Progress in Unfolding the Photon Structure Function F_2^γ

Edmund Clay

February 23, 1999

Abstract

Three programs to be used to unfold the photon structure function, F_2^γ are described: two regularised unfolding programs and one based on Bayes' theorem. Problems with the unfolding are highlighted. Two-dimensional unfolding is presented as a possible way to reduce model dependence in the unfolding result.

1 Introduction

The substructure of the photon can be studied in a similar way to that of the proton; by deep inelastic scattering of electrons. The case of the photon is particularly interesting because it can interact as a hadron or as a gauge boson, thus it gives a unique opportunity to examine the interface between perturbative and non-perturbative QCD [1].

At LEP, two photon events in which one photon (the 'target' photon) is nearly real can be selected by requiring that one electron is seen ('tagged') in the detector, while the other is not. The photon structure function $F_2^\gamma(x,Q^2)$ is related to the differential cross section for these events with respect to the invariants $Q^2 = -q^2$ and $x = \frac{Q^2}{p.q}$ where q is the four momentum of the virtual probe photon and p is that of the nearly real target photon.

x and Q^2 can be determined from the event kinematics using

$$Q^2 = 2E_b E_{\mathrm{tag}}(1 - \cos\theta_{\mathrm{tag}}) \tag{1}$$

and

$$x = \frac{Q^2}{Q^2 + W^2 + P^2}. \tag{2}$$

E_b is the beam energy and E_{tag} and θ_{tag} are the energy and polar angle of the tagged electron. W is the invariant mass of the hadronic final state and P^2 is the negative four momentum squared of the target photon, which is assumed to be small.

While it is possible to measure Q^2 to an accuracy of a few percent, measuring x is more difficult. This is because the electron producing the target photon is not seen, which makes it necessary to determine W from the hadronic final state. Because of losses in the forward region of the OPAL detector W cannot be measured accurately. To correct for the resulting poor resolution in x, the experimental distribution must be *unfolded.* Unfolding is a statistical technique described in more detail in the next section. It relies on good Monte Carlo simulation for accurate results. It has been demonstrated [2] that none of the present Monte Carlo models provide an adequate description of the experimental data, as a result of which previous OPAL measurements of F_2^γ show large systematic errors [3] [4]. In this report I present a study of new unfolding methods, with the aim of reducing modelling errors.

2 Unfolding Programs

Unfolding is now a standard technique in experimental particle physics, and there are several unfolding programs available. Previous OPAL analyses have used only the program RUN by Volker Blobel [5]. I have looked at two other programs to see if there were any differences in their behaviour compared to RUN: GURU, a regularised unfolding program using matrix methods by V.Kartvelishvili [6] and a method based on Bayes' theorem by G.D'Agostini [7].

2.1 The Unfolding Problem

A measurement $g(y)$ is made of a true distribution $f(x)$. The relationship between these distributions can be expressed in terms of a response function

$A(x,y)$;

$$g(y) = \int A(x,y) f(x)\, dx + e(y) \tag{3}$$

where $e(y)$ are statistical errors. There may also be a background contribution which is subtracted from the data. The response function can take into account both limited resolution, through which the true value x is transformed into a measured value y, and limited acceptance, such that some events are lost altogether.

In order to solve for real data the equation must be discretised:

$$g_j = \sum_{i=1}^{m} A_{ij} f_i + e_j. \tag{4}$$

Or equivalently, put into matrix form,

$$g = Af + e. \tag{5}$$

Here g, f and e represent histograms of the measured distribution, true distribution and errors respectively. The response matrix A is an $n \times m$ matrix which contains values between zero and one, representing the probability of an event in the ith bin of the true distribution ending up in the jth bin of the measured distribution. The problem is to extract the best estimate for f given g and A. A is usually found from a detailed simulation of the detector. Note that the errors e are only known statistically. This is what makes unfolding a statistical problem and not a purely algebraic one.

If the statistical errors e can be neglected and the inverse matrix A^{-1} exists, then

$$f = A^{-1} g. \tag{6}$$

This is perhaps the most obvious way to recover f. However, in practice it leads to unstable, oscillating solutions. The intuitive explanation for this behaviour involves looking at the effect of multiplying f by the matrix A^{-1}. If A is a smearing matrix, which removes high frequency components from the distribution, then A^{-1} has to put them back. But after having been smeared, the high frequency components are a small effect, easily swamped by statistical errors. So the inverse response matrix puts back a completely different set of high frequency components to those in the original distribution, making the result useless. Regularised unfolding programs reduce high frequency components by introducing a bias towards smooth solutions.

2.2 Regularised unfolding using B-splines (RUN)

RUN uses a maximum likelihood fit for unfolding. Included in the fit is a regularisation term to smooth unwanted oscillations.

The negative log likelihood function $S(f)$ is defined as

$$S(f) = -\sum_{j=1}^{n} \ln P(g_j|\hat{g}_j) \tag{7}$$

where $P(g_j|\hat{g}_j)$ is the initial probability of observing g_j events in the jth bin given an expectation of $\hat{g}_j = \sum_{i=1}^{m} A_{ij} f_i$. It is given by the Poisson distribution,

$$P(g_j|\hat{g}_j) = e^{\hat{g}_j} \frac{(\hat{g}_j)^{g_j}}{(g_j)!}. \tag{8}$$

$S(f)$ is minimised using an iterative procedure to give the unfolding result. RUN parameterises the data in terms of B-splines in order to perform the fit.

For regularisation another term is required:

$$r(f) = \int [f''(x)]^2 \, dx \tag{9}$$

where f'' is the second derivative of f. This is combined with $S(f)$ to give a new function to be minimised,

$$R(f) = S(f) + \tau r(f). \tag{10}$$

The factor τ is called the regularisation parameter. It controls the amount of smoothing; if $\tau = 0$ then there is no smoothing, if $\tau \to \infty$ then the fit is linear. The user has to choose τ so that spurious oscillating components are eliminated without overly biasing the result.

2.3 Regularised unfolding using Singular Value Decomposition (GURU)

The singular value decomposition of an $m \times n$ matrix A is the factorisation

$$A = USV^T \tag{11}$$

$U^T = U^{-1}, V^T = V^{-1}$ and $S_{ij} = 0$ for $i \neq j, S_{ii} \geq 0$.

It is useful in many problems involving matrices, and can be applied to unfolding.

Rather than maximum likelihood (as in RUN), GURU uses a least squares fit of the form

$$\sum_{j=1}^{n} \left(\frac{\sum_{i=1}^{m} A_{ij} f_i - g_j}{\hat{e}_j}\right)^2 = \min \tag{12}$$

where $\hat{e}_j$ is the error on the jth measurement. The general case can be written as

$$(Af - g)^T B^{-1} (Af - g) = \min \tag{13}$$

where B is the covariance matrix of the data g. B can be decomposed using SVD to rewrite this equation in a form that is easier to work with:

$$(A'f - g')^T (A'f - g') = \min. \tag{14}$$

Here A and b and B have been rotated and rescaled into A', b' and B', such that B' becomes equal to the unit matrix.

A regularisation term can be included in a similar way to the B-spline method,

$$(A'f - g')^T (A'f - g') + \tau (Cf)^T (Cf) = \min. \tag{15}$$

τ is the regularisation parameter as described in the previous section and C is a matrix giving the degree of curvature of f.

