An unfolding method for high energy physics experiments

Contribution to the Conference

Advanced Statistical Techniques in Particle Physics

Durham, 18 – 22 March 2002

Volker Blobel

Institut für Experimentalphysik, Universität Hamburg

Abstract

Finite detector resolution and limited acceptance require to apply unfolding methods in high energy physics experiments. Information on the detector resolution is usually given by a set of Monte Carlo events. Based on the experience with a widely used unfolding program (RUN) a modified method has been developed.

The first step of the method is a maximum likelihood fit of the Monte Carlo distributions to the measured distribution in one, two or three dimensions; the finite statistic of the Monte Carlo events is taken into account by the use Barlows method with a new method of solution. A clustering method is used before to combine bins in sparsely populated areas. In the second step a regularization is applied to the solution, which introduces only a small bias. The regularization parameter is determined from the data after a diagonalization and rotation procedure.

1 The unfolding problem

A standard task in high energy physics experiments is the measurement of a distribution $f(x)$ of some kinematical quantity x. With an ideal detector one could measure the quantity x in every event and could obtain $f(x)$ by a simple histogram of the quantity x. With *real* detectors the determination of $f(x)$ is complicated by three effects:

- **Limited acceptance**

  The probability to observe a given event, the *detector acceptance*, is less than 1, resulting in a certain loss of recorded events. The acceptance can and will in general depend on the kinematical variable x.

- **Transformation**

  Instead of the quantity x a different, but related quantity y is measured in the detector. The relation between x and y is given by some transformation. The transformation can for example be caused by the non-linear response of a detector component.

- **Finite resolution**

  The measured quantity y is smeared out due to the finite resolution (or limited measurement accuracy) of the detector. Thus there is only a *statistical* relation between the true kinematical variable x and the measured quantity y.

The really difficult effect in the data correction for experimental effects, or data transformation from y to x is the **finite resolution**, causing a *smearing* of the measured quantities. Mathematically the relation between the distribution $f(x)$ of the true variable x, to be determined in an experiment, and the measured distribution $g(y)$ of the measured quantity y is given by the integral equation,

$$g(y) = \int A(y,x) f(x) \, \mathrm{d}x + b(y) \; , \tag{1}$$

called a Fredholm integral equation of the first kind. In this equation the distribution $b(y)$ has been added to the right-hand side; it represents a *background* contribution to the measured distribution $g(y)$. This has in practice to be taken into account; the background contribution can be either independently measured or calculated and thus can be assumed to be *known.* The resolution function $A(y,x)$ represents the effect of the detector. For a given value $x = x_0$ the resolution function $A(y,x_0)$ describes the response of the detector in the variable y for that fixed value x_0. The problem to determine the distribution $f(x)$ from measured distributions $g(y)$ (and a known distribution $b(y)$) is called *unfolding*; it is called an inverse problem. Unfolding of course requires the knowledge of the resolution function $A(y,x)$, i.e. all the effects of limited acceptance, transformation and finite resolution.

In addition to the imperfections of the detector, there may be further effects between x and y, which are *outside of the experimental control*, even with an ideal detector. One example are radiative effects, which in experiments are often corrected afterwards *(radiative corrections)*, but are in their effect similar to detector effects. Another effect is the *Fermi motion* and has to do with the dynamics within the substructure of the target or beam particles. If the true kinematical quantity is defined at the *parton level*, further effects from the fragmentation process of partons to the (observable) hadrons influence the measured quantity y. All these effects are of statistical nature.

A typical example for these effects is shown in Figure 1 taken from a Monte Carlo simulation of all three effects. By unfolding an estimate of the original distribution has to be determined from the distorted measured distribution. Details on the Monte Carlo simulation are given later in section 3, where the unfolding in this example is discussed in detail.

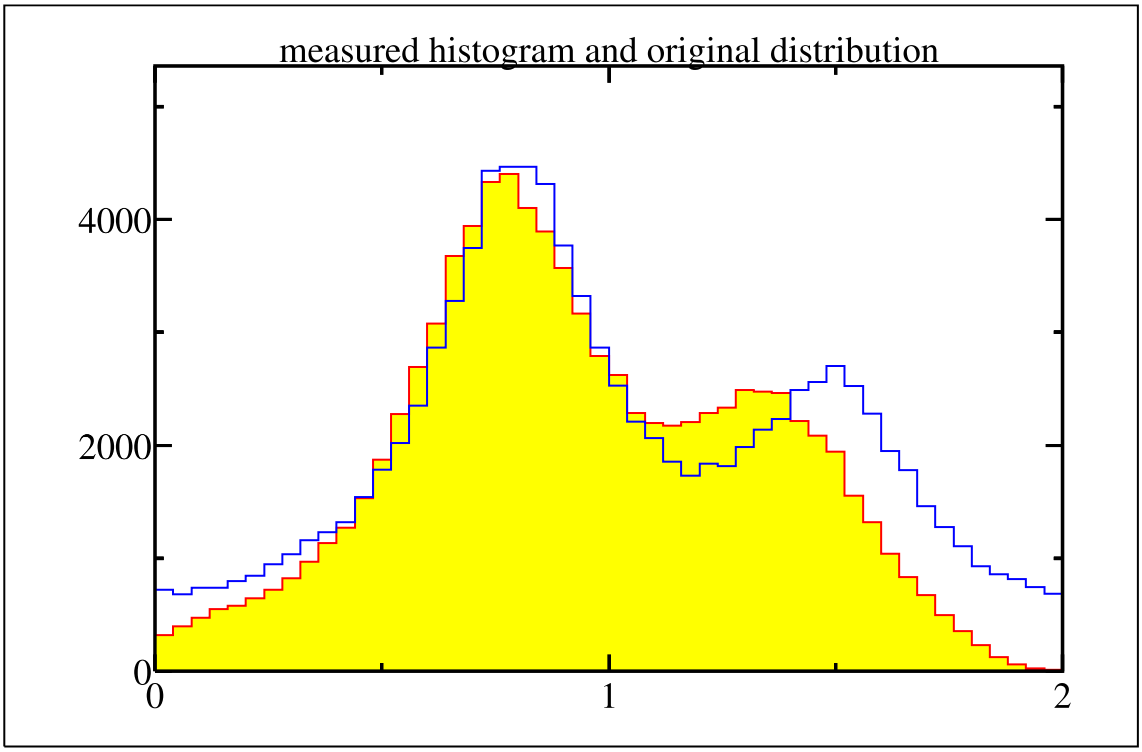

Figure 1: The Monte Carlo simulation of the effects of limited acceptance, transformation and finite resolution. Shown is the original (true) distribution and the "measured" distribution.

For the numerical solution of equation (1) the distributions have to represented by a finite set of parameters. One posssibility is to represent the distributions by histograms, and the resolution function by a matrix. Equation (1) can then be represented by the matrix equation

$$\boxed{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}\,, \tag{2}$$

which has to be solved for the vector $\boldsymbol{x}$, given the vectors $\boldsymbol{y}$ and $\boldsymbol{b}$ (known background histogram). The vector $\boldsymbol{y}$ (and $\boldsymbol{b}$) with n elements represent histograms of the measured quantity y, and the distribution $f(x)$ is represented by a histogram of vector $\boldsymbol{x}$ with m elements. The variables y and x may be multidimensional, and the multidimensional histograms can be mapped to n-bin and m-bin histograms, respectively. The transition from $\boldsymbol{x}$ to $\boldsymbol{y}$ is described by the n-by-m matrix $\boldsymbol{A}$. The elements a_{ij} is related to the probability observe an entry in histogram bin i of the histogram $\boldsymbol{y}$, if the true value x is from histogram bin j of the histogram $\boldsymbol{x}$ of true values.

Standard solution methods for the equation (2) may give unacceptable solution vectors $\boldsymbol{x}$ with wildly fluctuating bins contents, although it can be shown that the solution has zero bias and minimum covariance (Cowan [1]). In the case $n = m$ the equation can be solved exactly and in the case $n > m$ methods like the least squares or the maximum likelihood method may be used. The reason for the oscillations are large correlations between bins, usually with a correlation coefficient close to -1 for neighbour bins. These difficulties in the solution of the integral equation (1) or the discretization (2) are well known, even if there are no measurement errors [2, 3]. The problem is called ill-posed, because often small changes in the data may result in large changes of the solution.

In high energy physics experiments the problems is even more difficult than in other fields. Often the statistics of the measurement is not high and every $\boldsymbol{y}$-bin content (which is distributed due to the Poisson distribution around the expected value) has large statistical fluctuations. Furthermore the resolution function $\boldsymbol{A}(x,y)$ (or the matrix $\boldsymbol{A}$) is not known from theory, but represented by a Monte Carlo simulation of the process, based on some assumed distibution $f_{\mathrm{MC}}(x)$,

$$g_{\mathrm{MC}}(y) = \int A(y,x) f_{\mathrm{MC}}(x)\,\mathrm{d}x\ , \qquad \text{(Monte Carlo simulation)} \tag{3}$$

and is also statistically limited. Standard methods for the solution of integral equations or linear equations can not be used in this case.

A simple method used in many experiment is the so-called bin-by-bin correction which may be meaningful if the measurements y are very close to the true values x, i.e. most of the entries in a $\boldsymbol{y}$-histogram bin should have their origin in the corresponding $\boldsymbol{x}$ bin [4]. The result of this simple method is often biased towards the Monte Carlo input $f_{\mathrm{MC}}(x)$, which may or may not be close to the true distribution. In addition to this bias there is the problem of assigning correct errors to the result; this may be impossible because correlations between bins are ignored by construction of the method. Nevertheless the method is often used in practice and can give reasonable results if the main effect of the measurement is *limited acceptance.*

Real *unfolding* methods, taking all the correlations into account, are essential if there are larger effects of *transformation* and *finite resolution.* In the maximum likelihood method a function $F(\boldsymbol{x})$ is constructed as the negative log of the Likelihood function, which describes the statistical relations between data and result:

$$F(\boldsymbol{x}) = -\log L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}) \tag{4}$$

and the minimum of $F(\boldsymbol{x})$ is determined. Wildly fluctuating results $\boldsymbol{x}$ are due to large (negative) correlations and are not acceptable. The standard approach to get a more reasonable solution is to impose a measure of the smoothness on the result $\boldsymbol{x}$; this method is called **regularization**. This technique was proposed independently by Phillips [5] and by Thikhonov [6, 7]. For a function $f(x)$ the expression

$$C(f) = \int \left(\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} \right)^2 \mathrm{d}x \tag{5}$$

is often used in the regularization [8], which in the linearized version of the problem can be expressed by a quadratic form

$$C(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x} \tag{6}$$

with a positive-semidefinite matrix $\boldsymbol{C}$ (derivatives are replaced by finite differences). Equation (4) is then modified to the form

$$F(\boldsymbol{x}) = -\log L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}) + \tau \cdot C(\boldsymbol{x}) \tag{7}$$

with a factor τ called regularization parameter. Other expressions for the regularization term are possible, for example according to the principle of maximum entropy [1].

The result of the minimization of the modified function $F(\boldsymbol{x})$ of equation (7) will show less fluctuations than the original version obtained from equation (4) and may be more useful to describe the result of the measurement and to compare the measurement with theoretical predictions. However it is clear that unavoidably the regularization introduces a bias. The magnitude of the bias depends on the regularization parameter τ. It is not a-priori clear in which way the regularization influences the result and how an optimal value of the regularization parameter τ is determined. A very large value would result in a *linear* function $f(x)$ or distribution $\boldsymbol{x}$, respectively. It is however clear that the method requires an a-priori knowledge about a smooth behaviour of $f(x)$. The function $f_{\mathrm{MC}}(x)$ used in the Monte Carlo simulation of equation (3) is often very close to the final result $f(x)$, i.e. the ratio is rather smooth. This suggests to express $f(x)$ in the form $f(x) = f_{\mathrm{MC}}(x) \times f_{\mathrm{mult}}(x)$ and to rewrite equation (1) in the form

$$g(y) = \int \left[A(y,x) f_{\mathrm{MC}}(x)\right] f_{\mathrm{mult}}(x) \, \mathrm{d}x + b(y) \; . \tag{8}$$

For the discretized form the function $f_{\mathrm{MC}}(x)$ can be absorbed in a redefinition of matrix $\boldsymbol{A}$ and the vector $\boldsymbol{x}$ is interpreted as discretization of the hopefully *smooth* function $f_{\mathrm{mult}}(x)$. With this redefinition the equation (2) can remain unchanged.

The program RUN [9, 10] for regularized unfolding is available since almost two decades and has been used in many experiments; early applications are [11, 15] and [12, 13]. It is based on the reinterpretation of matix $\boldsymbol{A}$ and $\boldsymbol{x}$, as described above, and includes a method for the determination of the regularization parameter τ based on the available degrees of freedom. In the method described later in this paper a few details are treated differently.

