Statistical Methods in Particle Physics

9. Unfolding

Heidelberg University, WS 2023/24

Klaus Reygers, Martin Völkl (lectures)
Ulrich Schmidt, (tutorials)

Deconvolution

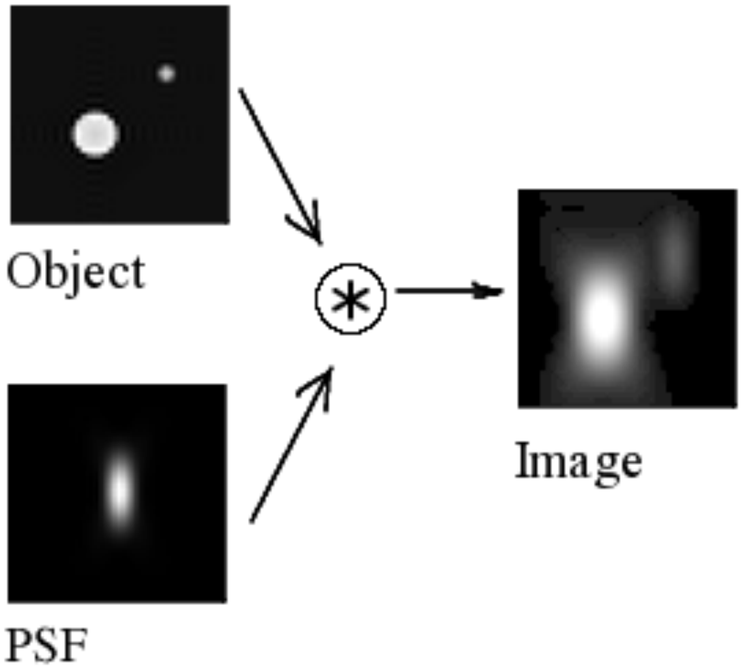
Finite resolution of the detector smears the quantities we're interested in.

Goal:
smeared information
→ original information

This is called *deconvolution* or *unfolding*

"Inverse problem"

Problem can be ill-posed in the sense that unfolded result can be very sensitive to small perturbations in the data



Example:
Smearing of a telescope image

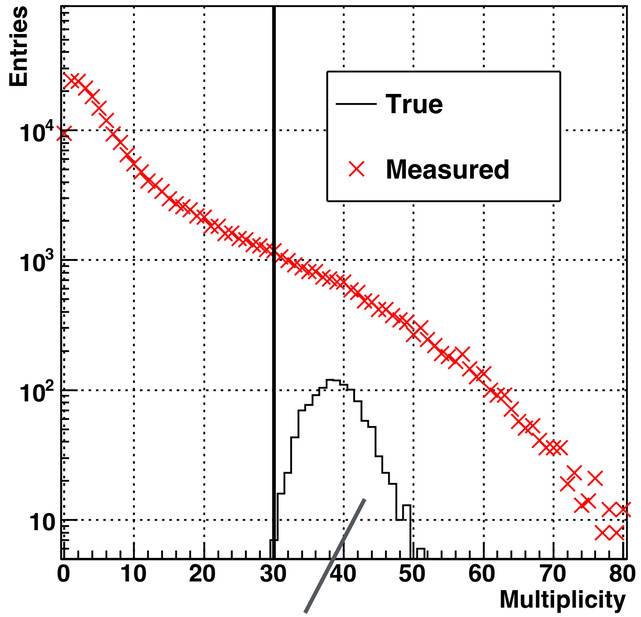
https://en.wikipedia.org/wiki/Point_spread_function

Examples in particle physics

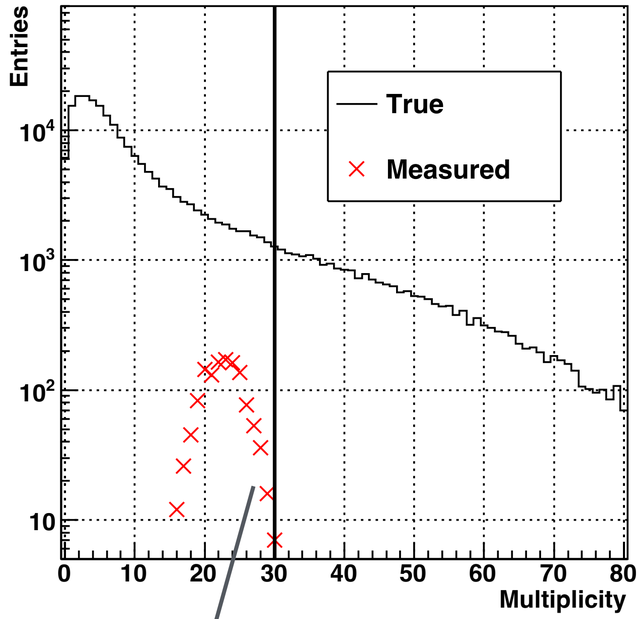
- Multiplicity distributions $P(N_{ch})$
  - Measured multiplicity differs from true charged particle multiplicity due to detector effects (efficiency, fake hits, …)
- p_T spectra, e.g., π^0 spectrum measured with a calorimeter
  - finite energy resolution and shower overlaps in a calorimeter affect the p_T of the reconstructed shower

Example: multiplicity distributions in pp collisions

arXiv:0912.0023



true N's contributing to measured $N = 30$



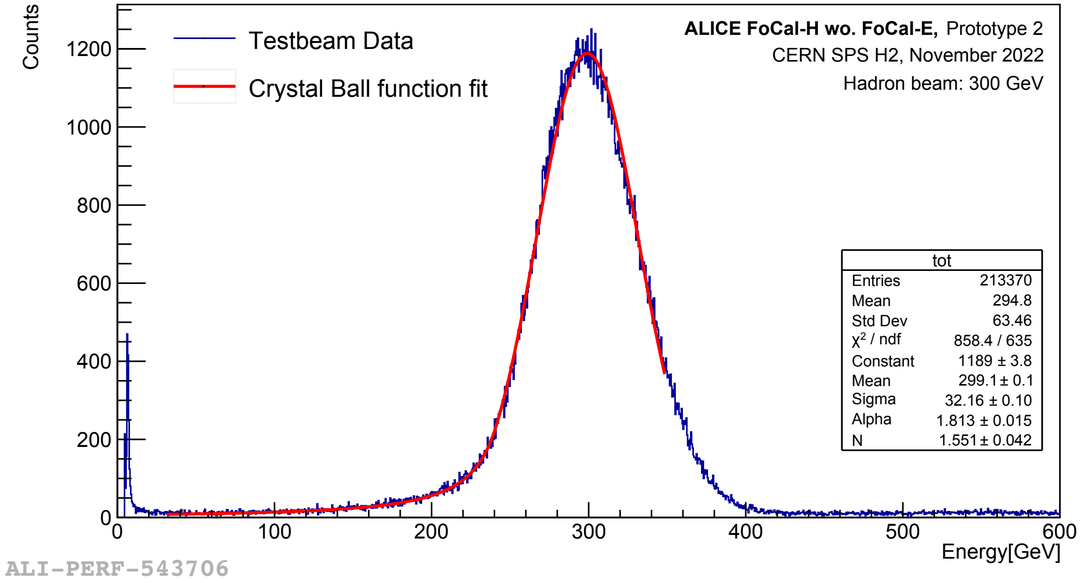
measured N's for a true $N = 30$

The response function

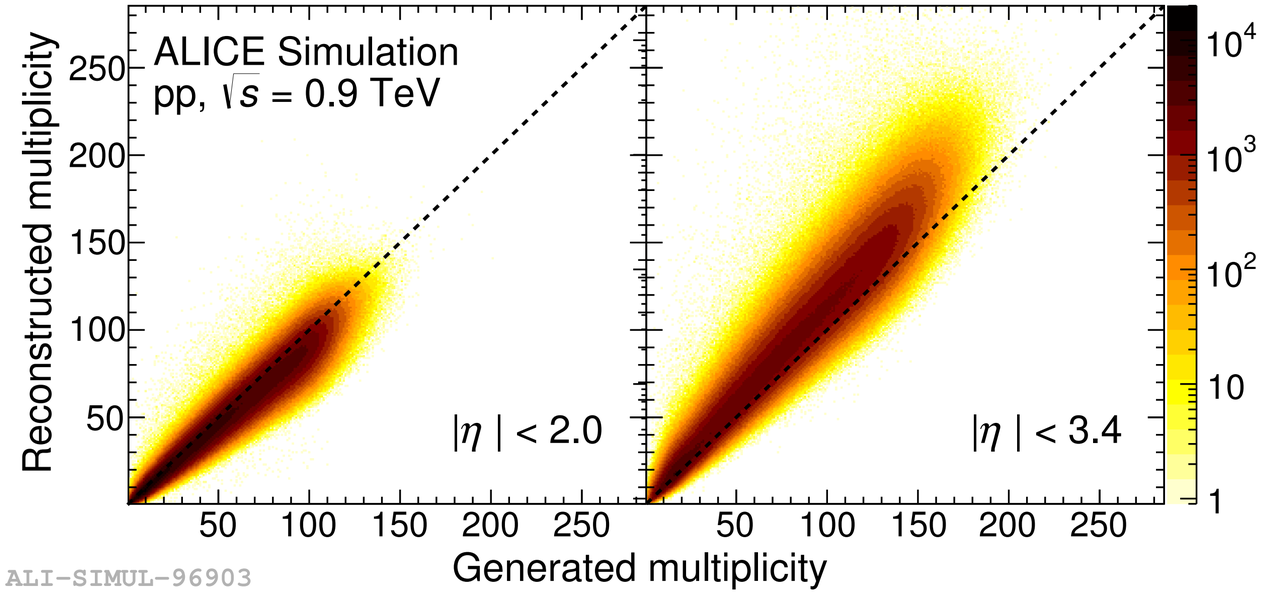
- Central object of the problem
- Give distribution of measured value for a given true value, e.g. for energy measurement:

$$R(E_{\text{meas}} \,|\, E_{\text{true}}) = p(E_{\text{meas}} \,|\, E_{\text{true}})$$

- In the simplest case a normal distribution
- To include physical effects, use simulations
- Usually contains a lot of physics
- When plotting generated vs. reconstructed, quantity, need to normalise for fixed generated
- We'll assume that we know the response perfectly in the following
  - This may not always be true in the real world



ALICE Collaboration



The response function

Suppose we deal with continues variables (e.g., transverse momentum)

$f_t(x_t)$: distribution of true values (normalized to unity)

$f_m(x_m)$: distribution of measured values (normalized to unity)

$f_b(x_m)$: distribution of background (normalized to unity)

Response function R:

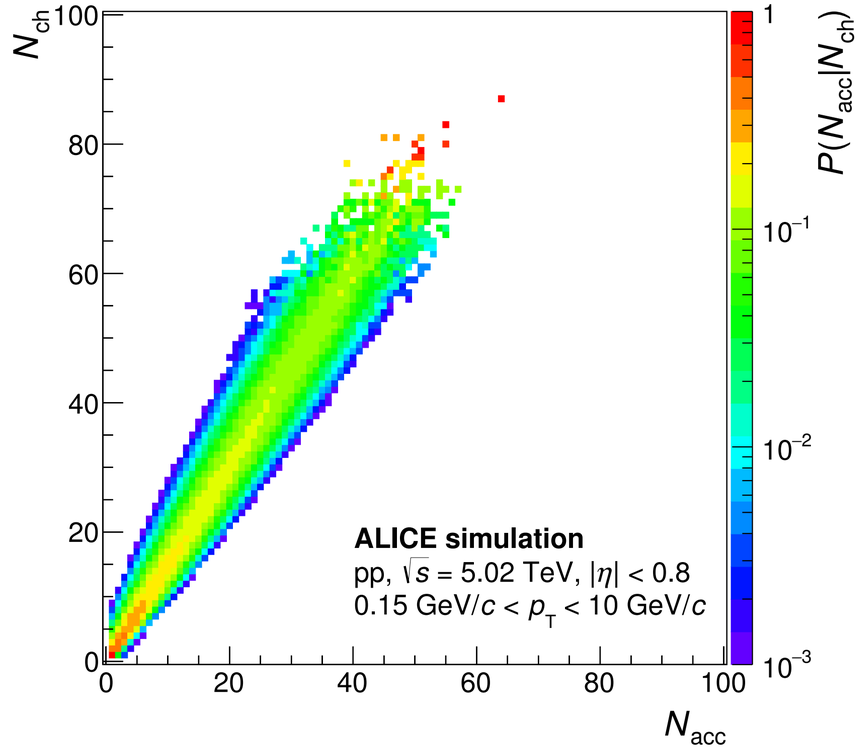
$$R(x_m|x_t) = r(x_m|x_t) \times \varepsilon(x_t)$$

probability (density) to observe x_m given x_t

"smearing" "efficiency"

