AN UNFOLDING METHOD FOR HIGH ENERGY PHYSICS EXPERIMENTS

Volker Blobel
Institut für Experimentalphysik, Universität Hamburg, Germany

Abstract

Finite detector resolution and limited acceptance require one 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 statistics of the Monte Carlo events is taken into account by the use of Barlow's method with a new method of solution. A clustering method is used before combining 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. The acceptance depends on the kinematical variable x.
- **Transformation:** Instead of the quantity x a different, but related quantity y is measured. The transformation from x to y can 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,

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

called a Fredholm integral equation of the first kind. In practice often a known (measured or simulated) background contribution $b(y)$ has to be added to the right-hand side of equation (1); this contribution is ignored in this paper. The resolution function $A(y,x)$ represents the effect of the detector. For a given value $x = x_0$ the function $A(y,x_0)$ describes the response of the detector in the variable y for that fixed value x_0. The problem in determining the distribution $f(x)$ from measured distributions $g(y)$ 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. 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 a statistical nature.

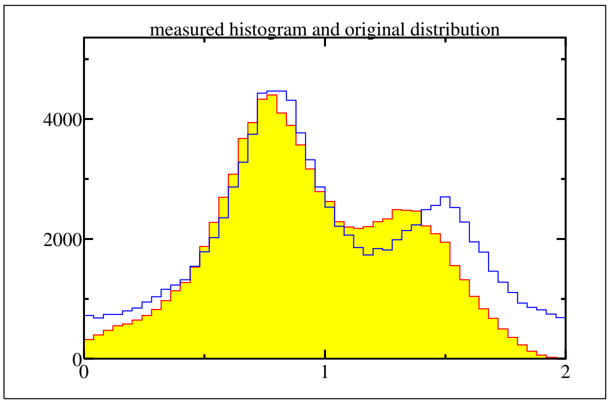

Fig. 1: The Monte Carlo simulation of the effects of limited acceptance, transformation and finite resolution. Shown is the original (true) distribution (line histogram) and the "measured" distribution (yellow/shaded histogram).

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.

For the numerical solution of equation (1) the distributions have to be 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

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

which has to be solved for the vector $\boldsymbol{x}$, given the vector $\boldsymbol{y}$ (data histogram). The vector $\boldsymbol{y}$ with n elements represents a histogram of the measured quantity y, and the distribution $f(x)$ is represented by a histogram of the vector $\boldsymbol{x}$ with m elements. The variables y and x may be multidimensional, and the multidimensional histograms can be mapped to n-bin ($\boldsymbol{x}$) and m-bin histograms ($\boldsymbol{y}$), respectively. The transition from $\boldsymbol{x}$ to $\boldsymbol{y}$ is described by the n-by-m matrix $\boldsymbol{A}$. The element a_{ij} is related to the probability to 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 $\boldsymbol{x}$. Problems with standard solutions are discussed in the next section.

In high energy physics experiments the problem 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 according to the Poisson distribution around the expected value) has a large statistical fluctuation. Furthermore the resolution function $\boldsymbol{A}(x, y)$ (or the matrix $\boldsymbol{A}$) is not known analytically, but is represented by a data set from Monte Carlo simulation of the process, based on some assumed distribution $f_{\text{MC}}(x)$,

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

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

A simple method like the so-called *bin-by-bin correction* may be meaningful if the measurements y are very close to the true values x. Real *unfolding* methods, taking all the correlations into account, are essential if there are larger effects of *transformation* and *finite resolution*. A solution $\boldsymbol{x}$ has to be found, with small deviations between the elements of $\boldsymbol{Ax}$ and the elements of the actually measured histogram $\widehat{\boldsymbol{y}}$. 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 $\boldsymbol{x}$ connected to large (negative) correlations between adjacent bins are not acceptable. The 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 [2] and by Thikhonov [3]. For a function $f(x)$ the integrated square of the second derivative

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

is often used in the regularization which in the linearized version of the problem can be expressed by a quadratic form $C(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}$ with a positive-semidefinite matrix $\boldsymbol{C}$ (derivatives are replaced by finite differences). Equation (4) is then modified to the form

$$F(\boldsymbol{x}) = -\log L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}) + \tau \cdot C(\boldsymbol{x}) \tag{6}$$

with a factor τ called regularization parameter.