2 Unfolding as an ill-posed problem

The problems inherent to unfolding are discussed in a simple special case, assuming a resolution matrix $\boldsymbol{A}$ with some smearing of data into neighbour bins. Assuming a true vector $\boldsymbol{x}$ the product $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ describes the distribution expected due to the migration effect. Assuming the same dimensions for the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ the matrix $\boldsymbol{A}$ is a square matrix and in the example later in this section the following simple symmetric form is assumed for the matrix $\boldsymbol{A}$, which depends on a single parameter ε (ε = migration parameter); for a 5-by-5 matrix the form is

$$\boldsymbol{A} = \begin{pmatrix} 1-\varepsilon & \varepsilon & 0 & 0 & 0 \\ \varepsilon & 1-2\varepsilon & \varepsilon & 0 & 0 \\ 0 & \varepsilon & 1-2\varepsilon & \varepsilon & 0 \\ 0 & 0 & \varepsilon & 1-2\varepsilon & \varepsilon \\ 0 & 0 & 0 & \varepsilon & 1-\varepsilon \end{pmatrix} \; . \tag{9}$$

In this case a direct solution for $\boldsymbol{x}$, given a measurement $\widehat{\boldsymbol{y}}$ is possible with inversion of the

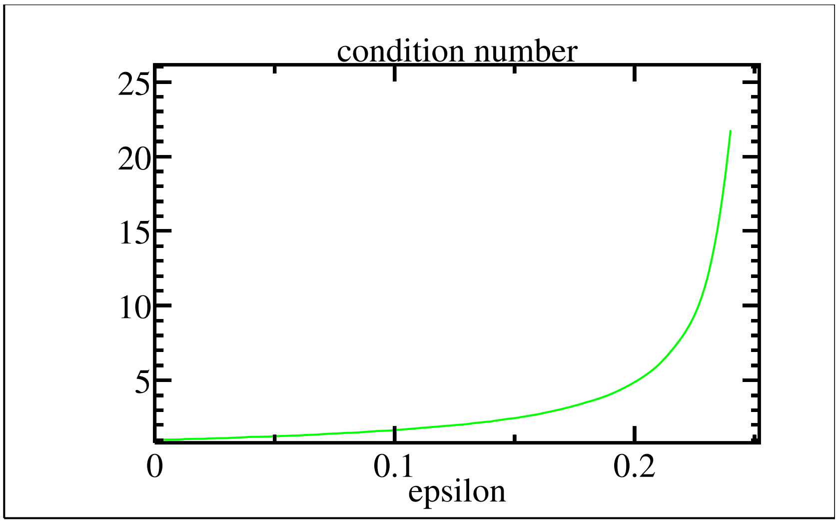

Figure 2: The condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ for the matrix of type of equation (9) for dimension $n = 20$ as a function of the migration parameter ε.

matrix $\boldsymbol{A}$:

$$\begin{array}{ll} \text{estimate} & \widehat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{y} \\ \text{error propagation} & \boldsymbol{V}(\widehat{\boldsymbol{x}}) = \boldsymbol{A}^{-1}\boldsymbol{V}_y \left(\boldsymbol{A}^{-1}\right)^T . \end{array}$$

The result has good statistical properties, for example it has no bias:

$$E[\boldsymbol{x}] = \boldsymbol{A}^{-1}E[\boldsymbol{y}] = \boldsymbol{A}^{-1}\boldsymbol{A}E[\boldsymbol{x}] = \boldsymbol{x} .$$

In practice the result is however satisfactory only for a matrix $\boldsymbol{A}$ with dominating diagonal; the result looks terrible if the matrix $\boldsymbol{A}$ describes a large migration to neighbour bins. Consequently the problem is called an ill-posed problem. In the following the solution of the equation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{10}$$

using an orthogonal decomposition is discussed; this will allow some insight into the unfolding problem. The left-hand side $\boldsymbol{y}$ represents the measured distribution, which will differ from the expectation $\boldsymbol{A}\boldsymbol{x}$ with the true vector $\boldsymbol{x}$ by (at least) statistical fluctuations. In the case considered here ($\boldsymbol{A}$ is a square matrix) an exact solution of the equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ for vector $\boldsymbol{x}$ would be possible.

The symmetric matrix $\boldsymbol{A}$ is decomposed by

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \tag{11}$$

with a diagonal matrix $\boldsymbol{D}$ and a transformation matrix $\boldsymbol{U}$:

$$\boldsymbol{D} = \begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ & & & & \lambda_n \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{1} .$$

The diagonal elements of matrix $\boldsymbol{D}$ are the eigenvalues λ_j of matrix $\boldsymbol{A}$, assumed here to be in the order of decreasing value. The transformation matrix $\boldsymbol{U}$ contains the corresponding eigenvectors with the eigenvector $\boldsymbol{u}_j$ in the j-th column. The condition number κ of a matrix is defined by the ratio of eigenvectors $\kappa = \lambda_{\max}/\lambda_{\min}$; the value of κ is important for the quality of unfolding (see below). Figure 2 shows the condition number for the matrix of the type of (9) for values of the migration parameter ε up to 0.24. For values above $\varepsilon = 0.20$ the condition number is rapidly increasing.
A transformation of equation (10) to a new basis is done by multiplication with matrix $\boldsymbol{U}^T$ (which is a rotation in the n-dimensional space):

$$\begin{array}{lrcl} \boldsymbol{U}^T \cdot \ | & \boldsymbol{y} & = & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{x} . \\ & \underbrace{\boldsymbol{U}^T\boldsymbol{y}}_{\boldsymbol{c}} & = & \boldsymbol{D}\underbrace{\boldsymbol{U}^T\boldsymbol{x}}_{\boldsymbol{b}} \end{array}$$

Thus the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are transformed to vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ using the matrix U of eigenvectors $\boldsymbol{u}_j$:

$$\begin{array}{rclccc} \boldsymbol{x} \to \boldsymbol{b} & = & \boldsymbol{U}^T\boldsymbol{x} & \text{or} & b_j = \boldsymbol{u}_j^T\boldsymbol{x} & j = 1\ldots n \\ \boldsymbol{y} \to \boldsymbol{c} & = & \boldsymbol{U}^T\boldsymbol{y} & \text{or} & c_j = \boldsymbol{u}_j^T\boldsymbol{y} & j = 1\ldots n . \end{array}$$

Of course the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ can be transformed back to the $\boldsymbol{x}/\boldsymbol{y}$-space by

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{b} \qquad\qquad \boldsymbol{x} = \boldsymbol{U}\boldsymbol{b} . \tag{12}$$

The equation $\boldsymbol{c} = \boldsymbol{D}\boldsymbol{b}$ with the diagonal matrix $\boldsymbol{D}$ shows that in this representation folding/unfolding is simply multiplication/division of the coefficients b_j and c_j by the eigenvalues or the inverse of eigenvalues, respectively, and depends only on the properties of the matrix $\boldsymbol{A}$. Each coefficient b_j and c_j of the linear combination is transformed *independently* of any other coefficient:

$$\text{Folding} \qquad \boldsymbol{x} \to \boldsymbol{y} \qquad b_j \to b_j \cdot \lambda_j = c_j \qquad j = 1 \ldots n \tag{13}$$

$$\text{Unfolding} \qquad \boldsymbol{y} \to \boldsymbol{x} \qquad c_j \to \frac{c_j}{\lambda_j} = b_j \qquad j = 1 \ldots n\,. \tag{14}$$

This operation does not depend on any assumption of the solution $\boldsymbol{x}$.

In an unfolding application the vector $\boldsymbol{y}$ is measured and transformed by $\boldsymbol{c} = \boldsymbol{U}^T\boldsymbol{y}$ to coefficients c_j, which have values influenced by statistical fluctuations of the elements of vector $\boldsymbol{y}$. In the unfolding the coefficients c_j are divided by the eigenvalues λ_j; the statistical fluctuation of coefficient c_j is magnified for small eigenvalues λ_j (i.e. $\lambda_j \ll 1$). Eventually, for very small eigenvalues λ_j, the final result $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{b}$ will be dominated by one or few of the coefficients b_j with small eigenvalues and large statistical errors, and the complete result is unsatisfactory.

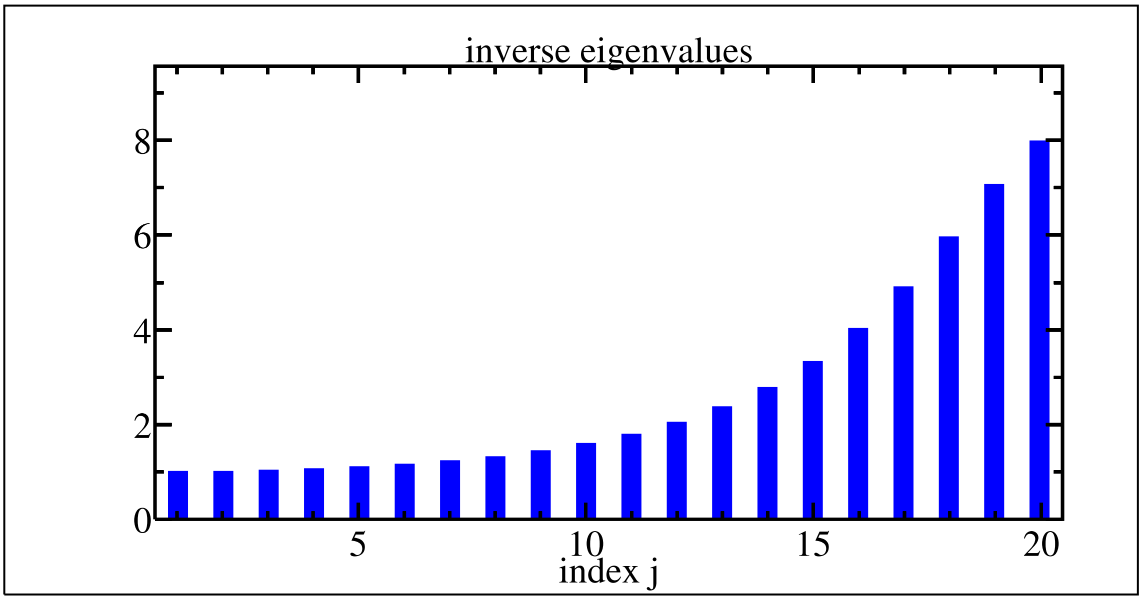

Figure 3: Inverse eigenvalues of the matrix of type (9) for dimension $n = 20$ and migration parameter $\varepsilon = 0.22$. The condition number is $\kappa = \lambda_{\max}/\lambda_{\min} = 7.9$ in this case.

Example: In a numerical example the matrix $\boldsymbol{A}$ has the form of equation (9) with $n = 20$ and a value of the migration parameter of $\varepsilon = 0.22$. The inverse of the eigenvalues of matrix $\boldsymbol{A}$ are shown in Figure 3.

For $\boldsymbol{x}$ the ideal distribution of Figure 4a is assumed; the underlying function is of the form $x\exp(-ax)$. The decomposition of the matrix $\boldsymbol{A}$ according to equation (11) is performed and the coefficients b_j and c_j are calculated as given by equations (12) and (12). These coefficients are shown in Figure 5a. In addition this figure shows, calculated by standard error propagation, the error level of the coefficients, of the folded distribution of Figure 4a with Poisson distributed bin contents. Figure 5a (histogram labeled true) shows, that the coefficients of the true distribution decrease rapidly with increasing value j of the index of the coefficient, by roughly three orders

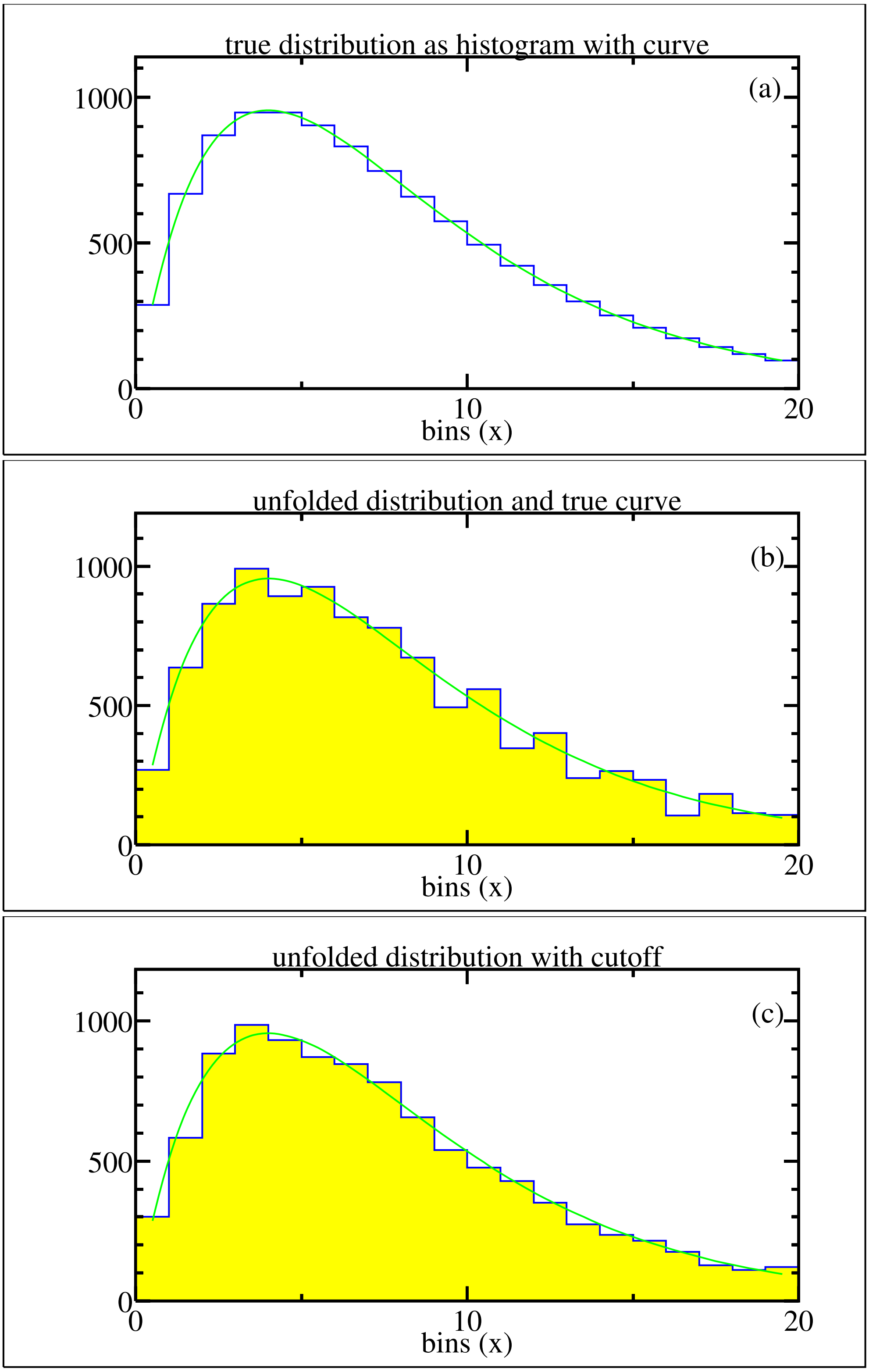

Figure 4: Original (true) distribution (a) and two results from unfolding ((b) and (c)). Result (b) has been obtained from all 20 coefficients, and for result (c) a sharp cut-off after 10 coeeficients has been applied.

of magnitude. The coefficients of the folded distribution, obtained from the true distribution application of the resolution matrix, drop even faster, because it is more smooth due to the migration effect. Of course the relation $b_j/c_j = \lambda_j$ is valid (compare equation (13)). The last coefficient b_j in Figure 5a is reduced by the inverse of the condition number of the matrix, which is $\kappa = 7.9$ in this case, and is identical to the last inverse eigenvalue of Figure 3a.

