nnpdf++ data layout

Maintainer: Nathan Hartland
n.p.hartland@vu.nl

May 2, 2020

Contents

1 Revision history

Date	Version	Major/Minor	Author	Comments
9/9/15	1.0.0	-	nh	First version.
16/9/15	1.1.0	Major	vb/sc/nh	Implementing many new theory parameters. Adding header to `CFACTOR` files.
18/10/15	1.1.1	Minor	nh	Updating paths after move to git repository.
19/10/15	1.2.0	Major	nh	New FK Header format.
3/12/15	1.2.1	Minor	nh	Commondata systematics format emphasis.
7/12/15	1.2.2	Minor	nh/ag	Commondata process types modified
10/12/15	1.2.3	Minor	nh	Higgs process types added
25/01/16	1.2.4	Minor	nh	Added asymmetry comment and quark production thresholds.
16/03/16	1.2.5	Minor	nh	Added EWK double-differential process types.
13/09/16	1.2.6	Minor	nh	Added the THEORY systematic type.
7/10/16	1.2.7	Minor	nh	Added the THEORYCORR and THEORYUNCORR systematic types, removed THEORY.
17/10/16	1.2.8	Minor	sc	Added SKIP systematic type.
29/11/16	1.3.0	Major	nh	Added uncertainties upon C-factors
19/02/18	1.4.0	Major	nh	Added global_nx parameter. Removed SYSTYPES file discussion, removed references to old scripts.
26/02/18	1.4.1	Minor	nh	Formatting improvements, discussed FK table compression.
05/06/18	1.4.2	Minor	nh	Addition of EScaleVar entry in theory database.

2 Introduction

In the `nnpdf++` project, data files used by the code may be grouped into two categories, theory and experiment. Experimental data and the information pertaining to the treatment of systematic errors are held in `CommonData` and `SYSTYPE` files. `FK` tables, `COMPOUND` and `CFACTOR` files store the precomputed information for use when calculating theoretical predictions corresponding to information held in the equivalent `CommonData` file. In this document the file formats and naming conventions for these files will be detailed, along with the directory structure employed by the `nnpdf++` code.

For NNPDF3.1 and later fits, a considerably larger number of theory options will be explored than in previous determinations. In NNPDF3.0 the main theory variations used were perturbative order, value of the strong coupling and the number of active flavours in the VFNS. For NNPDF3.1 and later, variations in additional parameters must be accommodated, such as treatments of the heavy quark mass (pole vs MS-bar), scale variations, intrinsic charm, resummation effects etc. The book-keeping used to enable efficient variations of the theoretical treatment used in fits post-3.0 will therefore also be outlined here.

This document will begin by detailing the specifications for the file formats used by the code, first with the experimental data file formats and layouts in Section 3 and secondly with the file formats used for theoretical predictions in Section 4. Finally the organisation of these files within the `nnpdf++` structure will be described in Section 5.

2.1 Important definitions

In order to clarify the later description, here are a few important terminological points to note.

Dataset vs *Experiment*

When referring to a collection of data points two words are used in the `nnpdf++` code which have specific meanings. *Dataset* refers to the result of a specific measurement, typically associated with a single experimental paper and corresponds to the `DataSet` class in the `nnpdf++` code. *Experiment* refers to a collection of *Datasets* which are associated by experimental cross-correlations. For example, the ATLAS 2010 R=0.4 inclusive jet measurement and the ATLAS 2011 high-mass Drell-Yan measurement are both examples of *Datasets* as used in the NNPDF3.0 analysis. Both of these datasets are grouped into the ATLAS *Experiment* as they have systematic uncertainties that are cross-correlated with each other. In this document, when using these terms in this sense, they will be italicised for clarity.

***Dataset* and *Experiment* names**

When referred to, the *Dataset* and *Experiment* names refer to the short identifying string used in the code for each *Dataset* and *Experiment*. For example, the *Dataset* name for the aforementioned ATLAS 2010 inclusive jet measurement with R=0.4 is ATLASR04JETS36PB.

3 Experimental data files

Data made available by experimental collaborations comes in a variety of formats. For use in a fitting code, this data must be converted into a common format that contains all the required information for use in PDF fitting. Existing formats commonly used by the community, such as in HepData, are generally unsuitable. Principally as they often do not fully describe the breakdown of systematic uncertainties. Therefore over several years an NNPDF standard data format has been iteratively developed, now denoted `CommonData`. In addition to the `CommonData` files themselves, in the `nnpdf++` project the user has the ability to vary the treatment of individual systematic errors by use of parameter files denoted `SYSTYPE` files. In this section we shall detail the specifications of these two files.

In principle, the file specification and classes described in this section are independent of the `nnpdf++` project and may be generated by whatever means the user sees fit. In practice, the `CommonData` and `SYSTYPE` files are generated by the `buildmaster` project of `nnpdf++` from the raw experimental data files.

3.1 Process types and kinematics

Before going into the file formats, we shall summarise the identifying features used for data in the `nnpdf++` code.