By construction, one has

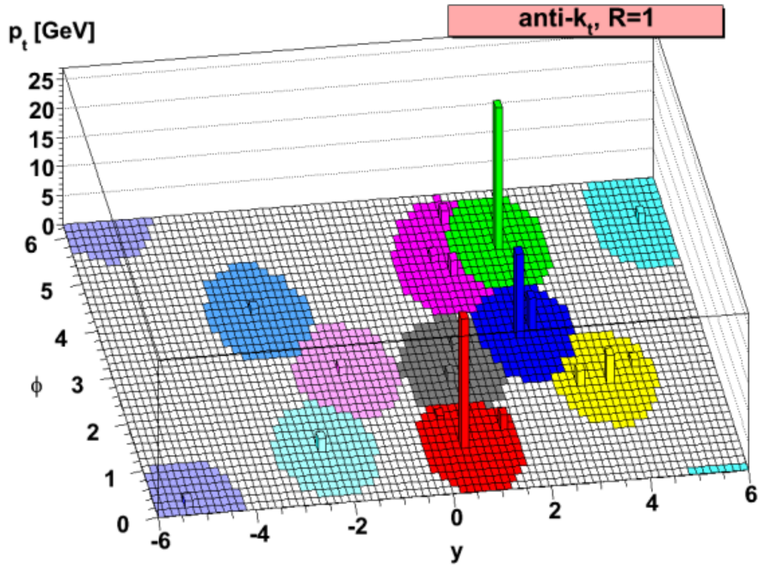
$$\int_{\Omega_m} r(x_m|x_t)\,\mathrm{d}x_m = 1$$



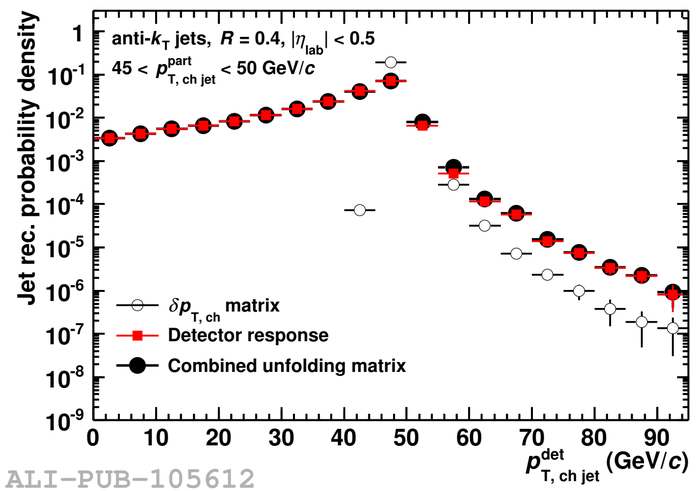
Comparison of created and accepted charged particles

Typical case: Jet energy reconstruction

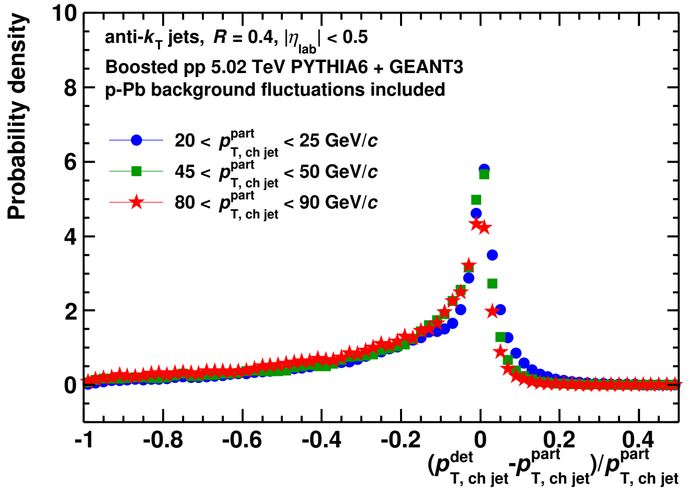
- Jets are usually supposed to represent partons
- But not all particles belonging to the Parton shower may be found
- The momentum of the reconstructed particles also has an uncertainty
- Particles from other processes may be added as well
- Much care needs to be taken to ask the right question and define the response function
- Which parts of the event actually belong to the jet and which don't?
- Much time and effort usually taken to define the simulations which yield the response function
- Much harder for heavy ion collisions due to large background



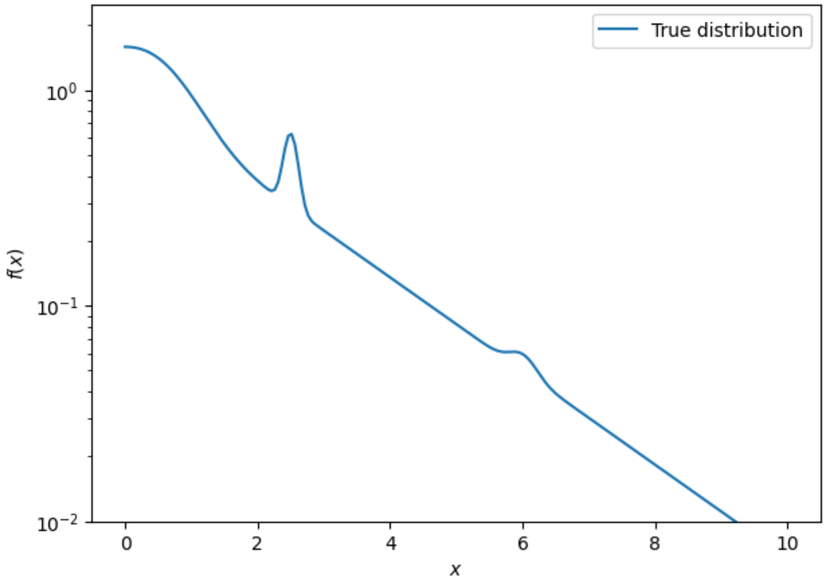
The anti-kt jet clustering algorithm, Cacciari, Salam

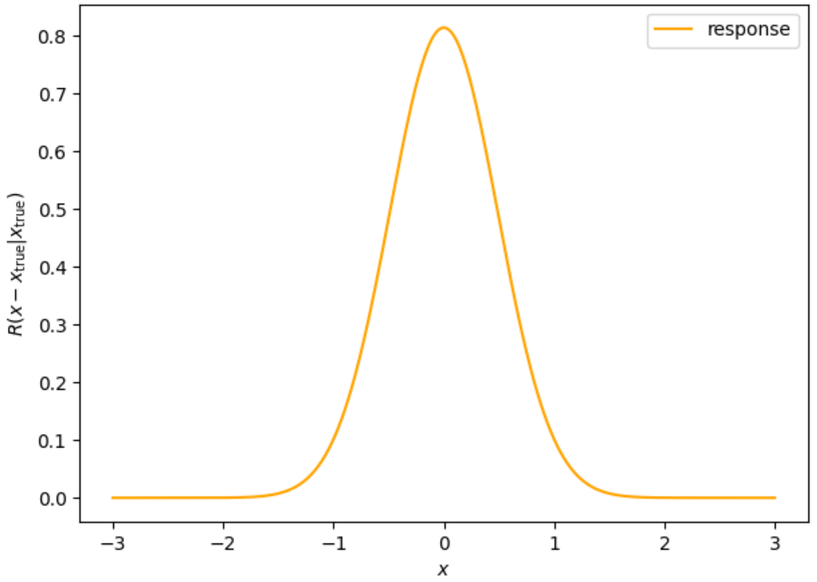


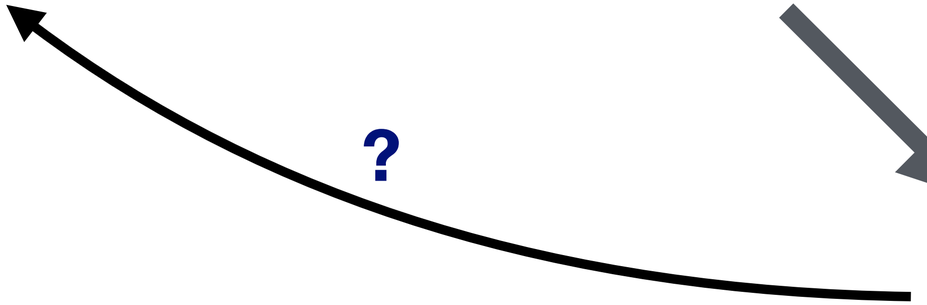
ALI-PUB-105612

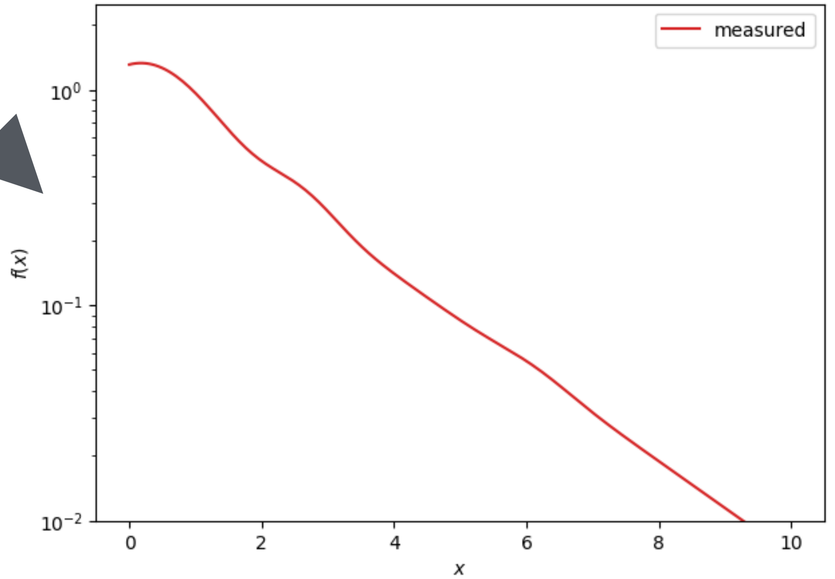


Convolution and Deconvolution





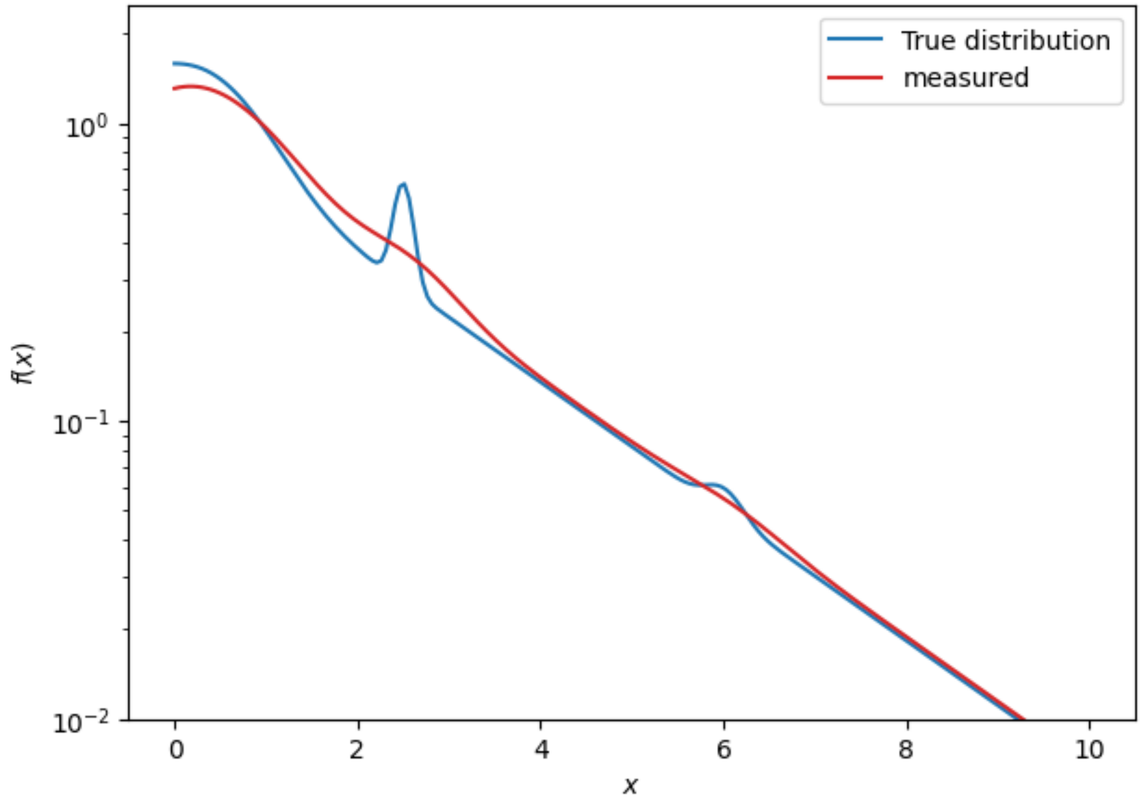




- For fixed response the effect is a convolution
- Can we undo the convolution somehow (“deconvolution”)?
- What about the uncertainties?

