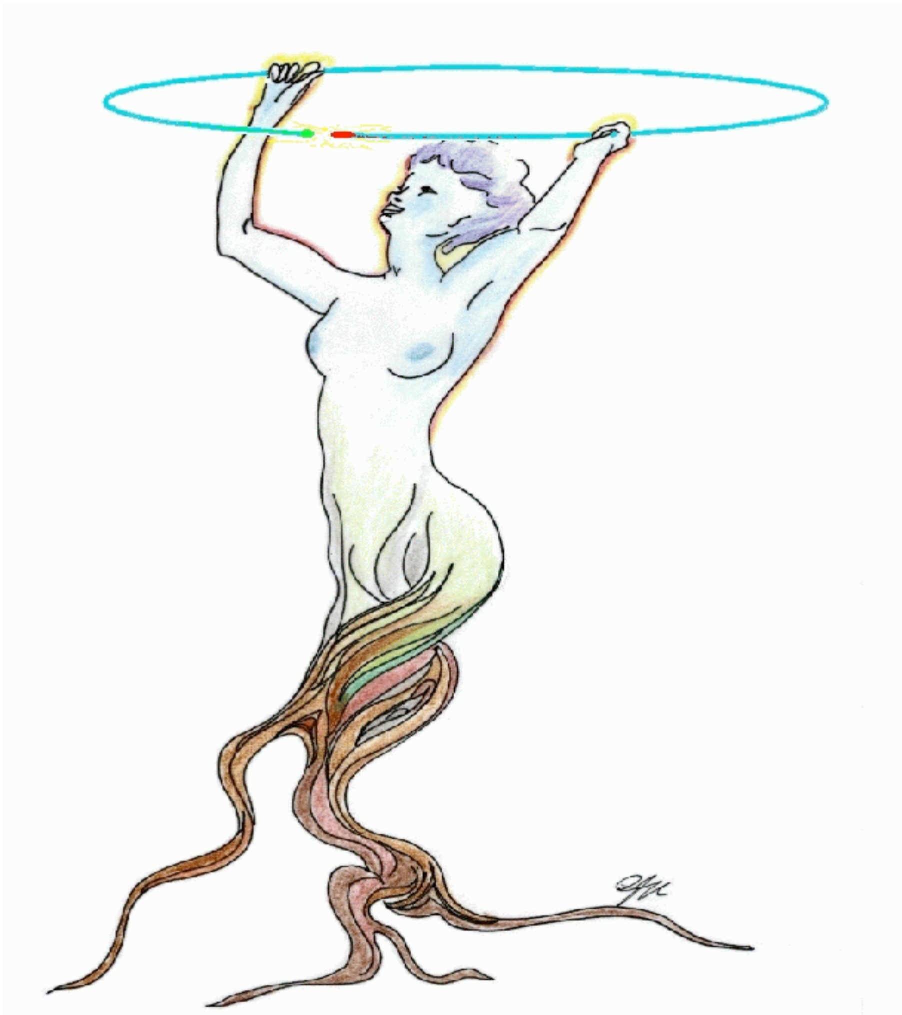
ROOT for beginners

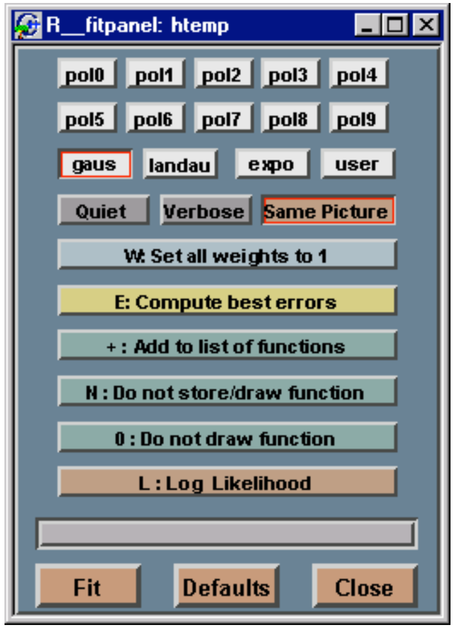
Third day

Data Fitting



Fits

- We know how to make a fit by using the graphical interface...
- How to fit with the command line? Or within a script ?



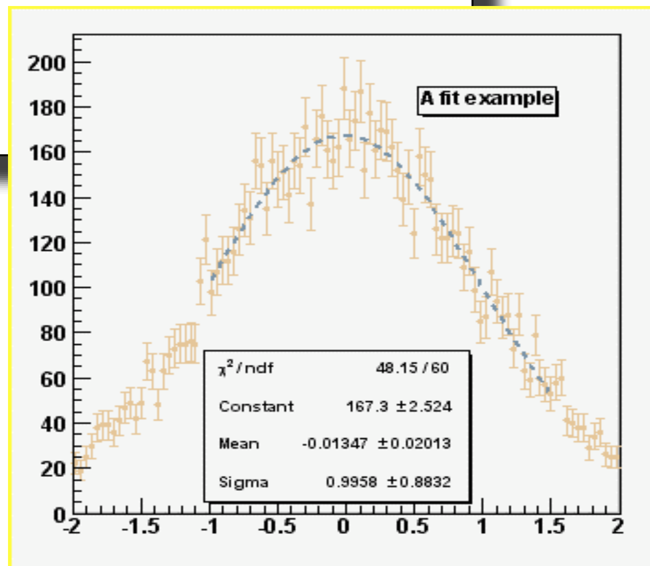

```
gStyle->SetOptFit(kTRUE)
TH1F *h = new TH1F("hg","Un exemple de fit",100,-2,2)
h->FillRandom("gaus",10000)
h->Fit("gaus","V","E1",-1,1.5)
```

function name

fit options

drawing options

fit limits

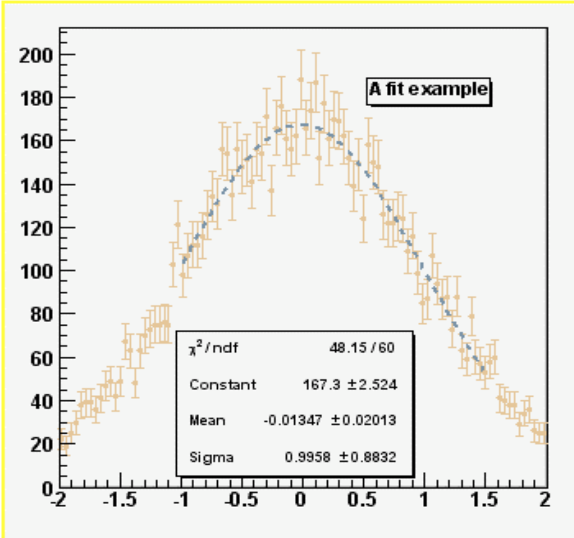


Basic fits…

Which fitting functions ?

