RooFit tutorial

Arnab Purohit
Purdue University

Outline

Purpose

Structure

Basic Classes

Implementation

Toy Monte Carlo

Fitting data

Fitting options & results

Why do we fit?

- To estimate (point/interval) parameters of a hypothetical distribution from the observed data.
  - Y = f(observations|parameters)
- Example: Measured mass and couplings of Higgs boson.

χ^2 and Likelihood

- $\chi^2(a) = \Sigma(O_i - f(x_i;a))/\sigma_i^2$

  We mainly use χ^2/dof which is also known as reduced χ^2

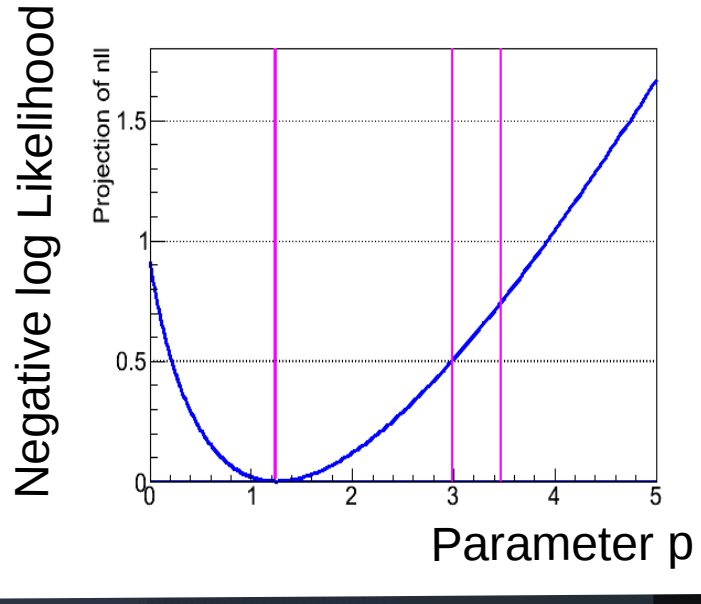
  Here ndof = number of observations - the number of fitted parameters.

- A **likelihood function** L(a) is proportional to the probability for the occurrence of a sample configuration x_1, ..., x_n given that the probability density f(x;a) with parameter a is known,

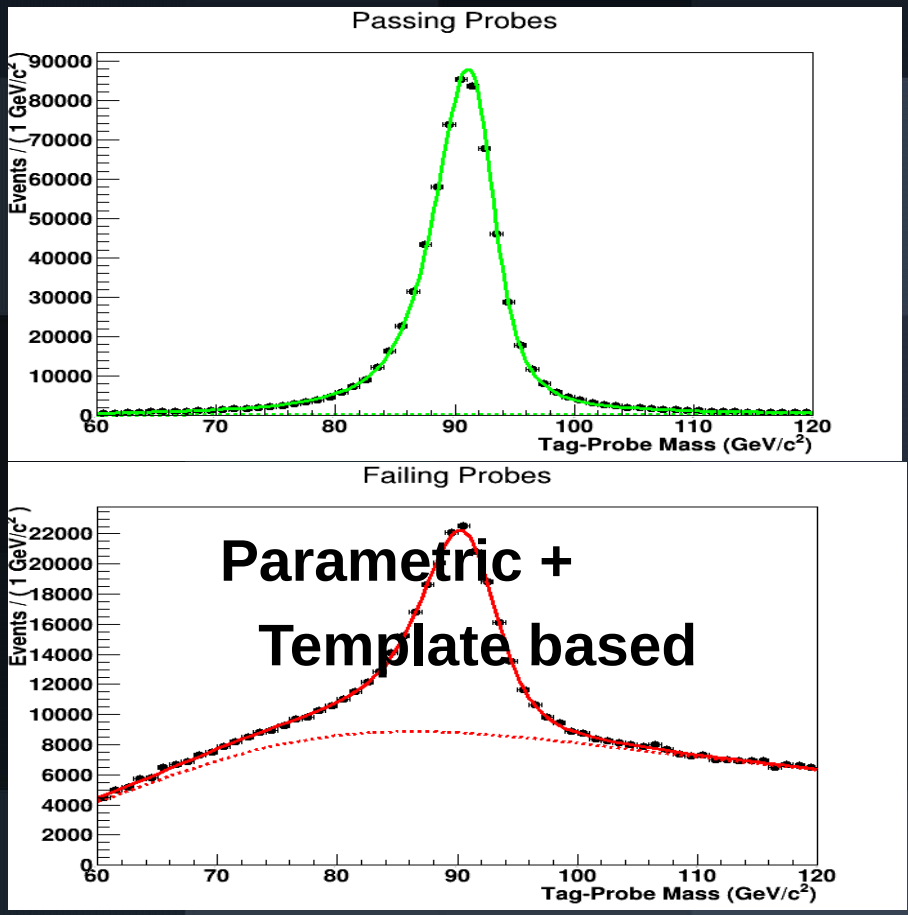
  - $L(a) = f(x_1;a)*f(x_2;a)*.....*f(x_n;a)$

