Regularised Unfolding – the RUN program

V. Blobel

II. Inst. für Experimentalphysik, Hamburg

2000-01-19

LEP wide WG on 2-Photon Physics

RUN is a general unfolding program. Early applications were:

- dσ/dy in NC ν interactions (CHARM collaboration),
- $\sigma_{\gamma\gamma}$ and F_2^γ in 2-photon physics (PLUTO collaboration).

Web home page: `http://www.desy.de/~blobel/`

I. The unfolding problem

Measurements in High Energy Physics … … are distorted by detector effects:

- finite detector resolution: *smearing* of measured quantities
- non-linear response
- limited acceptance (trigger, cuts): loss of events, loss of particles

… and by physical effects:

- radiation of photons (ISR, FSR)
- Fermi motion within the nucleus
- fragmentation (partons → hadrons)

HEP: Measured data have relatively large statistical errors. Response function known only by set of Monte Carlo events (usually limited in statistics).

Possibilities for data analysis:

- either compared folded model with uncorrected data,
- or correct/transform the data distribution.

The transformation is called *Unfolding*.

Basic equations

Integral equation simplified as matrix equation:

$$g(y) = \int A(y,x) f(x) \, \mathrm{d}x \qquad\qquad \boldsymbol{g} = A \, \boldsymbol{f} \, .$$

Representation of:

- true distribution $f(x)$ by histogram $\boldsymbol{f}$ with n bins
- measured distribution $g(y)$ by histogram $\boldsymbol{g}$ with m bins
- response function $A(y,x)$ by m-by-n matrix $\boldsymbol{A}$

A single element of vector $\boldsymbol{g}$:

$$g_i = \sum_{j=1}^{n} A_{ij} f_j$$

Interpretation of response matrix:

- A_{ij} = probability, to measure an event from true bin f_j in bin g_i,
- column $A_{\bullet j}$ is detector response to true element f_j,
- all elements $A_{ij} \geq 0$ by definition (probabilities).

Unfolding by inversion

Standard statistical method: inversion $\boldsymbol{f} = \boldsymbol{A}^{-1}\boldsymbol{g}$ (or ML/LSQ fit)
... has optimal statistical properties:

- unbiased, and
- efficient (smallest possible variance for an estimator with zero bias)

Correlations between bins are unavoidable *negative* and *large*; the result looks unacceptable. The problem are the statistical fluctuations in $\boldsymbol{g}$, which generate oscillating terms in $\boldsymbol{f}$.

Any reduction in variance can only be achieved by introducing a bias.

Art of Unfolding: construct biased estimator such that the bias will be small, if the assumptions (e.g. smoothness of solution) are in fact correct.

[Glen Cowan; Statistical Data Analysis. Oxford Science Publications, Clarendon, Oxford. 1998]

Is there a general, satisfactory, acceptable, reliable method for Unfolding?

"A new method has been invented … "

"A new unfolding method has been developed" based on …
Bin-by-Bin Correction, Unsmearing, Regularization, a Migration Matrix Technique, Bayes' Theorem, Wavelet Method, Degree of Belief, Singular Value Decomposition, Intuition, Maximum Entropy, Feeling, …

Unfolding means: reduced number of data points, enlarged statistical errors, enlarged systematic errors.

I prefer methods, that

- start from principle (e.g. Maximum Likelihood, Least Squares),
- derive estimator from principle (method),
- derive statistical properties of estimator.

Convergence is well defined, if the method is iterative.

There are several unfolding methods, which consist of a

- method of solution, usually iterative,

not derived from a principle. Iteration stopped, if corrections small. Their properties are studied experimentally (… the method works …).

Properties also depend on details of implementation.

II. The case $n = m = 2$

Only $n = 2$ possible true values and $m = 2$ possible observed values.

Example: classification of events into NC (neutral current) and CC (charged current).

$$\begin{pmatrix} NC \\ CC \end{pmatrix}_{\text{obs}} = \begin{pmatrix} NC \to NC & CC \to NC \\ NC \to CC & CC \to CC \end{pmatrix} \cdot \begin{pmatrix} NC \\ CC \end{pmatrix}_{\text{true}}$$

or short $\boldsymbol{g} = \boldsymbol{A}\boldsymbol{f}$ with matrix $\boldsymbol{A}$ containing *positive* probabilities.

Unfolding by matrix inversion (*what else?*):

$$\boxed{\begin{pmatrix} NC \\ CC \end{pmatrix}_{\text{true}} = \boldsymbol{f} \;=\; \boldsymbol{A}^{-1}\boldsymbol{g} = \boldsymbol{A}^{-1} \begin{pmatrix} NC \\ CC \end{pmatrix}_{\text{obs}}}$$

i.e. a linear transformation from the observed values to the true values ($\to$ straightforward error propagation).

Unfolding by inversion

Quantitative example: ε = probability for migration from original class into other class.

$$\begin{pmatrix} g_1 \\ g_2 \end{pmatrix}_{\text{obs}} = \begin{pmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}_{\text{true}}$$

or short: $\boldsymbol{g} = \boldsymbol{A}\boldsymbol{f}$.