- The predefined functions:
  - "gaus" = `p0*exp(-0.5*pow((x-p1)/p2),2)`
  - "expo" = `exp(p0+p1*x)`
  - "polN" = `p0 + p1*x + p2*pow(x,2) + p3*...`
  - "landau" (guess the formula!)
- How to obtain the values of the fit parameters ?

```
TF1 *gfit = (TF1 *)h->GetFunction("gaus")
gfit->GetParameter(0)
gfit->GetParameter(1) ...
gfit->GetParError(0) ...
double par[3]
gfit->GetParameters(par)
```

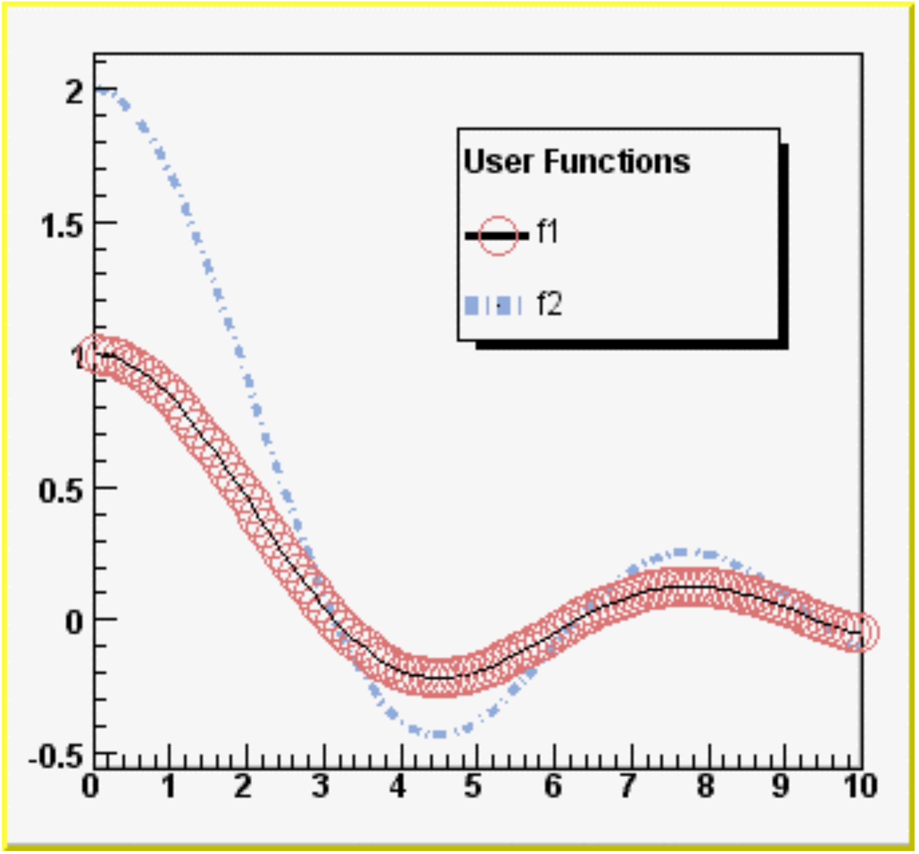


Creating a user defined function

```
TF1 *fu = new TF1("f1", "sin(x)/x", 0, 10)
TF1 *fd = new TF1("f2", "f1 * 2", 0, 10)
fu->Draw()
fd->Draw("same")
```

Only the function name is known!

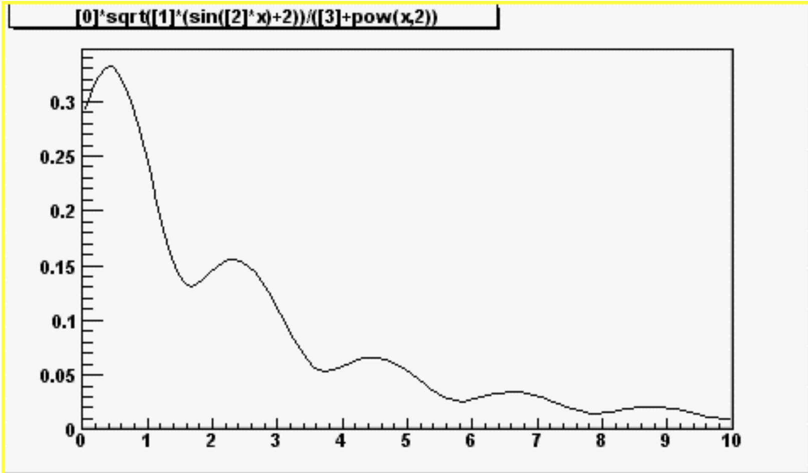
And many other combinations !



Including parameters

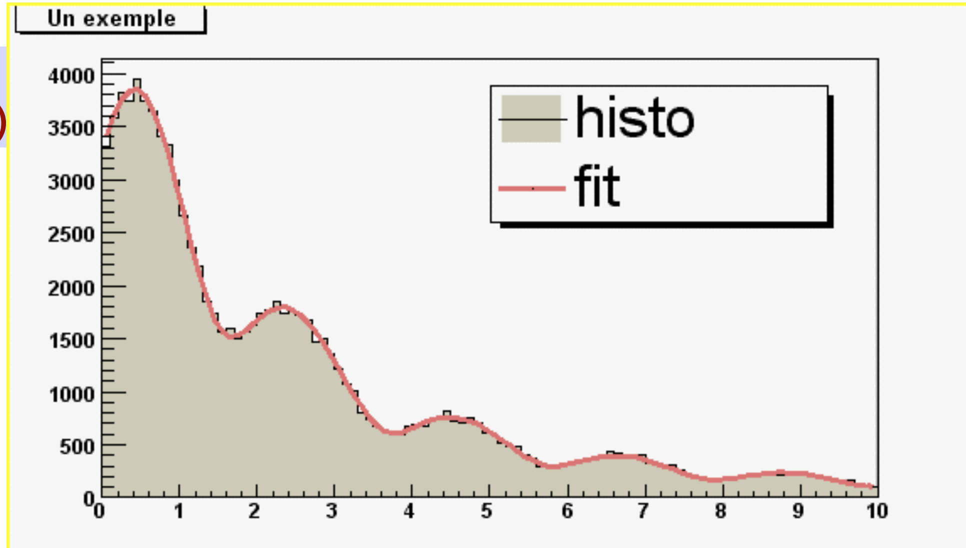
```
TF1 *ft = new TF1("f3", "[0]*sqrt([1]*(sin([2]*x)+2))
   /([3]+pow(x,2))", 0, 10)
ft->SetParameters(1,1,3,5)
ft->Draw()
```