NLL estimation

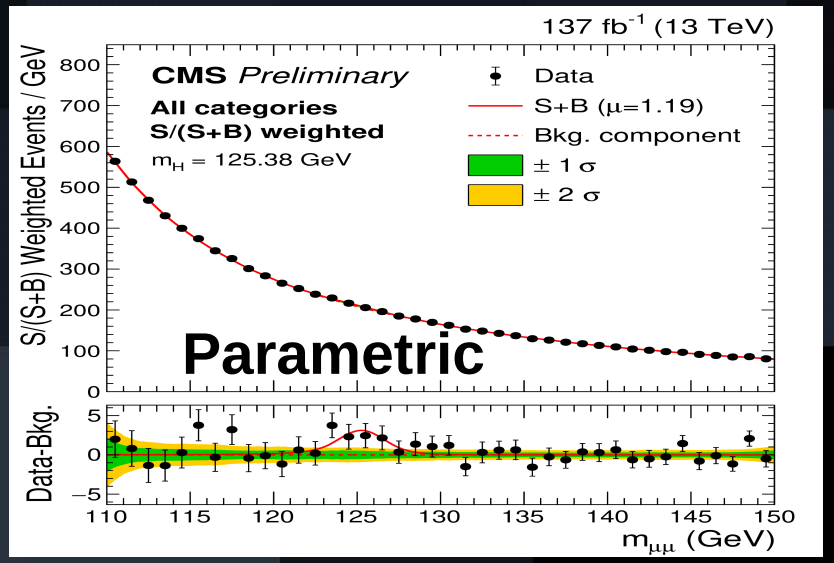
- The least-square fit and the maximum likelihood fit are equivalent when the distribution of observed events in each bin is normal.
- Best fit parameters p given by maximizing likelihood L or minimizing negative log likelihood (NLL).

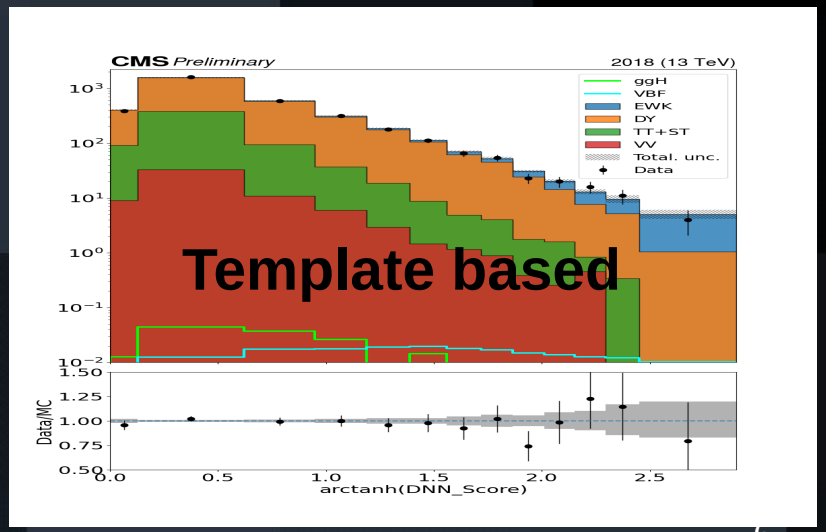


Purpose of RooFit



Tag and probe fit to measure cut efficiency and data-MC scale factor.

