

文 件: TFile

画 布: TCanvas, TPad, ...

随 机 数: TRandom, TRandom1, TRandom2, TRandom3

树, 树链: TTree, TChain (非常重要)

还有很多全局函数, 全局类, 全局变量, 多数以g开头, 如:

gRandom, gROOT, gStyle, gPad, gEnv, gFile...

Root的随机数

随 机 数: TRandom, TRandom1, TRandom2, TRandom3

周期	10^9	10^{171}	10^{26}	10^{6000}
速度(ns/call)	34	242	37	45

速度与**CPU**和编译器有关

gRandom是指向当前随机数产生子的指针,

该产生子默认**TRandom3**对象。

<http://root.cern.ch/root/html522/TRandom.html>

(为什么看TRandom? 因为TRandom1/2/3都继承自TRandom)

ROOT—随机数

<code>gRandom->Binomial(ntot, p):</code>	二项分布
<code>gRandom->BreiWigner(mean, gamma)</code>	Breit-Wigner分布
<code>gRandom->Exp(tau)</code>	指数分布
<code>gRandom->Gaus(mean, sigma)</code>	高斯分布
<code>gRandom->Integer(imax)</code>	(0,imax-1)随机整数
<code>gRandom->Landau(mean, sigma)</code>	Landau分布
<code>gRandom->Poisson(mean)</code>	泊松分布(返回int)
<code>gRandom->PoissonD(mean)</code>	泊松分布(返回double)
<code>gRandom->Rndm()</code>	(0,1]均匀分布
<code>gRandom->Uniform(x1,x2)</code>	(x1,x2]均匀分布
....	

思考：什么情况下需要PoissonD(mean)?

ROOT中的数学函数

画图时采用

```
root[1] fun_name.Draw();
```

□制作一维函数曲线图

```
TF1 *fun_name = new TF1("fun_name","expression",  
x_low,x_high);
```

```
root[0]TF1 *f1 = new TF1("f1","x*sin(x)",-5,5);
```

□制作二维函数曲线图

```
TF2 *fun_name = new TF2("fun_name","expression",  
x_low,x_high, y_low,y_high);
```

```
root[0]TF2 *f2 = new TF2("f2","x*sin(x)+y*cos(y)",  
-5,5,-10,10);
```

□制作三维函数曲线图

```
TF3 *fun_name = new TF3("fun_name","expression",  
x_low,x_high,y_low,y_high,z_low,z_high);
```

```
root[0]TF3 *f3 = new TF2("f3","x*sin(x)+y*cos(y)  
+z*exp(z)",-5,5,-10,10,-20,20);
```


数学函数的定义方式(1)

ROOT中定义数学函数的方式多种多样

❑ 利用C++数学表达式

```
TF1* f1 = new TF1("f1", "sin(x)/x", 0, 10);
```

❑ 利用TMath定义的函数

```
TF1 *f1 = new TF1("f1", "TMath::DiLog(x)", 0, 10);
```

❑ 利用自定义C++数学函数

```
Double_t myFun(x) {  
    return x+sqrt(x);  
}  
TF1* f1 = new TF1("f1", "myFun(x)", 0, 10);
```

以上函数都不含参数，但在数据拟合时，我们往往需要定义含未知参数的函数

数学函数的定义方式(2)

ROOT中定义含未知参数的数学函数

❑ ROOT已经预定义了几种常用的含参函数

gaus: 3个参数

$$f(x) = p_0 * \exp(-0.5 * ((x - p_1) / p_2)^2)$$

expo: 2个参数

$$f(x) = \exp(p_0 + p_1 * x)$$

polN: N+1个参数

$$f(x) = p_0 + p_1 * x + p_2 * x^2 + \dots$$

其中 $N=0, 1, 2, \dots$ ，使用时根据需要用 $pol0, pol1, pol2 \dots$

landau: 3个参数

朗道分布，没有解析表达式

这些预定义函数可直接使用，比如

```
histogram->Fit("gaus"); //对直方图进行高斯拟合  
TF1 *f1=new TF1("f1","gaus",-5,5);
```

数学函数的定义方式(3)

ROOT中自定义含未知参数的数学函数

❑ 利用C++数学表达式

```
TF1* f1 = new TF1("f1", "[0]*sin([1]*x)/x", 0, 10);
```

❑ 利用C++数学表达式以及ROOT预定义函数

```
TF1* f1 = new TF1("f1", "gaus(0)+[3]*x", 0, 3);
```

❑ 利用自定义的C++数学函数

```
Double_t myFun(Double_t *x, Double_t *par) {  
    Double_t f=par[0]*exp(-x[0]/par[1]);  
    return f;  
}
```

