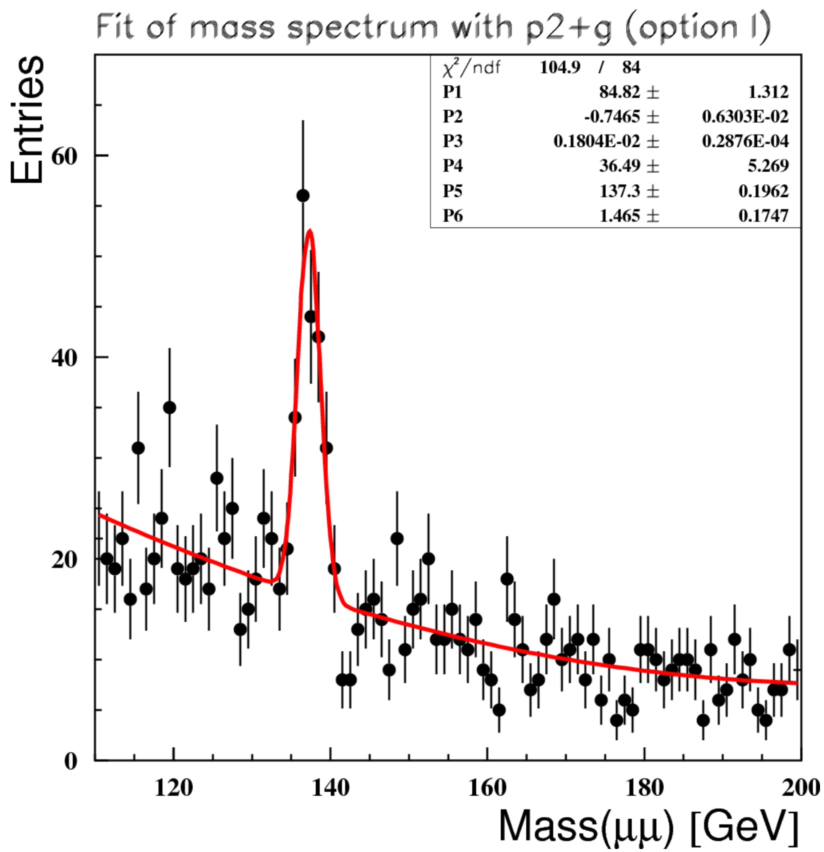
Mass peak fit tutorial

Terascale statistic school workshop april 2, 2014

Olaf Behnke (DESY), Vladyslav Libov (DESY)



What you really want to know:

- Number of signal events
- Mass of resonance
- Width of resonance

Practical questions:

- Choice of binning
- Use χ^2 variants or Poisson Likelihood?
- Parameterisation for signal and background
- + many more ☺

Mass peak fit tutorial with MINUIT

- We will use TMINUIT program
- Note: we will use practical features of MINUIT to perform mass-peak fit – we will not really look deeply behind the scenes to understand the MINUIT program/algorithms – focus is on getting *a feeling for mass peak fits and results*

MINUIT program

- Fortran code developed in the 1970s by Fred James (CERN)
- Minimization of a function FCN
- MIGRAD is main minimization algorithm:
  - uses a *variable metric algorithm (Davidson-Fletcher-Powell)*
  - difference to the basic Newton-Raphson method: the Hesse matrix is not completely re-evaluated at each iteration but only updated from last iteration (saves time) → Sometimes needed to call HESSE after MIGRAD to obtain correct parameter error matrix $V = H^{-1}$

MINUIT program phases

- Initialisation (e.g. defining parameters)
- Minimization (many commands possible)
- Finishing (retrieving results)

MINUIT commands we will use

- "SET PRI 2" — Defines printout level (range -1 to 3)
- "Fix 3 4" — Fix parameter 3 and 4 (they will be not fitted)
- "Release 3 4" — Free parameter 3 and 4 (they will be fitted)
- "MIGRAD" — Run main minimizer to minimize FCN
- "HESSE" — Determine Hesse matrix H = $\frac{1}{2}\frac{d^2(FCN)}{dpar^2}$
- "MINOS" — Determine negative and positive uncertainties from points where FCN=FCN_min+1 **(Profile Likelihood: FCN is minimised wrt all other free parameters)**
- "MNC 1 2" — Calculates contours of FCN=FCN_min+1 with respect to parameters 1 and 2 **(Profile Likelihood: FCN is minimised wrt all other free parameters)**

Tutorial code structure (in main directory)

- Generating masspeak pseudodata:
  - generate_data.exe*
  - Steering in config/generate_data/gauss_pol0

- Fit signal and background in pseudodata:
  - Mass_peak_fit.exe*
  - Steering in config/mass_peak_fit/gauss_pol0
  - Call to MINUIT commands in src/Fit.cxx

- Compile and run scripts (generating data and fitting)
  - ./run.sh (gen+fit) .rungen.sh (gen) .runfit.sh (fit)

Note: for most tasks you have to edit only the two steering files and rerun the run.sh script!

Get the tutorial code

Module load root

cp /afs/desy.de/group/school/statschool2014/mass_peak_tutorial.tar.gz .

tar –zvxf mass_peak_tutorial.tar.gz

cd mass_peak_tutorial

Tutorial topic

- gauss_pol0

$$N_s/(\sigma\sqrt{2\pi})\exp\left(-\frac{(m-\mu)^2}{2\sigma}\right) + c\,N_b$$

Note: Ns and Nb denote the total number of signal and background events in the sample (c is appropriate normalisation constant)

If time allows (probably not):

- gauss_pol3

$$N_s/(\sigma\sqrt{2\pi})\exp\left(-\frac{(m-\mu)^2}{2\sigma}\right) + p0 + p1\cdot m + p2\cdot m^2 + p3\cdot m^3$$

Tutorial Intro – step by step – 1. generate

- Generating masspeak pseudodata:
  - generate_data.exe*
  - Steering in config/generate_data/gauss_pol0

1) ./rungen.sh	Compile and generate data/test.root
2) gv data/test.eps	Look at generated mass spectrum
3) emacs config/generate_data/gauss_pol0	Edit generation steering file

Tutorial Intro – step by step – 1. generate

- Generating masspeak pseudodata:
  - generate_data.exe*
  - Steering in config/generate_data/gauss_pol0

```
output_file      test.root
histogram        test
bin_width   □    0.02
xmin             3
xmax             5
#Function Seed Parameters
distribution     gauss 1    100    4.0      0.02
distribution     pol0  1    1000
```

Ns

Random Seeds
(0=unix time, 1 = fixed)

Nb

Peak
position

Peak
width

Tutorial Intro – step by step – 1. generate

- Generating masspeak pseudodata:
  - generate_data.exe*
  - Steering in config/generate_data/gauss_pol0

1) ./rungen.sh	Compile and generate data/test.root
2) gv data/test.eps	Look at generated mass spectrum
3) emacs config/generate_data/gauss_pol0	Edit generation steering file

4) Change some parameter, save and repeat 1) and 2) see effect?

5) Restore default version, do it in editor or by
cp config/generate_data/gauss_pol0_ref config/generate_data/gauss_pol0

Tutorial Intro – step by step – 2. fit

- Fit Signal and background in the pseudo data
  - Mass_peak_fit.exe*
  - Steering in config/mass_peak_fit/gauss_pol0
  - Call to MINUIT commands in src/Fit.cxx

1) ./runfit.sh	Compile and fit – watch the screen output
2) gv results/fit.eps	Look at fitted data
3) emacs config/mass_peak_fit/gauss_pol0	Edit fit steering file

Tutorial Intro – step by step – 2. fit

Minuit print out

```
INFO: opened config/mass_peak_fit/gauss_pol0
 PARAMETER DEFINITIONS:
    NO.   NAME         VALUE      STEP SIZE      LIMITS
     1 Ns          1.00000e+02  1.00000e+01     no limits
     2 Mu          4.00000e+00  1.00000e-01     no limits
 **********
 **    1 **FIX          2
 **********
 PARAMETER DEFINITIONS:
    NO.   NAME         VALUE      STEP SIZE      LIMITS
     3 Sigma       2.00000e-02  1.00000e-01     no limits
 **********
 **    2 **FIX          3
 **********
 PARAMETER DEFINITIONS:
    NO.   NAME         VALUE      STEP SIZE      LIMITS
     4 Nb          1.00000e+03  1.00000e+01     no limits
 **********
 **    3 **FIX          4
 **********
 **********
 **    4 **MIGRAD
 **********
 FIRST CALL TO USER FUNCTION AT NEW START POINT, WITH IFLAG=4.
 START MIGRAD MINIMIZATION.  STRATEGY  1.  CONVERGENCE WHEN EDM .LT. 1.00e-04
 FCN=-3224.14 FROM MIGRAD    STATUS=INITIATE        2 CALLS           3 TOTAL
                     EDM= unknown      STRATEGY= 1      NO ERROR MATRIX
  EXT PARAMETER               CURRENT GUESS       STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1    Ns          1.00000e+02   1.00000e+01   1.00000e+01  -5.49348e-03
   2    Mu          4.00000e+00     fixed
   3    Sigma       2.00000e-02     fixed
   4    Nb          1.00000e+03     fixed
 MIGRAD MINIMIZATION HAS CONVERGED.
 MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
 COVARIANCE MATRIX CALCULATED SUCCESSFULLY
 FCN=-3224.15 FROM MIGRAD    STATUS=CONVERGED      12 CALLS          13 TOTAL
                     EDM=1.22711e-08    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1    Ns          1.00392e+02   1.19765e+01   3.31155e-01  -1.30806e-05
   2    Mu          4.00000e+00     fixed
   3    Sigma       2.00000e-02     fixed
   4    Nb          1.00000e+03     fixed
 EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  1    ERR DEF=1
  1.434e+02
```