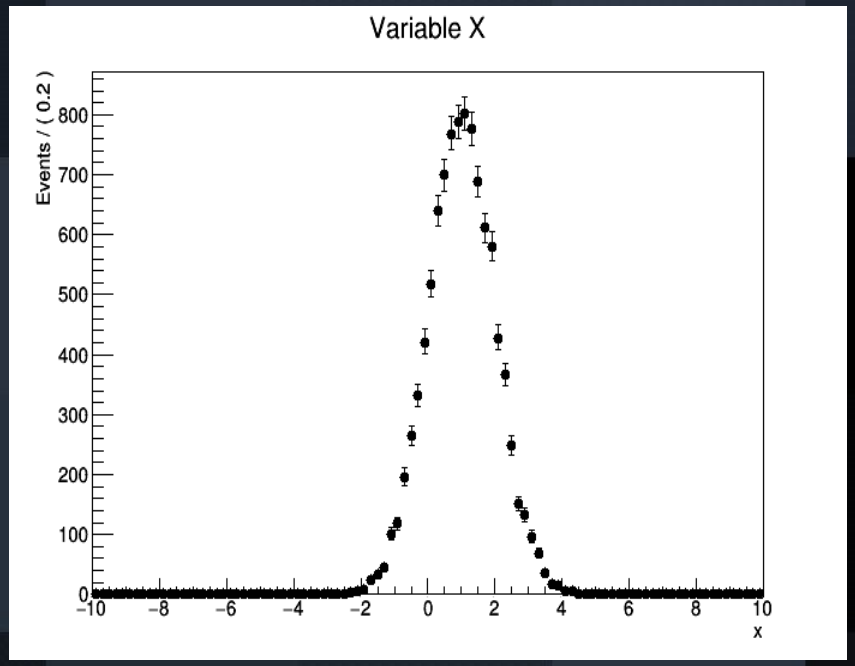




H → μμ

Purpose of RooFit

- The RooFit library provides a toolkit for modeling the expected distribution of events in a physics analysis.
  - Physical parameters
    - Position
    - Momentum
  - Detector effects
    - Resolution
    - Efficiency

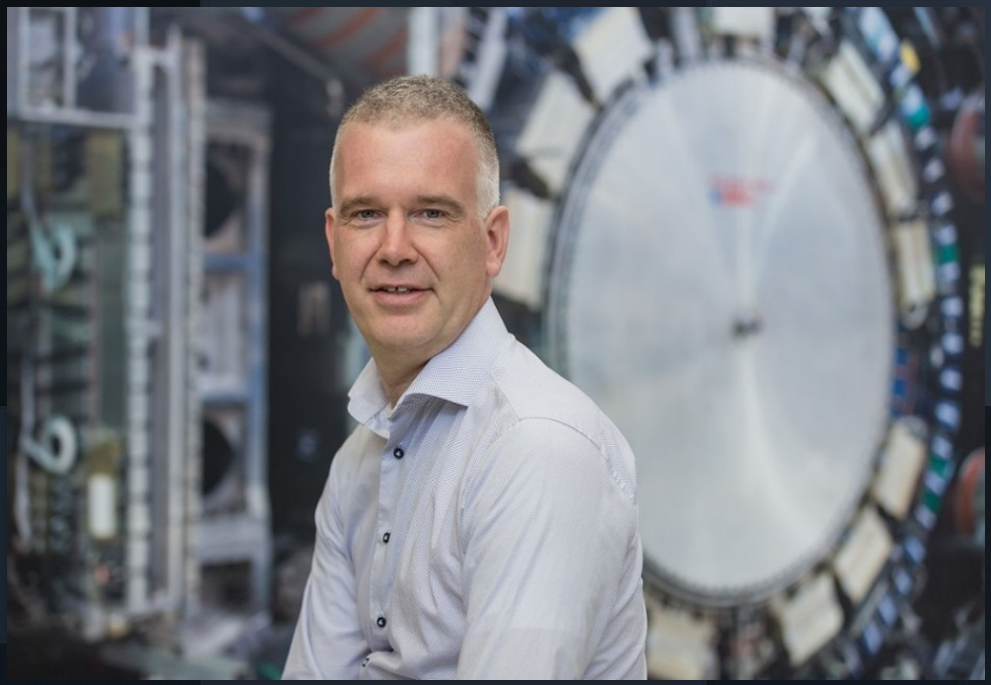


- Models can be used to perform unbinned maximum likelihood fits, produce plots, and generate "toy Monte Carlo" samples for various studies.

