6. miniworkshop difrakce a ultraperiferálních srážek

Fitting with correlated uncertainties

Guillermo Contreras

Czech Technical University in Prague

Děčín

September 14, 2023

Parameter estimation

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

There may be many parameters

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

There may be many parameters

The data may be multidimensional

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

There may be many parameters

The data may be multidimensional

An estimator $\hat{a}(x_1, \ldots, x_n)$ is a function of the data that gives a value for the parameter a

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

There may be many parameters

The data may be multidimensional

An estimator $\hat{a}(x_1, \ldots, x_n)$ is a function of the data that gives a value for the parameter a

An ideal estimator would be

- Consistent. $\hat{a}(x_1 \ldots x_N) \to a$ as $N \to \infty$,
- Unbiased: $\langle \hat{a} \rangle = a$,
- Efficient: $\langle (\hat{a} - a)^2 \rangle$ is as small as possible,
- Invariant: $\hat{f}(a) = f(\hat{a})$.

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Parameter estimation

Suppose that you know the probability density function $P(x; a)$ for some process and you have a data set $\{x_i\}$

What is the best value of a?

There may be many parameters

The data may be multidimensional

An estimator $\hat{a}(x_1, \ldots, x_n)$ is a function of the data that gives a value for the parameter a

There is no such a thing

An ideal estimator would be

- Consistent. $\hat{a}(x_1 \ldots x_N) \to a$ as $N \to \infty$,
- Unbiased: $\langle \hat{a} \rangle = a$,
- Efficient: $\langle (\hat{a} - a)^2 \rangle$ is as small as possible,
- Invariant: $\hat{f}(a) = f(\hat{a})$.

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Maximum likelihood

The maximum likelihood (ML) estimator adjusts a so that the likelihood of the sample is maximised

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Maximum likelihood

The maximum likelihood (ML) estimator adjusts a so that the likelihood of the sample is maximised

ML is

- It is consistent.
- It is biased, but bias falls like $1/N$.
- It is efficient for the large N.
- It is invariant—doesn't matter if you reparametrize a.

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Least squares estimation

If you have Gaussian measurements of y taken at various x values with a measurement error σ, and a prediction $y = f(x; a)$ then

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Least squares estimation

If you have Gaussian measurements of y taken at various x values with a measurement error σ, and a prediction $y = f(x; a)$ then

Gaussian probabilities

$$P(y; x, a) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(y - f(x; a))^2}{2\sigma^2}}$$

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Least squares estimation

If you have Gaussian measurements of y taken at various x values with a measurement error σ, and a prediction $y = f(x; a)$ then

Gaussian probabilities

$$P(y; x, a) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(y - f(x;a))^2}{2\sigma^2}}$$

→

Log-likelihood

$$\ln L = -\sum \frac{(y - f(x;a))^2}{2\sigma^2} + \text{constants}$$

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Least squares estimation

If you have Gaussian measurements of y taken at various x values with a measurement error σ, and a prediction $y = f(x; a)$ then

Gaussian probabilities

$$P(y; x, a) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(y - f(x;a))^2}{2\sigma^2}}$$

$\rightarrow$

Log-likelihood

$$\ln L = -\sum \frac{(y - f(x;a))^2}{2\sigma^2} + \text{constants}$$

To maximise $\ln L$ is equivalent to minimise:

χ^2 minimisation

$$\chi^2 = \sum \frac{(y - f(x;a))^2}{\sigma^2}$$

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Least squares estimation

If you have Gaussian measurements of y taken at various x values with a measurement error σ, and a prediction $y = f(x; a)$ then

Gaussian probabilities

$$P(y; x, a) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(y - f(x;a))^2}{2\sigma^2}}$$

Log-likelihood

$$\ln L = -\sum \frac{(y - f(x;a))^2}{2\sigma^2} + \text{constants}$$

To maximise $\ln L$ is equivalent to minimise:

χ^2 minimisation

$$\chi^2 = \sum \frac{(y - f(x;a))^2}{\sigma^2}$$

In case of a histogram with n entries, the uncertainty, given by Poisson statistics, is $\sigma^2 = n$ which breaks down for empty bins

R. J. Barlow, Practical statistics for particle physics
https://e-publishing.cern.ch/index.php/CYRSP/article/view/1384

Lagrange multipliers

In mathematical optimization, the **method of Lagrange multipliers** is a strategy for finding the local maxima and minima of a function subject to equation constraints (i.e., subject to the condition that one or more equations have to be satisfied exactly by the chosen values of the variables).[1] It is named after the mathematician Joseph-Louis Lagrange. The basic idea is to convert a constrained problem into a form such that the derivative test of an unconstrained problem can still be applied. The relationship between the gradient of the function and gradients of the constraints rather naturally leads to a reformulation of the original problem, known as the **Lagrangian function**.[2]

The method can be summarized as follows: In order to find the maximum or minimum of a function $f(x)$ subjected to the equality constraint $g(x) = 0$, form the Lagrangian function,

$$\mathcal{L}(x, \lambda) \equiv f(x) + \lambda \cdot g(x)$$

and find the stationary points of $\mathcal{L}$ considered as a function of x and the Lagrange multiplier λ . This means that all partial derivatives should be zero, including the partial derivative with respect to λ .[3]

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \qquad \text{and} \qquad \frac{\partial \mathcal{L}}{\partial \lambda} = 0\,;$$

or equivalently

$$\frac{\partial f(x)}{\partial x} + \lambda \cdot \frac{\partial g(x)}{\partial x} = 0 \qquad \text{and} \qquad g(x) = 0\,.$$

The solution corresponding to the original constrained optimization is always a saddle point of the Lagrangian function,[4][5] which can be identified among the stationary points from the definiteness of the bordered Hessian matrix.[6]

https://en.wikipedia.org/wiki/Lagrange_multiplier

Fitting with correlated uncertainties

$$\chi^2_{\mathrm{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Theory

$$\chi^2_{\exp}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Theory

Measurements

$$\chi^2_{\text{exp}}(\boldsymbol{m},\boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta^2_i} + \sum_j b_j^2. \qquad (27)$$

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Theory

Measurements

$$\chi^2_{\text{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Statistical and uncorrelated uncertainty

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Correlated uncertainty source *j* in measurement *i*

Theory

Measurements

$$\chi^2_{\exp}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Statistical and uncorrelated uncertainty

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Correlated uncertainty source *j* in measurement *i*

Theory

Measurements

$$\chi^2_{\exp}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Statistical and uncorrelated uncertainty

Nuisance parameters
(Lagrange multipliers)

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Correlated uncertainty source *j* in measurement *i*

Theory

Measurements

$$\chi^2_{\mathrm{exp}}(\boldsymbol{m},\boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Statistical and uncorrelated uncertainty

Nuisance parameters
(Lagrange multipliers)

This χ^2 prescription is suitable for measurements in which the uncertainties are absolute, i.e. do not depend on the central value of the measurement.