Migrad started

Migrad has converged

Min.FCN

Migrad results

Tutorial Intro – step by step – 2. fit

```
**********
**    5 **HESSE
**********
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-3224.15 FROM HESSE     STATUS=OK              5 CALLS          18 TOTAL
                    EDM=1.22603e-08    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR       STEP SIZE       VALUE
  1    Ns                    1.00392e+02   1.19765e+01   1.32462e-02   1.00392e+02
  2    Mu                    4.00000e+00     fixed
  3    Sigma                 2.00000e-02     fixed
  4    Nb                    1.00000e+03     fixed
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  1    ERR DEF=1
 1.434e+02
**********
**    6 **MINOS
**********
FCN=-3224.15 FROM MINOS     STATUS=SUCCESSFUL      4 CALLS          22 TOTAL
                    EDM=1.22603e-08    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                  PARABOLIC         MINOS ERRORS
 NO.   NAME      VALUE            ERROR      NEGATIVE      POSITIVE
  1    Ns                    1.00392e+02   1.19765e+01  -1.16290e+01   1.23283e+01
  2    Mu                    4.00000e+00     fixed
  3    Sigma                 2.00000e-02     fixed
  4    Nb                    1.00000e+03     fixed
```

Start determination of HESSE matrix

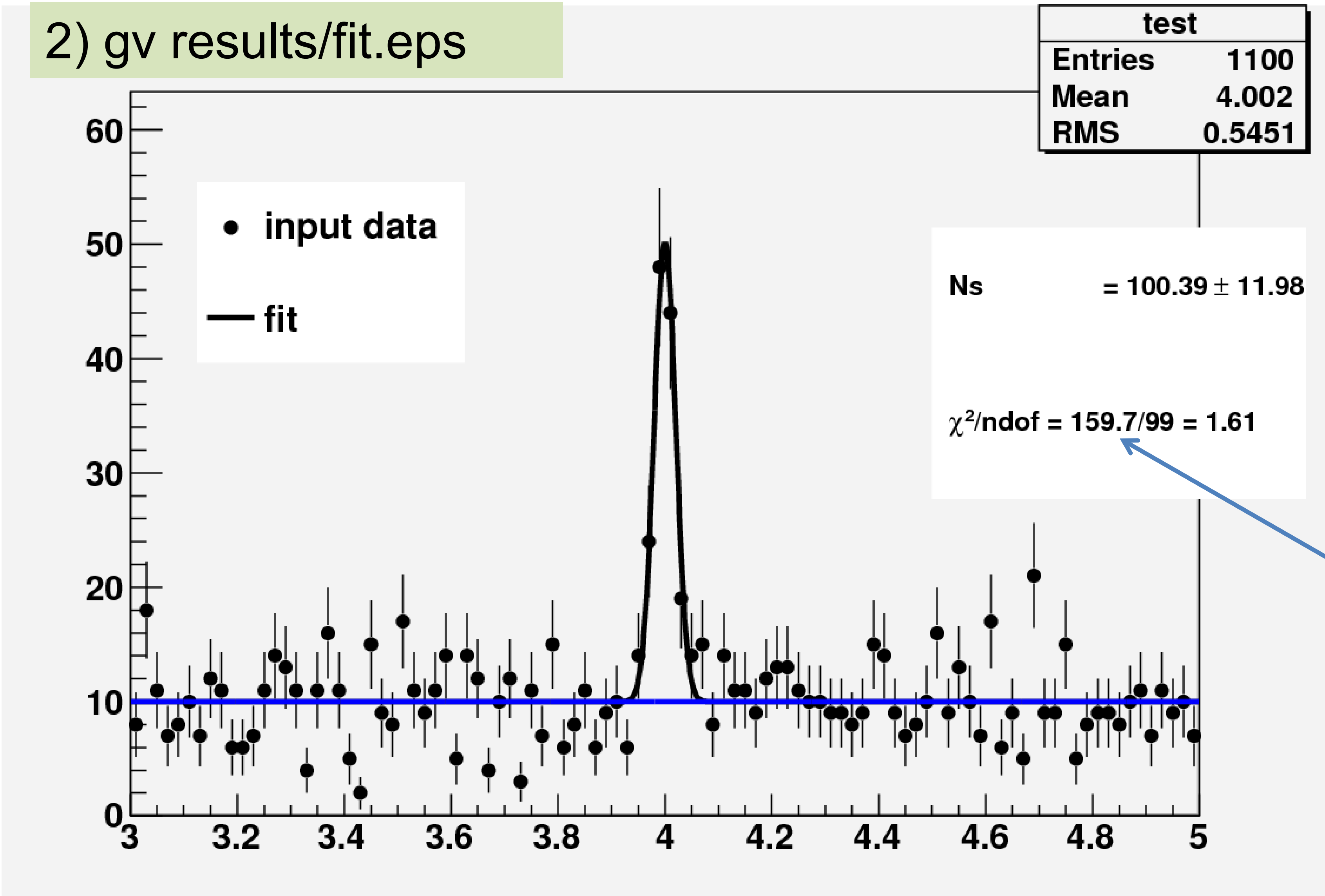
Errors on Ns from HESSE

Start MINOS Chi2= Chi2_min+1 search

Minos negative error

Minos positive error

Tutorial Intro – step by step – 2. fit



Neyman χ^2 for control of fit quality

Tutorial Intro – step by step – 2. fit

- Steering in config/mass_peak_fit/gauss_pol0

```
# specify input data
input_file            test.root
input_histogram       test

# possible values: gauss, pol0, pol3
# Any combination of these functions works, separate them with '+'.
fit_function          gauss+pol0
normalised_gauss      1
# possible values: chi2_pearson, chi2_neyman, log_likelihood
fit_type              log_likelihood
fit_range             3 5
start_values          100   4.0      0.02     1000
fix_parameters        0       1        1        1
lower_limit           0       0        0        0
upper_limit           0       0        0        0
step                  10      0.1      0.1      10
#covariance_ellipse_parameters 1 4
only_draw_initial     0
```

Choose Estimator

Note:log_likelihood means using -2 ln L

Nb

Fixed par.

Decomment for plotting chi2 contours

Fitted par.

Ns

mu

sigma

Tutorial Intro – step by step – 2. fit

- Fit Signal and background in the pseudo data
  - Mass_peak_fit.exe*
  - Steering in config/mass_peak_fit/gauss_pol0
  - Call to MINUIT commands in src/Fit.cxx

1) ./runfit.sh — Compile and fit – watch the screen output

2) gv results/fit.eps — Look at fitted data

3) emacs config/mass_peak_fit/gauss_pol0 — Edit fit steering file

4) Change some parameter, save and repeat 1) and 2) see effect?

5) Restore default version, e.g. by
cp config/mass_peak_fit/gauss_pol0_ref config/mass_peak_fit/gauss_pol0

Tutorial Intro – step by step – 2. fit

- Call to MINUIT commands in src/Fit.cxx

```
// ***********  USER EDITABLE PART  *********** //

// perform the minimization!
fMinuit -> Command("MIGRAD");
fMinuit -> Command("HESSE");
fMinuit -> Command("MINOS");

// ********  END OF USER EDITABLE PART  ******* //
```

6) emacs src/Fit.cxx
Add command line before the other three comands:
fMinuit -> Command(“Set Pri 3”);

1) ./runfit.sh

watch the **changed** screen output
ok, it’s somewhat tedious ☹
Take it out again then

Gauss_pol0 Task 1a: fit only Ns

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**

- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**

- Run script: **./run.sh**

Task 1a): Error estimates

- Fill the fitted Ns and its error from Hesse and MINOS in the table below
  (get all results from Minuit printout)

Ns	**Nb**	**Fitted Ns**	**Hesse** $\boldsymbol{\Delta(Ns)}$	**Minos** $\boldsymbol{\Delta Ns -}$	**Minos** $\boldsymbol{\Delta Ns +}$
100	1000				

Can you explain why the MINOS errors are asymmetric?