Convolution and Deconvolution

- The detector response tends to “wash out” peaks and other details
- Steeper parts of the distribution become less steep
- It seems like some information is lost due to the smearing
- Intuition: This should lead to larger uncertainties when inverting
- Details smaller than the detector resolution should be hard to detect



Response Matrix (1)

Further definitions:

m_{tot} : total number of true events

n_{tot} : total number of measured events

b_{tot} : total number of background events

$$\mu_{\text{tot}} = E[m_{\text{tot}}], \quad \nu_{\text{tot}} = E[n_{\text{tot}}], \quad \beta_{\text{tot}} = E[b_{\text{tot}}]$$

It is practical to work with discrete bins. E.g., probability to find x_t in bin j:

$$p_j = \int_{\text{bin } j} \mathrm{d}x_t \, f_t(x_t), \quad \mu_j = \mu_{\text{tot}} \times p_j$$

Ignoring backgrounds, the expected measured number of entries in bin i is:

$$\begin{aligned} \nu_i &= \mu_{\text{tot}} \int_{\Omega_t} \mathrm{d}x_t \, \text{Prob}(x_m \text{ in } i | \text{true } x_t, \text{ detected}) \\ &\quad \times \text{Prob}(\text{detect } x_t) \times \text{Prob}(\text{produce } x_t) \\ &= \mu_{\text{tot}} \int_{\text{bin } i} \mathrm{d}x_m \int_{\Omega_t} \mathrm{d}x_t \, r(x_m|x_t) \varepsilon(x_t) f_t(x_t) \end{aligned}$$

Response Matrix (2)

Further definitions:

$$\int_{\Omega_t} \mathrm{d}x_t = \sum_{j=1}^{M} \int_{\text{bin } j} \mathrm{d}x_t$$

$$\nu_i = \mu_{\text{tot}} \int_{\text{bin } i} \mathrm{d}x_m \sum_{j=1}^{M} \int_{\text{bin } j} \mathrm{d}x_t \, r(x_m|x_t)\varepsilon(x_t)f_t(x_t)$$

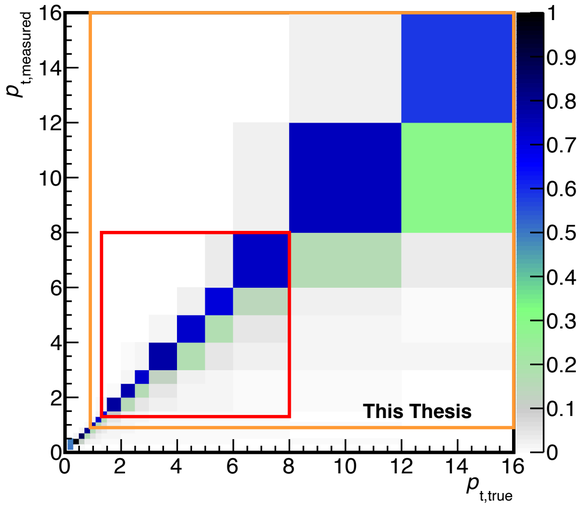
$$= \sum_{j=1}^{M} \int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \, \frac{r(x_m|x_t)\varepsilon(x_t)f_t(x_t)}{\mu_j/\mu_{\text{tot}}} \mu_j$$

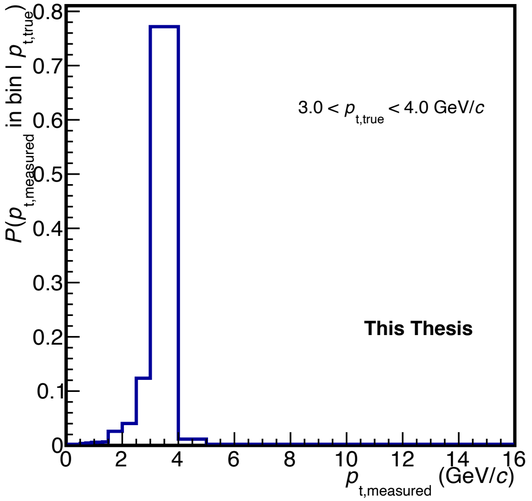
This may be written as

$$\nu_i = \sum_{j=1}^{M} R_{ij}\mu_j$$

with the components of the response matrix R_{ij} given by

$$R_{ij} = \frac{\int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \, r(x_m|x_t)\varepsilon(x_t)f_t(x_t)}{\int_{\text{bin } j} \mathrm{d}x_t f(x_t)}$$





PhD Thesis of M. Völkl

Response matrix (3)

In other words:

$$R_{ij} = \text{Prob}(\text{observed in bin } i | \text{true in bin } j)$$

Obviously, summing the response matrix over i gives the efficiency:

$$\sum_{i=1}^{N} R_{ij} = \varepsilon_j$$

In compact matrix form (including background):

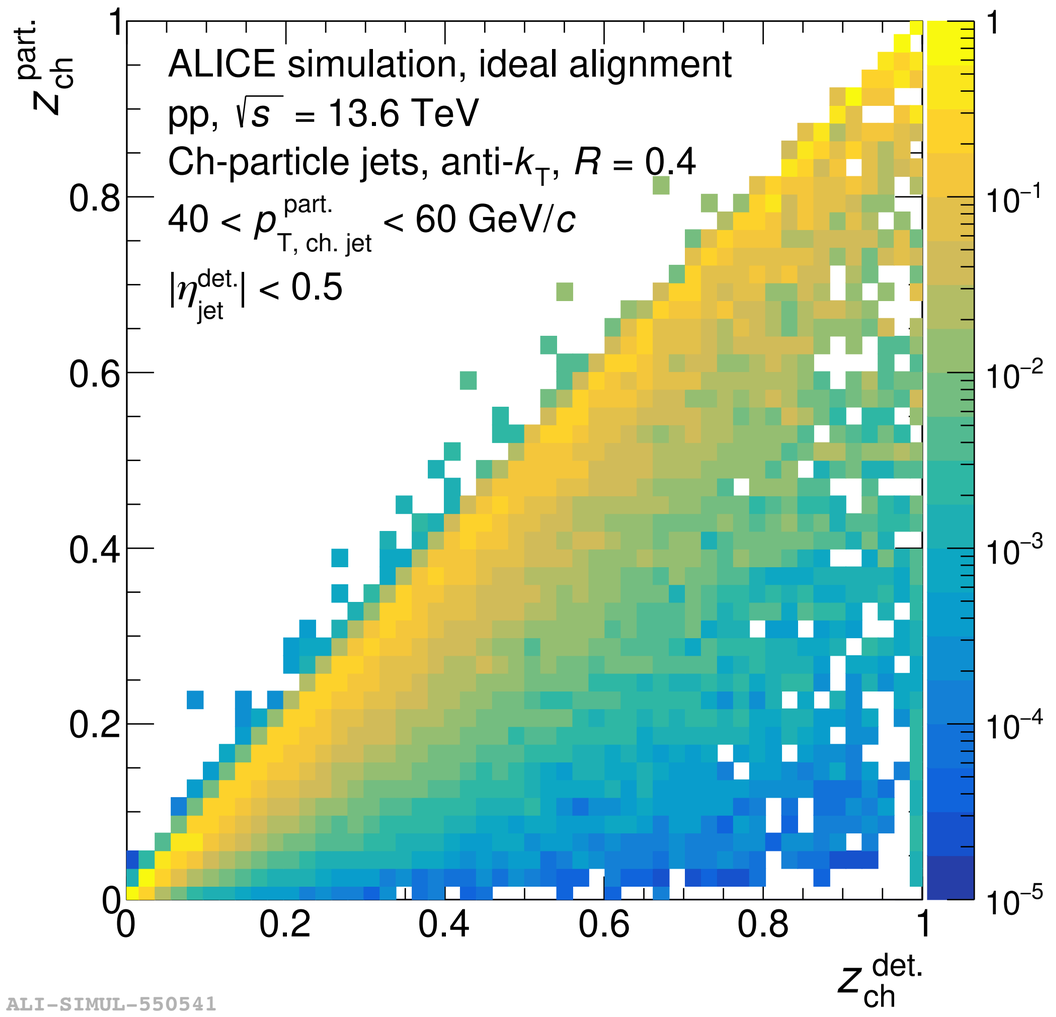
$$\nu_i = \sum_{j=1}^{M} R_{ij}\mu_j + \beta_i \qquad\qquad \vec{\nu} = R\vec{\mu} + \vec{\beta}$$

Response matrix depends on $f_t(x_t)$ which we want to know. However, if we make the bins small enough $f_t(x_t) \approx$ const. within a bin and drops from the ratio:

$$R_{ij} = \frac{\int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \; r(x_m|x_t)\varepsilon(x_t)f_t(x_t)}{\int_{\text{bin } j} \mathrm{d}x_t f(x_t)} \approx \frac{1}{\Delta x_{t,j}} \int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \; r(x_m|x_t)\varepsilon(x_t)$$

Response matrix (4)

- The response matrix usually comes from simulations
- Draw true and reconstructed bins
- If true distribution is different from simulated, this can give a bias
- That is only a problem, if the response varies strongly over a bin
- Smaller bins are better in this regard
- But it is not the only consideration
- We will ignore statistical fluctuations in the response matrix here



Unfolding by inverting the response matrix (1)

We have

$$\vec{\nu} = R\vec{\mu} + \vec{\beta}$$

Replace $\vec{\nu}$ by $\vec{n}$ to obtain and obvious estimator for the true distribution:

$$\hat{\vec{\mu}} = R^{-1}(\vec{n} - \vec{\beta})$$

This solution minimizes

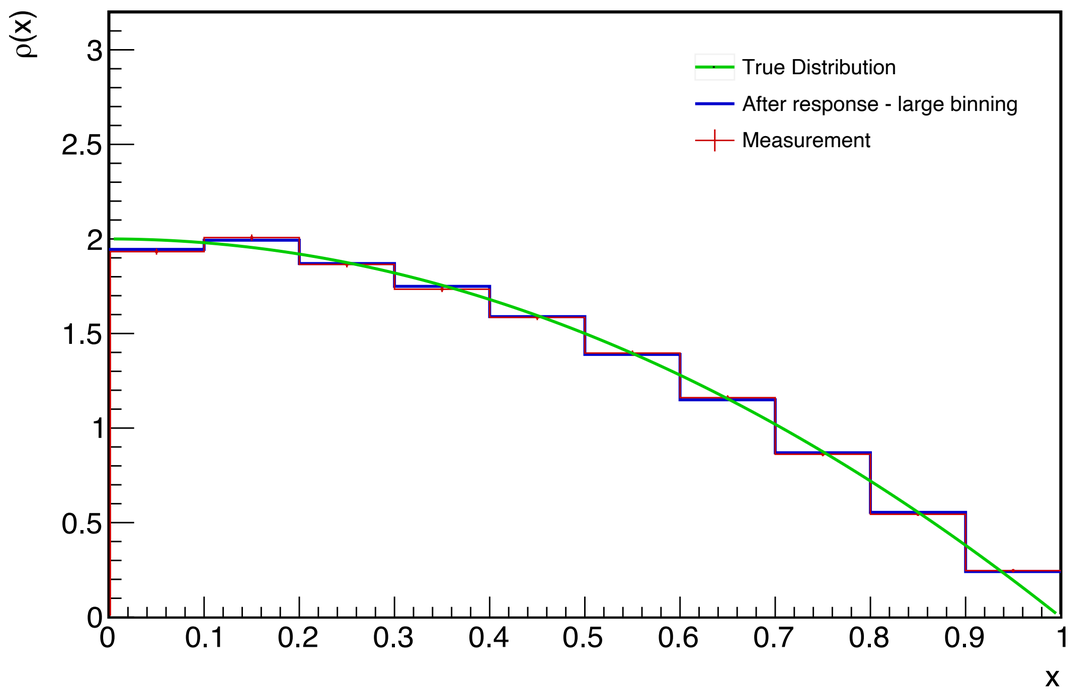
$$\chi^2(\vec{\mu}) = (\vec{\nu}(\vec{\mu}) - \vec{n})^{\mathsf{T}} V^{-1} (\vec{\nu}(\vec{\mu}) - \vec{n}) \qquad \text{where} \qquad V_{i,j} = \mathrm{cov}[n_i, n_j]$$

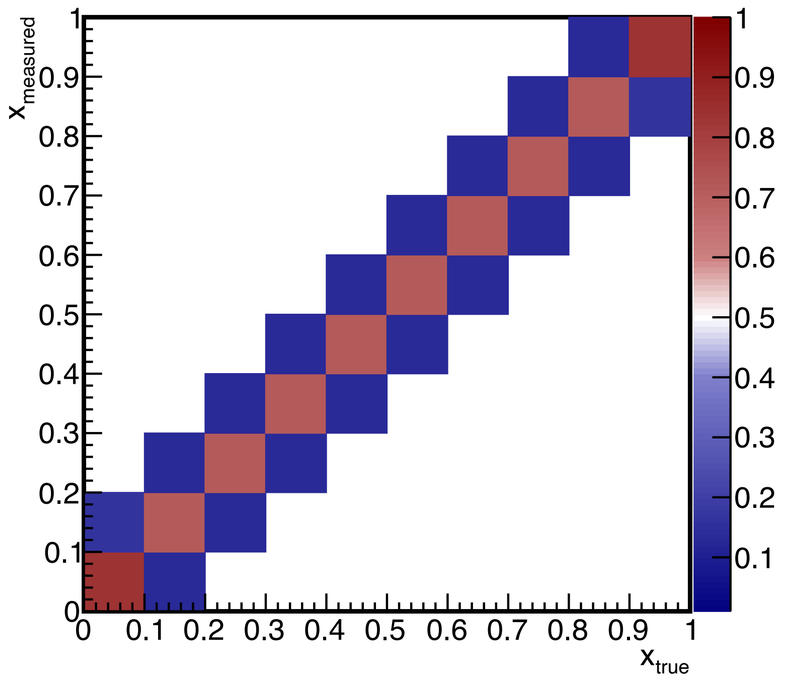
It can be shown that the covariance matrix of the solution is given by

$$U = R^{-1} V (R^{-1})^{\mathsf{T}}$$

This is the unique solution to the original problem!