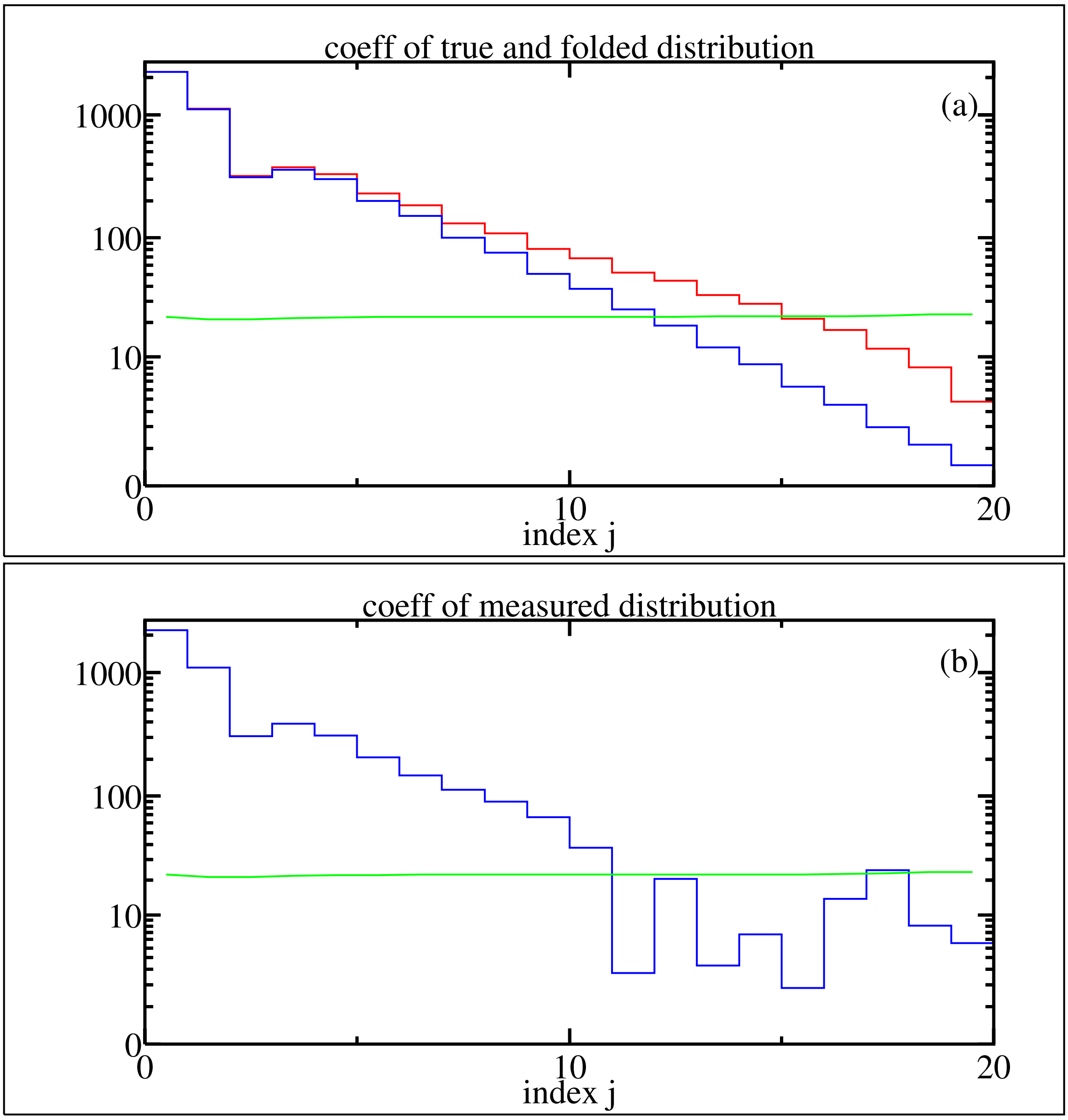

Figure 5: The absolute values of coefficients b_j and c_j are shown for $j = 1, 2, \ldots 20$. The coefficients b_j and c_j for the true distribution and the folded distribution (without meaurement errors) are shown in (a), together with the error estimate for the coeffients b_j calulated by error propagation. The coeffients c_j from the simulated measured distribution of Figure 7 are shown in (b), together with the error estimate. The statistical error of each coeficient is almost the same and thus for j above 12 the statistical error becomes larger than the smaller coefficient of the folded distribution. In (b) the coefficient for j above 12 are dominated by statistical errors and even the sign is not determined by the data.

It is interesting to look at the eigenvectors u_j. The components of the first eigenvector u_1, which

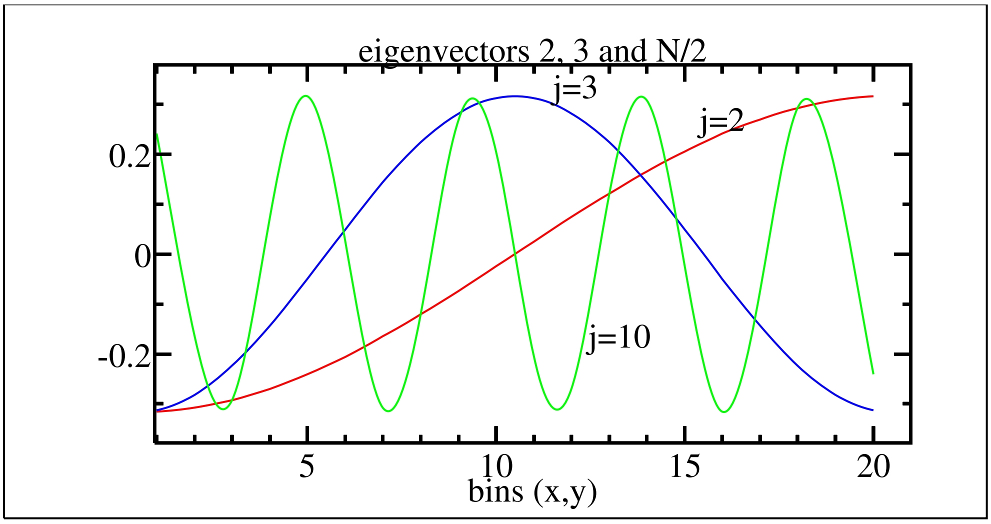

Figure 6: The eigenvectors corresponding to the eigenvalues λ_2, λ_3 and λ_{10} are visualized by interpolating curves. In general there are $j-1$ zeroes for the eigenvector λ_j. The eigenvector corresponding to the eigenvalue λ_1 is constant (i.e. all components have the same value).

corresponds to the eigenvalue $\lambda_1 = 1.0$, are all the same. Thus the coefficient c_1 is proportional to the total sum of the measured distribution and is not at all influenced by the migration. In Figure 6 some of the eigenvectors are visualized by functions, interpolating the components. In this visualization the various eigenvectors are rather similar to the set of Legendre polynomials. The eigenvector corresponding to eigenvalue λ_j has $j-1$ zeros, and the curvature of the visualized eigenvectors is rapidly increasing with index j; the components of the last eigenvector u_n have alternating sign for the bins. Thus coefficient b_1 and c_1 are identical (eigenvalue $\lambda_1 = 1.0$) and are proportional to the sum, which has a small relative statistical error. On the other hand the coefficient b_n with alternating signs of the components of the eigenvector has a small absolute value and its measured value will have a large relative statistical error. The value of c_n will be multiplied by $1/\lambda_n = 7.9$) in unfolding, introducing a large bin-to-bin oscillation into the result of unfolding.

Figure 7 shows the simulated measured distribution corresponding to the true distribution shown in Figure 4a. In order to obtain this distribution first the true distribution is folded by matrix $\boldsymbol{A}$; then each bin content is represented by a random number from the Poisson distribution with a mean of the expected value.
The coefficents for this measured distribution are calculated; they are shown in Figure 5, together with the expected statistical error. As expected from the size of the errors all coefficients with an index above about $j = 12$ are dominated by the statistical error and therefore do not significantly contribute to the precision of the measurement. For indices above $j = 12$ even the sign of the coeffient can not be determined by the measurement.
Using all the measured coefficients of Figure 7 for the unfolding the result of Figure 4b is obtained (this is equivalent to the complete solution of equation (10). This result shows large fluctuations around the expected values shown by the curve. The fluctuations are due to the contributions from indices above $j = 12$, which represent noise and are magnified in the unfolding because of the large values of ther inverse eigenvectors. The result is clearly unsatisfactory.

Because all measured coefficients c_j with j above a value of 12 are dominated by statistical errors (noise) their use in the unfolding makes no sense. A sharp cut-off after index $j = 12$

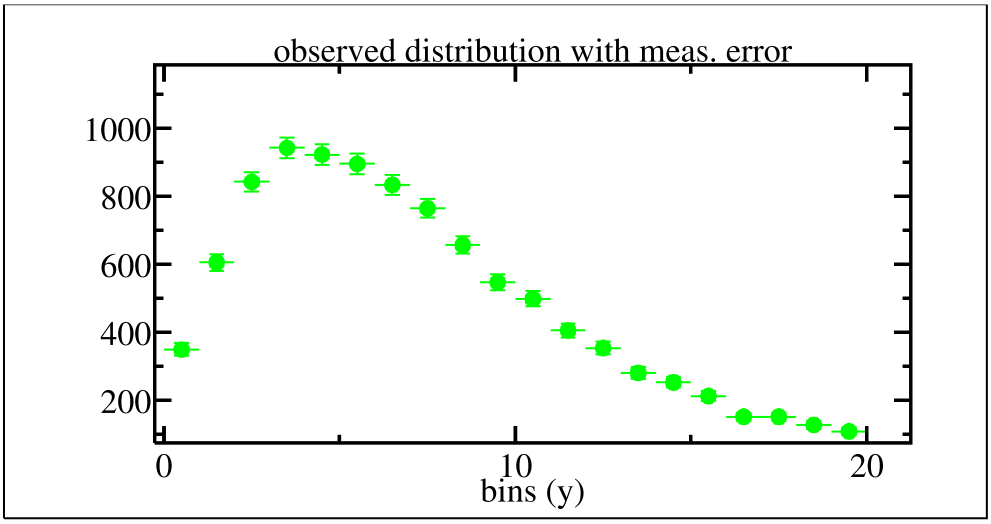

Figure 7: The simulated measured distribution corresponding to the true distribution shown in Figure 4a. The distribution is obtained by folding the true distribution, followed by simulation of the Poisson distribution for each bin.

or even after index $j = 10$ will not remove any useful information from the measurement. The unfolding result using only measured coefficients c_j up to $j = 10$ is shown in Figure 4c; compared to Figure 4b large fluctuations are suppressed and the results seems to be acceptable. Of course the fine structure of the true distribution expressed by the true coefficients b_j with $j > 10$ is not included in the solution and this may represent a bias. It is an unavoidable bias because these coefficients can not be measured.
The covariance matrix of the result can be calculated by standard error propagation and the standard deviations of the bin contents will correspond to the statistical fluctuations of each bin content. However it is clear that the covariance matrix is singular and has only rank 10 in this case, because the 20 bins are obtained from 10 measured coefficients (10 degrees of freedom). This property is inherent to the cut-off method and to the regularization method, and was already mentioned in [9]. Such singularity of the covariance matrix can be avoided if the final transformation is to a number of bins identical to the degree of freedoms; thus only a limited number of bins can be obtained in a measurement with large miration effects.

This method of using a sharp cut-off has to be compared to the regularization method. It has been shown [9] that the use of a regularization function of the type of equation (5) is equivalent to a *smooth* cut-off; essentially the measured coefficients c_j are multiplied by a factor depending on the curvature of the orthogonal contributions (see section 3).

