A small toolkit for the TMinuit package

AliTMinuitToolkit

ALICE week October 2007

Alexander Kalweit, Marian Ivanov (GSI Darmstadt)

Why do we need a toolkit for TMinuit?

- TMinuit is the standard package for non-linear fitting in ROOT

- based on an old FORTRAN code with a **complicated interface**
- impossible to fit a curve to a 2d-histogram with TH2::Fit()
- no option for robust fitting

using the fitter

1) set up the fitter

```
AliTMinuitToolkit tool;
```

2) define formula

```
tool.SetFitFunction(TFormula * formula)
```

3) add data points

a) histogram:

```
tool.FitHistogram(TH2F/TH1F * his)
```

b) **matrix:**

```
tool.SetPoints(TMatrixD * points)
tool.Fit()
```

4) set initial values

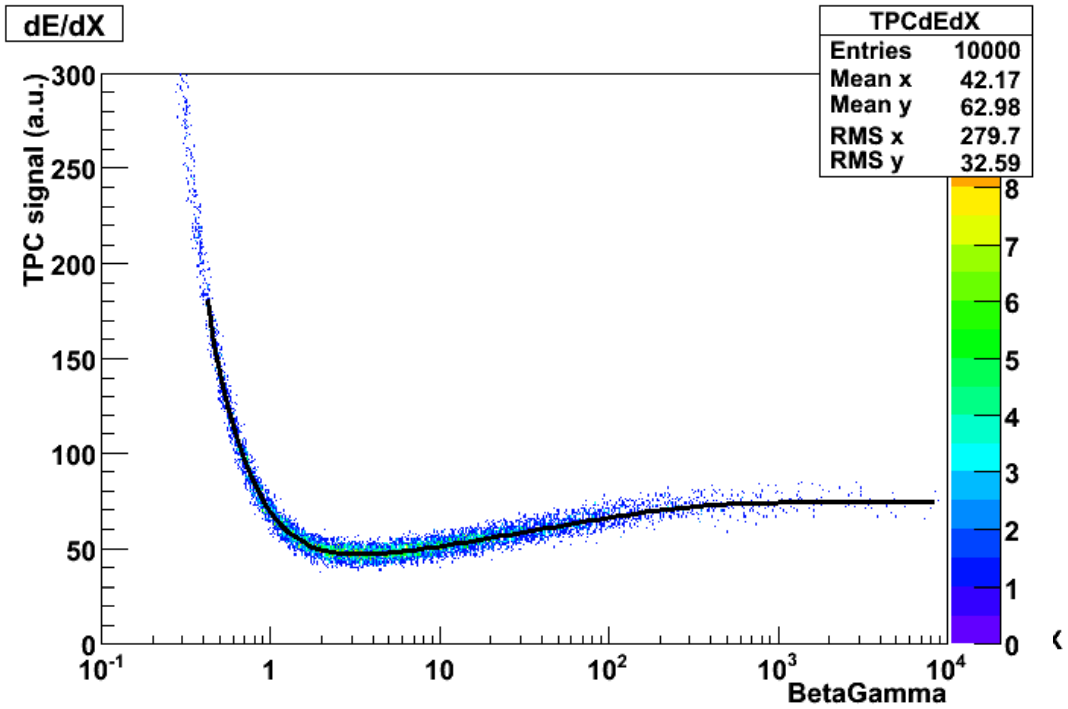
```
tool.SetInitialParam(TVectorD * param)
```