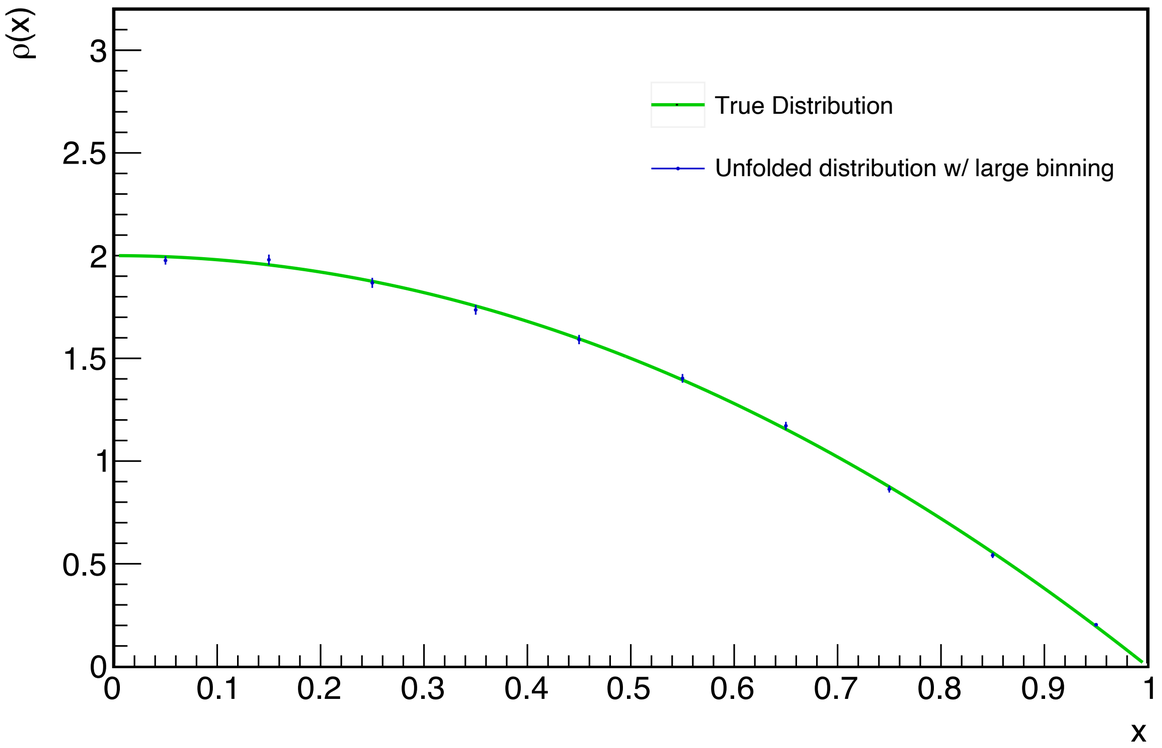
Example

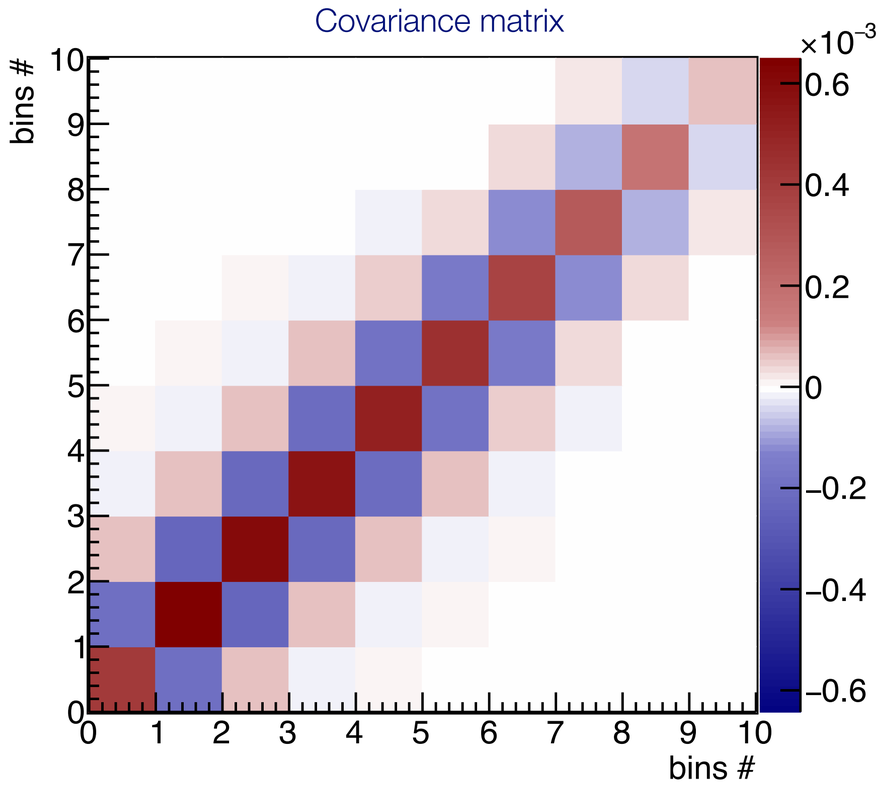




- Response is Gaussian, but with a width less than one bin
- Measurement fluctuates around the convoluted distribution

Example - continued





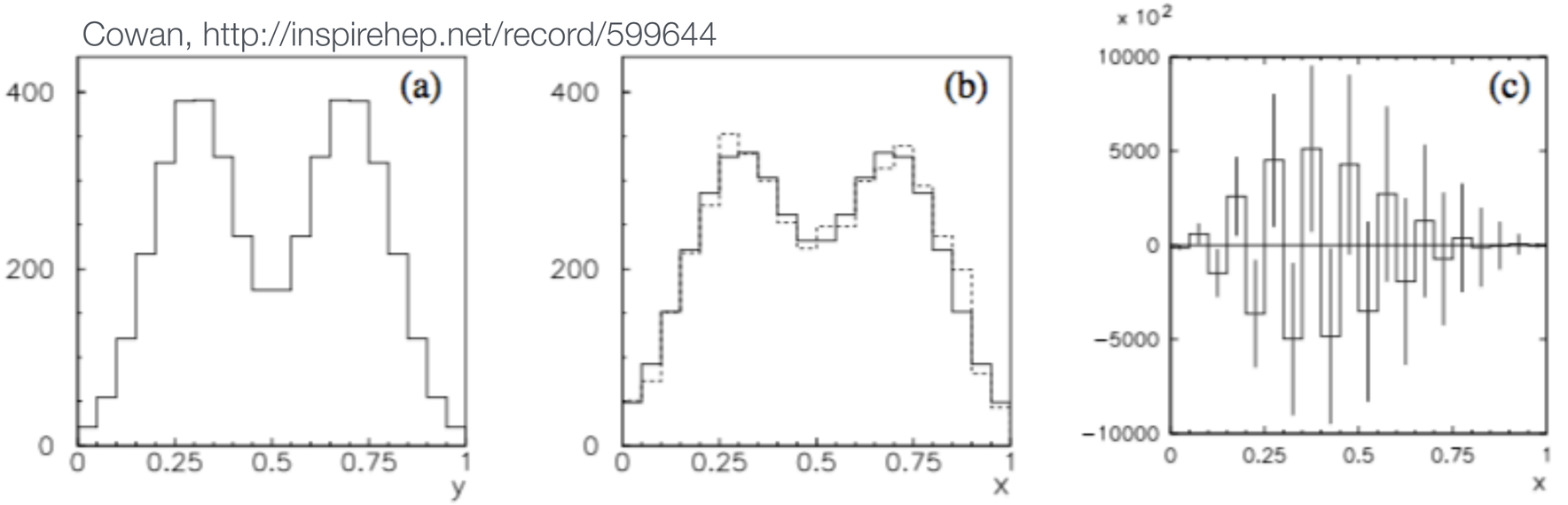
- Unfolded distribution agrees with true distribution - convolution is undone
- Covariance matrix shows anticorrelation between adjacent bins
  - All counts can come from current or adjacent bin

Unfolding by inverting the response matrix (2)

It can also be shown that matrix inversion is unbiased an has minimal variance.

This sounds good … let's try it.

Cowan, http://inspirehep.net/record/599644



true distribution $\vec{\mu}$

solid line: $\vec{\nu} = R\vec{\mu}$

dashed line:
sampled data $\vec{n}$ from $\text{Poisson}(n_i, \nu_i)$

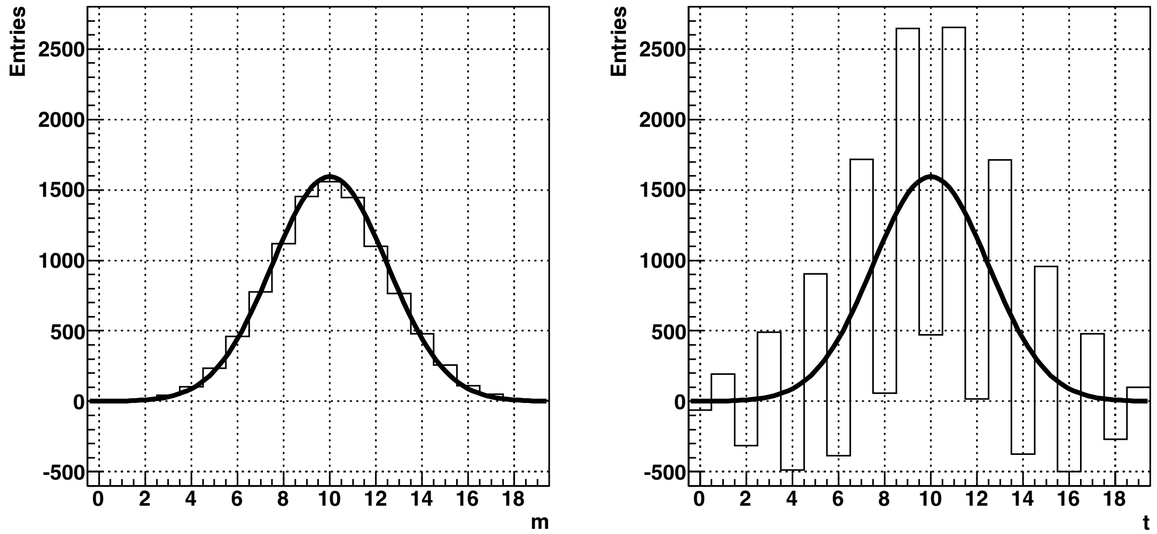
estimate $\hat{\vec{\mu}} = R^{-1}\vec{n}$

This looks like a disaster … unfolded distribution very different from the true one

Unfolding by inverting the response matrix (3)

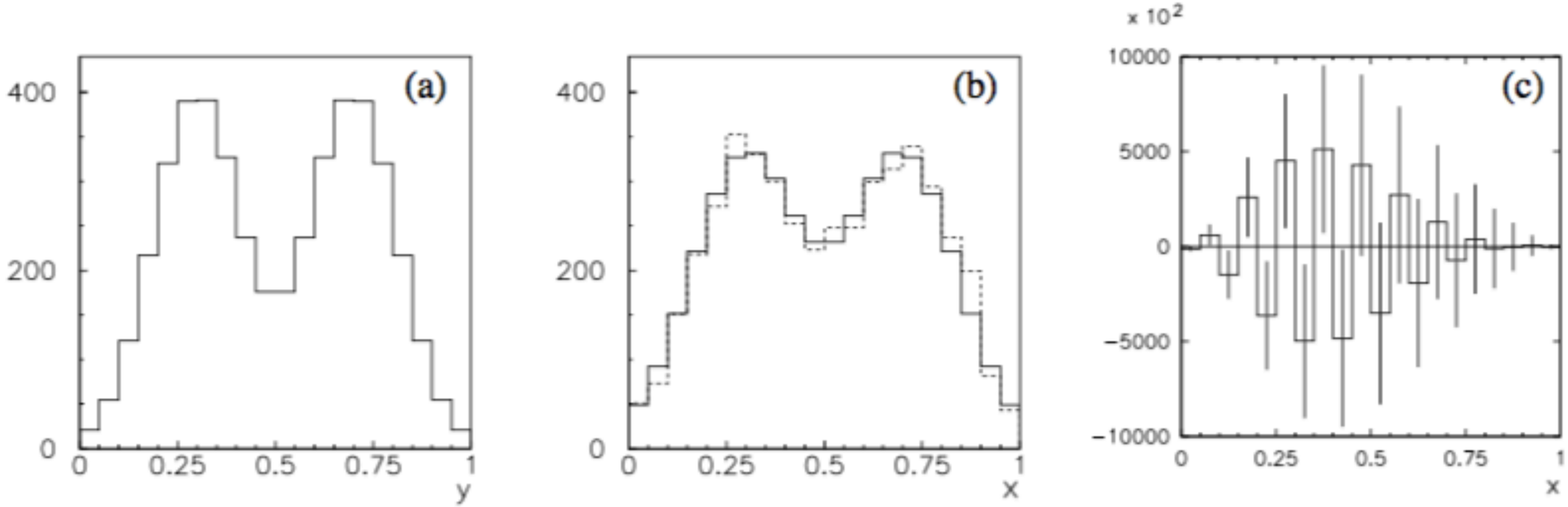
Another example:

$$R = \begin{pmatrix} 0.75 & 0.25 & 0 & & \cdots \\ 0.25 & 0.50 & 0.25 & 0 & \\ 0 & 0.25 & 0.50 & 0.25 & \\ & 0 & 0.25 & 0.50 & \\ \vdots & & & & \ddots \end{pmatrix}.$$



