xFitter

PDF Fitting package

xFitter developers

March 17, 2017

Abstract

The determination of the proton patron distribution functions is a complex endeavor involving several physics processes. The main process is deep-inelastic scattering and the central data set covering most of the proton structure phase space is provided at the HERA ep collider. Further processes (fixed target DIS, ppbar collisions etc.) provide further constraints for particular aspects: flavor separation, very high Bjorken-x etc. In particular, the precise measurements obtained or to come from LHC will continue to improve the knowledge of the PDF. The `xFitter` project aim at providing a framework for QCD analyses related to proton structure in the context of multi-processes and multi-experiments. The framework includes modules or interfaces enabling a large number of theoretical and methodological options, as well as a large number of relevant data sets from HERA, Tevatron and LHC. This manual explains the theoretical input used in the QCD analysis, the fit methodology and the installation procedure of the program. More information and the package downloads can be found on the web site `http://xfitter.org`.

Contents

1 Introduction

This manual provides a short description of the xFitter program which can be used to determine unpolarised proton parton distribution functions (PDFs). The parton distribution functions are needed to calculate cross sections for ep, pp, and $p\bar{p}$ colliders and thus they are required for interpretation of the data collected at the LHC and future colliders.

A schematic structure of the xFitter is illustrated in Fig. 1 which encapsulates all the current functionality of the platform.

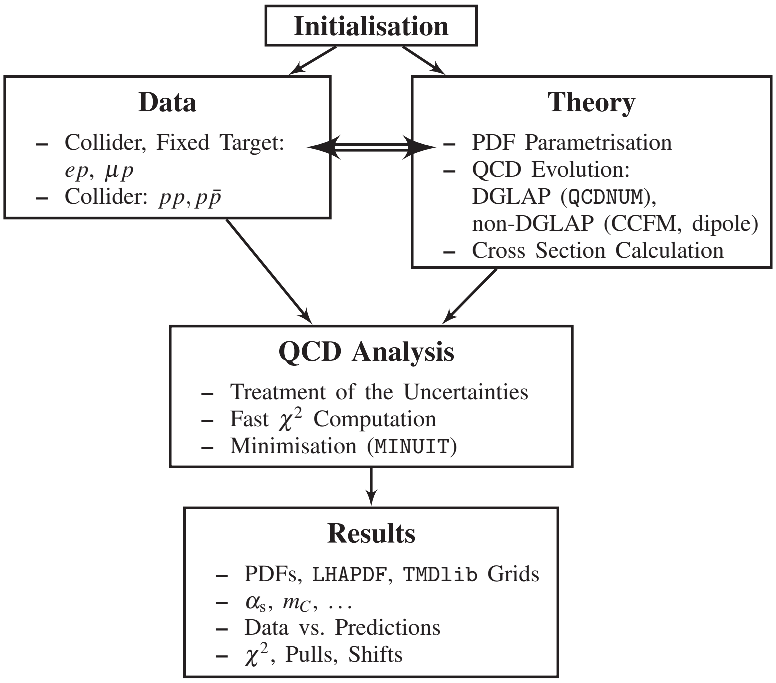

Initialisation
Data
– Collider, Fixed Target: ep, μp
– Collider: $pp, p\bar{p}$
Theory
– PDF Parametrisation
– QCD Evolution: DGLAP (QCDNUM), non-DGLAP (CCFM, dipole)
– Cross Section Calculation
QCD Analysis
– Treatment of the Uncertainties
– Fast χ^2 Computation
– Minimisation (MINUIT)
Results
– PDFs, LHAPDF, TMDlib Grids
– α_s, m_C, ...
– Data vs. Predictions
– χ^2, Pulls, Shifts

Figure 1: Schematic structure of the xFitter program.

This manual is structured such that it first describes briefly the theoretical input (section 2), followed by a description of the PDF parameterisation (section 3.1) and various χ^2 functions used in the minimisation (section 3.2). The minimisation is based on the standard MINUIT program [1] which is not discussed here. Section 5 is dedicated to program installation instructions for different fit scenarios (section 5.1) and provides a description of the program steering cards, with the output options given in section 5.2.

2 Theoretical Input

The main features of QCD theory are confinement (at short ranges the quarks are strongly bound inside protons) and asymptotic freedom (at large scales the coupling constant of the strong force decreases and quarks become quasi-free partons). The factorisation theorem exploits these features by separating short and long distances processes, such that structure functions can be written as a convolution between calculable parts (hard scattering coefficients) and non-calculable parts (parton distribution functions (PDFs)), which are therefore parametrised and determined from data.

Factorisation is most rigorously established for deep inelastic lepton-hadron scattering. For hadronic processes in which a colorless electroweak final state is produced (i.e. Higgs, a real or virtual W, Z or γ) factorisation is also well established and differential cross section calculations are currently available up to next-to-next-to leading order (NNLO) in perturbation theory.

Factorisation is also proved to work for sufficiently inclusive colored final states, such as the one-jet and dijet cross section. In this case no fully rigorous all-order demonstration is available, but no counter-example to factorisation has been found so far.

For the diffractive case, two factorisations are used: the Regge factorisation and the collinear factorisation for the hard scattering.

The proton PDFs are classically extracted from QCD fits by a measure of the agreement between data and theory models. The fit procedure used in the `xFitter` framework is common to all processes and it consists first in parametrising the PDFs at a starting scale Q_0^2, chosen to be below the charm mass threshold. The PDFs are then evolved using coupled, integro-differential Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) [2–6] evolution equations as implemented in the `QCDNUM` [7] program in the $\overline{\mathrm{MS}}$ scheme (LO, NLO nd NNLO evolutions are available [8, 9]). The PDFs calculated at a scale corresponding to a measured cross section are convoluted with the partonic cross sections to calculate the predicted cross section. For all measurement points, the predicted and measured cross sections together with their corresponding errors are used to build a global χ^2, minimized to determine the initial PDF parameters. This generic procedure includes a number of subtleties and assumptions, depending on the measurements, scale and available calculations.

In the following sections, the theoretical input for various processes is described, for example the electron-proton deep inelastic scattering (DIS) process in section 2.1, the Drell-Yan process in section 2.2. For the jet cross section calculations, `xFitter` uses APPLGRID or `fastNLO`, see section 2.5. Section 2.3 describes heavy-quark production in hadron collisions at NLO. In section 2.4 the $t\bar{t}$ cross sections are calculated based on the HATHOR package. Alternative approaches to collinear factorisation are also discussed, see sections 2.6 for Dipole models and 2.7 for unintegrated PDFs. The diffractive PDFs are discussed in section 2.8, in the context of the resolved Pomeron model which predicts such processes as seminclusive diffractive scattering in DIS.

2.1 Deep Inelastic Scattering Formalism and Schemes

The DIS experiments provide the cleanest approach to measure the proton structure. The DIS experiments are carried out either on fixed targets or at collider facilities using electrons, muons or neutrinos to probe the proton. In the DIS process a lepton is scattered off the constituents of the proton by a virtual exchange of a neutral (NC) or charged (CC) boson producing a hadronic shower and a scattered lepton in the final state. The DIS kinematic variables are illustrated in figure 2: the negative squared four-momentum of the exchange boson, Q^2, the scaling variable x, which can be related in the parton model to the fraction of momentum carried by the struck quark, and the inelasticity parameter y, which is the fraction of the energy transferred to the hadronic vertex. The NC (and similarly CC) cross section can be expressed in terms of generalised structure functions:

$$\frac{d^2\sigma_{NC}^{e^\pm p}}{dxdQ^2} = \frac{2\pi\alpha^2}{xQ^4}[Y_+\tilde{F}_2^\pm \mp Y_- x\tilde{F}_3^\pm - y^2\tilde{F}_L^\pm], \tag{1}$$

where $Y_\pm = 1 \pm (1-y)^2$. The structure function $\tilde{F}_2$ is the dominant contribution to the cross section, $x\tilde{F}_3$ is important at high Q^2 and $\tilde{F}_L$ is sizable only at high y. $\tilde{F}_2^\pm$ and $\tilde{F}_3^\pm$ can be expressed in terms of five structure functions describing the contributions from pure photon exchange, γZ interference and pure Z exchange:

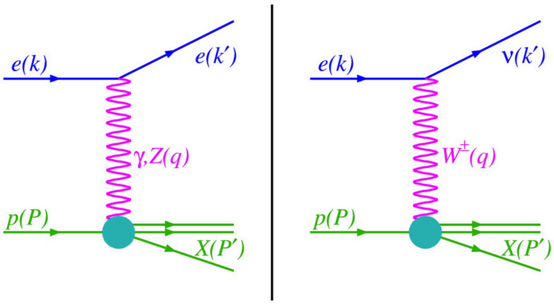


Figure 2: Diagram for a neutral and charged current DIS scattering.

$$\tilde{F}_2^{\pm} = F_2 - v_e\left(\frac{\kappa_W Q^2}{Q^2 + M_Z^2}\right)F_2^{\gamma Z} + (v_e^2 + a_e^2)\left(\frac{\kappa_W Q^2}{Q^2 + M_Z^2}\right)^2 F_2^Z \tag{2}$$

$$x\tilde{F}_3^{\pm} = \pm a_e\left(\frac{\kappa_W Q^2}{Q^2 + M_Z^2}\right)xF_3^{\gamma Z} \mp 2a_e v_e\left(\frac{\kappa_W Q^2}{Q^2 + M_Z^2}\right)^2 xF_3^Z \tag{3}$$

Here, the pure photon exchange is described by F_2, pure Z exchange by F_2^Z and xF_3^Z, and γZ interference by $F_2^{\gamma Z}$ and $xF_3^{\gamma Z}$. v_e is the weak vector and a_e the weak axial-vector coupling of the electron to the Z. The Weinberg angle θ_W enters the quantity κ_W in the following way: $\kappa_W = \frac{1}{4\sin^2\theta_W \cos^2\theta_W}$.

In the framework of perturbative QCD the structure functions are directly related to the parton distribution functions, i.e. in leading order (LO) F_2 is the momentum sum of quark and anti-quark distributions weighted by the quark charge squared:

$$[F_2, F_2^{\gamma Z}, F_2^Z] = x\sum_q [e_q^2, 2e_q v_q, v_q^2 + a_q^2]\{q(x,Q^2) + \overline{q}(x,Q^2)\} \tag{4}$$

$$[xF_3^{\gamma Z}, xF_3^Z] = x\sum_q [e_q^2 a_q, 2v_q a_q]\{q(x,Q^2) - \overline{q}(x,Q^2)\} \tag{5}$$

In analogy to neutral currents, the inclusive CC ep cross section can be expressed in terms of structure functions:

$$\frac{d^2\sigma_{CC}^{e^\pm p}}{dxdQ^2} = \frac{G_F^2}{4\pi x}\left(\frac{M_W^2}{Q^2 + M_W^2}\right)^2 [Y_+ W_2^{\pm} + y^2 W_L^{\pm} \mp Y_- xW_3^{\pm}], \tag{6}$$

Here, G_F is the Fermi constant which is related to the weak coupling g and electromagnetic coupling e, i.e. $G_F = \frac{g^2}{4\sqrt{2}M_W^2} = \frac{e^2}{4\sqrt{2}\sin^2\theta_W M_W^2}$. In LO the e^+p and e^-p cross sections are sensitive to different quark flavours:

$$\begin{aligned}\tilde{\sigma}_{CC}^{e^+p} &= x[\overline{u} + \overline{c}] + (1-y)^2 x[d + s]\\ \tilde{\sigma}_{CC}^{e^-p} &= x[u + c] + (1-y)^2 x[\overline{d} + \overline{s}].\end{aligned} \tag{7}$$

The cross-section predictions are obtained by convoluting the PDFs with the hard scattering coefficient functions. There are various schemes for separating in the structure functions into calculable processes and PDFs. For the DIS processes these are the Fixed-Flavour number (FFN) [10–12] or the general mass Variable-Flavour number (GM-VFN) [13] schemes. In the FFN scheme, heavy quark contributions are explicitly included in the hard cross sections. In the VFN scheme, PDFs corresponding to heavy quarks are introduced and the number of active flavors changes by one unit when the scale crosses the threshold for heavy quark distribution ($Q^2 > m_Q^2$).

The evolution program QCDNUM [7] used in xFitter provides the calculations of the deep inelastic structure functions in the zero-mass, generalised mass and the fixed flavour number schemes. VFN schemes with various treatments for the heavy quark thresholds are considered in xFitter :

- the Thorne Roberts (TR) scheme with its variants at NLO and NNLO [14, 15] as provided by the MSTW group,
- the ACOT scheme with its variants at LO and NLO as provided by the CTEQ group,
- the FONLL scheme at NLO and NNLO [16] provided by the external code APFEL [17],
- BMSN scheme at NLO and NNLO [1].

The fixed-flavour number scheme is available via the QCDNUM implementation and via the OPENQCDRAD [18] interface. Each of these schemes is briefly discussed below.

2.1.1 Zero-Mass Variable Flavour Scheme

In the zero-mass variable flavour number scheme (ZM-VFNS) heavy quark densities are included in the proton for $Q^2 >> m_h^2$ but they are treated as massless in both the initial and final states. This scheme is accurate in the region where Q^2 is much greater than m_h^2 but becomes unreliable for $Q^2 \sim m_h^2$.

2.1.2 General Mass Variable Flavour Scheme: Thorne-Roberts scheme

The Thorne-Roberts (TR) scheme, (referred as RT scheme in the xFitter) is a general-mass variable flavour number scheme (GM-VFNS) used as default for the MTSW PDF sets. GM-VFNSs smoothly connect the two regions: scales below ($Q^2 < m_h^2$) and scales much above the heavy quark scale threshold ($Q^2 >> m_h^2$). However, the connection is not unique. A GM-VFNS can be defined by demanding equivalence of the $n_f = n$ (FFNS) and $n_f = n + 1$ flavour (ZM-VFNS) descriptions above the transition point for the new parton distributions (they are by definition identical below this point), at all orders.

The TR scheme has two different variants: TR standard (as used in MSTW PDF sets [15, 19]) and TR optimal [20], with a smoother transition across the heavy quark mass scales. Both of these variants are accessible within the xFitter package. The calculations are available to NLO and NNLO. In addition, a fast version of the scheme is available (i.e. RT FAST) by using the k-factor technique. The k-factors are defined as the ratio between massless and massive scheme. They are applied to the fast massless scheme accessed by QCDNUM. However, the k-factors are only calculated correctly for the PDF parameters which enter the first iteration of the minimisation and are not updated with each iteration. Hence the RT-fast calculation must be repeated by inputting the final PDF parameters and iterating this procedure until the input and output PDFs are not significantly different

[1] The BMSN scheme as provided by the ABM group currently is not yet fully implemented in xFitter .

2.1.3 General Mass Variable Flavour Scheme: ACOT scheme

The Aivazis-Collins-Olness-Tung scheme belongs to the group of VFN factorisation schemes that use the renormalization method of Collins-Wilczek-Zee (CWZ) [21]. This scheme involves a mixture of the $\overline{\text{MS}}$ scheme for light partons (and for heavy partons when the factorisation scale is larger than the heavy quark mass) and the zero-momentum subtraction renormalisation scheme for graphs with heavy quark lines (if the factorisation scale is smaller than the mass of the heavy quark threshold). The DGLAP kernels and PDF evolution are pure $\overline{\text{MS}}$. Therefore, the ACOT scheme is considered to be a minimal extension of the $\overline{\text{MS}}$ scheme.

Within the ACOT package, different variants of the ACOT scheme are available: ACOT Full, S-ACOT Chi, ACOT ZM, $\overline{\text{MS}}$ at LO and NLO. For the longitudinal structure function higher order calculations are also available. The ACOT Full implementation fully takes into account the quark masses and it reduces to ZM $\overline{\text{MS}}$ scheme in the limit of masses going to zero, but it has the disadvantage of being quite slow. Therefore the k-factor technique has been adopted within the `xFitter` machinery in order to perform QCD fits. The k-factor can be defined in two different ways: on the one hand as the ratio between same order calculations but massless vs massive (i.e. NLO (ZM-VFNS)/NLO (ACOT), on the other hand one could speed up the calculations by defining the k-factors as the ratio between LO (massless)/NLO (massive). Both options are available in the `xFitter` package and give similar results. For convergence of the k-factors usually $2-3$ repetitions of the fit are needed. The different variants of this scheme are all integrated in the `xFitter` framework and can be selected via the namelist `HF_SCHEME` in the `steering.txt` (ACOT ZM, ACOT FULL, S-ACOT Chi).

The differences between TR and ACOT scheme types are summarised in the figure 3. One major issue in a complete GM-VFNS, is that of the ordering of the perturbative expansion. The equivalency of swapping the $O(m_H^2/Q^2)$ terms between Wilson coefficients (or hard-scattering amplitudes) without violating the definition of a GM-VFNS is what mainly distinguish the ACOT from TR schemes.

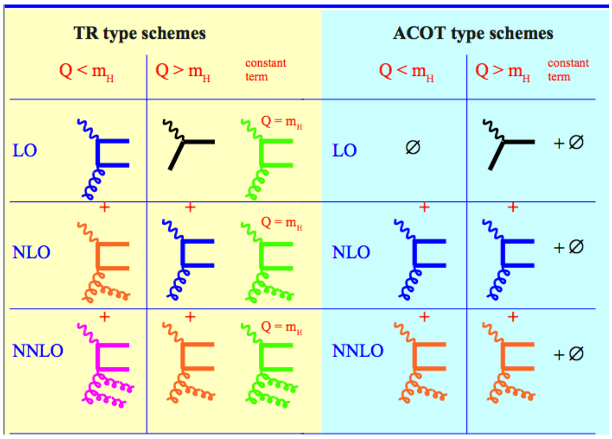


Figure 3: Schematic summary of ACOT and TR schemes.

2.1.4 General Mass Variable Flavour Scheme: FONLL scheme

The FONLL scheme was originally introduced to describe the transverse-momentum distribution of heavy flavours in hadronic collisions [22], while its application to DIS structure functions was presented more recently in Ref. [16] and is presently used by the NNPDF collaboration. The name is motivated by the fact that the method was originally used to combine a fixed (second) order (FO) calculation with a next-to-leading-log (NLL) one. However, the method is entirely general and can be used to combine consistently a fixed-order with a resummed calculations to any order.

The FONLL scheme relies on standard QCD factorization and calculations with massive quarks in the decoupling scheme of Ref. [23] and with massless quarks in the $\overline{\text{MS}}$ scheme. It is based upon the idea of looking at both the massless and massive calculations as power expansions in the strong coupling constant α_s, and replacing the coefficients of the expansion in the former with their exact massive counterparts in the latter. Exactly like the other GM-VFNSs presented above, also the FONLL method is intended to provide a framework capable to be accurate over a wide range of energies. Actually, it has been recently shown that the FONLL scheme is equivalent to the S-ACOT variant of the ACOT scheme to all orders, while, when an intrinsic charm component is included to the FONLL scheme, it is identical to ACOT FULL, again to all orders in perturbation theory [24, 25].