index	0	1	2	3
content	1	1	3	5




```
TH1F *hd = new TH1F("h2","Un exemple", 100, 0, 10)
hd->FillRandom("f3",100000)
ft->SetParameters
    (h2->GetMaximum(),1,2.8,6.)
hd->Fit("f3")
```

index	0	1	2	3
content	h2->GetMaximum()	1	2.8	6



Mixing functions

- Predefined functions can be mixed

```
TF1 *fq=new TF1("f4","gaus(2)+expo(0)",0,10)
```

Cst	X_0	σ	A	B

- Another example

```
TF1 *fc=new TF1("f5","pol3(0)+[4]*sin(gaus(5)+[8])",0,10)
```

P0	P1	P2	P3	Amp	Cst	X_0	σ	φ

Mixing functions

- Predefined functions can be mixed

```
TF1 *fq=new TF1("f4","gaus(2)+expo(0)",0,10)
```

A	B	Cst	X_0	σ
0	1	**2**	3	4

- Another example

```
TF1 *fc=new TF1("f5","pol3(0)+[4]*sin(gaus(5)+[8])",0,10)
```

P0	P1	P2	P3	Amp	Cst	X_0	σ	φ
0	1	2	3	**4**	**5**	6	7	**8**

Advanced fits

A complex fitting example

- Fitting a spectrum with a Maxwellian function:

  3 steps:

  - Step 1: Define the function
  - Step 2: Include it in a TF1
  - Step 3: Make the fit

http://caeinfo.in2p3.fr/root/Formation/en/Day3/FitMaxwell.root

Step 1: define the function

http://caeinfo.in2p3.fr/root/Formation/en/Day3/Maxwell.C

```
//
// Maxwell fitting function
//
#include “TMath.h”

Double_t Maxwell(Double_t *x, Double_t *par)
{
if(x[0] > par[1] && par[2] > 0 && par[0] > 0)
 {
 return par[0]*(x[0]-par[1])/par[2]*
        TMath::Exp(-(x[0]-par[1])/par[2]);
 }
else
 {
 return 0.;
 }
}
```

Arguments array

Parameters array

(E-B)/T*exp(-(E-B)/T)

x	E
	0

par	Cst	B	T
	0	1	2

Step 2: include the function in ROOT

```
root[0] .L Maxwell.C+
```
Compilation and loading of the function

```
root[1] TF1 *mw=new
  TF1("maxwell",Maxwell,0,200,3)
```
Creation of the TF1

Range (0,200) — *Number of parameters* (3)

```
root[2] mw->SetParNames("Const","B","T")
```
Parameters names

```
root[3] mw->SetParameters(100,5,10)
```
Initial parameters values

```
root[4] mw->Draw()
```
Drawing the function (just to see it)

What happens in memory…

Maxwell(x,par)

```
.L Maxwell.C+

TF1 *mw=
 new TF1("maxwell",Maxwell,0,200,3)
```

maxwell

P0	P1	P2
?	?	?

What happens in memory…

Maxwell(x,par)

```
mw->SetParNames("Const","B","T")

mw->SetParameters(100,5,10)
```

maxwell

Const	B	T
100	5	10

Step 3: fit

```
root[0] TH1F *h1=(TH1F *)gROOT->FindObject("TestMaxwell")
```

fetching the pointer of the histogram to fit

```
root[1] h1->Fit("maxwell")
```

performing fit

```
root[2] mw->GetParameter(2)
```

Obtaining the value of the 3rd parameter (T)

```
root[2] mw->GetParameter("B")
```