- Same conclusion: we don't get the desired (smooth) answer - what is going on here?
- Important: Don’t confuse symptom and cause → the large fluctuations show that we forgot something

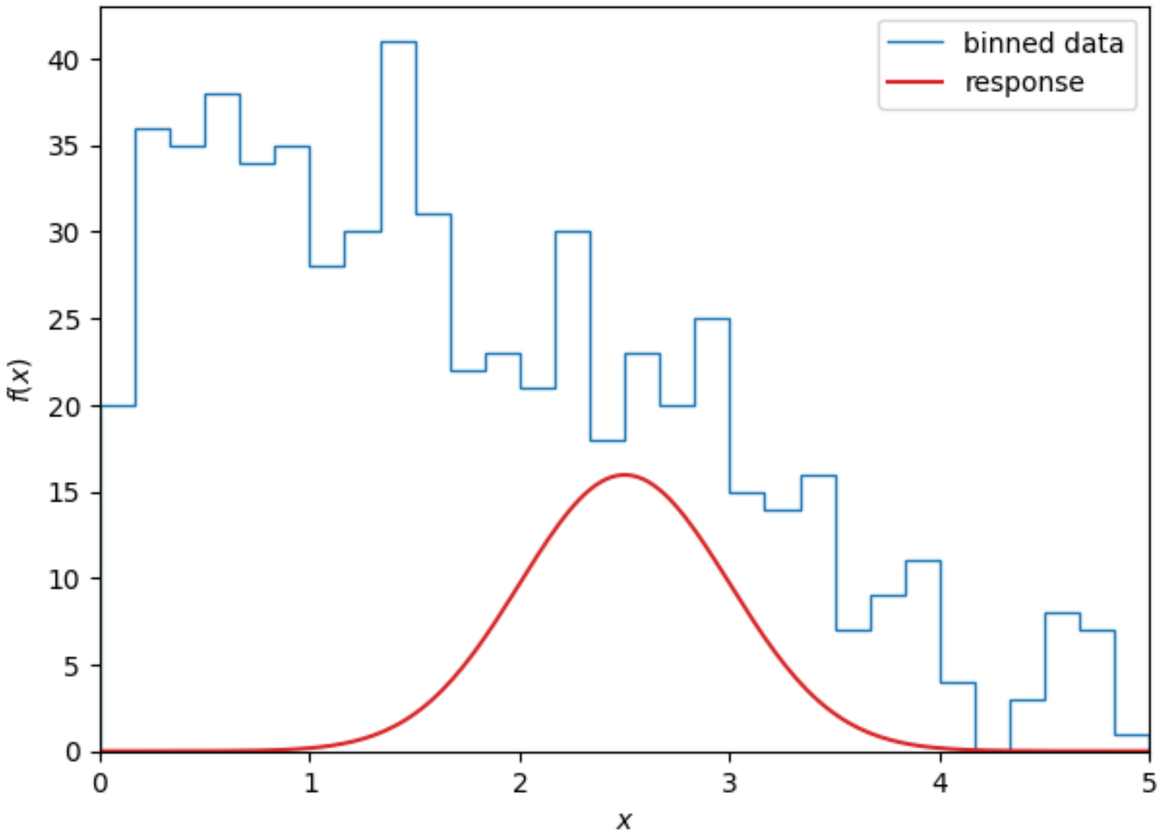
“Unphysical” solutions



- In the literature these are often called “unphysical”, “unacceptable”, “useless” or similar
- Usually in papers that propose a new type of regularisation
- The phrasing also suggests that they usually appear
- Have to be a bit careful about the language

Binning and resolution

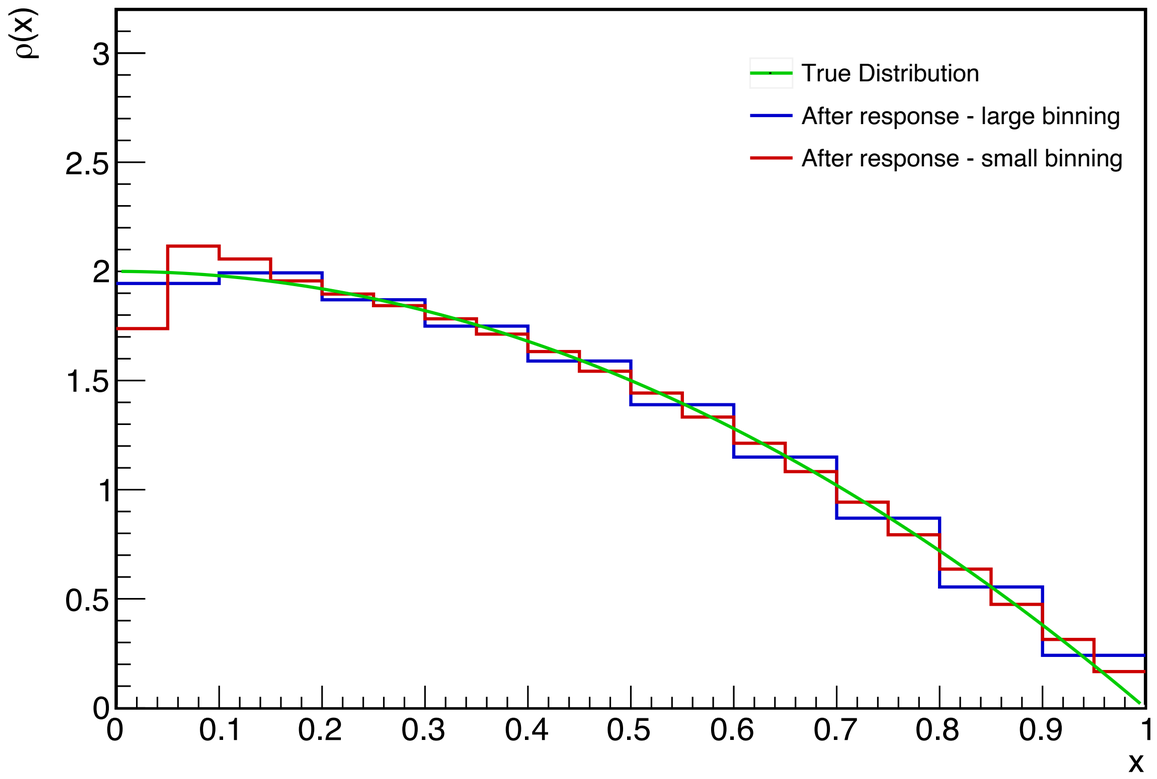
- The response function represents the detector resolution for the quantity on the horizontal axis
- E.g. for astronomy, the point spread function tells us something about the smallest resolvable angles
- When unfolding, we are asking: “How many true counts are in this bin?”
- If the bin size is smaller than the resolution, we would expect this to lead to problems
  - We are asking for an answer at a smaller scale than the resolution
- There is a variety of possible distributions that lead to a similar smeared out result
- In general, we *can* measure things smaller than the resolution, but it requires a lot of statistics



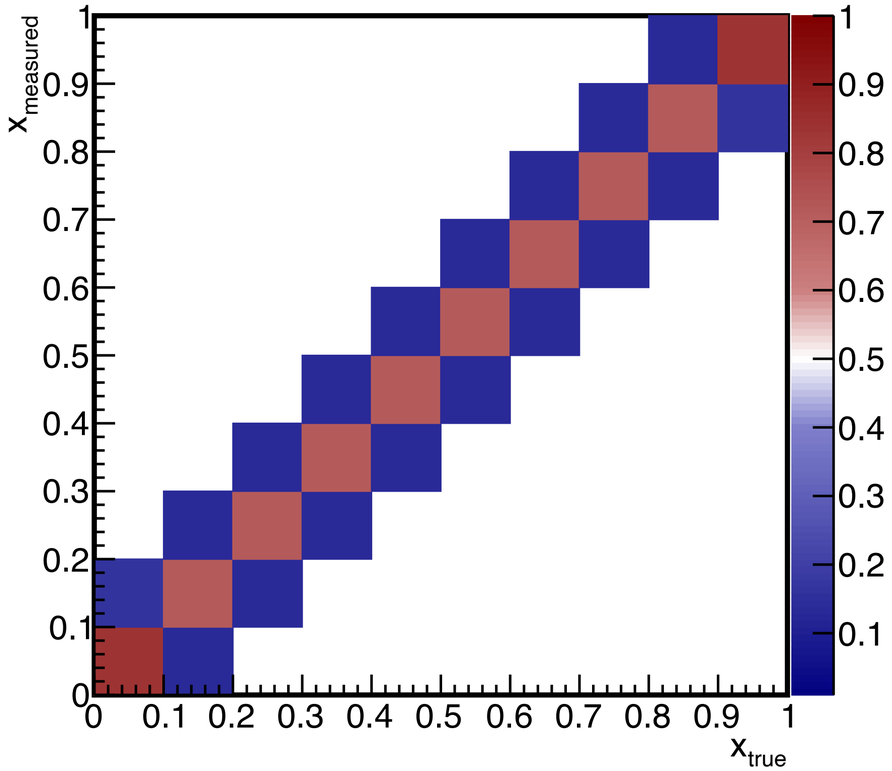
The bins are much smaller than the resolution here

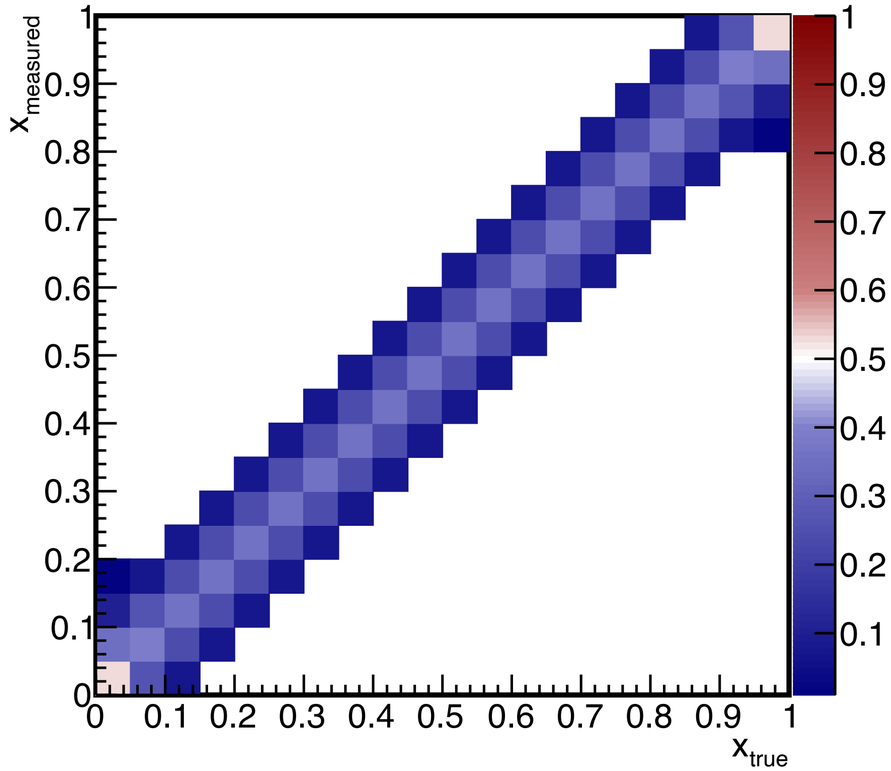
Example for comparison of bin width and resolution

- Same example as before
- Response is based on normal distribution with $\sigma = 0.055$
- Binning is 0.1 and 0.05
- We would expect that the second turns out to be a kind of ill-posed question
- However: Binning represents loss of information - so the smaller binning actually contains more information
- The actual effect requires a more subtle understanding