Sometimes the iterative solution of unfolding problems expressed by equations of the type (10) is proposed without explicit regularization, starting from a "good" initial distribution for $\boldsymbol{x}$. Of course equations of this type (with a square matrix) have a unique solution and iterative solutions are slow compared to the direct solution; after a large number of iterations with convergence the same result as by direct solution will be obtained. However in these proposal only a small number of iterations is recommended. It can be shown that iterative methods can in fact include an implicit regularization [14]: there is a different speed of convergence for the various orthogonal contributions and the small contributions with a small eigenvalue will converge very slowly. Thus after a few iterations the (large) coefficients with large eigenvalues are already accurate; the remaining coefficients are still almost unchanged and thus, for a stop after few iterations, their values are still close to the starting values. There seems to be some subjectivity in stopping "early" enough.

3 The proposed unfolding method

The proposed method is similar to the method used in RUN; the differences are emphasized in this section. It is expected that the proposed modifications results in more stable solutions. The proposed method requires a larger number of elements in the resolution matrix $\boldsymbol{A}$. Like in RUN the regularization is determined by the required number of degrees of freedom, which determines the regularization parameter. The number of degrees of freedom has to be small, but larger enough in order not to reduce significant components of the solution.
Throughout this section figures are shown to clarify the mathematical details. These figures all refer to the example already mentioned in section 1. In a Monte Carlo calculation of all three effects, limited (x-dependent) acceptance, non-linear transformation and finite resolution are simulated. Details on the function and the distorting effects are identical to the published examples [9] and are not repeated here. In total 100 000 "events" are simulated (for "data" and for the MC defining matrix $\boldsymbol{A}$). The input function $f_{\mathrm{MC}}(x)$ is a constant.

It is assumed that the resolution function is given implicitly by a sample of Monte Carlo events, generated according to some function $f_{MC}(x)$. As mentioned in section 1 this function is absorbed in the definition of the resolution matrix $\boldsymbol{A}$. The elements of matrix $\boldsymbol{A}$ are integers; element a_{ij} of matrix $\boldsymbol{A}$ is the number of Monte Carlo events from bin j of $\boldsymbol{x}$, which are measured in bin i of $\boldsymbol{y}$.

In RUN the discretization for $f(x)$ and for $A(y,x)$ was done using cubic B-spline functions [16]; the effect is the same as for simple histograms namely the integral equation is transformed to a system of linear equations (10), however the elements of the vectors are B-spline coefficients instead of bin contents. The advantage is that the parametrized solution is a *smooth* function and the curvature as defined by equation (5) can be exactly written as a quadratic form (6). However the accurate matrix determination of matrix $\boldsymbol{A}$ requires a good Monte Carlo statistic. In RUN statistical fluctuations of the elements of matrix $\boldsymbol{A}$ could not be treated.

Simple histograms are instead proposed here; the elements of the vector $\boldsymbol{y}$ are bin contents (integer numbers). A disadvantage is the fact that the curvature has to be defined by finite differences. For example the second derivative in bin j is proportional to $-x_{j-1} + 2x_j - x_{j+1}$. It is clear that some available resolution is lost if bins with a width as expected for the final resolution would be used. Therefore the number of bins in x and in y should be as large as possible; in fact it is recommended to use initially $m = 2n_{\mathrm{df}}$ bins for $\boldsymbol{x}$ for a final number of degrees of freedom of n_{df}. For y a larger number of bins n ($> m$) is recommended, to avoid a loss of resolution. Thus the number $n \times m$ of elements is large, and a large sample of Monte Carlo events is required to *fill* matrix $\boldsymbol{A}$. The statistical error of the elements a_{ij} eventually can not be neglected.

The elements of the vector $\boldsymbol{x}$ are initially determined by a fit without any regularization. The optimal fit method seems to be the maximum likelihood method which can be based on the Poisson distribution of the counts. Regularization is applied optionally in a second step.

Standard Poisson maximum likelihood fit. Ignoring initially eventual statistical errors of the elements a_{ij} the expected number of events in bin i of $\boldsymbol{y}$ is given by

$$y_i = \sum_{J=1}^{m} a_{ij}\, x_j \; . \tag{15}$$

For the expected number y, as given by this expression, the observed values $\widehat{y}$ follow the Poisson distribution:

$$P_y(\widehat{y}) = e^{-y}\,\frac{y^{\widehat{y}}}{\widehat{y}!} \qquad \text{with normalization} \quad \sum_{\widehat{y}=0}^{\infty} P_y(\widehat{y}) = 1\ . \tag{16}$$

Optimal estimates for the elements x_j are obtained by minimizing the (negative) logarithm of the total likelihood with respect to the elements x_j of vector $\boldsymbol{x}$:

$$F(\boldsymbol{x}) = -\ln \mathcal{L}(\boldsymbol{x}) = -\ln\left[\prod_{i=1}^{n} P_{y_i}(\widehat{y}_i)\right] = \sum_{i=1}^{n} (y_i - \widehat{y}_i \cdot \ln y_i) + \text{const.}\ , \tag{17}$$

where the constant term containing e.g $\widehat{y}_i!$ can be ommited. This expression (17) correctly accounts for small number of entries $\widehat{y}_i$.

An alternative would be to use the (linear) least squares method for the fit. The advantage would be the potential solution by singular value decomposition. However for small number of entries the use of the Poisson distribution seems to be essential. Furthermore the diagonalization used later in the method is almost equivalent to singular value decomposition (eigenvalues are the squares of the singular values).

Fitting with finite Monte Carlo samples. The problem of statistical fluctuations of the elements a_{ij} has been neglected so far. A method to treat the problem within the maximum-likelihood method has been developed by R.Barlow and Chr.Beeston [17]. In this method there is for each source bin x_j, in each bin, some (unknown) expected number of events A_{ji}. For each elements A_{ji} the corresponding number a_{ji} from the Monte Carlo sample is generated by a distribution which is taken to be Poisson. The nice feature of this method is that a bias which would be introduced by ignoring the statistical character of the values of the elements a_{ij} is avoided and the maximum likelihood error is more realistic. In the appendix of this paper the Barlow method is explained, together with a new kind of numerical method of solution.

Combining bins. The likelihood function is a sum over all bins. Combining bins with small number of entries does not introduce any systematic error. The number of elements of the matrix may be large, especially if x and/or y are multidimensional, and a small number of entries (or even zero) in an element may not be uncommon. With limited statistic for data and the Monte Carlo simulation or with a large number of bins the combination (addition) of certain bins can improve the solution.

The combination of bins is based on the total number of entries $\alpha_i = \sum_j a_{ji}$ in the simulation. Bins with $\alpha_i = 0$ (i.e. no Monte Carlo entry) but a nonzero number of data entries $d_i \neq 0$ can not correctly be treated in the likelihood function and are combined with neighbour bins. Bins with a very small number of Monte Carlo entries α_i (regardless of the number of data entries) have a very low weight in the fit and should be avoided. A lower limit k_{limit} of Monte Carlo entries in a bin can be defined; y-bins with $\alpha_i < k_{\text{limit}}$ (e.g. $k_{\text{limit}} = 5$) will be combined with neighbour bins. The combination of bins is done with a cluster algorithm, taking into account the distance between bins in one, two or three dimensions.

Bins i without data events (i.e. $d_i = 0$) *and* without Monte Carlo events (i.e. $\sum_j a_{ji} = 0$) do not contribute to the fit; in general these bins will be bins outside the physical region, but with limited statistic this case may also occur within the physical region.

First option: sharp cut-off of orthogonal contributions: This method is rather similar to the method discussed in section 2. The computational problem is to determine the minimum of $F(\boldsymbol{x})$ (see equation (17)). The standard iterative method is based on the representation for the correction $\Delta\boldsymbol{x}$

$$F(\Delta\boldsymbol{x}) = \Delta\boldsymbol{x}^T \boldsymbol{H} \Delta\boldsymbol{x} + \Delta\boldsymbol{x}^T \boldsymbol{g} + \ldots \tag{18}$$

with the Hessian $\boldsymbol{H}$ (matrix of second derivatives of $F(\Delta\boldsymbol{x})$) and the gradient vector $\boldsymbol{g}$ (first derivatives of $F(\Delta\boldsymbol{x})$). A Newton step is then calulated from equation

$$\boldsymbol{H}\Delta\boldsymbol{x} + \boldsymbol{g} = 0 \; . \tag{19}$$

Convergence is usually fast for good starting values and the covariance matrix is equal to the inverse of the Hessian. The starting values can be calculated by a linear least square fit, based on the approximation of the Poisson distribution by a Gaussian distribution for each bin.
A sharp cut-off as discussed in the example of section 2 requires a diagonalization of the symmetric matrix $\boldsymbol{H}$,

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \tag{20}$$

with a diagonal matrix $\boldsymbol{D}$ and a transformation matrix $\boldsymbol{U}$. By a transformation (rotation) in $\boldsymbol{x}$-space linear combinations of the $\boldsymbol{x}$-components are obtained with a diagonal covariance matrix, with variances of the linear combinations given by the inverse of the eigenvalues of matrix $\boldsymbol{D}$. A cut-off is done at some index j followed by backtransformation to the $\boldsymbol{x}$-space of bins-contents using the transformation matrix $\boldsymbol{U}$. Of course the number of degrees of freedom of the solution is given by the number of linear combinations kept after the sharp cut-off. It is recommended to choose the number of bins equal to twice the final number of degrees of freedom. Then it is simple to combine finally two bins of the result x to one bin, and the covariance matrix for this reduced number of bins should have the full rank (= number of degrees of freedom).

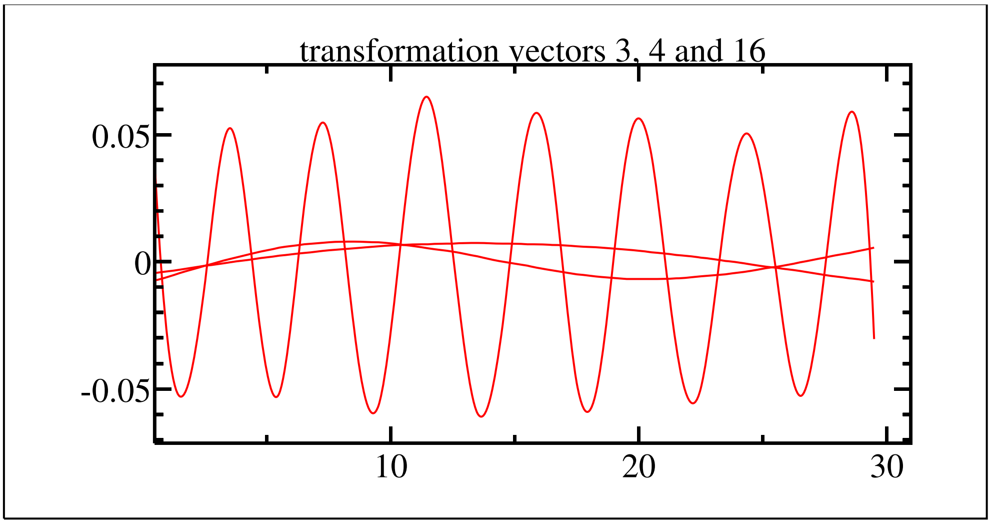

Figure 8: Selected column vectors of the complete transformation matrix defined in the regularization procedure. They correspond to the curvature eigenvalues S_{33}, S_{44} and $S_{16,16}$. Visualization is done by curves interpolating the components. The amplitude associated which each vector all have the same standard deviation of 1.

Second option: regularization. In this option the regularization is based on the second derivative of the result according to equation (5), which can be expressed by a quadratic form $\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}$ with a positive-semidefinite matrix $\boldsymbol{C}$. The second derivative in bin j is proportional to

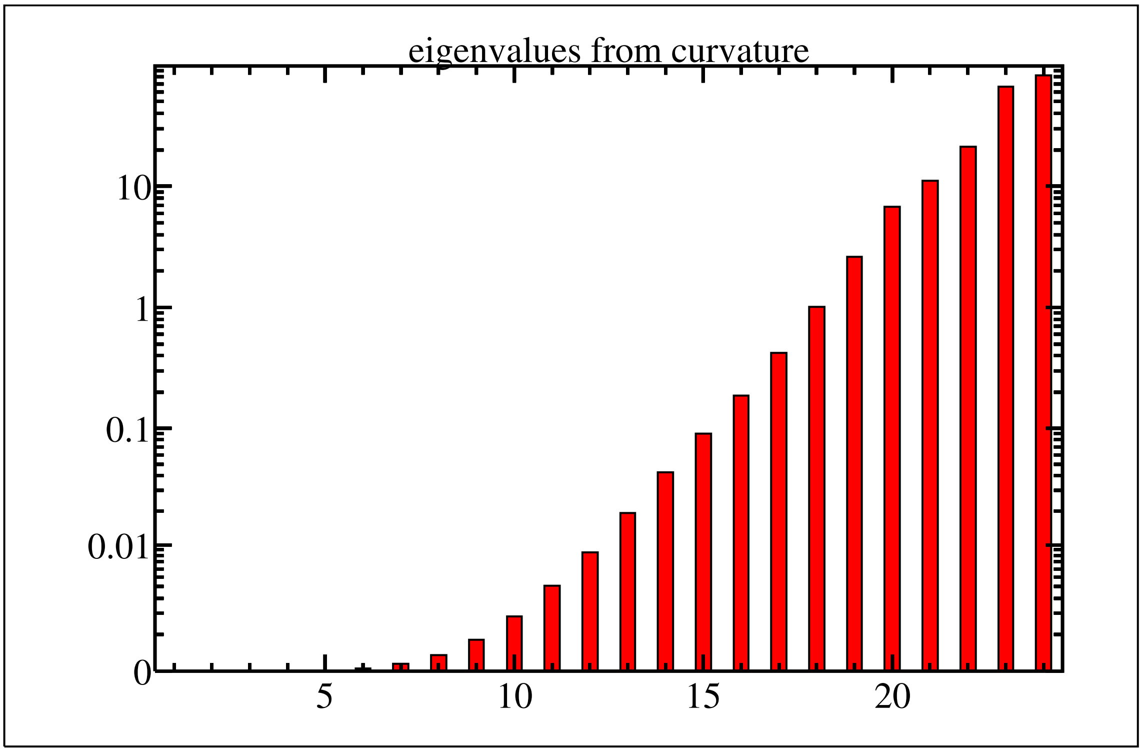

Figure 9: The eigenvalues after the curvature transformation. The values are very rapidly increasing for orthogonal contributions for increasing index value.