ROOFIT

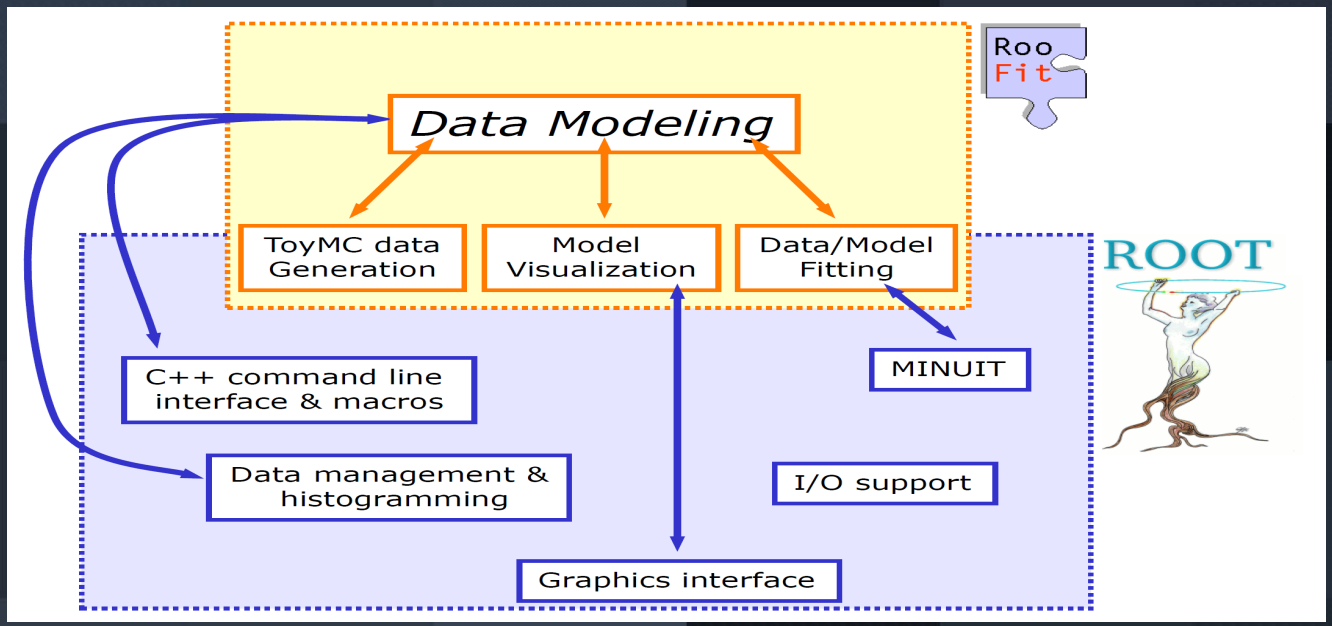
The RooFit toolkit for data modeling

Wouter Verkerke(UC, Santa Barbara) and David P. Kirkby(UC, Irvine)



ROOFIT and ROOT

- RooFit library comes with and depends on ROOT



To use RooFit in ROOT CINT:

```
gSystem->Load("libRooFit") ;
using namespace RooFit ;
```

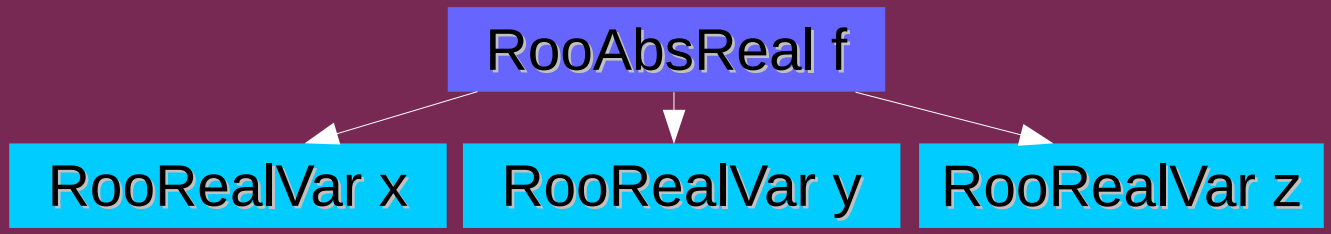
How to build a model?

Basic Structure

Mathematics

f(x,y,z)

RooFit



RooFit Code

```
RooRealVar x("x","x",5) ;
RooRealVar y("y","y",5) ;
RooRealVar z("z","z",5) ;
RooFunction f("f","f",x,y,z) ;
```

Data

- Generally speaking, data comes in two flavors:
  - unbinned data, represented in ROOT by class TTree
  - binned data, represented in ROOT by classes TH1,TH2 and TH3 .

RooFit can work with both.

Data

- Unbinned data can also be imported from ROOT TTrees
  - RooDataSet data(“data”,”data”,x,Import(*myTree)) ;
  - Imports TTree branch named “x”.
  - Can be of type Double_t, Float_t, Int_t or Uint_t. All data is converted to Double_t internally
- Binned data can be imported from ROOT THx histograms
  - RooDataHist data(“data”,”data”,x,Import(*myTH1)) ;
  - Imports values, binning definition.

How to fit model to data?