In one dimension a possible definition of curvature is

$$r = \sum_{i=1}^{n} [(f_{i+1} - f_i) - (f_i - f_{i-1})]^2 \tag{16}$$

and the corresponding form of C is

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & \ldots & & \\ 1 & -2 & 1 & 0 & \ldots & & \\ 0 & 1 & -2 & 1 & \ldots & & \\ & \ldots & & & & & \\ & \ldots & & 1 & -2 & 1 & \\ & \ldots & & & 1 & -1 & \end{pmatrix}. \tag{17}$$

2.4 Bayesian Unfolding

Bayes' theorem can be stated as:

$$P(\text{cause}|\text{effect}) \propto P(\text{effect}|\text{cause}) \times P_0(\text{cause}). \tag{18}$$

When applied to unfolding a 'cause' is an event in a certain bin of the true distribution, and an 'effect' is a event in another bin of the measured distribution. $P(\text{effect}|\text{cause})$ is the analogue of the response matrix in regularised unfolding. $P_0(\text{cause})$ is the initial probability of the cause, which can be estimated from Monte Carlo simulation. Although the output is independent of $P_0(\text{cause})$, if the initial estimate is poor the method takes longer to converge.

The normalised form of Bayes' theorem is

$$P(C_i|E_j) = \frac{P(E_j|C_i)P_0(C_i)}{\sum_{k=1}^{n} P(E_j|C_k)P_0(C_k)} \tag{19}$$

with the normalisation condition $\sum_{i=1}^{n} P(C_i|E_j) = 1$.

After N experimental observations one has a distribution of events, with the number of events in the jth bin $n(E_j)$. Events can then be assigned to causes:

$$n'(C_i) = \frac{1}{\epsilon_i} \sum_{j=1}^{m} n(E_j) P(C_i|E_j) \tag{20}$$

where E_i is the detector efficiency in the ith bin, obtained by simulation. The total number of events is

$$N' = \sum_{i=1}^{n} n'(C_i). \tag{21}$$

And the updated probabilities are given by

$$P'(C_i) = \frac{n'(C_i)}{N'}. \tag{22}$$

The probabilities $P'(C_i)$ can then be put into equation (19) as an improved guess, and the procedure repeated until some criterion is satisfied, e.g. that χ^2 between two iterations is less than a certain value.

2.5 Comparison of Unfolding Programs

The programs RUN and GURU are both programs for regularised unfolding and are very similar in principle; it is therefore not surprising that they give similar output. There is a difference in the type of regularisation; RUN smoothes the shape of the unfolded function and so will tend to give a smoother curve than reality, while GURU smoothes the difference between the input Monte Carlo function and the unfolded function, so it allows for rapidly changing functions as long as the Monte Carlo mirrors that behaviour. This may be important at low x, if there is a rise in the structure function in that region. Of course if the Monte Carlo is very different from the data there will be additional bias in the GURU method. There is also some difference in the way I treated the two programs: while RUN is a 'black box' program with a complex internal structure, GURU uses standard matrix techniques to do most of the unfolding work, so it is relatively easy to follow the code. This made it possible to adapt GURU to perform two dimensional unfolding simply by changing the curvature matrix.

Another difference between RUN and GURU is a potential source of error: RUN uses a probability matrix for the detector response, while GURU uses a number-of-events matrix. The difference is not just one of normalisation; RUN can allow for the possibility that some events came from outside the range of the true variable considered in the unfolding, while GURU cannot. It is therefore important to include the full range of the true variable when unfolding with GURU.

The Bayes unfolding program works in a rather different way to the other two programs. Without regularisation it is equivalent to them (there is only one solution to the problem) but unlike RUN and GURU it will not do any regularisation itself; this is left to the user. While there are some advantages to this approach, such as the ability to fit to any desired function while unfolding, it is impossible to balance smoothness against bias through the use of a regularisation parameter. Also, as the smoothing is not an integral part of the unfolding process, the errors do not take it into account. After smoothing, one expects smaller, but more correlated errors. Figure (figure 1) shows an example of this. When unfolding a slanted distribution, although all three programs return the correct shape, the errors from Bayes' are smaller at the low end, while the errors from RUN and GURU are effectively constant over the whole range. The main advantage of the Bayes program is that it can be used for two dimensional unfolding with no alterations, although then it is necessary to smooth in two dimensions also.

RUN is the most comprehensive of the three programs, and also the most stable when changing parameters such as the number of bins and regularisation parameter (this could be due to the use of spline functions to parameterise the input distributions - in effect, a pre-smoothing of the data). I encountered problems only when unfolding with limited Monte Carlo statistics. With all three unfolding programs, it is important to have much more Monte Carlo than data, about 10 times as much, as they neglect the statistics of the Monte Carlo sample in the errors given.

To test the unfolding programs, I used all three for the same problem: unfolding of data with the HERWIG Monte Carlo. There is no significant difference between the three curves, and the errors are also consistent, even from Bayes since in this case the function is fairly flat (figure 2). For subsequent unfoldings, I show only the result from GURU.

3 Unfolding Results

Using the three programs described above, I unfolded OPAL data taken at a centre of mass energy of 172 GeV and average $Q^2 = 41$ GeV^2. I see the same features as the previous analysis [3]:

- Using the same generator (HERWIG [8]) as mock data and Monte Carlo, but with a different parameterisation of F_2^γ in each (SAS1D [12] for the mock data, GRV [11] for the Monte Carlo), it is possible to recover F_2^γ over most of the x range. There is still a problem for $x < 0.1$ (figure 3a).
- Using a different generator (PYTHIA [9] with HERWIG), there is only agreement in the central region of x, and large differences at both high x and low x (this is the source of the large systematic errors in the F_2^γmeasurement) (figure 3b).
- The results of unfolding the data with HERWIG and PYTHIA are inconsistent at both high x and low x. (figure 4).

I also did the unfolding based only on number of events, without using the structure function, and found that there is only disagreement between

HERWIG and PYTHIA at low x (figure 5). The problems at high x arise from differences between the true x distributions. Since unfolding deals only with the transition from true to measured variables, any difference on the level of true variables cannot be corrected by an unfolding program. At low x the problem is quite different. Here, although true x agrees, the measured value x_{vis} does not, indicating a problem in the final state modelling.

4 Monte Carlo Differences

The stages in a Monte Carlo model are roughly as follows: the generator produces a two photon state, then accepts or rejects events using a given structure function, based on the values of x and Q^2 for each event. It makes some cuts on Q^2 and W before generating the final state. The events are smeared, passed through a detector simulation and reconstructed to give the final measured value x_{vis}. Thus the distribution of true x depends on the initial two photon state, the structure function and generator level cuts. The x_{vis} distribution depends on the true x distribution and final state model; everything else is the same for all the Monte Carlos, and relatively reliable.