All presently available variants of the FONLL scheme, *i.e.* A, B and C, are available in `xFitter` by means of their implementation in the `APFEL` code [17].

2.1.5 Fixed -Flavour Number Scheme

As mentioned before, in the FFN scheme only the gluon and the light quarks are considered as partons within the proton, massive quarks are produced perturbatively in the final state. `xFitter` includes the FFN scheme from ABM [18] and can also use the `QCDNUM` implementation.

In addition, a recent variation of the fixed-flavour number scheme in which the running mass definition of the heavy quark mass is used in the $\overline{\text{MS}}$ scheme [26] is implemented in `xFitter` .This variant is realised via the interface to the open-source code OPENQCDRAD [18]. This scheme has the advantage of reducing the sensitivity of the DIS cross sections to higher order corrections, and improving the theoretical precision of the mass definition. In `QCDNUM`, the calculation of the heavy quark contributions to DIS structure functions are available at NLO and only electromagnetic exchange contributions are taken into account. In the ABM implementation, the QCD corrections to the massive Wilson coefficients up to the currently best known approximate NNLO for the neutral-current (NC) heavy-quark production [27] and up to NLO for the charged-current (CC) case are available.

2.1.6 Electroweak corrections for ep scattering

To properly compare the experimental data with theoretical predictions, QED corrections are necessary. In the `xFitter` the electroweak corrections for the DIS process are based on the EPRC package [28].

The calculations of higher-order electroweak corrections to DIS scattering at HERA are performed in the on-shell scheme where the gauge bosons masses M_W abd M_Z are treated symmetrically as basic parameters together with the top and Higgs masses, besides the fine structure constant α and other fermion masses.

The code provides the running of α using the most recent parametrisation of the hadronic contribution to Δ_α [29], as well as an older one from Burkhard [30].

2.2 Drell Yan processes

This section presents calculations of Drell Yan processes that can be used to predict lepton pair production at the LHC or Tevatron. The Drell-Yan process is shown in Fig. 4. The calculations of the Drell Yan processes are known for many observables up to NNLO order. For example, there are packages such as FEWZ [31] and DYNNLO [32] for NNLO or `MCFM` [33] for NLO calculations. However, due to the complicated nature of these calculation involving an increased number of diagrams with each additional order, these calculations are too slow to be used iteratively in a fit. There are various methods to overcome this shortage: using the k-factors approximation from lower to higher order, or using the so-called grid technique (storing the matrix elements on grids such that the cross sections maybe calculated later by convoluting these grids with the input PDFs) when available.

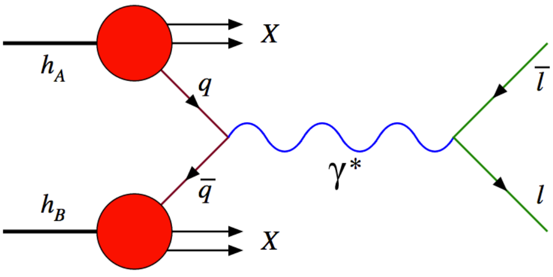


Figure 4: Diagram for a generic DY scattering.

`xFitter` provides two implementations for pp Drell Yan processes. The first implementation uses calculations at LO which can be extended to NLO using k-factors, the second uses the APPLGRID interface. A short description of both implementations is given below while for details of the theoretical modules we direct the user to the references for these packages provided in the description.

The leading order Drell-Yan [34, 35] triple differential cross section in invariant mass M, boson rapidity y and CMS lepton scattering angle $\cos\theta$, for the neutral current, can be written as

$$\frac{\mathrm{d}^3\sigma}{\mathrm{d}M\mathrm{d}y\mathrm{d}\cos\theta} = \frac{\pi\alpha^2}{3MS}\sum_q P_q\left[F_q(x_1,Q^2)F_{\bar{q}}(x_2,Q^2) + (q \leftrightarrow \bar{q})\right], \tag{8}$$

where S is the squared CMS beam energy, $x_{1,2} = \frac{M}{\sqrt{S}}\exp(\pm y)$, $F_q(x_1,Q^2)$ is the parton number density, and

$$\begin{aligned} P_q &= e_l^2 e_q^2(1+\cos^2\theta) \\ &+ e_l e_q \frac{2M^2(M^2-M_Z^2)}{\sin^2\theta_W\cos^2\theta_W[(M^2-M_Z^2)^2+\Gamma_Z^2M_Z^2]}[aA_q(1+\cos^2\theta)+2bB_q\cos\theta] \\ &+ \frac{M^4}{\sin^4\theta_W\cos^4\theta_W[(M^2-M_Z^2)^2+\Gamma_Z^2M_Z^2]}[(a^2+b^2)(A_q^2+B_q^2)(1+\cos^2\theta)+8abA_qB_q\cos\theta]. \end{aligned} \tag{9}$$

Here θ_W is the Weinberg angle, M_Z and Γ_Z are Z boson mass and width, and

$$\begin{aligned} a &= -\frac{1}{4} + \sin^2\theta_W, \\ b &= -\frac{1}{4}, \\ A_q &= \frac{1}{2} I_q^3 - e_q \sin^2\theta_W, \\ B_q &= \frac{1}{2} I_q^3, \\ I_u^3 &= -I_d^3 = \frac{1}{2}, \\ e_l &= -1, e_u = \frac{2}{3}, e_d = -\frac{1}{3} \end{aligned} \tag{10}$$

give the electro-weak couplings.

The expression for charged current scattering has a simpler form.

$$\frac{d^3\sigma}{dMdyd\cos\theta} = \frac{\pi\alpha^2}{48S\sin^4\theta_W} \frac{M^3(1-\cos\theta)^2}{(M^2-M_W^2)+\Gamma_W^2 M_W^2} \sum_{q_1,q_2} V_{q_1q_2}^2 F_{q_1}(x_1,Q^2) F_{q_2}(x_2,Q^2), \tag{11}$$

where $V_{q_1q_2}$ is the CKM quark mixing matrix and M_W and Γ_W are W boson mass and decay width.

The simple form of these expressions allows the calculation of integrated cross sections without utilization of Monte-Carlo techniques. This is particularly useful for PDF fitting purposes because statistical fluctuations are avoided in this case. In both neutral and charged current expressions the parton distribution functions factorise as functions dependent only on boson rapidity y and invariant mass M. The integral in $\cos\theta$ can be computed analytically and integrations in y and M can be performed with the Simpson method. The $\cos\theta$ parts are kept in the equation explicitly because their integration is asymmetric for data in lepton η bins and also because of the need to apply lepton $p_\perp$ cuts.

The fact that PDF functions factorise, allows high speed calculations when performing parameter fits over lepton rapidity data. In this case the factorised part of the expression which is independent of PDFs can be calculated only once for all minimisation iterations. The leading order code in `xFitter` package implements this optimisation and uses fast convolution routines provided by `QCDNUM`. Currently the full width LO calculations are optimised for lepton pseudorapidity and boson rapidity distributions with the possibility to apply lepton $p_\perp$ cuts. This flexibility allows the calculations to be performed within the phase space corresponding to the available measurement.

The calculated leading order cross sections are multiplied by k-factors to obtain predictions at NLO or NNLO precision.

Alternatively, one can obtain the NLO predictions directly by using APPLGRID or `fastNLO` techniques, which rely on the factorisation theorem by decoupling the hard scattering coefficients from PDFs. The hard scattering coefficients are calculated once and stored into a grid for a given kinematic bin, speeding up the convolution process with the PDFs and thus allowing to for fast QCD fits. These methods are described in more detail in section 2.5. An independent treatment for the electro-weak corrections is applied as the independent k-factors, using packages such as SANC and FEWZ.

2.3 Heavy-quark production in hadron collisions at NLO

Theoretical predictions for heavy-quark production in pp collisions are obtained using the massive NLO calculations [36–38] in the FFNS, also available as part of the Mangano-Nason-Ridolfi (MNR) calculations [39]. In `xFitter` the one-particle inclusive variant of the calculations is implemented [40, 41].

The pole mass definition is employed. Additionally non-perturbative quark to parton fragmentation is implemented (including most popular Kartvelishvili [42], Peterson [43] and BCFY functions [44]). All flexibility of the original MNR code is retained: the factorisation and renormalisation scales, heavy-quark mass, strong coupling constant, fragmentation function and PDFs may be changed in each iteration, i.e. treated as fit parameters.

2.4 Cross Sections for $t\bar{t}$ production in pp or $p\bar{p}$ collisions

Top-quark pairs ($t\bar{t}$) are mainly produced at hadron colliders via gg fusion and $q\bar{q}$ annihilation. There are also qq' and qg production modes. The program HATHOR [45] allows the calculation of the expected total $t\bar{t}$ cross section at $p\bar{p}$ and pp colliders up to approximate NNLO accuracy. Version 1.3 of HATHOR includes the exact NNLO for $q\bar{q} \to t\bar{t}$ [46] as well as a new high-energy constraint on the approximate NNLO calculation obtained from soft-gluon resummation [47]. The default choice for renormalization and factorization scale in $t\bar{t}$ production is the top-quark mass, m_t. The pole mass scheme is typically employed for m_t but HATHOR also supports calculations in the $\overline{\text{MS}}$ scheme.

2.5 Jets

This sections presents various fast calculational techniques for jet production based on the factorization formalism.

The calculation of higher order jet cross sections is very demanding in terms of computing power. The reasons are the large number of contributing Feynman diagrams and also the large number of infrared divergences. For an accurate cancellation of these singularities, the dipole subtraction method is often applied in such calculations. During the necessary Monte Carlo integration a very fine phase space sampling has to be performed in order to account for the accurate cancellation of the counter terms.

In order to enable the inclusion of jet-cross section measurements in PDF and α_s fits, the perturbative coefficients have to be pre-computed in a PDF and α_s independent way. For this purpose, two similar tools are interfaced to the `xFitter`.

2.5.1 `fastNLO`

The `fastNLO` project [48–50] enables the inclusion of hadron-induced differential data in PDF and α_s fits and is mostly used for jet cross sections in DIS and hadron-hadron collisions. This tool uses multi-dimensional interpolation techniques to convert the convolutions of perturbative coefficients with parton distribution functions and the strong coupling into simple products. The phase space of the interpolations are optimized for each individual data point during the generation of the `fastNLO` tables. The perturbative coefficients for jet-production in NLO are calculated by the `NLOJet++` program [51] where calculations in DIS [52] as well as in hadron-hadron collisions [53, 54] with threshold-corrections of $\mathcal{O}$(NNLO) for inclusive jet cross sections [55] are available.

The `fastNLO` code is included as source code in the `xFitter` package and no further requirements or compilation options are needed. In order to include a new measurement into the PDF-fit, the `fastNLO` tables have to be specified. These tables include all necessary information about the perturbative coefficients and the calculated process for all bins of a certain dataset. Tables for almost all published jet measurements are available through the project website `http://fastnlo.hepforge.org`. The calculation of a new `fastNLO` table requires to download the full `fastNLO` toolkit package and a suitable generator code from the website.

Features of the fastNLO concept are the very quick convolution of the perturbative coefficients with the PDFs, of $O(100ms)$, and the very high accuracy of the interpolation procedure. The fastNLO tables are conventionally calculated for multiple factors of the factorization scale, and the renormalization scale factor can be chosen freely. Some of the fastNLO tables already allow for the free choice [50] of the renormalization and the factorization scale as a function of two pre-defined observables. The evaluation of the strong coupling constant, which enters the cross section calculation, is taken consistently from the QCDNUM evolution code, or from LHAPDF if specified.

2.5.2 APPLGRID

The APPLGRID [56] package allows the fast computation of NLO cross sections for particular processes for arbitrary sets of proton parton distribution functions. The package implements calculations of Drell Yan cross sections of electroweak boson (Z, W) production as well as jet production in proton-(anti)proton collisions and DIS processes.

The approach is based on storing the perturbative coefficients of NLO QCD calculations of final-state observables measured in hadron colliders in look-up tables. The PDFs and the strong couplings are included during the final calculations, e.g. during PDF fitting. The method allows variation of factorization and renormalization scales in calculations.

The look-up tables (grids) can be generated with modified versions of MCFM [57, 58] or NLOJet++ [54] software as distributed with the full version of APPLGRID package.

APPLGRID supports an interface to the MCFM parton level generators, hence model input parameters such as electroweak parameters are in fact pre-set following the MCFM input steering card, while binning and definitions of the observables for which the differential cross sections are needed are set in the APPLGRID code. The grid parameters, Q^2 binning and interpolation orders are also defined in the code.

APPLGRID constructs the grid tables in two steps: *(i)* exploration of the phase space in order to optimize the memory storage and *(ii)* actual grid construction in the phase space corresponding to the requested observables.

Afterwards the NLO cross sections are restored from the grids using externally provided PDFs, α_S, factorization and renormalization scales. QCD NNLO k-factors can be applied if requested.

In order to use APPLGRID tables in xFitter , the APPLGRID package has to be downloaded and installed first. In addition, the xFitter code has to be configured with a special option (for details see section 5.1).

2.6 DIPOLE models

The dipole picture provides an alternative approach to the virtual photon-proton scattering at low x because it allows the description of both inclusive and diffractive processes. In this approach, the virtual photon fluctuates into a $q\bar{q}$ (or $q\bar{q}g$) dipole which interacts with the proton [59]. The dipoles can be viewed as quasi-stable quantum mechanical states, which have very long life time $\propto 1/m_p x$ and a size which is not changed by scattering. A schematic view of dipole factorisation at small x in DIS is illustrated in figure 5. The virtual photon fluctuates into a quark-antiquark pair and subsequently interacts with the target, and the dynamics of the interaction are embedded in the dipole scattering amplitude.

Several dipole models have been developed to describe various DIS reactions. They vary due to different assumption made about the behavior of the dipole-proton cross sections. In the xFitter three representative models are implemented: the original Golec-Biernat-Wüsthoff (GBW) [60] dipole saturation model, the colour glass condensate approach to the high parton density regime Iancu-Itakura-Munier (IIM) model [61] and a modified GBW model which takes into account the effects of DGLAP evolution Bartels-Golec-Kowalski(BGK) [62].

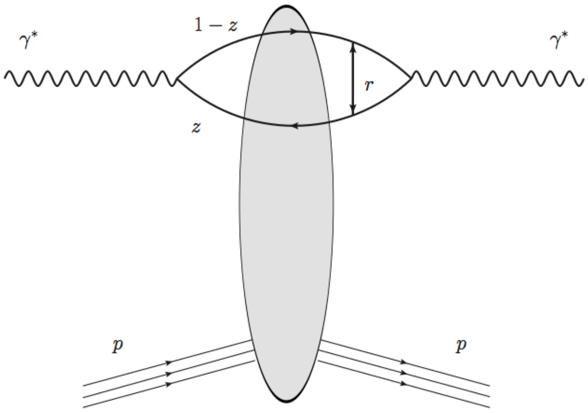


Figure 5: Schematic diagram of dipole factorisation for the inclusive cross section in DIS.

2.6.1 GBW model

In the GBW model the dipole-proton cross section σ_{dip} is given by

$$\sigma_{\text{dip}}(x, r^2) = \sigma_0 \left(1 - \exp\left[-\frac{r^2}{4R_0^2(x)}\right]\right), \tag{12}$$

where r corresponds to the transverse separation between the quark and the antiquark, and R_0^2 is an x dependent scale parameter which has corresponds to a saturation radius, $R_0^2(x) = (x/x_0)^\lambda$. The free fitted parameters are the cross-section normalisation σ_0 as well as x_0 and λ.

2.6.2 IIM model

The IIM model assumes an improved expression for the dipole cross section which is based on the Balitsky-Kovchegov equation [63]. The explicit formula for σ_{dip} can be found in [61]. The free fitted parameters are an alternative scale parameter $\tilde{R}$, x_0 and λ.

2.6.3 BGK model

The BGK model modifies the GBW model by taking into account the DGLAP evolution of the gluon density. The dipole cross section is given by

$$\sigma_{\text{dip}}(x, r^2) = \sigma_0 \left(1 - \exp\left[-\frac{\pi^2 r^2 \alpha_s(\mu^2) x g(x, \mu^2)}{3\sigma_0}\right]\right).$$

The factorization scale μ^2 has the form $\mu^2 = C_{bgk}/r^2 + \mu_0^2$. In this model the gluon density, which is parametrized at some starting scale Q_0^2 by

$$xg(x, Q_0^2) = A_g x^{-\lambda_g} (1-x)^{C_g}.$$

is evolved to larger Q^2's using LO and NLO DGLAP evolution. The free fitted parameters for this model are σ_0, μ_0^2 and three parameters for the gluon density: A_g, λ_g, C_g. The parameter C_{bgk} is kept fixed: $C_{bgk} = 4.0$.

2.6.4 BGK model with valence quarks

The dipole models are valid in the low-x region only, where the valence quark contribution is small, of the order of 5%. The new HERA F_2 data have a precision which is better than 2 %. Therefore, in the xFitter the contribution of the valence quarks is taken from the PDF fits and added to the original BGK model, this is uniquely possible within the xFitter framework. The quality of the fits of the BGK dipole model with valence quarks and without valence quarks are the same. The sample input steering and output fits are discussed in section 5.2.

2.7 Transverse Momentum Dependent (unintegrated PDF) with CCFM

In this subsection another alternative approach to collinear DGLAP evolution is presented. In high energy factorization [64] generally the measured cross section is written as a convolution of the partonic cross section $\hat{\sigma}(k_t)$, which depends on the transverse momentum k_t of the incoming parton, with the k_t-dependent parton distribution function $\tilde{\mathcal{A}}(x, k_t, p)$ (transverse momentum dependent (TMD) or unintegrated uPDF):

$$\sigma = \int \frac{dz}{z} d^2k_t \hat{\sigma}(\frac{x}{z}, k_t) \tilde{\mathcal{A}}(x, k_t, p) \tag{13}$$

The evolution of $\tilde{\mathcal{A}}(x, k_t, p)$ can proceed via the BFKL, DGLAP or via the CCFM evolution equations. In xFitter an extension of the CCFM [65–68] evolution has been implemented. Since the evolution cannot be easily obtained in a closed form, first a kernel $\tilde{\mathcal{A}}(x'', k_t, p)$ is determined from the MC solution of the CCFM evolution equation, and is then folded with the non-perturbative starting distribution $\mathcal{A}_0(x)$ [69]:

$$\begin{aligned} x\mathcal{A}(x, k_t, p) &= x \int dx' \int dx'' \mathcal{A}_0(x) \tilde{\mathcal{A}}(x'', k_t, p)\, \delta(x' \cdot x'' - x) && (14) \\ &= \int dx' \int dx'' \mathcal{A}_0(x) \tilde{\mathcal{A}}(x'', k_t, p)\, \frac{x}{x'} \delta(x'' - \frac{x}{x'}) && (15) \\ &= \int dx' \mathcal{A}_0(x') \cdot \frac{x}{x'} \tilde{\mathcal{A}}\left(\frac{x}{x'}, k_t, p\right) && (16) \end{aligned}$$

The kernel $\tilde{\mathcal{A}}$ includes all the dynamics of the evolution, Sudakov form factors and splitting functions and is determined in a grid of $50 \otimes 50 \otimes 50$ bins in x, k_t, p. In xFitter the evolution of $\mathcal{A}$ is performed using the method implemented in [70, 71].