Gauss_pol0 Task 1a: fit only Ns SOL

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**

- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**

- Run script: **./run.sh**

Task 1a): Error estimates

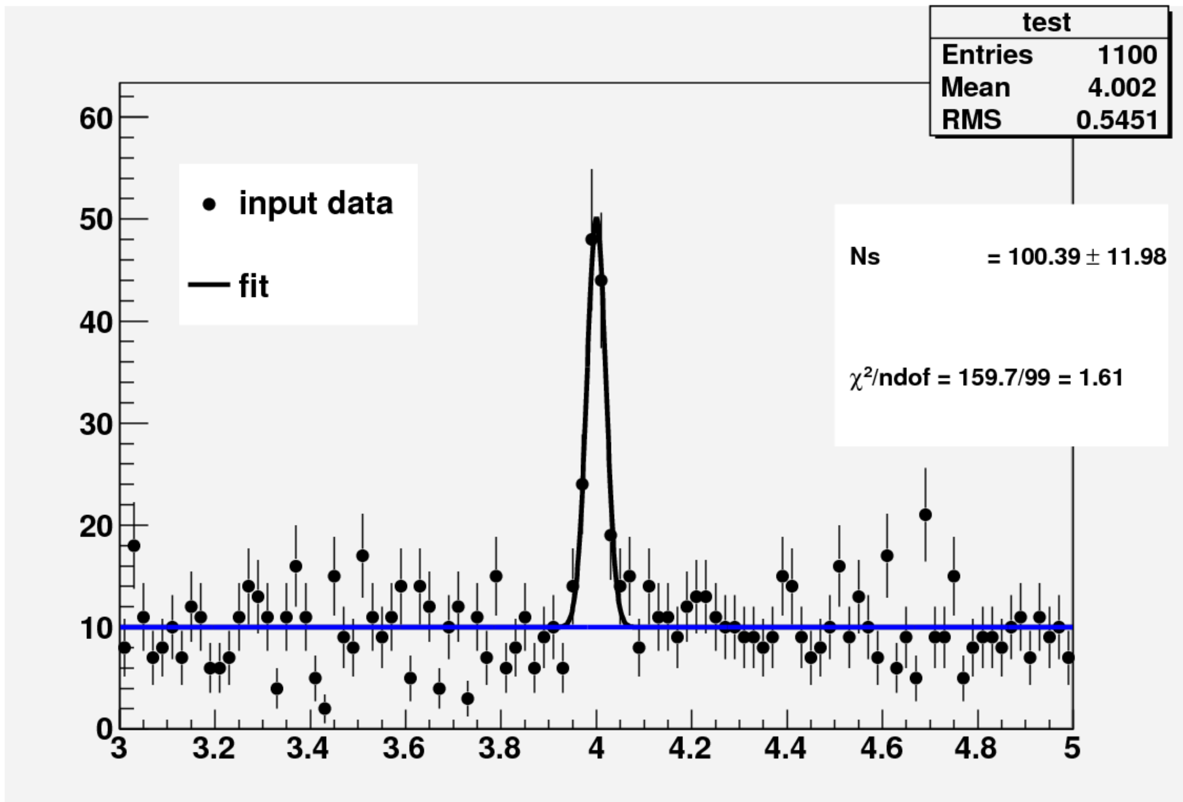
- Fill the fitted Ns and its error from Hesse and MINOS in the table below (get all results from Minuit printout).

Ns	**Nb**	**Fitted Ns**	**Hesse $\Delta(Ns)$**	**Minos $\Delta Ns-$**	**Minos $\Delta Ns+$**
100	1000	100.4	11.9	-11.6	12.3
10	10	9.15	3.14	-2.8	3.5

Can you explain why the MINOS errors are asymmetric?

➤ **Due to Poisson Likelihood**

Gauss_pol0 Task 1a: fit only Ns SOL



- Error on Ns ~ sqrt(NT) with NT = total number of events in peak region

Gauss_pol0 Task 1b: fit only Ns

Task 1b): Migrad convergence

- Change **the fit start value for Ns** to different values (see table below) and rerun run.sh for each value
- Check in the printout messages if MIGRAD converged and mark in the table below

Ns	**Nb**	**Fit start value for Ns**	**Migrad converged?**
100	1000	**100**	
100	1000	**300**	
100	1000	**10**	

What is the “fitted” Ns value (given in the Minuit printout) if MIGRAD did **not** converge?

Gauss_pol0 Task 1b: fit only Ns SOL

Task 1b): Migrad convergence

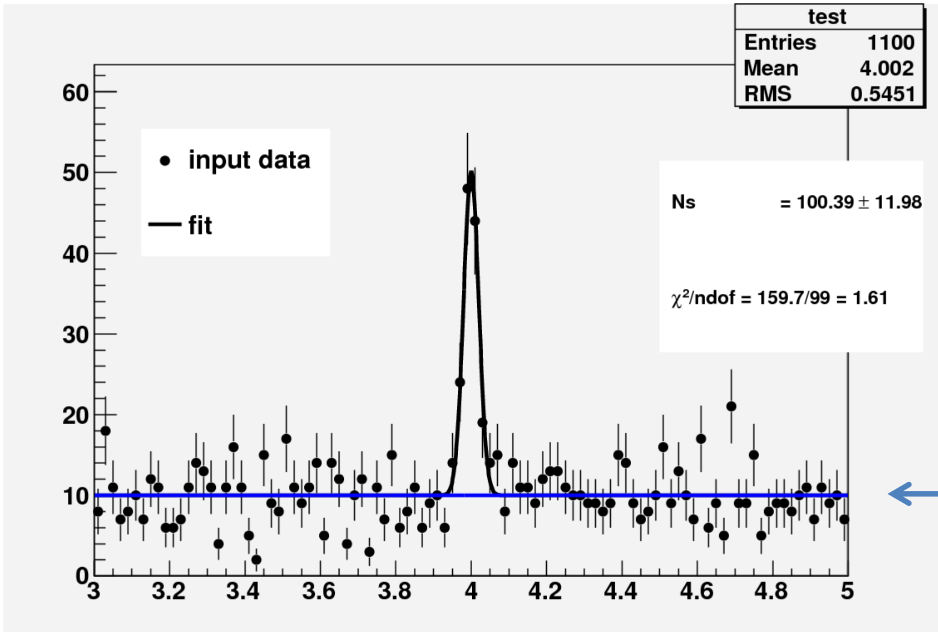
- Change **the fit start value for Ns** to different values (see table below) and rerun run.sh for each value
- Check in the printout messages if MIGRAD converged and mark in the table below

Ns	Nb	Fit start value for Ns	Migrad converged?
100	1000	**100**	yes
100	1000	**300**	no
100	1000	**200**	yes
100	1000	**10**	yes

What is the "fitted" Ns value (given in the Minuit printout) if MIGRAD did **not** converge?

The initial parameter values are taken!

Gauss_pol0 Task 2: fit Ns and Nb



Nb = ???

- Depending on the process Nb can be estimated from
  - NLO QCD or QEW
  - from **MCs using scaling factors determined in background enriched samples (can be risky!)**
  - fitting the mass spectrum → what we will do ☺

Gauss_pol0 Task 2a: fit Ns and Nb

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**
  Change: fix parameters 0 1 1 0
  covariance_ellipse_parameters 1 4
- Run script: **./run.sh**

- Fitting: src/Fit.cxx
  Change to:
  fMinuit -> Command("MIGRAD");
  fMinuit -> Command("HESSE");
  fMinuit -> Command("Set Pri -1");

Task 2a): Correlation of parameters

- Fill fitted Ns, Nb, errors **and correlation coefficient** ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters (shown are contours for values FCN_min+0.25,+1,+2.25,+4,+6.25,+9): **gv results/cov_ellipse.eps**
  → are the two parameters correlated or anticorrelated? (and can you explain why?)

Ns	Nb	Fitted Ns	Fitted Nb	$\Delta(Ns)$	$\Delta(Nb)$	Correlation Coefficient (ρ)
100	1000					

Gauss_pol0 Task 2a: fit Ns and Nb SOL

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**
  Change: fix parameters 0 1 1 0
  covariance_ellipse_parameters 1 4
- Run script: **./run.sh**

- Fitting: src/Fit.cxx
  Change to:
  fMinuit -> Command("MIGRAD");
  fMinuit -> Command("HESSE");
  fMinuit -> Command("Set Pri -1");