In all the Monte Carlos (HERWIG, PYTHIA, F2GEN [10]), the Q^2 distribution for the virtuality of the probe photon is consistent with the data sample, but the virtuality of the target photon is calculated using the equivalent photon approximation [13] and is not treated the same way in HERWIG and PYTHIA. There is also a W cut to eliminate charm events below the charm threshold, but HERWIG and PYTHIA use different charm masses. This leads to a different x distribution. Both the EPA and the charm mass affect mostly the high x region. When unfolding, the assumption is made that a given F_2^γ will give the same true x distribution in Monte Carlo and data. There is no information directly about F_2^γ in the unfolding problem, therefore any discrepancies must be corrected in the Monte Carlos. However, unfolding can in principle deal with modelling differences as long as there is another measured variable that can be used to distinguish between the models. This requires multi-dimensional unfolding.

Since HERWIG uses F_2^γ in generating parton showers as well as in calculating the cross section, there is some model dependence involved in the choice of F_2^γ. To investigate this it would be necessary to separate out the F_2^γ used for the x and Q^2 distributions from F_2^γ used elsewhere. This remains to be studied.

5 2D Unfolding

It has been suggested [14] that if the x_{vis} distribution depends not only on true x but also on the final state model, then it may be possible to find a variable related to the model and unfold in that variable as well as x. Then the solution will be independent of both the true x distribution used in the Monte Carlo and also to a certain extent the choice of Monte Carlo.

Full 2D unfolding requires two variables both on generator level and in the final data. It is also possible to unfold with only one generator level variable and two measured variables. This is an option in the program RUN. The two types of 2D unfolding are not quite the same. The difference is in the regularisation. In the two measured, one true variable case, events close together in x will have similar weighting, regardless of how they differ in any other variable. With two true variables only events close in both variables will be smoothed together, giving the unfolding program more freedom.

Candidates for the second unfolding variable are the transverse energy of the hadronic system out of the tag plane ($E_{\mathrm{t,out}}$), the polar angle of the thrust axis (θ_{thr}) and the number of jets (N_{jet}). All of these are some way related to the ability of the detector to measure x, so if they are different in two models, the x distributions will also be different.

One way to identify suitable variables for 2D unfolding is to look at the distribution of x_{vis} for restricted ranges of true x and the second variable. If the plots have the same shape for two different models, then an unfolding in these two variables will be model independent, because the response matrix will be the same for both models. Figure 6 shows a comparison of HERWIG, F2GEN and PYTHIA for $x < 0.1$ in bins of $E_{\mathrm{t,out}}$. While the overall x_{vis} distributions look rather different for the three Monte Carlos, when separated into bins of the second variable the plots become more similar, particularly for F2GEN and PYTHIA. This indicates that 2D unfolding with $E_{\mathrm{t,out}}$ may be useful for these two Monte Carlos

The same comparison using N_{jet} (figure 7) also shows improved agreement using the second variable, but to a lesser extent. However, θ_{thr} does not help at all (figure 8).

A useful test of 2D unfolding is to unfold one Monte Carlo against another with a different final state model. I did this with F2GEN as mock data and PYTHIA as the reference Monte Carlo, using GURU as the unfolding program (figure 9). The 2D result is significantly closer to the true distribution of x in F2GEN. The same test with HERWIG as mock data is shows less improvement with 2D unfolding. It seems that the $E_{\rm t,out}$ distributions in HERWIG and PYTHIA are too similar to account for all the difference in the $x_{\rm vis}$ distributions.

Figure 10 shows 2D unfoldings of the OPAL data with HERWIG and PYTHIA. The scale for these plots is $\log x$ because it is for the low x region that 2D unfolding is most useful. As a test of the 2D unfolding I ran it with a random number as the second variable. Since there is no extra information compared to the 1D problem the result should not be very different. Figure 10c shows that this is indeed the case. The agreement between the Monte Carlos is significantly improved when using $E_{\rm t,out}$ as the second variable. There was also noticeable improvement using $N_{\rm jet}$.

6 Conclusions

To study the photon structure function at LEP2, it is necessary to unfold the data, because the total hadronic energy is poorly measured. Several unfolding programs are available, using different unfolding methods, but all of those tested give consistent results. It should not be necessary to include any systematic error in F_2^γ due to the choice of unfolding method, but as any regularised unfolding program introduces bias, there will be some uncertainty from the choice of regularisation parameter.

Most of the systematic error in previous OPAL measurements of F_2^γ comes from the model dependence. There are very large differences between the unfolding results using different Monte Carlos, and also differences when using the same Monte Carlo with a different input F_2^γ. There are two separate sources of discrepancy. Firstly, the same input F_2^γ in different Monte Carlos gives a different distribution of true x. This could be due to the EPA for the target photon, generator level cuts, or both. Secondly, even if the true x distributions in two Monte Carlos are the same, the $x_{\rm vis}$ distributions may be different. This is a problem with the angular distribution of the hadrons in the final state. The first problem can only be solved by improving the Monte Carlos. The second can be approached either by modifying the

Monte Carlos or using multidimensional unfolding. 2D unfolding with $E_{\mathrm{t,out}}$ as the second variable shows a considerable improvement in the agreement between HERWIG and PYTHIA when used to unfold the same data sample, compared to 1D unfolding. With the other variables used, θ_{thr} and N_{jet}, there was less improvement. 2D unfolding also proved useful when unfolding F2GEN with PYTHIA, but no firm conclusions can be drawn from this because the unfolding of HERWIG with PYTHIA was not strongly affected by the introduction of a second variable.

Besides the possibility of providing a model-independent measurement of F_2^γ, 2D unfolding returns information about the distribution of the second variable in the data (independently of x), which could be used to improve the Monte Carlos.

If the problems with the Monte Carlos can be solved, and 2D unfolding reduces modelling uncertainty sufficiently, it may be possible with the present data to distinguish between different parameterisations of F_2^γ and to probe low enough in x to see a rise in the structure function.

References

[1] E.Witten, Nucl. Phys. B120 (1977) 189-202.

[2] OPAL Coll., K.Ackerstaff *et al.*, Zeit. fur Physik C74 (1997) 33-48.

[3] OPAL Coll., K.Ackerstaff *et al.*, Phys. Lett. B412 (1997) 225-234.

[4] OPAL Coll., K.Ackerstaff *et al.*, Phys. Lett. B411 (1997) 387-401.

[5] V.Blobel, *Regularised Unfolding for High-Energy Physics Experiments* OPAL Technical Note TN361 (1996).

[6] A. Höcker, V. Kartvelishvili, Nucl. Instr. and Meth. A372 (1996) 469-481.

[7] G. D'Agostini, *A Multidimensional Unfolding Method Based on Bayes' Theorem* DESY 94-099 (1994).

[8] G.Marchesini *et al.*, Comp. Phys. Comm. 67 (1992) 465.

[9] T.Sjöstrand, Comp. Phys. Comm. 82 (1994) 74-89.

[10] A.Buijs *et al.*, Comp. Phys. Comm. 79 (1994) 523-532.

[11] M.Glück, E.Reya and A.Vogt, Phys. Rev. D46 (1995) 1973;
M.Glück, E.Reya and A.Vogt, Phys. Rev. D45 (1996) 3986.

[12] G.A.Schuler and T.Sjöstrand, Z. Phys. C68 (1995) 607-623.

[13] C. F. von Weizsäcker, Z.Phys. 88 (1934) 612;
E.J. Williams, Phys.Rev. 45 (1934) 729;
V.M.Budnev *et al.*, Phys. Rep. 15 (1975) 181.