The calculation of the cross section according to Eq.(13) involves a multidimensional Monte Carlo integration which is time consuming and suffers from numerical fluctuations, and therefore cannot be used directly in a fit procedure involving the calculation of numerical derivatives in the search for a minimum. Instead the following procedure is applied:

$$\begin{aligned} \sigma_r(x, Q^2) &= \int_x^1 dx_g \mathcal{A}(x_g, k_t, p) \hat{\sigma}(x, x_g, Q^2) && (17) \\ &= \int_x^1 dx' \mathcal{A}_0(x') \cdot \tilde{\sigma}(x/x', Q^2) && (18) \end{aligned}$$

The kernel $\tilde{\mathcal{A}}$ has to be provided separately and is not calculable within the program. A starting distribution $\mathcal{A}_0$, at the starting scale Q_0, of the following form is used:

$$x\mathcal{A}_0(x, k_t) = N x^{-B_g} \cdot (1 - x)^{C_g} \left(1 - D_g x + E_g \sqrt{x} + F_g x^2\right) \tag{19}$$

with free parameters N, B_g, C_g, D_g, E_g, F_g.

The calculation of the ep cross section follows eq.(13), with the off-shell matrix element including quarks masses taken from [64] in its implementation in CASCADE [72]. In addition to the boson gluon fusion process, valence quark initiated $\gamma q \to q$ processes are also included, with the valence quarks taken from [73].

2.8 Diffractive PDFs

In this section the diffractive process is briefly described. It was observed at HERA that about 10% of deep inelastic interactions are diffractive leading to events in which the interacting proton stays intact ($ep \to eXp$). In the diffractive process the proton appears well separated from the rest of the hadronic final state by a large rapidity gap, in all other respects the events look similar to normal deep inelastic events. This process is usually interpreted as the diffractive dissociation of the exchanged virtual photon to produce a hadronic system X with mass much smaller than W and the same net quantum numbers as the exchanged photon. Figure 6 illustrates the kinematic variables used to describe the inclusive diffractive DIS process. For this, the proton vertex factorisation approach is assumed such that the diffractive DIS is mediated by the exchange of hard Pomeron and a secondary Reggeon. The factorisable pomeron picture has proved remarkably successful for the description of most of these data.

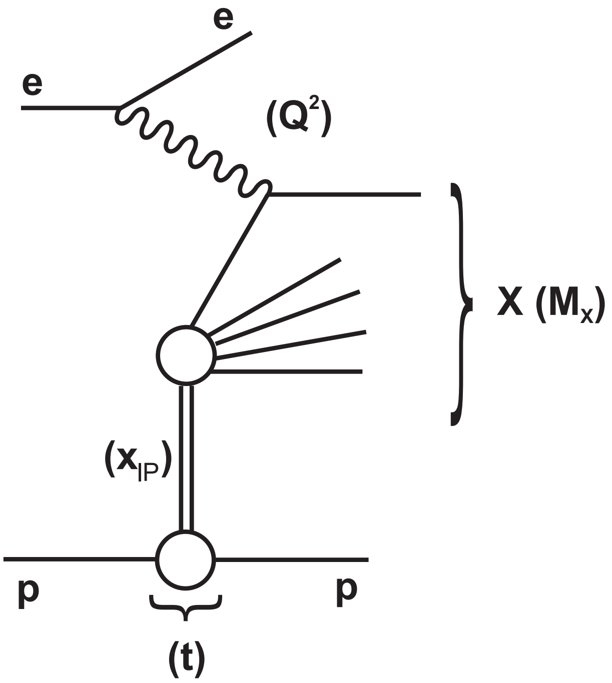


Figure 6: Schematic diagram of the kinematic variables used to describe the inclusive diffractive DIS process.

In addition to x, Q^2 and the squared four-momentum transfer t (the undetected momentum transfer to the proton system), the mass M_X of the diffractively produced final state provides a further degree of freedom. In practice, the variable M_X is often replaced by β,

$$\beta = \frac{Q^2}{M_X^2 + Q^2 - t}. \tag{20}$$

In models based on a factorisable pomeron, β may be viewed as the fraction of the pomeron longitudinal momentum which is carried by the struck parton, $x = \beta x_{I\!P}$.

2.8.1 Cross-section

As for the inclusive case, the diffractive cross-section can be expressed as:

$$\frac{d\sigma}{d\beta\, dQ^2 dx_{I\!P}\, dt} = \frac{2\pi\alpha^2}{\beta Q^4}\left(1+(1-y)^2\right)\overline{\sigma}^{D(4)}(\beta, Q^2, x_{I\!P}, t) \tag{21}$$

where the "reduced cross-section" , $\overline{\sigma}$, is defined as

$$\overline{\sigma}^{D(4)} = F_2^{D(4)} - \frac{y^2}{1+(1-y)^2} F_L^{D(4)} = F_T^{D(4)} + \frac{2(1-y)}{1+(1-y)^2} F_L^{D(4)} \tag{22}$$

The dimension of $F_k^{D(4)}(\beta, Q^2, x_{I\!P}, t)$ is GeV^{-2} and thus quantities integrated over t.

$$F_k^{D(3)}(\beta, Q^2, x_{I\!P}) \equiv \int_{t_{\min}}^{t_{\max}} dt F_k^{D(4)}(\beta, Q^2, x_{I\!P}, t) \tag{23}$$

are dimensionless. The maximum kinematically allowed value of t is given by

$$t_{\mathrm{MAX}} = -\frac{x_{I\!P}^2 m_p^2 + p_\perp^2}{1-x_{I\!P}} \approx -\frac{x_{I\!P}^2}{1-x_{I\!P}} m_p^2 \tag{24}$$

where m_p is the proton mass. As $x = x_{I\!P}\beta$ we can normalize to the standard DIS formula

$$\frac{d\sigma}{d\beta\, dQ^2\, dx_{I\!P}\, dt} = \frac{2\pi\alpha^2}{x\, Q^4}\left(1+(1-y)^2\right) x_{I\!P}\overline{\sigma}^{D(4)}(\beta, Q^2, x_{I\!P}, t) \tag{25}$$

which upon integration over t reads

$$\frac{d\sigma}{d\beta\, dQ^2\, dx_{I\!P}} = \frac{2\pi\alpha^2}{xQ^4}\left(1+(1-y)^2\right) x_{I\!P}\overline{\sigma}^{D(3)}(\beta, Q^2, x_{I\!P}). \tag{26}$$

The diffractive structure functions can be expressed as convolutions of the calculable coefficient functions with diffractive quark and gluon distribution functions, which in general depend on all of $x_{I\!P}$, Q^2, β, t.

2.8.2 Regge factorization

For a better description of data, a contribution from a secondary Reggeon, $I\!R$, is included, hence

$$F_k^{D(4)}(\beta, Q^2, x_{I\!P}, t) = \sum_{X=I\!P, I\!R} \phi_X(x_{I\!P}, t)\, F_k^X(\beta, Q^2) \tag{27}$$

or

$$F_k^{D(3)}(\beta, Q^2, x_{I\!P}) = \sum_{X=I\!P, I\!R} \Phi_X(x_{I\!P})\, F_k^X(\beta, Q^2) \tag{28}$$

where

$$\Phi_X(x_{I\!P}) = \int_{t_{\min}}^{t_{\max}} dt\, \phi_X(x_{I\!P}, t)\,. \tag{29}$$

The fluxes are parametrized as

$$\phi_X(x_{I\!P}, t) = \frac{A_X\, e^{b_X t}}{x_{I\!P}^{2\alpha_X(t)-1}} \tag{30a}$$

where

$$\alpha_X(t) = \alpha_X(0) + \alpha'_X t\,. \tag{30b}$$

The function $F_k^{I\!R}(\beta, Q^2)$ is taken to be that of the pion.

3 Methodology for PDF fits

In this section the main fit formalism is presented and detail and various options implemented in `xFitter` are described. Such options come in three broad classes: different functional forms used to parametrise the PDFs at the starting scale (section 3.1); different defintions of the χ^2 (section 3.2); and different treatement of experimental uncertainties (section 3.2).

3.1 PDF Parameterisation

In this Section the different possible parametrisations for the PDFS at the staring scale which are implemented in the `xFitter` are described. Sec. 3.1.1 covers standard functional forms, Sec. 3.1.2 covers the bi-log normal functional for and Sec. 3.1.3 covers more exotic forms based on generalised polynomials such as the Chebyshev polynomials.

3.1.1 Standard Functional form

The term standard functional form is understood to refer to a simple polynomial that interpolates between the low and high x regions:

$$xf(x) = Ax^B(1-x)^C P_i(x), \tag{31}$$

Standard forms are commonly used by PDF groups.

The notation used throughout this text reflects the notation used in the code.

HERAPDF style

The parametrised PDFs at HERA are the valence distributions xu_v and xd_v, the gluon distribution xg, and the u-type and d-type sea $x\bar{U}$, $x\bar{D}$, where $x\bar{U} = x\bar{u}$, $x\bar{D} = x\bar{d} + x\bar{s}$. The following standard functional form is used to parametrise them

$$xf(x) = Ax^B(1-x)^C(1 + Dx + Ex^2), \tag{32}$$

where the normalisation parameters, A_{uv}, A_{dv}, A_g, are constrained by the number sum-rules and the momentum sum-rule. The B parameters $B_{\bar{U}}$ and $B_{\bar{D}}$ are set equal, $B_{\bar{U}} = B_{\bar{D}}$, such that there is a single B parameter for the sea distributions. The strange quark distribution is already present at the starting scale and it is assumed that $x\bar{s} = f_s x\bar{D}$ at Q_0^2. The strange fraction is chosen to be $f_s = 0.31$, which is consistent with determinations of this fraction using neutrino induced di-muon production. In addition, to ensure that $x\bar{u} \to x\bar{d}$ as $x \to 0$, $A_{\bar{U}} = A_{\bar{D}}(1-f_s)$. The D and E parameters are introduced one by one until no further improvement in χ^2 is found. This procedure allows for more flexibility when adding more precision data into the fit, for example when adding HERA-II data. The best fit results in a total of 10 free parameters when performing fits to solely HERA I data (fits are then referred to as HERAPDF1.0), and of 13 free parameters when adding preliminary HERA II data on top (fits are then referred to as HERAPDF1.5).

In HERAPDF fits the assumption is made that the strange sea quark density follows the same shape as the down quark density. However, and alternative functional form is provided to parametrise the strange density at the starting scale:

$$xs(x) = f_s \frac{1}{1 + \tanh(-(x - x_{hs}h_{hr}))}, \tag{33}$$

with $x_{hs} = 0.07$ and $h_{hr} = 20$, corresponding to a sharp turn on of the strange density at $x \sim 0.07$. This form is inspired by the HERMES analysis [74] which measure semi-inclusive production of the strange mesons.

Strange style

For studies of the strange quark sea density, the parametrisation of $x\bar{D}$ is replaced by a sum of $x\bar{d}(x) + x\bar{s}(x)$ parametrised densities:

$$x\bar{s}(x) = \frac{fs}{1-fs} A_{\bar{d}} x^{B_{\bar{s}}} (1-x)^{C_{\bar{s}}}. \tag{34}$$

Flexible style

Flexible style is the extension of the "HERAPDF style" by allowing 2 extra free parameters for every PDF distribution, namely the D and E parameters which give flexibility in the medium x range. This can be used to study the sensitivity of the data to the PDF description. In this case the total number of free parameters is 22.

CTEQ style

$$xf(x) = a_0 x^{(a_1+n)}(1-x)^{a_2} e^{a_3 x}(1 + e^{a_4} x + e^{a_5} x^2), \tag{35}$$

3.1.2 Bi-Log-Normal Functional Form

A bi-log-normal distribution is proposed by [75] to parametrise the x dependence of the PDFs. This parametrisation is motivated from multi-particle statistics. The follwing parametrisation is proposed as a general ansatz.

$$xf(x) = x^{p-b\log(x)}(1-x)^{q-\log(1-x)}. \tag{36}$$

This function can be regarded as a generalisation of parametrisations commonly used by global fit groups. In order to satisfy the QCD sum rules this parametric form requires numerical integration.

3.1.3 Chebyshev Polynomial Functional Form

A flexible Chebyshev polynomial based parametrisation can be used for the gluon and sea densities. The polynomials use $\log x$ as an argument to emphasize the low x behavior. The parametrisation is valid for $x > x_{min} = 1.7 \times 10^{-5}$. The PDFs are multiplied by $1 - x$ to ensure that they vanish as $x \to 1$. The resulting parametric form is

$$\begin{aligned} xg(x) &= A_g (1-x) \sum_{i=0}^{N_g-1} A_{g_i} T_i \left(-\frac{2\log x - \log x_{min}}{\log x_{min}} \right), & (37) \\ xS(x) &= (1-x) \sum_{i=0}^{N_S-1} A_{S_i} T_i \left(-\frac{2\log x - \log x_{min}}{\log x_{min}} \right). & (38) \end{aligned}$$

Here the sum over i runs up to $N_{g,S} = 15$ order Chebyshev polynomials of the first type T_i for the gluon, g, and sea-quark, S, density, respectively. The normalisation A_g is given by the momentum sum rule.

The advantages of this parametrisation are that the momentum sum rule can be evaluated analytically and that for $N \geq 5$ the fit qulaity is already similar to a standard Regge-inspired parametrisation with a similar number of parameters.

3.1.4 Diffractive parametrisation Functional Form

Pomeron parametrisation

The Pomeron is parametrised at the initial Q_0^2 in terms of two singlet distributions, f_g and f_+, which evolve as follows:

$$\frac{d}{dt}f_+ = \frac{\alpha_s}{2\pi}\left[\mathcal{P}_{\mathrm{FF}}f_+ + \mathcal{P}_{\mathrm{FG}}f_g\right] \tag{39a}$$

$$\frac{d}{dt}f_g = \frac{\alpha_s}{2\pi}\left[\mathcal{P}_{\mathrm{GF}}f_+ + \mathcal{P}_{\mathrm{GG}}f_g\right] \tag{39b}$$

As $I\!P$ is neutral, $f_q = f_{\bar{q}}$ for each flavour q. Assuming that all light quark PDFs are equal

$$f_d = f_u = f_s\,, \tag{40}$$

we have

$$f_{q-} \equiv 0 \tag{41a}$$

$$f_{q+} \equiv 2f_q \tag{41b}$$

At $n_f = 3$

$$f_{q+} = f_+/3,\ q = d, u, s\,. \tag{42}$$

i.e.

$$\tilde{f}_{q+} \equiv f_{q+} - \frac{1}{n_f}f_+ = 0,\ \text{for } q = d, u, s\,. \tag{43}$$

This gives all PDFs for the FFNS, while for VFNS f_{h+} for $h = c, b, t$ are generated dynamically above the respective transition scales Q_h^2. Hence at $n_f > 3$ the singlet has contributions from the heavy quarks and we get non-trivial nonsinglet distributions $\tilde{f}_{h+}$ satisfying

$$\frac{d}{dt}\tilde{f}_{h+} = \frac{\alpha_s}{2\pi}\mathcal{P}_{(+)}\tilde{f}_{h+} \tag{44}$$

Parametrisation at Q_0^2

Full PDFs are given in analogy to Eq. 28

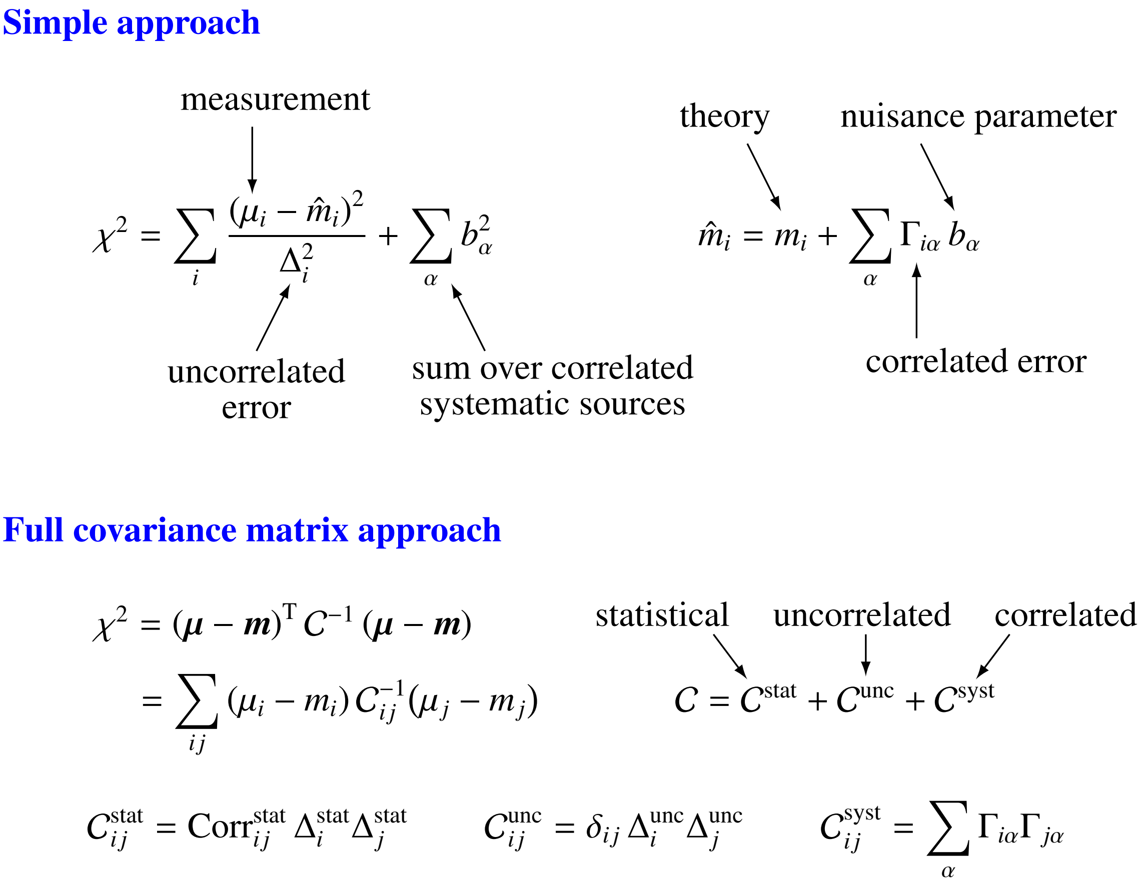
$$f_k^{D(3)}(\beta, Q^2, \xi) = \hat{\Phi}_{I\!P}(\xi)\, f_k^{I\!P}(\beta, Q^2) + \Phi_{I\!R}(\xi)\, f_k^{I\!R}(\beta, Q^2) \tag{45}$$

where $\hat{\Phi}_{I\!P} \equiv \Phi_{I\!P}/A_{I\!P}$, with the fluxes given by Eq. 29 and Eq. 30.