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Correlated uncertainty source *j* in measurement *i*

Theory

Measurements

$$\chi^2_{\mathrm{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \Gamma^i_j b_j - \mu^i]^2}{\Delta_i^2} + \sum_j b_j^2. \qquad (27)$$

Statistical and uncorrelated uncertainty

Nuisance parameters
(Lagrange multipliers)

This χ^2 prescription is suitable for measurements in which the uncertainties are absolute, i.e. do not depend on the central value of the measurement.

But correlated and uncorrelated systematic errors are frequently, to a good approximation, proportional to the central values (multiplicative errors)

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

$$\chi^2_{\mathrm{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \gamma^i_j m^i b_j - \mu^i]^2}{\delta^2_{i,\mathrm{stat}} \mu^i (m^i - \sum_j \gamma^i_j m^i b_j) + (\delta_{i,\mathrm{uncor}}\, m^i)^2} + \sum_j b_j^2. \quad (31)$$

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Relative correlated uncertainty source *j* in measurement *i*

$$\chi^2_{\rm exp}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{[m^i - \sum_j \gamma^i_j m^i b_j - \mu^i]^2}{\delta^2_{i,\rm stat}\mu^i(m^i - \sum_j \gamma^i_j m^i b_j) + (\delta_{i,\rm uncor}\, m^i)^2} + \sum_j b_j^2. \quad (31)$$

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Relative correlated uncertainty source *j* in measurement *i*

$$
\chi^2_{\mathrm{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{\left[m^i - \sum_j \gamma^i_j m^i b_j - \mu^i\right]^2}{\delta^2_{i,\mathrm{stat}} \mu^i \left(m^i - \sum_j \gamma^i_j m^i b_j\right) + \left(\delta_{i,\mathrm{uncor}}\, m^i\right)^2} + \sum_j b_j^2. \quad (31)
$$

Relative uncorrelated uncertainty

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Fitting with correlated uncertainties

Relative correlated uncertainty source *j* in measurement *i*

Relative statistical uncertainty

$$\chi^2_{\text{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{\left[m^i - \sum_j \gamma^i_j m^i b_j - \mu^i\right]^2}{\delta^2_{i,\text{stat}} \mu^i \left(m^i - \sum_j \gamma^i_j m^i b_j\right) + \left(\delta_{i,\text{uncor}}\, m^i\right)^2} + \sum_j b_j^2. \quad (31)$$

Relative uncorrelated uncertainty

This is the formula we will use

Measurement of the Inclusive ep Scattering Cross Section at Low Q^2 and x at HERA
H1 Collaboration, Eur.Phys.J. C63 (2009) 625-678
https://inspirehep.net/record/817368?ln=en

Example: power law

The cross section for the exclusive production of J/ψ is expected to follow a power law for the dependence on the centre-of-mass energy of the γp system [1]

[1] Energy dependence of exclusive J/ψ photoproduction off protons in ultra-peripheral p-Pb collisions at $\sqrt{s_{\rm{NN}}}$ = 5.02 TeV
ALICE Collaboration, Eur.Phys.J. C79 (2019) no.5, 402
https://inspirehep.net/record/1693305?ln=en

Example: power law

The cross section for the exclusive production of J/ψ is expected to follow a power law for the dependence on the centre-of-mass energy of the γp system [1]

1

Use a power law with ALICE-like parameters

[1] Energy dependence of exclusive J/ψ photoproduction off protons in ultra-peripheral p-Pb collisions at $\sqrt{s_{\rm{NN}}}$ = 5.02 TeV
ALICE Collaboration, Eur.Phys.J. C79 (2019) no.5, 402
https://inspirehep.net/record/1693305?ln=en

Example: power law

The cross section for the exclusive production of J/ψ is expected to follow a power law for the dependence on the centre-of-mass energy of the γp system [1]

1 Use a power law with ALICE-like parameters

2 Generate cross sections using realistic uncorrelated uncertainties

[1] Energy dependence of exclusive J/ψ photoproduction off protons in ultra-peripheral p-Pb collisions at $\sqrt{s_{\rm{NN}}}$ = 5.02 TeV
ALICE Collaboration, Eur.Phys.J. C79 (2019) no.5, 402
https://inspirehep.net/record/1693305?ln=en

Example: power law

The cross section for the exclusive production of J/ψ is expected to follow a power law for the dependence on the centre-of-mass energy of the γp system [1]

1 Use a power law with ALICE-like parameters

2 Generate cross sections using realistic uncorrelated uncertainties

3 Add correlated uncertainties of 4 types

[1] Energy dependence of exclusive J/ψ photoproduction off protons in ultra-peripheral p-Pb collisions at $\sqrt{s_{\rm{NN}}}$ = 5.02 TeV
ALICE Collaboration, Eur.Phys.J. C79 (2019) no.5, 402
https://inspirehep.net/record/1693305?ln=en

Example: power law

The cross section for the exclusive production of J/ψ is expected to follow a power law for the dependence on the centre-of-mass energy of the γp system [1]

1 Use a power law with ALICE-like parameters

2 Generate cross sections using realistic uncorrelated uncertainties

3 Add correlated uncertainties of 4 types

- Overall normalisation (eg. luminosity or BR)
- Affecting only 'forward' data
- Affecting only 'semi-forward' data
- Affecting only 'central-barrel' data

[1] Energy dependence of exclusive J/ψ photoproduction off protons in ultra-peripheral p-Pb collisions at $\sqrt{s_{\rm{NN}}}$ = 5.02 TeV
ALICE Collaboration, Eur.Phys.J. C79 (2019) no.5, 402
https://inspirehep.net/record/1693305?ln=en