Each datapoint has an associated *process type* string. This can be specified by the user, but **must** begin with the appropriate identifying base process type. Additionally for each datapoint three kinematic values are given, the *process type* being primarily to identify the nature of these values. Typically the first kinematic variable is the principal differential quantity used in the measurement. The second kinematic variable defines the scale of the process. The third is generally the centre-of-mass energy of the process, or inelasticity in the case of DIS. The allowed basic process types, and their corresponding three kinematic variables are outlined below.

- **DIS** - Deep inelastic scattering measurements: (x, Q^2, y)
- **DYP** - Fixed-target Drell-Yan measurements: $(y, M^2, \sqrt{s})$
- **JET** - Jet production: $(\eta, p_T^2, \sqrt{s})$
- **DIJET** - Dijet production: $(\eta, m_{12}, \sqrt{s})$

- **PHT** - Photon production: $(\eta_\gamma, E^2_{T,\gamma}, \sqrt{s})$
- **INC** - A total inclusive cross-section: $(0, \mu^2, \sqrt{s})$
- **EWK_RAP** - Collider electroweak rapidity distribution: $(\eta/y, M^2, \sqrt{s})$
- **EWK_PT** - Collider electroweak p_T distribution: $(p_T, M^2, \sqrt{s})$
- **EWK_PTRAP** - Collider electroweak p_T, y distribution: $(\eta/y, p_T^2, \sqrt{s})$
- **EWK_MLL** - Collider electroweak lepton-pair mass distribution: $(M_{ll}, M_{ll}^2, \sqrt{s})$
- **EWJ_(J)RAP** - Collider electroweak + jet boson(jet) rapidity distribution: $(\eta/y, M^2, \sqrt{s})$
- **EWK_MLLRAPCOS** - Collider electroweak M_{ll}, y, $\cos\theta^*$ distribution: $(\eta/y, M_{ll}^2, \sqrt{s})$
- **EWJ_(J)PT** - Collider electroweak + jet boson(jet) p_T distribution: $(p_T, M^2, \sqrt{s})$
- **EWJ_(J)PTRAP** - Collider electroweak + jet boson(jet) p_T, y distribution: $(\eta/y, p_T^2, \sqrt{s})$
- **EWJ_MLL** - Collider electroweak+jet lepton-pair mass distribution: $(M_{ll}, M_{ll}^2, \sqrt{s})$
- **HQP_YQQ** - Heavy diquark system rapidity $(y^{QQ}, \mu^2, \sqrt{s})$
- **HQP_MQQ** - Heavy diquark system mass $(M^{QQ}, \mu^2, \sqrt{s})$
- **HQP_PTQQ** - Heavy diquark system p_T $(p_T^{QQ}, \mu^2, \sqrt{s})$
- **HQP_YQ** - Heavy quark rapidity $(y^Q, \mu^2, \sqrt{s})$
- **HQP_PTQ** - Heavy quark p_T $(p_T^Q, \mu^2, \sqrt{s})$
- **HIG_RAP** - Higgs boson rapidity distribution $(y, M_H^2, \sqrt{s})$

As examples of *process type* strings, consider **EWK_RAP** for a collider W boson asymmetry measurement binned in rapidity, and **DIS_F2P** for the F_2^p structure function in DIS. The user is free to choose something identifying for the second segment of the process type, the important feature being the basic process type. However, users are encouraged to only use this freedom when absolutely necessary (such as when used in combination with APFEL).

One special case is that of W boson lepton asymmetry measurements, which being cross-section asymmetries may occasionally have negative datapoints. Therefore asymmetry measurements must have the final tag **ASY** to ensure that artificial data generation permits negative data values. An example *process type* string would be **EWK_RAP_ASY**.

Notes for the future

In the future it would be nice to have a more flexible treatment of the kinematic variables, both in their number and labelling.

3.2 `CommonData` file format

Each experimental *Dataset* has its own `CommonData` file. `CommonData` files contain the bulk of the experimental information used in the `nnpdf++` project, with the only other experimental data files controlling the treatment and correlation of systematic errors. Each `CommonData` file is a plaintext file with the following layout.

The first line begins with the *Dataset* name, the number of systematic errors, and the number of datapoints in the set, whitespace separated. For example for the ATLAS 2010 jet measurement the first line of the file reads:

ATLASR04JETS36PB 91 90

Which demonstrates that the set *name* is 'ATLASR04JETS36PB', that there are 91 sources of systematic uncertainty, 90 datapoints, one associated `FK` table, and that the `FK` table corresponds to a proton initial state. As another example, consider the NMCPD *Dataset*:

NMCPD 5 211

Here there are 5 sources of systematic uncertainty and 211 datapoints.

Following this, each line specifies the details of a single datapoint. The first value being the datapoint index $1 < i_{\rm dat} \leq N_{\rm dat}$, followed by the *process type* string as outlined above, and the three kinematic variables in order. These are followed by the value of the experimental datapoint itself, and the value of the statistical uncertainty associated with it (absolute value). Finally the systematic uncertainties are specified. The layout per datapoint is therefore

$i_{\rm dat}$ *ProcessType* $\rm kin_1$ $\rm kin_2$ $\rm kin_3$ data_value stat_error [.. systematics ..]