Unfolding with error propagation:

$$\boxed{\begin{array}{rcl} \boldsymbol{f} & = & \boldsymbol{A}^{-1}\boldsymbol{g} \\ \boldsymbol{V}_f & = & \boldsymbol{A}^{-1}\boldsymbol{V}_g\boldsymbol{A}^{-1} \end{array}}$$

Inverse matrix (with *negative* off-diagonal elements):

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

Resulting covariance matrix:

$$\boldsymbol{V}_x = \frac{1}{(1-2\varepsilon)^2}\begin{pmatrix} (1-\varepsilon)^2 g_1 + \varepsilon^2 g_2 & -\varepsilon(1-\varepsilon)(g_1+g_2) \\ -\varepsilon(1-\varepsilon)(g_1+g_2) & (1-\varepsilon)^2 g_2 + \varepsilon^2 g_1 \end{pmatrix}$$

(... becomes singular in case of $\varepsilon = 0.5$)

A numerical example

Measurement and covariance matrix are:

$$\begin{array}{rcrcr} g_1 & = & 100 & \pm & 10 \\ g_2 & = & 81 & \pm & 9 \end{array} \qquad V_g = \begin{pmatrix} 100 & 0 \\ 0 & 81 \end{pmatrix}$$

Result of unfolding by inversion:

ε	$\hat{f}_1$	$\hat{f}_2$	ρ
0	100 ± 10	81 ± 9	0
0.1	102.4 ± 11.3	78.6 ± 10.2	-0.22
0.4	138 ± 35	43 ± 33	-0.92
0.5	$\pm\infty$	$\pm\infty$	singular

Statistical errors magnified because of finite resolution (migration probability ε)!

Effect in detail for $\varepsilon = 0.4$:

$$A^{-1} = \frac{1}{1-2\varepsilon}\begin{pmatrix} 1-\varepsilon & -\varepsilon \\ -\varepsilon & 1-\varepsilon \end{pmatrix} = \begin{pmatrix} 3 & -2 \\ -2 & 3 \end{pmatrix}$$

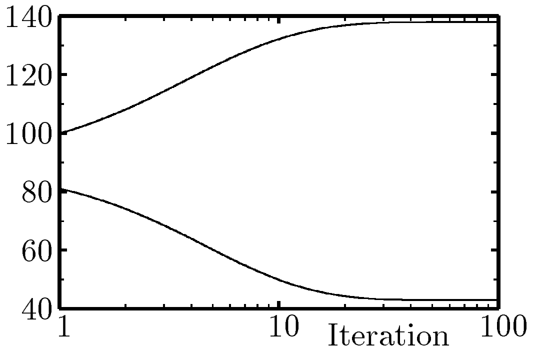
$$\begin{array}{lrcl} \text{for example} & \hat{f}_1 & = & 3 \cdot g_1 - 2 \cdot g_2 = 300 - 162 = 138 \\ \text{with} & \sigma & = & \sqrt{3^2 \cdot 100 + 2^2 \cdot 81} = 35 \qquad (\gg 10) \end{array}$$

SVD would combine the two bins into one bin.

… *the bin-by-bin correction method*

What about the bin-by-bin acceptance correction method in this case?

Parameter f_1 and f_2 versus iteration number (for $\varepsilon = 0.4$:



The method converges (of course) to the *same* values as the inversion method, but extremely slow (linear convergence).

$$f_1 = 138 \pm 13.8 \qquad f_2 = 43 \pm 4.8$$

The errors are too small (correlation by finite resolution is ignored).

A general comment on *linear* convergent methods: $x_k \to x_{k+1}$

$$\text{ratio} \qquad \frac{F(x_{k+1}) - F(x^*)}{F(x_k) - F(x^*)} \approx \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 \qquad \text{often close to 1}$$

… *Bayes' Unfolding*

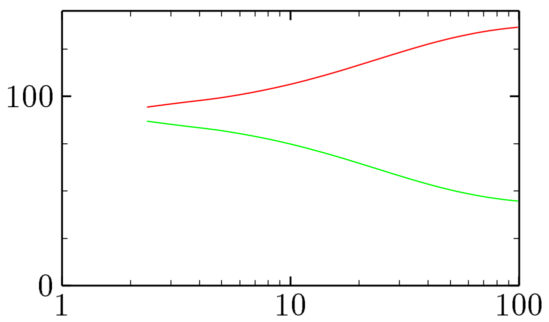
What about Unfolding "based on Bayes Theorem"?

Case $\varepsilon = 0.5$ (means *no* measurement, matrix singular):

initial values unchanged during iterations (interpreted as "no additional information from measurement").

Case $\varepsilon = 0.4$:

The two values as a function of iteration number …



Slow convergence (> 100 iterations) until a result identical to the inverse matrix method is reached.

Result of **folding**

The result of **folding** of $f(x)$ with a Gaussian resolution function (standard deviation σ) is the *observed* distribution $g(y)$:

$$g(y) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right) f(x)\,\mathrm{d}x \; ,$$

which is again periodic, with Fourier coefficients

$$\begin{aligned} \alpha_\nu &= a_\nu \quad \exp\left(-\frac{\nu^2\sigma^2}{2}\right) \\ \beta_\nu &= b_\nu \quad \exp\left(-\frac{\nu^2\sigma^2}{2}\right) \end{aligned}$$

Note: each single Fourier coefficient is modified (damped), independent of all other coefficients!

The basic elements affected by folding/unfolding are the cos and sin terms (full functions as contributions to $f(x)$ and $g(y)$), and **not** the isolated data point.

III. Unfolding of periodic functions

Discussion of a special case, where **unfolding** can be done using an analytical formula.

Assumption:

- true distribution $f(x)$ periodic (period $= 2\pi$)
- Gaussian x resolution (standard deviation $= \sigma$)

Representation of $f(x)$ as Fourier sum:

$$f(x) = \frac{a_0}{2} + \sum_{\nu=1}^{\infty} \left(a_\nu \cos \nu x + b_\nu \sin \nu x\right)$$

The sin and cos functions form an orthogonal system of functions, with coefficients:

$$\begin{aligned} a_\nu &= \frac{1}{\pi}\int_0^{2\pi} f(x) \cos \nu x \,\mathrm{d}x \\ b_\nu &= \frac{1}{\pi}\int_0^{2\pi} f(x) \sin \nu x \,\mathrm{d}x \end{aligned}$$

(or finite sum for finite set of data). In general a *few* Fourier terms sufficient for an accurate description of function $f(x)$ (higher order terms become soon very small).