The Pomeron PDFs are parametrised as

$$f_N^{I\!P} = A_1^{(N)}x^{A_2^{(N)}}(1-x)^{A_3^{(N)}}\ \exp\left(-\frac{d}{1.00001-x}\right), \tag{46}$$

where the ‘dumping factor’ d is taken as 0.01 or 0.001. N = G for gluon and N = S for ‘singlet’: $f_{\mathrm{S}} \equiv f_+(n_f = 3)$, cf. Eq. 42.



$$\chi^2 = \sum_i \frac{(\mu_i - \hat{m}_i)^2}{\Delta_i^2} + \sum_\alpha b_\alpha^2 \qquad \hat{m}_i = m_i + \sum_\alpha \Gamma_{i\alpha}\, b_\alpha$$

$$\chi^2 = (\boldsymbol{\mu} - \boldsymbol{m})^{\mathrm{T}}\, C^{-1}\, (\boldsymbol{\mu} - \boldsymbol{m}) = \sum_{ij} (\mu_i - m_i)\, C^{-1}_{ij} (\mu_j - m_j) \qquad C = C^{\mathrm{stat}} + C^{\mathrm{unc}} + C^{\mathrm{syst}}$$

$$C^{\mathrm{stat}}_{ij} = \mathrm{Corr}^{\mathrm{stat}}_{ij}\, \Delta^{\mathrm{stat}}_i \Delta^{\mathrm{stat}}_j \qquad C^{\mathrm{unc}}_{ij} = \delta_{ij}\, \Delta^{\mathrm{unc}}_i \Delta^{\mathrm{unc}}_j \qquad C^{\mathrm{syst}}_{ij} = \sum_\alpha \Gamma_{i\alpha} \Gamma_{j\alpha}$$

Figure 7: Various χ^2 representations in `xFitter`.

3.2 Chisquare Definition and Treatment of Experimental Uncertainties

In this section various forms of χ^2 allowing for the inclusion of systematic and statistical correlations are presented. They are based on the use of nuisance parameters or on the full covariance matrix. A schematic picture of χ^2 definitions is displayed in Fig. 7. The description starts with the most simple cases and extends to more evolved forms, which take into account possible biases arising from low statistics data.

3.2.1 Using Nuisance Parameters

In this subsection the focus is on the form of χ^2 using nuisance parameters, which take into account the correlated systematic error sources. Several variants are briefly discussed. For more detailed presentations see Refs. [76–78].

From the statistical point of view a measurement result, μ_i, at point i is a random variable which can be modelled as

$$\mu_i = m_i(\boldsymbol{p}) + r_i \sigma_i + \sum_{\alpha=1}^{N_{\mathrm{syst}}} \Gamma^i_\alpha b_\alpha \tag{47}$$

where
$m_i(\boldsymbol{p})$ is the 'true', physical model value depending on parameters $\boldsymbol{p} = (p_1, p_2, \ldots)$,

σ_i describes the statistical and uncorrelated systematic uncertainties,
Γ^i_α quantifies the sensitivity of the i-th measurement to the correlated systematic error source α,
r_i are normal random variables (fluctuating around 0 with unit dispersion).
Finally b_α are nuisance parameters, quantifying the strength of correlated error source α.

The probability density to obtain a measurement $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots)$ is proportional to $e^{-\chi^2(\boldsymbol{m},\boldsymbol{b})/2}$ and the model parameters $\boldsymbol{p}$ are obtained via the minimisation of χ^2. To this end the nuisance parameters must be determined. In the following, different forms of χ^2, as well as different approaches to the nuisance parameters are presented. Moreover, the relative rather than absolute uncertainties are used, e.g. $\Gamma^i_\alpha = \gamma^i_\alpha \mu_i$ or $\sigma_i = \sqrt{\delta^2_{i,\text{stat}} + \delta^2_{i,\text{uncor}}}\,\mu^i$.

Simple Form

For a single data set, the χ^2 function can be defined in a simple additive form

$$\chi^2_{\text{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{\left[m^i - \sum_\alpha \gamma^i_\alpha \mu^i b_\alpha - \mu^i\right]^2}{\left(\delta_{i,\text{stat}}\,\mu^i\right)^2 + \left(\delta_{i,\text{uncor}}\,\mu^i\right)^2} + \sum_\alpha b^2_\alpha. \tag{48}$$

where the $\boldsymbol{m}$ dependence on $\boldsymbol{p}$ has been suppressed. In the formula, $\delta_{i,\text{stat}}$ and $\delta_{i,\text{uncor}}$ denote the relative statistical and relative uncorrelated systematic uncertainties, respectively. The latin index i runs over the data points, while the greek index α runs over the correlated systematic error sources.

This formula for χ^2, as well as the following ones, is obtained under the assumption of normal distribution of the nuisance parameters. This assumption results in the trailing term, $\sum_\alpha b^2_\alpha$, expressing the penalty for correlated shifts away from the central values.

Scaled Form

Equation 48 can be evolved as in [78]:

$$\chi^2_{\text{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_i \frac{\left[m^i - \sum_\alpha \gamma^i_\alpha m^i b_\alpha - \mu^i\right]^2}{\delta^2_{i,\text{stat}}\,\mu^i m^i \prod_\alpha \exp\left(-\gamma^i_\alpha b_\alpha\right) + \left(\delta_{i,\text{uncor}}\,m^i\right)^2} + \sum_\alpha b^2_\alpha\,. \tag{49}$$

The above definitions of χ^2 use systematic uncertainties that are proportional to the central values (multiplicative errors) whereas the statistical errors scale with the square roots of the expected number of events. Other scaling properties for the statistical and uncorrelated systematic uncertainties are discussed later.

Covariance Matrix

In the case of correlated (off-diagonal) statistical uncertainties, the χ^2 function reads

$$\chi^2_{\text{exp}}(\boldsymbol{m}, \boldsymbol{b}) = \sum_{ij}\left(m^i - \sum_\alpha \Gamma^i_\alpha(m^i)\,b_\alpha - \mu^i\right) C^{-1}_{\text{stat}\;ij}(m^i, m^j)\left(m^j - \sum_\alpha \Gamma^j_\alpha(m^j)\,b_\alpha - \mu^j\right) + \sum_\alpha b^2_\alpha\,. \tag{50}$$

Here the scaling properties of the correlated systematic uncertainties Γ^i_α and of the covariance matrix C_{stat} are expressed in terms of m^i, and the dependence of C_{stat} on b_α is ignored. The uncorrelated statistical uncertainties if provided are included in the matrix C_{stat} in the equation above.

Three methods of treatment of the nuisance parameters are implemented in `xFitter`. Quadratic dependence of χ^2 on b_α allows for a fast determination of the minimum, without the need to include formal nuisance parameters into the Minuit minimisation.

In the first method a minimisation of Eq. 50 wrt. b_α is used to define the covariance matrix for the systematic uncertainties, which is determined as

$$C_{\mathrm{syst}\;ij} = \sum_\alpha \Gamma^i_\alpha \Gamma^j_\alpha\,. \tag{51}$$

The total covariance matrix is given by the sum of the statistical and systematic covariance matrices

$$C_{\mathrm{tot}} = C_{\mathrm{stat}} + C_{\mathrm{syst}}\,, \tag{52}$$

and the χ^2 function takes the form

$$\chi^2(\boldsymbol{m}) = \sum_{ij} (m^i - \mu^i)\, C^{-1}_{\mathrm{tot}\;ij}\, (m^j - \mu^j)\,. \tag{53}$$

The second method is used to determine optimal shifts of the nuisance parameters at each iteration. The minimisation wrt. $\boldsymbol{b}$ leads to a system of linear equations

$$\sum_\beta \sum_{ij} C^{-1}_{\mathrm{stat}\;ij} \Gamma^i_\alpha \Gamma^j_\beta\, b_\beta + b_\alpha = \sum_{ij} C^{-1}_{\mathrm{stat}\;ij} \Gamma^i_\alpha (m^j - \mu^j)\,, \tag{54}$$

where, $1 \le \alpha \le N_{\mathrm{syst}}$, the total number of correlated systematic uncertainties. The methods given by Eq. 53 and Eq. 54 are equivalent algebraically but Eq. 54 is more efficient numerically when the number of nuisance parameters is smaller than the number of measurements. Additional advantage of this method is that it provides information on the nuisance parameter shifts and constraints on them which could be important for other applications, e.g. PDF profiling (see Section 4.2).

In the third approach the nuisance parameters $\boldsymbol{b}$ are excluded from the χ^2 minimisation. In this case, which is referred to as the offset method, the minimum is determined for the values of $\boldsymbol{b}$ set to zero while uncertainties on the parameters $\boldsymbol{p}$ are determined by shifting each nuisance parameter b_α by ±1 (one standard deviation). The total covariance matrix for parameters p^i is determined as

$$C^{\mathrm{offset}}_{\mathrm{par}\;ij} = \sum_{\alpha=1}^{N_{\mathrm{syst}}} \Delta p^i_\alpha \Delta p^j_\alpha\,, \tag{55}$$

where $\Delta p^i_\alpha = 0.5(p^i(b_\alpha = +1) - p^i(b_\alpha = -1))$ and the quality of the fit is estimated by fixing $\boldsymbol{p}$ to the values determined at the minimum and minimising with respect to $\boldsymbol{b}$.

Finally, all three approaches can be combined together. For example, some of the systematic uncertainties can be treated using the matrix method while others can be treated using the Hessian method. In this case, the covariance matrix C_{syst} is build using the corresponding sub-set of systematic sources and C_{stat} is replaced by $C_{\mathrm{stat}} + C_{\mathrm{syst}}$ in Eq. 50. Similarly, some of the systematic uncertainties can be treated using offset method and then $C^{\mathrm{total}}_{\mathrm{par}} = C^{\mathrm{Hessian}}_{\mathrm{par}} + C^{\mathrm{offset}}_{\mathrm{par}}$ where offset and Hessian covariance matrices are calculated using corresponding systematic error sources.

Bias corrections

The correlated and uncorrelated systematic uncertainties can be treated as additive, $\Gamma^i_\alpha(m^i) = \gamma^i_\alpha \mu^i$ or multiplicative, $\Gamma^i_\alpha(m^i) = \gamma^i_\alpha m^i$.

The statistical uncertainties can be treated as additive, $\Delta^i(m^i) = \delta^i \mu^i$ or as Poisson, $\Delta^i(m^i) = \delta^i \sqrt{\mu^i m^i}$. More complex scaling from Eq. 49, which depends on shifts of b_α, is implemented using an iterative approach: for the first iteration $b_\alpha = 0$ is used to determine values of b_α which are then applied in the second iteration. The statistical covariance matrix is scaled in a similar manner. In this case the correlation matrix is assumed to be fixed, the diagonal elements are updated using the prescription describe above and the covariance matrix is rescaled accordingly.

CHI2SettingsName:	StatScale	UncorSysScale	CorSysScale	Scaling rule
CHI2Settings				
Poisson	+	+	−	$\sqrt{m^i \mu^i}$
Linear	−	+	+	m^i
NoRescale	+	+	+	μ^i
LogNorm	Reserved, not implemented			

Table 1: Global scaling rules for statistical, uncorrelated and correlated systematic uncertainties. The scaling rule is given with respect to corresponding relative uncertainty. E.g. for the Poisson statistical uncertainty the absolute statistical uncertainty is $\Delta_i = \delta_{i,\mathrm{stat}} \sqrt{m^i \mu^i}$

CorChi2Type value	Description
Hessian	Use nuisance parameters.
Matrix	Use covariance matrix.
Offset	Use offset method

Table 2: Possible values of the CorChi2Type parameter which defines treatment of the correlated systematic uncertainties.

The modifications of the covariance matrix at each iteration of the Minuit minimisation may lead to systematic biases. There are two approaches that can be used to avoid these biases. In the first approach the covariance matrix is calculated using the expected values at the first iteration of the minimisation and kept fixed to these values for further iterations. This method requires several repetitions of the minimisation, to ensure that values close to optimal are obtained already at the first iteration. The second method [79] modifies the χ^2 function by adding a term corresponding to a non constant value of the covariance matrix:

$$\chi^2_{\log} = 2 \log \frac{\Delta^i(m^i)}{\Delta^i(\mu^i)} \tag{56}$$

3.2.2 `xFitter` implementation

The form of the χ^2 function and the scaling properties of the uncertainties are controlled globally by the CHI2SettingsName and Chi2Settings variables and individually using ":" modifiers. The global scaling properties of the uncertainties are described in Table 1. The global form of the χ^2 function is defined by the CorChi2Type parameter, see Table 2.

The default behavior can be changed for each correlated systematic source by ":" modifiers. They are described in Table 3. The modifiers should appear at the end of the systematic source name, e.g. 'H3:M'. Several modifiers can be used, e.g. 'H3:M:C'.

The names of systematic error sources are read first from the ListOfSources variable of the &Systematics namelist, located in the steering.txt file. Next the names are read from the data files following the sequence given by the InputFileNames list. The properties of each systematic error source are defined by its first occurrence. That means that if, for example, 'H3:M:C' is defined in the ListOfSources variable, the source 'H3' is treated as multiplicative and using covariance matrix approach regardless definitions in the data files. If, however, ListOfSources defines a source without any modifiers, e.g. 'H3', the default treatment, following the Chi2Settings variable is enforced for this source. Thus the ListOfSources variable is a convenient way to modify behavior of the correlated systematic sources.

Modifier	Description
	Scaling properties
:M	Multiplicative scaling, m^i
:A	Additive scaling, μ^i
:P	Poisson scaling, $\sqrt{m^i \mu^i}$
	χ^2 treatment
:N	Nuisance parameter treatment
:C	Covariance matrix treatment
:O	Offset method treatment
:E	Nuisance parameter, included in MINUIT ("External")

Table 3: Modifiers for correlated systematic uncertainty sources.

The shifts of the systematic sources are reported in the RESULTS.TXT file. The uncertainty on the shift is however estimated only approximately, neglecting the correlation with the theory parameters. An accurate determination of the uncertainty can be achieved by using the toy MC method (see section 3.2.3) or by using ':E' modifier. In the latter case the systematic source is treated using the MINUIT minimisation. Note, however, that this approach can slow the minimisation convergence considerably.

When CORCHI2TYPE is set to OFFSET all fits are run in a single job, each fit driven by initial parameters read from `minuit.in.txt`. Two optional parameters can be set in the CSOFFSET namelist, e.g.

```
&CSOffset
 CorSysIndex  =  0
 UsePrevFit = 1
&End
```

Setting `CorSysIndex` to any value $\in [-N_{\text{syst}}, N_{\text{syst}}]$ restricts the job to a single fit to data shifted (down or up) by the corresponding correlated error source. `CorSysIndex` = 0 corresponds to the central fit. If `CorSysIndex` is omitted then all fits are performed.

The parameter `UsePrevFit` determines how to use the results of previous fits, if such results are present in the `output` folder.

0 — Do not use any previous fit results (Default)

1 — Use previously obtained parameters as starting values for the current fit. Read initial parameters from `minuit.save_<CSI>.txt` — e.g. `minuit.save_001m.txt` for `CorSysIndex` = −1. If the file does not exist and `CorSysIndex` ≠ 0 try to read `minuit.save_0.txt`.

2 — Do not perform the fit if a corresponding `Results_<CSI>.txt` file exists, otherwise switch to mode 1.

3.2.3 Monte Carlo Method

The PDF uncertainties can be estimated using a Monte Carlo technique [80, 81]. The method consists in preparing replicas of data sets by allowing the central values of the cross sections to fluctuate within their systematic and statistical uncertainties taking into account all point-to-point correlations. The preparation of the data is repeated for a large N ($>$ 100 times) and for each of these replicas a NLO QCD fit is performed to extract the PDF set. The PDF central values and uncertainties are estimated using the mean values and RMS over the replicas.

3.2.4 Implementation in xFitter

The steering flags to activate the MC method are located in the steering.txt via:

```
&MCErrors

  lRAND    = False
  lRANDDATA = True
  ISeedMC = 123456
  ! --- Choose what distribution for the random number generator
  ! STATYPE (SYS_TYPE)  =   1  gauss
  ! STATYPE (SYS_TYPE)  =   2  uniform
  ! STATYPE (SYS_TYPE)  =   3  lognormal
  ! STATYPE (SYS_TYPE)  =   4  poisson (only for lRANDDATA = False !)
  STATYPE =  1
  SYSTYPE =  1
&End
```

To activate the MC method for error estimation set `lRand = True` . To use data (true, default) or theory (false) for the central values of the MC replica the the flag `lRANDDATA` is used. The seed for random number generation is selected via `ISeedMC` . The smearing of the uncertainties can be treated differently for correlated or uncorrelated source and four distributions are supported for random number generators: Gauss, uniform, lognormal, and Poisson. If the flags are set to 0 then no smearing is produced.

3.2.5 Regularisation methods

Regularisation methods are aiming to study the parametrisation assumptions of PDFs. When more flexible parametrisation styles is used the shape of the PDFs must be constrained and various methods are used. The xFitter framework provides the means to study and compare various methods.

Data Driven Regularisation

This method was first applied by the NNPDF group. It uses a redundant paremeters and introduces a stopping criterion based on data. This method splits data randomly into "fit" and "control" samples. The "fit" sample is used to determine the PDF parameters. The χ^2 of this sample is observed to decrease semi-monotonically. The "control" sample is used to protect against overfitting and for this sample the χ^2 will first decrease and then will start to increase due to fluctuation of the data.

