

CERN ROOT Minuit2 Demo

E.C. Birdsall

What this is

- A brief introduction to Minuit2
- Intended for new PhD students to “get something working fast”
- Produced quickly one evening, there could be a bug lurking, don't use this as a “de-facto standard”

What this is not

- Not a guide to best practices
- Not a detailed overview of technicalities relating to obtaining correct minima or verification of results
- Not an in depth guide to all of Minuit2 functionality

What we will do

- example.C
- Find minimum value of x^2
- Find minimum value of x^2 with a Gaussian constraint

What we will do

- example2.C
- Create some fake data using TRandom3 random number generator
- Scale this data, and create a new dataset from it
- Use Minuit2 to find the scaling coefficients in a mock parameter fit

Download the example

- <https://github.com/edbird/Minuit2-Examples>
- Video for the first 2 demos (3rd one – the best one – not yet online)
-
- <https://www.youtube.com/watch?v=OVvgcfyNCBY>

Download the example

- If the examples do not work, then the following slides show their output

Example 1

- 2-in-1: First minimize x^2 starting from $x=3$
- Then minimize x^2 with a Gaussian constraint
- \$ root
- root[0] .L example.C
- root[1] example()


```
root [2] example()
Example without constraint enabled
operator()(param[0]=3)
operator() operator()=9
operator()(param[0]=3)
operator() operator()=9
operator()(param[0]=3.005)
operator() operator()=9.03003
operator()(param[0]=2.995)
operator() operator()=8.97003
operator()(param[0]=3.00154)
operator() operator()=9.00927
operator()(param[0]=2.99846)
operator() operator()=8.99074
operator()(param[0]=9.9476e-11)
operator() operator()=9.89547e-21
operator()(param[0]=0.000488281)
operator() operator()=2.38419e-07
operator()(param[0]=-0.000488281)
operator() operator()=2.38418e-07
operator()(param[0]=9.9476e-11)
operator() operator()=9.89547e-21
operator()(param[0]=0.000488281)
operator() operator()=2.38419e-07
operator()(param[0]=-0.000488281)
operator() operator()=2.38418e-07
operator()(param[0]=9.76563e-05)
operator() operator()=9.53676e-09
operator()(param[0]=-9.76561e-05)
operator() operator()=9.53672e-09
operator()(param[0]=9.9476e-11)
operator() operator()=9.89547e-21
fval changed from 9 to 9.89547e-21
param[0] changed from 3 +- 0.5 to 9.9476e-11 +- 1
done - it was easy
```

Example 1

Example 1

```
Example with constraint enabled
operator()(param[0]=3)
x * x = 9
c = 4
operator() operator()=13
operator()(param[0]=3)
x * x = 9
c = 4
operator() operator()=13
operator()(param[0]=3.005)
x * x = 9.03003
c = 3.98003
```

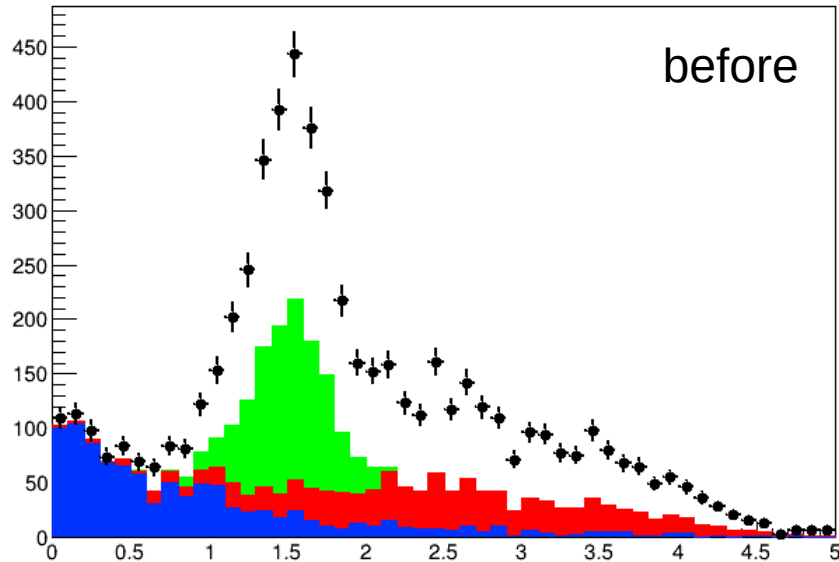
...

```
operator()(param[0]=2.49975)
x * x = 6.24873
c = 6.25127
operator() operator()=12.5
operator()(param[0]=2.5)
x * x = 6.25
c = 6.25
operator() operator()=12.5
fval changed from 13 to 12.5
param[0] changed from (3 +- 0.5) to (2.5 +- 0.707107)
done - it was easy
```