Snippets (1/2)

```
// ---------------------------------------------------------------------------------
// parameters for ideal power law
// ---------------------------------------------------------------------------------
double lambdaPL = 0.7;
double nPL = 70; // nb
double wPL = 90; // GeV

// ---------------------------------------------------------------------------------
// info to generate power law
// ---------------------------------------------------------------------------------
const int nPoints = 9; // number of measurements to be simulated
const double wPoints[nPoints] = {24, 30, 40, 50,  70,  130,  190,  390,  700}; // GeV
const double staUnc[nPoints] = {0.05, 0.03, 0.05, 0.07, 0.07, 0.03, 0.03, 0.07,0.04}; // rel stat unc
const double uncUnc[nPoints] = {0.08, 0.08, 0.08, 0.07,
                                0.05, 0.07, 0.07, 0.07, 0.10}; // uncorrelated unceratainty in percent
const int nCor = 4; // number of correlated sources of uncertainty
const double normUnc = 0.03; // global normalization correlated error
const double fwdUnc = 0.05; // correlated uncertainty for fwd points
const double sfUnc = 0.04; // correlated uncertainty for semi-fwd points
const double cbUnc = 0.04; // correlated uncertainty for central barrel points
const double corUnc[nCor][nPoints] = // relative correlated uncertainty
  {
    normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc,
    fwdUnc, fwdUnc, fwdUnc, 0.0, 0.0, 0.0, 0.0, 0.0, fwdUnc, // only fwd
     0.0, 0.0, 0.0, sfUnc, sfUnc,  0.0, 0.0, sfUnc, 0.0,  // only semifwd
     0.0, 0.0, 0.0,  0.0,  0.0, cbUnc, cbUnc,  0.0, 0.0  // only semifwd
  };
```

Snippets (1/2)

```
// -------------------------------------------------------------------------------
// parameters for ideal power law
// -------------------------------------------------------------------------------
double lambdaPL = 0.7;
double nPL = 70; // nb
double wPL = 90; // GeV

// -------------------------------------------------------------------------------
// info to generate power law
// -------------------------------------------------------------------------------
const int nPoints = 9; // number of measurements to be simulated
const double wPoints[nPoints] = {24, 30, 40, 50,  70,  130,  190,  390,  700}; // GeV
const double staUnc[nPoints] = {0.05, 0.03, 0.05, 0.07, 0.07, 0.03, 0.03, 0.07,0.04}; // rel stat unc
const double uncUnc[nPoints] = {0.08, 0.08, 0.08, 0.07,
                                0.05, 0.07, 0.07, 0.07, 0.10}; // uncorrelated unceratainty in percent
const int nCor = 4; // number of correlated sources of uncertainty
const double normUnc = 0.03; // global normalization correlated error
const double fwdUnc = 0.05; // correlated uncertainty for fwd points
const double sfUnc = 0.04; // correlated uncertainty for semi-fwd points
const double cbUnc = 0.04; // correlated uncertainty for central barrel points
const double corUnc[nCor][nPoints] = // relative correlated uncertainty
  {
    normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc, normUnc,
    fwdUnc, fwdUnc, fwdUnc, 0.0, 0.0, 0.0, 0.0, 0.0, fwdUnc, // only fwd
     0.0, 0.0, 0.0, sfUnc, sfUnc,  0.0, 0.0, sfUnc, 0.0,  // only semifwd
     0.0, 0.0, 0.0,  0.0,  0.0, cbUnc, cbUnc,  0.0, 0.0  // only semifwd
  };
```



```
// ------------------------------------------------------------------------------------
//  generated power law
// ------------------------------------------------------------------------------------
double csGen[nPoints]; // nb
double csPL[nPoints]; // nb
double suUnc[nPoints]; // stat+unc uncertaity in nb
void generateCS()
{
  // initialise random number generator
  gRandom->SetSeed(0);
  cout << endl << " Random seed: " << gRandom->GetSeed() << endl;
  // generate cross section
  for(int i=0; i<nPoints;i++) {
    csPL[i] = nPL*TMath::Power(wPoints[i]/wPL,lambdaPL);
    csGen[i] = csPL[i]*gRandom->Gaus(1,staUnc[i]);
    csGen[i] *= gRandom->Gaus(1,uncUnc[i]);
    suUnc[i] = csGen[i]*TMath::Sqrt(staUnc[i]*staUnc[i]+uncUnc[i]*uncUnc[i]);
  }

  // shift full cross section
  double normCS = gRandom->Gaus(1,normUnc);
  for(int i=0; i<nPoints;i++) csGen[i] *= normCS;
  cout << " normalisation shift: " << normCS << endl;

  // shift fwd cross sections
  double normFW =  gRandom->Gaus(1,fwdUnc);
  csGen[0] *= normFW; csGen[1] *= normFW; csGen[2] *= normFW; csGen[8] *= normFW;
  cout << " fwd shift: " << normFW << endl;

  // shift sf cross sections
  double normSF =  gRandom->Gaus(1,sfUnc);
  csGen[3] *= normSF; csGen[4] *= normSF; csGen[7] *= normSF;
  cout << "  semi-fwd shift: " << normSF << endl;

  // shift cb cross sections
  double normCB =  gRandom->Gaus(1,cbUnc);
  csGen[5] *= normCB; csGen[6] *= normCB;
  cout << " central barrel shift: " << normCB << endl;
  cout << endl;
}
```

Snippets (2/2)

```
// -----------------------------------------------------------------------------
// chi2 function to fit data according to prescription in
// Eur.Phys.J. C63 (2009) 625-678, section 9, eq (31)
// -----------------------------------------------------------------------------

void fcnChi2Model(int &npar, double *gin, double &f, double *par, int iflag)

{
  // use unused parguments to avoid compiler message
  double tmp = npar; tmp = gin[0]; tmp = iflag;

  // define model according to:
  // ch2 = sum_i [m_i-mu_i-Sij]^2/D + Sbj
  // Sij = sum_j g_ij*m_i*b_j
  // Sbj = sum_j (b_j)^2
  // D = d_i_stat^2*mu_i*(m_i-Sij)+(d_i,unc*m_i)^2
  // mu_i is the measurement
  // g_ij relative normalization uncertainty at point i from source j
  // d_i = relative uncertainty (either stat or uncorr)

  double bj[nCor] = {par[2],par[3],par[4],par[5]};
  double Sbj = 0;
  for(int j=0;j<nCor;j++) Sbj += (bj[j]*bj[j]);
  double chi2 = 0;
  for (int i=0; i<nPoints;i++) {
    double mu_i = csGen[i];
    double m_i = par[0]*TMath::Power(wPoints[i]/wPL,par[1]);
    double Sij = 0;
    for(int j=0;j<nCor;j++) Sij += m_i*bj[j]*corUnc[j][i];
    double d_stat_i = staUnc[i];
    double d_unc_i = uncUnc[i];
    double D = (d_stat_i*d_stat_i*mu_i*(m_i-Sij))+((d_unc_i*m_i)*(d_unc_i*m_i));
    chi2 += ((m_i-mu_i-Sij)*(m_i-mu_i-Sij)/D);
  }
  chi2 += Sbj;
  f = chi2;
  gChi2 = f;
}
```