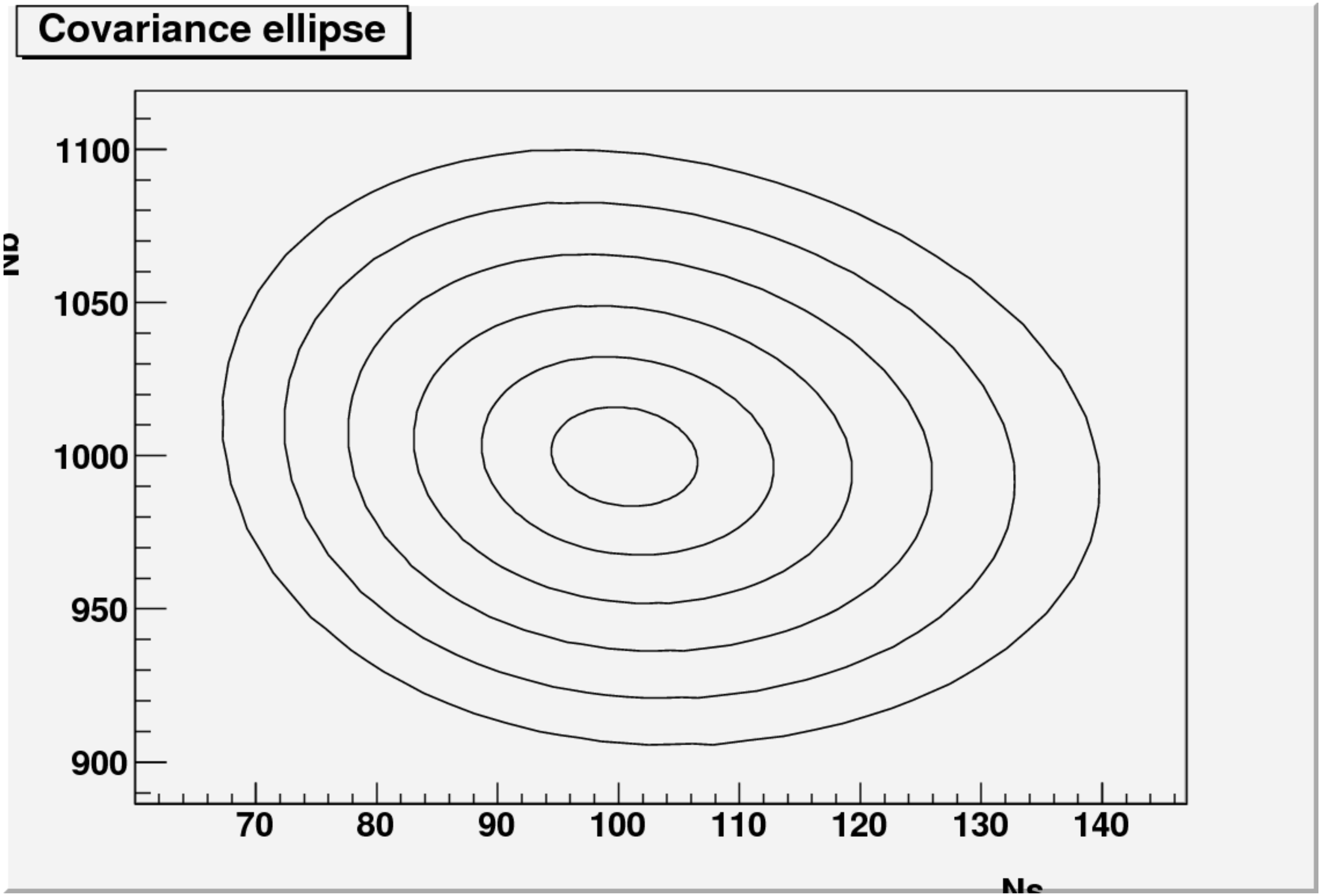
Task 2a): Correlation of parameters

- Fill fitted Ns, Nb, errors **and correlation coefficient ρ** in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters (shown are contours for values FCN_min+0.25,+1,+2.25,+4,+6.25,+9): **gv results/cov_ellipse.eps**
  → are the two parameters correlated or anticorrelated? (and can you explain why?)

Ns	Nb	Fitted Ns	Fitted Nb	$\Delta(Ns)$	$\Delta(Nb)$	Correlation Coefficient (ρ)
100	1000	100.4	99.9	12.1	32.2	-0.115

They are anticorrelated: compensate background increase by lowering signal

Gauss_pol0 Task 2a: fit Ns and Nb SOL



- Asymmetric errors (Poisson stat.)

Gauss_pol0   Task 2b: fit Ns and Nb

Task 2b): Correlations and fit range

- Perform the fit for various fit ranges and fill the correlation of Ns and Nb in the table.
- Note: the fit range is steered in the fit steering, e.g. fit_range 3.5 4.5

Fit range	$\boldsymbol{\rho}$
3 5	
3.5 4.5	
3.95 4.05	

Why does the anticorrelation increase for smaller fit ranges?

Gauss_pol0 Task 2b: fit Ns and Nb SOL

Task 2b): Correlations and fit range

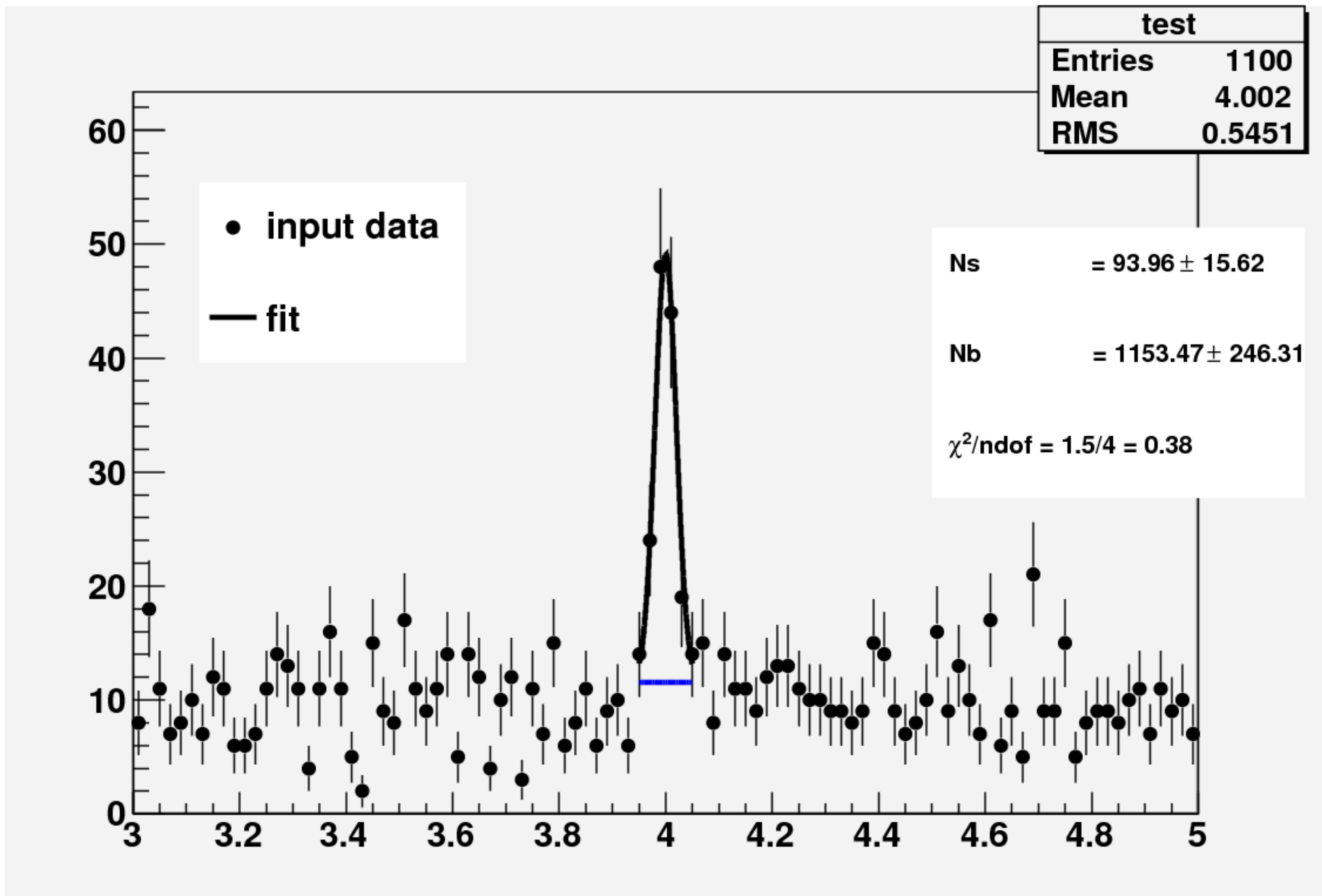
- Perform the fit for various fit ranges and fill the correlation of Ns and Nb in the table.
- Note: the fit range is steered in the fit steering, e.g. fit_range 3.5 4.5

Fit range	ρ
3 5	-0.115
3.5 4.5	-0.166
3.9 4.1	-0.431
3.95 4.05	-0.648

Why does the anticorrelation increase for smaller fit ranges? **In small fit range around peak the signal and background cannot be distinguished anymore (but effect is only strong for very narrow range!)**

