O. Inverse problems – unfolding/deconvolution

V. Blobel

Inst. für Experimentalphysik, Hamburg

Sommersemester 2002

1. Fast Fourier transfrom

A quantity related to a physical process can be described either

- in the time domain, by a function $h(t)$ (with complex values), or else
- in the frequency domain, where the process is specified as its amplitude $H(f)$ as a function of frequency f, with $-\infty < f < +\infty$; the amplitude is a complex number (or amplitude and phase).

The two different representations $h(t)$ and $H(f)$ of the same function are related by the *Fourier transform* equations ($i = \sqrt{-1}$)

$$H(f) = \int_{-\infty}^{+\infty} h(t)\, e^{2\pi i f t}\, dt \qquad h(t) = \int_{-\infty}^{+\infty} H(f)\, e^{-2\pi i f t}\, df$$

Use of the frequency f instead of the angular frequency $\omega (= 2\pi f)$ reduces the number of factors of 2π.

Properties of transforms

Transform pairs are indicated by the symbol $\Longleftrightarrow$:

$$h(t) \Longleftrightarrow H(f)$$

Fourier transformation is a *linear* operation: the transform of the sum of two functions is equal to the sum of the transforms.

"Scaling" in time or frequency:

$$h(at) \Longleftrightarrow \frac{1}{|a|} H\left(\frac{f}{a}\right) \qquad \frac{1}{|b|} h\left(\frac{t}{b}\right) \Longleftrightarrow H(bf)$$

"Shifting" in time or frequency:

$$h(t-t_0) \Longleftrightarrow H(f)\, e^{2\pi i f t_0} \qquad h(t)\, e^{-2\pi i f t_0} \Longleftrightarrow H(f-f_0)$$

Total power in a signal

Parseval's theorem:

$$\text{total power} \equiv \int_{-\infty}^{+\infty} |h(t)|^2 \, \mathrm{d}t = \int_{-\infty}^{+\infty} |H(f)|^2 \, \mathrm{d}f$$

(the power in a signal is the same in the time and the frequency domain).

The one-sided power spectral density $P_n(f)$ gives the power in a frequency interval, regardless of the sign of f:

$$P_n(f) \equiv |H(f)|^2 + |H(-f)|^2 \quad 0 \le f < \infty$$

Convolution

Statistics: If the random variables t_1 and t_2 have probability density functions $g(t)$ and $h(t)$, then the probability density of the sum

$$t = t_1 + t_2$$

has to consider all possible pairs of values that add to give t. It is given by the *convolution* (german: Faltung) of the two functions $h(t)$ and $g(t)$:

$$(g * h)(t) \equiv \int_{-\infty}^{+\infty} g(\tau)\, h(t-\tau)\, \mathrm{d}\tau \qquad \text{with} \quad g * h = h * g$$

Theorem: The Fourier transform of the convolution is the product of the individual Fourier transforms:

$$g * h \Longleftrightarrow G(f)\, H(f)$$

Discretely sampled data

Values of function $h(t)$ are recorded at evenly spaced intervals in time with $\Delta =$ time interval:

Sequence of values $\qquad h_n = h(n\Delta) \quad n = \ldots, -2, -1, 0, 1, 2, \ldots$

Definition of a special frequency f_c for time interval Δ:

$$\text{Nyquist critical frequency} \qquad f_c \equiv \frac{1}{2\Delta}$$

(1) If a continuous function $h(t)$ is *bandwidth-limited* to frequencies smaller than f_c, then the function $h(t)$ is completely determined by its samples h_n.

(2) If $h(t)$ is **not** *bandwidth-limited*, then all the power spectral density that lies outside of the frequency range $-f_c < f < +f_c$ is spuriously moved into that range (*aliasing*).

The discrete Fourier transform

Integrals are replaced by sums:

$$H_n \equiv \sum_{k=0}^{N-1} h_k \, e^{2\pi i k n/N} \qquad\qquad h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n \, e^{-2\pi i k n/N}$$

Parseval's theorem (discrete form):

$$\text{total power} \equiv \sum_{k=0}^{N-1} |h_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |H_n|^2$$

Theorem: If the 2π-periodic function $h(t)$ has an absolute continuous r-th derivative $h^{(r)}(t)$, then

$$|H_k| = \mathcal{O}\left(\frac{1}{|k|^{r+1}} \right)$$

i.e. smooth functions are well described by few H_k.

How much computation ...

... is involved in computing the discrete Fourier transform of N points?

Standard method: N^2 complex multiplications, plus some additional operations, with

$$\mathcal{O}(N^2)$$

Answer since the mid-1960s: with *Fast Fourier Transform* FFT

$$\mathcal{O}(N \log_2 N)$$

by J.W.Cooley and J.W.Tukey (and before Danielsson and Lanczos (1942) and even Gauss (1805)).