$-x_{j-1} + 2x_j - x_{j+1}$ and the contribution of this bin to the curvature matrix $\boldsymbol{C}$ is given by

$$\begin{pmatrix} & & \dots & & \\ & C_{j-1,j-1} & C_{j-1,j} & C_{j-1,j+1} & \\ \dots & C_{j,j-1} & C_{jj} & C_{j,j+1} & \dots \\ & C_{j+1,j-1} & C_{j+1,j} & C_{j+1,j+1} & \\ & & \dots & & \end{pmatrix} = \begin{pmatrix} & & \dots & & \\ & +1 & -2 & +1 & \\ \dots & -2 & +4 & -2 & \dots \\ & +1 & -2 & +1 & \\ & & \dots & & \end{pmatrix} . \tag{21}$$

In principle the same procedure is used as in RUN; the mathematical details are given elsewhere [9, 15]. Here a simple explanation is given on the standard mathematical operations[1] used. Regularization is done by adding the term $\tau \cdot \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}$ to the function $F(\Delta \boldsymbol{x})$ of equation (18). Exactly as in the first option the Hessian is diagonalized.

$$\boldsymbol{H} = \boldsymbol{U} \boldsymbol{D} \boldsymbol{U}^T \qquad \qquad \boldsymbol{H}^{-1} = \boldsymbol{U} \boldsymbol{D}^{-1} \boldsymbol{U}^T = \left[\boldsymbol{U} \boldsymbol{D}^{-1/2} \right] \left[\boldsymbol{D}^{-1/2} \boldsymbol{U}^T \right] \tag{22}$$

Up to this step everything is identical to the cut-off option. Using transformation matrix $\boldsymbol{U}\boldsymbol{D}^{-1/2}$ the vector $\boldsymbol{x}$ is transformed to linear combinations $\widehat{\boldsymbol{x}}$, which are orthogonal with all variances equal to 1 (the covariance matrix is the unit matrix). Because the covariance matrix is equal to the unit matrix, every additional pure rotation will not change the (unit) covariance matrix. In terms of the transformed vector the regularization term can now be written in the form $\tau \cdot \widehat{\boldsymbol{x}}^T C_U \widehat{\boldsymbol{x}}$, where C_U is the transformed curvature matrix $\boldsymbol{C}$. Now a further diagonalization can be done of matrix C_U:

$$\tau \cdot \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x} \;\rightarrow\; \tau \cdot \widehat{\boldsymbol{x}}^T \boldsymbol{C}_U \widehat{\boldsymbol{x}} = \tau \cdot \widehat{\boldsymbol{x}}^T \boldsymbol{U}_C \; \boldsymbol{S} \; \boldsymbol{U}_C^T \widehat{\boldsymbol{x}} \tag{23}$$

[1] In a publication the method has been described to "*have certain mathematical complications*", but it is based only on standard linear algebra of symmetric matrices.

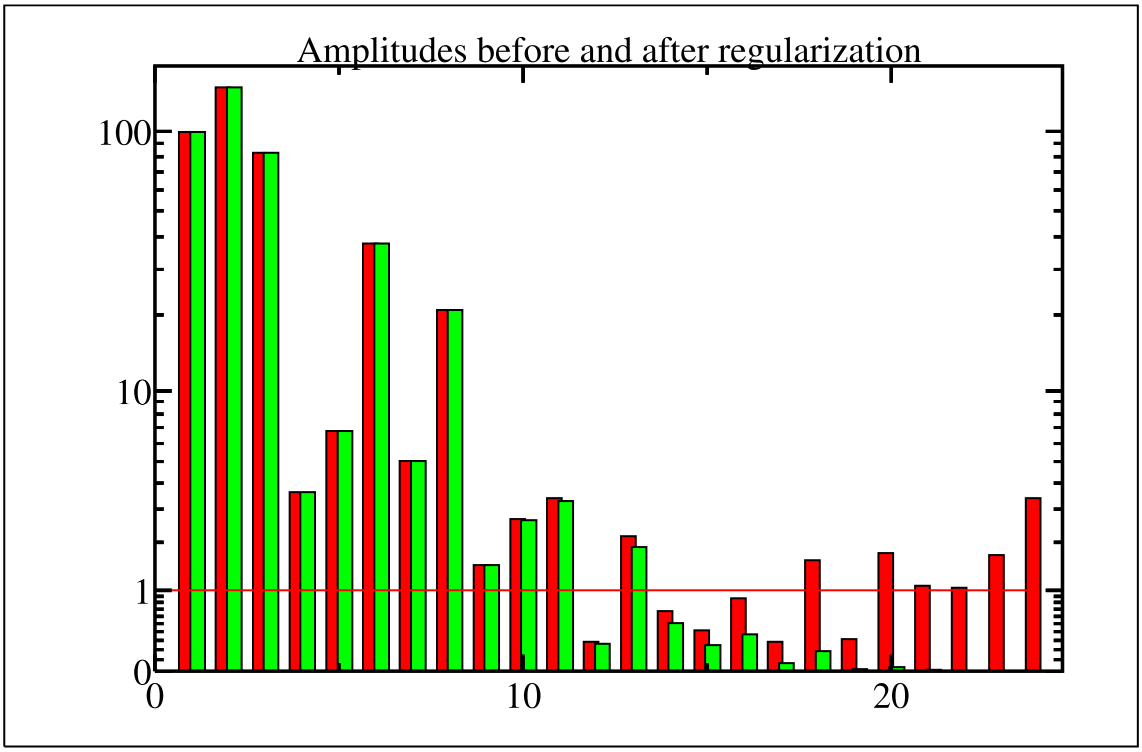

Figure 10: The amplitudes before (left bars) and after regularization (right bars). The statistical error of all amplitudes is equal to 1, which is indicated by the horizontal line. The vertical scale is linear at the bottom and makes a transition to a logarithmic scale at the top.

with a diagonal matrix $\boldsymbol{S}$ and a rotation matrix $\boldsymbol{U}_C$. This diagonalization can be used to define a pure rotation from the linear combination $\widehat{\boldsymbol{x}}$ to another linear combination $\widetilde{\boldsymbol{x}}$

$$\widehat{\boldsymbol{x}} \rightarrow \widetilde{\boldsymbol{x}} = \boldsymbol{U}_C^T \widehat{\boldsymbol{x}} \tag{24}$$

The components of the new vector $\widetilde{\boldsymbol{x}}$ still have the unit matrix as covariance matrix. The complete transformation from $\boldsymbol{x}$ to $\widetilde{\boldsymbol{x}}$ is the effect of the transformation by $\boldsymbol{U}\boldsymbol{D}^{-1/2}$ and by $\boldsymbol{U}_C$. The algebra can be explained in other words: the error ellipsoid related to the Hessian is first rotated to have the axes parallel to the axes of the new system. By a change of the scales the ellipsoid is transformed to a sphere, which will remain a sphere for any further rotation. Now a last rotation is done to bring the axes into the order of increasing curvature.

Some columns of the complete (product) transformation are shown in Figure 8. All linear combinations obtained have the same precision (standard deviation of the coefficient is one). As seen in the Figure linear combinations with large index j are oscillating with large amplitude. The diagonal elements S_{jj} are the (statistically independent) contributions of the elements of $\widetilde{\boldsymbol{x}}$ to the total curvature. If sorted according to increassing value of S_{jj} the first values S_{11} and S_{22} will be very small (almost constant and linear term with zero curvature), and with increasing index j the value of S_{jj} will increase rather fast. The spectrum of eigenvalues S_{jj} is shown in Figure 9. The rapid increase over many orders of magnitude with increasing index is clearly visible.

In terms of the linear combinations $\widetilde{\boldsymbol{x}}$ regularization is simply given by

$$(\widetilde{x}_j)_{\text{reg}} = \frac{1}{1 + \tau \cdot S_{jj}} (\widetilde{x}_j)_{\text{unreg}} \ . \tag{25}$$

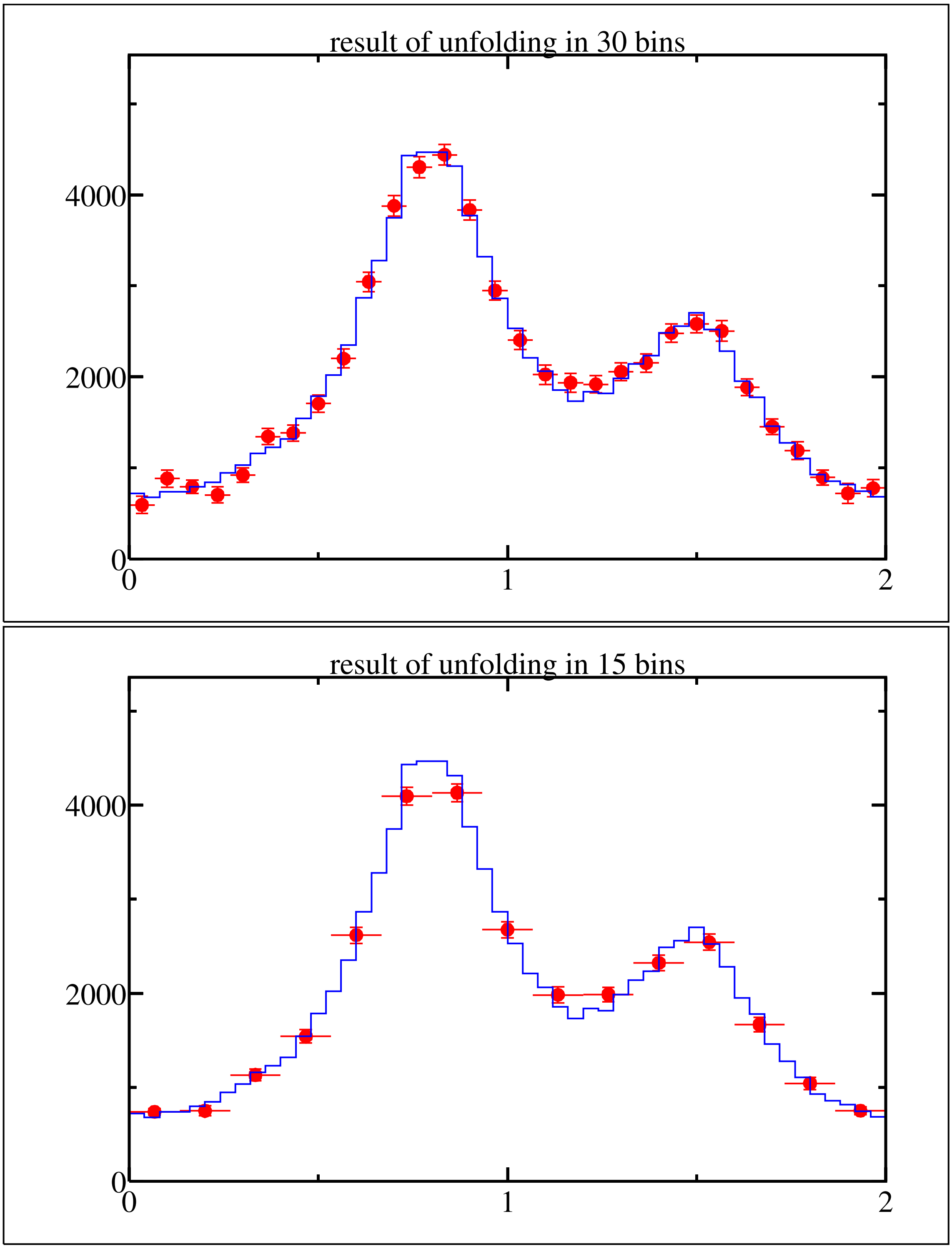

Figure 11: The unfolding result after regularization with 15 degrees of freedom with 30 bins (top) and with 15 bins (bottom). For comparison the original (true) distribution is shown by a histogram. The input to unfolding are the data from Figure 1.

and this simple form is the reason for the transformations made before. The value of the regularization parameter τ is not yet specified, but will be new determined now. The first factors (small j) on the right-hand-side of equation (25) will be close to 1.0; for a value $\tau = 1/S_{kk}$ the