The result of the minimization of the modified function $F(\boldsymbol{x})$ of equation (6) will show smaller fluctuations than the result obtained from equation (4) and may be more useful 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 value of regularization parameter τ. A very large value would result in a *linear* function $f(x)$ or distribution $\boldsymbol{x}$, respectively. It is clear that the method requires an a-priori knowledge about a smooth behaviour of $f(x)$. The function $f_{\text{MC}}(x)$ 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 expressing $f(x)$ in the form $f(x) = f_{\text{MC}}(x) \times f_{\text{mult}}(x)$ and rewriting equation (1) in the form

$$g(y) = \int \left[A(y, x) f_{\text{MC}}(x) \right] f_{\text{mult}}(x) \, \mathrm{d}x \ . \tag{7}$$

For the discretized form the function $f_{\text{MC}}(x)$ can be absorbed in a redefinition of matrix $\boldsymbol{A}$ and the vector $\boldsymbol{x}$ is interpreted as discretization of the hopefully *smooth* function $f_{\text{mult}}(x)$. With this redefinition the equation (2) can remain unchanged. The program RUN [4, 5] for regularized unfolding is available for almost two decades and has been used in many experiments; early applications are [6] and [7]. It is based on the reinterpretation of matrix $\boldsymbol{A}$ 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 some 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 $\boldsymbol{y} = \boldsymbol{Ax}$ describes the distribution expected due to the migration effect. With 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

$$\boldsymbol{A} = \begin{pmatrix} 1-\varepsilon & \varepsilon & 0 & 0 & 0 \\ \varepsilon & 1-2\varepsilon & \varepsilon & 0 & 0 \\ 0 & \varepsilon & 1-2\varepsilon & \varepsilon & 0 \\ 0 & 0 & \varepsilon & 1-2\varepsilon & \varepsilon \\ 0 & 0 & 0 & \varepsilon & 1-\varepsilon \end{pmatrix} . \tag{8}$$

A direct solution for $\boldsymbol{x}$, given a measurement $\widehat{\boldsymbol{y}}$ differing from the expectation $\boldsymbol{A}\boldsymbol{x}$ with the true vector $\boldsymbol{x}$ by statistical fluctuations, is possible with inversion of the matrix $\boldsymbol{A}$:

$$\text{estimate} \quad \widehat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\widehat{\boldsymbol{y}} \qquad\qquad \text{error propagation} \quad \boldsymbol{V}(\widehat{\boldsymbol{x}}) = \boldsymbol{A}^{-1}\boldsymbol{V}_y\left(\boldsymbol{A}^{-1}\right)^T .$$

The result has certain good statistical properties, for example it has no bias: $E[\widehat{\boldsymbol{x}}] = \boldsymbol{A}^{-1}E[\boldsymbol{y}] = \boldsymbol{A}^{-1}\boldsymbol{A}E[\boldsymbol{x}] = \boldsymbol{x}$. 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 $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ using an orthogonal decomposition is discussed; this will allow some insight into the unfolding problem.

The symmetric matrix $\boldsymbol{A}$ can be written as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \tag{9}$$

with a transformation matrix $\boldsymbol{U}$ with property $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{1}$, and a diagonal matrix $\boldsymbol{D}$, where the diagonal elements of matrix $\boldsymbol{D}$ are the eigenvalues λ_j of matrix $\boldsymbol{A}$ (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 $\kappa = \lambda_{\text{max}}/\lambda_{\text{min}}$; the value of κ is important for the quality of unfolding (see below). For values above $\varepsilon = 0.20$ the condition number κ is very rapidly increasing.

A transformation of equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ to a new basis is done by multiplication with matrix $\boldsymbol{U}^T$ (which is a rotation in the n-dimensional space):

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

The matrix $\boldsymbol{U}$ of eigenvectors $\boldsymbol{u}_j^T$ allows one to transform the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to vectors $\boldsymbol{b} = \boldsymbol{U}^T\boldsymbol{x}$ and $\boldsymbol{c} = \boldsymbol{U}^T\boldsymbol{y}$, and to transform these vectors back by $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{b}$ and $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{c}$. The transformed equation $\boldsymbol{c} = \boldsymbol{D}\boldsymbol{b}$ with the diagonal matrix $\boldsymbol{D}$ shows, that each of the coefficients b_j and c_j is transformed independently of any other coefficient by the simple relation $c_j = \lambda_j \cdot b_j$. This operation does not depend on any assumption of the solution $\boldsymbol{x}$, and depends only on the properties of the matrix $\boldsymbol{A}$. Folding ($\boldsymbol{x} \to \boldsymbol{y}$) and unfolding ($\boldsymbol{y} \to \boldsymbol{x}$) is multiplication and division by the eigenvalues λ_j, respectively, of the coefficients in the transformed space.