This makes really a difference: for example

$$\text{computation for } N = 10^6 \quad \begin{cases} \text{2 weeks of cpu time} & \text{(standard)} \\ \text{30 seconds of cpu time} & \text{(FFT)} \end{cases}$$

The FFT – Fast Fourier Transform

Danielsson and Lanczos: a discrete Fourier transform of length N can be rewritten as the sum of two discrete Fourier transforms, each of length $N/2$. One of the two is from the even-numbered points of the original N, the other from the odd-numbered points. This method can be used recursively (N, $N/2$, $N/4, \ldots 1$) for $N = 2^k$.

Proof (with the complex constant $W \equiv e^{2\pi i/N}$) for $f_n \Longleftrightarrow F_n$:

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \\ &= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \\ &= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/-1} e^{2\pi ikj/(N/2)} f_{2j+1} \\ &= F_k^e + W^k F_k^o \end{aligned}$$

Case of a real function $h(t)$

If $h(t)$ is real, then $\qquad H(-f) = |H(f)|^*$

Discrete case: take the real array of f_n and treat it as a complex array h_j of half the length:

$$h_j = f_{2j} + i\, f_{2j+1}\,, \quad j = 0, \ldots N/2 - 1$$

Use the standard FFT for complex data, get complex F_n, and rearrange:

- the values F_0 and $F_{N/2}$ are both real; $F_{N/2}$ is returned as the imaginary part of F_0;
- the values F_1 to $F_{N/2-1}$ are complex and can be rewritten as real (positive) amplitudes and phases.

Convolution and deconvolution using the FFT

Often a signal $s(t)$ is smeared by a response function $r(t)$:

$s(t)$	signal, which goes indefinitely in time
$r(t)$	response function, typically a peaked function that falls to zero in both directions

The "smeared" signal is given by the convolution $r * s = s * r$.

Discrete convolution theorem: if the signal is periodic with period N (completely defined by the N values s_0, s_1, $\ldots s_{N-1}$), then

$$(r * s)_j \equiv \sum_{k=-N/2+1}^{N/2} s_{j-k} r_k \quad \Longleftrightarrow \quad S_n R_n$$

Convolution and … contnd.

What happens if the signal is of finite length, but not periodic?

Convolution will "pollute" first output channels $(r * s)_0$ with some wrapped-around data from the far end of the data stream: s_{N-1}, s_{N-2}, …

Treatment of end effects by zero padding:
add sufficient zero-padded values at the end of the s_j vector.

Deconvolution ...

...is the process of *undoing* the smearing in a data set thet has occurred under the influence of a known response function.

The response function describes the known effect of a less-than-perfect measuring apparatus.

Deconvolution is simple in the Fourier transformed representation:

$$m = s * r \qquad \Longleftrightarrow \qquad M_n = S_n \, R_n \qquad \rightarrow \qquad S_n = \frac{M_n}{R_n}$$

Note: the process of deconvolution is generally quite sensitive to noise in the input data $\rightarrow$ *optimal filtering*.

Optimal filtering with the FFT

Filtering = removal of “noise” from a “corrupted” signal.

$u(t)$	signal, uncorrupted (we want to measure this)
$r(t)$	response function
$s(t)$	smeared signal, from $u * r$
$n(t)$	noise component
$c(t)$	measured corrupted signal $= s(t) + n(t)$

Deconvolution is done by $C(f)/R(f)$.

Filtering is done by a filter, $\Phi(f)$, which, when applied to the measured signal $C(f)$, then deconvolved by $R(f)$, produces a signal $\hat{u}(t)$ or $\hat{U}(f)$ as close as possible to the uncorrupted signal:

$$\hat{U}(f) = \frac{C(f) \, \Phi(f)}{R(f)}$$

Optimal (Wiener) filter

Optimal (Wiener) filter is defined by minimizing

$$\int_{-\infty}^{+\infty} |\hat{u}(t) - u(t)|^2 \, dt$$

i.e. the result $\hat{u}(t)$ should be close to the truth $u(t)$ in the *least-square* sense.

The result is the optimal filter:

$$\Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2} \approx \frac{|S(f)|^2}{|C(f)|^2}$$

Often the noise is constant or almost constant in f and can be determined from the data (i.e. $C(f)$) in the high frequency region.

2. The unfolding problem

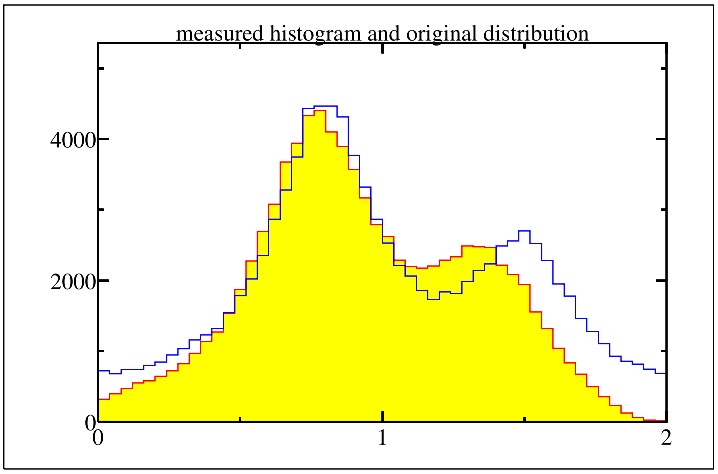
Measurement of a distribution $f(x)$ with a *real* detector in HEP:

- **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 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.