A numerical example

Test function =
sum of three Breit-Wigner functions:

$$f(x) = \sum_{k=1}^{3} b_k \frac{g_k^2}{(x - x_k)^2 + g_k^2}$$

... interpreted as a periodic function function within interval $0 \ldots 2\pi$

The following figures show:

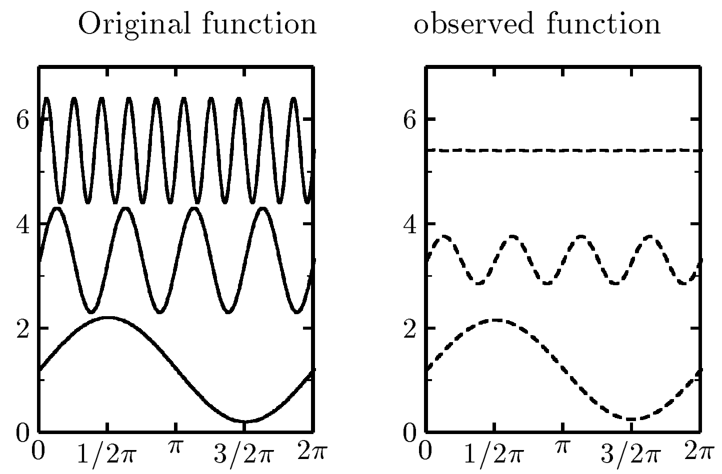
- the Fourier analysis,
- the effect of finite resolution,
- the effect from statistical errors, and
- the combination of the two effects.

Unfolding

Unfolding is very simple: the *measured* coefficients α_ν und β_ν of cos and sin terms are corrected by multiplication with factors

$$\exp\left(+\frac{\nu^2\sigma^2}{2}\right)$$

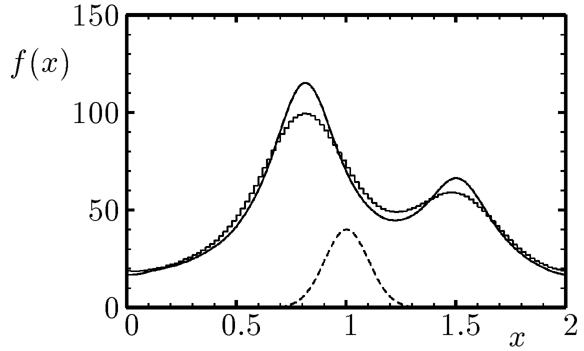
in order to get the coefficients a_ν und b_ν back.



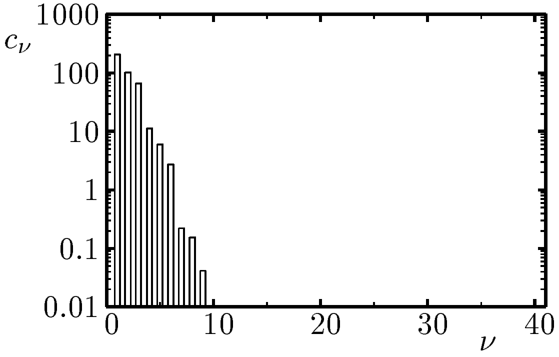
Strongly fluctuating higher order terms have to be ignored!

Effects of finite resolution

Histogram of the function(curve) after **folding** with a Gaussian resolution function ($\sigma = 0.1$, dashed curve):



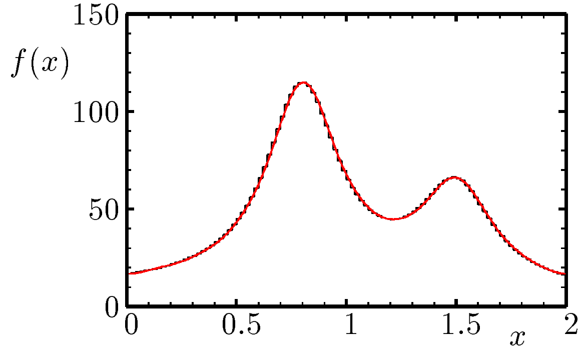
Fourier coefficients on a log. scale:



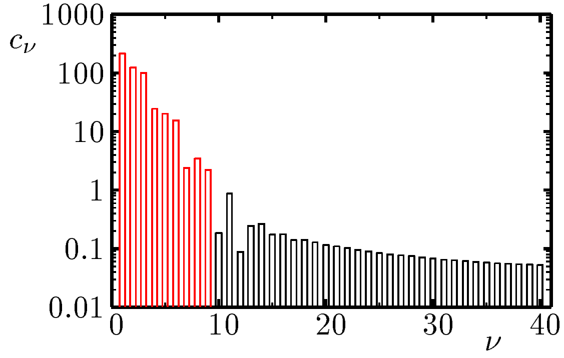
Coefficients $c_\nu \to 0$ much faster because of finite resolution.