Fitting

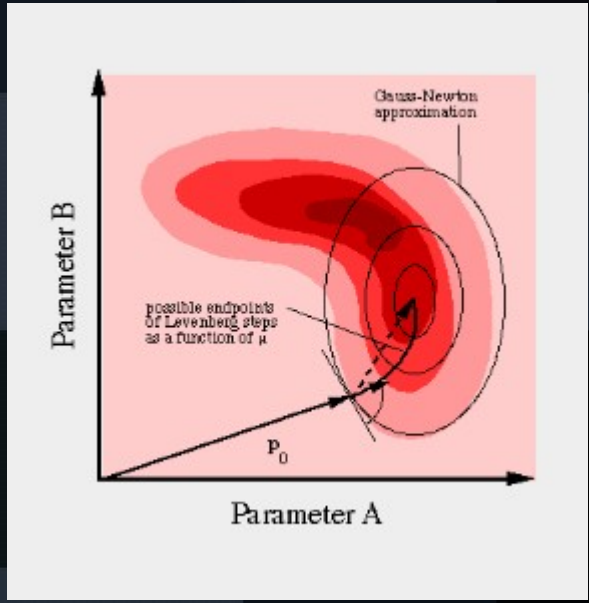
- Fitting a model to data involves the construction of a test statistic from the model and the data, most common choices are
  - χ^2 and
  - –log(likelihood)

  and minimizing that test statistics with respect to all parameters that are not considered fixed.
- The default fit method in RooFit is the unbinned maximum likelihood fit for unbinned data and the binned maximum likelihood fit for binned data.

How to minimize the test statistics?

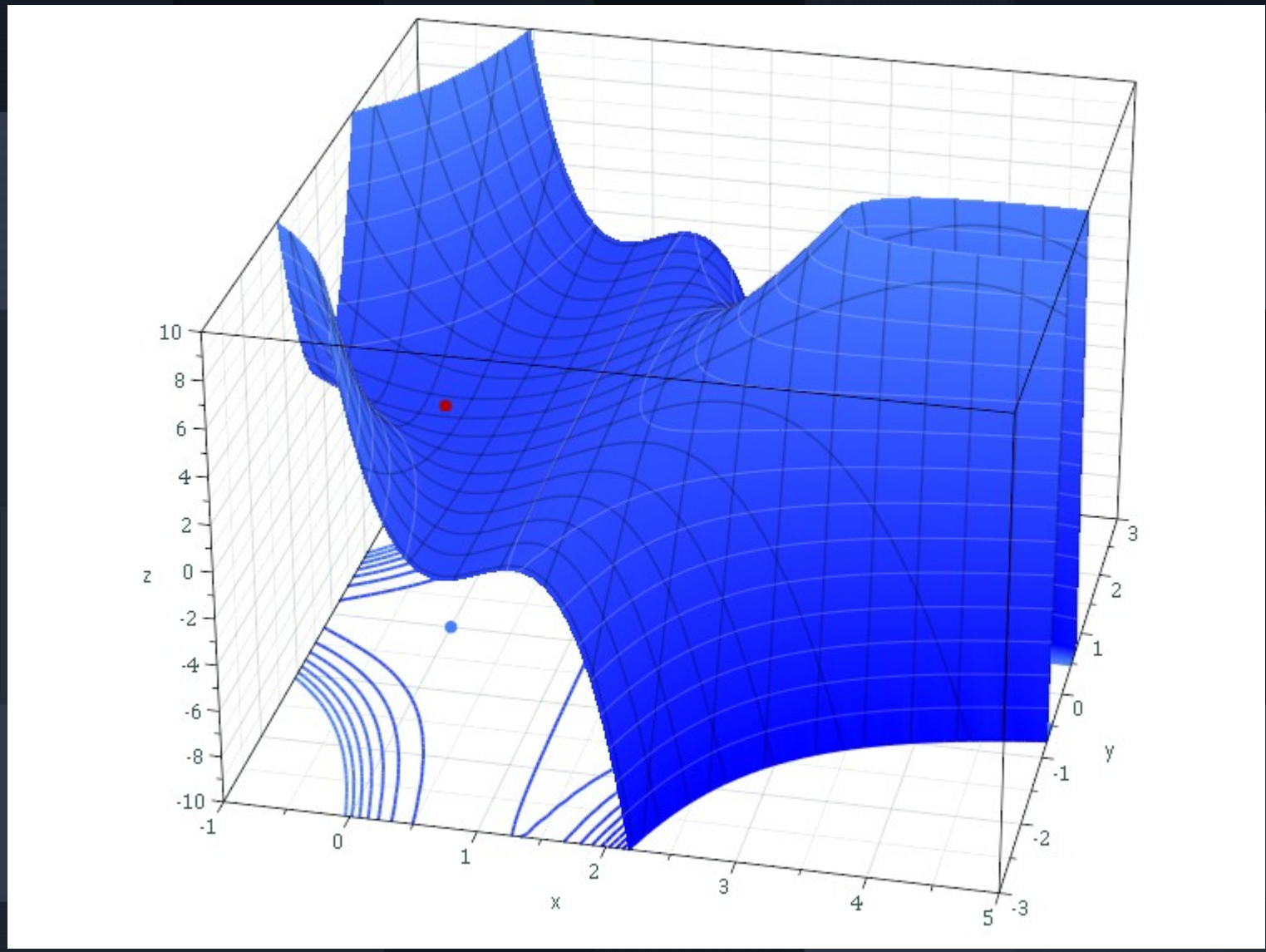
How to minimize

- In either case, the test statistic is calculated by RooFit.
- Suppose we have a statistic L(a).
- Minimization always proceeds by evaluating L(a) repeatedly at different points 'a' (i.e for different values of the parameters) determined by the minimization algorithm(s) used, until some minimum value is attained.
- At that point the user wants to have an idea about how sensitive the solution is to variations in the parameters