In *reality* a distribution $g(y)$ is measured instead of $f(x)$.

Example of the distorting effects

True distribution is given by blue (line) histogram, measured is the yellow histogram (from a simulation):



The task is to reconstruct the true distribution from the measured histogram (this simulation is later used as an example for unfolding).

Integral equation

Relation between $f(x)$ and $g(y)$

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

(Fredholm integral equation of the first kind)
or in discretized form:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \qquad\qquad y_i = \sum_{J=1}^{m} a_{ij} \, x_j + b_i$$

where

$f(x)$ **or** $\boldsymbol{x}$ distribution or histogram of true variable

$f(y)$ **or** $\boldsymbol{y}$ distribution or histogram of measured variable

$A(y,x)$ **or** $\boldsymbol{A}$ resolution function or matrix

$b(y)$ **or** $\boldsymbol{b}$ background distribution or histogram (known)

Note: $\boldsymbol{x}$ and/or $\boldsymbol{y}$ may be multidimensional.

3. Unfolding as an ill-posed problem

Discussion of a special case of folding/unfolding with resolution matrix

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

Numerical values in simulation: dimension $n = m = 20$ and $\varepsilon = 0.22$.

Solution of system of linear equations

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

in this special case is simple:

$$\widehat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{y} \qquad \boldsymbol{V}(\widehat{\boldsymbol{x}}) = \boldsymbol{A}^{-1}\boldsymbol{V}_y\left(\boldsymbol{A}^{-1}\right)^T$$

but result often wildly oscillating.

Orthogonal decomposition

Matrix $\boldsymbol{A}$ is decomposed into a diagonal matrix $\boldsymbol{D}$ and a rotation matrix $\boldsymbol{U}$:

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

Transform: $\boldsymbol{U}^T \cdot \;|$

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{x}\,. \\ \underbrace{\boldsymbol{U}^T\boldsymbol{y}}_{\boldsymbol{c}} &= \boldsymbol{D}\underbrace{\boldsymbol{U}^T\boldsymbol{x}}_{\boldsymbol{b}} \end{aligned}$$

Vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are transformed to vectors $\boldsymbol{b}$ and $\boldsymbol{c}$:

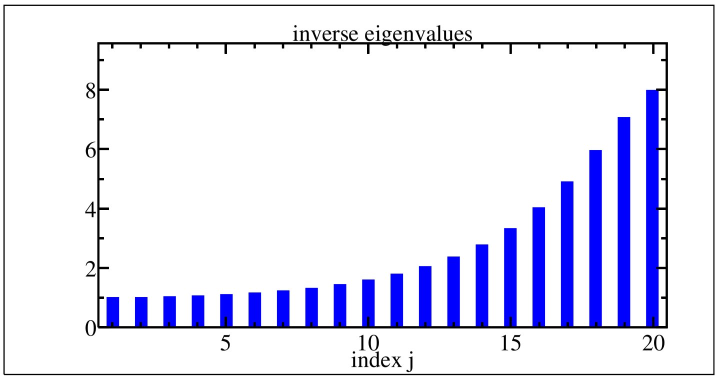
$$\begin{aligned} \boldsymbol{x} \to \boldsymbol{b} &= \boldsymbol{U}^T\boldsymbol{x} \qquad \text{or} \qquad b_j = \boldsymbol{u}_j^T\boldsymbol{x} \qquad j = 1\ldots n \\ \boldsymbol{y} \to \boldsymbol{c} &= \boldsymbol{U}^T\boldsymbol{y} \qquad \text{or} \qquad c_j = \boldsymbol{u}_j^T\boldsymbol{y}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

Unfolding and eigenvalues

Folding and unfolding is now done by multiplication/divison of the coefficients b_j and c_j by the eigenvalues:

Folding	$\boldsymbol{x} \to \boldsymbol{y}$	$b_j \to b_j \cdot \lambda_j = c_j$	$j = 1 \dots n$
Unfolding	$\boldsymbol{y} \to \boldsymbol{x}$	$c_j \to \dfrac{c_j}{\lambda_j} = b_j$	$j = 1 \dots n\,.$

... but look at the inverse of the eigenvalue for $n = m = 20$ and $\varepsilon = 0.22$:

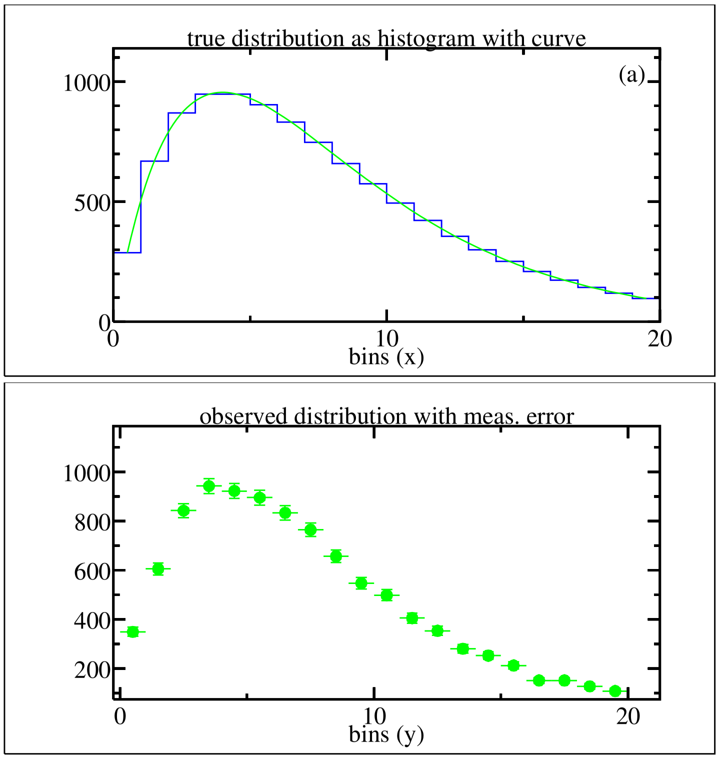


Simulation of a measurement

Starting from an ideal distribution (shape of $x\exp(-ax)$) a measured distribution is simulated:

top original true distribution

bottom simulated measured distribution

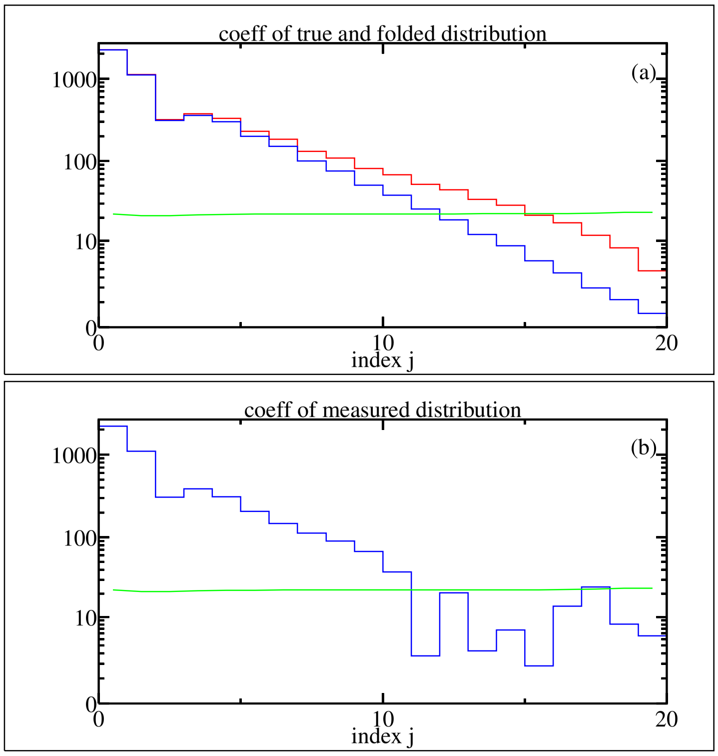


Coefficients

Transformation from distribution to coefficients:

top coefficients b_j and c_j of the true and the folded distribution (no measurement errors)

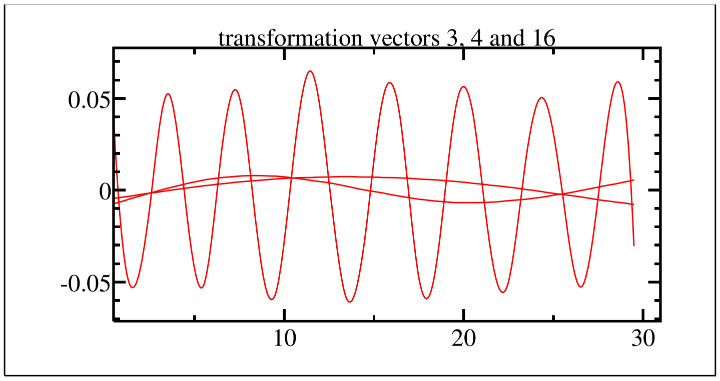
bottom coefficients c_j of the measured distribution (with measurement errors)



Green constant line indicates level of statistical errors.

Eigenvectors

The eigenvectors can be visualized by a function interpolating the bins.



Components of $\boldsymbol{u}_1$ with eigenvalue 1 all have the same value $\rightarrow$ coefficient c_1 is proportional to the sum of the distribution.