Obtaining the value of the parameter named "B"

```
root[3] double para[3]
```

Creating an array with 3 real numbers

```
root[4] mw->GetParameters(para)
```

Obtaining the value of the parameters In the array

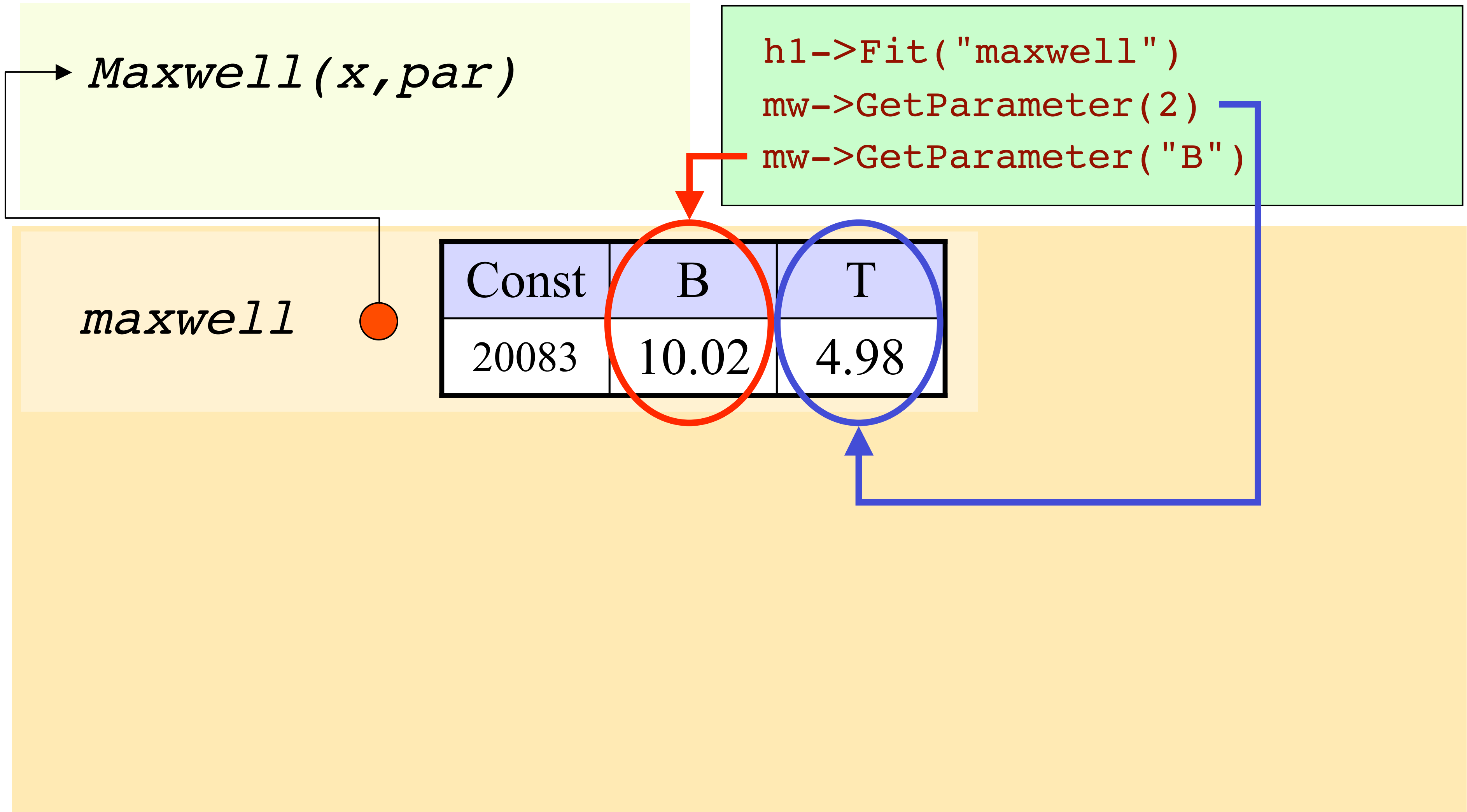
```
root[5] mw->GetChisquare()
```

Obtaining the value of the Chi2

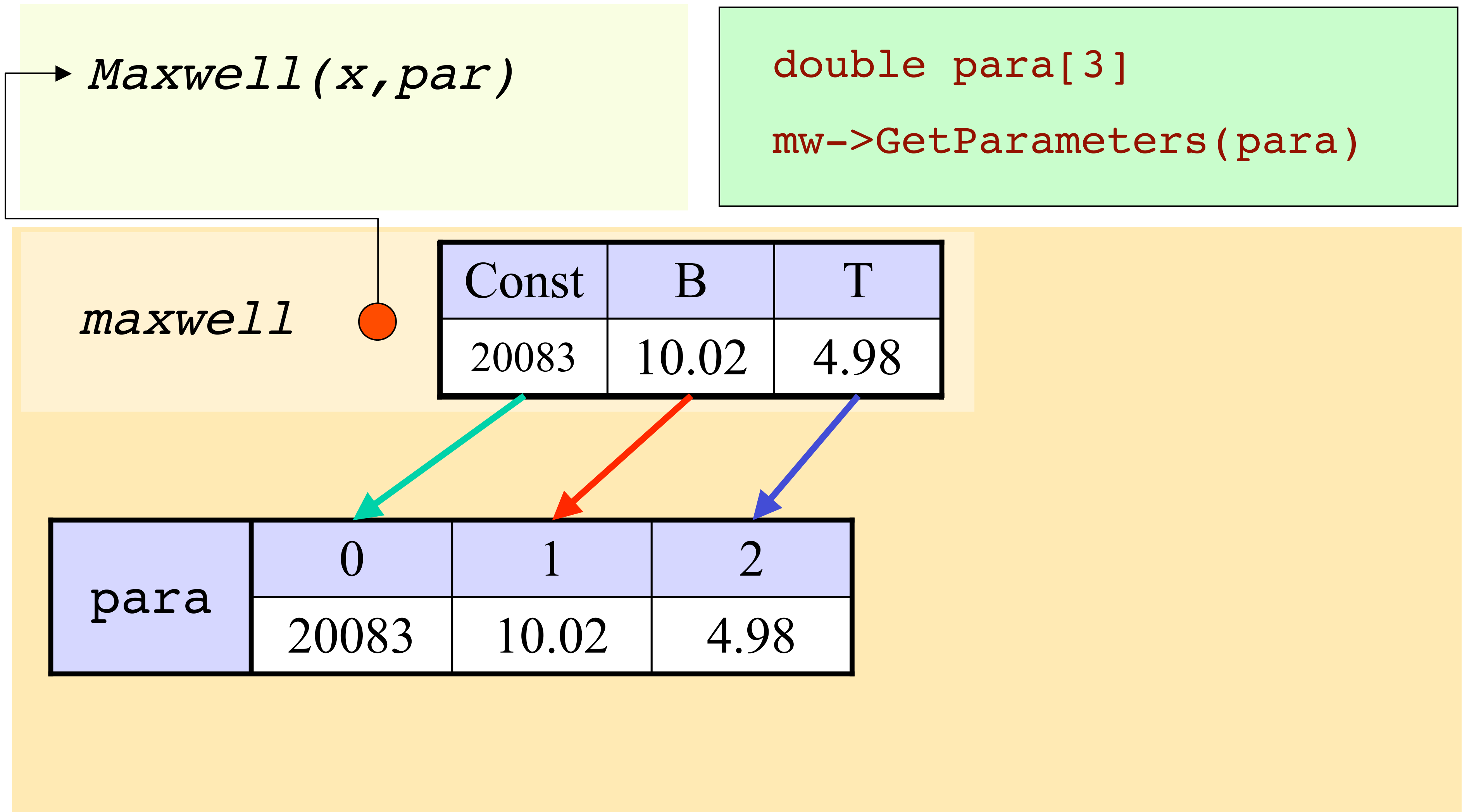
```
root[6] mw->GetNDF()
```

Obtaining the number degrees of freedom of the fit

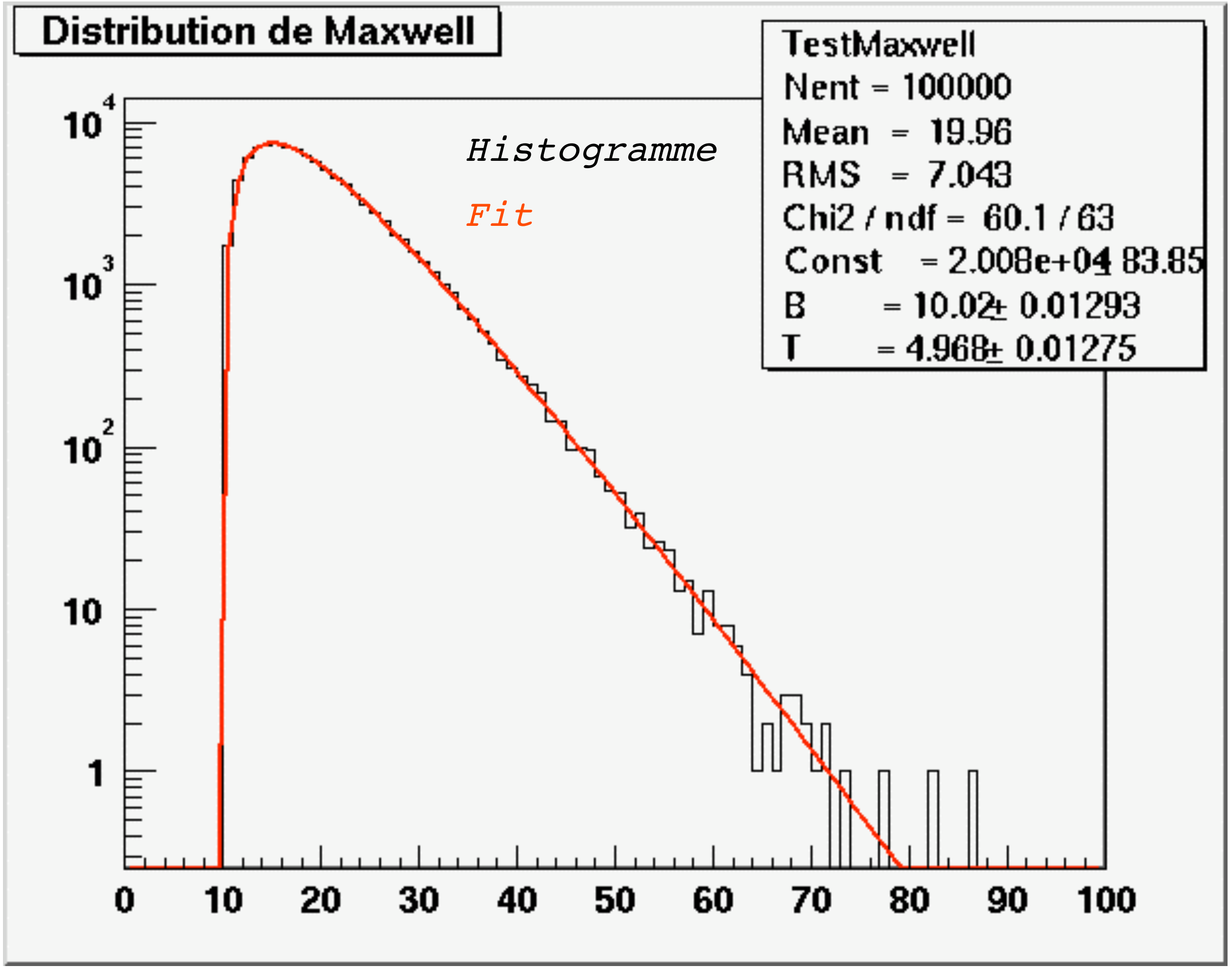
What happens in memory…



What happens in memory...



What a beautiful fit!



More complex: 2D gaussian

http://caeinfo.in2p3.fr/root/Formation/en/Day3/Gaus2D.C

```
//
// 2D Gaussian fit function
//
#include "TMath.h"

Double_t Gaus2D(Double_t *x, Double_t *par)
{
if(par[2] > 0 && par[4] > 0)