[14] S.Cartwright *et al.*, RAL-TR-97-042 (1997).

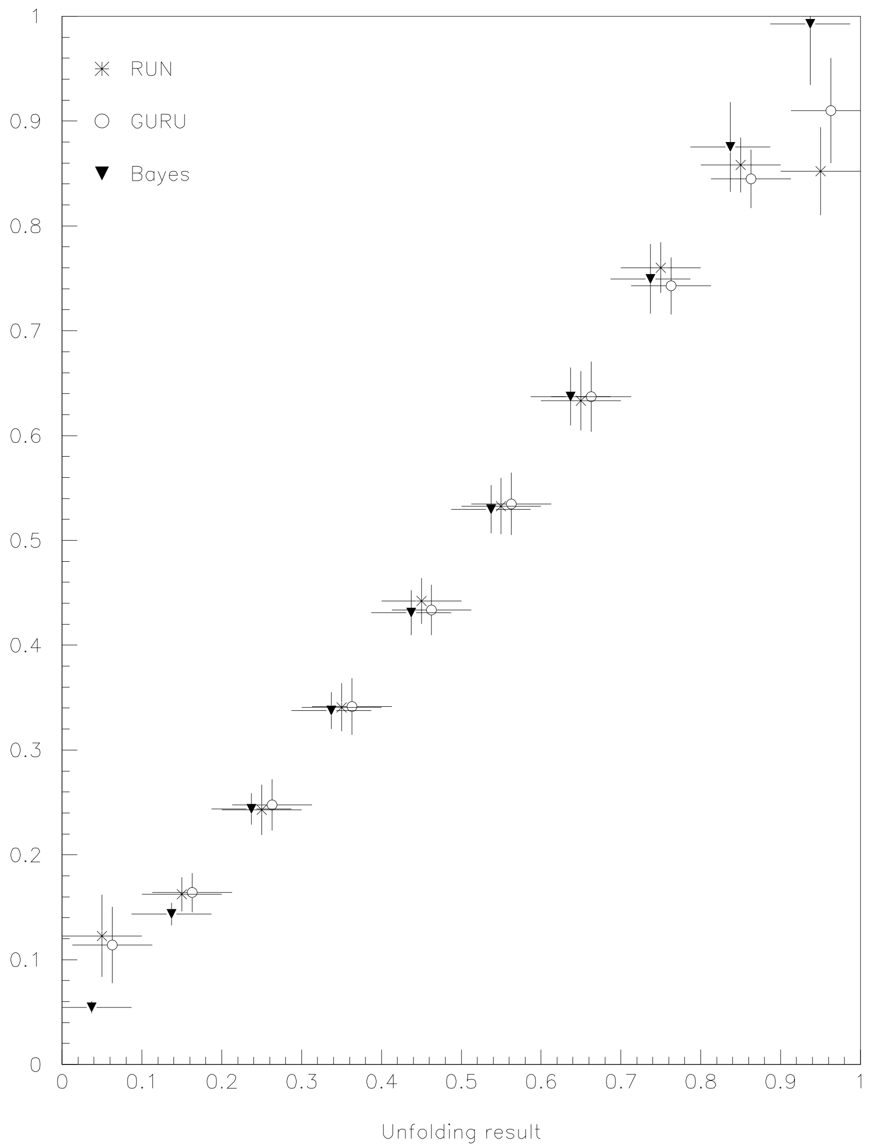

Figure 1: A simple test of the unfolding programs. The Monte Carlo used to unfold had a flat distribution, while the data had a slanted distribution.

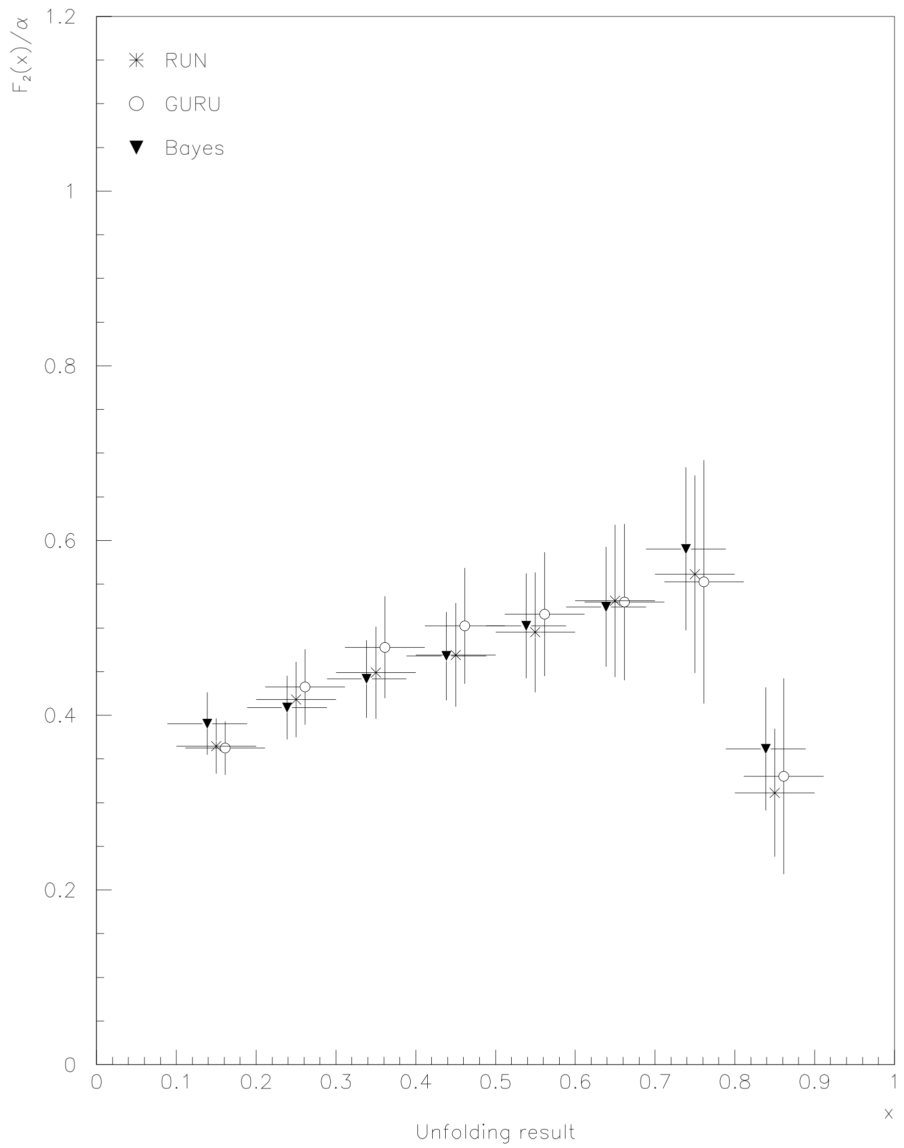

Figure 2: Comparison of the unfolding programs using data with $\sqrt{s} = 172$ GeV, average $Q^2 = 41\mathrm{GeV}^2$. The Monte Carlo used was HERWIG with the GRV parameterisation of F_2^γ.