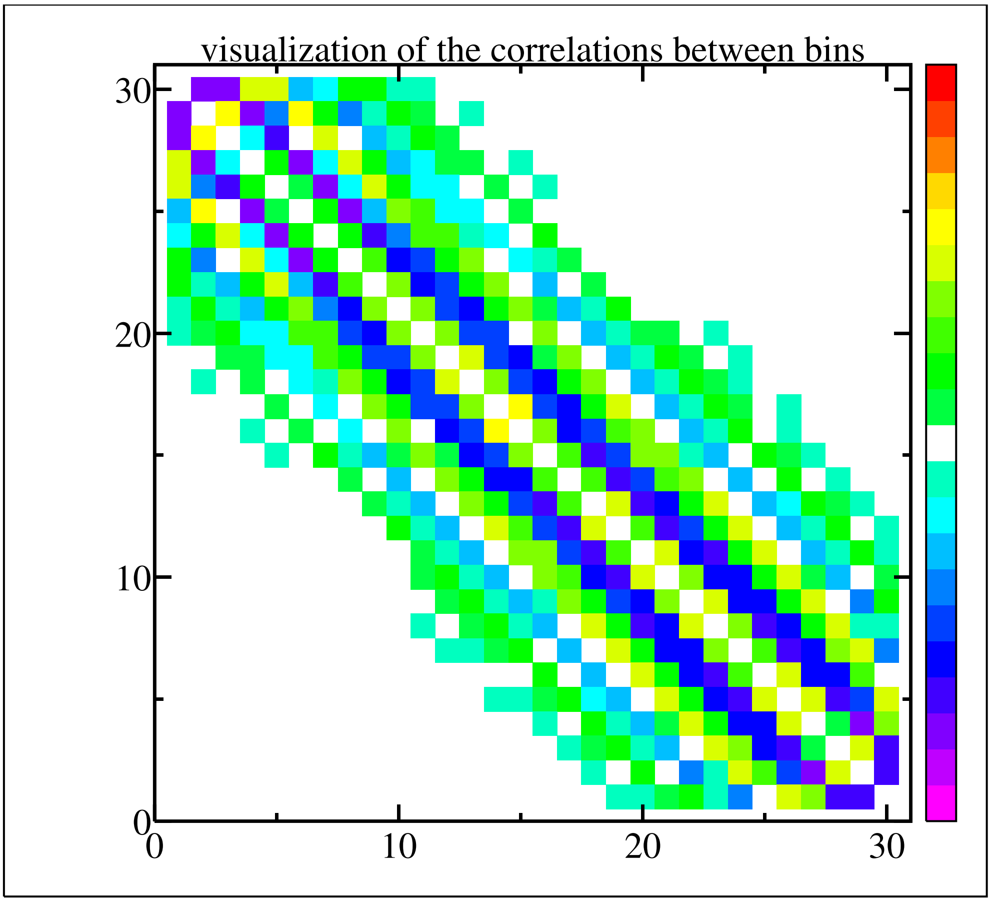

Figure 12: A visualization of the correlations ρ between bin contents by colour display, taken from the covariance matrix of the 30-bin regularization. The magnitude of the correlation is expressed by a colour according to the bar on the right, which shows the colours from $\rho = +1$ (top, red) to $\rho = -1$ (bottom, magenta).

factor will be $1/2$ and for indices $j > k$ will rapidly decrease towards zero. The sum of all factors can be called the *effective* number of degrees of freedom, and this fact can be used to determine the value of the regularization parameter τ from the required number of degrees of freedom, i.e the regularization parameter τ is determined from the value of n_{df} in the equation

$$n_{\mathrm{df}} = \sum_{j=1}^{m} \frac{1}{1 + \tau \cdot S_{jj}} . \tag{26}$$

Thus instead of the direct value of the regularization parameter τ the required number of degrees of freedom has to be specified and determines the degree of regularization. This number can be taken from the spectrum of the coefficients (see Figure 10). The insignificant part (large j) is clearly visible in the spectrum and separated from the significant part (small j). The selected value of n_{df} should be equal to or larger than the number of significant terms.

Returning to the example the effect of the regularization is shown in Figure 10. The elements of the vector $\widetilde{\boldsymbol{x}}$ are called amplitudes; these amplitude determine the contribution of the column vectors (compare Figure 8) to the final solution. In Figure 10 the unregularized amplitudes, which have standard deviation one, are shown by the left bars; amplitudes above index 15 are compatible with one and represent noise. They would however make a large contribution to the solution, because the corresponding column vectors (Figure 8) are large. the regularization effectively damps the amplitude around and above index 15, which has been chosen as the degree of freedom here. The significant amplitudes are not affected by the regularization.

The final result of the example (measured distribution in Figure 1) is shown in Figure 11. The top figure shows 30 data points with error bars together with the original (true) distribution; within errors the original distribution is nicely reproduced. The rank of the covariance matrix is about 15, which was chosen as the effective number of degrees of freedom; thus inversion of the covariance matrix, needed e.g. for a least-square fit of a model to the data, is not possible. Figure 12 shows a visualization (at least in a coloured version) of the correlations between bins of the solution. Correlations (range $-1 \ldots +1$) are shown by colours (top $+1$ to -1). Correlations between neighbor bins are about $+0.5$ and correlations to the next bin about -0.5. These correlation are a direct consequance of the finite resolution. Although the large number of 30 data points seems to be attractive, the data points should be reduced to 15 data points by combining two bins to one, which then have a full-rank covariance matrix. This set of data points is shown in the bottom of Figure 11. The broader bins of this set of data points are a consequence of the limited acceptance and finite resolution of the measurement.

Appendix: Fit of binned distributions

The unfolding fit is essentially a fit of binned distribution. This appendix describes the methods used in unfolding for the case of good statistics (statistical error can be neglected) and for limited statistics. Since the latter is based on the method by R.Barlow and Chr.Beeston [17] the nomenclature and some explanations are taken from their paper. In the unfolding application the *number of sources* is equal to the original bins.

As the result of measurements there is a sample of real data of size N_D, each member of which consists of a set of values $\{\boldsymbol{x}\}$. The data are binned, dividing the one- or multi-dimensional space of the $\{\boldsymbol{x}\}$ into n bins. This subdivision gives a set of numbers $\{d_1, d_2, \ldots, d_n\}$, where d_i is the number of events in the real data that fall into bin i. The values d_i are integers 0, 1, ...

$$d_i = \text{number of events in the real data that fall into bin } i \tag{27}$$

$$N_D = \sum_{i=1}^{n} d_i = \text{total number in the data sample} \tag{28}$$

The real data arise from a number of sources (or physical processes) and the aim is to determine the proportions P_j of the different sources in the data from the statistical data d_i and from models for the sources. One has to distinguish the case, where analytic forms are available for the distribution of the sources and where the distribution has to be calculated by a Monte Carlo simulation.

Models with analytic forms. Analytic forms available for the distribution of the sources can be used to calculate, by integration over the bins, numbers a_{ij} proportional to the expected number of events from source j in bin i; these values have no statistical errors.

$$P_j = \text{proportion of source } j \text{ in the data} \tag{29}$$

$$N_j = \sum_{i=1}^{n} a_{ij} \tag{30}$$

$$a_{ij}/N_j = \text{fraction of events from source } j \text{ expected in bin } i \tag{31}$$

The predicted number of events in bin i is, for the proportions P_j, given by the sum

$$f_i = N_D \sum_{j=1}^{m} P_j \, \frac{a_{ij}}{N_j} \; , \tag{32}$$

where the proportions P_j should sum to unity. It is convenient to introduce strength factors p_j and to calculate the number of event expected in a bin i from the strength factors and the numbers a_{ij}:

$$\begin{aligned} p_j &= N_D\, P_j / N_j = \text{strength factors} && (33)\\ f_i &= \sum_{j=1}^{m} p_j\; a_{ij} = \text{number of events expected to fall into bin } i && (34) \end{aligned}$$

A standard method to estimate the factors p_j is the least squares method, where the p_j are adjusted to minimize the function $F(\boldsymbol{p})$, defined by

$$\boxed{F(\boldsymbol{p}) = \sum_{i=1}^{n} \frac{(d_i - f_i)^2}{d_i}}\,, \tag{35}$$

assuming that the distribution for d_i is Gaussian with a standard deviation of $\sqrt{d_i}$. The distribution for d_i in reality is Poisson, but the Gaussian is assumed to be a good approximation to the Poisson distribution for large numbers, say $d_i > 10$. This linear least squares fit can be solved directly without iterations and without the need for initial values of the factors p_j. The value $F(\boldsymbol{p})$ at the minimum will follow a χ^2-distribution for $n - m$ degrees of freedom, if the Gaussian distribution is justified, and thus provides a number which can be used to check the goodness-of-fit.

In the approximation of the Poisson distribution by a Gaussian, values d_i fluctuating down will get a smaller absolute standard deviation and thus a larger weight, compared to values d_i fluctuating up. For smaller values d_i the result will be biased and the fit is distorted. However the fit may still be useful because of its simplicity and because the parameter values obtained can be used as starting values for a more appropriate nonlinear fit. A better approximation of the Poisson distribution is obtained by replacing d_i in the denominator by f_i; however then the minimization requires iterations and initial values.

Often some or many of the data d_i are small or even zero; this is especially the case for multi-dimensional data, where even a modest subdivision of the axes will give a large total number n of bins. The least squares methid is no longer appropriate.

Instead the correct statistical distribution, the Poisson distribution, has to be used. If the mean value of entries in a bin is f, then the observed values $d = 0, 1, \ldots$ follow the Poisson distribution

$$P_f(d) = e^{-f}\,\frac{f^d}{d!} \qquad \text{with normalization} \quad \sum_{d=0}^{\infty} P_f(d) = 1\;. \tag{36}$$

The variance (or square of the standard deviation) is equal to the mean value f. Estimates of the factors are found by maximizing the total likelihood

$$\mathcal{L}(\boldsymbol{p}) = \prod_{i=1}^{n} P_{f_i}(d_i) = \prod_{i=1}^{n} e^{-f_i}\,\frac{f^{d_i}}{d_i!} \tag{37}$$

with respect to the parameters $\boldsymbol{p}$ or equivalently, by minimizing the negative logarithm of the likelihood $F(\boldsymbol{p}) = -\ln \mathcal{L}(\boldsymbol{p})$:

$$\boxed{F(\boldsymbol{p}) = \sum_{i=1}^{n} (f_i - d_i \cdot \ln f_i)} \tag{38}$$

omitting constant factors in the product like $1/d_i!$. This expression correctly accounts for small numbers of data events d_i in a bin or even zero data events in some bins. The method is called *binned maximum likelihood* fit.

Constant factors in the likelihood product can be omitted or added, because they do no influence the position of the minimum and do not change the shape of the function. There is some advantage in redefining the function F to be minimized by adding some constants as

$$\boxed{F(\boldsymbol{p}) = \sum_{i=1}^{n} g_i \qquad \text{with} \quad g_i = \begin{cases} (f_i - d_i) - d_i \cdot \ln(f_i/d_i) & \text{if } d_i > 0 \\ f_i & \text{if } d_i = 0 \end{cases}} \,. \tag{39}$$

The position of the minimum and the shape of F is not changed by his modification, but now the minimum value of F can be interpreted, as shown below. Considering one term of the sum only and assuming $d_i > 0$ the approximation

$$\begin{aligned}(f_i - d_i) - d_i \cdot \ln \frac{f_i}{d_i} &= (f_i - d_i) - d_i \cdot \ln \left(1 + \frac{f_i - d_i}{d_i}\right) \\ &\approx (f_i - d_i) - d_i \left(\frac{(f_i - d_i)}{d_i} - \frac{1}{2}\frac{(f_i - d_i)^2}{d_i^2}\right) = \frac{1}{2}\frac{(f_i - d_i)^2}{d_i} \end{aligned} \tag{40}$$

based on the expansion $\ln(1+\varepsilon) \approx \varepsilon - \varepsilon^2/2$ can be done, which shows, that the value F multiplied by 2 is (within this approximation) identical to $\chi^2/2$. Thus, as in least squares methods, the value of the function can be interpreted in terms of the goodness-of-fit χ^2.

Minimization is now a nonlinear problem requiring iterations. A good method to calculate a step vector $\Delta \boldsymbol{p}$ in each iteration is the Newton method, based on the quadratic approximation of F. The m-dimensional step vector is defined by

$$\boldsymbol{H} \Delta \boldsymbol{p} = -\boldsymbol{g} \tag{41}$$

where the elements of the gradient $\boldsymbol{g}$ and of the Hessian $\boldsymbol{H}$ can be calculated by the following sums;

$$g_j = \frac{\partial F}{\partial p_j} = \sum_i a_{ij} - \frac{d_i}{f_i} a_{ij} \qquad\qquad H_{jk} = \frac{\partial^2 F}{\partial p_j \partial p_k} = \sum_i \frac{d_i}{f_i^2} a_{ij} a_{ki} \tag{42}$$

Models with MC simulation. Often no analytic forms are available for the distributions of the sources. Instead there are samples of data, generated according to the model of the source by Monte Carlo simulation (and perhaps measured background samples). These data samples can be binned in the same way as the real data, giving a set of integer numbers $\{a_{1j}, a_{2j}, \ldots a_{nj}\}$ for source j. Now both the real data d_i and the data a_{ij} are of statistical nature with integer values 0, 1, ...

$$a_{ij} \;=\; \text{number of Monte Carlo events from source } j \text{ in bin } i \tag{43}$$

$$N_j \;=\; \sum_{i=1}^{n} a_{ij} = \text{total number in the MC sample for source } j \tag{44}$$

The Monte Carlo samples are finite, leading to statistical fluctuations in the numbers a_{ij}. If the Monte Carlo samples are much larger than the data sample, one may ignore these fluctuations and use

$$f_i = \sum_{j=1}^{m} p_j \, a_{ij} \tag{45}$$

as before; the fluctuations in the a_{ij} are damped by the factor N_D/N_j. Usually it is assumed that a factor of 10 in the size of the Monte Carlo samples compared to the date sample should be large enough.