For example, in the case of a DIS datapoint from the BCDMSD *Dataset*:

1 DIS_F2D 7.0e-02 8.75e+00 5.666e-01 3.6575e-01 6.43e-03 [.. systematics ..]

In these lines the systematic uncertainties are laid out as so. For each uncertainty, additive and multiplicative versions are given. The additive uncertainty is given by absolute value, and the multiplicative as a percentage of the data value (that is, relative error multiplied by 100). The systematics string is formed by the sequence of $N_{\rm sys}$ pairs of systematic uncertainties:

$$[.. \text{ systematics } ..] = \sigma_0^{\rm add} \quad \sigma_0^{\rm mul} \quad \sigma_1^{\rm add} \quad \sigma_1^{\rm mul} \quad .... \quad \sigma_n^{\rm add} \quad \sigma_n^{\rm mul}$$

where σ_i^{add} and σ_i^{mul} are the additive and multiplicative versions respectively of the systematic uncertainty arising from the ith source. While it may seem at first that the multiplicative error is spurious given the presence of the additive error and data central value, this may not be the case. For example in a closure test scenario, the data central values may have been replaced in the `CommonData` file by theoretical predictions. Therefore if you wish to use a covariance matrix generated with the original multiplicative uncertainties via the t_0 method, you must also store the original multiplicative (percentage) error. For flexibility and ease of I/O this is therefore done in the `CommonData` file itself.

For a *Dataset* with N_{dat} datapoints and N_{sys} sources of systematic uncertainty, the total `CommonData` file should therefore be $N_{\text{dat}} + 1$ lines long. Its first line contains the set parameters, and every subsequent line should consist of the description of a single datapoint. Each datapoint line should therefore contain $7 + 2N_{\text{sys}}$ columns.

3.3 `SYSTYPE` file format

The explicit presentation of the systematic uncertainties in the `CommonData` file allows for a great deal of flexibility in the treatment of these errors. Specifically, whether they should be treated as additive or multiplicative uncertainties, and how they are correlated, both within the *Dataset* and within a larger *Experiment*. A specification for how the systematic uncertainties should be treated is provided by a `SYSTYPE` file. As there is not always an unambiguous method for the treatment of these uncertainties, these information is kept outside the (unambiguous) `CommonData` file. Several options for this treatment are often provided in the form of multiple `SYSTYPE` files which may be selected between in the fit.

Each `SYSTYPE` file begins with a line specifying the total number of systematics. Naturally this must match with the N_{sys} variable specified in the associated `CommonData` file. This is presented as a single integer. For example, in the case of the BCDMSD `SYSTYPE` files, the first line is

```
8
```

As there are $N_{\text{sys}} = 8$ sources of systematic uncertainty for this *Dataset*. Following this line are N_{sys} lines, describing each source of systematic uncertainty. For each source two parameters are provided, the *uncertainty treatment* and the *uncertainty description*. These are laid out for each systematic as:

$$i_{\text{sys}} \quad [\textit{uncertainty treatment}] \quad [\textit{uncertainty description}]$$

Where $1 < i_{\text{sys}} \leq N_{\text{sys}}$ enumerates each systematic. The *uncertainty treatment* determines whether the uncertainty should be treated as additive, multiplicative, or in cases where the choice is unclear, as randomised on a replica by replica basis. These choices are selected by using the strings **ADD**, **MULT**, or **RAND**. The *uncertainty description* specifies how the systematic is to be correlated with other datapoints. There are three special cases for the *uncertainty description*, specified by the strings **CORR**, **UNCORR**, **THEORYCORR**,

THEORYUNCORR and **SKIP**. The first two specify whether the systematic is fully correlated **only** within the *Dataset* (**CORR**), or whether the systematic is totally uncorrelated (**UNCORR**). The **THEORY** descriptor is used to describe theoretical systematics due to e.g missing NNLO corrections, which are treated as either **CORR** or **UNCORR** according to their suffix, but are not included in the generation of artificial replicas (their only contribution is to the fitting error function). If the user wishes to correlate a specific uncertainty between multiple *Datasets* within an *Experiment*, then they should use a custom *uncertainty description*. When building a covariance matrix for an *Experiment*, the `nnpdf++` code checks for matches between the *uncertainty descriptions* of systematics of its constituent *Datasets*. If a match is found, the code will correlate those systematics over the relevant datasets. The **SKIP** descriptor removes the systematic from the covariance matrices for debug purposes.