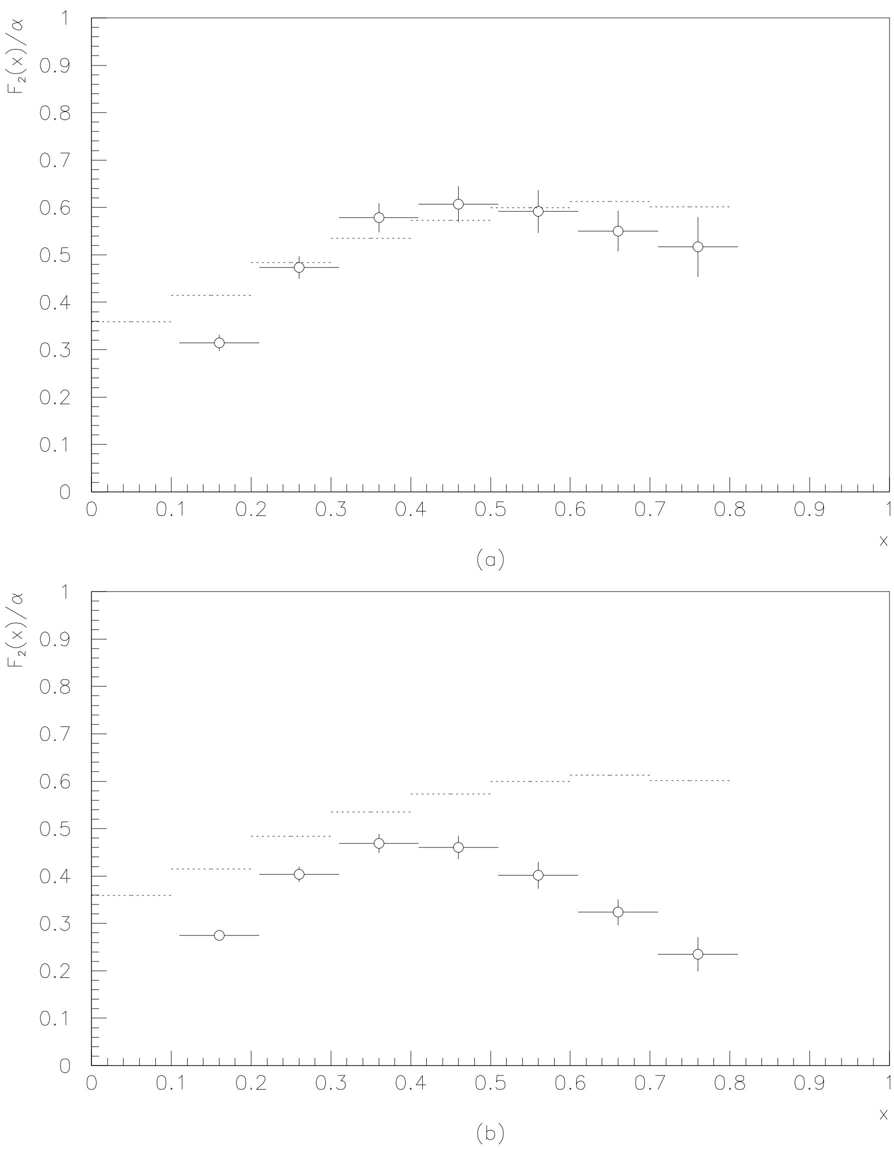

Figure 3: Unfolding tests using Monte Carlos as mock data. (a) HERWIG with the SAS1D parameterisation unfolded using HERWIG with the GRV parameterisation. The dotted lines show SAS1D. Part (b) is the same except that PYTHIA with SAS1D was used for the data sample.

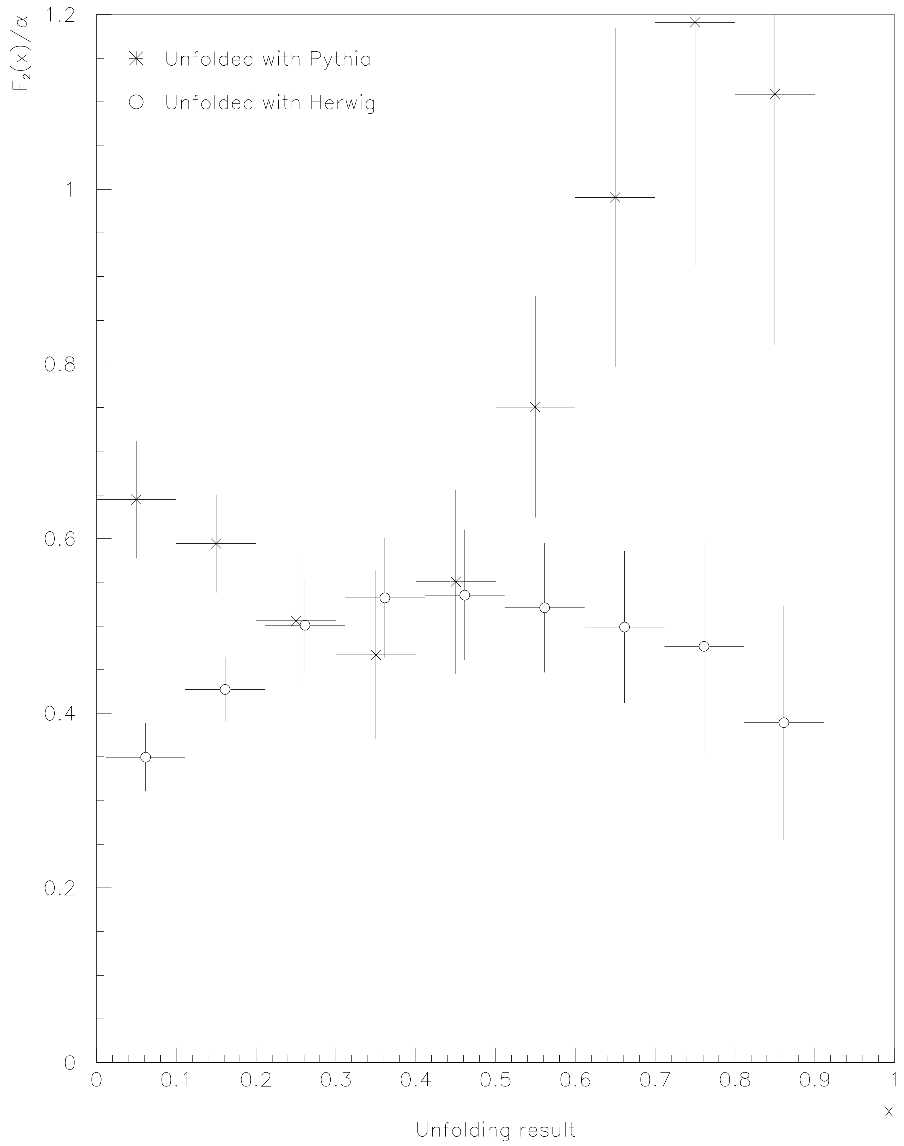

Figure 4: The $\sqrt{s} = 172$ GeV, average $Q^2 = 41$ GeV^2 data sample unfolded using HERWIG (GRV) and PYTHIA (SAS1D).