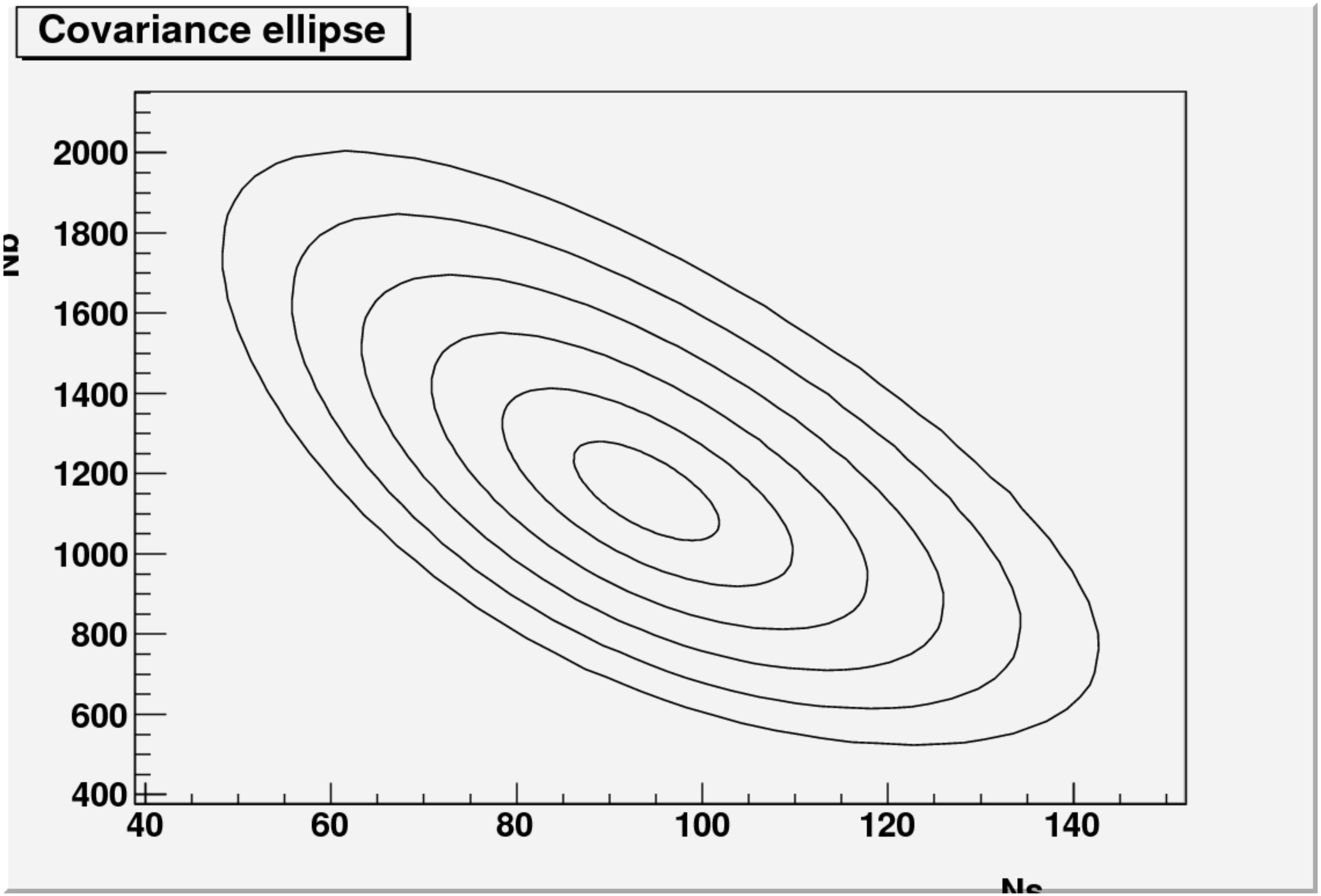
Gauss_pol0 Task 2b: fit Ns and Nb SOL

Fit from 3.95 to 4.05



Gauss_pol0 Task 2b: fit Ns and Nb SOL

Fit from 3.95 to 4.05



Gauss_pol0 Task 2c: fit Ns and Nb

Task 2c): Log likelihood vs Neyman chi2 and Pearson chi2

- Switch back the fit range to: fit_range 3 5
- Run the fit with log_likelihood, chi2_neyman and chi2_pearson, for this change in the fit steering: fit_type log_chi2_neyman etc.
- Fill FCN_min, fitted Ns and Nb and errors in the table and the sum Ns+Nb
- Have also (for fun) a look at the mass peak fits: **gv results/fit.eps**

Ns	Nb	Estimator	FCN_min	Fitted Ns	Fitted Nb	$\Delta(Ns)$	$\Delta(Nb)$	Fitted Ns+Nb
100	1000	Log_likelihood						
100	1000	chi2_neyman						
100	1000	chi2_pearson						

Do the fits look ok? Can you see any biases in the results?

Gauss_pol0 Task 2c: fit Ns and Nb SOL

Task 2c): Log likelihood vs Neyman chi2 and Pearson chi2

- Switch back the fit range to: fit_range 3 5
- Run the fit with log_likelihood, chi2_neyman and chi2_pearson, for this change in the fit steering:
  fit_type log_chi2_neyman etc
- Fill FCN_min, fitted Ns and Nb and errors in the table and the sum Ns+Nb
- Have also (for fun) a look at the mass peak fits: **gv results/fit.eps**

Ns	**Nb**	**Estimator**	**FCN_min**	**Fitted Ns**	**Fitted Nb**	$\Delta(Ns)$	$\Delta(Nb)$	**Fitted Ns+Nb**
100	1000	Log_likelihood	-3224.15	100.4	999.6	12.1	32.3	1100
100	1000	chi2_neyman	137.1	105.4	857.5	12.0	30.0	962.9
100	1000	chi2_pearson	111.7	98.5	1057.4	12.1	33.2	1155.9

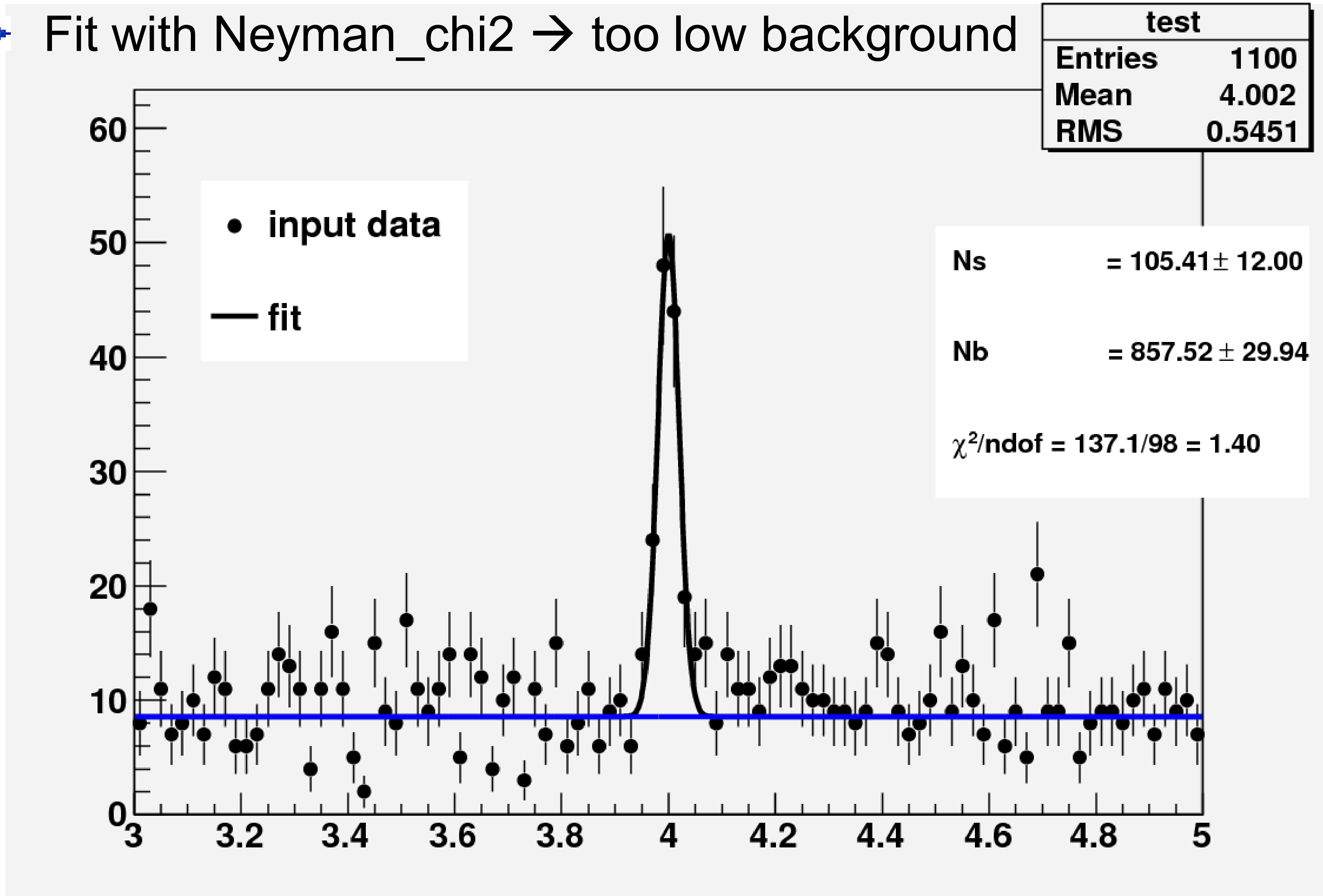
- **Fitted Ns+Nb – generated Ns+Nb:**

= 0 (log_likelihood); = -chi2 (chi2_neyman); = +chi2/2 (chi2_pearson);