Fourier analysis of test function

Test function as a histogram (100 bins, no errors):



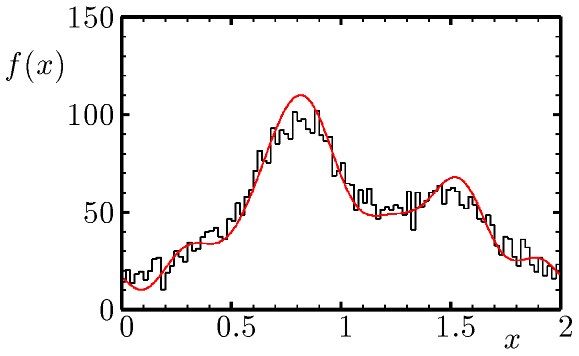
Fourier coeff. $c_\nu = \sqrt{a_\nu^2 + b_\nu^2}$ on a *log.* scale:



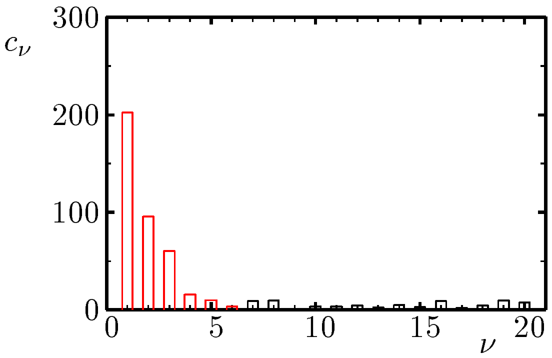
Coefficients $c_\nu \to 0$ for large ν as $1/\nu^2$.
Coefficients up to $\nu = 9$ are sufficient for the accurate parametrization of the test function (see curve in top figure = representation of terms up to $\nu = 9$).

Effects ...

by the finite resolution **plus** statistical errors:



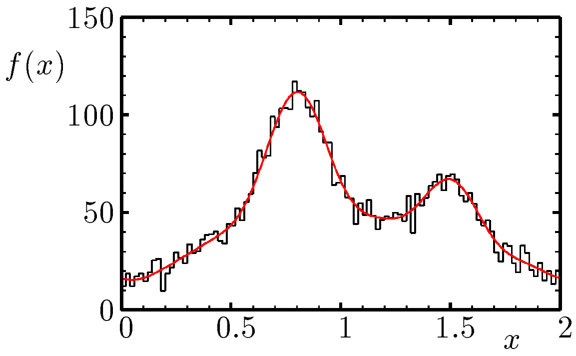
Coefficients on a linear scale.



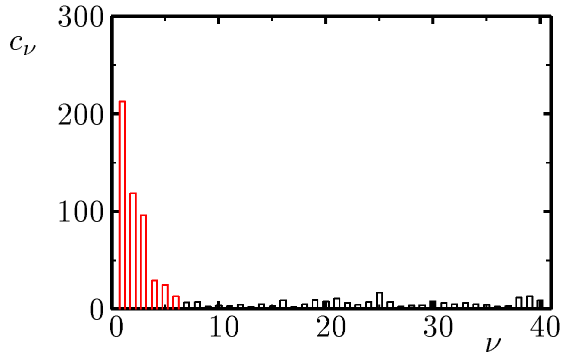
Coefficients up to $\nu \approx 6$ significant. Curve (top figure) reconstructed from the coefficients up to $\nu = 6$.

Effects due to statistical errors

Histogram with error ± 5 in each bin:



Now coefficients c_ν on a *linear* scale.



Statistical measurement errors generate a *constant* error (noise) for the coefficients.
Only the first coefficients up to $\nu = 6 \ldots 8$ are **significant** (curve in top figure reconstructed from the first 6 coefficients).

Lower limit of the bin width

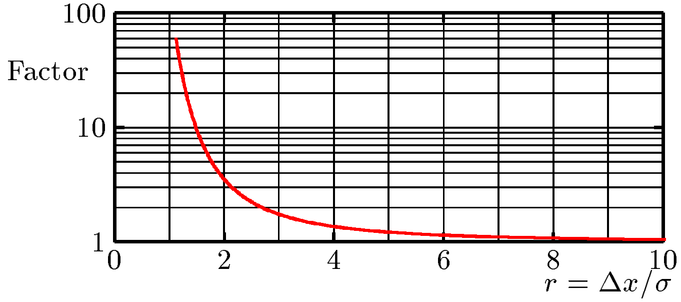
Cut-off at $\nu = \nu_0$ means:

$1 + 2 \times \nu_0$ **significant** coefficients only

and rank of covar. matrix is $1 + 2 \times \nu_0$.

Representation of result in bins (with integration of corrected function within bins) with limitation to $1 + 2 \times \nu_0$ bins (same number of degrees of freedom).

Magnification factor for statistical errors due to limited resolution (migration):



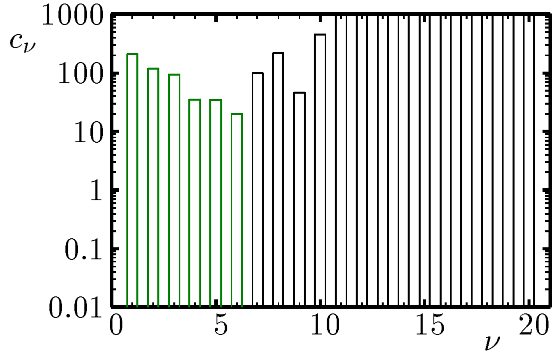
- Cut-off essential to suppress enlarged statistical fluctuations (regularisation)
- Cut-off index critical
- smallest tolerable bin width ≥ 2 standard deviations

Corrections of Fourier coefficients

Unfolding: coefficients α_ν and β_ν corrected by factor:

$$\exp\left(+\frac{\nu^2\sigma^2}{2}\right)$$

Plot of corrected coefficients c_ν:



Cut-off necessary at $\nu = 6$ in order to avoid huge fluctuations by *noise*.

Either

- sharp cut-off, or
- smooth cut-off by *regularisation*.

IV. The regularisation method – RUN

Use $m > n$ (usually $m \gg n$) and start from standard solution: in first iteration χ^2 minimization[a] with

$$(\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f})^T \ \boldsymbol{W} \ (\boldsymbol{g} - \boldsymbol{A}\boldsymbol{f}) = \text{minimum}$$

and in later iterations binned maximum likelihood (Poisson statistic in each bin)

$$-2 \ln \mathcal{L}(\boldsymbol{f}) = \text{minimum}$$

in order to allow case of few (even zero) events in bins.