External Regularisation based on a penalty term in χ^2

Another method to constrain the PDF shape is to simply apply a penalty term to the χ^2 function. One method is the so called "length penalty" which selects PDF solutions with a smoother shape in $W \approx Q\sqrt{\frac{1-x}{x}}$:

$$L = \int_{W_{min}}^{W_{max}} \sqrt{1 + \left(\frac{dxf(W)}{dW}\right)^2} dW \qquad (57)$$

This method can be applied when using Chebyshev polynomials to parametrise PDFs. For more details, the reader is invited to consult reference [82]. This method is implemented in xFitter via steering flags under Namelist `&Cheb` as follows:

```
&Cheb
  ! Set following > 0 to turn on:
   NCHEBGLU = 0   ! number of parameters for the gluon (max 15)
   NCHEBSEA = 0   ! number of parameters for the sea   (max 15)

  ! Cheb. polynomial type: multiply by (1-x) (1) or not (0)
   ichebtypeGlu = 1
   ichebtypeSea = 1

  ! Starting point in x:
   chebxmin = 1.E-5

   ILENPDF  = 0   ! use pdf length constraint

  ! PDF length constraint strength for different PDFs:
   PDFLenWeight = 1., 1., 1., 1., 1.

  ! Range in W where length constraint is applied:
   WMNLen =  20.
   WMXLen = 320.
&end
```

Alternatively, when using the flexible parametrisation style a χ^2 penalty term can be applied to account for th deviation from a simpler parametric form, for example

$$\chi^2_{reg} = T \sum_f \left(\left(\frac{D_f}{\Delta_D} \right)^2 + \left(\frac{E_f}{\Delta_E} \right)^2 \right), \tag{58}$$

with $\Delta D = \Delta E = 100$, such that for large D and E the ratio will approach 1. T is the regularisation parameters, such that for $T = 0$ there is no penalty term. For large T there is strong penalty. This method of regularisation is accessed via Namelist: `&ExtraMinimisationParameters` with the name `'Temperature'` which is T in the above description.

4 Alternative Methods: Profiling and Reweighting

In this section a alternative approaches to PDF studies are described. These are PDF profiling and reweighting. The later allows to update probability distribution based PDF (e.g. NNPDF PDF sets) to be updated with new data inputs. PDF profiling is a bit more universal, it compares data and MC predictions and based on the χ^2-minimization technique implemented in the `xFitter`, it constraints the individual PDF eigenvector sets of the input PDFs taking into account also the data uncertainties. Both of these techniques are implemented in the `xFitter` and are described below.

4.1 Bayesian Reweighting

Bayesian reweighting of PDF sets is a way to include new data into an existing PDF set without actually carrying out a full-blown fitting procedure. It was first suggested by Giele and Keller [80] and first pursued in practice by the NNPDF Collaboration [83, 84]. Watt and Thorne [85] have also proposed a scheme for how to implement the Bayesian reweighting technique for PDF predictions based on central values with errors determined using the Hessian Eigenvector Method.

The `xFitter` package allows these methods to be used to update any PDF that is available either as a probability distribution (i.e. a `LHAPDF`.LHgrid file in NNPDF format) or as a PDf eigenvector set (i.e. any PDF set in `LHAPDF`.LHgrid file format with errors determined using the Hessian Eigenvector Method). This enables the user to assess the impact of new data not only on the `HERAPDF` using the full-blown fit procedure but also on other standard PDF sets. This one can investigate how the data impact different PDF sets.

The Bayesian Reweighting technique essentially uses PDF probability distributions as input, applies weights to these distributions based on how well the new data is described and outputs an updated PDF probability distribution. In the following paragraphs, firstly the construction of these PDF probability distributions is described, then the calculation of the weights to update the PDF probability distribution is introduced and lastly, the configuration of the module within the `xFitter` framework is explained.

4.1.1 PDF probability distributions

PDF probability distributions are constructed as finite ensembles of N_{rep} parton distribution functions PDF_k, $\mathcal{E} = \{PDF_k, k = 1, ..., N_{\text{rep}}\}$. Observables $O(\text{PDF})$ are conventionally calculated from the average of the predictions obtained from the ensemble:

$$\langle O(\text{PDF})\rangle = \frac{1}{N_{\text{rep}}}\sum_{k=1}^{N_{\text{rep}}} O(\text{PDF}_k) \tag{59}$$

Their uncertainties are calculated as the standard deviation, defined as:

$$\sigma_{O(\text{PDF})} = \sqrt{\frac{1}{N_{\text{rep}}-1}\sum_{k=1}^{N_{\text{rep}}}(O(\text{PDF}_k) - \langle O(\text{PDF})\rangle)^2} \tag{60}$$

While the standard PDF sets from the NNPDF collaboration are already available as ensembles of parton distribution functions, the PDF predictions of other PDF fitting groups need to be converted to PDF probability distributions. This is possible provided that the PDF sets have associated uncertainties that can be used to create replicas of the central PDF set with random variations that lie within the uncertainties.

In the case of uncertainties provided by standard Hessian eigenvector error sets, this can be easily achieved by creating the k-th random replica by introducing introducing random fluctuations around the central PDf set, PDF_0.

If the PDF eigenvectors are asymmetric, that is they come in pairs of negative and positive PDF error sets, corresponding to negative and positive deviations from the central value, these random fluctuations can be created by drawing a random number R_{jk} and adding, depending on the sign of the random number, the difference of the positive or respectively negative PDF of the j-th PDF eigenvector pair from the central value, scaled by the absolute value of the random number:

$$\text{PDF}_k = \text{PDF}_0 + \sum_{j=0}^{n}\left[\text{PDF}_j^{\pm} - \text{PDF}_0\right]|R_{jk}| \tag{61}$$

Here, k denotes the number of the random replica and runs from $k = 1, ..., N_{\text{rep}}$; j denotes the eigenvector pair and runs from $j = 1, ..., n$, where n is the number of eigenvectors, e.g. $n = 20$ for MSTW08.

In case that the Hessian eigenvectors have been symmetrised and only one error set is given per eigenvector, the above prescription simplifies to:

$$\text{PDF}_k = \text{PDF}_0 + \sum_{j=0}^{n} \left[\text{PDF}_j - \text{PDF}_0\right] R_{jk} \tag{62}$$

4.1.2 Bayesian Reweighting of PDF sets

Once PDF probability distributions are available as inputs, they can be updated to incorporate the new data. This is achieved by applying weights to the PDF probability distributions such that the prediction for observable $\langle O(\text{PDF})\rangle$ from Eq. 59 changes to:

$$\langle O^{\text{new}}(\text{PDF})\rangle = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} w_k O(\text{PDF}_k) \tag{63}$$

The weights w_k calculated are here according to:

$$w_k = \frac{(\chi_k^2)^{\frac{1}{2}(N_{\text{data}}-1)} \exp^{-\frac{1}{2}\chi_k^2}}{\frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} (\chi_k^2)^{\frac{1}{2}(N_{\text{data}}-1)} \exp^{-\frac{1}{2}\chi_k^2}}, \tag{64}$$

where N_{data} is the number of new data points, k denotes the specific replica for which the weight is calculated and χ_k^2 is between a given data point y_i and its theoretical prediction obtained with the k-th PDF replica:

$$\chi^2(y, \text{PDF}_k) = \sum_{i,j=0}^{N_{\text{data}}} (y_i - y_i(\text{PDF}_k))\sigma_{ij}^{-1}(y_j - y_j(\text{PDF}_k)) \tag{65}$$

The weighted PDF probability distribution can be turned into a new ensemble of PDF replicas, based on which predictions for any observable can be calculated. This new, reweighted PDF probability distribution commonly is chosen to be based upon a smaller number of PDF sets compared to the input PDF probability distribution, because replicas that are incompatible with the data are discarded in order to create a more stream-lined PDF set.

4.2 Profiling

The impact of a new data set on a given PDF set can be quantitatively estimated with a profiling procedure [86]. The profiling is performed using a χ^2 function which includes both the experimental uncertainties and the theoretical uncertainties arising from PDF variations:

$$
\begin{aligned}
&\chi^2(\boldsymbol{b}_{\mathbf{exp}}, \boldsymbol{b}_{\mathbf{th}}) = \\
&\quad \sum_{i=1}^{N_\mathrm{data}} \frac{\left(\sigma_i^\mathrm{exp} + \sum_\alpha \Gamma_{i\alpha}^\mathrm{exp} b_{\alpha,\mathrm{exp}} - \sigma_i^\mathrm{th} - \sum_\beta \Gamma_{i\beta}^\mathrm{th} b_{\beta,\mathrm{th}}\right)^2}{\Delta_i^2} \\
&\quad + \sum_\alpha b_{\alpha,\mathrm{exp}}^2 + \sum_\beta b_{\beta,\mathrm{th}}^2 \,. \qquad (66)
\end{aligned}
$$

The correlated experimental and theoretical uncertainties are included using the nuisance parameter vectors $\boldsymbol{b}_{\mathbf{exp}}$ and $\boldsymbol{b}_{\mathbf{th}}$, respectively. Their influence on the data and theory predictions is described by the $\Gamma_{i\alpha}^\mathrm{exp}$ and $\Gamma_{i\beta}^\mathrm{th}$ matrices. The index i runs over all N_data data points, whereas the index α (β) corresponds to the experimental (theoretical) uncertainty nuisance parameters. The measurements and the uncorrelated experimental uncertainties are given by σ_i^exp and Δ_i , respectively, and the theory predictions are σ_i^th. The χ^2 function of Eq. 66 can be generalised to account for asymmetric PDF uncertainties:

$$
\Gamma_{i\beta}^\mathrm{th} \to \Gamma_{i\beta}^\mathrm{th} + \Omega_{i\beta}^\mathrm{th} b_{\beta,\mathrm{th}} \,, \qquad (67)
$$

where $\Gamma_{i\beta}^\mathrm{th} = 0.5(\Gamma_{i\beta}^\mathrm{th+} - \Gamma_{i\beta}^\mathrm{th-})$ and $\Omega_{i\beta}^\mathrm{th} = 0.5(\Gamma_{i\beta}^\mathrm{th+} + \Gamma_{i\beta}^\mathrm{th-})$ are determined from the shifts of predictions corresponding to up ($\Gamma_{i\beta}^\mathrm{th+}$) and down ($\Gamma_{i\beta}^\mathrm{th-}$) PDF uncertainty eigenvectors.

The minimisation of Eq. 66 in its original form leads to a system of linear equations. The generalised function, with asymmetric PDF uncertainties, is minimised iteratively: the values of $\Gamma_{i\beta}^\mathrm{th+}$ are updated using $\beta_{k,\mathrm{th}}$ from the previous iteration and following the substitution of Eq. 67. Several iterations are required to converge, and the procedure is verified using the `MINUIT` program which yields identical results.

The value at the minimum of the χ^2 function provides a compatibility test of the data and theory. In addition, the values at the minimum of the nuisance parameters $b_{\beta,\mathrm{th}}^\mathrm{min}$ can be interpreted as optimisation ("profiling") of PDFs to describe the data [86]. Explicitly, the profiled central PDF set f_0' is given by

$$
f_0' = f_0 + \sum_\beta b_{\beta,\mathrm{th}}^\mathrm{min} \left(\frac{f_\beta^+ - f_\beta^-}{2} - b_{\beta,\mathrm{th}}^\mathrm{min} \frac{f_\beta^+ + f_\beta^- - 2f_0}{2} \right) , \qquad (68)
$$

where f_0 is the original central PDF set and $f_\beta^\pm$ represents the eigenvector sets corresponding to up and down variations.

The shifted PDFs have reduced uncertainties. In general, the shifted eigenvectors are no longer orthogonal, but can be transformed to an orthogonal representation using a standard diagonalisation procedure, as in Ref. [78]. In this method the covariance matrix C of the PDF nuisance parameters is diagonalised as

$$
\begin{aligned}
\boldsymbol{b}_{\mathbf{th}}^{\boldsymbol{T}} C \boldsymbol{b}_{\mathbf{th}} &= \boldsymbol{b}_{\mathbf{th}}^{\boldsymbol{T}} G^T D G \boldsymbol{b}_{\mathbf{th}} = \boldsymbol{b}_{\mathbf{th}}^{\boldsymbol{T}} (\sqrt{D}G)^T \sqrt{D} G \boldsymbol{b}_{\mathbf{th}} \\
&= (G' \boldsymbol{b}_{\mathbf{th}})^T G' \boldsymbol{b}_{\mathbf{th}} = (\boldsymbol{b}'_{\mathbf{th}})^{\boldsymbol{T}} \boldsymbol{b}'_{\mathbf{th}} \,, \qquad (69)
\end{aligned}
$$

where G is an orthogonal matrix, D is a positive definite diagonal matrix, and $\sqrt{D}$ is a diagonal matrix built of $\sqrt{D_{ii}}$. The matrices G and D can be constructed using the eigenvectors and eigenvalues of the matrix C. The transformation G' can be adjusted, using orthogonal transformations, to keep the new eigenvector basis aligned along the original as much as possible. As a result of this adjustment, the transformation matrix can take a triangular form with all diagonal elements greater than zero.

The method can be extended to PDF sets with asymmetric uncertainties: the transformation matrix is determined using symmetrised uncertainties as in Eq. 69, and the orthogonal up and down PDF eigenvectors $f_\beta^{+'}$ and $f_\beta^{-'}$ are calculated as

$$\begin{aligned} f_\alpha^{+'} &= f_0' + \sum_\beta G'_{\beta\alpha} \left(\frac{f_\beta^+ - f_\beta^-}{2} + G'_{\beta\alpha} \frac{f_\beta^+ + f_\beta^- - 2f_0}{2} \right), \\ f_\alpha^{-'} &= f_0' - \sum_\beta G'_{\beta\alpha} \left(\frac{f_\beta^+ - f_\beta^-}{2} - G'_{\beta\alpha} \frac{f_\beta^+ + f_\beta^- - 2f_0}{2} \right). \end{aligned}$$

4.3 Usage of the alternative approaches in the `xFitter` framework

The `xFitter` framework can be used to apply either PDF reweighting for NNPDF-style PDF probability distributions as well as for PDF profiling for PDF sets with Hessian PDF eigenvector error sets. The `xFitter` will automatically determine which method to use based on the specified input PDF.

This requires that the `LHAPDF` module is installed, see sections 5.1.4. In the `xFitter` steering files, as `RunningMode` the following parameter has to be chosen: `'LHAPDF Analysis'`. This will write out the following files into the `output` directory:

- **NNPDF-style PDFs**:
  - `pdf_BAYweights.dat`
  - `pdf_GKweights.dat`
- **Hessian PDFs**:
  - `pdf_vector_cor.dat`
  - `pdf_shifts.dat`
  - `pdf_rotate.dat`

These files can be used to either perform reweighting or profiling as described in the following. The module `tools/process` is used for this purpose. Therefore the user has to change into this directory for further steps.

4.3.1 `xfitter-process`: Bayesian Reweighting

To get the results as LHAPDF files, the xfitter-process has to be run in order:

bin/xfitter-process reweight usage: xfitter-process reweight number_output_replicas pdf_weights pdf_dir_in pdf_dir_out

here, the following parameters need to be given:

- `number_output_replicas` is the number of PDF sets that the replica should contain after the reweighting.

- `pdf_weights` is the file with the weights (the output of the main part of the `xFitter` so either `pdf_BAYweights.dat` or `pdf_GKweights.dat`, depending on which weighting scheme you want to use.
- `pdf_dir_in` is the directory of the input PDF set, e.g. `/share/LHAPDF/NNPDF30_nlo_as_0118`
- `pdf_dir_out` is the directory of the input PDF set, e.g. `/share/LHAPDF/NNPDF30_myNewPDFset`

two checks plots are automatically created when running the reweighting:

- `./weights.pdf`: weight distributions (used in the reweighting procedure - replicas with high weights are kept, low weight replicas are thrown out)
- `./palpha.pdf` (only for Bayesian weighting): distribution of the probability, that the uncertainties of the new data should be re-scaled by a factor of alpha. The rescaling factor alpha should therefore ideally be 1. It is essentially a measure of the compatibility of the new data with the old data (it should be around around 1, if it is larger than that, say around 1.7, then then the new data are incompatible with the ones included in the fit - 0.5 for example however suspiciously good).

To plot the results as comparions with the input data, the bin/xfitter-draw program can be run just as for the other fits, e.g. using the command:

bin/xfitter-draw reweight-BAY:output:”BAYreweighted” reweight-GK:output:”GKreweighted”

Help on the drawing program in general can be obtained running: **bin/xfitter-draw –help**

4.3.2 `xfitter-process`: Profiling

5 Program Manual

This section begins with a presentation of the installation instructions for various scenarios supported by the xFitter platform. There then follows a basic user manual which is intended to guide the user in his/her analysis.

5.1 Program Installation Instructions

To install xFitter together with the most commonly used modules it is suggested to use the default installation script:

```
https://wiki-zeuthen.desy.de/xFitter/xFitter/DownloadPage?action=AttachFile&do=view&target=install-xfitter
```

If you want to install the package manually, use the following instruciton.

5.1.1 Core Installation and Modules

The Installation Instructions are dependent on which modules are activated via the configuration option. To see complete list of options and possible modules, run

```
./configure --help
```

5.1.2 Pre-requirements

The xFitter program has been tested on various platforms:
SL6 (64 bit), Ubuntu 14.04, Mac OS. The following programs and system libraries are required:

- gfortran, gcc and g++ comiliers.
- lapack, blas libraries

The following package is required in order to build the xFitter package:

- QCDNUM [7] versions starting from qcdnum-17-01-13 should be used. These can be found at http://www.nikhef.nl/~h24/qcdnum/QcdnumDownload.html

5.1.3 Default Minimal Installation

- Makesure that QCDNUM installation directory is added to the executabel path and that LD_LIBRARY_PATH points to the QCDNUM libraries. To verify that, type

  ```
   qcdnum-config  --libdir
  ```

  and check that the printed directory is indeed included in the LD_LIBRARY_PATH list.

- Run:

  ```
  ./configure
  make
  make install
  ```

  If your system does not have root analysis package installed, you can use the xFitter core functionality, however certain packages such as xfitter-draw can not be built. The support of these programs is included by default and thus root package is required when you run configure script without additional options. You may disable root by configuring using configure --disable-root option.

  After these commands are finished, the executable bin/xfitter file should be installed

- Run a check:

  ```
  bin/xfitter
  ```

5.1.4 Installation with external packages (APPLGRID, APFEL, MELA, LHAPDF)

- Make sure that QCDNUM bin directory and libraries are included in the paths as described in the previous section

- Make sure that $PATH and $LD_LIBRARY_PATH variables point to the external package(s) enviroment.

- Run:

  ```
  # For applgrid only:
   ./configure --enable-applgrid
  # For mela only:
  # ./configure --enable-mela
  # For apfel only:
  # ./configure --enable-apfel
  # For lhapdf only
  # ./configure --enable-lhapdf
  # For all packages:
  #  ./configure --enable-applgrid  --enable-mela  --enable-apfel  --enable-lhapdf

   make
   make install
  ```

  After these commands are finished, the executable bin/xfitter file should be installed

- Run a check:

  ```
  bin/xfitter
  ```

5.1.5 Installation with HATHOR

Note that support of HATHOR is suspended in `xFitter`. Please use this option with extra caution.