In order to unfold a measured vector $\boldsymbol{y}$, the vector is 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 to obtain $b_j = c_j/\lambda_j$; 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 by few of the coefficients b_j with small eigenvalues and large statistical errors, and the complete result is unsatisfactory.

Example. In a numerical example the matrix $\boldsymbol{A}$ has the form of equation (8) with $n = 20$ and a value of the migration parameter of $\varepsilon = 0.22$. The first eigenvalue is $\lambda_1 = 1.0$, and the last one is $\lambda_{20} = 1/7.9$, giving a condition number $\kappa = 7.9$. For $\boldsymbol{x}$ the ideal distribution of Figure 2a is assumed; the underlying function is of the form $x\exp(-ax)$. The decomposition of the matrix $\boldsymbol{A}$ according to equation (9) is performed and the coefficients b_j and c_j are calculated. These coefficients are shown in Figure 3a (with $b_j \geq c_j$). In addition this figure shows, calculated by standard error propagation, the almost constant error level of the coefficients, of the folded distribution of Figure 2a with Poisson distributed bin contents. Figure 3a shows that the coefficients b_j of the true distribution decrease rapidly with increasing value j of the index of the coefficient, by roughly three orders of magnitude. The coefficients c_j of the folded distribution drop even faster, because it is more smooth due to the migration effect. Of course the relation

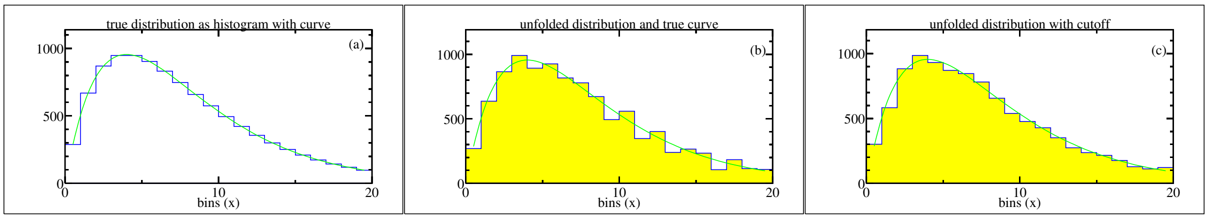

Fig. 2: Original (true) distribution (a) and two results from unfolding ((b) and (c)). Result (b) has been obtained from all 20 coefficients, and for result (c) a sharp cut-off after 10 coefficients has been applied (i.e. the coefficients 11 to 20 are ignored).

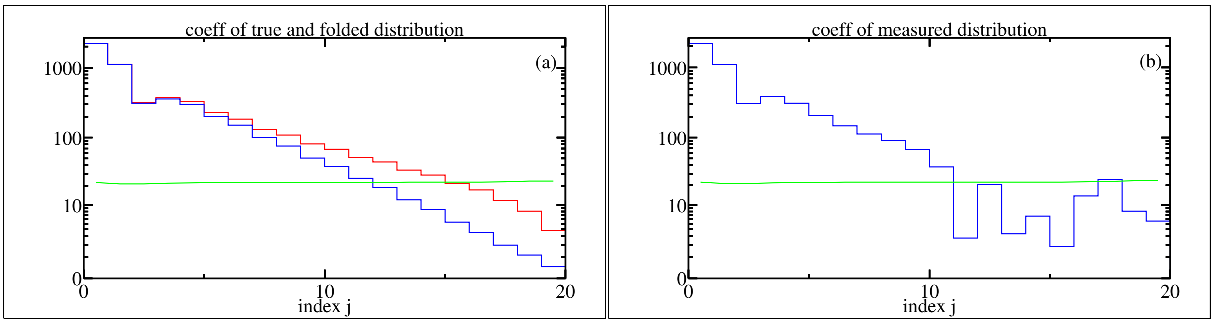

Fig. 3: The absolute values of coefficients b_j and c_j are shown for $j = 1, 2, \ldots 20$. The coefficients b_j and c_j for the true distribution and the folded distribution (without measurement errors) are shown in (a), together with the (almost constant) error estimate for the coefficients b_j calulated by error propagation. The coefficients c_j from the simulated measured distribution are shown in (b), together with the error estimate. For j above 12 the smaller coefficients of the folded distribution become smaller than the statistical error. In (b) the coefficients for j above 12 are dominated by statistical errors and even the sign is not determined by the data.