 {
 double rx=(x[0]-par[1])/par[2];
 double ry=(x[1]-par[3])/par[4];
 return par[0]*TMath::Exp(-(rx*rx+ry*ry)/2.);
 }
else
 return 0.;
}
```

Parameters array

Arguments array

x	X	Y
	0	1

par	Cst	X_0	σ_X	Y_0	σ_Y
	0	1	2	3	4

Include the function in ROOT

```
root[0] .L Gaus2D.C+
root[1] TF2 *g2D=new TF2("g2d",Gaus2D,-10,10,
                                             -10,10,5)

root[2] g2D->SetParNames("Const","X_{0}","#sigma_{x}",
                                      "Y_{0}","#sigma_{y}")

root[3] g2D->SetParameters(100,5,10,2,3)

root[4] g2D->Draw("surf")
```

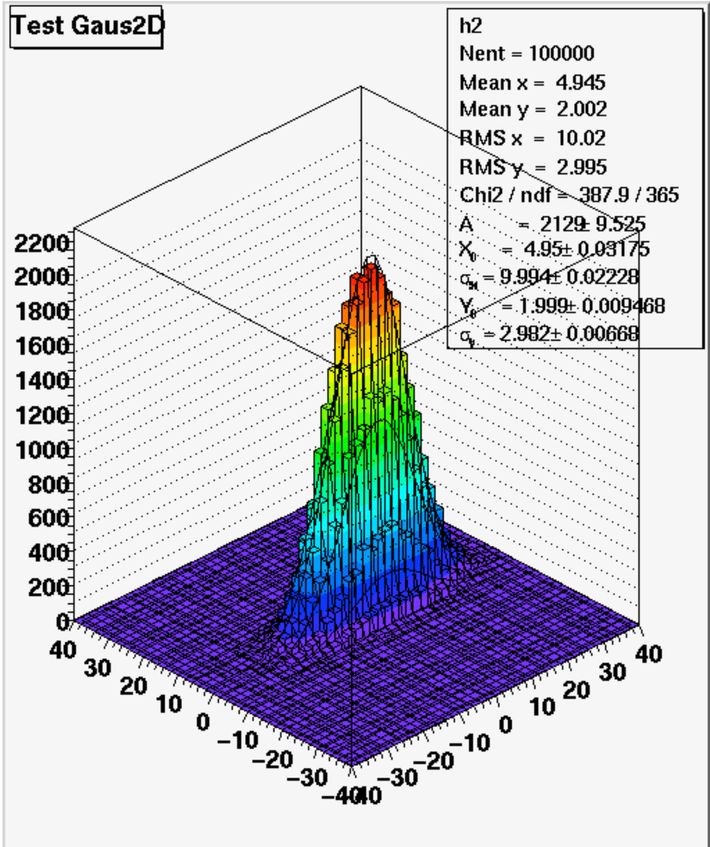
Make the fit

```
root[0] TH2F *h2=(TH2F *)gROOT->FindObject("TestGaus2D")
```

```
root[1] h2->Fit("g2d","V")
root[2] h2->Draw("lego2")
root[3] g2D->Draw("surf,same")
```