Snippets (2/2)

```
// ---------------------------------------------------------------------------
// chi2 function to fit data according to prescription in
// Eur.Phys.J. C63 (2009) 625-678, section 9, eq (31)
// ---------------------------------------------------------------------------

void fcnChi2Model(int &npar, double *gin, double &f, double *par, int iflag)

{
  // use unused pargmuents to avoid compiler message
  double tmp = npar; tmp = gin[0]; tmp = iflag;

  // define model according to:
  // ch2 = sum_i [m_i-mu_i-Sij]^2/D + Sbj
  // Sij = sum_j g_ij*m_i*b_j
  // Sbj = sum_j (b_j)^2
  // D = d_i_stat^2*mu_i*(m_i-Sij)+(d_i,unc*m_i)^2
  // mu_i is the measurement
  // g_ij relative normalization uncertainty at point i from source j
  // d_i = relative uncertainty (either stat or uncorr)

  double bj[nCor] = {par[2],par[3],par[4],par[5]};
  double Sbj = 0;
  for(int j=0;j<nCor;j++) Sbj += (bj[j]*bj[j]);
  double chi2 = 0;
  for (int i=0; i<nPoints;i++) {
    double mu_i = csGen[i];
    double m_i = par[0]*TMath::Power(wPoints[i]/wPL,par[1]);
    double Sij = 0;
    for(int j=0;j<nCor;j++) Sij += m_i*bj[j]*corUnc[j][i];
    double d_stat_i = staUnc[i];
    double d_unc_i = uncUnc[i];
    double D = (d_stat_i*d_stat_i*mu_i*(m_i-Sij))+((d_unc_i*m_i)*(d_unc_i*m_i));
    chi2 += ((m_i-mu_i-Sij)*(m_i-mu_i-Sij)/D);
  }
  chi2 += Sbj;
  f = chi2;
  gChi2 = f;
}
```



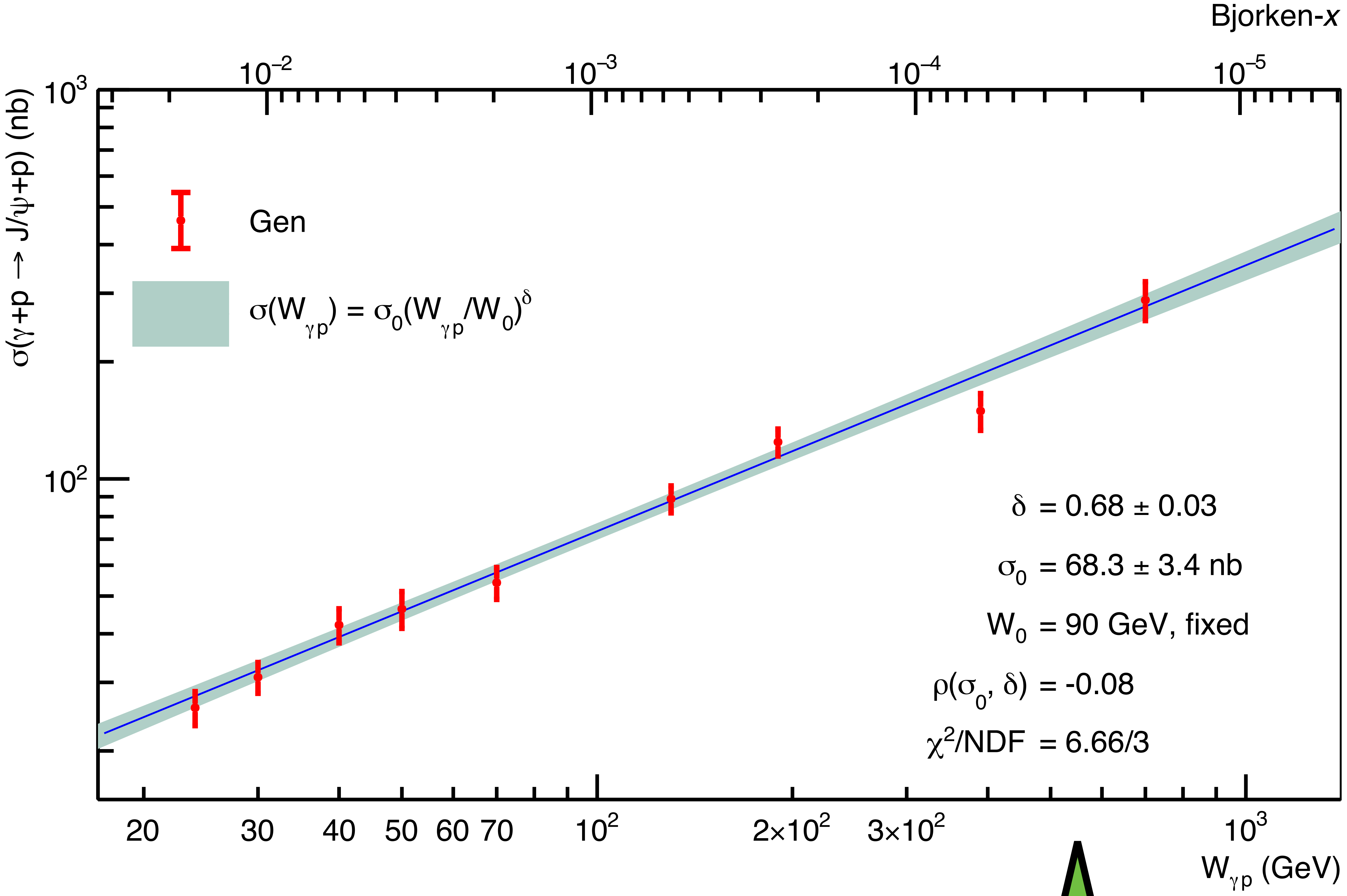
```
void powerLawFit()
{
  //////////////////////////////////////////////////////////////////////////
  // generate data set
  //////////////////////////////////////////////////////////////////////////
  generateCS();

  //////////////////////////////////////////////////////////////////////////
  // now do TMinut
  //////////////////////////////////////////////////////////////////////////
  cout << endl<< endl<< endl;
  cout << " - o - o - o - o - o - o - o - o -  Minuit   - o - o - o - o - o - o -
  cout << endl<< endl<< endl;
  // initialize minuit with a maximum of parameters
  const int nPar = 2+nCor;
  TMinuit *myMinuit = new TMinuit(nPar);

  // set the function with the minimization process
  myMinuit->SetFCN(fcnChi2Model);

  // define parameters
  double minBj = -10;
  double maxBj = -minBj;
  myMinuit->DefineParameter(0,"N",70,1,0.0,1000.);
  myMinuit->DefineParameter(1,"#delta",0.7,0.1,0.1,10.);
  myMinuit->DefineParameter(2,"global norm",0.001,0.0001,minBj,maxBj);
  myMinuit->DefineParameter(3,"fwd norm",0.001,0.0001,minBj,maxBj);
  myMinuit->DefineParameter(4,"sf norm",0.001,0.0001,minBj,maxBj);
  myMinuit->DefineParameter(5,"cb norm",0.001,0.0001,minBj,maxBj);

  // migrad
  myMinuit->SetMaxIterations(500);
  myMinuit->Migrad();

  //////////////////////////////////////////////////////////////////////////
  // get results
  //////////////////////////////////////////////////////////////////////////
```