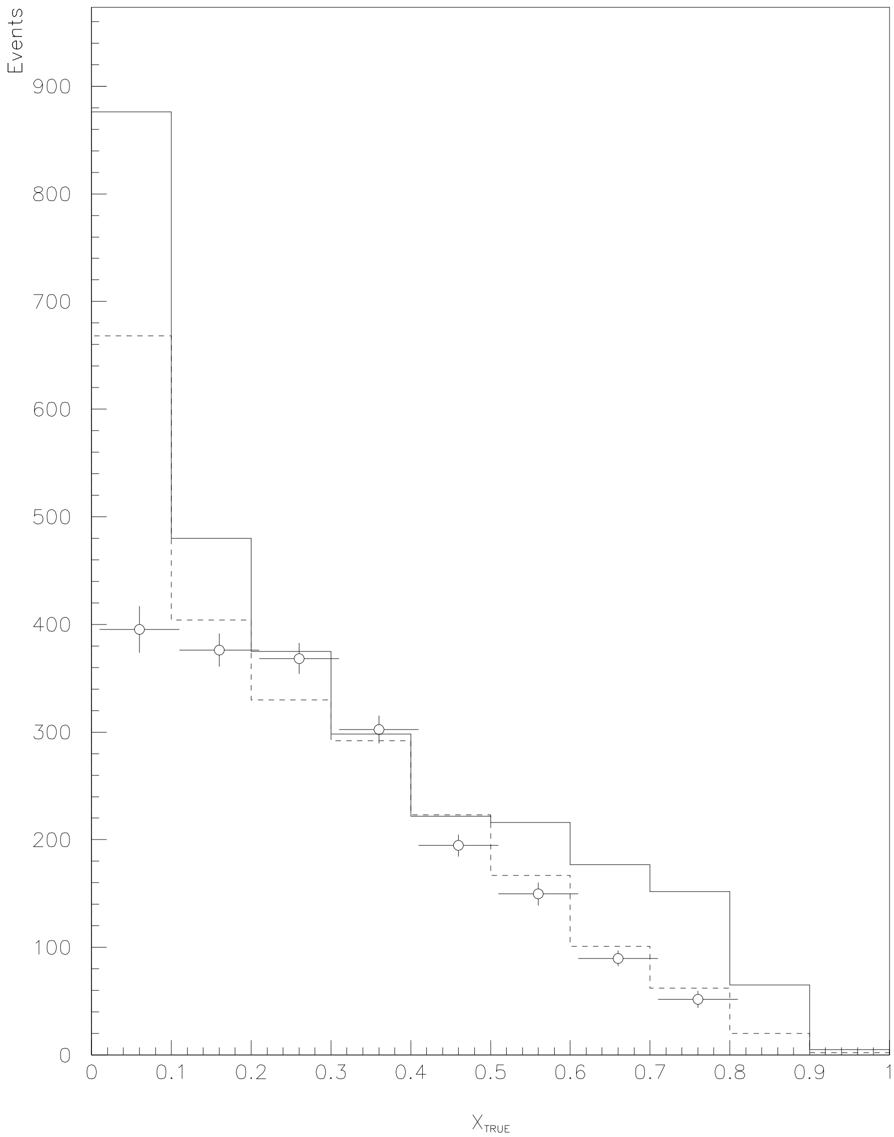

Figure 5: The solid lines show the x distribution for HERWIG with GRV; the dashed lines for PYTHIA with SAS1D. The points are PYTHIA unfolded with HERWIG.

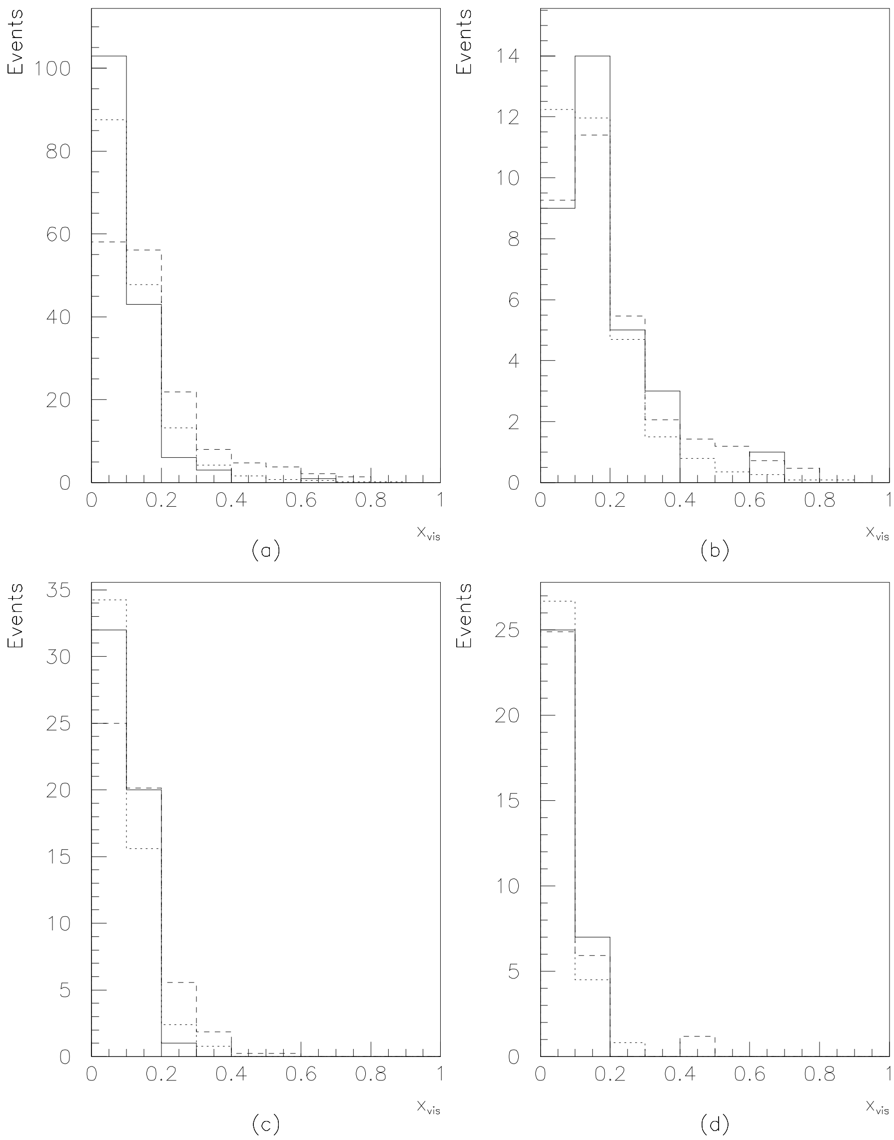

Figure 6: x_{vis} for $x < 0.1$ and for restricted ranges of $E_{\text{t,out}}$. The solid lines show F2GEN, the dashed lines PYTHIA and the dotted lines HERWIG. (a) All events, (b) $E_{\text{t,out}} < 4$ GeV, (c) 4 GeV $< E_{\text{t,out}} < 8$ GeV, (d) 8 GeV $< E_{\text{t,out}} < 12$ GeV. Plots are normalised to the same area.