$b_j/c_j = \lambda_j$ is valid. The last coefficient b_j in Figure 3a is reduced to c_j by the inverse of the condition number of the matrix, which is $\kappa = 7.9$ in this case.

The components of the first eigenvector u_1 (eigenvalue = 1) are all the same. Thus the coefficients b_i and c_i are identical, and proportional to the total sum of the measured distribution, not at all influenced by the migration. If visualized by functions, interpolating the components the eigenvector u_j (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; it has a small absolute value and its measured value will have a large relative statistical error. The value of b_{20} is obtained by $b_{20} = 7.9 \cdot c_{20}$ in unfolding, introducing a large bin-to-bin oscillation into the result of unfolding.

In a simulation, Poisson distributed bin contents are assumed in the measurement vector $\boldsymbol{y}$. The coefficients for this measured distribution are shown in Figure 3b, together with the level of the 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 information content of the measurement. For indices above $j = 12$ even the sign of the coefficient cannot be determined by the measurement.

Using all the "measured" coefficients for the unfolding, the result of Figure 2b is obtained. 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 their inverse eigenvalues. 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 $j = 12$ 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 2c; compared to Figure 2b the 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. 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 [4]. 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; 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 [4] 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).[1]

3 THE PROPOSED UNFOLDING METHOD

The method proposed here 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 large number of rows and columns 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.

Figures in this section 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 [4]. In total 100 000 “events” are simulated for “data” and for the MC defining matrix $\boldsymbol{A}$. The input function $f_{\text{MC}}(x)$ (equation (7)) is a constant.

In RUN the discretization for $f(x)$ and for $A(y,x)$ was done using cubic B-spline functions; the effect is the same as for simple histograms, namely, the integral equation is transformed to a system of linear equations. 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. However the accurate 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). The curvature of the solution is constructed by finite differences: the second derivative in bin j is proportional to $x_{j-1} - 2x_j + x_{j+1}$. In a histogram some resolution is lost if bins with a width as large as expected for the final resolution would be used. It is recommended to use initially $m = 2n_{\text{df}}$ 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

[1] Sometimes the iterative solution of unfolding problems expressed by the equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is proposed in the literature 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 unsatisfactory result as by direct solution will be obtained. However in these proposals only a small number of iterations is recommended. It can be shown that iterative methods can in fact include an implicit regularization [8]: 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 is of course some subjectivity in stopping “early” enough.

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* the matrix $\boldsymbol{A}$. The statistical error of the elements a_{ij} eventually cannot be neglected.

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 $y_i = \sum_{j=1}^{m} a_{ij}\, x_j$. For the expected number y_i, as given by this expression, the observed values $\widehat{y}_i$ follows the Poisson distribution. 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 $\boldsymbol{x}$, assuming the Poisson distribution:

$$F(\boldsymbol{x}) = -\ln \mathcal{L}(\boldsymbol{x}) = -\ln \left[\prod_{i=1}^{n} P_{y_i}(\widehat{y}_i) \right] = \sum_{i=1}^{n} (y_i - \widehat{y}_i \cdot \ln y_i) + \text{const.}\,, \tag{10}$$

where the constant term containing e.g $\widehat{y}_i!$ can be ommited. This expression (10) correctly accounts also for bins with a small number of histogram entries $\widehat{y}_i$.

An alternative would be to use the (linear) least squares method with singular value decomposition for the fit. 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 [9]. In this method there is for each source bin x_j some (unknown) expected number of events A_{ij}. For each element A_{ij} the corresponding number a_{ij} from the Monte Carlo sample is generated by a distribution which is taken to be Poisson too. 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. A large number of slack variables (one for each bin) is introduced and has to be treated in the optimzation. A new fast numerical solution method has been developed (see [1]).