To plot with nice colours!

```
root[3] gStyle->SetPalette(1)
```



Fits with many functions…

Even harder: mixing 2 functions

Maxwell.C

```
//
// Sum of 2 Maxwellian functions
//
Double_t DeuxMaxwell(Double_t *x, Double_t *par)
{
  return Maxwell(x,par)+Maxwell(x,&par[3]);
}
```

Maxwell with par[0], par[1] and par[2]

Maxwell with par[3], par[4] and par[5]

x	E
	0

par	C_1	B_1	T_1	C_2	B_2	T_2
	0	1	2	**3**	4	5

Performing the fit

```
root[0] .L Maxwell.C+
```
Compilation and loading of the functions

```
root[1] TF1 *deuxmw=new TF1("deuxmax",DeuxMaxwell,0,200,6)
```
Creating the TF1

```
root[2] deuxmw->SetParNames("C1","B1","T1","C2","B2","T2")
```
Parameters names

```
root[3] deuxmw->SetParameters(1,1,1,2,2,2)
```
initial values of the parameters

```
root[4] gStyle->SetOptFit(kTRUE)
```
To plot the parameters values in the Statistics box

```
Root[5] TH1F *h2m=(TH1F *)gROOT->FindObject("Test2Maxwell")
```
Fetching the pointer of the histogram to fit

`root[6] h2m->Fit("deuxmax")`	*Performing the fit*
`root[7] double param[6]`	*array of doubles*
`root[8] deuxmw->GetParameters(param)`	*Getting the parameters values*
`root[9] mw->SetParameters(param)`	*Values for the first Maxwellian*
`root[10] mw->SetLineColor(kRed)`	*Its colour is set to red*
`root[11] mw->DrawClone("same")`	*Drawing a copy (why?)*
`root[12] mw->SetParameters(&param[3])`	*Values for the second Maxwellian*
`root[13] mw->SetLineColor(kBlue)`	*Its colour is set to blue*
`root[14] mw->DrawClone("same")`	*Drawing a copy*

What happens in memory…

Maxwell(x,par)

DeuxMaxwell(x,par)

```
.L Maxwell.C+
TF1 *deuxmw=new TF1("deuxmax",DeuxMaxwell,0,200,6)
deuxmw->SetParNames("C1","B1","T1","C2","B2","T2")
h2m->Fit("deuxmax")
```

maxwell

Const	B	T
?	?	?

deuxmax

C1	B1	T1	C2	B2	T2
901.02	10.00	5.02	1990.26	5.08	24.58

What happens in memory…

Maxwell(x,par)

DeuxMaxwell(x,par)

```
double param[6]

deuxmw->GetParameters(param)
```

maxwell

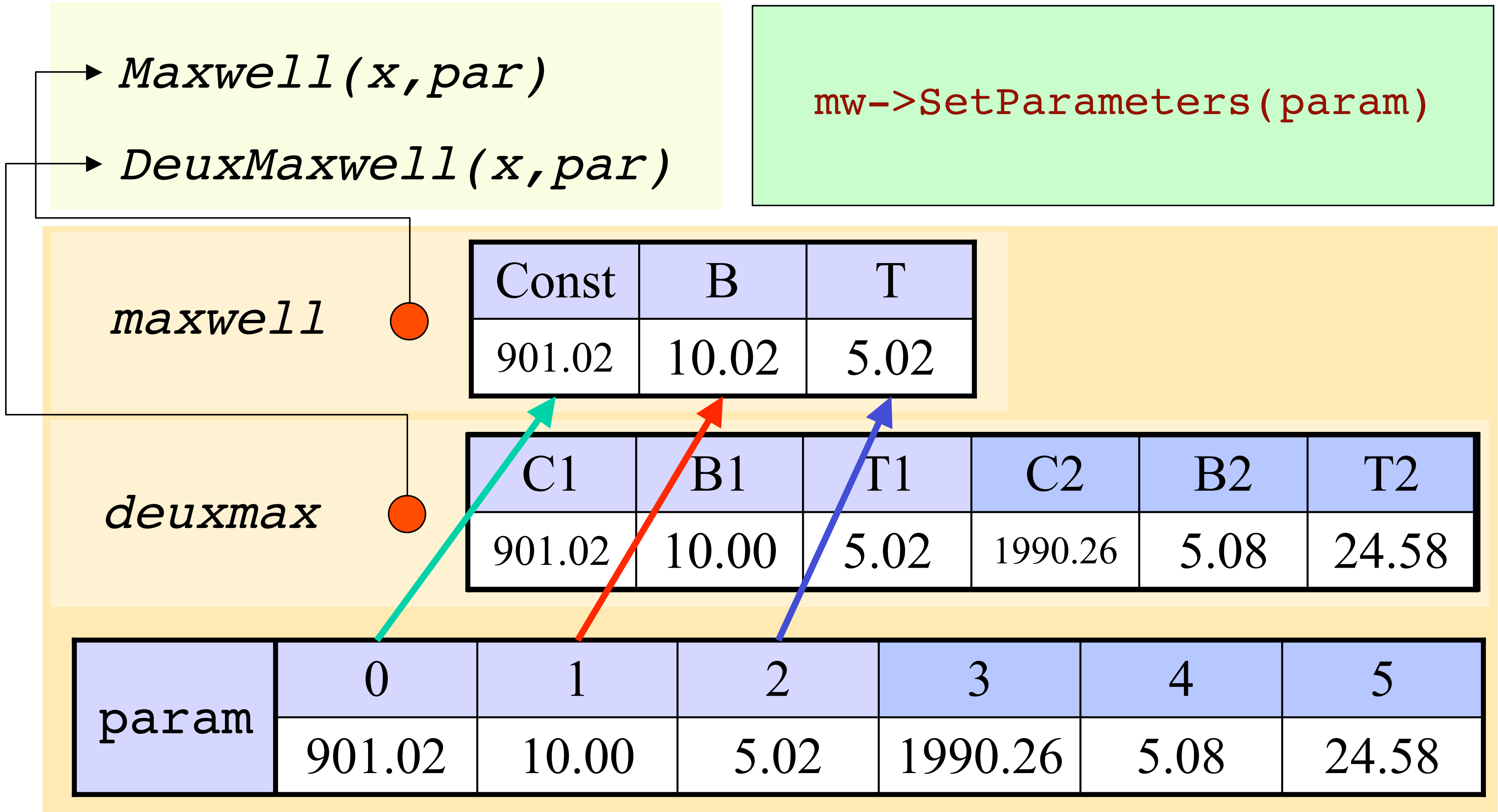
Const	B	T
?	?	?