Add regularisation term $\tau \cdot R(\boldsymbol{f})$, e.g.

$$R(\boldsymbol{f}) = \int |f''(x)|^2 \, \mathrm{d}x$$

with regularisation parameter τ of unknown magnitude:

- τ small: no effect of regularisation
- τ large: resulting function will be **linear** → bias.

[a] Corresponds to matrix inversion in the case $m = n$

Technical details

Solution function $f(x)$ expressed relative to (known!) MC function $f_{\mathrm{MC}}(x)$:

$$f(x) = f_{\mathrm{mult}}(x) \times f_{\mathrm{MC}}(x)$$

and $f_{\mathrm{mult}}(x)$ parameterized (in order to avoid discontinuities) as sum cubic B-splines $B_j(x)$ (3. order)[a] with coefficients $\boldsymbol{a}$:

$$f_{\mathrm{mult}}(x) = \sum_j a_j \, B_j(x)$$

Curvature simple for cubic B-splines: quadratic expression

$$\int |f''(x)|^2 \ \mathrm{d}x = \boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a}$$

with fixed symmetric matrix $\boldsymbol{C}$.
Problem: find minimum of

$$-2 \ln \mathcal{L}(\boldsymbol{f}) + \tau \cdot R(\boldsymbol{f}) \approx \ldots \boldsymbol{a}^T \boldsymbol{h} + \frac{1}{2} \boldsymbol{a}^T \boldsymbol{H} \boldsymbol{a} + \frac{1}{2} \tau \cdot \boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a} \, .$$

[a] Normal histograms are B-Splines of order 0.

Which value of τ?

Method: simultaneous diagonalization of symmetric matrices $\boldsymbol{H}$ and $\boldsymbol{C}$

$$\ldots \frac{1}{2}\boldsymbol{a'}^T\boldsymbol{a'} + \frac{1}{2}\tau \cdot \boldsymbol{a'}^T\boldsymbol{S}\boldsymbol{a'} = \text{minimum}$$

with diagonal matrix $\boldsymbol{S}$ of eigenvalues S_{jj} (corresponding to square of singular values in SVD), is equivalent to parameter transformation ($\boldsymbol{a} \to \boldsymbol{a'}$):

$$f_{\text{mult}}(x) = \sum_j a'_j P_j(x)$$

The $P_j(x)$ are orthonormal linear combinations of B-splines, the coefficients a'_j have variances of 1 and are independent.

Simultaneous diagonalization allows to see clearly effect of regularization

$$a'^{\,\text{reg}}_j = \left(\frac{1}{1+\tau S_{jj}}\right) a'^{\,\text{unreg}}_j$$

Curvature S_{jj} is increasing rapidly, resulting in a damping of coeffients with high of index value j.

Effective number of degrees of freedom

Define effective number of degrees of freedom (dof) from sum of factors:

$$m_0 = \sum_{j=1}^{m} \left(\frac{1}{1+\tau S_{jj}}\right) .$$

Approximate relation between effective number of dof and regularization parameter τ:

$$\tau \approx 1/S_{jj} \quad \text{with } j = m_0$$

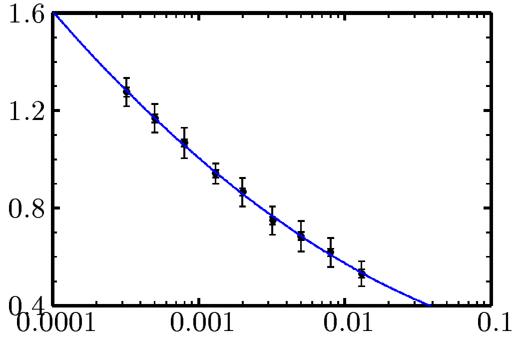
Error propagation is straightforward (standard formalism of maximum likelihood) and gives $m \times m$ covariance matrix $\boldsymbol{V}$ of coefficients $\boldsymbol{a}$, which has effective rank m_0 with $m_0 < m$); matrix $\boldsymbol{V}$ is singular.

Final transformation to m_0 data points with covariance matrix of full rank.

Example from a publication

What happens, if many "data" points are generated from a low-rank data set?

Nine strongly (+) correlated data points, while effective rank is only two … three:



Fit of parabola (3 parameters) to the data shows: **either**

- errors to big (by factor > 5), **or**
- fluctuations too small.

- Shape of MC assumption copied to the "data"!

Comments from other publications

Comments about the *simultaneous diagonalization* in RUN from different papers:

" … lead to satisfactory results, though concrete implementations happen to be quite lengthy and complicated."

" … produces satisfactory results, but it has never been widely used, probably because of certain technical complications. Unfortunately, since the true distribution is decomposed into orthogonal polynomials whose coefficients are estimated, this method only works in solving one dimensional problems."

Practical problems in unfolding

Knowledge of resolution function $A(y,x)$ essential.

The most important practical problem in unfolding is **control** of the resolution function! There are potential …

- problems in the event generation,
- problems in the detector simulation,
- problems in the measurement (data taking, reconstruction),

and usually MC statistics is too small.

$\rightarrow$ Unfolding procedure has to include back-projection of the result. This is included in RUN.

Input to RUN are n-tuples; unfolding is done in two passes through the data. In a third pass the MC events are reweighted according to the result of unfolding; this allows to generate distributions of many measured variables for comparison of data and MC simulations.

Any discrepancy has to be understood. Understanding has to be used to improve calculation.