How to minimize

- At that point the user wants to have an idea about how sensitive the solution is to variations in the parameters: How steeply does L(a) increases away from a_{min} in parameter space?
- Physicists call this information the “errors of the parameters a”.
- In some cases the function may have more than one “local minimum”. Then the user must decide whether it is sufficient to know the location of any one local minimum or knowledge of “global minimum” is required.


10
8
6
4
2
0
-2
-4
-6
-8
-10
z
-1
0
1
2
3
4
5
x
3
2
1
0
-1
-2
-3
y

Minuit

- The minimization of the test statistic is performed by MINUIT through its TMinuit implementation in ROOT to perform the minimization and error analysis.
- Minuit offers 3 different minimization methods, each of which may be used alone or in combination with others depending on the behavior of L(a).
- Three minimization subroutines SEEK, SIMPLX and MIGRAD.

Minuit

a) SEEK – a monte carlo searching subroutine. It may be used at the beginning of a fit when no reasonable starting point is known or when it is suspected that there are several minima.

b) SIMPLX – A minimization subroutine using simplex method by Nelder and Mead.It is very “safe” and reasonably fast when far from minimum and may also be used to converge to the exact minimum. It does not compute covariance matrix but gives order of magnitude of the parameter error.

c) MIGRAD – this is based on a variable metric method by Fletcher. It is extremely fast near a minimum but slow if the function is badly behaved.

Minuit

- Some “global” logic is built into the program. For example, if MIGRAD fails it automatically calls SIMPLX to make another attempt.

The covariance matrix or error matrix

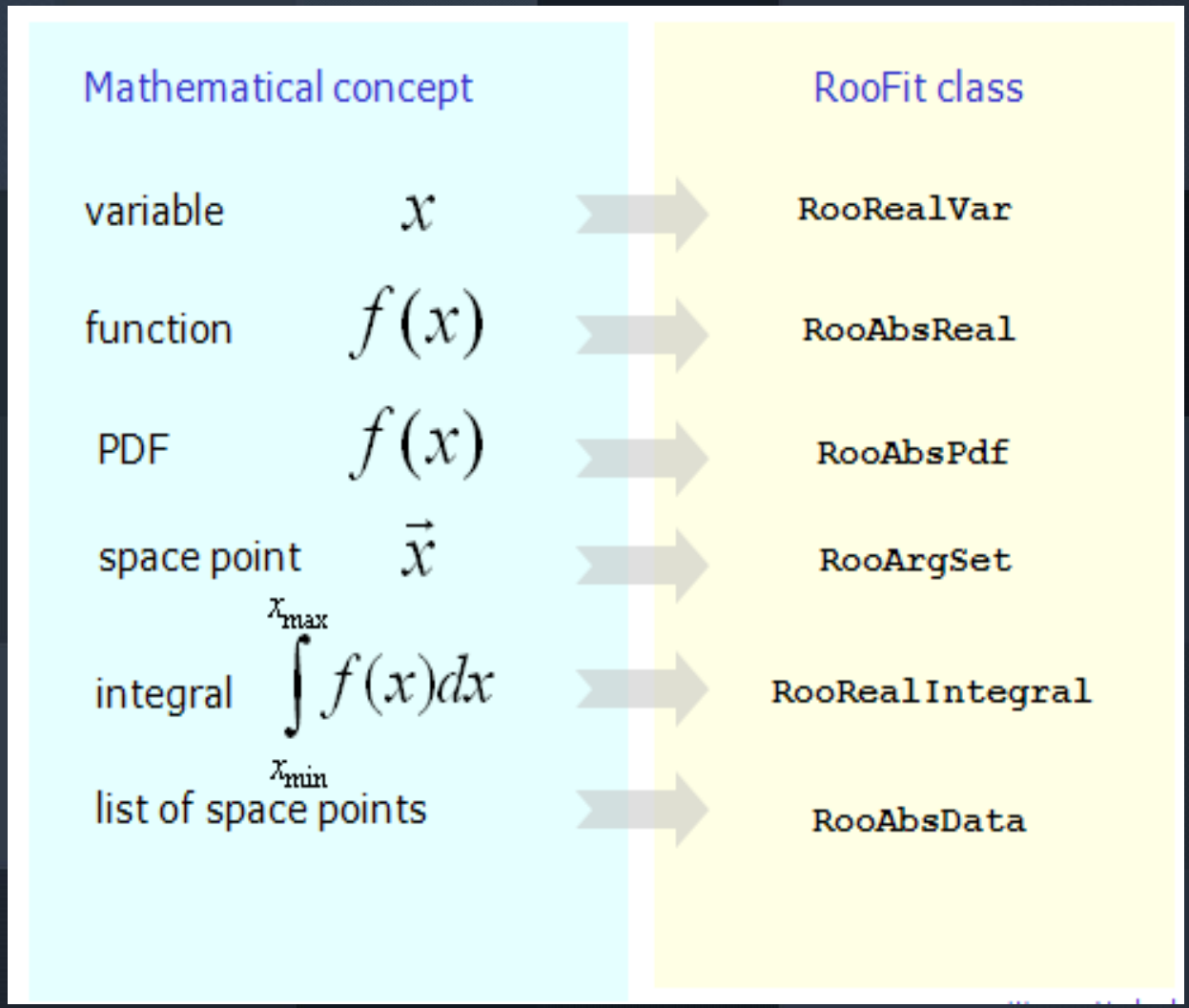
- The Hessian matrix describes the local curvature of a function of many variables.
- The Error matrix is inverse of the Hessian matrix.