Response matrices for example

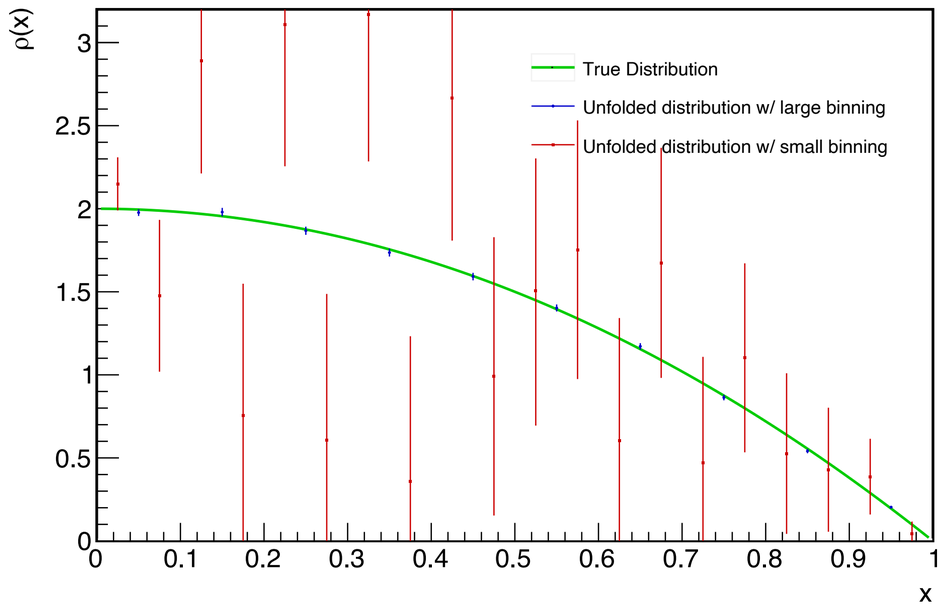




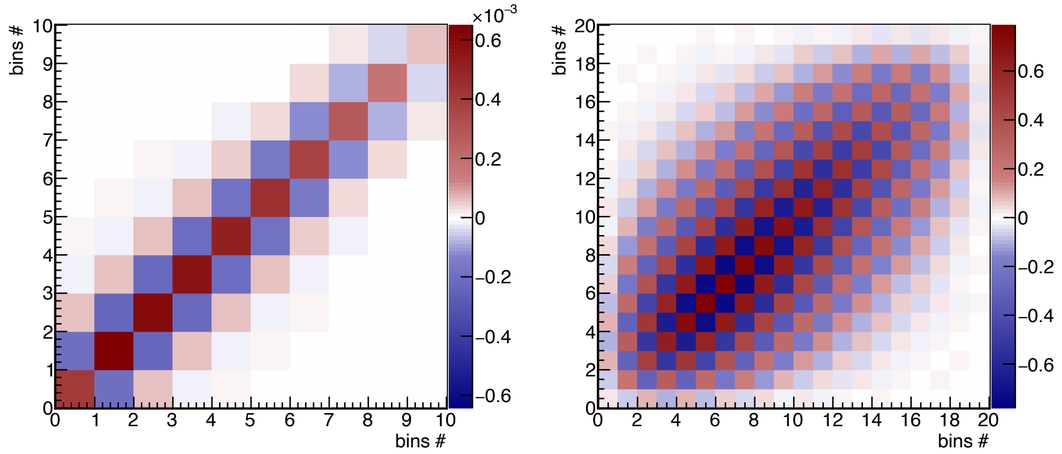
Large and small binning comparison

- As expected - large uncertainties for the small binning result
- But how is this possible? The smaller binning should contain more information!
- The plotted error bars are only the diagonal elements of the covariance matrix
- We can refold to get back the measured data - no information is lost in matrix inversion method
- How can we use the information in the covariance?

unfolded results

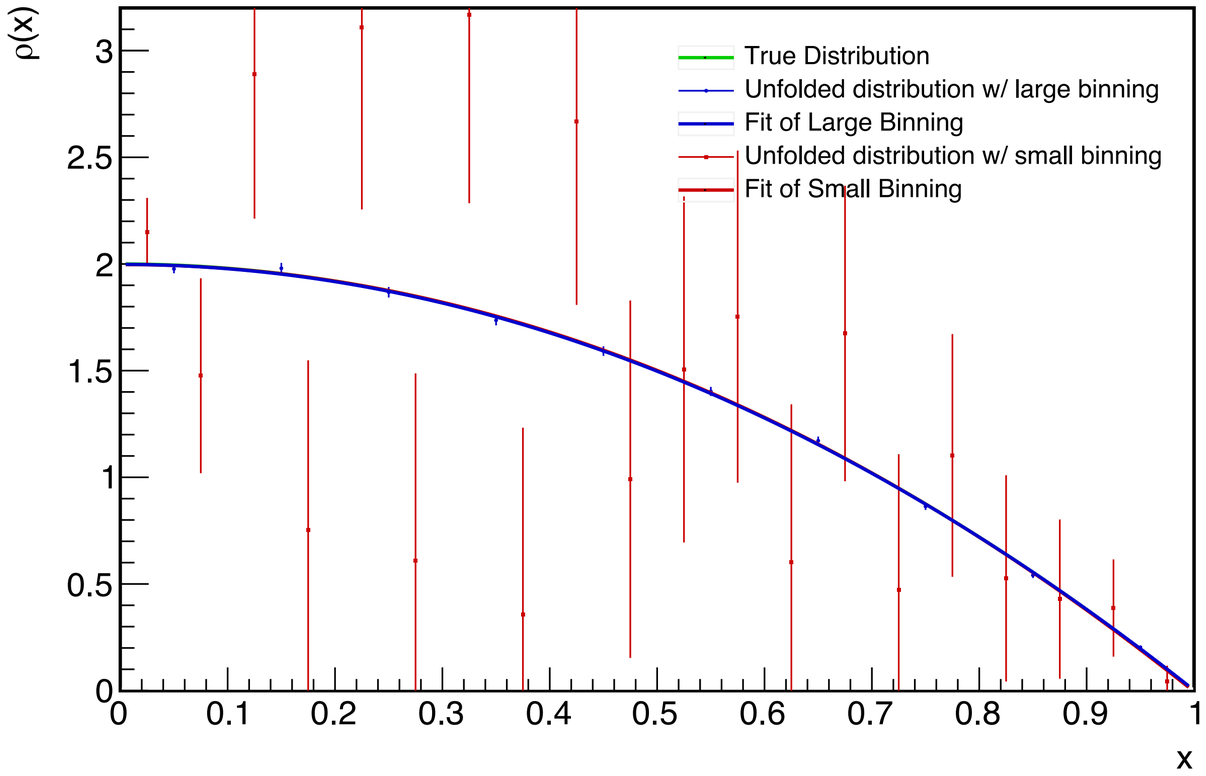


covariance matrices



Fit to unfolded results

- We learned how to fit with correlated uncertainties
- Fitting with use of the entire covariance matrix gives almost identical results
- For model comparison, the large diagonal elements in the covariance matrix do not matter!
- Thus, the problem exits only for visual presentation - the answer is fine in both cases



What's wrong with the matrix inversion method?

Unbiased, minimum variance, actually also a ML estimator … all very nice!

The result is not wrong, it is just not nice for plotting

- Does not really look like the original distribution
- Large correlation between bins - not visible in the error bars

"Applying the response matrix R smears out fine structure
→ applying R^{-1} creates (usually unwanted) structure"

More desirable solution by adding (smoothness) constraints.
However, this will produce a bias.

Depending on use case, find an acceptable balance between bias and smoothness.

To unfold or not to unfold?

[C. Pruneau,
Data Analysis Techniques for Physical Scientists]

From S. Oser's lecture:

> The most important advice I can give about deconvolution is “Don't”.
>
> It's a lot of work, and often produces biased or otherwise unsatisfactory results. Moreover it's often unnecessary.

"Forward fitting" is much easier

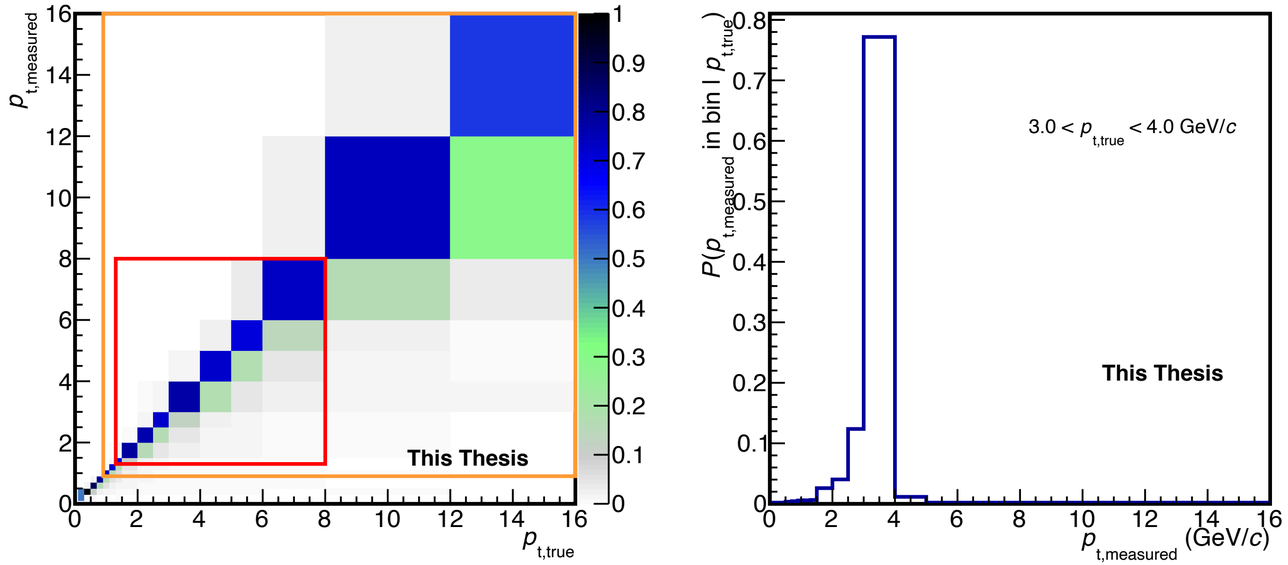
- Take theory prediction
- Convolve it with the response of the detector
- Compare smeared theory directly with the data

Approaching unfolding

1. Easiest case: No unfolding at all
   - Publish result on level of reconstructed quantities
   - Also publish response function
   - Comparison to models then by applying the response to them
2. Unfold using matrix inversion/least squares
   - Unbiased result, good for comparison to models
   - Need to publish covariance matrix
   - Good idea to consider binning to avoid making bins smaller than the resolution
   - Useful if applying response is not feasible for users (e.g very complex)
3. Regularised unfolding
   - Accept a bias in the result in exchange for smoothening the fluctuations
   - If this decreases correlations, the result is easier to understand by eye

Edge of the measured region

- Often times, measurements are only done in some range
- E.g. p_{T}, E, p_{jet} often have a minimum given by what the detector can measure, and a maximum because statistics run out
- However, the response connects regions outside the measured range
- This needs to be considered!
- In many cases, the contribution is small, e.g.:
  - Measure these regions with large uncertainties
  - Treat as systematic - assume probability distribution for content and marginalise
- The propagated uncertainty contribution will often be small in the measured region



The red region is the main one of the result. The adjacent bins were still measured, the contribution of the ones after that was treated as a systematic by checking the result of wildly varying contributions. (beauty hadron decay electron measurement)

Unfolding software: RooUnfold (ROOT Unfolding framework)

https://gitlab.cern.ch/RooUnfold/RooUnfold

RooUnfold is a framework for unfolding (AKA "deconvolution" or "unsmearing"). It currently implements six methods:

- iterative ("Bayesian"; as proposed by D'Agostini);
- singular value decomposition (SVD, as proposed by Höcker and Kartvelishvili and implemented in TSVDUnfold);
- bin-by-bin (simple correction factors);
- an interface to the TUnfold method developed by Stefan Schmitt;
- simple inversion of the response matrix without regularisation; and
- iterative dynamically stabilized unfolding (IDS) by Bogdan Malaescu, implemented by Chris Meyer.

It can be used from the ROOT prompt, from C++ (Cling) or Python (PyROOT) scripts, or linked against the ROOT libraries.

Bin-by-bin method (1)

Used very often, but has issues …

Assume shape of true spectrum and determine correction factor for each bin (usually determined from Monte Carlo simulation):

$$\mu_i = C_i(n_i - \beta_i) \qquad\qquad C_i = \frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}}$$

Works if smearing (bin-to-bin sharing) is negligible, only loss due to finite efficiency:

$$R_{ij} \approx \delta_{ij}\varepsilon_j$$

Obviously works, too, if MC = nature.

Expectation value for corrected data:

$$E[\hat{\mu}_i] = C_i E[n_i - \beta_i] = C_i(\nu_i - \beta_i) \equiv C_i \nu_i^{\mathrm{sig}}$$

Bin-by-bin method (2)

Inserting the C_i's one can determine the bias:

$$E[\hat{\mu}_i] = \frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}}\nu_i^{\mathrm{sig}} = \underbrace{\left(\frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}} - \frac{\mu_i}{\nu_i^{\mathrm{sig}}}\right)\nu_i^{\mathrm{sig}}}_{\mathrm{bias}} + \mu_i$$

no bias only if MC = nature

Covariance matrix of the corrected data (smearing fluctuations independent between bins)

$$U_{ij} = \mathrm{cov}[\hat{\mu}_\mathrm{i}, \hat{\mu}_\mathrm{j}] = C_i C_j \underbrace{\mathrm{cov}[n_i^{\mathrm{sig}}, n_j^{\mathrm{sig}}]}_{0 \text{ for } i \neq j} = C_i^2 \mathrm{Var}[n_i^{\mathrm{sig}}]\delta_{ij}$$

Iterative bin-by-by method

- Start with plausible guess of true spectrum
- Apply correction to measurement
- Generate new correction factors from corrected spectrum of previous iteration
- And so on … usually a few iterations sufficient

Regularized unfolding

Matrix inversion is the maximum likelihood solution:

Independent Poisson fluctuations:

$$\ln L(\vec{\mu}) = \sum_{i=1}^{M} (n_i \ln \nu_i - \nu_i)$$

ML estimator:

$$\hat{\vec{\nu}} = \vec{n}$$

$$\rightarrow \ \hat{\vec{\mu}} = R^{-1}(\vec{n} - \vec{\beta})$$

Idea: accept solutions that are close to maximum likelihood estimate:

$$\ln L(\vec{\mu}) \geq \ln L(\vec{\mu}_{\text{max}}) - \Delta \ln L(\vec{\mu})$$

Define a smoothness function S that gets bigger when the unfolded solution becomes smoother.