5) access results

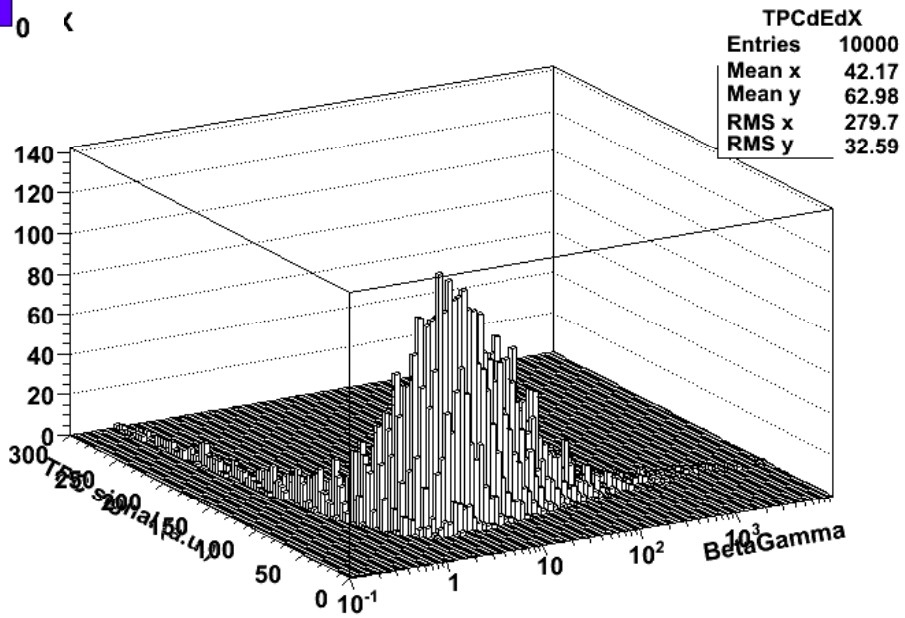
```
TVectorD * GetParameters()
```

example

- with AliTMinuitToolkit one can fit sth. like this:



- TH2F::Fit() “sees” only this:

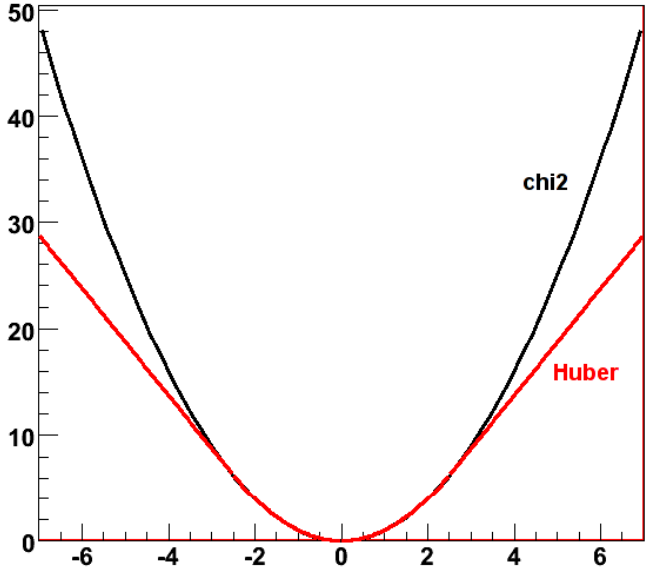


robust fitting

robust* = *reduce the influence of outliers

working method (1)

- Ordinary fit algorithms are based on χ^2-minimization.
- Therefore outliers have large weights.
- This can be avoided if the so called Huber function is used:



$$h(x) = x^2 \quad , \quad |x| < k$$
$$h(x) = 2kx - k^2 \quad , \quad |x| > k$$

► a rough guess for the mean distance of the “good” points to the curve (sigma) is needed:

```
AliTMinuitToolkit::EnableRobust(true, sigma)
```

working method (2)

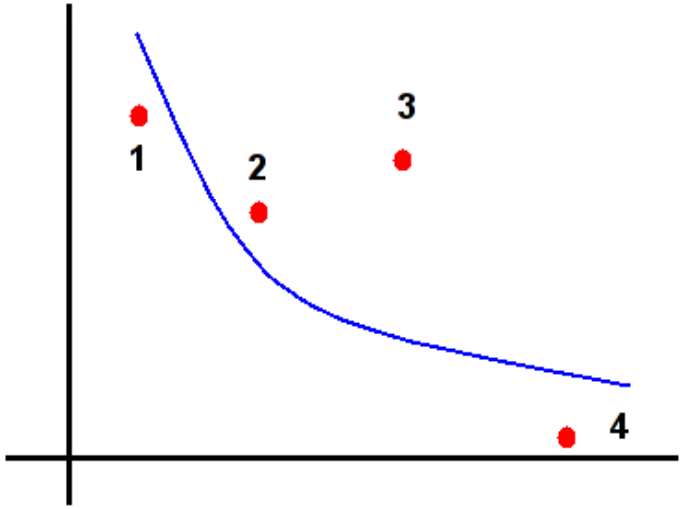
If **in addition** a weighting function is specified a 2nd run is started:

1) calculate the distance of each point before starting the iteration

2) sort the points according to their distance → index list

3) assign a weight to each point according to its position in the index list

The weighting function w(x) is defined between 0 and 1:
w(x=0) = weight of the nearest point
w(x=1) = weight of the point with the largest distance

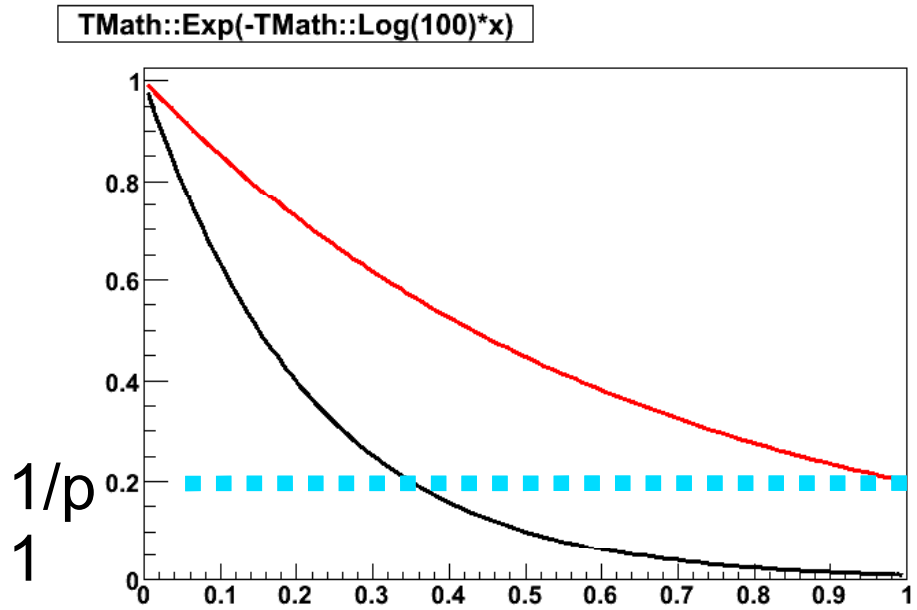


index list and weight:

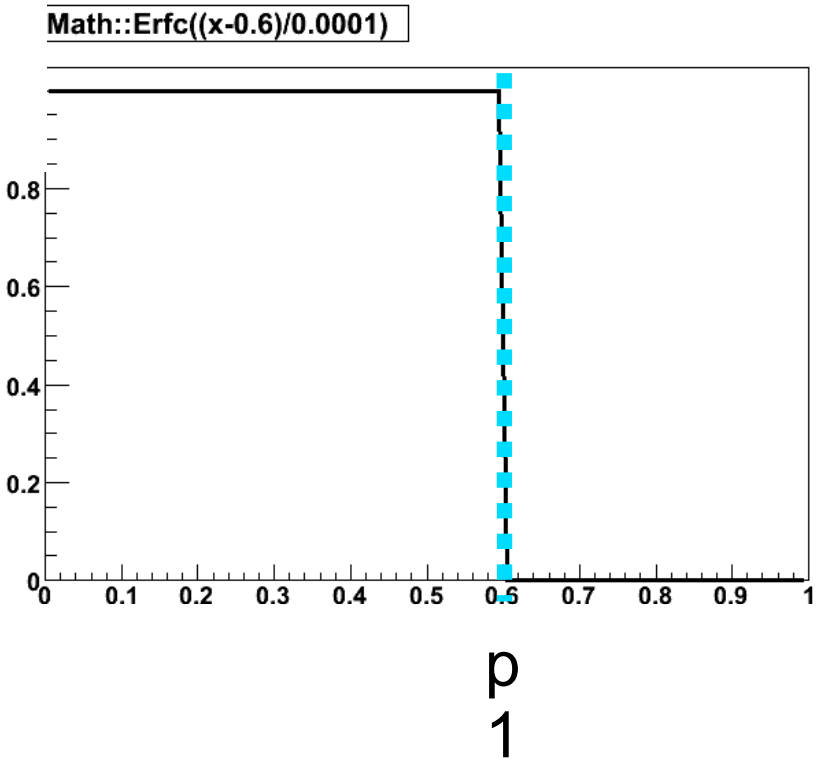
2 → 1
1 → 1
4 → 0.7

predefined weighting functions (1)