$$\Sigma_a = H(a^*)^{-1}$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Simple example of complete maximum likelihood fit

- you define everything with RooFit classes:



```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
              1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
              1,0.1,10) ;

RooGaussian gauss("gauss","gaussian PDF",
              x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
                1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
                1,0.1,10) ;
RooGaussian gauss("gauss","gaussian PDF",
                x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

1. define 3 variables:
   - observable x
   - free parameters mean, sigma

1. Defining variables

- In ROOFIT variables are defined as -

```
RooRealVar(“name”, “title”, value, minValue, maxValue, “unit”)
```

- observables (i.e. x, y, energy, time) and parameters of a PDF (i.e. mean, sigma, slope) are both variables

  → the data set “tells” a PDF what it's observable is

  → all other variables must be parameters

- when fitting a PDF model to data: all free floating (= not fixed) parameters are fitted

- you can later on define and exclude a parameter from being fitted by the method

  `RooRealVar.setValue(value)` and `RooRealVar.setConstant()`

- construct flexible variable:

```
RooFormulaVar
mean_shifted(“mean_shifted”,”@0+@1”,RooArgList(mean,shift))
```

Simple example of complete maximum likelihood fit

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
              1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
              1,0.1,10) ;

RooGaussian gauss("gauss","gaussian PDF",
                x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

1. define 3 variables:
   - observable x
   - free parameters mean, sigma

2. Create PDF model with these variables

2. About PDFs

- construction of PDF is one of the most important steps
- bad PDF → bad fit
- the PDF contains the parameters which are fitted:

  this can either be parameters defining the shape of a PDF (like decay constant, Gaussian width, ...) or often fractions of different PDF components (i.e. signal vs. background component)
- PDFs are automatically normalized within RooFit

Some predefined pdfs

Gaussian	$\exp\left(-0.5\left(\frac{x-m}{s}\right)^2\right)$	`RooGaussian(name,title,x,m,s)`
Exponential	$\exp(a\cdot x)$	`RooExponential(name,title,x,a)`
Breit-Wigner	$\frac{1}{(x-m)^2+\frac{1}{4}g^2}$	`RooBreigWigner(name,title,x,m,g)`
Crystal Ball	$\left.\frac{(\frac{n}{\lvert a\rvert})^n e^{-\frac{1}{2}a^2}}{(\frac{n}{\lvert a\rvert}-\lvert a\rvert-x)^n}\right\rvert_{x<-\lvert a\rvert}$, $\left.\exp\left(-\frac{1}{2}\left(\frac{x-m}{s}\right)^2\right)\right\rvert_{x>-\lvert a\rvert}$	`RooCBShape(name,title,x,m,s,a,n)`

Simple example of complete maximum likelihood fit

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
              1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
              1,0.1,10) ;

RooGaussian gauss("gauss","gaussian PDF",
              x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

1. define 3 variables:
   - observable x
   - free parameters mean, sigma

2. Create PDF model with these variables

3. generate 10000 toy events

Simple example of complete maximum likelihood fit

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
             1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
             1,0.1,10) ;

RooGaussian gauss("gauss","gaussian PDF",
             x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

1. define 3 variables:
   - observable x
   - free parameters mean, sigma

2. Create PDF model with these variables

3. generate 10000 toy events

4. fit PDF and all floating parameters to data

Simple example of complete maximum likelihood fit

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
             1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
             1,0.1,10) ;

RooGaussian gauss("gauss","gaussian PDF",
             x,mean,sigma) ;
RooDataSet* data = gauss.generate(x,10000) ;
gauss.fitTo(*data) ;
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

1. define 3 variables:
   - observable x
   - free parameters mean, sigma

2. Create PDF model with these variables

3. generate 10000 toy events

4. fit PDF and all floating parameters to data

5. plot data and PDF

Accessing the fit results

// Construct function object representing –log(L)

- RooNLLVar nll(“nll”,”nll”,pdf,data) ;

// Minimize nll w.r.t its parameters

- RooMinuit m(nll) ;
- m.migrad() ; // find min NLL
- m.hesse() ; // symmetric errors assuming parabola
- m.minos() ; // asymmetric errors from min NLL+0.5
- nll->plotOn(frame,ShiftToZero()) ;

- `RooFitResult* r = gauss.fitTo(*data,Save()) ;`
- `r->Print() ;`