- Download Hathor from

  `http://www-zeuthen.desy.de/~moch/hathor/`

  and install it according to the instructions given there (requires LHAPDF library)

- Define a variable HATHOR_ROOT such that HATHOR_ROOT points to the directory of your Hathor installation

- Install the `xFitter` as described above but configuring it with the option ”–enable-hathor” before building it

5.1.6 Installation for TMD (uPDF) in high-energy factorisation (using CASCADE)

- Installation with TMD requires Cascade and Pythia generators, they can be downloaded from `http://cascade.hepforge.org/` and `https://pythia6.hepforge.org/` respectively.

  After installation of the generator packages, the CASCADE_ROOT and PYTHIA_ROOT environment variables have be specified and point to the corresponding libraries. In the DESY afs environment the pre-installed versions of Cascade and Pythia can be used:

  ```
  export CASCADE\_ROOT=/afs/desy.de/group/alliance/mcg/public/MCGenerators/cascade/2.2.04/\$SYSNAME
  export PYTHIA\_ROOT=/afs/desy.de/group/alliance/mcg/public/MCGenerators/pythia6/425/\$SYSNAME}
  ```

  where SYSNAME = i586_rhel50 or similar.

- Run:

  ```
  ./configure --enable-updf --enable-lhapdf
  make
  make install
  ```

- use steering and MINUIT input files from ”input_steering”:

  ```
  cp input-steering/steering.txt.kt-factorisation steering.txt
  cp input-steering/minuit.in.txt.kt-factorisation minuit.in.txt
  cp input-steering/steer-ep-CASCADE steer-ep
  cp input-steering/steer_gluon-evolv steer_gluon-evolv
  ```

- edit steering.txt:

  ```
  \&CCFMFiles: give name for output grid file for uPDF
  \&\fitter\
  TheoryType = 'uPDF3' ! 'DGLAP'  -- collinear evolution
               ! 'uPDF'   -- un-integrated PDFs:
               !  uPDF1 fit with kernel ccfm-grid.dat file
               !  uPDF2 fit evolved uPDF, fit just normalisatio
               !  uPDF3 fit using precalculated grid of sigma_hat
               !  uPDF4 fit calculating kernel on fly, grid of sigma_hat
  ```

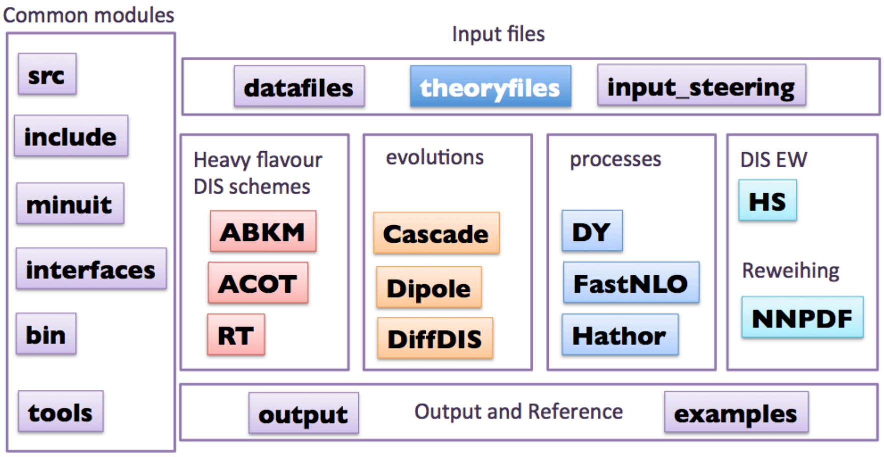


Figure 8: Schematic structure of the `xFitter` program organisation in different modules.

The recommended option is `uPDF4`, which evolves the evolution kernel for gluons and valence quarks (evolution parameters are set in `steer_gluon-evolv`). After evolution of the kernel, $\hat{\sigma}$ is calculated in a grid in x at the Q^2 values used in the data sets selected. The $\hat{\sigma}$ values are stored for transverse and longitudinal cross sections for light quarks ($n_f \leq 3$), charm and beauty quarks.

- run the program: bin/xfitter
- plotting F_2 fit results:
  xfitter-draw can be used to draw F_2 results.
  The uPDFs need to be plotted with an external package (currently not available).

5.2 User Manual

In this section a user manual is presented. The section starts with a general overview of the code organisation and it follows with a more detailed explanation for the most frequently used functions.

5.2.1 Code Organisation

A general diagram of available modules is illustrated in figure 8. The flow is depicted such that it follows the structure of the `xFitter` .

In addition, an inventory list with short description of existing subroutines is presented in Table 4. Here we choose to enlist only the routines from the common target module to guide the user of available functionalities.

steerings	• steering.txt:	free PDF parameters to be varied by MINUIT
	• minui.in.txt:	main steering card
	• ewparam.txt:	settings of electroweak parameters, as well as masses
src	• main.f:	main program
	• read_steer.f:	access steer parameters from steering card
	• read_data.f:	reading the datatables and storing data information
	• init_theory.f:	initialising theory modules
	• dataset_tools.f:	allocating bin indices
	• error_logging.f:	error logging information
	• minuit_ini.f:	initialise MINUIT module
	• fcn.f:	passes to MINUIT the χ^2 to be minimised
	• pdf_param.f:	parametrisation of the PDFs at starting scale
	• sumrules.f:	PDF constraints at starting scale, such as QCD sum rules.
	• evolution.f:	evolution of PDFs
	• theory_dispatcher.f:	distribution of theory prediction calculations for a given dataset
	• dis_sigma.f	calulates the DIS cross sections
	• GetChisquare.f	calculates the χ^2
	• GetCovChisquare.f	calculates the χ^2 using covariance matrix
	• GetPointScaledErrors.f	calculates the rescaled statistical, uncorrelated and constant errors
	• prep_corr.f	prepare systematic correlation matrix
	• systematics.f	build the matrix for systematic uncertainties and invert it
	• error_bands_pumplin.f	Hessian error calculations
	• mc_errors.f	MC method for creating replicas of data through smearing.
	• GetDiffDisXsection.f	calulates the diffractive DIS cross sections
	• FixModelParams.f	used for diffractive DIS cross sections
	• lhapdf_dum.f	(used only with ENABLE_LHPDF)
	• reweighting.f	main subroutine for PDF rewighting (used only with ENABLE_NNPDF)
	• nnpdfreweighting.f	main subroutine for NNPDF rewighting (used only with ENABLE_NNPDF)
	• dy_cc_sigma.f	calulates the DY cross sections
	• applgrids_dum.f	protective file against miss-use of flags in steering
	• fappl_grid.cxx	(used only with ENABLE_APPLGRID)
	• applgrids.f	passing PDFs to APPLGRID (used only with ENABLE_APPLGRID)
	• pp_jets_applgrid.f	calulates *pp* jets cross sections
	• ep_jets_fastnlo.f	calulates *ep* jets cross sections
	• getncxskt.f	access the NC cross sections grids for uPDFs
	• Getgridkt.f	acess the grids for uPDFs
	• ttbar_hathor_dum.f	protective file against miss-use of flags in steering
	• ttbar_hathor.f	(used only with ENABLE_HATHOR)
	• offset_fns.f	collects results from Offset method and stores them
	• g_offset.cc	file used for Offest method
	• matrix.cc	inversion of matrix as used for Offset method
	• FitPars_base.cc	file used for Offest method
	• FTNFitPars.cc	file used for Offest method
	• Xstring.cc	file used for Offest method
	• decor.cc	file used for Offest method
	• store_output.f	write the output
	• store_h1qcdfunc.f	store structure functions

Table 4: A list of main subroutines are listed with a short description of their function.

5.2.2 Steering files

The software behavior is controlled by three files with steering commands. These files have predefined names:

- `steering.txt` – controls main ”stable” (un-modified during minimisation) parameters. The file also contains names of data files to be fitted and definitions of kinematic cuts
- `minuit.in.txt` – controls minimisation parameters and minimisation strategy. Standard `MINUIT` commands can be provided in this file
- `ewparam.txt` – controls electroweak parameters such as W and Z boson masses and CKM matrix parameters.

Steering.txt

Different options are activated via steering flags in the main steering file.

The format of the steering file follows standard ”namelist” conventions. Comments start with exclamation mark (similarly used for data file format). The following namelist blocks are encountered:

- `xFitter`: Main steering cards.
- `InFiles`: Namelist to specify input data
- `InTheory`: (Optional) Namelist to provide fixed theory predictions with uncertainties as text files.
- `InCorr`: (Optional) Namelist to control statistical correlation files
- `Scales`: (Optional) Namelist to modify renormalisation/factorisation scale
- `CovarToNuisance` (Optional) Namelist to convert covariance matrix to nuisance parameter representation for data sets were some of the uncertainties are represented using covarainace or correlation matricies.
- `QCDNUM`: (Optional) Namelist to modify `QCDNUM` evolution parameters such as x and Q^2 grids.
- `Systematics`: (Optional) Namelist to modify behavior of different systematic sources.
- `ExtraMinimisationParameters`: Namelist to add extra to `MINUIT` parameters.
- `OutDir`: (Optional) allows to modify default output directory
- `Output`: Namelist that outputs steering cards
- `Cuts`: Namelist for process dependent cuts
- `MCErrors` (Optional):Namelist for MC errors steering cards
- `Cheb` (Optional): Chebyshev study namelist
- `Poly` (Optional): pure polynomial parameterisation for valence quarks
- `HQScale` (Optional): choose the factorisation scale for HQs
- `lhapdf` (Optional):`LHAPDF` steering card

These namelist blocks are described in greater details in the User’s example 5.3.

The main namelist `xFitter` specifies most important features of the program. They are controlled by the following keywords:

RunningMode:

defines the mode in which the program can be used. Possible values are:

- `RunningMode = 'Fit'` – the default mode which uses `MINUIT`-based minimisation of PDFs and other parameters.
- `RunningMode = 'LHAPDF Analysis'` – uses PDFs from the `LHAPDF` library to produce predictions for the central PDF set as well as eigenvector or MC replica variations. This mode requires `--enable-lhapdf` flag at the package configuration stage. The resulting predictions can be analysed using the `xfitter-process` program to perform PDF profiling or reweighting, following the formalism discussed in Section 4.
- `RunningMode = 'PDF Rotate'` – dedicated mode, which uses `InTheory` tabulated predictions to perform re-diagonalization of PDFs along data predictions. The output of this mode can be processed using `rotate` module of the `xfitter-process` program

TheoryType:

is a steering flag which defines the theory type via the chosen evolution. The following types are supported:

- `TheoryType = 'DGLAP'` as used for collinear evolution theories, using default `QCDNUM` evolution. Other versions of DGLAP evolution include: `TheoryType = 'DGLAP_APFEL'`, where `QCDNUM` is replaced by `APFEL`, requires `--enable-apfel`; `TheoryType = 'DGLAP_QEDEVOL'`, where `QCDNUM` evolution includes photon PDF; `TheoryType = 'DGLAP_APFEL_QED'`, where DGLAP evolution is performed by `APFEL` and includes photon PDF. For these types another flag is needed to specify the order of the perturbative series in α_S:
`Order` which can be leading order ('LO'), next-to-leading order ('NLO') and when available 'NNLO'.
- `TheoryType = 'DIPOLE'` as used for the dipole models;
- `TheoryType = 'uPDF'` as used for the un-integrated PDFs (with 4 variants)

Starting scale: The evolution starting scale is set via flag `Q02`, commonly set below charm threshold, as imposed by `QCDNUM`.

DIS Heavy Flavour Scheme type:

For the DIS process, several schemes are available for heavy quark treatments via `HF_SCHEME` flag.

- VFNS (Variable Flavour Number Schemes):
  - RT-VFNS schemes [from Robert Thorne], `HF_SCHEME = RT, RT OPT`, as well as the fast variants based on k-factors `RT FAST, RT OPT FAST`
  - `Zero Mass VFNS [qcdnum], ZM-VFNS`
  - ACOT (ACOT-Full, ACOT-ZM, S-ACOT-Chi) schemes [from Fred Olness], `HF_SCHEME = ACOT Full, ACOT Chi, ACOT ZM`, they are all based on k-factors.
  - FONLL schemes, from 'A' to 'C' with MSbar or pole masses for the heavy quarks, e.g. `HF_SCHEME = 'FONLL-A', 'FONLL-A RUNM OFF', 'FONLL-A RUNM ON'`.
- FFNS (Fixed Flavour Number Scheme)
  - via `QCDNUM`, `HF_SCHEME = FF`
  - via ABM (openqcdrad-1.6) [from Sergey Alekhin], `HF_SCHEME = FF ABM`

IMPORTANT to note if running with FFNS (nf=3):

- only neutral current DIS data should be used in FF scheme due to missing NLO coefficient functions in charged current process, in this cases valence quark parameters need to be fixed in minuit.in.txt file.
- In FF ABM implementation the charged current coefficients are available therefore valence parameters do not need to be fixed.

- $\alpha_s(Q^2)$ in FFNS is 3-flavour and recommended to be set to the value of 0.105 such that is not too high at low energies
- the scale in FFNS is defined as $\mu^2 = Q^2 + 4m_h^2$ by default, can be changed in HQScale in `steering.txt`(scale variation in ABM not yet implemented)
- the pole mass definition for heavy quarks is set in ABM by default, the running mass definition [26] can be switched in by setting `HF_SCHEME = FF ABM RUNM` in `steering.txt` .

PDFType:

can be set to 'proton' (default) or 'lead' (experimental option)

PDF parametrisation style:

There are various types of parametric functional form supported by `xFitter` . They are accessed via the steering flag called `PDFStyle`. Available styles are summarised as follows:

`'HERAPDF'`	– HERAPDF-like with u_v, d_v, $\bar{U}$, $\bar{D}$, and g evolved pdfs
`'CTEQ'`	– CTEQ-like parameterisation
`'CTEQHERA'`	– Hybrid, with valence like CTEQ, rest as HERAPDF.
`'CHEB'`	– CHEBYSHEV parameterisation based on g, Σ, u_v, d_v evolved pdfs
`'LHAPDFQ0'`	– use `LHAPDF` library to define PDFs at starting scale and evolve with local `QCDNUM` parameters
`'LHAPDF'`	– use `LHAPDF` library to define PDFs at all scales at the `QCDNUM` grid points. Uses `QCDNUM` to perform PDF interpolation and to evaluate α_S.
`'LHAPDFNATIVE'`	– native access to `LHAPDF` PDFs and α_S for all x, Q^2 values. To compute predictions for the ZMVFNS structure function this option falls back to the behaviour of '`LHAPDF`' flag.
`' DDIS'`	– use Diffractive DIS
`'BiLog'`	– bi-lognormal parametrisation

These styles were described in details in section 3.1. The `LHAPDF`-type styles can be used only with proper configuration settings, as explained in the section 5.1.

Definition of Chisquares:

The χ^2 function definitions and `xFitter` implementation are given in 3.2. Briefly, different options for various χ^2 terms are controlled by the flag `'CHI2SettingsName'` to specify the term and by the flag `Chi2Settings` to specify the options. `'CHI2SettingsName'` can have following values:

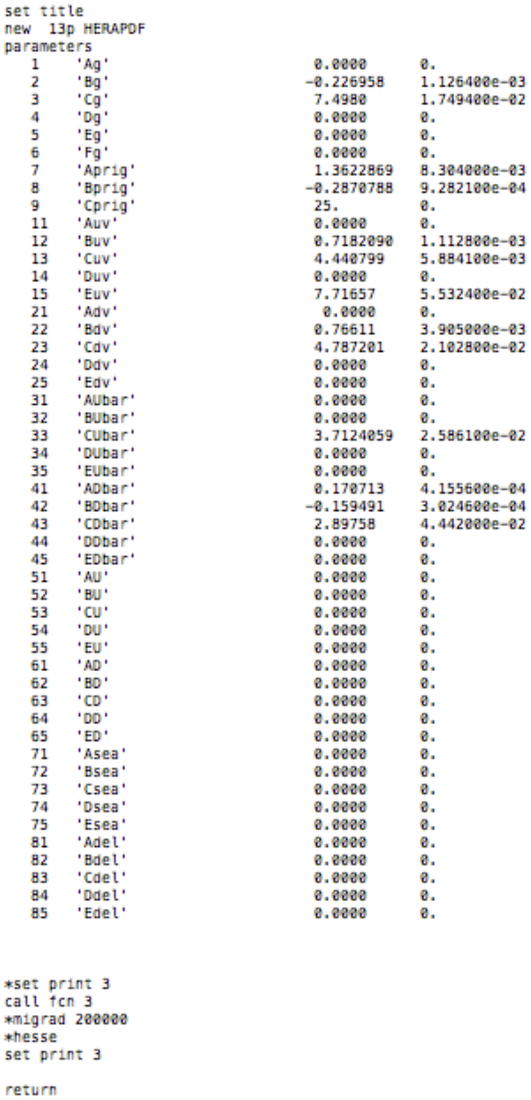
- `'StatScale'`, `'UncorSysScale'`, `'CorSysScale'` for statistical, uncorrelated, and correlated systematic uncertainties. Their scaling properties can be `'NoRescale'`, `'Poisson'` or `'Linear'`
- `'UncorChi2Type'`, `'CorChi2Type'` control treatment of systematics uncertainties correlation. They can take values of 'Diagonal', 'Hessian'

(logical) debug flag: The debug flag will be turned on for more print outs via `LDEBUG.`

Selection of the data:

The namelist &Cuts, located inside the `steering.txt` file can be used to apply simple process dependent cuts. The cuts are limited to bin variables. Simple low and high limits are allowed. For example, a cut on $Q^2 > 3.5$ GeV^2 for NC ep scattering is specified as

```
! Rule #1: Q2 cuts
 ProcessName(1)     = 'NC e+-p'
 Variable(1)        = 'Q2'
 CutValueMin(1)     = 3.5
 CutValueMax(1)     = 1000000.0
```



```
set title
new  13p HERAPDF
parameters
    1    'Ag'                          0.0000      0.
    2    'Bg'                         -0.226958    1.126400e-03
    3    'Cg'                          7.4980      1.749400e-02
    4    'Dg'                          0.0000      0.
    5    'Eg'                          0.0000      0.
    6    'Fg'                          0.0000      0.
    7    'Aprig'                       1.3622869   8.304000e-03
    8    'Bprig'                      -0.2870788   9.282100e-04
    9    'Cprig'                       25.         0.
    11   'Auv'                         0.0000      0.
    12   'Buv'                         0.7182090   1.112800e-03
    13   'Cuv'                         4.440799    5.884100e-03
    14   'Duv'                         0.0000      0.
    15   'Euv'                         7.71657     5.532400e-02
    21   'Adv'                          0.0000     0.
    22   'Bdv'                         0.76611     3.905000e-03
    23   'Cdv'                         4.787201    2.102800e-02
    24   'Ddv'                         0.0000      0.
    25   'Edv'                         0.0000      0.
    31   'AUbar'                       0.0000      0.
    32   'BUbar'                       0.0000      0.
    33   'CUbar'                       3.7124059   2.586100e-02
    34   'DUbar'                       0.0000      0.
    35   'EUbar'                       0.0000      0.
    41   'ADbar'                       0.170713    4.155600e-04
    42   'BDbar'                      -0.159491    3.024600e-04
    43   'CDbar'                       2.89758     4.442000e-02
    44   'DDbar'                       0.0000      0.
    45   'EDbar'                       0.0000      0.
    51   'AU'                          0.0000      0.
    52   'BU'                          0.0000      0.
    53   'CU'                          0.0000      0.
    54   'DU'                          0.0000      0.
    55   'EU'                          0.0000      0.
    61   'AD'                          0.0000      0.
    62   'BD'                          0.0000      0.
    63   'CD'                          0.0000      0.
    64   'DD'                          0.0000      0.
    65   'ED'                          0.0000      0.
    71   'Asea'                        0.0000      0.
    72   'Bsea'                        0.0000      0.
    73   'Csea'                        0.0000      0.
    74   'Dsea'                        0.0000      0.
    75   'Esea'                        0.0000      0.
    81   'Adel'                        0.0000      0.
    82   'Bdel'                        0.0000      0.
    83   'Cdel'                        0.0000      0.
    84   'Ddel'                        0.0000      0.
    85   'Edel'                        0.0000      0.


*set print 3
call fcn 3
*migrad 200000
*hesse
set print 3

return
```

Figure 9: An example of a `MINUIT` steering card.