- SetWeightFunction("exponential", p1)

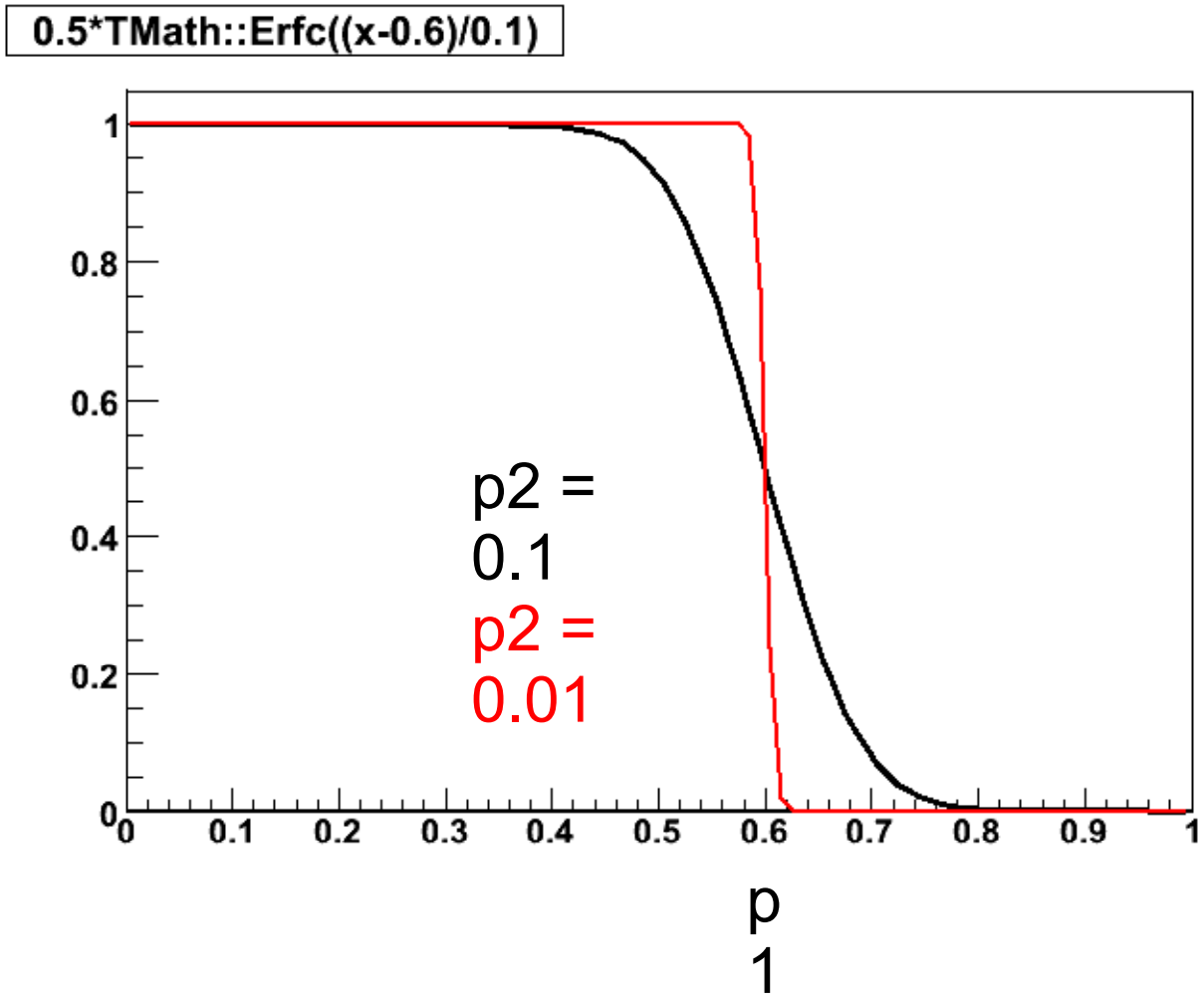


- SetWeightFunction("box", p1)