Combining bins. The likelihood function is a sum over all bins. Combining almost empty bins does not introduce a systematic error. The total 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. The combination of almost empty bins is done with a cluster algorithm, taking into account the distance between bins in one, two or three dimensions.

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 (10)). The standard iterative method is based on the representation for the correction $\Delta\boldsymbol{x}$

$$F(\Delta\boldsymbol{x}) = \frac{1}{2}\Delta\boldsymbol{x}^T \boldsymbol{H} \Delta\boldsymbol{x} + \Delta\boldsymbol{x}^T \boldsymbol{g} + \ldots \tag{11}$$

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 the equation $\boldsymbol{H}\Delta\boldsymbol{x} + \boldsymbol{g} = 0$. 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 $\boldsymbol{H}$ by $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ 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 transformation back to the $\boldsymbol{x}$-space of bin contents using the transformation matrix $\boldsymbol{U}$.

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 $\boldsymbol{C}$. In principle the same procedure is used as in RUN; the mathematical details are given elsewhere [4]. Here a simple explanation is given on the standard mathematical operations[2] 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 (11). Exactly as in the first option the Hessian is diagonalized.

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \, , \qquad \boldsymbol{H}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T = \left[\boldsymbol{U}\boldsymbol{D}^{-1/2}\right]\left[\boldsymbol{D}^{-1/2}\boldsymbol{U}^T\right] \, . \tag{12}$$

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 (unit covariance 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 another diagonalization can be done of matrix C_U:

$$\tau \cdot \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} \;\rightarrow\; \tau \cdot \widehat{\boldsymbol{x}}^T\boldsymbol{C}_U\widehat{\boldsymbol{x}} = \tau \cdot \widehat{\boldsymbol{x}}^T\boldsymbol{U}_C\,\boldsymbol{S}\,\boldsymbol{U}_C^T\widehat{\boldsymbol{x}} \tag{13}$$

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 $\widetilde{\boldsymbol{x}}$

$$\widehat{\boldsymbol{x}} \;\rightarrow\; \widetilde{\boldsymbol{x}} = \boldsymbol{U}_C^T\,\widehat{\boldsymbol{x}} \, . \tag{14}$$

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. A last rotation is done to bring the axes into the order of increasing curvature.

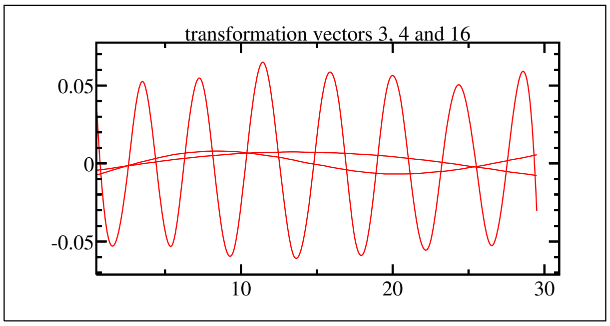

Fig. 4: Selected column vectors of the complete transformation matrix defined in the regularization procedure. They correspond to the curvature eigenvalues S_{33}, S_{44} and $S_{16,16}$. Visualization is done by curves interpolating the components. The amplitude associated which each vector all have the same standard deviation of 1.

Some columns of the complete (product) transformation are shown in Figure 4. 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. Sorted according to increasing value of S_{jj} the value of S_{jj} will increase rather fast with increasing index j. The spectrum of eigenvalues S_{jj} is shown in Figure 5. In terms of the linear combinations $\widetilde{\boldsymbol{x}}$ regularization is simply given by

$$\left(\widetilde{x}_j\right)_{\text{reg}} = \left(\frac{1}{1+\tau \cdot S_{jj}}\right)\left(\widetilde{x}_j\right)_{\text{unreg}} \, . \tag{15}$$