Example 2

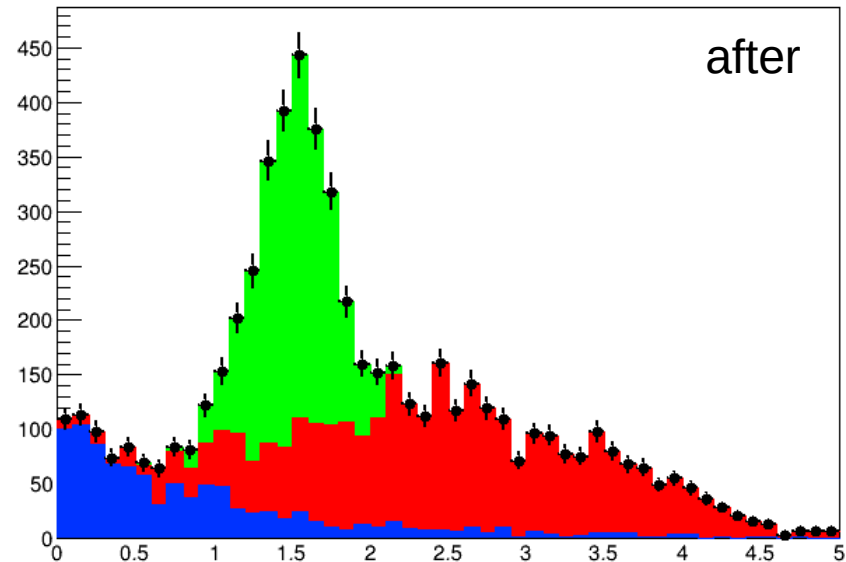
- \$ root
- root[0] .L example2.C
- root[1] example()

Example 2



```
const double A_coeff = 3.0;  
const double B_coeff = 2.0;  
const double C_coeff = 1.0;
```

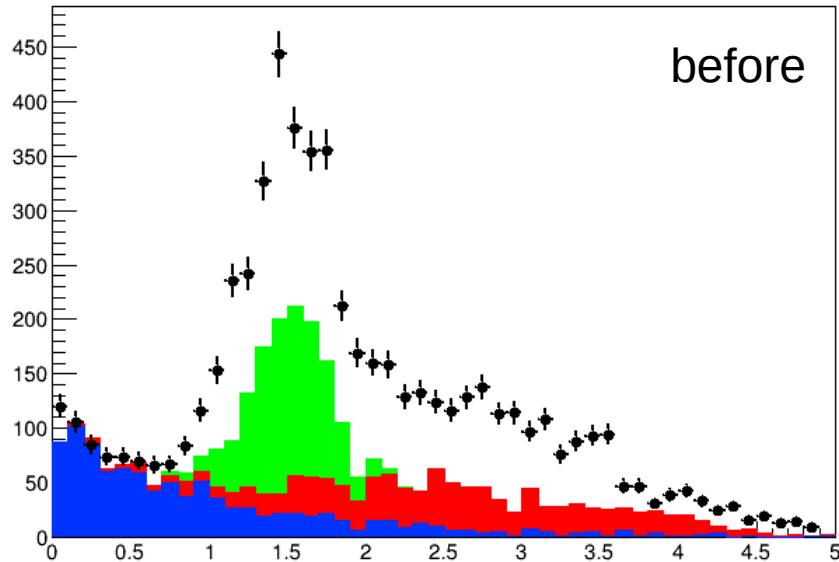
```
fval changed from 3860.84 to 5.4714e-06  
param[0] changed from 1 +- 0.1 to 3.00016 +- 0.069779  
param[1] changed from 1 +- 0.1 to 1.99997 +- 0.0579954  
param[2] changed from 1 +- 0.1 to 0.999961 +- 0.0447801  
operator()(A=3.06994, B=1.99997, C=0.999961)  
operator() operator()=1.14591  
fval_after_2=1.14591 fval_after=5.4714e-06  
done - it was easy
```



Example 3

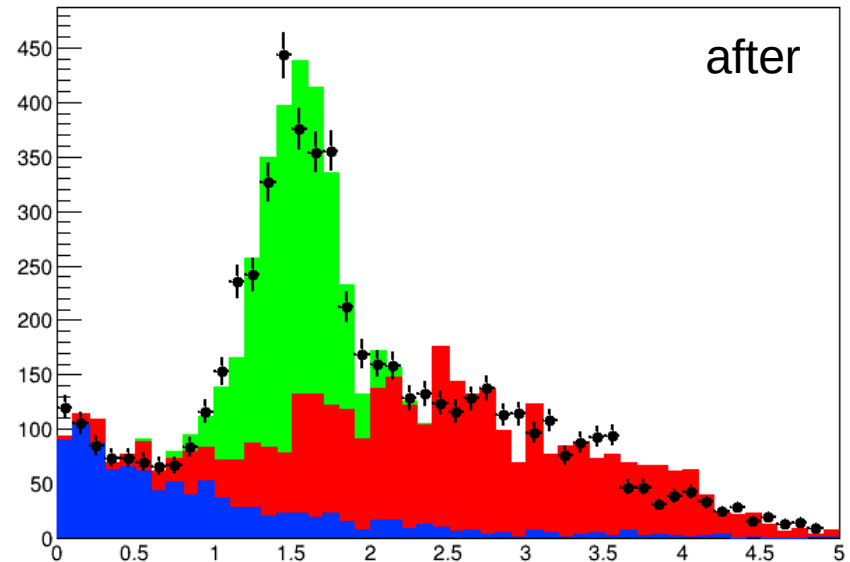
- Same as Example 2, but pseudo data generated from separate random MC
- \$ root
- root[0] .L example3.C
- root[1] example()

Example 3



```
const double A_coeff = 3.0;  
const double B_coeff = 2.0;  
const double C_coeff = 1.0;
```

```
fval changed from 4461.89 to 258.07  
param[0] changed from 1 +- 0.1 to 3.11829 +- 0.0703806  
param[1] changed from 1 +- 0.1 to 1.97363 +- 0.0587401  
param[2] changed from 1 +- 0.1 to 1.0367 +- 0.0463264  
operator()(A=3.18867, B=1.97363, C=1.0367)  
operator() operator()=259.201  
fval_after_2=259.201 fval_after=258.07  
done - it was easy
```



End

- End