As an example, let us consider an NNPDF2.3 standard `SYSTYPE` for the BCDMSD *Dataset*.

```
8
1 ADD BCDMSFB
2 ADD BCDMSFS
3 ADD BCDMSFR
4 MULT BCDMSNORM
5 MULT BCDMSRELNORMTARGET
6 MULT CORR
7 MULT CORR
8 MULT CORR
```

Here the first five systematics have custom *uncertainty descriptions*, thereby allowing them to be cross-correlated with other *Datasets* in a larger *Experiment*. Systematics six to eight are specified as being fully correlated, but only within the BCDMSD *Dataset*. Additionally note that the first three systematics are specified as additive, and the remainder are multiplicative. If we compare now to the equivalent `SYSTYPE` file for the BCDMSP *Dataset*:

```
11
1 ADD BCDMSFB
2 ADD BCDMSFS
3 ADD BCDMSFR
4 MULT BCDMSNORM
5 MULT BCDMSRELNORMTARGET
6 MULT CORR
7 MULT CORR
8 MULT CORR
9 MULT CORR
10 MULT CORR
11 MULT CORR
```

It is clear that the first five systematics are the same as in the BCDMSD *Dataset*, and therefore should the two sets be combined into a common *Experiment*, the code will cross-correlate them appropriately. The combination of `SYSTYPE` and `CommonData` is quite flexible. As stated previously, once generated from the original raw experimental data, the `CommonData` file is fixed and should not be altered apart from for the purpose of correcting errors. In practice the full details on the systematic correlation and their treatment is often not precisely specified. This system allows for the safe variation of these parameters for testing purposes.

4 Theory data files

In the `nnpdf++` project, `FK` tables (or grids) are used to provide the information required to compute pQCD cross sections in a compact fashion. With the `FK` method a typical hadronic observable datapoint $\mathcal{O}$, is computed as,

$$\mathcal{O}_d = \sum_{\alpha,\beta}^{N_x} \sum_{i,j}^{N_{\rm pdf}} \sigma^{(d)}_{\alpha\beta ij} N_i^0(x_\alpha) N_j^0(x_\beta). \tag{1}$$

Where $\sigma^{(d)}_{\alpha\beta ij}$, the `FK` table, is a five index object with two indices in flavour (i, j), two indices in x (α, β) and a datapoint index d. $N_i^0(x_\alpha)$ is the $i^{\rm th}$ initial scale PDF in the evolution basis at x-grid point $x = x_\alpha$. Each `FK` table has an internally specified x-grid upon which the PDFs are interpolated. The full 14-PDF evolution basis used in the `FK` tables is given by:

$$\{\gamma, \Sigma, g, V, V3, V8, V15, V24, V35, T3, T8, T15, T24, T35\}\,. \tag{2}$$

Additional information may be introduced via correction factors known internally as C-factors. These consist of datapoint by datapoint multiplicative corrections to the final result of the `FK` convolution $\mathcal{O}$. These are provided by `CFACTOR` files, typical applications being the application of NNLO and electroweak corrections. For processes which depend non-linearly upon PDFs, such as cross-section ratios or asymmetries, multiple FK tables may be required for one observable. In this case information is provided in the form of a `COMPOUND` file which specifies how the results from several `FK` tables may be combined to produce the target observable. In this section we shall specify the layout of the `FK`, `COMPOUND` and `CFACTOR` files.

4.0.1 FK table compression

It is important to note that the FK table format as described here pertains to the *uncompressed* tables. Typically FK tables as found and read by the NNPDF code are compressed individually with gzip.

4.1 `FK` file format

`FK` preamble layout

The FK preamble is constructed by a set of data segments, of which there are two configurations. The first configuration consists of a list of key-value pairs, and the second is a simple data 'blob' with no requirements as to it's formatting. Each segment is begun by a delineating line, for key-value pairs the segment opening line is

```
_SegmentName_____________________________________________
```

and for data blobs the opening line is

```
{SegmentName_____________________________________________
```

The key difference being in the first character, underscore (_) for key-value pair segments, and open curly brace ({) for data blobs. The name of the segment is specified from the second character, to a terminating underscore (_). The line is then typically padded out with underscores up to 60 characters. Following this delineating line, for a key-value segment, the following lines must all be of the format

```
*KEY: VALUE
```

with the first character required to be an asterisk (*), then specifying the key, and value for that segment. For blob-type segments, no constraints are placed upon the format, aside from that each line **must not** begin with one of the delineating characters { or _, as these will trigger the construction of a new segment.

While the user may specify additional segments, both key-value pair and blob-type for their own use, there are seven segments required by the code. These are, specified by their segment name:

- **GridDesc** [BLOB]
  This segment provides a 'banner' with a short description for the FK table. The contents of this banner are displayed when the table is read from file.

- **VersionInfo** [K-V]
  A list specifying the versions of the various pieces of code used in the generation of this FK table (minimally libnnpdf and apfel).

- **GridInfo** [K-V]
  This list specified various architectural points of the FK table. The required keys are specified in Table 1.

- **TheoryInfo** [K-V]
  A list of all the theory parameters used in the generation of the table. The required keys are specified in Table 2.