指定参数数目

```
TF1* f1 = new TF1("f1", "myFun", 0, 10, 2);  
f1->SetParameter(0, 100);  
f1->SetParameter(1, 2);
```

ROOT脚本文件示例：Macro文件

课上练习

`#include "TF1.h"` ← 非编译运行情况下，非必须的

```
Double_t DiExp(Double_t *x, Double_t *par)
{
    Double_t f=par[0]*exp(-x[0]/par[1]);
    return f;
}
```

```
void myFunc() { ← 解释执行的条件下，要与文件名同名
    TF1* f1 = new TF1("f1",DiExp,0,10,2);
    f1->SetParameter(0,100);
    f1->SetParameter(1,2);
    f1->Draw();
}
```

ROOT中统计直方图

❑ 定制一维直方图

```
TH1F *hist_name = new TH1F("hist_name", "hist_title",  
num_bins, x_low, x_high);
```

❑ 定制二维图

```
TH2F *hist_name = new TH2F("hist_name", "hist_title",  
num_bins_x, x_low, x_high, num_bins_y, y_low, y_high);
```

❑ 定制三维图

```
TH3F *hist_name = new TH3F("hist_name", "hist_title",  
num_bins_x, x_low, x_high, num_bins_y, y_low, y_high,  
num_bins_z, z_low, z_high);
```

❑ 填充统计图

```
hist_name.Fill(x);  
hist_name.Fill(x, y);  
Hist_name.Fill(x, y, z);
```

绘图:

```
root[0]hist_name.Draw();
```

ROOT脚本文件示例:直方图, 随机数

```
void histw() {  
    gROOT->Reset();  
    const Int_t NEntry = 10000 ;  
    TFile *file = new TFile("Landau.root","RECREATE");  
    TH1F *h1 = new TH1F("h1","A simple histo",100,0,1);  
    for (int i=0;i<NEntry;i++) {  
        h1->Fill( gRandom->Landau(0.2,0.1) );  
    }  
    h1->GetXaxis()->SetTitle("X Title,  $\sqrt{\Delta_2}$ ");  
    h1->GetXaxis()->CenterTitle();  
    h1->GetYaxis()->SetTitle("Y Title, Entries/0.01");  
    h1->Draw();  
    file->Write();  
    c1->SaveAs("Landau.ps");  
}
```

1. 文件准备, 写入
2. 生成直方图
3. 填图
4. 生成图片

打开root文件

1. 打开已有的root文件，如刚刚生成的Landau.root:
终端提示行下:

```
> root -l Landau.root
```

2. ROOT环境下:

```
> root
```

```
root [0] TFile f1("hist1.root");
```

```
root [1] .ls
```

```
root [2] h1->Draw();
```

3. 利用TBrowser 课上练习

```
> root
```

```
root [0] TBrowser b
```

```
利用你的鼠标 .....
```

4. 当然也可以利用C++代码执行

打开一个文件，查看一个直方图

```
//  
// open a root file and get a histogram  
//  
void histr() {  
  
    TFile *file = new TFile("Landau.root", "READ");  
    TH1F *h1 = (TH1F*) file->Get("h1");  
    h1->Draw();  
}
```

1. 文件打开
2. 得到直方图的指针
3. 画图

很多情况下，这些命令可以在**root**的提示符下逐行键入，并进行测试。

小结

- ROOT简介

C++，面向对象，实验数据处理的强大工具

- 安装与登录以及体验

- 运行ROOT脚本

- 数学函数，直方图，随机数，

- 新建root文件，查看root文件

练习

1. 把**Tutorial**目录拷贝到自己目录下，“看一看”！
2. 完成课上的例子
3. 自己生成两维的直方图，利用随机数生成两维的高斯分布，存成文件，然后自己用两种以上的方法打开，重新查看。
4. 继续熟悉**emacs**或者**vi**，**C++**的概念等等

参考资料

- ROOT手册第2章, 第3章
- <http://root.cern.ch>
- <http://root.cern.ch/root/Reference.html>
- <http://root.cern.ch/root/Tutorials.html>
- <http://root.cern.ch/root/HowTo.html>
- \$ROOTSYS/tutorials中的各个例子
- <http://hep.tsinghua.edu.cn/training/index.html> (一个三段的视频教程)