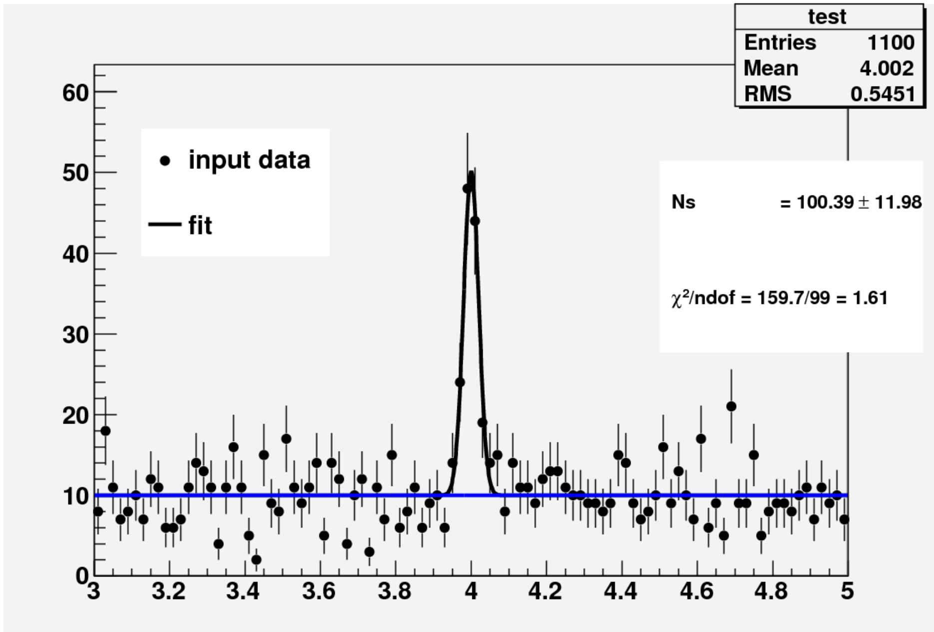
Note: these relations hold in general!

Gauss_pol0 Task 2c: fit Ns and Nb SOL

- Fit with Neyman_chi2 → too low background



Gauss_pol0   Task 3: fit Ns and mu



mu = ???

- Particle masses crucial parameters in the SM, e.g. Higgs mass
- Detectors: for track based mass peaks e.g. D0→ K-pi+ one can usually achieve high precision, often limited only by systematics

Gauss_pol0 Task 3a: fit Ns and mu

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref (don't forget to switch back to log_likelihood)**
  Change: fix parameters 0 0 1 1
  covariance_ellipse_parameters 1 2
- Run script: **./run.sh**

Task 3a): Lets do the fit!

- Fill the fitted Ns, mu, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters: **gv results/cov_ellipse.eps**

Ns	Nb	Fitted Ns	Fitted mu	$\Delta(Ns)$	$\Delta(mu)$	ρ
100	1000					

The lowest possible error on mu for a background free sample is $\Delta(mu)$ = sigma/sqrt(Ns) where sigma=0.02 is the fixed peak width. How much larger is our error?

Gauss_pol0 Task 3a: fit Ns and mu SOL

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**
  Change: fix parameters 0 0 1 1
  covariance_ellipse_parameters 1 2
- Run script: **./run.sh**

Task 3a): Lets do the fit!

- Fill the fitted Ns, mu, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters **gv results/cov_ellipse.eps**

Ns	Nb	Fitted Ns	Fitted mu	$\Delta(Ns)$	$\Delta(mu)$	ρ
100	1000	100.5	3.9977	12.0	0.0029	0.002

The lowest possible error on mu for a background free sample is $\Delta(mu)$ = sigma/sqrt(Ns) where sigma=0.02 is the fixed peak width. How much larger is our error?

Sigma/sqrt(Ns) = error on mean value for background free peak, our error is 45% larger

Gauss_pol0 Task 3b: fit Ns and mu

Task 3b): Stupid start values ☺

- Change start values in fit steering: start_values 10 4.4 0.02 1000 and rerun the fit
- Fill fitted Ns, mu, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the fit: **gv results/fit.eps**

Ns	Nb	Fitted Ns	Fitted mu	$\Delta(Ns)$	$\Delta(mu)$	ρ
100	1000					

How do you judge the fit quality?
Could you rule out that the fitted local peak is a real one?

Gauss_pol0 Task 3b: fit Ns and mu SOL

Task 3b): Stupid start values ☺

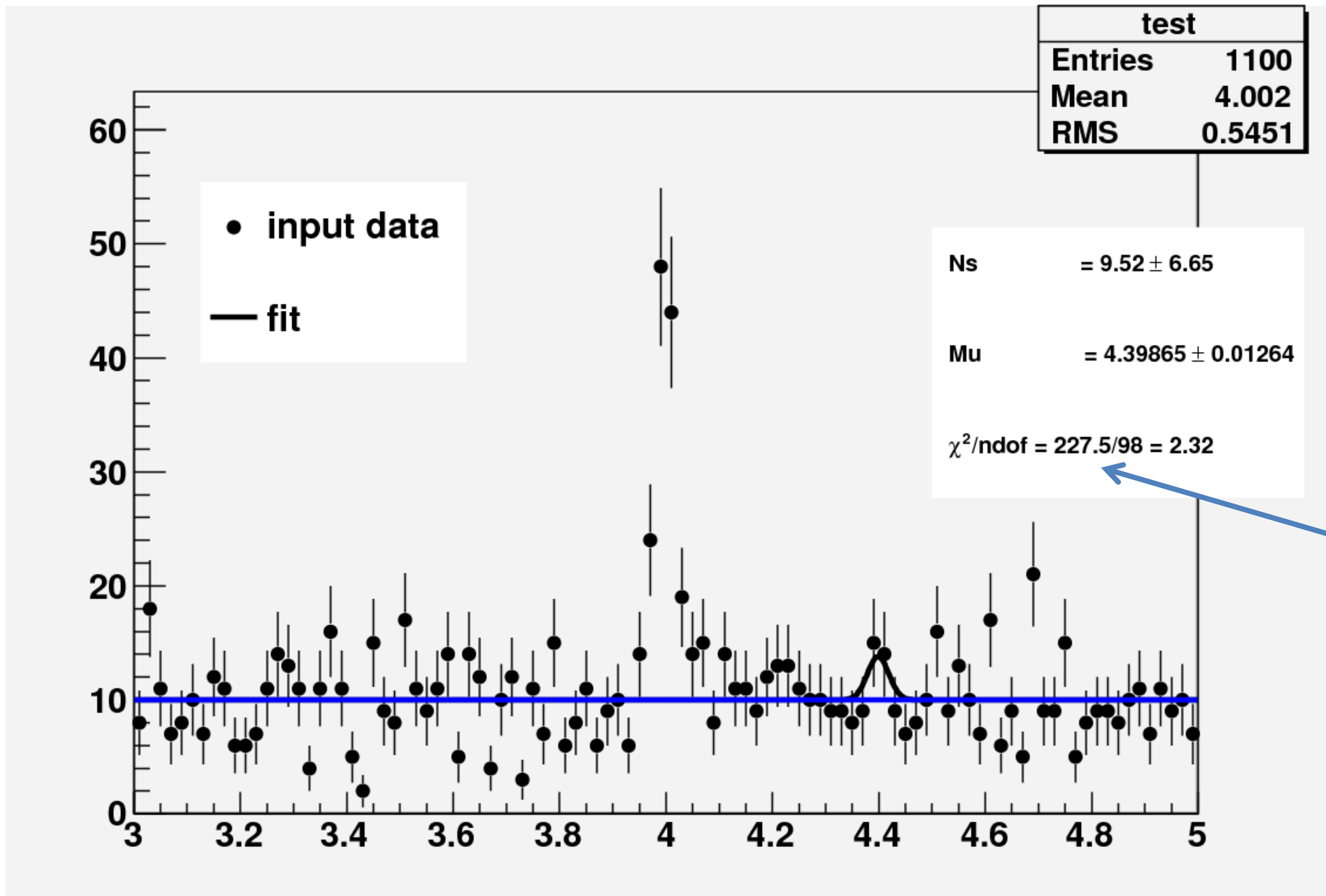
- Change start values in fit steering: start_values 10 4.4 0.02 1000 and rerun the fit
- Fill fitted Ns, mu, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the fit: **gv results/fit.eps**

Ns	Nb	Fitted Ns	Fitted mu	$\Delta(Ns)$	$\Delta(mu)$	ρ
100	1000	9.5	4.40	6.6	0.01	0.001

How do you judge the fit quality?
Could you rule out that the fitted local peak is a real one?