The task then is to maximize

$$\phi(\vec{\mu}) = \ln L(\vec{\mu}) + \tau S(\mu)$$

parameter that controls the strength go the regularization

smoothness function

Tikhonov regularization

Smoothness measure for the deconvoluted function *f*:

$$S[f] = -\int \mathrm{d}x \left(\frac{\mathrm{d}^k f}{\mathrm{d}^k x} \right)^2 \qquad k = 1, 2, 3, \ldots$$

Minus sign makes *S* small when *k*-th derivative is large

Tikhonov for $k = 2$:

numerical estimate of the second derivative:

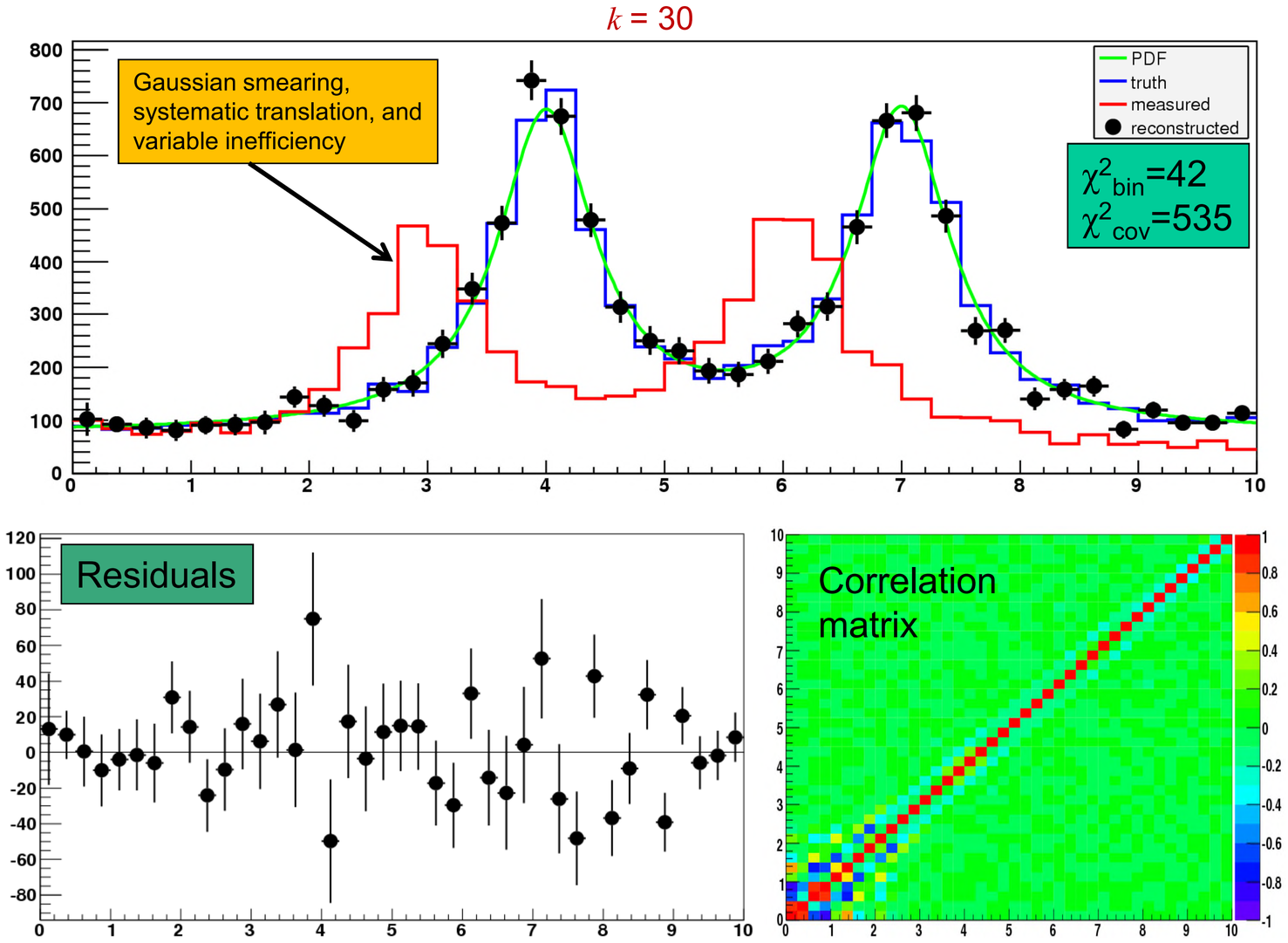
$$\phi(\vec{\mu}) = \ln L(\vec{\mu}) + \tau S(\mu), \qquad S(\vec{\mu}) = -\sum_{i=1}^{M-2} (-\mu_i + 2\mu_{i+1} - \mu_{i+2})^2$$

Implementation by A. Höcker, V. Kartvelishvili: *Singular Value Decomposition*
(NIM A372 (1996) 469, hep-ph/9509307, TSVDUnfold in ROOT)

Minimizes $\chi^2(\vec{\mu}) + \tau \sum_i [(\mu_{i+1} - \mu_i) - (\mu_i - \mu_{i-1})]^2$

- Introduces a penalty term for changes in the slope
- Possible interpretation: Prior information about the smoothness of the result

RooUnfold with SVD algorithm

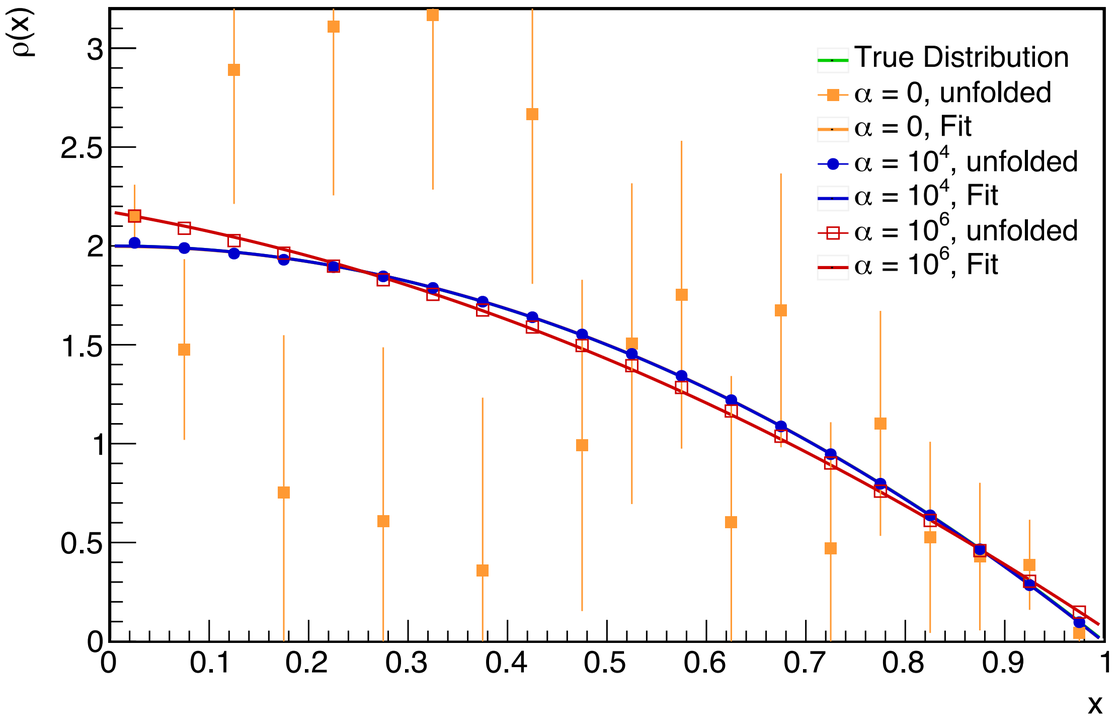


http://hepunx.rl.ac.uk/~adye/software/unfold/RooUnfold.html

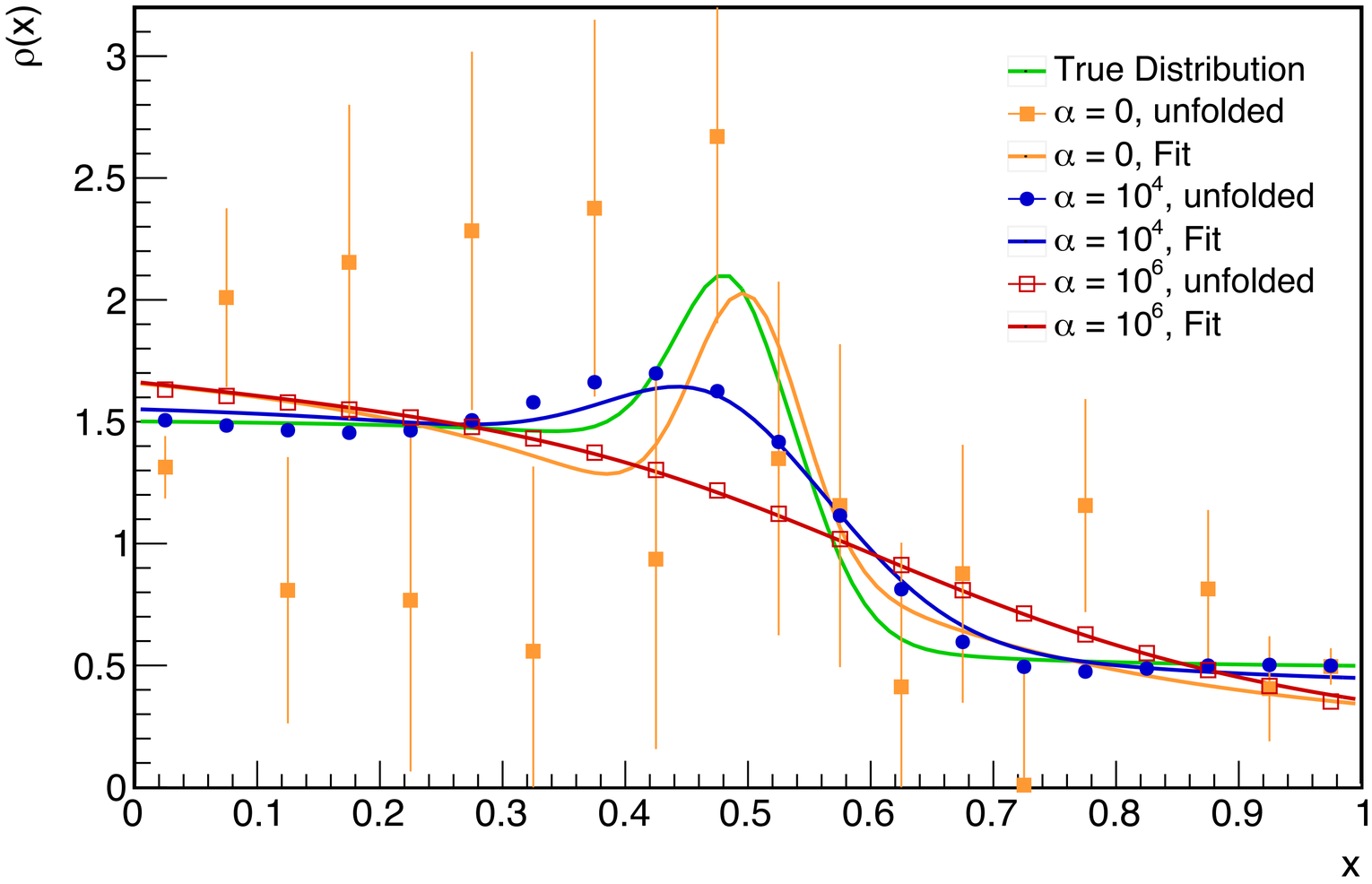
Tim Adye, Unfolding in HEP, https://indico.cern.ch/event/671301/contributions/2745801

Regularisation strength

- Varying the refactor of the penalty term changes the regularisation strength
- Structure in true distribution will be washed out
  - Dangerous for distributions with a lot of structure (e.g. peaks)
- Due to the additional introduced information, fluctuations can be smaller than $\sqrt{N}$
- Think about what the actual prior information is
- The penalty term for each bin should depend on the distance to the adjacent bins - consider for unequal bins
- Bias depends on the true distribution



Another example



Here the regularisation increases the bias more quickly

Regularised unfolding algorithms

- Maximum likelihood with Tychonov regularisation
  - Real number for regularisation strength
  - Include prior information about smoothness this way
- Singular Value Decomposition
  - Used as a way to implement the previous for Gaussian uncertainties
- Bin-by-bin unfolding
  - Simplest; assumes that the simulated distribution is similar to the true one
- Iterative (“Bayesian”) unfolding
  - Uses an iterative method, starting at smooth distribution, converging to unregularised
  - Regularisation by stopping before convergence - unclear what prior information this would correspond to
  - Computationally efficient (compared to inverting large matrices)
  - Sometimes additional smoothing is done within the steps