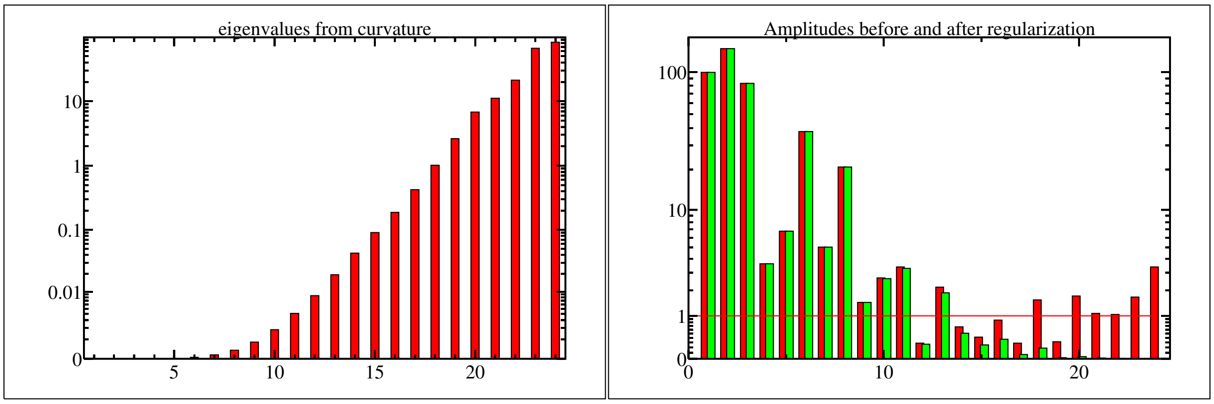

Fig. 5: The eigenvalues after the curvature transformation. The values are very rapidly increasing for orthogonal contributions for increasing index value (left). The amplitudes before (left bars) and after regularization (right bars). The statistical error of all amplitudes is equal to 1, which is indicated by the horizontal line. The vertical scale is linear at the bottom and makes a transition to a logarithmic scale at the top (right).

and this simple form is the reason for the transformations made before.

Determination of the regularization parameters τ. The first factors (small j) on the right-hand-side of equation (15) will be close to 1; for a value $\tau = 1/S_{kk}$ the 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 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

$$n_{\mathrm{df}} = \sum_{j=1}^{m} \left(\frac{1}{1 + \tau \cdot S_{jj}} \right) \quad . \tag{16}$$

Thus 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 or amplitudes, shown in Figure 5. 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. 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 4) are large. The regularization effectively damps the amplitude (right bars) around and above index 15, which has been chosen as the number of degrees 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 6. The left 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. 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 Figure 6 (right). The broader bins of this set of data points are a consequence of the limited acceptance and finite resolution of the measurement.

[2] 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.

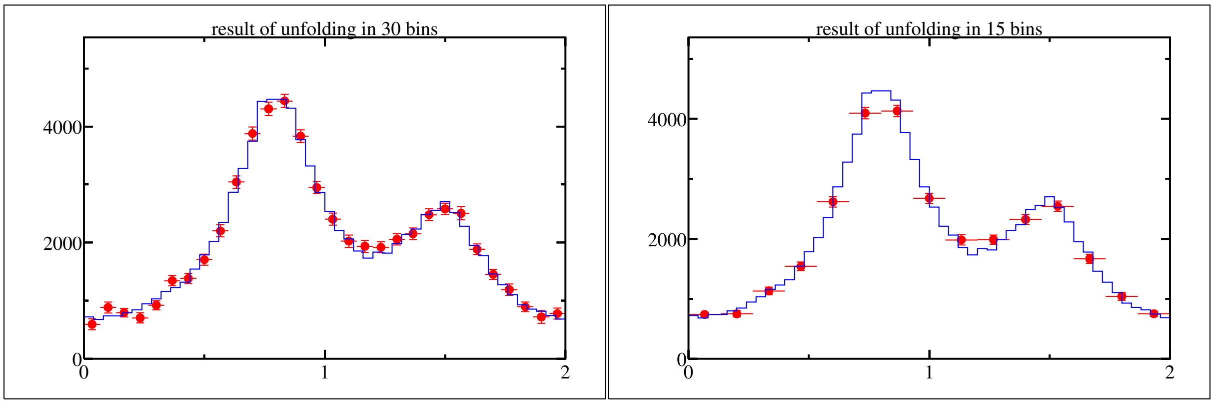

Fig. 6: The unfolding result after regularization with 15 degrees of freedom with 30 bins (left) and with 15 bins (right). For comparison the original (true) distribution is shown by a histogram. The data from Figure 1 are used as input.

ACKNOWLEDGEMENTS

I would like to thank the organizers of the conference on Advanced Statistical Techniques in Particle Physics for their hospitality and the stimulating atmosphere in Durham.

References

[1] A more detailed text is available via `http://www.desy.de/~blobel/`.

[2] 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)

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

[4] 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

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

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

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

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

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