Test 1

Random seed: 2584683516
normalisation shift: 0.954077
fwd shift: 0.985996
semi-fwd shift: 0.957315
central barrel shift: 1.00353

Test 1

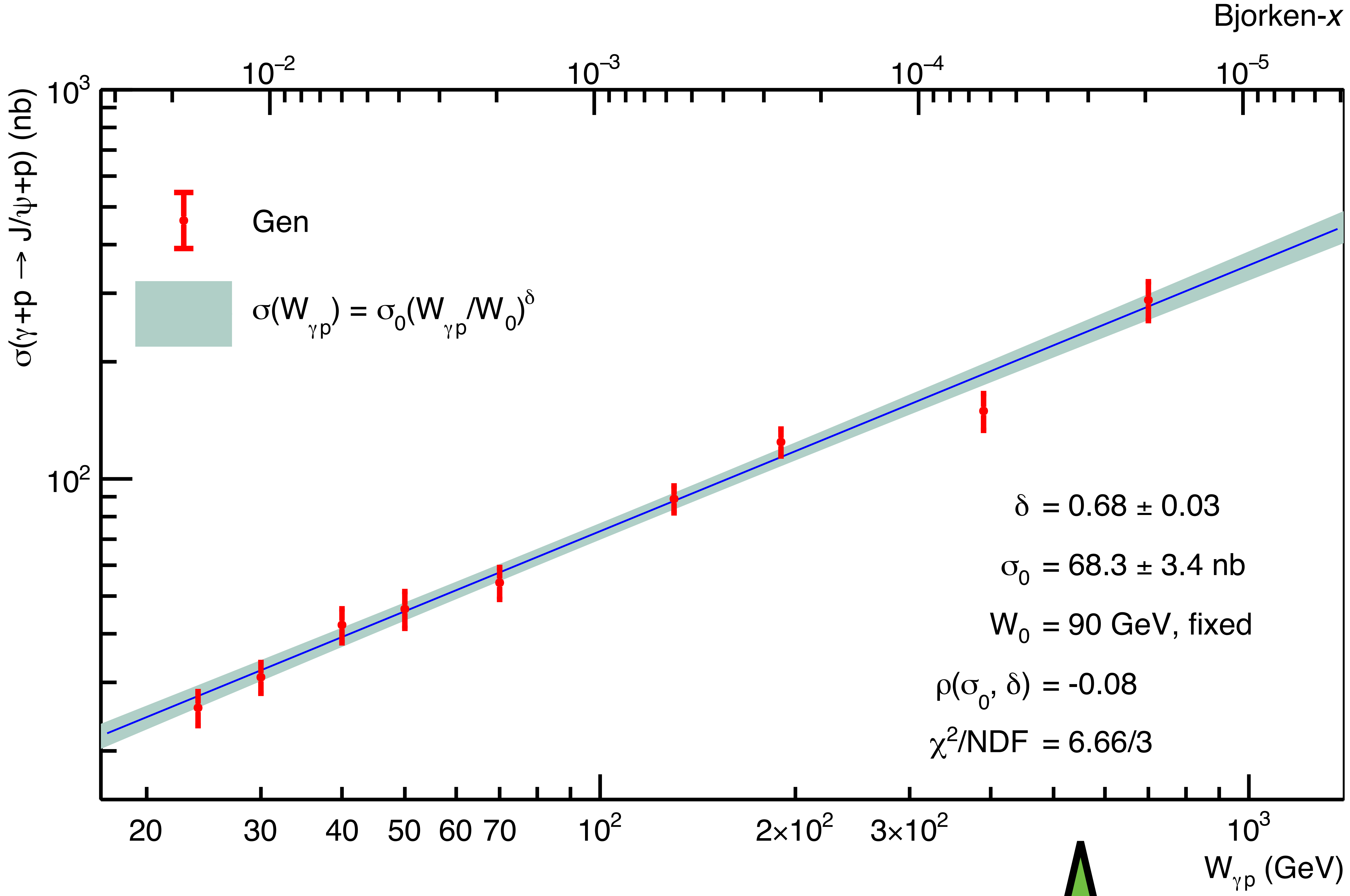
Random seed: 2584683516
normalisation shift: 0.954077
fwd shift: 0.985996
semi-fwd shift: 0.957315
central barrel shift: 1.00353