However often the statistical fluctuations in the a_{ij} of the Monte Carlo sample can not be ignored, and one has to consider them together with the statistical fluctuations of the data d_i. A method to treat the problem within the maximum-likelihood method has been developed by R. Barlow and Chr. Beeston [17] and the following derivation follows closely this paper. In this method there is for each source, in each bin, some (unknown) expected number of events A_{ij}

$$f_i = \sum_{j=1}^{m} p_j A_{ij} \; . \tag{46}$$

From each A_{ij} the corresponding a_{ij} is generated by a distribution which can be taken as Poisson. The total likelihood is the combined probability of the observed $\{d_i\}$ and of the observed $\{A_{ij}\}$ and one has have to minimize

$$F(\boldsymbol{p}, \boldsymbol{A}) = -\ln \mathcal{L} = \sum_{i=1}^{n} [f_i - d_i \cdot \ln f_i] + \sum_{i=1}^{n} \sum_{j=1}^{m} [A_{ij} - a_{ij} \cdot \ln A_{ij}] \tag{47}$$

or, more complete, showing directly the dependence on the strength factors p_j and the expected numbers of events A_{ij}:

$$F(\boldsymbol{p}, \boldsymbol{A}) = -\ln \mathcal{L} = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} p_j A_{ij} - d_i \cdot \ln \sum_{j=1}^{m} p_j A_{ij} \right] + \sum_{i=1}^{n} \sum_{j=1}^{m} [A_{ij} - a_{ij} \cdot \ln A_{ij}] \; . \tag{48}$$

Estimates for the p_j and the A_{ij} are found by minimising the function F. The interest is of course in the determination of the p_j. This is a minimisation problem in $m + m \times n$ unknowns, which is a rather large number, which has to be determined from the n observed data values $\{d_1, d_2, \ldots, d_n\}$ and to the $m \times n$ Monte Carlo values a_{ij}. Conditions for the minimum are obtained by the fact, that the derivatives of F with respect to the strength factors p_j and the event numbers A_{ij} have to be zero. Two sets of equations are obtained, the set

$$\sum_{i=1}^{n} \left(1 - \frac{d_i}{f_i} \right) A_{ij} = 0 \qquad\qquad j = 1, 2, \ldots m \tag{49}$$

from the differentials with respect to the strength factors p_j, and

$$1 - \frac{a_{ij}}{A_{ij}} + p_j - \frac{d_i \, p_j}{f_i} = 0 \qquad\qquad j = 1, 2, \ldots m \qquad i = 1, 2, \ldots n \tag{50}$$

from the differentials with respect to the A_{ij}. Thus one has to solve a system of $m + m \times n$ simultaneous, nonlinear equations for the $m + m \times n$ unknowns.

In the original paper by Barlow the equations (50) are rewritten in the form

$$1 - \frac{d_i}{f_i} = \frac{1}{p_j} \left(\frac{a_{ij}}{A_{ij}} - 1 \right) \qquad\qquad j = 1, 2, \ldots m \qquad i = 1, 2, \ldots n \tag{51}$$

and then it is shown that the problem can be remarkably simplified. The left hand side depends on index i only, so it can be written as

$$t_i = 1 - \frac{d_i}{f_i} \tag{52}$$

and one can express A_{ij} in the form

$$A_{ij} = \frac{a_{ij}}{1 + p_j \, t_i} \; . \tag{53}$$

For a given set of values p_j the $m \times n$ unknowns A_{ij} are given by the n unknowns t_i, which are defined above. If $d_i = 0$ then $t_i = 1$, and if not, then

$$\frac{d_i}{1-t_i} = f_i = \sum_{j=1}^{m} p_j A_{ij} = \sum_{j=1}^{m} p_j \frac{a_{ij}}{1+p_j\ t_i} \tag{54}$$

The equations for different bin indices are *not coupled* and the values t_i are determined independently for each bin i. This is a great simplification: instead of $m \times n$ unknowns A_{ij} there are only n unknowns t_i. Since, for a given bin i, all the m values A_{ij} are combined to one value f_i, which has to be compared to one measured value d_i, it is clear that this reduction of the number of parameters per bin from m to 1 has to be possible.

The following treatment is slightly different from BARLOWs treatment. From now on the t_i are considered as the essential parameters, not the A_{ij}, which are not used in the following, but eliminated using the relation (53). The function (47) to be minimised is written in the form

$$F(\boldsymbol{p},\boldsymbol{t}) = -\ln \mathcal{L} = \sum_{i=1}^{n} \left[\left(\sum_{j=1}^{m} \frac{p_j a_{ij}}{1+p_j t_i} \right) - d_i \cdot \ln \left(\sum_{j=1}^{m} \frac{p_j a_{ij}}{1+p_j t_i} \right) \right] \\ + \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{a_{ij}}{1+p_j t_i} - a_{ij} \cdot \ln \frac{a_{ij}}{1+p_j t_i} \right) \tag{55}$$

with the m unknowns p_j and the n unknowns t_i.

Fitting using finite Monte Carlo samples.
Minimization of the function $F(\boldsymbol{p},\boldsymbol{t})$ can be done iteratively based on the quadratic approximation at each iteration. The following system of linear equations with m unknown elements of the vector $\Delta \boldsymbol{p}$ and n unknown elements of the vector $\Delta \boldsymbol{t}$ has to be solved:

$$\left(\begin{array}{c|c} \boldsymbol{H} & \boldsymbol{C} \\ \hline \boldsymbol{C}^T & \boldsymbol{T} \end{array} \right) \cdot \left(\begin{array}{c} \Delta \boldsymbol{p} \\ \hline \Delta \boldsymbol{t} \end{array} \right) = \left(\begin{array}{c} \boldsymbol{g} \\ \hline \boldsymbol{\gamma} \end{array} \right) \tag{56}$$

Here the elements of matrix $\boldsymbol{H}$ and vector $\boldsymbol{g}$ are defined as before (see equations). The elements of the remaining matrices and vectors are:

$$T_{ii} = \frac{\partial^2 F}{\partial t_i^2} \qquad C_{ji} = \frac{\partial^2 F}{\partial p_j \partial t_i} \qquad \gamma_i = \frac{\partial F}{\partial t_i} \tag{57}$$

(matrix $\boldsymbol{T}$ is a diagonal matrix). The complete matrix can be very large ($n+m$-by-$n+m$). As shown below the fact that the large n-by-n submatrix $\boldsymbol{T}$ is diagonal can be used to speed up the calculation.

Solution by partitioning. The general method of partitioned matrices is used for the solution. In the first step the parameters $\boldsymbol{p}$ are assumed to be fixed values. The corrections $\Delta \boldsymbol{t}$ can then be determined from the simpler system

$$\left(\ \boldsymbol{T} \ \right) \cdot \left(\ \Delta \boldsymbol{t} \ \right) = \left(\ \boldsymbol{\gamma} \ \right) \qquad \text{i.e.} \quad \Delta t_i = \gamma_i / T_{ii} \ . \tag{58}$$

After this (local) solution for the parameters $\boldsymbol{t}$ the matrices $\boldsymbol{T}$ and $\boldsymbol{C}$ are redetermined; then the vector $\boldsymbol{\gamma}$ is zero.

The second step is the solution for the corrections $\Delta \boldsymbol{p}$. The matrix $\boldsymbol{H}$ and the vector $\boldsymbol{g}$ are determined using the fitted parameters $\boldsymbol{t}$. This can be done using a modified matrix according to

$$\Delta \boldsymbol{p} = \boldsymbol{H}'^{-}1 \boldsymbol{g} \qquad \text{with} \quad \boldsymbol{H}' = \boldsymbol{H} - \boldsymbol{C}^T \boldsymbol{T} \boldsymbol{C} \tag{59}$$

where the modified Hessian $\boldsymbol{H}'$ (a smaller m-by-m matrix) correctly takes into account the correlation between the $\boldsymbol{p}$ and the $\boldsymbol{t}$, expressed by the derivative matrix $\boldsymbol{C}$. The solution for $\Delta\boldsymbol{p}$ is identical to the complete solution of matrix equation. The inverse matrix $\boldsymbol{H}'^{-1}$ is the covariance matrix of vector $\boldsymbol{p}$. The final values of the parameters $\boldsymbol{t}$ remain undetermined at this iteration; they may differ slightly from the values determined in the first step, which neglected the correlation between the $\boldsymbol{p}$ and the $\boldsymbol{t}$.

The solution by partitioned matrix is particular suited for the solution of equation, because the large matrix is a diagonal matrix. Thus the elements t_i of the parameter vector $\boldsymbol{t}$ can be determined element by element. The elements t_i correspond, according to equation (54), to the zero of the nonlinear function

$$g(t_i) = (1 - t_i) \sum_{j=1}^{m} \frac{a_{ij} p_j}{1 - p_j t_i} - d_i \tag{60}$$

which can be determined by Newtons method, using the derivative

$$\frac{\mathrm{d}g(t_i)}{\mathrm{d}t_i} = - \sum_{j=1}^{m} \frac{a_{ij} p_j \left(1 + p_j\right)}{\left(1 - p_j t_i\right)^2} \tag{61}$$

Starting from the value $t_i = 0$ the zero of the function is usually found in few iterations.

The maximum likelihood method allows to estimate the covariance matrix of the fitted parameters from the inverse of the Hessian $\boldsymbol{H}'$. These estimates can be checked by performing large numbers of Monte Carlo experiments, which allow to determine distributions of the estimated parameters.
From equation (59) it is clear that at least the diagonal elements of the matrix $\boldsymbol{H}'$ are *smaller* compared to $\boldsymbol{H}$, and thus the estimated standard deviations of Barlows method are *larger* than the Poisson method. If the likelihood function of Barlows method is more realistic, the likelihood function of the Poisson method is too narrow because the fluctuations of the Monte Carlo data are neglected. Thus one expects that the "propagated errors" are larger for Barlows method compared to the Poisson method. However, as was shown already in the paper by Barlow, the Poisson method can have a biased result in certain cases.

In a loop over the n bins the parameters t_i are determined and, using these fitted parameters, the elements of the matrices $\boldsymbol{H}$, $\boldsymbol{C}$ and $\boldsymbol{T}$. The elements of the modified matrix $\boldsymbol{H}'$ are in this case:

$$H'_{jk} = H_{jk} - \sum_{i=1}^{n} \frac{C_{ji} C_{ki}}{T_{ii}} = \frac{\partial^2 F}{\partial p_j \partial p_k} - \sum_{i=1}^{n} \frac{\dfrac{\partial^2 F}{\partial p_k \partial t_i} \dfrac{\partial^2 F}{\partial p_j \partial t_i}}{\dfrac{\partial^2 F}{\partial t_i^2}} \tag{62}$$

Example: The numerical example is identical to example C used by R. Barlow in his paper. The set is two-dimensional with values x_1 and x_2 within the square $0 \leq x_1 \leq 1;\ 0 \leq x_2 \leq 1$. The densities of the first and the second source are given by

$$F_1(x_1, x_2) \propto \frac{x_1 + x_2}{2} \qquad\qquad F_2(x_1, x_2) \propto \frac{1 + x_1 - x_2}{2} \tag{63}$$

i.e. both are rising with x_1, while the first is rising with x_2, and the second is falling with x_2. For both distributions there is an edge of the square with zero density. "Data" events were generated in the ratio $1:2$.

In the test 1000 "data" and 1000 MC events for the two sources were generated, with 10×10 bins for x_1 and x_2; thus on average there are 10 entries per MC source per bin. The fitted

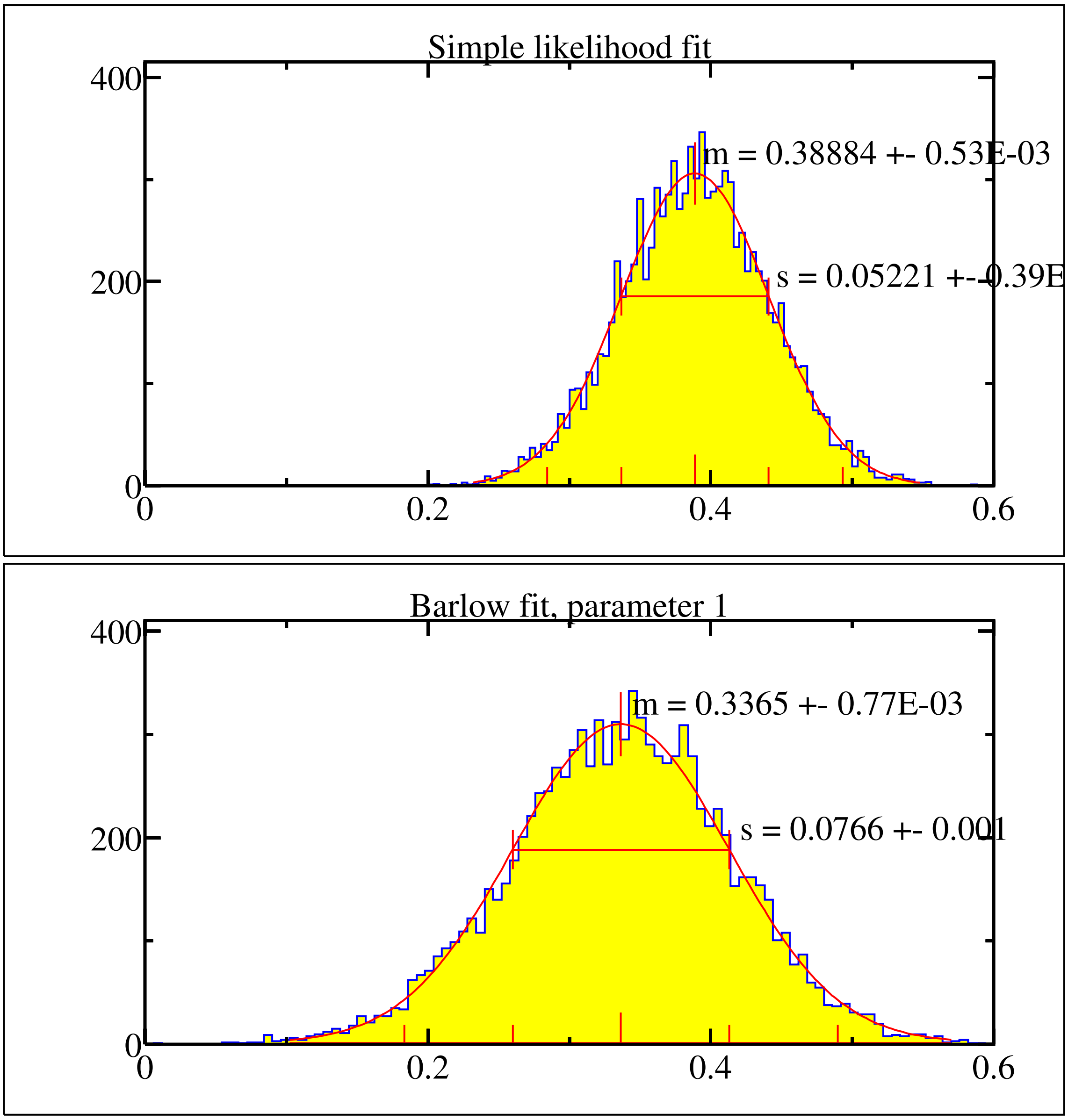

Figure 13: Distribution of the resulting value of the first parameter obtained by the simple likelihood method (top) and by Barlows method (bottom). The distribution is the result of 10 000 MC experiments with a true value of the first parameter of 0.3333. The result of the simple likelihood method is systematically biased.