deuxmax

C1	B1	T1	C2	B2	T2
901.02	10.00	5.02	1990.26	5.08	24.58

param	0	1	2	3	4	5
	901.02	10.00	5.02	1990.26	5.08	24.58

What happens in memory…



What happens in memory…

Maxwell(x,par)

DeuxMaxwell(x,par)

```
mw->SetParameters(&param[3])
```

maxwell

Const	B	T
1990.26	5.08	24.58

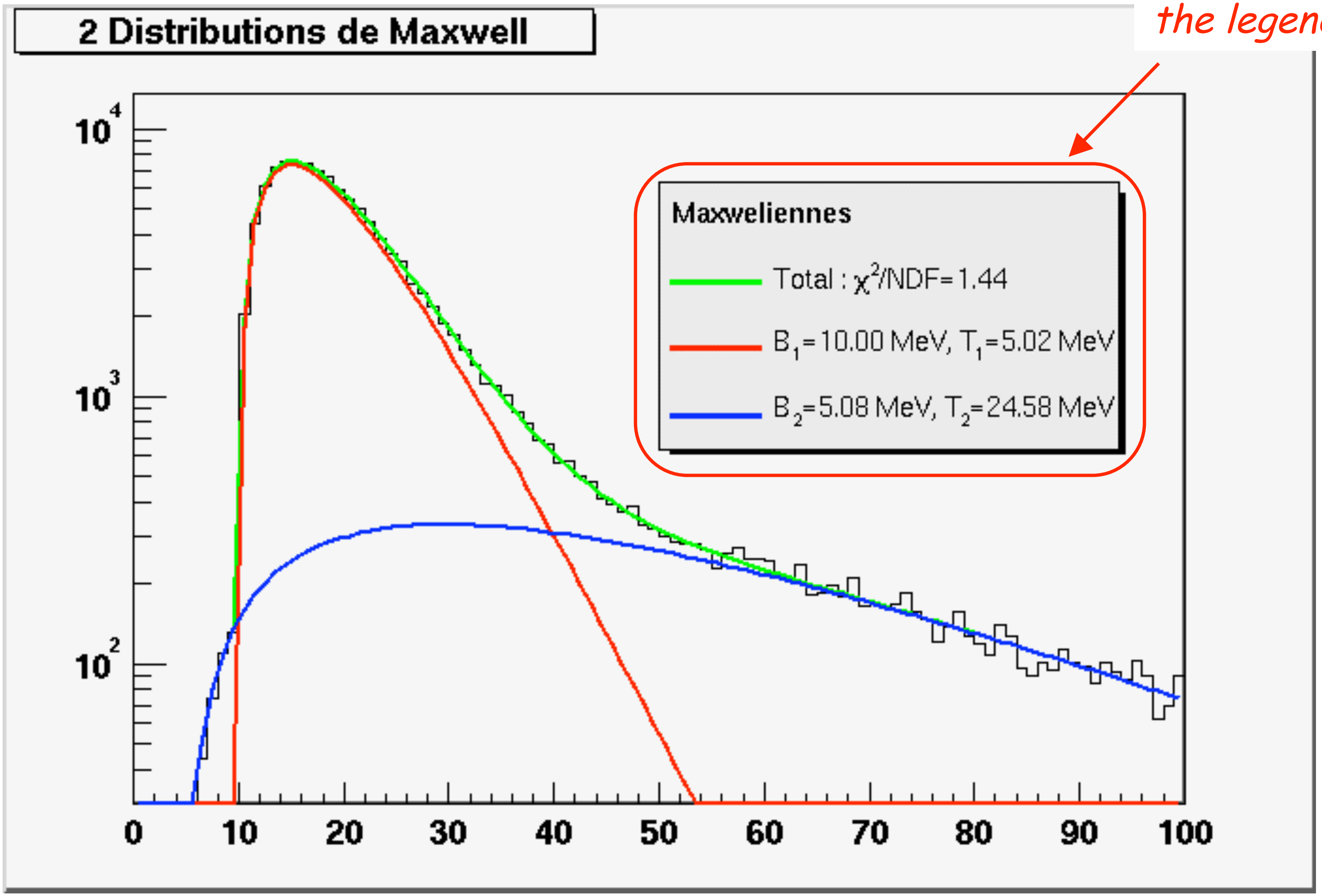
deuxmax

C1	B1	T1	C2	B2	T2
901.02	10.00	5.02	1990.26	5.08	24.58

param	0	1	2	3	4	5
	901.02	10.00	5.02	1990.26	5.08	24.58

It's really very beautiful!

How to draw the legend?