- **FlavourMap** [BLOB]
  The segment describes the flavour structure of the grid by means of a flavour map. This map details which flavour channels are active in the grid, using the basis specified in Eqn. 2. For DIS processes, an example section would be

```
{FlavourMap_____________________________________________
0 1 1 0 0 0 0 0 0 0 1 0 0 0
```

  Which specifies that only the Singlet, gluon and T_8 channels are populated in the grid. In the case of hadronic FK tables, the full 14×14 flavour combination matrix is specified in the same manner. Consider the flavourmap for the CDFR2KT *Dataset*:

```
{FlavourMap_____________________________________________
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

  This flavourmap contains 9 nonzero entries, demonstrating the importance of only computing those flavour combinations that are relevant to the process. Additionally this map instructs the `nnpdf++` convolution code as to which elements of the FastKernel grid should be read, to minimise holding zero entries in memory.

- **xGrid** [BLOB]
  This segment defines the x grid upon which the `FK` grid is defined, given as an N_x long list of the x-grid points. This grid should be optimised to minimise `FK` grid zeros in x-space. The blob is a simple list of the grid points, here is an example of an x-grid with $N_x = 5$ entries:

```
{xGrid__________________________________________________
```

```
0.10000000000000001
0.13750000000000001
0.17499999999999999
0.21250000000000002
1.00000000000000000
```

For examples of complete DIS and hadronic FK table headers, see Appendix C.

FK grid layout

To start the section of the file with the FK grid itself, we begin with a blob-type segment delineator:

```
{FastKernel_______________________________________________
```

The grid itself is now written out. For hadronic data, the format is line by line as follows:

$$d\ \alpha\ \beta\ \sigma^d_{\alpha\beta 11}\ \sigma^d_{\alpha\beta 12}\ ....\ \sigma^d_{\alpha\beta nn} \tag{3}$$

Where d is the index of the data point for that line, α is the x-index of the first PDF, β is the x-index of the second PDF, the $\sigma^d_{\alpha\beta ij}$ are the values of the FastKernel grid for data point d as in Eqn 1, and $n = 14$ is the total number of parton flavours in the grid. Therefore the full 14×14 flavour space for one combination of the indices $\{d, \alpha, \beta\}$ is written out on each line.

These lines should be written out first in β, then α and finally d so that the FK grids are written in blocks of datapoints. All FK grid values should be written out in double precision. For DIS data the FK grids must be written out as

$$d\ \alpha\ \sigma^d_{\alpha 1}\ \sigma^d_{\alpha 2}\ ....\ \sigma^d_{\alpha n} \tag{4}$$

Therefore here all $n = 14$ values are written out for each combination of $\{d, \alpha\}$.

When writing out the grids, note that only x-grid points for which there are nonzero FK entries are written out. For example, there should be no lines such as:

d α β 0 0 0 ... 0

However, for those x-grid points which do have nonzero σ contributions, the full set of flavour contributions must be written out regardless of the number of zero entries. This choice was made in order that the nonzero flavour entries may be examined/optimised by hand after the FK table is generated.

The FK file should end on the last entry in the grid, and without empty lines at the end of file.

4.2 `CFACTOR` file format

Additional multiplicative factors to be applied to the output of the `FK` convolution may be introduced by the use of `CFACTOR` files. These files have a very simple format. They begin with a header providing a description of the C-factor information stored in the file. This segment is initialised and terminated by a line beginning with a star (*) character and consists of six mandatory fields

- **SetName** - The *Dataset* name.
- **Author** - The author of the `CFACTOR` file.
- **Date** - The date of authorship.
- **CodesUsed** - The code or codes used in generating the $C-$factors.
- **TheoryInput** - Theory input parameters used in the $C-$factors (e.g α_S, scales).
- **PDFset** - The PDF set used in the $C-$factors.

These fields are formatted as

FieldName: FieldEntry

and may be accompanied by any additional information, within the star delineated header region. Consider the following as a complete example of the header,

```
****************************************
SetName: D0ZRAP
Author: John Doe john.doe@cern.ch
Date: 2014
CodesUsed: MCFM 15.01
TheoryInput: as 0.118, central scale 91.2 GeV
PDFset: NNPDF30_as_0118_nnlo
Warnings: None
Additional Information here
****************************************
```

The remainder of the file consists of the C-factors themselves, and the error upon the C-factors. Each line is now the C-factor for each datapoint, with the whitespace separated uncertainty. For example, for *Dataset* with five points, the data section of a `CFACTOR` file may be:

```
1.1 0.1
1.2 0.12
1.3 0.13
1.4 0.14
1.5 0.15
```

Where the i^{th} line corresponds to the C-factor to be applied to the `FK` prediction for the $(i-1)^{\text{th}}$ datapoint. The first column denotes the value of the C-factor and the second column denotes the uncertainty upon it (in absolute terms, not as a percentage or otherwise relative to the C-factor). For a complete example of a `CFACTOR` file, please see Appendix B.

4.3 `COMPOUND` file format

Some *Datasets* cover observables that depend non-linearly upon the input PDFs. For example, the NMCPD *Dataset* is a measurement of the ratio of deuteron to proton structure functions. In the `nnpdf++` code such sets are denoted *Compound Datasets*. In these cases, a prescription for how the results from FK convolutions as in Eqn 1 should be combined must be given.

The `COMPOUND` files are a simple method of providing this information. For each *Compound Dataset* a `COMPOUND` file is provided that contains the information on how to build the observable from constituent `FK` tables. The following operations are currently implemented.

Operation ($N_{\texttt{FK}}$)	Code	Output Observable
Null Operation(1)	**NULL**	$\mathcal{O}_d = \mathcal{O}_d^{(1)}$
Sum (2)	**ADD**	$\mathcal{O}_d = \mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)}$
Normalised Sum (4)	**SMN**	$\mathcal{O}_d = (\mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)})/(\mathcal{O}_d^{(3)} + \mathcal{O}_d^{(4)})$
Asymmetry (2)	**ASY**	$\mathcal{O}_d = (\mathcal{O}_d^{(1)} - \mathcal{O}_d^{(2)})/(\mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)})$
Ratio (2)	**RATIO**	$\mathcal{O}_d = \mathcal{O}_d^{(1)}/\mathcal{O}_d^{(2)}$

Here $N_{\texttt{FK}}$ refers to the number of tables required for each compound operation. $\mathcal{O}_d$ is final observable prediction for the d^{th} point in the *Dataset*. $\mathcal{O}_d^{(i)}$ refers to the observable prediction for the d^{th} point arising from the i^{th} `FK` table calculation. Note that here the ordering in i is important.

The `COMPOUND` file layout is as so. The first line is once again a general comment line and is not used by the code, and therefore has no particular requirements other than its presence. Following this line should come a list of the FK tables required for the calculation. This must be given as the table's filename *without* its path, preceded by the string '**FK:** '. For example,

```
FK: FK_SETNAME_1.dat
FK: FK_SETNAME_2.dat
```

The ordering of the list is once again important, and must match the above table. For example the observables $\mathcal{O}^{(i)}$ arise from the computation with the i^{th} element of this list.

The final line specified the operation to be performed upon the list of tables, and must take the form

OP: [**CODE**]

where the [**CODE**] is given in the above table. Here is an example of a complete `COMPOUND` file

```
# COMPOUND FK
FK: FK_NUMERATOR.dat
FK: FK_DENOMINATOR.dat
OP: RATIO
```

5 Organisation of data files

The `nnpdf++` code needs to be able to handle a great deal of different options with regard to the treatment of both experimental data and theoretical choices. In the code, every effort has been made to keep experimental and theoretical parameters strictly separate.

In this section we shall specify the layout of the various `nnpdf++` data directory. It is in this directory that all of the read-only data to be used in the fit are accessed. The data directory is located in the `nnpdfcpp` git repository, under the path `/nnpdfcpp/data/`.

5.1 Experimental data storage

The central repository for `CommonData` in use by `nnpdf++` projects is located in the `nnpdfcpp` git repository at

```
/nnpdfcpp/data/commondata/
```

Where a separate `CommonData` file is stored for each *Dataset* with the filename format

```
DATA_<SETNAME>.dat
```

Information on the treatment of systematic uncertainties, provided in `SYSTYPE` files, are located in the subdirectory

```
/nnpdfcpp/data/commondata/systypes
```

Here several `SYSTYPE` files may be supplied for each *Dataset*. The various options are enumerated by suffix to the filename. The filename format for `SYSTYPE` files is therefore

```
SYSTYPE_<SETNAME>_<SYSID>.dat
```

Where the default systematic ID is **D**EFAULT. As an example, consider the first `SYSTYPE` file for the D0ZRAP *Dataset*:

```
SYSTYPE_D0ZRAP_DEFAULT.dat
```

5.2 Theory lookup table

In order to organise the various different theoretical treatments available, a lookup table is provided in `sqlite3` format. This lookup table can be found in the `nnpdfcpp` repository data directory at:

```
/nnpdfcpp/data/theory.db
```

This file should only be edited in order to add new theory options. It may be edited with any appropriate sqlite3-supported software. A script is provided to give a brief overview of the various theory options available. It can be found at

```
/nnpdfcpp/data/disp_theory.py
```

and should be run without any arguments.

Theory options are enumerated by an integer *TheoryID*. The parameters of each theory option are described in the lookup table under the appropriate ID. The current available parameters are summarised in Table 2.

5.3 Theory storage

Each theory configuration is stored as a GZIP compressed Tar archive with filename format

```
theory_<THEORYID>.tgz
```

and are stored at the location specified in the default nnprofile.yaml. For easy access, they can be downloaded through the `vp-get` utility. Each archive contains the following directory structure

```
theory_X/
     -cfactor/
     -compound/
     -fastkernel/
```

Inside the directory `theory_X/cfactor/` are stored `CFACTOR` files with the filename format

```
CF_<TYP>_<SETNAME>.dat
```

where `<TYP>` is a three-letter designation for the source of the C-factor (i.e EWK or QCD) and `<SETNAME>` is the typical *Dataset* designation.

The directory `theory_X/compound/` contains the `COMPOUND` files described earlier, this time with the filename format

```
FK_<SETNAME>-COMPOUND.dat
```

Finally the `FK` tables themselves are stored in `theory_X/fastkernel/` with the filename format