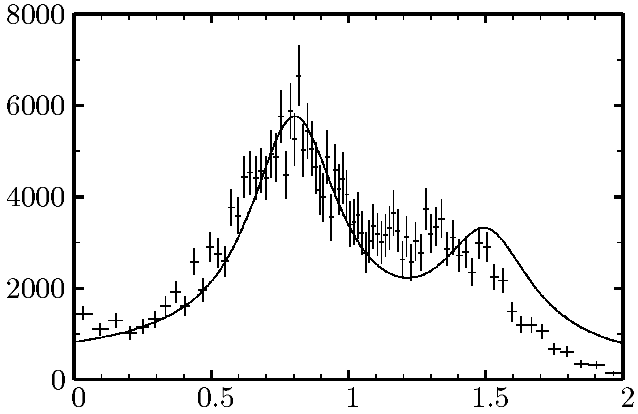
Measured distribution in RUN may be up to three-dimensional[a] in order to reduce systematic effects from incorrect MC

[a]Needs high statistics MC sample – improve RUN to use Roger Barlows method to take into account low MC statistic per bin.

Numerical example

Curve = test function (Sum $\sum_i \mathrm{BW}_i$). Histogram of 10000 generated events:

- Gaussian resolution function with $\sigma = 0.1$
- nichtlinear detector response
- reduced acceptance at both ends



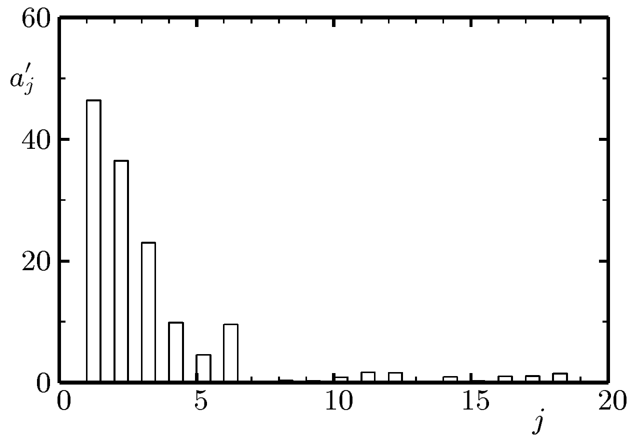
(Width of Histogram bins smaller in the center)

Minimum between the two peaks is filled up!

Amplitudes

$$f(x) = \sum_j a_j \, B_j(x) = \sum_j a'_j P_j(x)$$

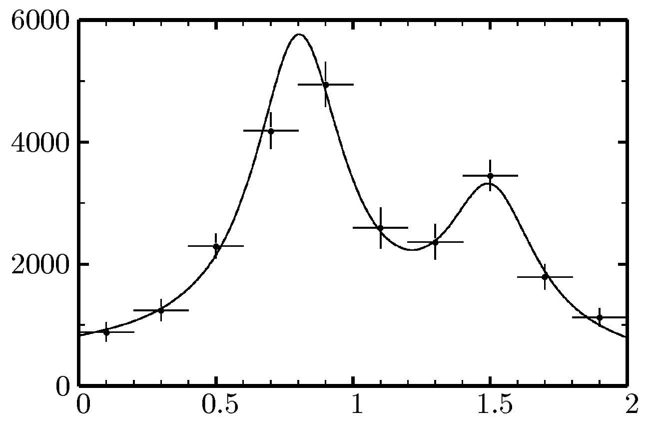
$$\begin{aligned} B_j(x) &= \text{B-Spline, } \neq 0 \text{ only in small area} \\ P_j(x) &= \text{orthonormal linear combinations} \end{aligned}$$



- Coefficients a'_j have variances of 1 and are independent!

Result of unfolding

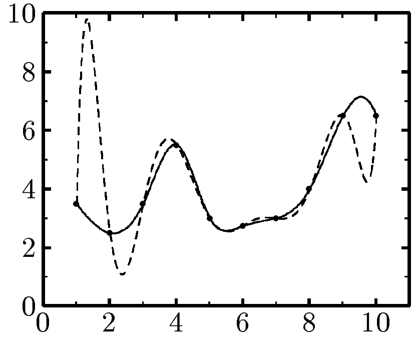
- 10 data points by integration over interval (only 10 data points)
- comparison with input curve



- Minimum between two peaks *visible* again

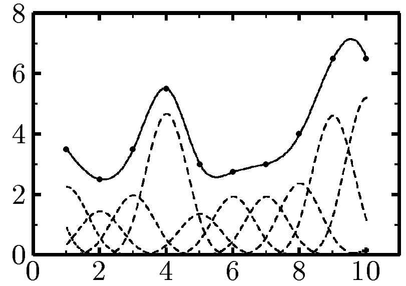
Interpolation by spline functions

Example: Interpolation of 10 data points by cubic spline functions (full curve) und polynomial of 9.th degree (dashed):



Representation by B splines (with local support):

$$f(x) = \sum_j a_j \, B_j(x) \qquad B_j(x) = \text{ cubic B spline}$$



V. Conclusions and outlook

- Unfolding is necessary for many measurements in HEP experiments
- Unfolding is a difficult statistical and mathematical problem
- It requires a detailed understanding of the detector response
- Regularisation methods, based on statistical principles, exist, but are not easy to use.

Improvement possible for $\sigma_{\gamma\gamma}$ and F_2^γ? Problem is the MC generation of the process.