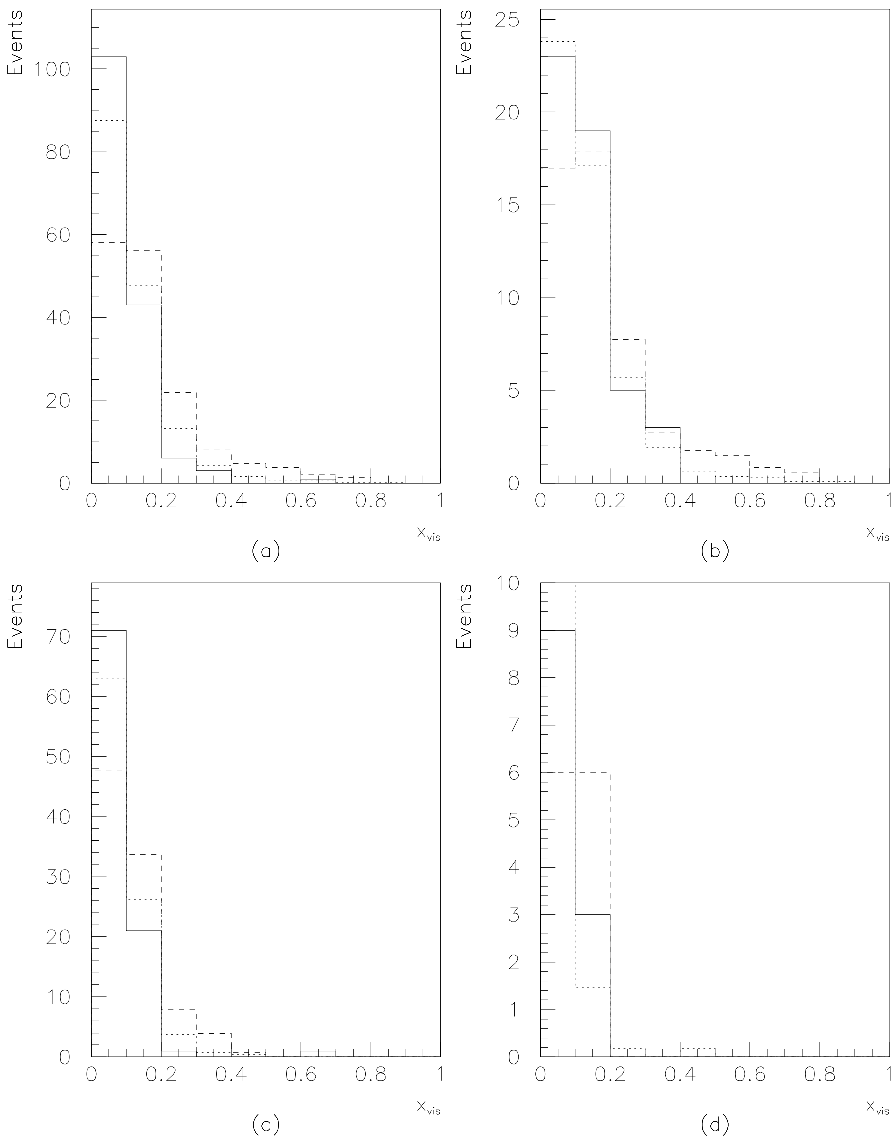

Figure 7: x_{vis} for $x < 0.1$ and for bins of N_{jet}. The solid lines show F2GEN, the dashed lines PYTHIA and the dotted lines HERWIG. (a) All events, (b) $N_{\mathrm{jet}} < 2$, (c) $N_{\mathrm{jet}} = 2$, (d) $N_{\mathrm{jet}} > 2$. Plots are normalised to the same area.

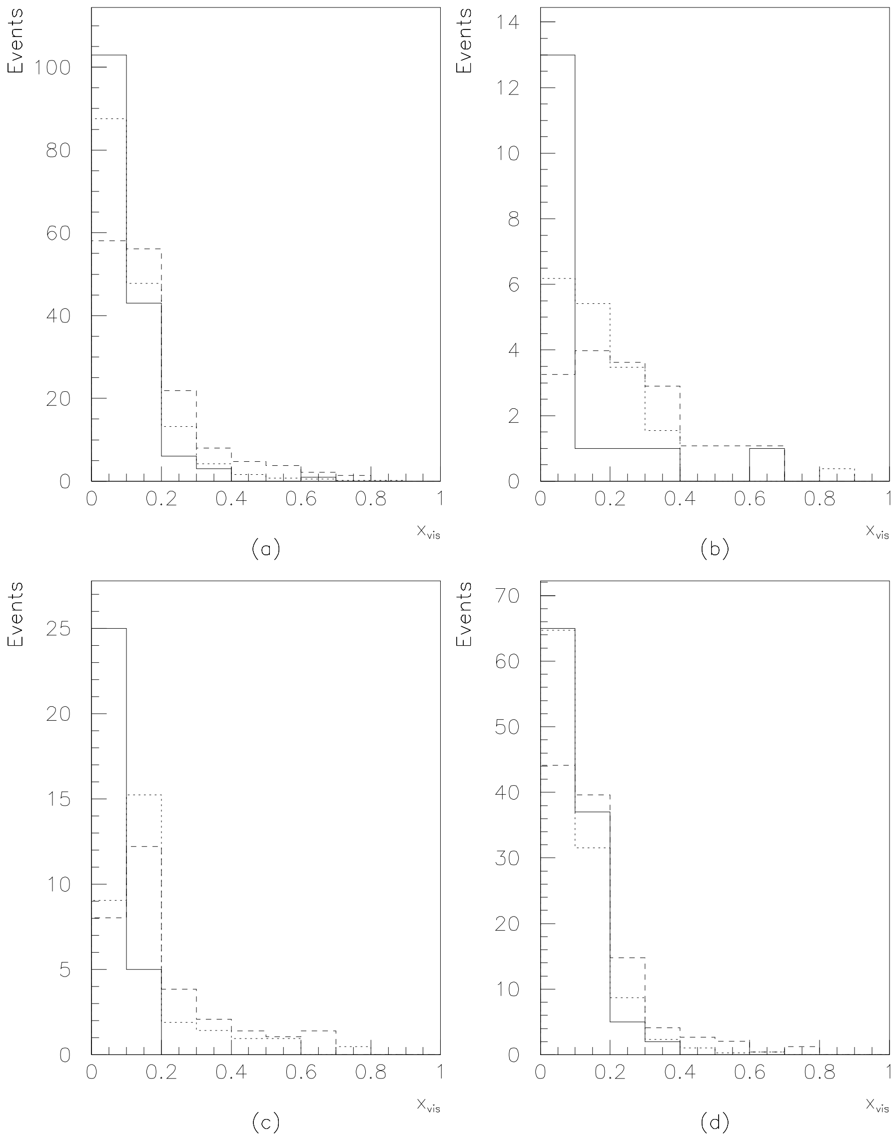

Figure 8: x_{vis} for $x < 0.1$ and for restricted ranges of θ_{thr}. The solid lines show F2GEN, the dashed lines PYTHIA and the dotted lines HERWIG. (a) All events, (b) $\theta_{\text{thr}} < 0.5$, (c) $0.5 < \theta_{\text{thr}} < 0.75$ (d) $\theta_{\text{thr}} > 0.75$, Plots are normalised to the same area.