A figure with a legend

- Just add a `TLegend` object

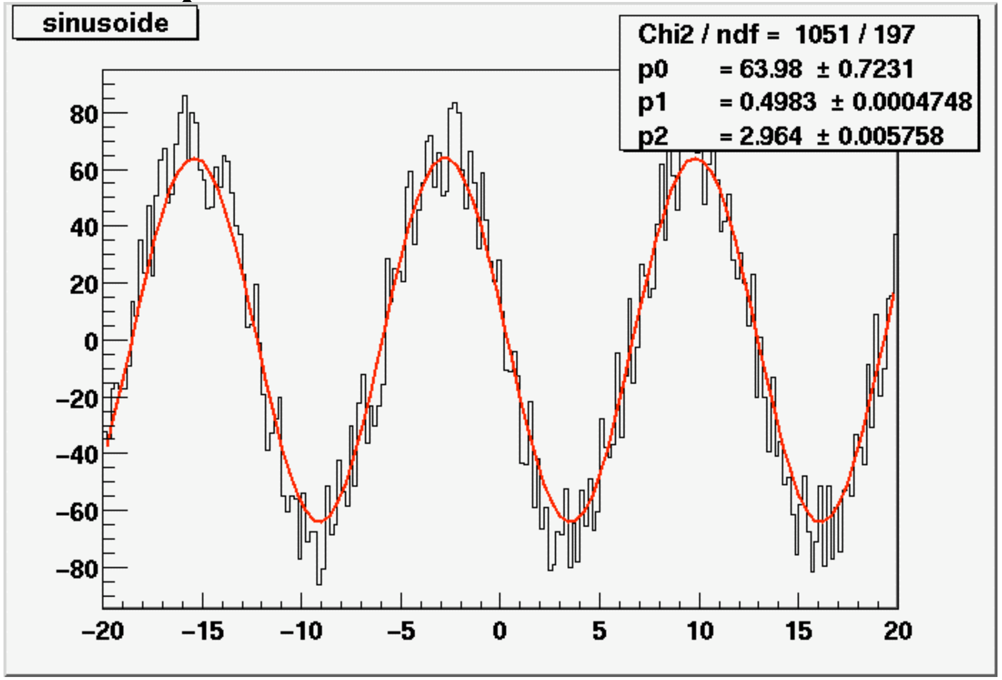
http://caeinfo.in2p3.fr/root/Formation/en/Day3/MakeFits.C

```
TLegend *legend=new TLegend(0.5,0.5,0.8,0.8,"Maxwelliennes");   Title
Char_t message[80];                                               Coordinates of the frame
TF1 *fun=h2m->GetFunction("deuxmax");   Fit function linked to the histogram
sprintf(message,"Total : #chi^{2}/NDF = %.2f",fun->GetChisquare()/fun->GetNDF());
legend->AddEntry(fun,message);   Adding the first line
TList *liste = gPad->GetListOfPrimitives();   List of objects in the current TPad
for(Int_t i=0;i<2;i++)   Loop on the two "maxwell" functions
 {
 fun=(TF1 *)liste->FindObject("maxwell");   Fetching the pointer of an object named "maxwell"
 fun->SetName(Form("maxwell%d",i+1));   Changing its name
 sprintf(message,"%s = %.2f MeV, %s = %.2f MeV",
          deuxmax->GetParName(3*i+1),fun->GetParameter(1),
          deuxmax->GetParName(3*i+2),fun->GetParameter(2));
 legend->AddEntry(fun,message);   Adding the line to the TLegend
 }
legend->Draw();   Drawing the TLegend
```

Exercise 1

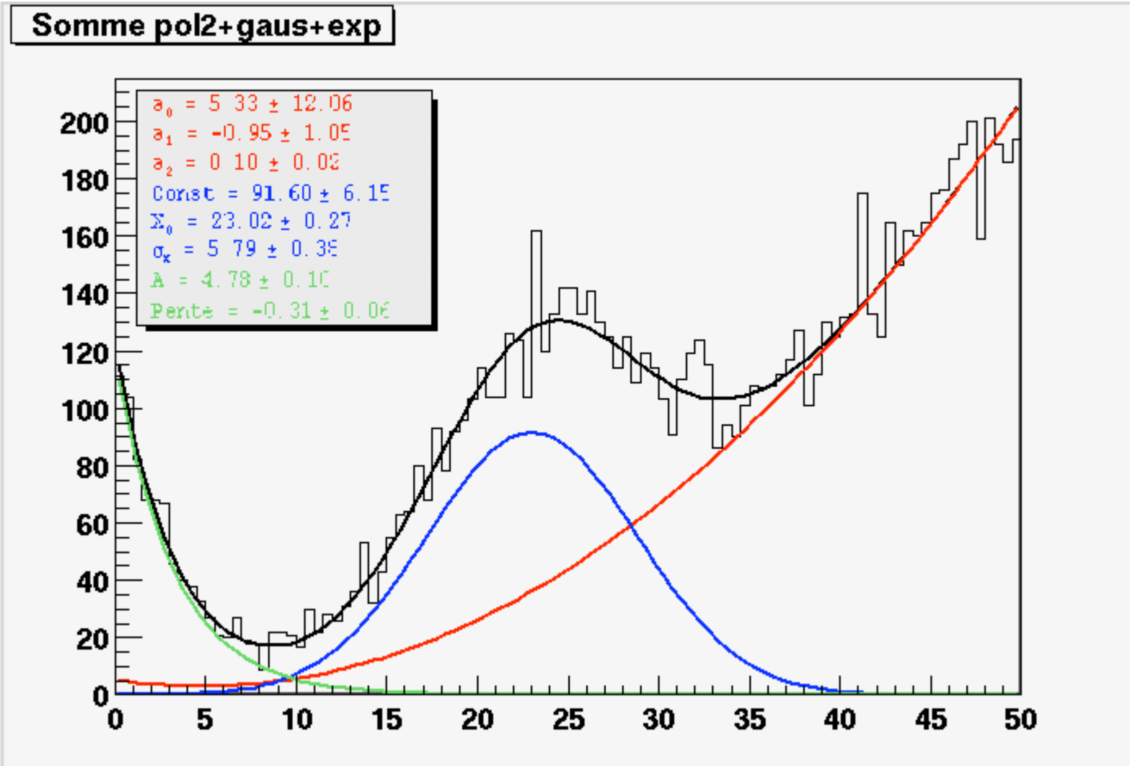
http://caeinfo.in2p3.fr/root/Formation/en/Day3/Fits.root

- Fit the histogram named `hfs` in the file `Fits.root` with a sinusoidal function and display the fit parameters.



Exercise 2

- Fit the histogram named `hfsomme` in fhe file `Fits.root` with the sum of an exponential function (expo), of a gaussian function (gaus) and a second degree polynomial function (pol2). Plot the total fit function on the histogram plus the 3 component functions.



Exercise 3

- Fit the histogram named **`hData`** in the file **`Fits.root`** with the sum of

- a background $$Background(x) = \frac{Ax}{1+\exp\left[(x-B)/C\right]}$$

- a signal $$Signal(x) = \begin{cases} G\exp\left(-\frac{1}{2}\left(\frac{x-x_0}{\sigma_g}\right)^2\right) & if\ x \le x_0 \\ G\exp\left(-\frac{1}{2}\left(\frac{x-x_0}{\sigma_d}\right)^2\right) & if\ x > x_0 \end{cases}$$

- Plot the total fit function and the background and the signal separately. Add a legend (**`TLegend`**) with an entry for each function.

Solution of exercise 3