Possible improvements: specialized unfolding

- Two-dimensional unfolding (successful?) (which variables?)
- Mixing of ≥ 2 MC models in unfolding (adds another parameter)
- Simultaneous fit of data at different energy

Regularisation by curvature

$$\text{Curvature } = \int |f''(x)|^2 \, \mathrm{d}x = \boldsymbol{a}^T C \boldsymbol{a}$$

with symmetric matrix C:

$$C = \begin{pmatrix} 2 & -3 & 0 & 1 & 0 & 0 & \ldots \\ -3 & 8 & -6 & 0 & 1 & 0 & \\ 0 & -6 & 14 & -9 & 0 & 1 & \\ 1 & 0 & -9 & 16 & -9 & 0 & \\ 0 & 1 & 0 & -9 & 16 & -9 & \\ 0 & 0 & 1 & 0 & -9 & 16 & \\ \vdots & & & & & & \ddots \end{pmatrix}$$

for cubic splines with equidistant knots.

The minimum of

$$-2\ln\mathcal{L}(a) + \frac{1}{2}\tau \cdot \boldsymbol{a}^T C \boldsymbol{a} = \ldots \boldsymbol{a}^T \boldsymbol{h} + \frac{1}{2}\boldsymbol{a}^T H \boldsymbol{a} + \frac{1}{2}\tau \cdot \boldsymbol{a}^T C \boldsymbol{a}$$

has to be determined. Which value of τ?

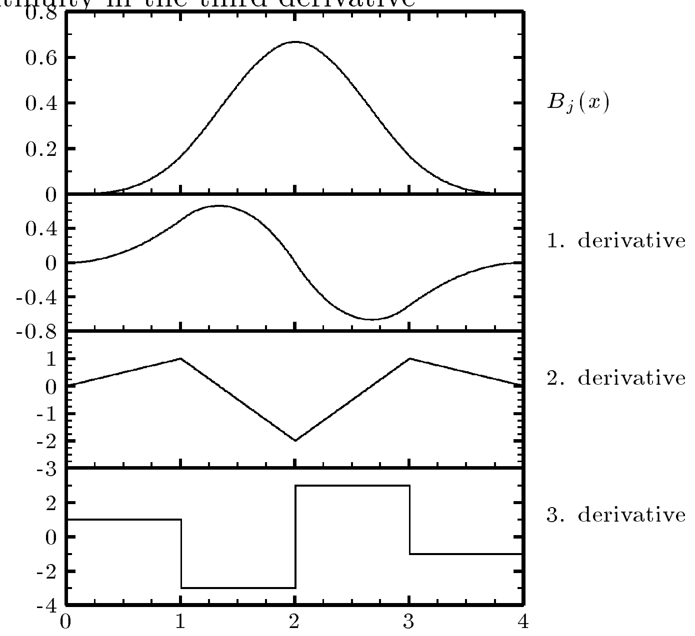
Simultaneous diagonalisation of H and C with parameter transformation ($\boldsymbol{a} \to \boldsymbol{a}'$):

$$= \ldots \boldsymbol{a}'^T \boldsymbol{a}' + \tau \cdot \boldsymbol{a}'^T \boldsymbol{S} \boldsymbol{a}'$$

B splines

Cubic B splines:

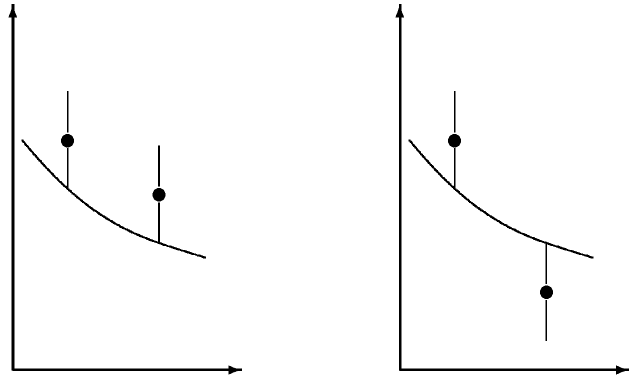
- not equal to zero in limited x interval,
- made out of four polynomials of third degree
- discontinuity in the third derivative



Numerical Example

$\sigma_1 = \sigma_2 = \sigma$ and two deviations $\Delta = \pm 1\,\sigma$

$$\chi^2 = \begin{cases} 2/(1+\rho) & \text{same sign} \\ 2/(1-\rho) & \text{opposite sign} \end{cases}$$



Probability from χ^2

	same sign	opposite sign
$\rho = 0$	37 %	37 %
$\rho = +0.9$	59 %	0.0005 %
$\rho = -0.9$	0.0005 %	59 %

Difficult interpretation of *common* deviations of > 1 data point in case of *correlated* ($\rho \neq 0$) data points.

Correlations

Case of two *correlated* variables:

$$\text{Covariance matrix} \quad \boldsymbol{V} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$\begin{aligned} \sigma_{1,2} &= \text{standard deviations of the variables} \\ \rho &= \text{correlation coefficient} \qquad -1 < \rho < +1 \end{aligned}$$

The χ^2 value for a deviation $\boldsymbol{\Delta}$ from a theroretical prediction is given by:

$$\chi^2 = \boldsymbol{\Delta}^T \left(\boldsymbol{V}^{-1}\right) \boldsymbol{\Delta}$$

with the inverse of the covariance matrix

$$\boldsymbol{V}^{-1} = \frac{1}{(1-\rho^2)} \begin{pmatrix} 1/\sigma_1^2 & -\rho/\sigma_1\sigma_2 \\ -\rho/\sigma_1\sigma_2 & 1/\sigma_2^2 \end{pmatrix}$$

The case $m = n = 4$

Case $m = n = 4$: standard solution of $\boldsymbol{g} = \boldsymbol{A}\boldsymbol{f}$ by inversion

$$\begin{aligned} \boldsymbol{f} &= \boldsymbol{A}^{-1}\,\boldsymbol{g} \\ V_f &= \left(\boldsymbol{A}^{-1}\right)^T \boldsymbol{V}_g \left(\boldsymbol{A}^{-1}\right) \qquad \text{error propagation} \end{aligned}$$

$$\boldsymbol{A} = \begin{pmatrix} 0.8 & 0.2 & 0.0 & 0.0 \\ 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 \\ 0.0 & 0.0 & 0.2 & 0.8 \end{pmatrix} \qquad \boldsymbol{A}^{-1} = \begin{pmatrix} 1.38 & -0.52 & 0.19 & -0.05 \\ -0.52 & 2.10 & -0.76 & 0.19 \\ 0.19 & -0.76 & 2.10 & -0.52 \\ -0.05 & 0.19 & -0.52 & 1.38 \end{pmatrix}$$

Inverse matrix $\boldsymbol{A}^{-1}$ has mainly negative off-diagonal elements.