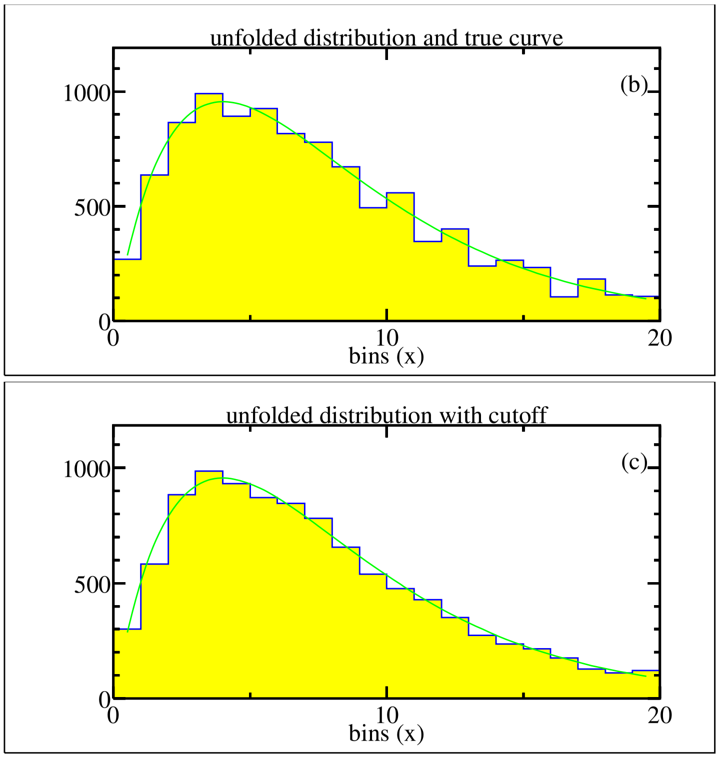
There are $j-1$ zeros in eigenvector $\boldsymbol{u}_j$ $\rightarrow$ more and more oscillating with increasing index j.

Result of unfolding

Unfolding without and with sharp cut in the number of contributions:

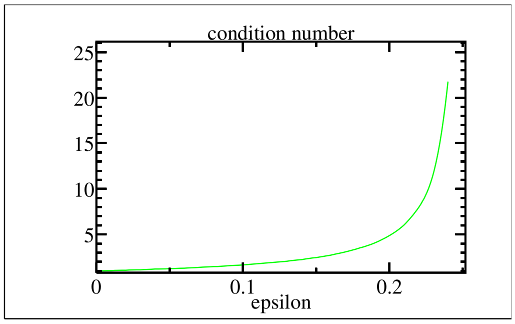
top unfolding with all 20 coefficients

bottom unfolding with a sharp cut-off after 10 coefficients



Condition number, iterative methods ...

The plot below shows the condition number as a function of the migration parameter ε:



Sharp increase above $\varepsilon = 0.2$:
$\rightarrow$ too many bins are used.

A remark to iterative methods without explicit regularization, starting from a "good" initial distribution and requiring only few iterations:

Iteration may contain an implicit regularization: convergence of first orthogonal contributions is fast, but convergence of last contributions may be very slow and are suppressed by stopping "early" enough.

4. ML fit of binned distributions

Expectation value for bin content y_i, $i = 1, 2, \ldots n$:

$$y_i = \sum_{J=1}^{m} a_{ij}\, x_j \qquad a_{ij} \text{ are known (without statistical errors)}$$

Distribution of expected number Poisson distribution $P_y(\widehat{y}) = e^{-y}\, \frac{y^{\widehat{y}}}{\widehat{y}!}$ is used to construct (negative log of) likelihood function:

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

which has to be minimized – or better ($F_{\text{min}} \sim \chi^2$)

$$F(\boldsymbol{x}) = \sum_{i=1}^{n} g_i \qquad \text{with} \quad g_i = \begin{cases} (y_i - \widehat{y}_i) - \widehat{y}_i \cdot \ln(y_i/\widehat{y}_i) & \text{if } \widehat{y}_i > 0 \\ f_i & \text{if } \widehat{y}_i = 0 \end{cases}$$

Fitting using finite Monte Carlo samples

… but the a_{ij} have itself statistical errors, if determined by Monte Carlo simulation.

Method of R.Barlow and Chr.Beeston (Comp. Phys. Comm. 77 (1993)): redefine expectation

$$y_i = \sum_{j=1}^{m} A_{ij} x_j \quad \text{with (unknown) expected number of events } A_{ij}$$

From each A_{ij} the corresponding a_{ij} is generated by a distribution taken to be Poisson, and included in the definition of the likelihood function.
This introduces a large number $n \times m$ of parameters, which however can be reduced to n parameters t_i, $\; i = 1, 2, \ldots n$ (see paper above).

Solution

There is still a large number $(n+m)$ of parameters to be determined in the fit:

$$\boldsymbol{x}: \quad m \text{ parameters} \qquad\qquad \boldsymbol{t}: \quad n \text{ parameters}$$

(time-consuming calculation).

Special fast solution: the $n+m$-by-$n+m$ Hessian has a *diagonal* submatrix of the n-by-n derivatives w.r.t. the t_i and can effectively be reduced by partitioning to a small m-by-m matrix.