values for the first parameter, which in the simulation has the value 0.3333, obtained from 10 000 experiments is shown in Figure 13. The result from the simple likelihood fit for the first parameter is 0.389 ± 0.052 (mean and standard deviation of distribution), while the result is 0.336 ± 0.077 for the Barlow fit. Ignoring the finite MC statistic produces a substantial bias in the simple method, which is avoided in the Barlow method. These result agree with the results from [17]. The distributions of the error of the first parameter in the simple method and in Barlows fit are shown in Figure 14. In both fits the calculated standard deviation is slightly smaller than the standard deviation calculated from the distribution of the fit result. Finally the value of the likelihood function from the fit is shown in Figure 15 for the two fits; this function value can be interpreted in terms of the goodness-of-fit χ^2 (see equation (39) and the discussion

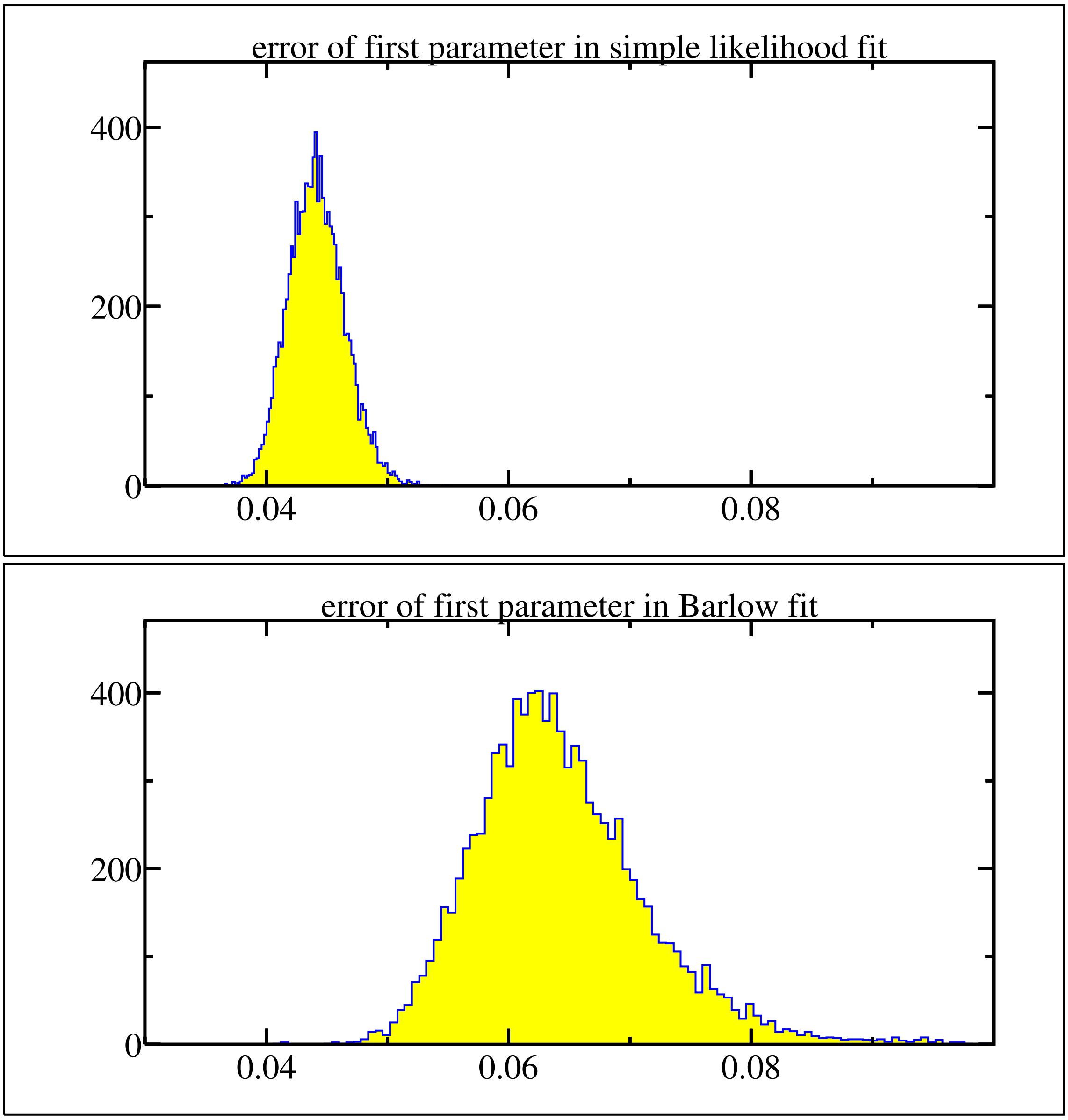

Figure 14: Distribution of the standard deviation for the MC simulation of Figure 13 by the simple likelihood method (top) and by Barlows method (bottom). The standard deviation is calculated in the standard way from the curvature of the likelihood function (inverse of Hessian). In both cases the standard deviation seems to be slighly underestimated.

there). The expected mean value is about 100 (for 100 bins) and the result from the Barlow fit agrees nicely with the expected value, while the simple likelihood fit has a larger mean value around 150. This also demonstrates the improvement by Barlows method.

References

[1] G. Cowan, Statistical Data Analysis, Clarendon Press, Oxford (1998)

[2] B. W. Rust and W. R. Burrus, Mathematical Programming and the Numerical Solution of Linear Equations, American Elsevier Publishing Company, Inc., New York, 1972

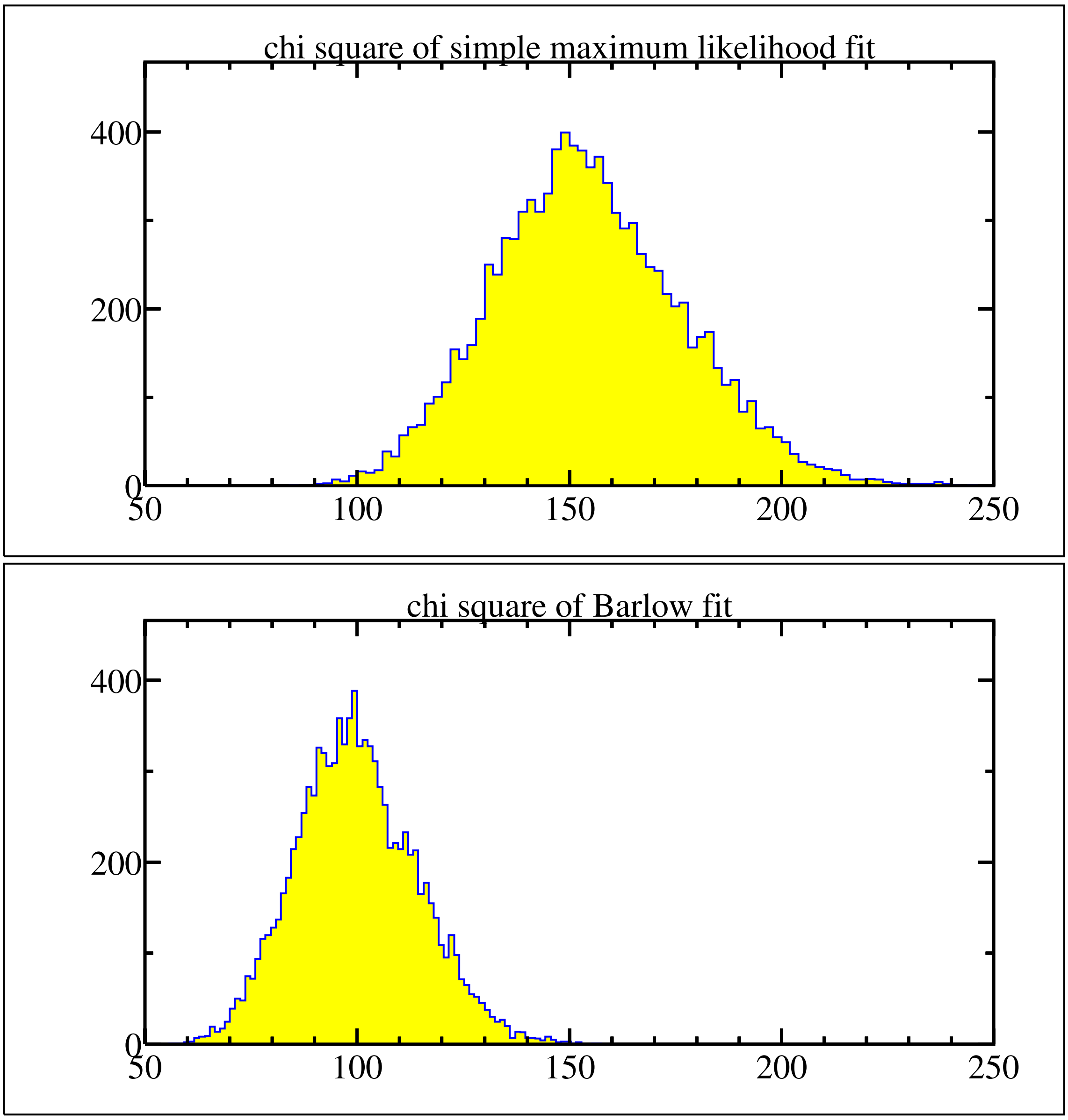

Figure 15: The goodness of the fit is shown by the "χ^2"-like expression explained in the text for the simple likelihood method (top) and for Barlows method (bottom). The number of degrees of freedom is $100 - 2 = 98$. This is the value obtained on average in the Barlow fit, while the value of the simple likelihood method is shifted to much larger values.

[3] S. Christiansen, Integral equations: An outline, *Formulae and Methods in Experimental Data Evaluation with Special Emphasis on High Energy Physics* Vol. 3, European Physical Society, CERN, 1984

[4] J. Carr, Analysis of deep inelastic muon and electron scattering experiments, *Formulae and Methods in Experimental Data Evaluation with Special Emphasis on High Energy Physics* Vol. 2, European Physical Society, CERN, 1984

[5] D. L. Phillips , A technique for the numerical solution of certain integral equations of the first kind, J. Assoc. Comput. Mach. **9**, 84-97, 1962

[6] A.N. Thikhonov, On the solution of improperly posed problems and the method of regularization, *Sov. Math.* **5**, 1035 (1963)

[7] A.N. Thikhonov and V.Ya. Arsenin, Solutions of Ill-posed Problems, John Wiley, New York (1977)

[8] S. W. Provencher, A constrained regularization method for inverting data represented by a linear algebraic or integral equation, *Comp. Phys. Comm.*, **27**, 213-227 (1982)

[9] V. Blobel, Unfolding methods in high energy physics experiments, in *Proceedings of the 1984 CERN School of Computing*, CERN 85-09 (1985) and DESY 84-114

[10] V. Blobel, The $\mathcal{RUN}$ manual, Regularized Unfolding for High-Energy Physics Experiments, OPAL Technical Note TN361 (1996)

[11] M. Jonker et al. (CHARM Collaboration), Experimental study of differential cross sections $d\sigma/dy$ in neutral current neutrino and antineutrino interactions Physics Letters **102 B**, 62-72, 1981

[15] M. Jonker et al. (CHARM Collaboration), Experimental study of x-distributions in semileptonic neutral current neutrino interactions, Physics Letters **128 B**, 117-123, 1983

[12] Ch. Berger et al. (PLUTO Collaboration), Measurement of the photon structure function $F_2^\gamma(x,Q^2)$, Physics Letters **142 B**, 111-118, 1984

[13] Ch. Berger et al. (PLUTO Collaboration), Measurement of deep inelastic electron scattering off virtual photons, Physics Letters **142 B**, 119-124, 1984

[14] A.K. Louis, Inverse und schlecht gestellte Probleme, Teubner, Stuttgart und Leipzig (1989)

[15] V. Blobel and E. Lohrmann, Statistische und numerische Methoden der Datenanalyse, Teubner, Stuttgart und Leipzig (1998)

[16] C. de Boor, A Practical Guide to Splines, Springer, New York, Heidelberg, Berlin, 1978

[17] Roger Barlow and Christine Beeston, Fitting finite Monte Carlo samples, *Computer Physics Communications* **77** (1993) 219–228