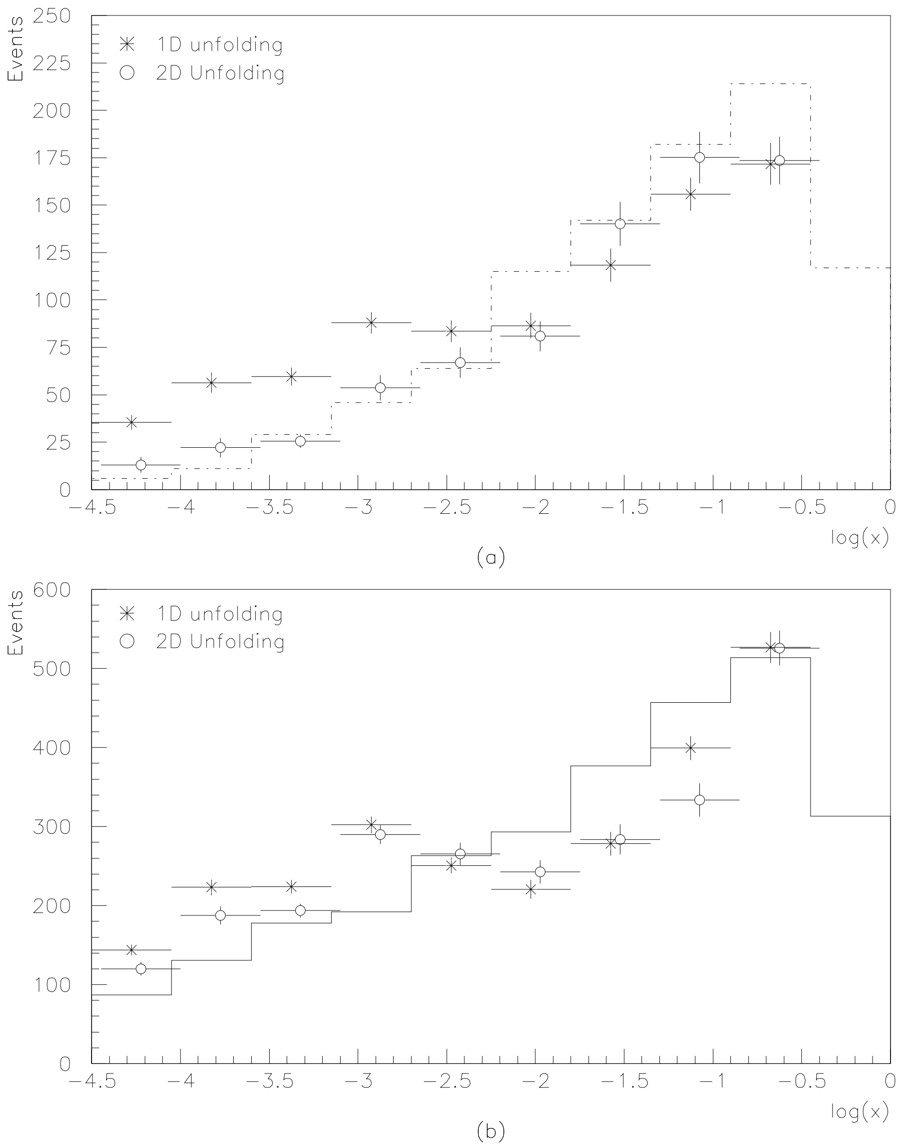

Figure 9: (a) Unfolding of F2GEN with PYTHIA in 1D and 2D. The dotted lines show the true x distribution in F2GEN. Part (b) is the same but HERWIG provides the data sample, and the solid line shows the true x distribution in HERWIG.

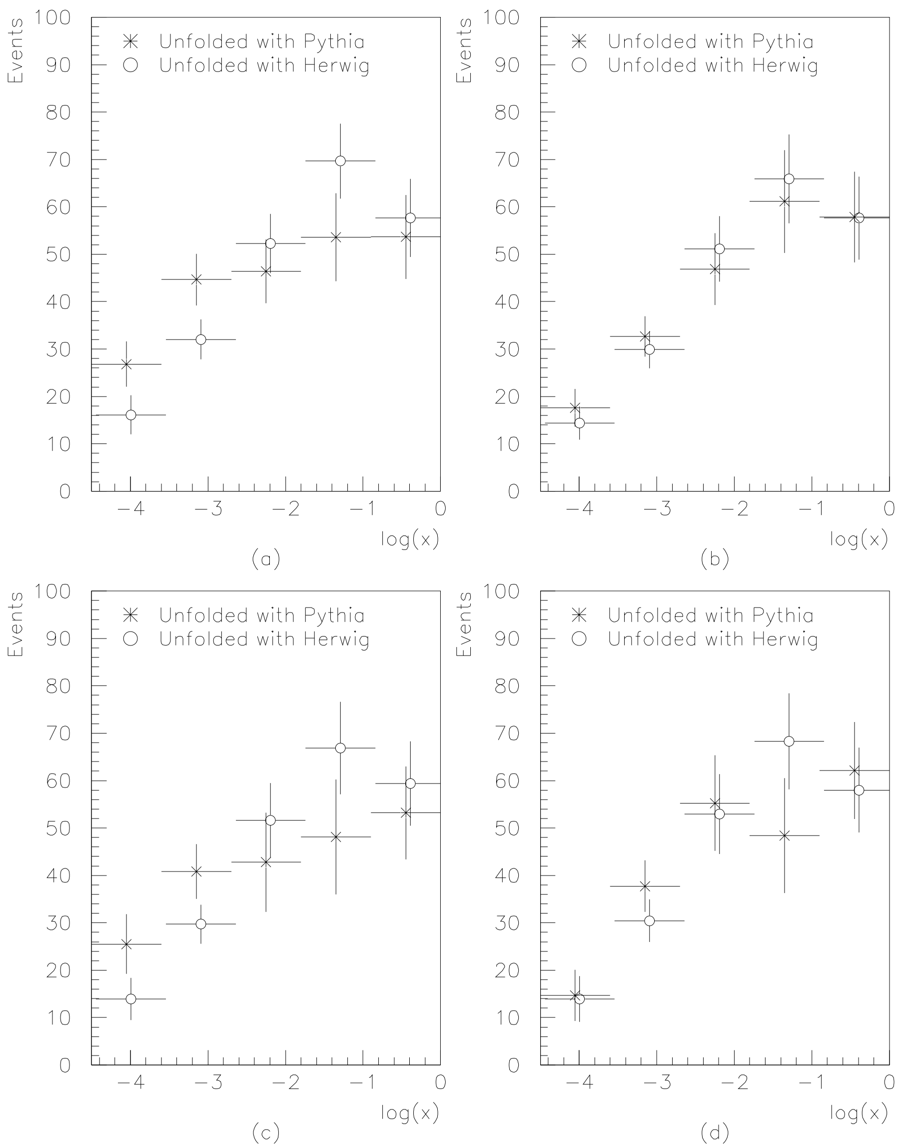

Figure 10: The $\sqrt{s} = 172$ GeV, average $Q^2 = 41\text{GeV}^2$ data sample unfolded using HERWIG and PYTHIA. (a) 1D unfolding. (b) 2D unfolding using $E_{\text{t,out}}$. (c) 2D unfolding with a random number as the second variable. (d) 2D unfolding using N_{jet}.