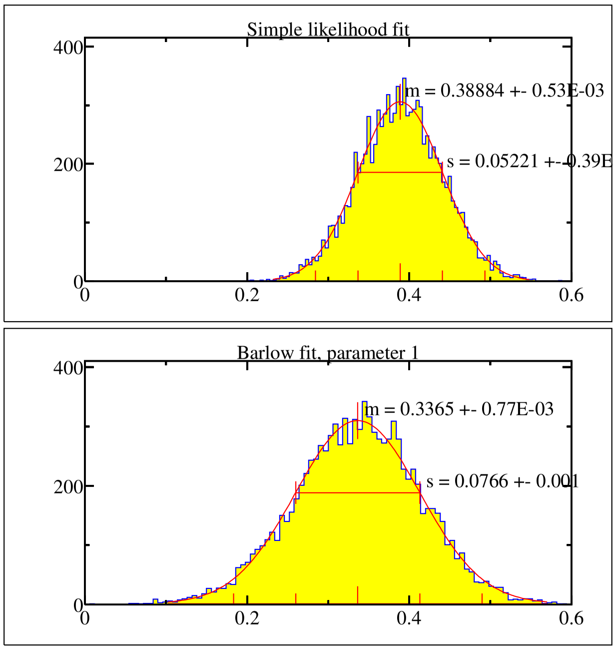
Test with example C of Barlow paper on the next two slides:
100 bins with 1000 entries for "data" and MC with two parameters.

1. parameter $= 1/3$ 2. parameter $= 2/3$

Fit of binned distributions: result on first parameter

Value of first parameter in simulation is 1/3. Shown is the result of 10 000 "experiments".

top Simple (Poisson) likelihood fit: result biased
bottom Method of Barlow: result unbiased

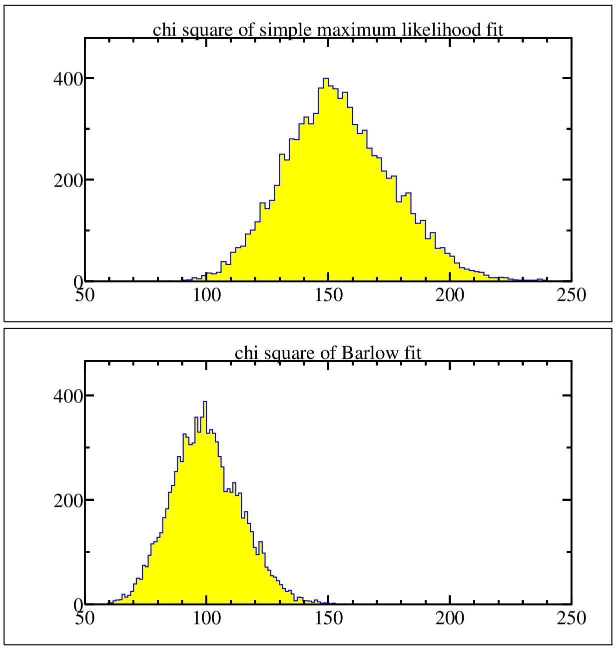


"χ^2" *of the two fits methods*

Goodness of fit can be checked by (modified) value of (negative log) likelihood function. Expected value is $100 - 2 = 98$.

top Simple (Poisson) likelihood fit: large "χ^2", because fluctuation of MC simulation partly neglected.

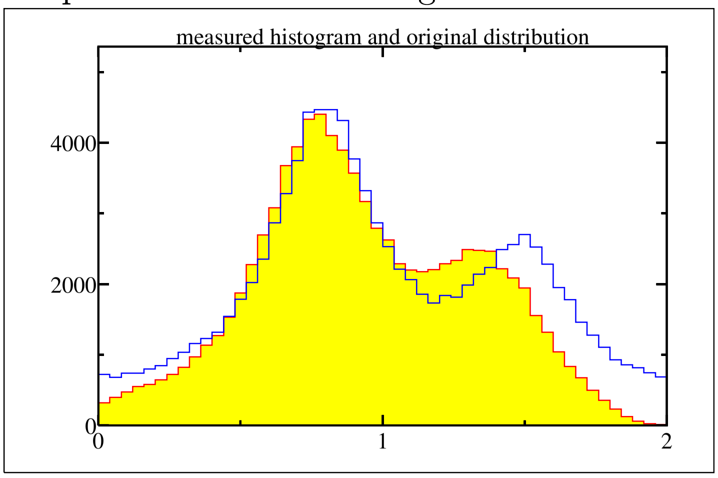
bottom Method of Barlow: value as expected.



5. A general unfolding method

Task: reconstruct the true distribution from the measured histogram

Example – result of unfolding will be shown later



True distribution is given by blue (line) histogram, measured is the yellow histogram (from a simulation)

Mathematics

Proposed method similar to program RUN, in use since ≈ 20 years (see Proc. of 1984 CERN School of Computing), but hopefully more stable.

RUN: cubic B-splines (continuous, smooth), curvature integral can be expressed by quadratic form; requires accurate determination of elements of matrix $\boldsymbol{A}$ (statistical errors are ignored in RUN).