```
FK_<SETNAME>.dat
```

Naturally, all of the FastKernel and C-factor files within the directory `theory_X/` have been determined with the theoretical parameters specified in the theory lookup table under ID `X`.

A Theory/FK configuration variables

Key	Type	Description	Comments
SETNAME	String	*SetName*	
HADRONIC	Boolean	Hadronic flag	0 or 1
NDATA	Integer	$N_{\rm dat}$	Number of data points
NX	Integer	N_x	Number of $x-$points in grid

Table 1: Table specifying the required elements of the GridInfo `FK` header segment. The Key column specifies the exact format of the Key in the K-V pair used in the GridInfo segment.

Field/Key	Type	Description	Comments
ID	Integer	*TheoryID*	Theory enumerating ID
PTO	Integer	pQCD order	(0/1/2 = LO/NLO/NNLO)
FNS	Text	Flavour Number Scheme	e.g FONLL-A/B/C or ZM-VFNS or FFNS
DAMP	Integer	FONLL damping factor switch	Boolean
IC	Integer	Intrinsic charm switch	Boolean
ModEv	Text	DGLAP solution mode	EXA/EXP/TRN
XIR	Real	ξ_R	μ_R/Q
XIF	Real	ξ_F	μ_F/Q
EScaleVar	Real	Switch for DGLAP scale variation	Boolean
NfFF	Integer	Number of flavours in the FFNS	3/4/5/6
MaxNfAs	Integer	$n_{f,\max}^{(\alpha_S)}$	Max active flavours in α_s
MaxNfPdf	Integer	$n_{f,\max}^{(\mathrm{PDF})}$	Max active flavours in PDFs
Q0	Real	Q_0	FK Table initial scale
alphas	Real	Strong coupling	Format: $\alpha_S(Q_{\mathbf{ref}})$
Qref	Real	$Q_{\mathbf{ref}}$	Reference scale for α_S in GeV
QED	Integer	QED switch	Boolean
alphaqed	Real	QED coupling	Format: $\alpha_{\mathrm{QED}}(Q_{\mathbf{qedref}})$
Qedref	Real	$Q_{\mathbf{qedref}}$	QED reference scale (GeV)
SxRes	Integer	small-x resummation switch	Boolean
SxOrd	Text	small-x pt order	("LL", "NLL", "NNLL")
HQ	Text	HQ mass treatment	POLE/MSBAR
mc	Real	c quark mass $M_c/m_c(Q_{\mathrm{mc}})$	Units: GeV
Qmc	Real	Q_{mc}	c reference scale (GeV)
kcThr	Real	c production threshold ratio	Ratio to mc
mb	Real	b quark mass $M_b/m_b(Q_{\mathrm{mb}})$	Units: GeV
Qmb	Real	Q_{mb}	b reference scale (GeV)
kbThr	Real	b production threshold ratio	Ratio to mb
mt	Real	t quark mass $M_t/m_t(Q_{\mathrm{mt}})$	Units: GeV
Qmt	Real	Q_{mt}	t reference scale (GeV)
ktThr	Real	t production threshold ratio	Ratio to mt
CKM	Text	CKM matrix elements	unpacked 3X3 matrix
MZ	Real	M_Z	Z mass (GeV)
MW	Real	M_W	W mass (GeV)
GF	Real	G_F	Fermi coupling constant
SIN2TW	Real	$\sin^2\theta_W$	
TMC	Integer	Target mass corrections	Boolean
MP	Real	M_P Proton mass	Units: GeV
global_nx	Integer	Global x-grid precision	Default (0) uses set-by-set precision
Comments	Text	General comments	

Table 2: Table describing the variables used to specify theory choices in the `nnpdf++` project. Here they are shown directly as the fields of the `theory.db` lookup table. The Type column refers directly to the `sqlite3` data type of the field.

B CFACTOR file format example

```
****************************************
SetName: D0ZRAP
Author: John Doe john.doe@cern.ch
Date: 2014
CodesUsed: MCFM 15.01
TheoryInput: as 0.118, central scale 91.2 GeV
PDFset: NNPDF30_as_0118_nnlo
Warnings: None
7 datapoints in total. No smoothing applied
****************************************
1.1 0.1
1.2 0.2
1.3 0.3
1.4 0.4
1.5 0.5
1.6 0.6
1.7 0.7
```

C FK preamble examples

C.1 DIS preamble - BCDMSD

```
{GridDesc_____________________________________________________
------------------------------
FK_BCDMSD.dat
------------------------------
_VersionInfo__________________________________________________
*APFEL: 2.6.1
*libnnpdf: 1.1.0b
_GridInfo_____________________________________________________
*HADRONIC: 0
*NDATA: 254
*NX: 50
*SETNAME: BCDMSD
{FlavourMap___________________________________________________
0 1 1 0 0 0 0 0 0 0 1 1 0 0
_TheoryInfo___________________________________________________
*DAMP: 1
*FNS: FONLL-C
*GF: 1.16638e-05
*HQ: MSBAR
*IC: 0
*MP: 0.938
*MW: 80.398
*MZ: 91.1876
*MaxNfAs: 5
*MaxNfPdf: 5
*ModEv: TRN
*NfFF: 5
*PTO: 2
*Q0: 1
*QED: 0
*Qedref: 1.777
*Qmb: 4.18
*Qmc: 3
*Qmt: 162.7
*Qref: 91.2
*SIN2TW: 0.23126
*SxOrd: LL
```