Maximum 100 cuts can be used by default.

The specific input files are stored in the *input_steerings* directory and it contains the following ready to use inputs (with corresponding `MINUIT` files):

- `steering.txt.ALLdata`: all data files
- `steering.txt.DIFFRACTION`: diffraction specific settings
- `steering.txt.kt-factorisation`: kt factorisation specific settings
- `steering.txt.dipole`: dipole model specific settings

`MINUIT` steering cards

The `MINUIT` steering card is described below, a sample file is presented in Fig. 9

The first three lines set the title and specify the list of `MINUIT` parameters which are to follow. The index of parameters is the first column and it is hardwired to the source code:

1 -10	gluon parameters
11-20	uval parameters
21-30	dval parameters
31-40	Ubar parameters
41-50	Dbar parameters
51-60	U parameters
61-70	D parameters
71-80	Sea parameters
81-90	Delta parameters
91-100	other parameters: alphas (95), fs=Dbar/str (96), fc=Ubar/ch (97)

The second column represents just user defined names, the third column is the input starting value for the parameter. The forth column sets the step size (usually chosen the same order as the error). If the step size is zero this parameter is FIXED. The fifth column sets the lower bound of the fit parameter, The sixth column sets the upper bound of the fit parameter if these columns are not filled then there are no bounds.

Only parameters that have non-zero stepsize are varied in the fit (free parameters). Another way to fix the parameters is simply by typing at the end of the list of parameters “FIX parameter number”. (make sure there is one line free before the `MINUIT` list). Examples of commands taken by `MINUIT` are:

call fcn 3	fit is not performed, only 1 iteration, useful for testing `MINUIT` parameters ARE NOT minimized.
migrad	fit is performed (default number of calls 2000).
migrad 20000	fit is performed up to 20000 calls, then terminates.
hesse	Hessian estimate of the `MINUIT` parameters (more reliable than `MINUIT`

The output of the fit is stored in the output/ directory as `minuit.out.txt`. Statements in minuit.out.txt which are useful for interpreting the results of the fit:

- `FCN=575.16` this is total chisquare
- `FROM MIGRAD STATUS=CONVERGED` this is desirable for a fit that converged
- `FROM HESSE STATUS=OK` this is desirable for a fit that converged
- `ERROR MATRIX ACCURATE` errors estimated with HESSE method

`xFitter` parameters for diffractive fits

Parameter	`xFitter` name	input file
$A_1^{(G)}$	Ag	minuit.in.txt
$A_2^{(G)}$	Bg	minuit.in.txt
$A_3^{(G)}$	Cg	minuit.in.txt
$A_1^{(S)}$	Auv	minuit.in.txt
$A_2^{(S)}$	Buv	minuit.in.txt
$A_3^{(S)}$	Cuv	minuit.in.txt
$\alpha_{I\!P}(0)$	Pomeron_a0	steering.txt
$A_{I\!R}$	Reggeon_factor	steering.txt
$\alpha_{I\!R}(0)$	Reggeon_a0	steering.txt

`xFitter` parameters for dipole fits

The default initial parameters for the fit without valence quarks are :

σ_0	A_g	λ_g	C_g	$cBGK$	$eBGK$
37.490	3.3446	0.0298	2.6302	4.0	15.362

For the BGK dipole model fits with valence quarks the initial parameters and the obtained χ^2 are:

No	Q^2		σ_0	A_g	λ_g	C_g	C_{BGK}	μ_0^2	Np	χ^2/Np
1	$Q^2 \geq 3.5$	LO	66.6	4.0	-0.039	18.6	4.0	5.3	196	0.930
2	$Q^2 \geq 3.5$	NLO	79.4	3.2	-0.021	13.7	4.0	6.7	196	0.927

5.2.3 Data file format

Experimental data are provided by the standard ASCII text files. The files contain a ”header” which describes the data format and the ”data” in terms of a table. Each line of the data table corresponds to a data point, the meaning of the columns is specified in the file header.

For example, a header for HERA-I combined H1-ZEUS data for e+p neutral current scattering cross section is given in the file

```
datafiles/H1ZEUS_NC_e-p_HERA1.0.dat
```

The format of the file follows standard ”namelist” conventions. Comments start with an exclamation mark. Pre-defined variables are:

- `Name` — (string) provides the name of the data set
- `Reaction` — (string) reaction type of the data set. Reaction type is used to trigger the corresponding theory calculation. The following reaction types are currently supported by the `xFitter`:
  - `'NC e+-p'` – double differential NC ep scattering (ZM-VFNS and RT-VFNS schemes)
  - `'CC e+-p'` – double differential CC ep scattering (ZM-VFNS scheme)
  - `'CC pp'` – single differential $d\sigma_{W^\pm}/deta_{\ell^\pm}$ production and W asymmetry at pp and $p\bar{p}$ colliders (LO+kfactors and APPLGRID interface)
  - `'NC pp'` – single differential $d\sigma_Z/dy_Z$ at pp and $p\bar{p}$ colliders (`LO` with k-factors and `APPLGRID` interface)
  - ’pp jets APPLGRID’ – $pp \rightarrow$ inclusive jet production, using `APPLGRID`
  - ’pp jets FastNLO – $pp \rightarrow$ inclusive jet production, using `fastNLO`.
  - ’FastNLO jets’ – jet cross sections using `fastNLO` interface. All ep, pp and $p\bar{p}$ colliders are supported.
  - ’FastNLO ep jets normalised’ – jet cross sections in the ep collisions using `fastNLO` interface and normalised to the inclusive DIS cross sections.
- `NData` — (integer) specifies number of data points in the file. This corresponds to the number of table rows which follow after the header.
- `NColumn` — (integer) number of columns in the data table.

- `ColumnType` — (array of strings) Defines layout of the data table. The following column types are pre-defined: 'Bin', 'Sigma', 'Error' and 'Dummy' The keywords are case sensitive. 'Bin' correspond to an abstract bin definition, 'Sigma' corresponds to the data measurement, 'Error' - to various type of uncertainties and 'Dummy' indicates that the column should be ignored.

- `ColumnName` — (array of strings) Defines names of the columns. The meaning of the name depends on the ColumnType. For ColumnType 'Bin', ColumnName gives a name of the abstract bin. The abstract bins can contain any variable names, but some of them must be present for correct cross section calculation. For example, 'x', 'Q2' and 'y' are required for DIS NC cross-section calculation.

  For ColumnType 'Sigma', ColumnName provides a label for the observable, which can be any string.

  For ColumnType 'Error', the following names have special meaning:

  - 'stat' – specifies column with statistical uncertainties, request Poisson re-scaling;
  - 'stat const' – specifies column with statistical uncertainties, request no re-scaling of the errors;
  - 'uncor' – specifies column with uncorrelated uncertainties. Any name containing keyword "uncor" is treated as an uncorrelated error source, e.g. "h1 uncor";
  - 'uncor const' – specifies column with uncorrelated uncertainties, request no re-scaling of the errors;
  - 'total' – specifies column with total uncertainties. Total uncertainties are not used in the fit, however there is an additional check is performed if 'total' column is specified: sum in quadrature of statistical, uncorrelated and correlated systematic uncertainties is compared to the total and a warning is issued if they differ significantly.
  - 'ignore' - specifies column to be ignored (for special studies).
  - Other names specifies columns of correlated systematic uncertainty. For a given data file, each column of the correlated uncertainty must have unique name. To specify correlation across data files, same name must be used for different files.

- `SystScales` — (array of float) For special studies, systematic uncertainties can be scaled The numbering of uncertainties starts from the first column with the ColumnType 'Error'. For example, setting

  ```
  SystScale(1) = 2.
  ```

  in `datafiles/H1ZEUS_NC_e-p_HERA1.0.dat` would scale systematic uncertainty by factor of two.

- `Percent` — (array of bool) For each uncertainty specify if it is given in absolute ("false") or in percent ("true"). The numbering of uncertainties starts from the first column with the `ColumnType` 'Error' (see example above).

- `NInfo` — (integer) Calculation of the cross-section predictions may require additional information about the data set. The number of information strings is given by NInfo

- `CInfo` — (array of strings) Names of the information strings. Several of them are predefined for different cross-section calculations.

- `DataInfo` — (array of float) Values, corresponding to `CInfo` names.

- `IndexDataset` – (integer) Internal `xFitter` index of the data set. Provide unique numbers to get extra info for χ^2/dof for each data set.

- `TheoryInfoFile` — (string) Optional additional theory file with extra information for cross-section calculation. This could be k-factors, `APPLGRID` file or `fastNLO` table.

- `TheoryType` — (string) Theory file type can be set to 'kfactor', 'applgrid', 'fastnlo' or 'expression'. See further for more details.

- `TermInfo` — (string) list of specific options for term processors.

- `TermName` — (string) names of terms used in the theory expression.

- `TermSource` — (string) pathes from where the term numerical values should be taken.

- `TheorExpr` — (string) theory epression in simple algebraic form.

- `NKFactor` — (integer) For kfactor files, number of columns in `TheoryInfoFile`.

- `KFactorNames` — (array of strings) For kfactor files, names of columns in `TheoryInfoFile`.

Depending on the chosen process specific requirements for the header might be present. Dataset-wise options are provided by a `CInfo` / `DataInfo` variable set. In case the information varies between data points (e.g. bin borders, hadronisation corrections etc.) it is provided in the data file and recognised by the program using reserved column names. In the following all these requirements are listed and briefly explained.

The theory type *expression* allows a flexibility in theory definition making it possible to set a simple formula in `TheorExpr` string variable. The expression terms must be preliminary defined in `TermName` – a string value, starting with alphabetical character, `TermType` – recognized string values are 'kfactor', 'applgrid' or 'virtgrid', `TermInfo` – additional options for the term processing code, and `TermSource` – the files from where the predictions are taken. `TermType` defines which type of the expression term is used:

kfactor is a term wich denotes an array of K-factors corresponding to the data bins. The `TermSource` for this term must point to a file with the K-factor table, containing three columns: bin lower and higher edges, and the K-factor value. The comments starting with "#" are ignored.

applgrid term tells the parser to initialize the `APPLGRID` grid for the cross section evaluation. The grid options are defined with additional options in `DataInfo` and `CInfo`.

virtgrid can be used if the fit is performed on the multidimensional measurement results. The `TermSource` in this case is a text file with a virtual grid definition. Consider for example the data are a triple differential measurement of a cross section in X, Y and Z observables. Internally the theory prediction for such measurement is represented as a linear array – a sequence of `APPLGRID` grids for Z, each corresponding to the hyperbin in X and Y. The virtual grid file in this case is a table, where each row denotes the hyperbin with its edges, `APPLGRID` file location and number of bins in it:

X_0^{low}	X_0^{high}	Y_0^{low}	Y_0^{high}	path/to/applgrid_0_0.root	$M(0;0)$
X_0^{low}	X_0^{high}	Y_1^{low}	Y_1^{high}	path/to/applgrid_0_1.root	$M(0;1)$
…					
X_0^{low}	X_0^{high}	$Y_{N_Y}^{\text{low}}$	$Y_{N_Y}^{\text{high}}$	path/to/applgrid_0_N_Y.root	$M(0;N_Y)$
X_1^{low}	X_1^{high}	Y_0^{low}	Y_0^{high}	path/to/applgrid_1_0.root	$M(1;0)$
X_1^{low}	X_1^{high}	Y_1^{low}	Y_1^{high}	path/to/applgrid_1_1.root	$M(1;1)$
…					
$X_{N_X}^{\text{low}}$	$X_{N_X}^{\text{high}}$	Y_0^{low}	Y_0^{high}	path/to/applgrid_N_X_0.root	$M(N_X;0)$
…					
$X_{N_X}^{\text{low}}$	$X_{N_X}^{\text{high}}$	$Y_{N_Y}^{\text{low}}$	$Y_{N_Y}^{\text{high}}$	path/to/applgrid_N_X_N_Y.root	$M(N_X;N_Y)$

where N_X, N_Y are the numbers of hyperbins in X, Y repectively and M is the number of bins in the corresponding `APPLGRID`. The order of hyprebin listing must be same as for the data table and the slowest changing hyperbin must go first. In case the fitted data is a double-differential cross section, only one hyperbin needs to be listed, i.e.:

X_0^{low}	X_0^{high}	path/to/applgrid_0.root	$M(0)$
X_1^{low}	X_1^{high}	path/to/applgrid_1.root	$M(1)$
…			
$X_{N_X}^{\text{low}}$	$X_{N_X}^{\text{high}}$	path/to/applgrid_N_X.root	$M(N_X)$

The comments starting with “#” are ignored.

The expression recognises simple arithmetic operations (+,-,/,*) and ’sum()’ function which returns prediction summed over bins. Example:

```
TheoryType     = 'expression'
TermName = 'A1', 'K'
TermType = 'applgrid','kfactor'
TermSource = 'path/to/grid.root' ,
             'path/to/kfactor.txt'
TheorExpr= 'K*A1'
```

The expression also recognises numerical terms, e.g. ’K*A+0.1’, which do not require preliminary definition. Due to technical limitations, no spaces are allowed in `TheorExpr` value.

By default the numeric result of the expression is divided by the (hyper)bin width. In order to obtain initial values or use ’sum()’ operation (integral of the differential distribution e.g. for normalization purposes) one should add ‘_norm’ suffix to the the TermType of ’applgrid’ or ’virtgrid’. For example:

```
TheoryType     = 'expression'
TermName = 'V', 'A'
```

```
TermType = 'virtgrid','applgrid_norm'
TermSource = 'path/to/virtgrid.txt' ,
             'path/to/applgrid.root'
TheorExpr= 'V/sum(A)'
```

In order to obtain cross sections for a multidimensional measurement divided by the last bin width only (i.e. not divided by the hyperbin width), one should use TermType of 'applgrid_normhyperbin' or 'virtgrid_normhyperbin'.

An example of virtgrid.txt file:

```
# y1     y2     applgrid                                     n_grid_bins
0.0     0.3     theoryfiles/atlas/Jets2010-vg/R04/eta1.root      17
0.3     0.8     theoryfiles/atlas/Jets2010-vg/R04/eta2.root      17
0.8     1.2     theoryfiles/atlas/Jets2010-vg/R04/eta3.root      17
1.2     2.1     theoryfiles/atlas/Jets2010-vg/R04/eta4.root      16
2.1     2.8     theoryfiles/atlas/Jets2010-vg/R04/eta5.root      13
2.8     3.6     theoryfiles/atlas/Jets2010-vg/R04/eta6.root      10
3.6     4.4     theoryfiles/atlas/Jets2010-vg/R04/eta7.root       7
```

Data format requirements for DIS

In this subsection we describe specific requirements for files using 'NC e+-p' and 'CC e+-p' reaction types. Examples of such input files are:

`datafiles/H1ZEUS_NC_e-p_HERA1.0.dat`

`datafiles/H1ZEUS_CC_e-p_HERA1.0.dat`.

The properly formatted DIS input files will have the following fields available in the `CInfo` variable list:

- `'sqrt(S)'` — the ep collision centre-of-mass energy in GeV. In particular, for HERA based results the the corresponding `DataInfo` value should be 300. for measurements based on data collected prior to 1997 (inclusive) and 318. for data collected after 1997.
- `'reduced'` — a field indicating whether calculated cross section should be reduced (1.) or not (0.) (reference to proper equation somewhere in this manual).
- `'e charge'` — electric charge of the colliding lepton beam. Supported `DataInfo` values are '1.' for electron and '-1.' for positron.
- `'e polarity'` — polarity of the lepton beam. The corresponding `DataInfo` value should be between -1.0 and 1.0 (is this true?) with abs(1.0) indicating fully polarised beam and 0.0 fully unpolarised one.

  In case of non-vanishing polarity the following additional fields are required:
- `'pol err unc'` — explain
- `'pol err corLpol'` — explain
- `'pol err corTpol'` — explain

The inclusive DIS cross sections are calculated on an x-Q^2-y grid. Correspondingly, the following columns need to present in the correctly formatted input file: `'x'`, `'Q2'` and `'y'`.

Data format requirements for `fastNLO`

In this subsection we describe data format specific for the `fastNLO` implementation accessed by choosing 'FastNLO jets' and 'FastNLO ep jets normalised' reaction types. Examples of properly formatted files are:

`datafiles/HERA/ZEUS_InclJets_HighQ2_98-00.dat`

`datafiles/HERA/H1_NormInclJets_HighQ2_99-07.dat`.

`TheoryType = 'FastNLO'` indicates usage of the `fastNLO`. The variable `ThoryInfoFile` should contain the proper path to the `fastNLO` table in version 2.0 or higher. `xFitter` supports both flexible and inflexible scales. Older `fastNLO` tables can be still accessed through the `APPLGRID` interface.