Proposed method: simple histogram bins (discontinuous, but many bins), taking into account statistical Monte Carlo errors by Barlow method.
Either sharp cut-off or regularization with integral of square of second derivatives.
Second derivative of bin j is proportional to $-x_{j-1} + 2x_j - x_{j+1} \quad \rightarrow$ integral is quadratic form $R(\boldsymbol{x}) = \boldsymbol{x}^T C \boldsymbol{x}$, i.e. minimize

$$\begin{aligned} F(\boldsymbol{x}) &= -\ln \mathcal{L}(\boldsymbol{x}) + \tau \cdot \boldsymbol{x}^T C \boldsymbol{x} \\ F(\boldsymbol{x} + \Delta \boldsymbol{x}) &= \Delta \boldsymbol{x}^T \boldsymbol{H} \Delta \boldsymbol{x} + \Delta \boldsymbol{x}^T \boldsymbol{g} + \ldots + \tau \cdot R(\boldsymbol{x}) \end{aligned}$$

Mathematics, contnd.

Initial solution without regularization starts with diagonalization of the Hessian:

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

followed by transformation $\qquad \widehat{\boldsymbol{x}} = \left[\boldsymbol{D}^{-1/2}\boldsymbol{U}^T\right]\boldsymbol{x}$

Property of $\widehat{\boldsymbol{x}}$: components are orthogonal with variance 1:
covariance matrix in m-dimensional space is a sphere.
(this would already allow an acceptable solution by a sharp cut-off).

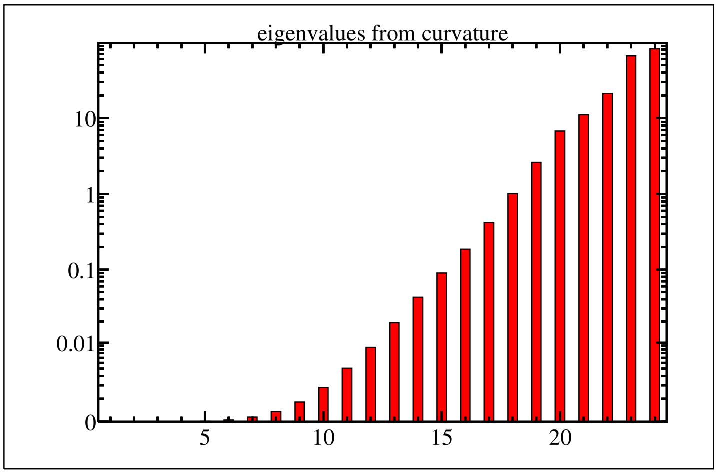
Every additional pure rotation will not change the property of $\widehat{\boldsymbol{x}}$:

Now, for regularization, an additional pure rotation is determined by diagonalization of the (transformed) curvature matrix $\boldsymbol{C}_U$.

Curvature eigenvalues S_{jj}

Linear combinations $\widehat{\boldsymbol{x}}$ are orthogonal with all variances equal to 1 (unit covariance matrix).
Further diagonalization based on regularization term: $\rightarrow S_{jj}$.

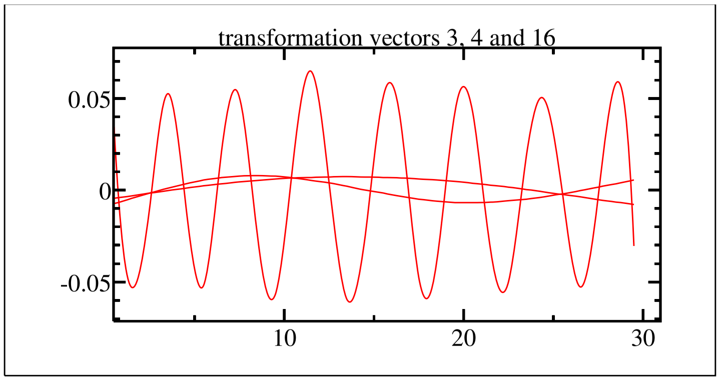
$$\tau \cdot \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x} \rightarrow \tau \cdot \widehat{\boldsymbol{x}}^T \boldsymbol{C}_U \widehat{\boldsymbol{x}} = \tau \cdot \left[\widehat{\boldsymbol{x}}^T \boldsymbol{U}_C \right] \boldsymbol{S} \left[\boldsymbol{U}_C^T \widehat{\boldsymbol{x}} \right]$$



... and another linear combination: $\widehat{\boldsymbol{x}} \rightarrow \widetilde{\boldsymbol{x}} = \boldsymbol{U}_C^T \widehat{\boldsymbol{x}}$

Transformation vectors

Visualization of vectors (= columns of the complete transformation matrix) corresponding to S_{33}, S_{44} and $S_{16,16}$ by interpolating curves.