```
RooFitResult: minimized FCN value: 25055.6, estimated distance to
minimum: 7.27598e-08

coviarance matrix quality:Full, accurate covariance matrix

Floating Parameter     FinalValue +/- Error

--------------------------------------------------------

Mean                   1.7233e-02 +/- 3.00e-02

Sigma                  2.9809e+00 +/- 2.17e-02
```

- `r->correlationMatrix().Print() ;`

  Easy way visualize correlation matrix:

- `r->correlationHist->Draw("colz") ;`

Goodness of Fit

- frame->makePullHist() ;
- frame->makeResidHist() ;

Fit result

```
[#1] INFO:Minization -- RooMinuit::optimizeConst: activating const optimization
 **********
 **    13 **MIGRAD        1000           1
 **********
 FIRST CALL TO USER FUNCTION AT NEW START POINT, WITH IFLAG=4.
 START MIGRAD MINIMIZATION.  STRATEGY  1.  CONVERGENCE WHEN EDM .LT.1.00e-003
 FCN=25019.2 FROM MIGRAD    STATUS=INITIATE       10 CALLS          11 TOTAL
                     EDM= unknown      STRATEGY= 1      NO ERROR MATRIX
  EXT PARAMETER                CURRENT GUESS       STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  mean        1.00000e+000   2.00000e+000   2.02430e-001  -1.99022e+002
   2  sigma       3.00000e+000   9.90000e-001   2.22742e-001   1.98823e+002
                               ERR DEF= 0.5
 MIGRAD MINIMIZATION HAS CONVERGED.
 MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
 COVARIANCE MATRIX CALCULATED SUCCESSFULLY
 FCN=25018.5 FROM MIGRAD    STATUS=CONVERGED      32 CALLS          33 TOTAL
                     EDM=5.79448e-007    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  mean        1.01746e+000   3.00149e-002   3.29345e-004  -8.34497e-002
   2  sigma       2.97870e+000   2.19221e-002   5.32112e-004   1.48773e-001
                               ERR DEF= 0.5
 EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 9.009e-004 1.839e-005
 1.839e-005 4.806e-004
 PARAMETER  CORRELATION COEFFICIENTS
       NO.  GLOBAL      1      2
        1  0.02795   1.000  0.028
        2  0.02795   0.028  1.000
 **********
 **    18 **HESSE         1000
 **********
 COVARIANCE MATRIX CALCULATED SUCCESSFULLY
 FCN=25018.5 FROM HESSE     STATUS=OK             10 CALLS          43 TOTAL
                     EDM=5.79794e-007    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                INTERNAL      INTERNAL
  NO.   NAME      VALUE            ERROR       STEP SIZE       VALUE
   1  mean        1.01746e+000   3.00144e-002   6.58691e-005   1.01922e-001
   2  sigma       2.97870e+000   2.19217e-002   2.12845e-005  -4.31732e-001
                               ERR DEF= 0.5
 EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 9.009e-004 1.792e-005
 1.792e-005 4.806e-004
 PARAMETER  CORRELATION COEFFICIENTS
       NO.  GLOBAL      1      2
        1  0.02723   1.000  0.027
        2  0.02723   0.027  1.000
[#1] INFO:Minization -- RooMinuit::optimizeConst: deactivating const optimization
RooRealVar::mean  = 1.01746  +/- 0.0300144  L(-10 - 10)
RooRealVar::sigma  = 2.9787  +/- 0.0219217  L(0 1 - 10)
```