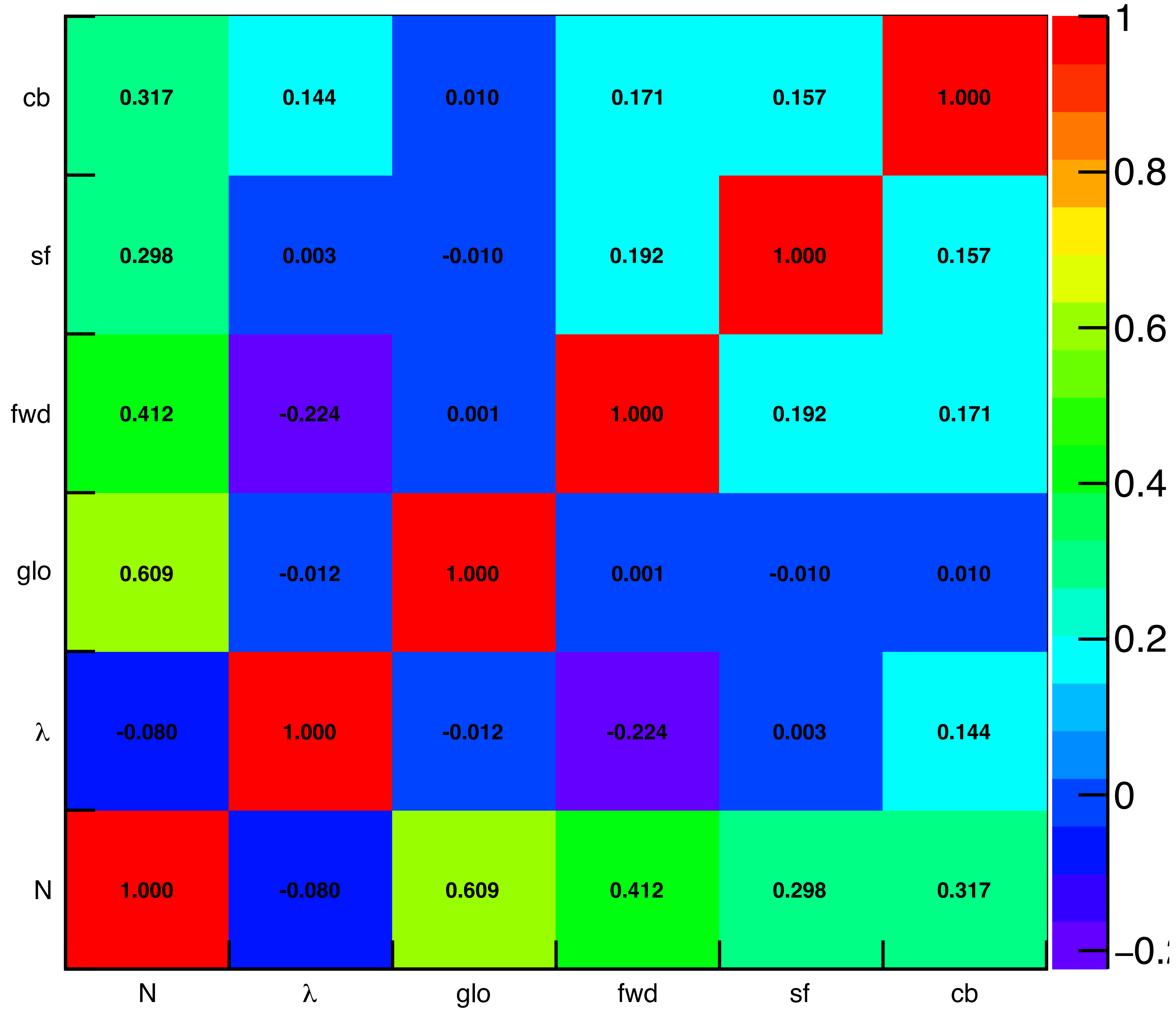
Input parameter recovered with small correlation

Test 1

Random seed: 2584683516
normalisation shift: 0.954077
fwd shift: 0.985996
semi-fwd shift: 0.957315
central barrel shift: 1.00353