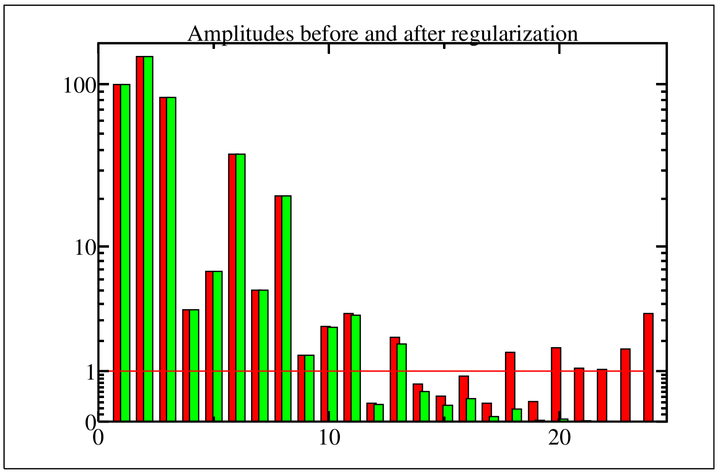


The amplitudes associated with each of the vectors (linear combinations of result bins) all have $\sigma = 1$.

Amplitudes and regularization

Regularization in terms of linear combinations $\widetilde{\boldsymbol{x}}$ is simply

$$(\widetilde{x}_j)_{\text{reg}} = \frac{1}{1+\tau \cdot S_{jj}} (\widetilde{x}_j)_{\text{unreg}} \qquad \text{with } \tau \text{ defined by} \quad n_{\text{df}} = \sum_{j=1}^{m} \frac{1}{1+\tau \cdot S_j j}$$

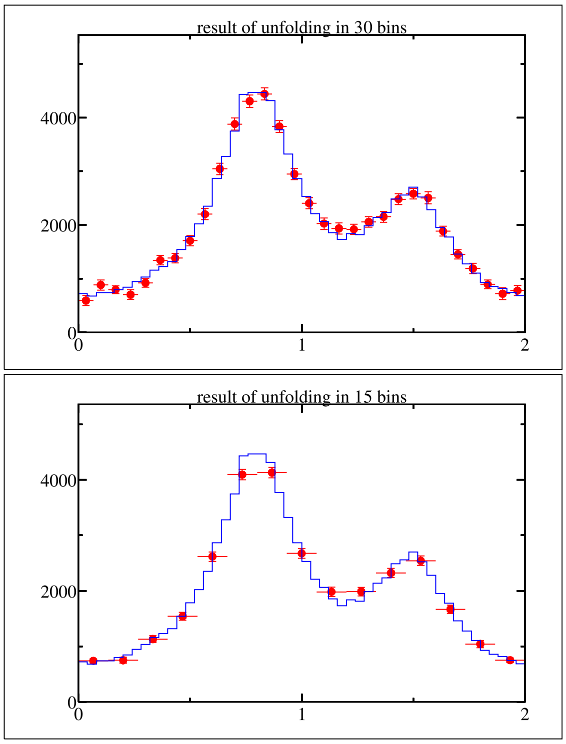


Amplitudes before (left red bars) and after (right green bars) regularization.

Result of unfolding with regularization

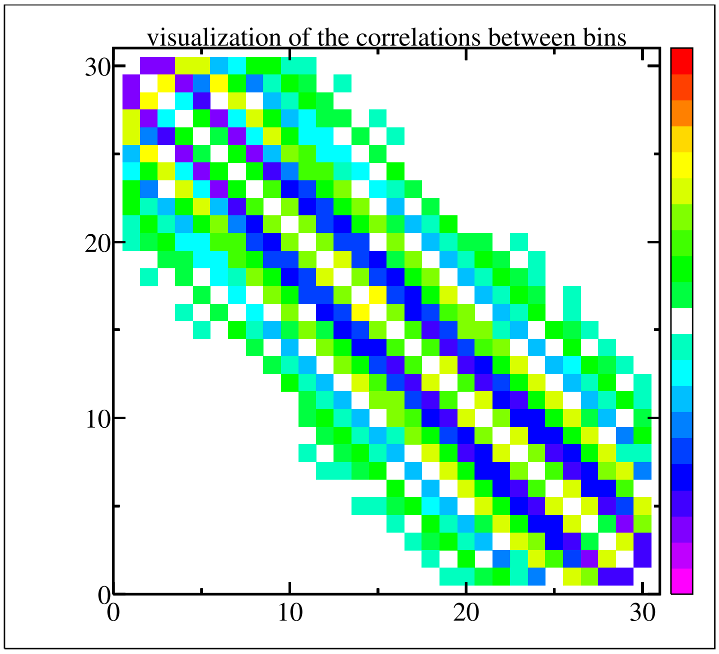
Regularization parameter τ determined from $n_{\text{df}} = 15$ (effective number of degrees of freedom.
Rank of covariance matrix is about 15.

top result with 30 bins
bottom result reduced to 15 bins
together with original distribution.



How big are the bin-to-bin correlations?

Visualization of correlation coefficients ρ between bins contents from the covariance matrix.



Magnitude of ρ expressed by colour according to the colour bar on the right:
top (red) $= +1$
bottom (magenta) $=-1$.

Summary

- Unfolding should produce detector-independent results from "distorted" measured distributions
- …and allow a sensitive comparison with theoretical predictions
- large bin-by-bin correlations in oscillating results would make the comparison impossible
- regularization can give acceptable results with correlations close to zero and with a small bias, but a reduced number of bins.

Of course a better detector is always better then a better unfolding.