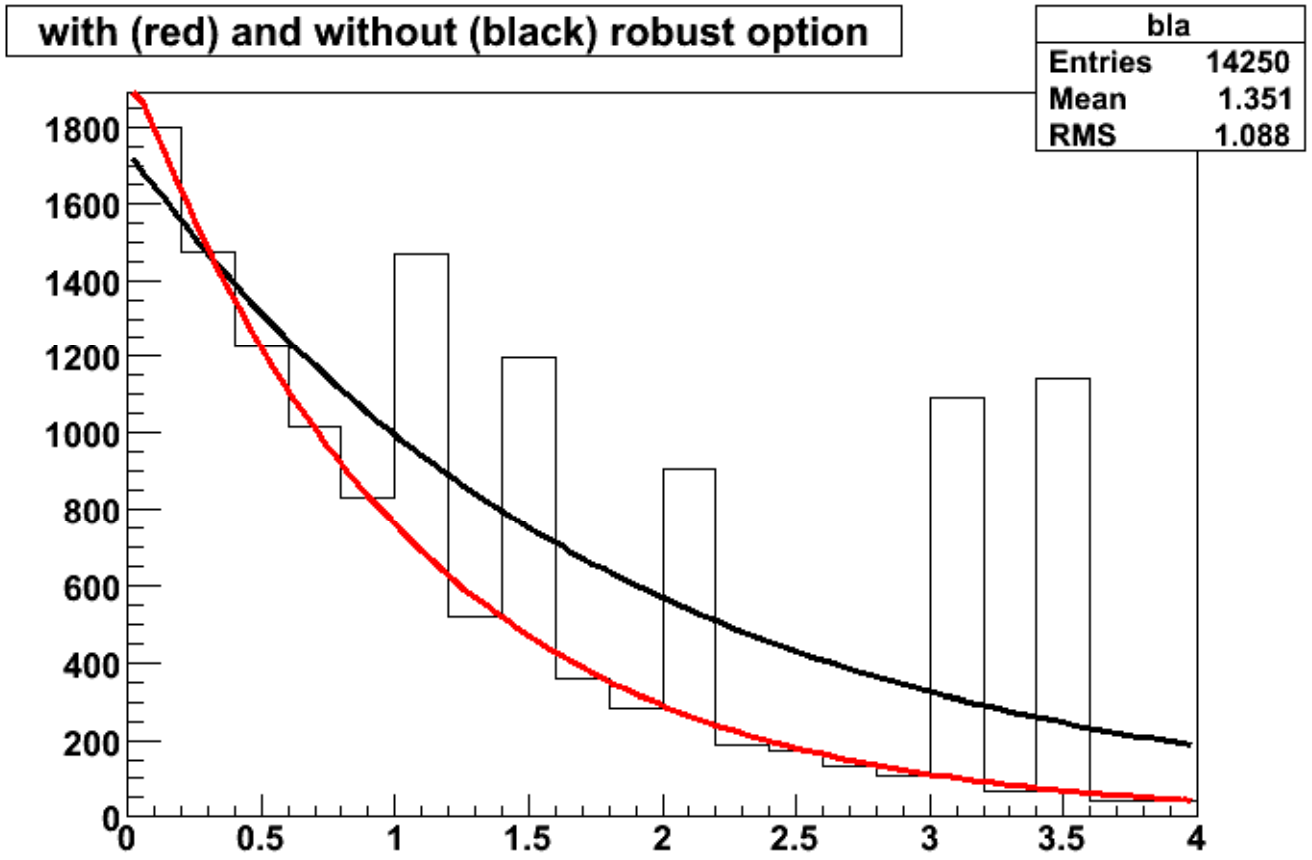
predefined weighting functions (2)

- SetWeightFunction("ErrorFunction", p1, p2)