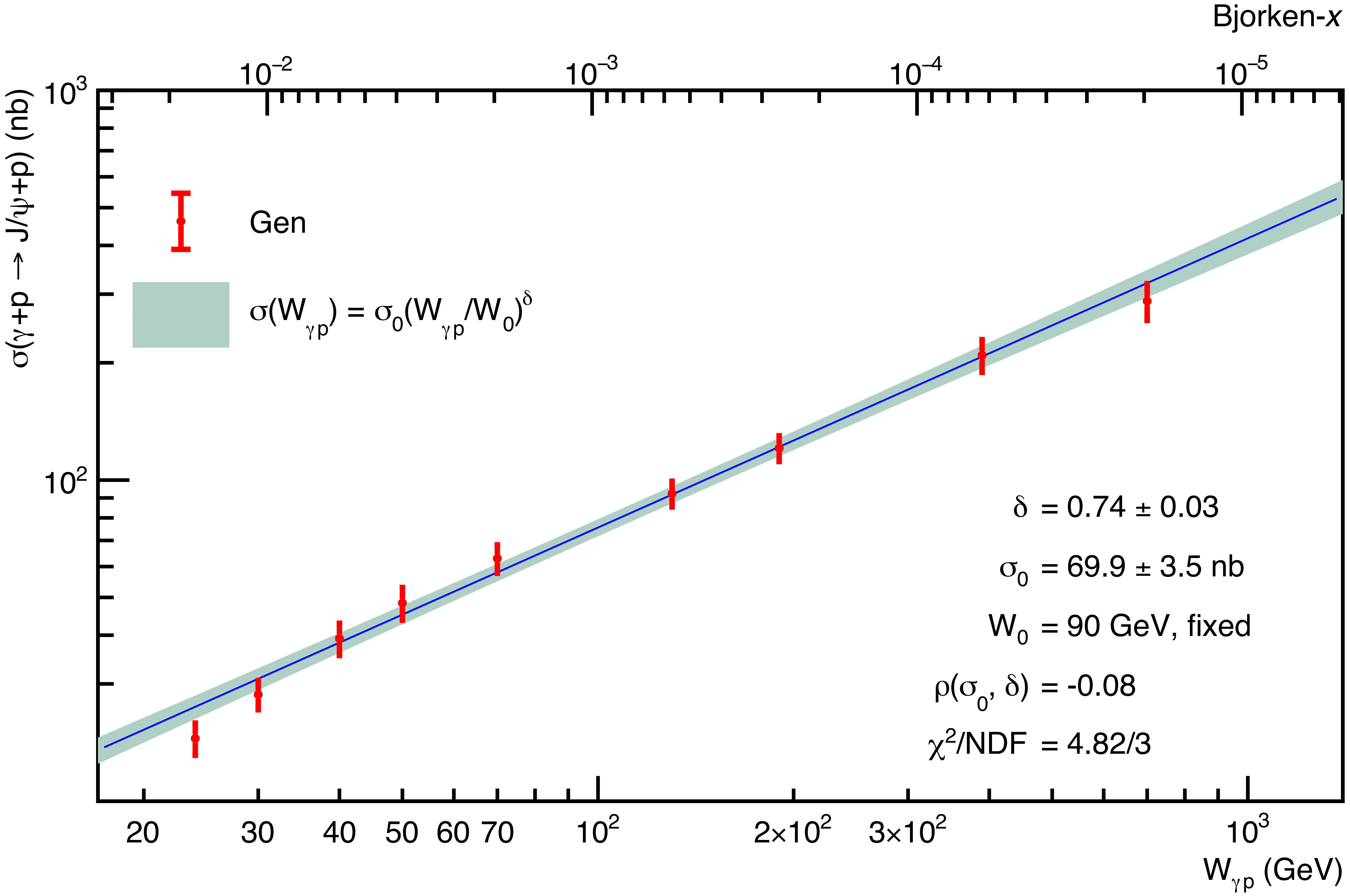
Correlation matrix

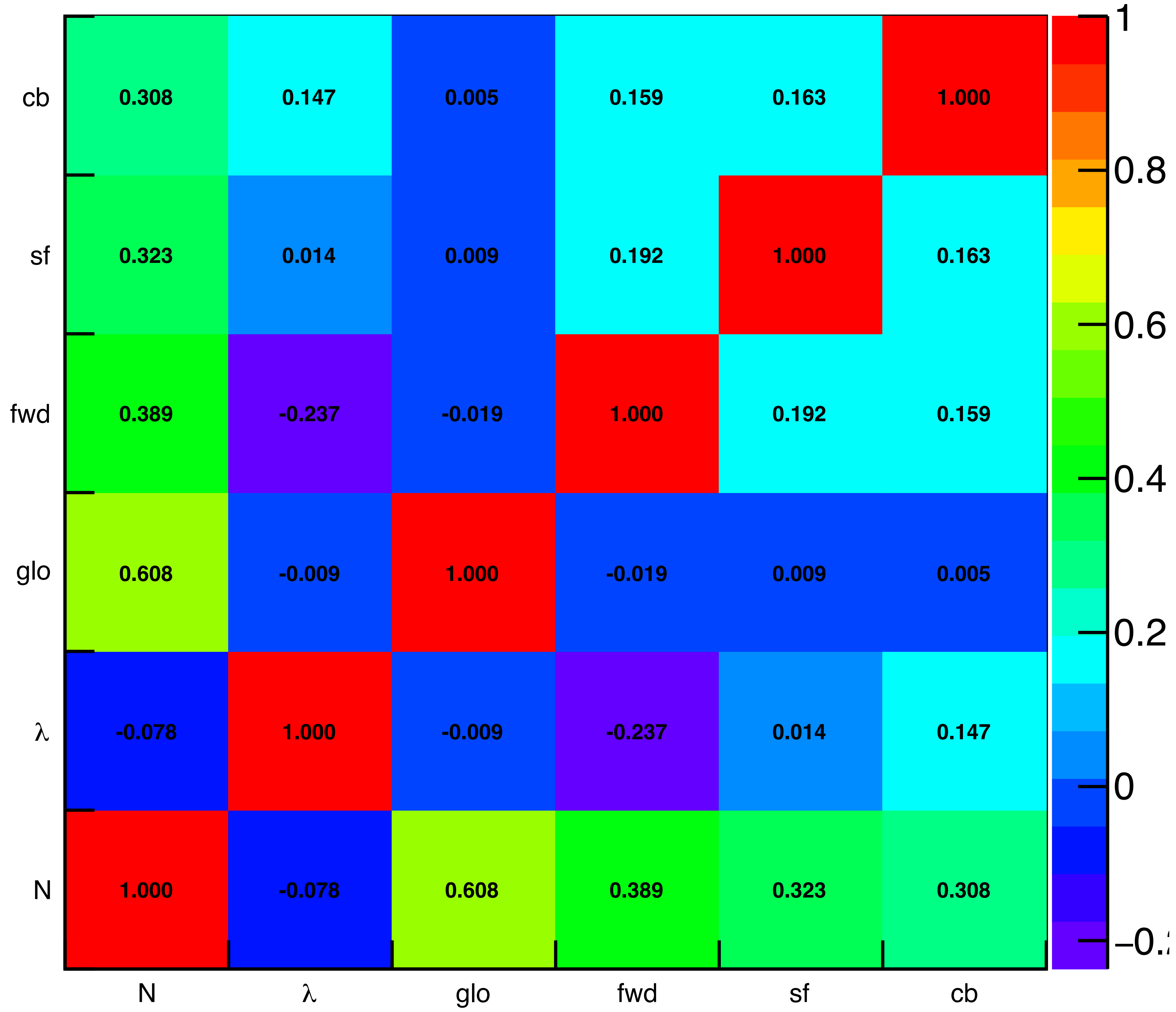


Input parameter recovered with small correlation

Test 2

Random seed: 3900432653
normalisation shift: 0.986186
fwd shift: 1.00663
semi-fwd shift: 1.04337
central barrel shift: 1.01852