Regularization based on entropy

Shannon entropy of a discrete probably distribution:

$$H = -\sum_{i=1}^{M} p_i \ln p_i$$

all p_i equal → maximum entropy (max. smoothness)

$p_i = 1$, all others 0 → minimum entropy

Use entropy as regularization function:

$$S(\vec{\mu}) = H(\vec{\mu}) = -\sum_{i=1}^{M} \frac{\mu_i}{\mu_{\text{tot}}} \ln \frac{\mu_i}{\mu_{\text{tot}}}$$

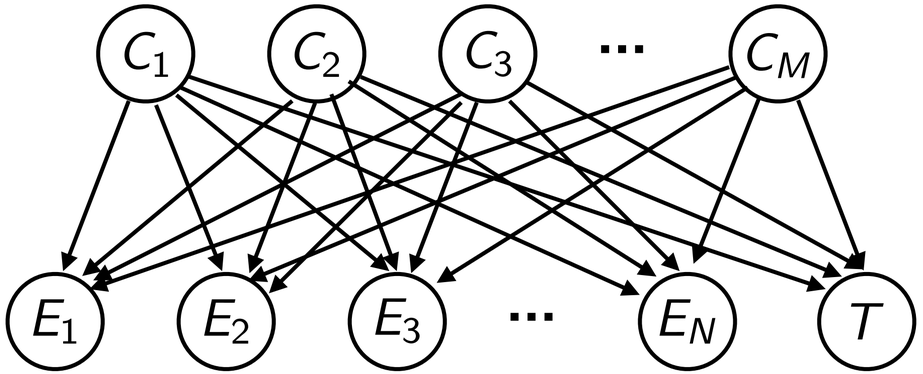
This gives the distribution with the maximum entropy consistent within the selected tolerance with the ML solution

Entropy related to number of different ways μ_{tot} objects can be distributed to obtains histogram $\vec{\mu} = (\mu_1, \ldots, \mu_M)$

$$\Omega(\vec{\mu}) = \frac{\mu_{\text{tot}}!}{\mu_1! \cdot \ldots \cdot \mu_M!} \quad \overset{\text{Stirling's approximation}}{\rightsquigarrow} \quad \ln \Omega(\vec{\mu}) \approx \mu_{\text{tot}} H(\vec{\mu})$$

Iterative unfolding (1) [a.k.a. “Bayesian” unfolding]

Causal network:



M causes

N effects

trash bin for
undetected events

Response matrix:

I: prior knowledge about probabilities of the causes C_i

$$R_{ji} = P(E_j|C_i, I)$$

$$E[n_j|\mu_i] = P(E_j|C_i, I) \cdot \mu_i = R_{ji} \cdot \mu_i$$

Iterative unfolding (2)

Bayes theorem:

$$P(C_i|E_j, I) = \frac{P(E_j|C_i, I) \cdot P(C_i|I)}{\sum_{k=1}^{M} P(E_j|C_k, I) \cdot P(C_k|I)}$$

We can write this as

$$\theta_{ij} := P(C_i|E_j, I) = \frac{R_{ji} \cdot P(C_i|I)}{\sum_{k=1}^{M} R_{jk} \cdot P(C_k|I)}$$

Estimate for number of true events in bin i given that we measure n_j events in bin j:

$$\mu_i|_{n_j} = \frac{P(C_i|E_j, I) \cdot n_j}{\varepsilon_i} = \frac{\theta_{ij} \cdot n_j}{\varepsilon_i}$$

Iterative unfolding (3)

Summing over all observed bins:

$$\mu_i|_{\vec{n}} = \frac{1}{\varepsilon_i} \sum_{j=1}^{N} \theta_{ij} \cdot n_j$$

By definition we can write the efficiency as

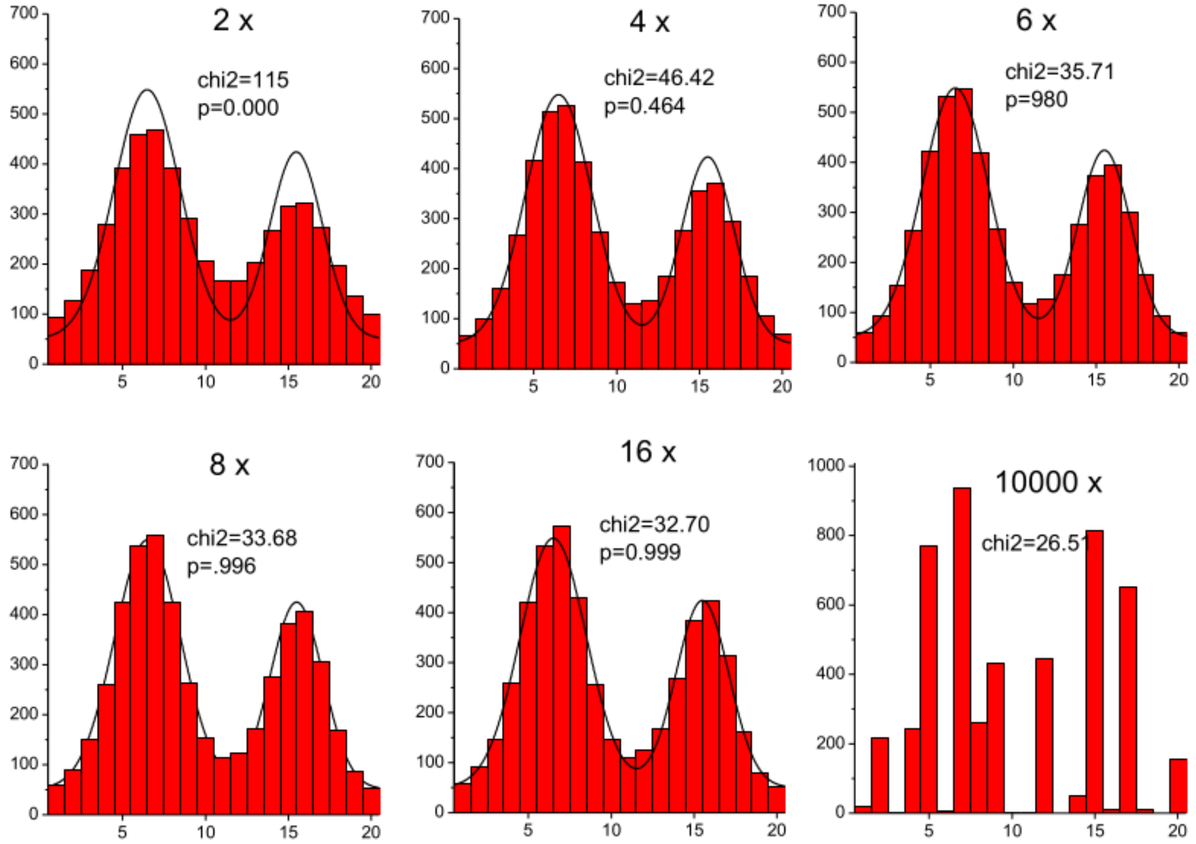
$$\varepsilon_i = \sum_{j=1}^{N} P(E_j|C_i, I) = \sum_{j=1}^{N} R_{ji}$$

Response matrix usually from Monte Carlo simulation

Iterative unfolding (4)

This procedure is applied iteratively:

- Choose prior distribution $P(C_i,I)$
- Often prior = measured distribution
- Update prior according to measured values
- iterate
- Converges to unregularised case
- Limited number of iterations provides implicit regularization



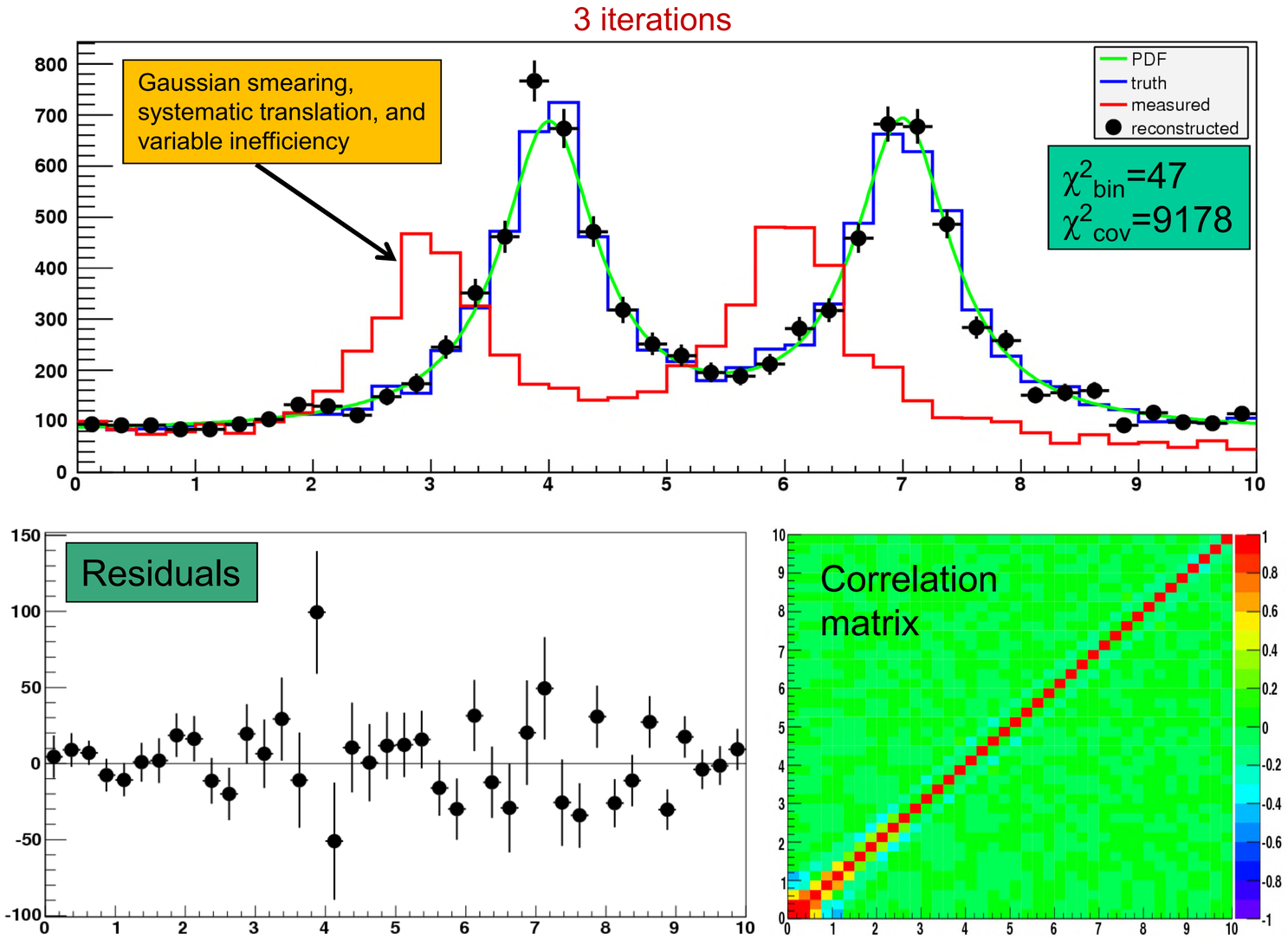
From talk by Günter Zech 2011

Shepp/Vardi 1982, Mülthei/Schorr 1986

G. D'Agostini, A Multidimensional unfolding method based on Bayes' theorem'', Nucl. Instrum. Meth. A 362 (1995) 487 (see also arXiv:1010.0632)

V. Blobel, Unfolding methods in high-energy physics experiments, https://cds.cern.ch/record/157405

RooUnfold with iterative Bayesian algorithm



http://hepunx.rl.ac.uk/~adye/software/unfold/RooUnfold.html

Tim Adye, Unfolding in HEP, https://indico.cern.ch/event/671301/contributions/2745801

Summary

- Measurement “folds” true distribution with response matrix
- Unfolding inverts this step
- Sometimes easier to skip and publish only measured distribution + response
- Unbiased result from finding minimum χ^2 solution to folding equation; equivalent to inverting response matrix
- If response is wide compared to binning, individual bins have large uncertainties and strong correlations
- For nicer plots: Regularisation methods
  - Essentially implement prior knowledge about smoothness, but may also be ad hoc
  - Tychonov regularisation makes this explicit, while e.g. iterative unfolding leaves this very vague
- Regularisation makes the result easier to interpret, but introduces a bias
  - Make sure the added information about smoothness is information you actually have

Some more practical tips

- Some problems are phrased as statistics problems, but are actually about physics
  - Which model should we use to describe the detector/Is the background linear or exponential?
  - How do we combine the different models into an uncertainty/what is their relative status?
  - Which systematics do we need to check?
- For Bayesians:
  - Some frequentist methods are quite useful e.g. χ^2/n_{dof} tests for quickly reject a model or to find systematic uncertainties
  - Much of particle physics is done within the frequentist paradigm, so knowing the methods is not optional
- For Frequentists:
  - For many systematic uncertainties, it is very natural to think of them in a Bayesian way (e.g. theory uncertainties) - even if you do the rest of the analysis in a frequentist approach
- Whenever someone proposes an (ad hoc) approach to a problem, it can often be useful to ask, which fundamental approach it is supposed to approximate - this helps avoid bad ideas

Goodbye

martin.andreas.volkl@cern.ch