Numerical example: $\boldsymbol{f} = \begin{pmatrix} 100 \\ 121 \\ 81 \\ 64 \end{pmatrix}$ yields measurement $\boldsymbol{g} = \begin{pmatrix} 104.2 \\ 108.8 \\ 85.6 \\ 67.4 \end{pmatrix}$

Unfolding $A^{-1} \begin{pmatrix} 104.2 \\ 108.8 \\ 85.6 \\ 67.4 \end{pmatrix} \Rightarrow \begin{pmatrix} 100 \\ 121 \\ 81 \\ 64 \end{pmatrix}$

Numerical example

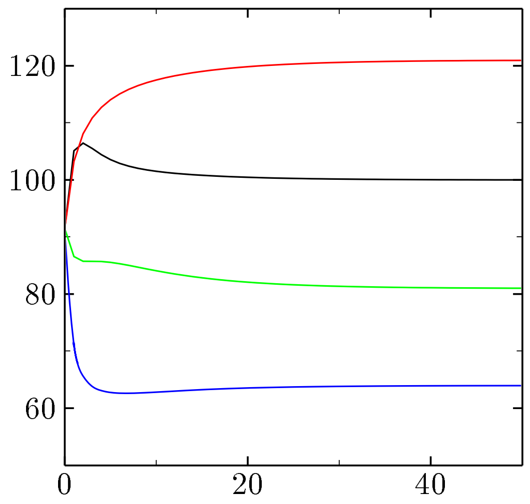
Response matrix $\boldsymbol{A}$ as before:

$$\boldsymbol{A} = \begin{pmatrix} 0.8 & 0.2 & 0.0 & 0.0 \\ 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 \\ 0.0 & 0.0 & 0.2 & 0.8 \end{pmatrix}$$

Same $\boldsymbol{f}$ and $\boldsymbol{g}$ as before, but now:

Unfolding based on Bayes' Theorem

Components of solution $\boldsymbol{f}$ as a function of iteration number



... convergence after ≈ 50 iterations to same result.

Numerical example

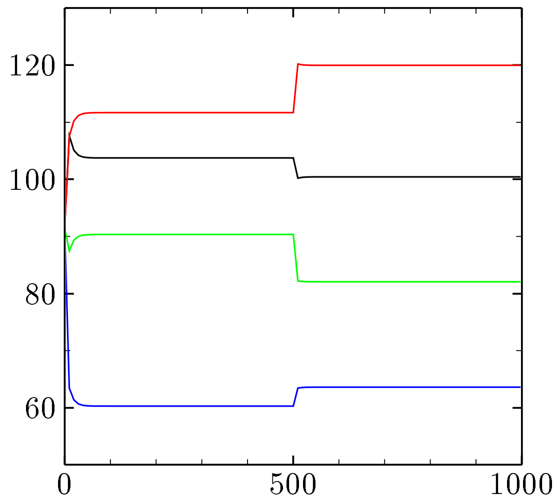
Response matrix $\boldsymbol{A}$ again slightly changed:

$$\boldsymbol{A} = \begin{pmatrix} 0.5 & 0.5 & 0.0 & 0.0 \\ 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 \\ 0.0 & 0.0 & 0.5 & 0.5 \end{pmatrix}$$

Same $\boldsymbol{f}$ as before:

Unfolding based on Bayes' Theorem

Components of solution $\boldsymbol{f}$ as a function of iteration number



... no change of result after 50 iterations, still wrong after 500 iterations. Change of components *by hand* after 500 iterations almost to result – no change of result in further iterations.

Numerical example

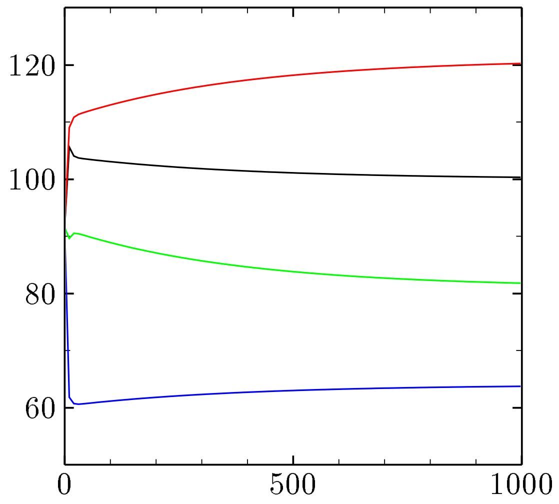
Response matrix $\boldsymbol{A}$ slightly changed:

$$\boldsymbol{A} = \begin{pmatrix} 0.6 & 0.4 & 0.0 & 0.0 \\ 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 \\ 0.0 & 0.0 & 0.4 & 0.6 \end{pmatrix}$$

Same $\boldsymbol{f}$ as before:

Unfolding based on Bayes' Theorem

Components of solution $\boldsymbol{f}$ as a function of iteration number



... convergence after ≈ 1000 iterations to same result as inverse method.