```
*SxRes: 0
*TMC: 1
*TheoryID: 7
*XIF: 1
*XIR: 1
*alphaqed: 0.00749625
*alphas: 0.118
*mb: 4.18
*mc: 0.986
*mt: 162.7
{xGrid_____________________________________________________
6.9265888619991195e-02
7.7677574001058236e-02
8.6760599033455912e-02
9.6515727077269992e-02
1.0693847246838524e-01
1.1801962180968653e-01
1.2974586013120251e-01
1.4210045166737728e-01
1.5506393063634324e-01
1.6861476611854062e-01
1.8272997502743873e-01
1.9738566676226815e-01
2.1255751145471796e-01
2.2822113029454361e-01
2.4435241115381084e-01
2.6092775579054239e-01
2.7792426659347097e-01
2.9531988146590743e-01
3.1309346535041777e-01
3.3122486633420206e-01
3.4969494345265562e-01
3.6848557237516494e-01
3.8757963421332448e-01
4.0696099179998674e-01
4.2661445698241623e-01
4.4652575176931059e-01
4.6668146557197077e-01
4.8706901027900074e-01
5.0767657449372061e-01
5.2849307792672917e-01
```

```
5.4950812667484750e-01
5.7071196990123374e-01
5.9209545827198862e-01
6.1365000437161166e-01
6.3536754522794392e-01
6.5724050700057512e-01
6.7926177183385794e-01
7.0142464683629069e-01
7.2372283512038826e-01
7.4615040881848282e-01
7.6870178397770295e-01
7.9137169723166323e-01
8.1415518414141896e-01
8.3704755910070550e-01
8.6004439670038091e-01
8.8314151445118372e-01
9.0633495676848319e-01
9.2962098012648797e-01
9.5299603929602150e-01
9.7645677458414570e-01
{FastKernel_________________________________________________
```

C.2 Hadronic preamble - CDFR2KT

```
{GridDesc_____________________________________________________
-------------------------------------------------------------
FK_CDFR2KT.dat
-------------------------------------------------------------
_VersionInfo__________________________________________________
*APFEL: 2.6.1
*libnnpdf: 1.1.0b
{Readme_______________________________________________________
*********************************************************************
ExpName: CDFR2KT
Author: FastNLO authors
Date: 2010
CodesUsed: NLOjet++/FastNLO (scenario fnt2004 from FastNLO webpage)
AdditionalInfo: incl. jets, kT algo D=0.7
*********************************************************************
_GridInfo_____________________________________________________
*HADRONIC: 1
*NDATA: 76
*NX: 30
*SETNAME: CDFR2KT
{FlavourMap___________________________________________________
0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 1 0 0 0 1 1 0 0
0 1 1 1 0 0 1 0 0 0 1 1 0 0
0 1 1 1 0 1 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 1 0 0 0 1 0 0 0
0 1 1 1 0 1 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 1 0 0 0 0 1 1 0 0
0 1 1 1 0 0 1 0 0 0 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
_TheoryInfo___________________________________________________
*DAMP: 1
*FNS: FONLL-C
*GF: 1.16638e-05
```

```
*HQ: MSBAR
*IC: 0
*MP: 0.938
*MW: 80.398
*MZ: 91.1876
*MaxNfAs: 5
*MaxNfPdf: 5
*ModEv: TRN
*NfFF: 5
*PTO: 2
*Q0: 1
*QED: 0
*Qedref: 1.777
*Qmb: 4.18
*Qmc: 3
*Qmt: 162.7
*Qref: 91.2
*SIN2TW: 0.23126
*SxOrd: LL
*SxRes: 0
*TMC: 1
*TheoryID: 7
*XIF: 1
*XIR: 1
*alphaqed: 0.00749625
*alphas: 0.118
*mb: 4.18
*mc: 0.986
*mt: 162.7
{xGrid_____________________________________________________
4.0941945000024672e-03
5.9356426849003037e-03
8.5647477735742213e-03
1.2278230204351056e-02
1.7448602544560710e-02
2.4515641282009264e-02
3.3957625320032526e-02
4.6241012256902900e-02
6.1757804939792604e-02
8.0769759935090835e-02
1.0337895878919207e-01
```

```
1.2953267418094364e-01
1.5905525671030885e-01
1.9169158055350388e-01
2.2714813737177489e-01
2.6512436628283159e-01
3.0533281023729242e-01
3.4750997595899380e-01
3.9142071068612511e-01
4.3685860760309952e-01
4.8364426537988547e-01
5.3162257521672562e-01
5.8065972288573631e-01
6.3064027352226959e-01
6.8146451295139832e-01
7.3304610913825119e-01
7.8531009886079706e-01
8.3819117643580765e-01
8.9163224991215573e-01
9.4558322764065939e-01
{FastKernel_________________________________________________
```