The overall fit description poor, in particular at true peak mass at 4.0, but locally around 4.4 it is ok, cannot rule out a peak there, but it is not significant either

Gauss_pol0 Task 3b: fit Ns and mu SOL



Bad Neyman chi2

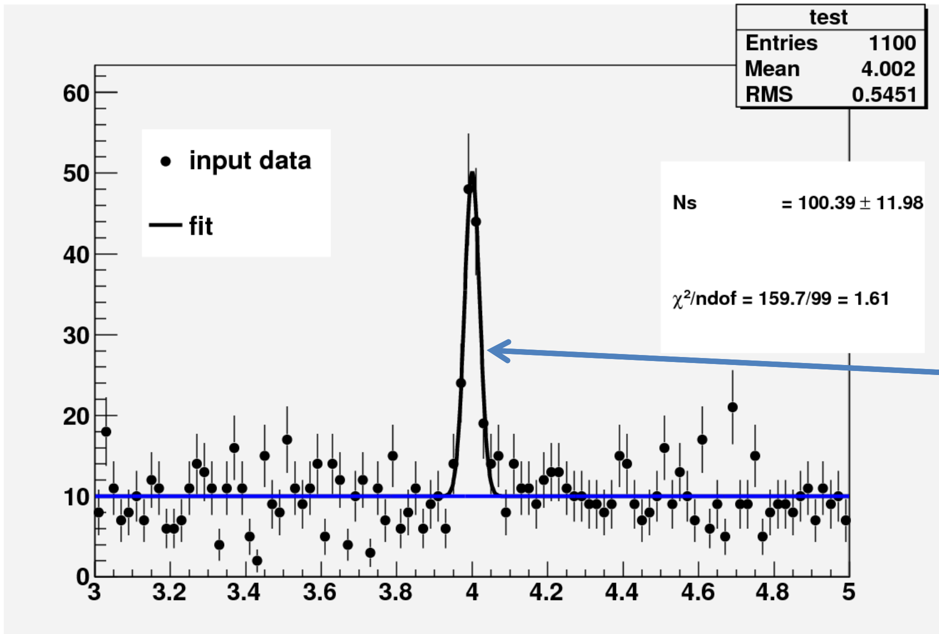
Gauss_pol0 Task 3c: fit Ns and mu

Task 3c): Fit challenge – try to find most significant side peak

- Try yourself different start values **Ns** and **mu** in fit steering:
  start_values Ns mu 0.02 1000 and rerun the fit, try to find the most significant side peak and fill in table

Ns	**Nb**	**Fitted Ns**	**Fitted mu**	$\boldsymbol{\Delta(Ns)}$	$\boldsymbol{\Delta(mu)}$
100	1000				

Gauss_pol0 Task 4: fit Ns and sigma



sigma = ???

Two contributions to observable width of peak:

- 1. Natural width of particle, Breit Wigner Resonance
- 2. Detector resolution (often dominating, e.g. D0→ K-pi+)
  - often not perfectly modelled by MC, need to fit to data
  - often simplified: parametrise by single gaussian, rarely correct, but might be ok for very small statistics

Gauss_pol0 Task 4a: fit Ns and sigma

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**
  Change: fix parameters 0 1 0 1
  covariance_ellipse_parameters 1 3
- Run script: **./run.sh**

Task 4a): Lets do the fit!

- Fill the fitted Ns, sigma, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters: **gv results/cov_ellipse.eps**

Ns	**Nb**	**Fitted Ns**	**Fitted sigma**	$\boldsymbol{\Delta(Ns)}$	$\boldsymbol{\Delta(sigma)}$	$\boldsymbol{\rho}$
100	1000					

The lowest possible error on sigma for a background free sample is $\Delta(sigma)$ = sigma/sqrt(2(Ns-1))
How much larger is our error?

Gauss_pol0 Task 4a: fit Ns and sigma SOL

- Generation: config/generate_data/gauss_pol0
  → **Use reference gauss_pol0_ref**
- Fitting: config/mass_peak_fit/gauss_pol0
  → **Use reference gauss_pol0_ref**
  Change: fix parameters 0 1 0 1
  covariance_ellipse_parameters 1 3
- Run script: **./run.sh**

Task 4a): Lets do the fit!

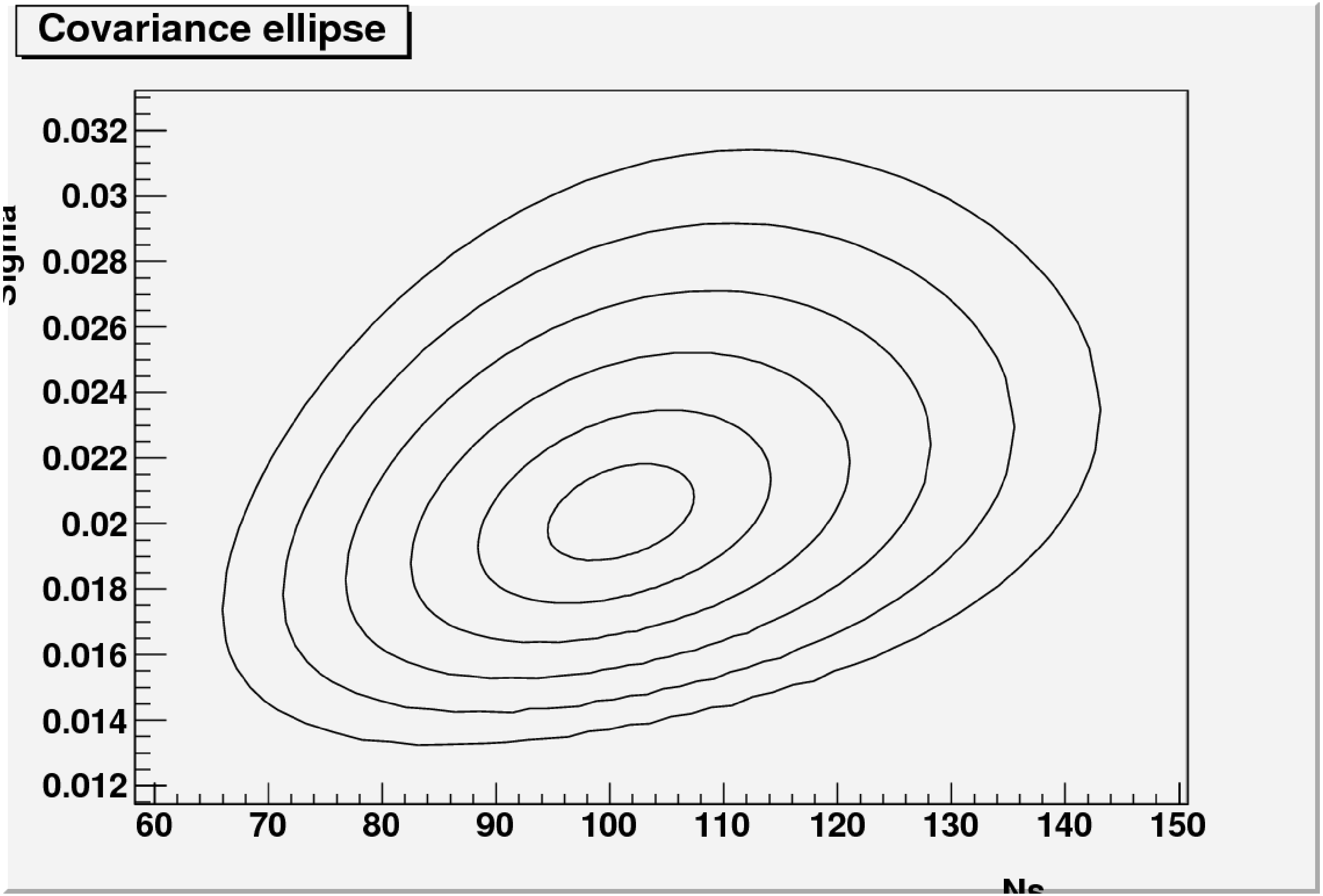
- Fill the fitted Ns, sigma, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters: **gv results/cov_ellipse.eps**

Ns	Nb	Fitted Ns	Fitted sigma	$\Delta(Ns)$	$\Delta(sigma)$	ρ
100	1000	100.9	0.0203	12.8	0.0029	0.354

The lowest possible error on sigma for a background free sample is $\Delta(sigma)$ = sigma/sqrt(2(Ns-1))

How much larger is our error? **sigma/sqrt(2(Ns-1)) = 0.0014 , our error is ~two times larger**

Gauss_pol0 Task 4a: fit Ns and sigma SOL



Gauss_pol0 Task 4b: fit Ns and sigma

Task 4b): Switch to non-normalised gaussian function

- We will fit the signal now with $gu(m) = \boldsymbol{S}\,\mathbf{exp}(-\frac{(\boldsymbol{m}-\boldsymbol{mu})^2}{\boldsymbol{2sigma}^2})$
- Change in the fit steering: **normalised_gauss 0**
- Fill the fitted S, sigma, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters: **gv results/cov_ellipse.eps**

Ns	**Nb**	**Fitted S (Maximum value)**	**Fitted sigma**	$\boldsymbol{\Delta(S)}$	$\boldsymbol{\Delta(sigma)}$	$\boldsymbol{\rho}$
100	1000					

Can you explain why the correlation is negative and has a large value?