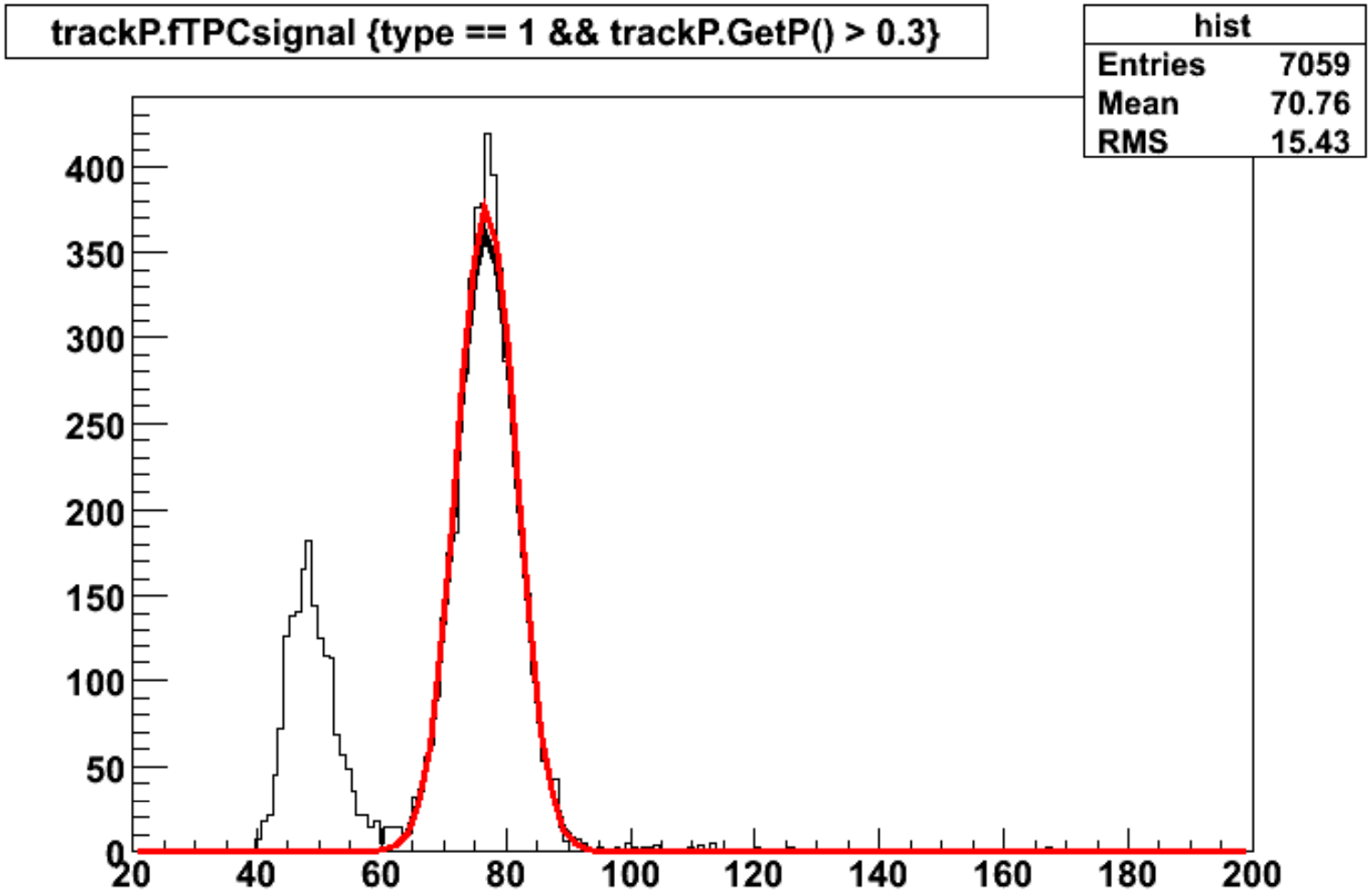


example (1)

- in AliTMinuitToolkit::Test()



example (2)


trackP.fTPCsignal {type == 1 && trackP.GetP() > 0.3}
hist
Entries 7059
Mean 70.76
RMS 15.43
400
350
300
250
200
150
100
50
0
20
40
60
80
100
120
140
160
180
200

problems with robust fitting

- very strong dependence on starting values
- no convergence guaranteed (the same with TMinuit)
- converging to local minima

additional information

- reference for robust fitting:

  Ekblom H. and Madsen K. (1988), Alogrithms for non-linear Huber estimation, BIT Numerical Mathematics 29 (1989) 60-76.

  internet: http://www.springerlink.com/content/m27721854298344/

- manual for TMinuit (description of algorithms etc.):
  http://wwwasdoc.web.cern.ch/wwwasdoc/WWW/minuit/minmain/minmain.html