The following fields are required to be present in the `CInfo` list:

- `'PublicationUnits'` — The desired units in which the cross sections are calculated by the `fastNLO` code. If the corresponding `DataInfo` field is set to '1.' the cross sections will be given in the same units as used in the relevant publication. In the case it is set to '0.', absolute cross section units will be used.
- `'MurDef', 'MufDef'` — The renormalisation and factorisation scale definitions used with variable scale `fastNLO` tables. If the chosen `fastNLO` table does not support variable scales, these fields will be ignored and the scale embedded within the table will be used instead. The values of the corresponding `DataInfo` fields set the renormalisation scale μ_r and factorisation scale μ_f following the `fastNLO` standard:

$$\begin{aligned}
\text{value}:&\quad \text{definition}\\
0:&\quad \mu^2_{r/f} = \mu^2_1\\
1:&\quad \mu^2_{r/f} = \mu^2_2\\
2:&\quad \mu^2_{r/f} = (\mu^2_1 + \mu^2_2)\\
3:&\quad \mu^2_{r/f} = (\mu^2_1 + \mu^2_2)/2\\
4:&\quad \mu^2_{r/f} = (\mu^2_1 + \mu^2_2)/4\\
5:&\quad \mu^2_{r/f} = ((\mu_1 + \mu_2)/2)^2\\
6:&\quad \mu^2_{r/f} = ((\mu_1 + \mu_2))^2\\
7:&\quad \mu^2_{r/f} = \max(\mu^2_1, \mu^2_2)\\
8:&\quad \mu^2_{r/f} = \min(\mu^2_1, \mu^2_2)\\
9:&\quad \mu^2_{r/f} = (\mu_1 * exp(0.3 * \mu_2))^2
\end{aligned}$$

  where μ_1 and μ_2 are specific scales chosen during production of the table. In particular for jet production at HERA traditionally

$$\mu^2_1 = Q^2 \qquad \mu^2_2 = p^2_T$$

- `sqrt(S)` — Should be defined only for 'FastNLO ep jets normalised' reaction type. The ep collision centre-of-mass energy in GeV. In particular, for HERA based results the the corresponding `DataInfo` value should be 300. for measurements based on data collected prior to 1997 (inclusive) and 318. for data collected after 1997.
- `'lumi(e-)/lumi(tot)'` — Should be defined only for 'FastNLO ep jets normalised' reaction type. The normalisation depends on the ratio of the luminosities of the positron and electron data used for the cross section measurement. This ratio should be given in a format (lumi(e^-) / (lumi(e^-) + lumi(e^+)) and ashould take values between [0., 1.].

- `'UseZMVFNS'` — Should be defined for ’FastNLO ep jets normalised’ reaction type. The calculation of the integrated inclusive DIS cross sections could be time consuming. This option provides an opportunity to use a ”Zero Mass Variable Flavour Number Scheme” approximation which is very fast and provides enough precision for normalisation purposes. ZMVNS is used if the corresponding `DataInfo` field is set to 1. Otherwise, the same scheme is used as defined globally with the variable ’HF_SCHEME’ defined in steering.txt file.

In addition there are some specific values within the `ColumnName` field which allow information specific to each data point to be passed. They are listed below:

- `'Z0Corr'` — (optional) The correction due to the Z_0 boson exchange. If it is given, each point calculated by the `fastNLO` code will be multiplied by the `Z0Corr` value.
- `'NPCorr'` — (optional) The non-perturbative correction. If it is given, each point calculated by the `fastNLO` code will be multiplied by the `NPCorr` value. `Z0Corr` and `NPCorr` can be added simultaneously, and in this case the calculated cross sections will be multiplied by the product `Z0Corr` * `NPCorr`.
- `'q2min', 'q2max', 'ymin', 'ymax', 'xmin', 'xmax'` — Should be defined for ’FastNLO ep jets normalised’ reaction type and are used to define DIS phase space for the normalisation. Since these three (`q2, y, x`) are connected by the relation

  $$Q^2 = x \cdot y \cdot s \tag{70}$$

  only two are required to be present to unambiguously define the DIS phase space for each data point.

5.2.4 Understanding the output

The results of the minimization are printed to the standard output and written to files in the `output/` directory.

The quality of the fit can be judged based on the total χ^2 per degrees of freedom. It is printed for each iteration as

```
                     Iteration   Chi2   NDF        Chi2/NDF
  xfitter f,ndf,f/ndf      3      588.64 579         1.02
```

The resulting χ^2 is reported at the end of minimisation for each data set and for the correlated systematic uncertainties separately. This information is printed and written to the `output/Results.txt` file. The `Results.txt` file contains additional information about shifts of the correlated systematic uncertainties.

The minimization information from the `MINUIT` program is stored using the standard `MINUIT` output format in the `output/minuit.out.txt` file. The level of verbosity for this information can be changed by `MINUIT` commands in the `minuit.in.txt` file. It is a good idea to check that `MINUIT` does not report any errors or warnings at the end of minimisation.

Point by point comparison of the data and predictions after the minimization is provided in the file stored in `output/fittedresults.txt`. The file reports three columns corresponding to the three first bins of the input tables, data value, sum in quadrature of statistical and uncorrelated systematic uncertainty, total uncertainty and then the predicted value, before and after applying correlated systematic shifts, the pull between the data and theory and the data set index. The pull p is calculated as

$$p = \frac{\mu - m}{\sigma_{\text{uncor}}} \tag{71}$$

where μ is the data value, m is the prediction and σ_{uncor} is the total uncorrelated uncertainty. Similar information is stored in the `pulls.first.txt` and `pulls.last.txt` files (dataset index, first bin, second bin, third bin, theory, data, pull). Theory is adjusted for systematic error shifts in this case.

The output PDFs are stored in `output/pdfs_q2val_XX.txt` files. Each of the files reports values of gluon, and quark PDFs as a function of x for fixed Q^2 points. The Q^2 values and x grid are specified by `&Output` namelist in the `steering.txt` file.

The PDF information and data to theory comparisons can be plotted using the `bin/DrawResults` program. Calling it without arguments plots results from the `output/` directory. Giving the programme one argument specifies the sub-directory where the information is read. Calling the `bin/DrawResults` program with two arguments provides a comparison of the PDFs obtained in the two fits.

In addition the `xFitter` package provides PDFs in the `LHAPDF` format as `output/lhapdf.block.txt` file. To obtain the `LHAPDF` grid file, run the `tools/tolhapdf.cmd` script. The script provides a `PDFs.LHgrid` file which can be read by the `LHAPDF` version lhapdf-5.8.6.tar.gz or later.

5.3 User Example

Two examples are available in `xFitter` for benchmarking purposes. The first example describes the default fit with HERA DIS inclusive data, the second is a fit where all the available data in `xFitter` are fitted simultaneously.

5.3.1 DIS inclusive only

By default the `xFitter` steering files (`steering.txt` and `minuit.in.txt`) are set to fit the DIS inclusive cross section data. For this fit no any additional configuration options are required. The results of this example fit (total and partial $\chi 2$, systematic shifts, data to theory comparison and histograms, see section 8.4 ”Understanding the output” for more details) can be compared to the result provided for benchmarking in ”examples”.

5.3.2 All processes

In order to run the second example with all data, the corresponding `steering.txt` from ”input_steering” has to be copied to the main directory:

cp input-steering/steering.txt.ALLdata steering.txt

The user must make sure that all data sets as given in `steering.txt.ALLdata` together with corresponding theory files have been downloaded before running this example.
Since the included sets contain various ep, pp, $p\overline{p}$ and fix target data, it is necessary to have `xFitter` configured with `APPLGRID`, hathor and `LHAPDF` options (corresponding shared library linking as explained in section 5.1 is required before configuration):

./configure –enable-applgrid –enable-lhapdf –enable-hathor

It is recommended to do ”make clean” before each configuration.
As in the previous case, the fit result can be compared to the one provided in ”examples”.

A How to add new data

Inclusion of the data files is controlled by `&InFiles` namelist in the `steering.txt` file. For example, by default the following four HERA-I files are included:

```
&InFiles
    NInputFiles = 4
    InputFileNames(1) = 'datafiles/H1ZEUS_NC_e-p_HERA1.0.dat'
    InputFileNames(2) = 'datafiles/H1ZEUS_NC_e+p_HERA1.0.dat'
    InputFileNames(3) = 'datafiles/H1ZEUS_CC_e-p_HERA1.0.dat'
    InputFileNames(4) = 'datafiles/H1ZEUS_CC_e+p_HERA1.0.dat'
&End
```

To include more files:

- Increase the `NInputFiles` variable.
- Specify the additional file by providing corresponding `InputFileNames()` variable.

Details about data file format can be found in section 5.2.3.

References

[1] F. James and M. Roos, Comput. Phys. Commun. **10**, 343 (1975).

[2] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 438 (1972).

[3] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 675 (1972).

[4] L. N. Lipatov, Sov. J. Nucl. Phys. **20**, 94 (1975).

[5] Y. L. Dokshitzer, Sov. Phys. JETP **46**, 641 (1977).

[6] G. Altarelli and G. Parisi, Nucl. Phys. B **126**, 298 (1977).

[7] M. Botje (2010), http://www.nikef.nl/h24/qcdnum/index.html, [arXiv:1005.1481].

[8] G. Curci, W. Furmanski, and R. Petronzio, Nucl.Phys. **B175**, 27 (1980).

[9] W. Furmanski and R. Petronzio, Phys.Lett. **B97**, 437 (1980).

[10] E. L. *et al.*, Phys. Lett. **B291**, 325 (1992).

[11] E. L. *et al.*, Nucl. Phys. **B392**, 162, 229 (1993).

[12] S. Riemersma, J. Smith, and van Neerven. W.L., Phys. Lett. **B347**, 143 (1995), [hep-ph/9411431].

[13] R. Demina, S. Keller, M. Kramer, S. Kretzer, R. Martin, *et al.* (1999), [hep-ph/0005112].

[14] R. S. Thorne and R. G. Roberts, Phys. Rev. D **57**, 6871 (1998), [hep-ph/9709442].

[15] R. S. Thorne, Phys. Rev. **D73**, 054019 (2006), [hep-ph/0601245].

[16] S. Forte, E. Laenen, P. Nason, and J. Rojo, Nucl. Phys. **B834**, 116 (2010), [arXiv:1001.2312].

[17] V. Bertone, S. Carrazza, and J. Rojo, Comput. Phys. Commun. **185**, 1647 (2014), [1310.1394].

[18] S. Alekhin, *OPENQCDRAD*, a program description and the code are available via: http://www-zeuthen.desy.de/~alekhin/OPENQCDRAD.

[19] A. D. Martin, Eur. Phys. J. C **63**, 189 (2009).

[20] R. S. Thorne (2012), [arXiv:1201.6180].

[21] J. C. Collins, Phys.Rev. **D58**, 094002 (1998), [hep-ph/9806259].

[22] M. Cacciari, M. Greco, and P. Nason, JHEP **05**, 007 (1998), [hep-ph/9803400].

[23] J. C. Collins, F. Wilczek, and A. Zee, Phys. Rev. **D18**, 242 (1978).

[24] R. D. Ball, V. Bertone, M. Bonvini, S. Forte, P. G. Merrild, J. Rojo, and L. Rottoli (2015), [1510.00009].

[25] R. D. Ball, M. Bonvini, and L. Rottoli, JHEP **11**, 122 (2015), [1510.02491].

[26] S. Alekhin and S. Moch, Phys. Lett. **B699**, 345 (2011), [arXiv:1011.5790].

[27] K. H., N. Lo Presti, S. Moch, and A. Vogt, Nucl.Phys. **B864**, 399 (2012).

[28] H. Spiesberger, Private communication.

[29] Jegerlehner, Proceedings, LC10 Workshop **DESY 11-117** (2011).

[30] H. Burkhard, F. Jegerlehner, G. Penso, and C. Verzegnassi, in CERN Yellow Report on ”Polarization at LEP” 1988.

[31] Y. Li and F. Petriello, Phys.Rev. **D86**, 094034 (2012), [arXiv:1208.5967].

[32] G. Bozzi, J. Rojo, and A. Vicini, Phys.Rev. **D83**, 113008 (2011), [arXiv:1104.2056].

[33] A. Falkowski, M. L. Mangano, A. Martin, G. Perez, and J. Winter (2012), [arXiv:1212.4003].

[34] S. D. Drell and T.-M. Yan, Phys. Rev. Lett. **25**, 316 (1970).

[35] M. Yamada and M. Hayashi, Nuovo Cim. **A70**, 273 (1982).

[36] P. Nason, S. Dawson, and R. K. Ellis, Nucl.Phys. **B303**, 607 (1988).

[37] P. Nason, S. Dawson, and R. K. Ellis, Nucl.Phys. **B327**, 49 (1989).

[38] M. L. Mangano, P. Nason, and G. Ridolfi, Nucl.Phys. **B373**, 295 (1992).

[39] A program description and the code are available via: `http://www.ge.infn.it/~ridolfi/hvqlibx.tgz`.

[40] O. Zenaiev, A. Geiser, K. Lipka, J. Blmlein, A. Cooper-Sarkar, *et al.* (2015), [1503.04581].

[41] O. Zenaiev, Doctoral thesis, Hamburg University (2015).

[42] V. Kartvelishvili, A. Likhoded, and V. Petrov, Phys.Lett. **B78**, 615 (1978).

[43] C. Peterson, D. Schlatter, I. Schmitt, and P. M. Zerwas, Phys.Rev. **D27**, 105 (1983).

[44] E. Braaten, K.-m. Cheung, S. Fleming, and T. C. Yuan, Phys.Rev. **D51**, 4819 (1995), [hep-ph/9409316].

[45] M. Aliev, H. Lacker, U. Langenfeld, S. Moch, P. Uwer, *et al.*, Comput.Phys.Commun. **182**, 1034 (2011), [arXiv:1007.1327].

[46] P. Bärnreuther, M. Czakon, and A. Mitov (2012), [arXiv:1204.5201].

[47] S. Moch, P. Uwer, and A. Vogt, Phys.Lett. **B714**, 48 (2012), [hep-ph/1203.6282].

[48] T. Kluge, K. Rabbertz, and M. Wobisch, pp. 483–486 (2006), [hep-ph/0609285].

[49] M. Wobisch, D. Britzger, T. Kluge, K. Rabbertz, and F. Stober [fastNLO Collaboration] (2011), [arXiv:1109.1310].

[50] D. Britzger, K. Rabbertz, F. Stober, and M. Wobisch [fastNLO Collaboration] (2012), [arXiv:1208.3641].

[51] Z. Nagy and Z. Trocsanyi, Phys.Rev. **D59**, 014020 (1999), [hep-ph/9806317].

[52] Z. Nagy and Z. Trocsanyi, Phys.Rev.Lett. **87**, 082001 (2001), [hep-ph/0104315].

[53] Z. Nagy, Phys.Rev. **D68**, 094002 (2003), [hep-ph/0307268].

[54] Z. Nagy, Phys.Rev.Lett. **88**, 122003 (2002), [hep-ph/0110315].

[55] N. Kidonakis and J. Owens, Phys.Rev. **D63**, 054019 (2001), [hep-ph/0007268].

[56] T. Carli *et al.*, Eur. Phys. J. **C66**, 503 (2010), [arXiv:0911.2985].

[57] J. M. Campbell and R. K. Ellis, Phys. Rev. **D60**, 113006 (1999), [arXiv:9905386].

[58] J. M. Campbell and R. K. Ellis, Nucl. Phys. Proc. Suppl. **205-206**, 10 (2010), [arXiv:1007.3492].

[59] N. N. Nikolaev and B. Zakharov, Z.Phys. **C49**, 607 (1991).

[60] K. Golec-Biernat and M. Wüsthoff, Phys. Rev. D **59**, 014017 (1999), [hep-ph/9807513].

[61] E. Iancu, K. Itakura, and S. Munier, Phys. Lett. **B590**, 199 (2004), [hep-ph/0310338].

[62] J. Bartels, K. Golec-Biernat, and H. Kowalski, Phys. Rev. D **66**, 014001 (2002), [hep-ph/0203258].

[63] I. Balitsky, Nucl. Phys. B **463**, 99 (1996), [hep-ph/9509348].

[64] S. Catani, M. Ciafaloni, and F. Hautmann, Nucl. Phys. B **366**, 135 (1991).

[65] M. Ciafaloni, Nucl. Phys. B **296**, 49 (1988).

[66] S. Catani, F. Fiorani, and G. Marchesini, Phys. Lett. B **234**, 339 (1990).

[67] S. Catani, F. Fiorani, and G. Marchesini, Nucl. Phys. B **336**, 18 (1990).

[68] G. Marchesini, Nucl. Phys. B **445**, 49 (1995).

[69] H. Jung and F. Hautmann (2012), [arXiv:1206.1796].

[70] G. Marchesini and B. R. Webber, Nucl. Phys. **B349**, 617 (1991).

[71] G. Marchesini and B. R. Webber, Nucl. Phys. **B386**, 215 (1992).

[72] H. Jung, S. Baranov, M. Deak, A. Grebenyuk, F. Hautmann, *et al.*, Eur.Phys.J. **C70**, 1237 (2010), [arXiv:1008.0152].

[73] M. Deak, F. Hautmann, H. Jung, and K. Kutak, *Forward-Central Jet Correlations at the Large Hadron Collider* (2010), [arXiv:1012.6037].

[74] A. Airapetian *et al.* [HERMES Collaboration], Phys.Lett. **B666**, 446 (2008), [arXiv:0803.2993].

[75] A. Schöening (2011), Private communication.

[76] D. Stump *et al.*, Phys. Rev. **D65**, 014012 (2002), [hep-ph/0101051].

[77] M. Botje, J.Phys. **G28**, 779 (2002), [hep-ph/0110123].

[78] F. Aaron *et al.* [H1 Collaboration], Eur. Phys. J. **C63**, 625 (2009), [arXiv:0904.0929].

[79] F. Aaron *et al.* [H1 Collaboration], JHEP **1209**, 061 (2012), [arXiv:1206.7007].

[80] W. T. Giele and S. Keller, Phys.Rev. **D58**, 094023 (1998), [hep-ph/9803393].

[81] W. T. Giele, S. Keller, and D. Kosower (2001), [hep-ph/0104052].

[82] A. Glazov, S. Moch, and V. Radescu, Phys. Lett. B **695**, 238 (2011), [arXiv:1009.6170].

[83] R. D. Ball, V. Bertone, F. Cerutti, L. Del Debbio, S. Forte, *et al.*, Nucl.Phys. **B855**, 608 (2012), [arXiv:1108.1758].

[84] R. D. Ball *et al.* [NNPDF Collaboration], Nucl.Phys. **B849**, 112 (2011), [arXiv:1012.0836].

[85] G. Watt and R. Thorne, JHEP **1208**, 052 (2012), [arXiv:1205.4024].

[86] H. Paukkunen and P. Zurita (2014), [arXiv:1402.6623].