Gauss_pol0 Task 4b: fit Ns and sigma SOL

Task 4b): Switch to non-normalised gaussian function

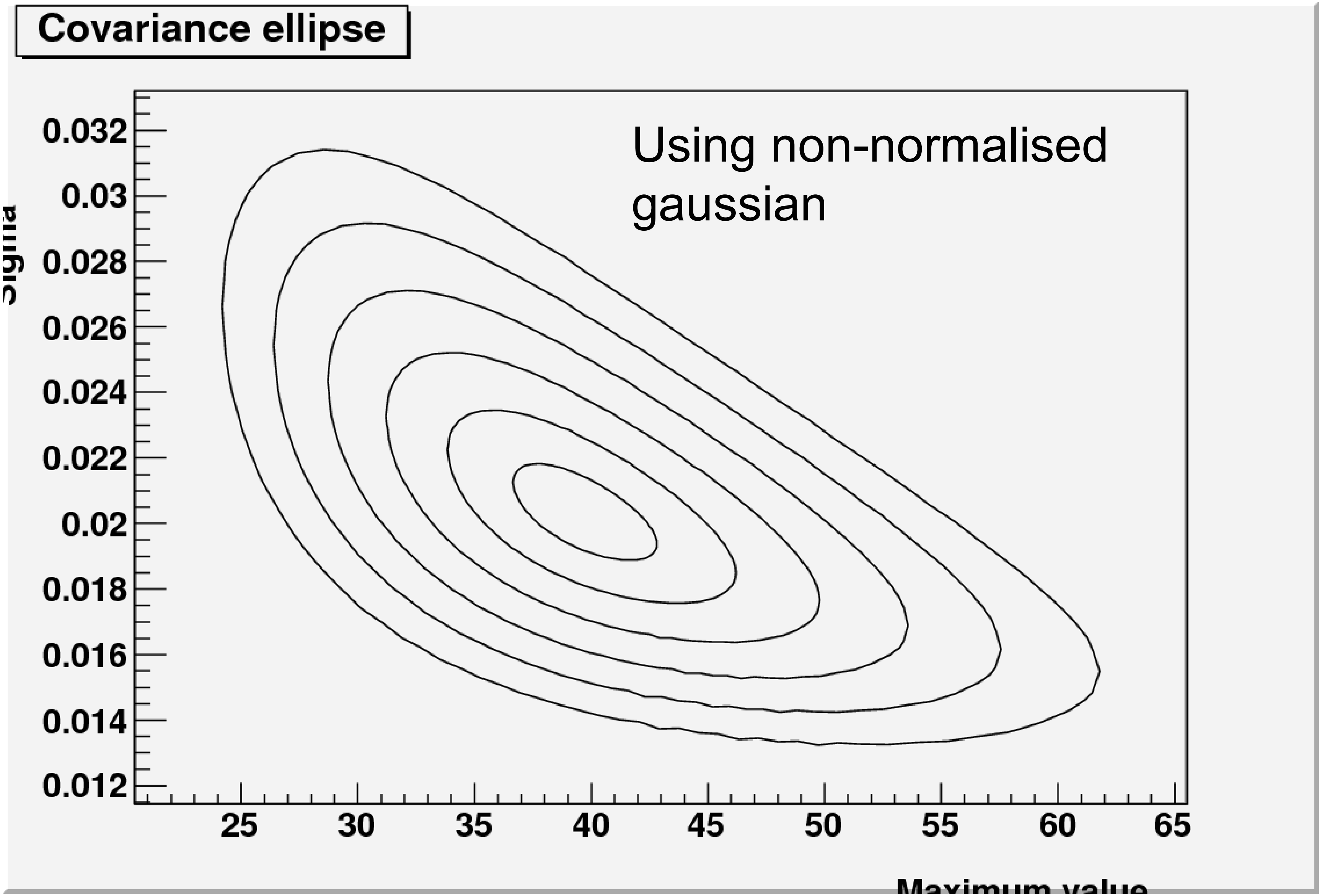
- We will fit the signal now with $gu(m) = \boldsymbol{S}\,\mathbf{exp}(-\frac{(\boldsymbol{m}-\boldsymbol{mu})^2}{\mathbf{2}\boldsymbol{sigma}^2})$
- Change in the fit steering: **normalised_gauss 0**
- Fill the fitted S, sigma, errors and correlation coefficient ρ in the table (get all results from Minuit printout)
- Have a look at the FCN contours around the fitted parameters: **gv results/cov_ellipse.eps**

Ns	**Nb**	**Fitted S (Maximum value)**	**Fitted sigma**	$\boldsymbol{\Delta(S)}$	$\boldsymbol{\Delta(sigma)}$	$\boldsymbol{\rho}$
100	1000	39.6	0.00203	6.2	0.0029	-0.602

Can you explain why the correlation is negative and has a large value?

- **For non-normalised function increasing the sigma will increase the integral of the function and this can be compensated by decreasing S.**
- **Note: the relative error for signal was 12.7% for normalised gaussian and has increased to 15.6%. This can be explained by factor 1/sqrt(1-rho**2) = 1.25**
- **One can restore the proper number of signal events: Ns ~ S sigma sqrt(2pi) and when applying error propagation the error on Ns will be again 12.7% (due to the anti-correlation of Ns and S). ALWAYS USE NORMALISED GAUSSIAN! (be on the bright side of life!)**

Gauss_pol0 Task 4b: fit Ns and sigma SOL



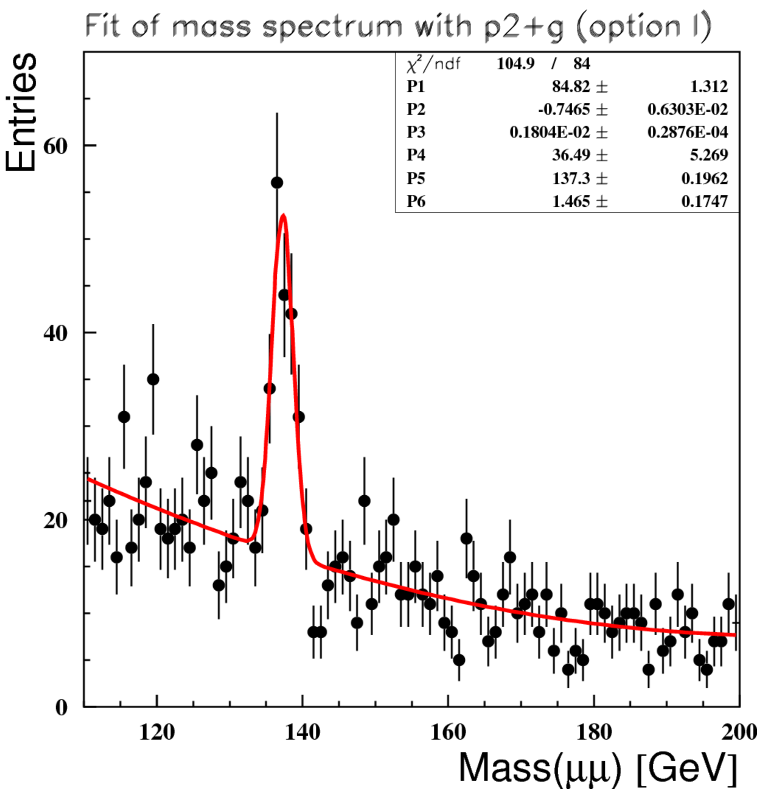
Gauss_pol3 Task 5: fit Ns and sigma

- Generation: config/generate_data/gauss_pol3
- Fitting: config/mass_peak_fit/gauss_pol3
- Run script: **./run.sh Change to**
  make
  .generate_data.exe - - config gauss_pol3
  .mass_peak_fit.exe - - config gauss_pol3

Task 5): Lets do the fit!

- Run the fit ./run.sh with the default fit steering, does it look good?
- Try to play with the steering parameters in config/mass_peak_fit/gauss_pol3 to obtain a reasonable fit of signal and background, e.g.
  - free fixed parameters for the polynomial pol3 (in the default steering only the constant term is **not** fixed)
  - and/or change start values of parameters.
- Note down a configuration (which parameters are fixed and start values) for which you are happy with the resulting fit!

Mass peak fit tutorial Summary



- Mass peak fitting **is an art:**
- Use Poisson likelihood
- Always take a look if the fit curve describes the data
- Neyman-chi2 often useful as another fit quality check
- Parameter start values are crucial
- Consider carefully signal and background parametrisation (Don't add extra parameters that do not substantially decrease -2 lnL)
- +Many more aspects (fit range, binning etc.)

Thanks for your attention!