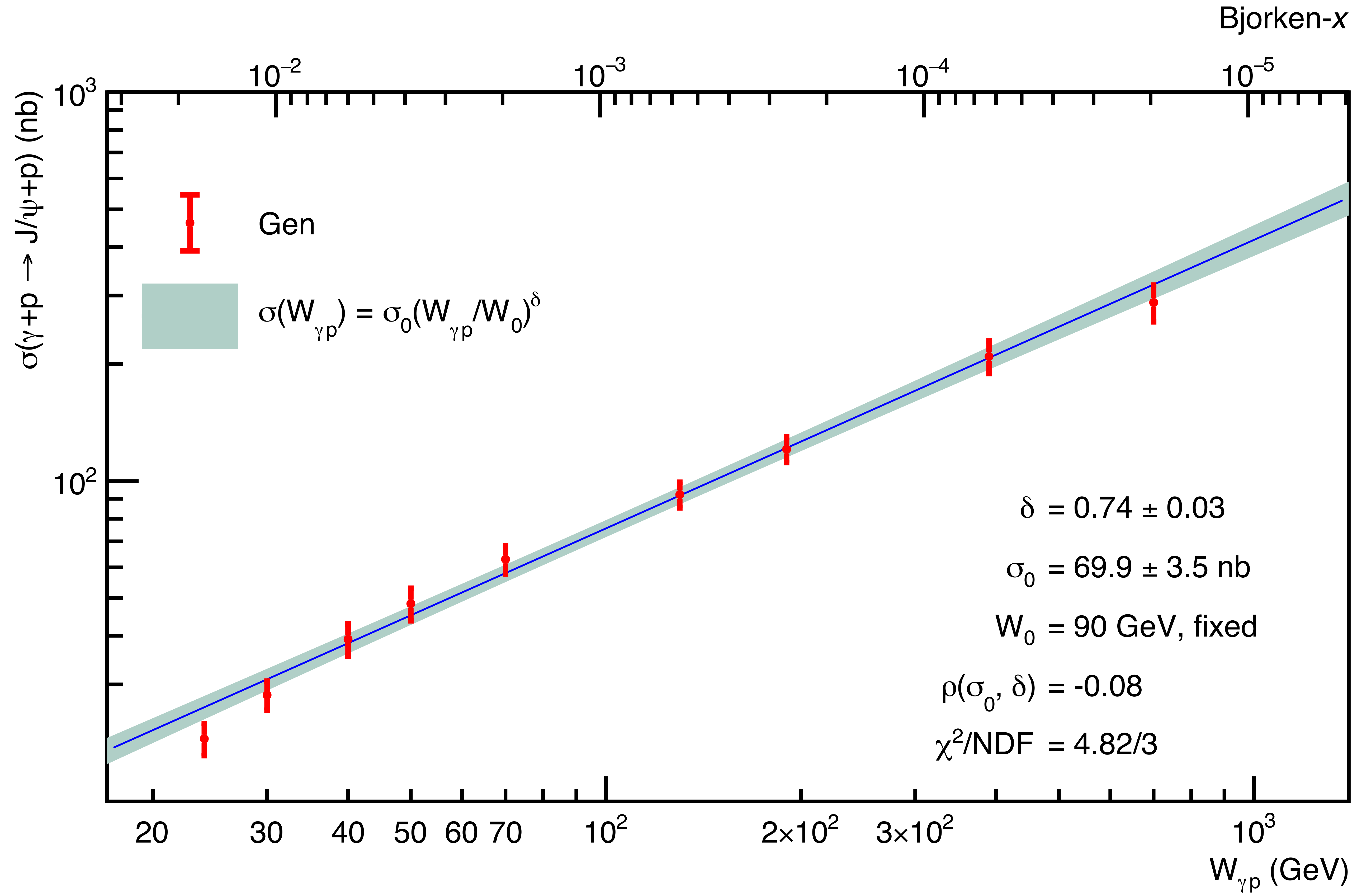


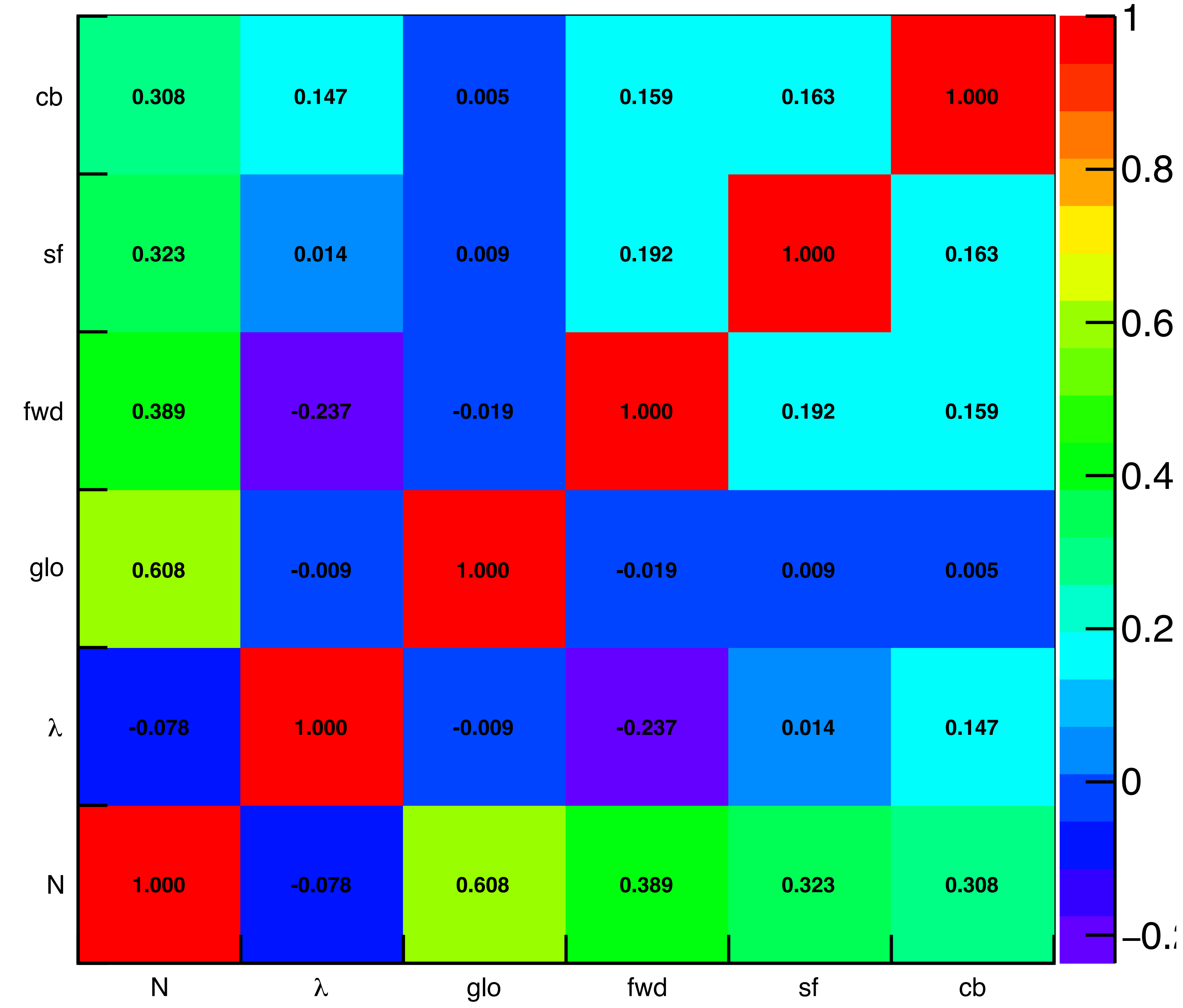


Test 2

Random seed: 3900432653
normalisation shift: 0.986186
fwd shift: 1.00663
semi-fwd shift: 1.04337
central barrel shift: 1.01852

Correlation matrix





Summary

Formalism to estimate parameters when there are correlated uncertainties

Code tested, and attached to the indico page