PROCEEDINGS OF SCIENCE

Theory pipeline for PDF fitting

Andrea Barontini,[a] **Alessandro Candido,**[a,*] **Juan M. Cruz-Martinez,**[a,b] **Felix Hekhorn,**[a] **Giacomo Magni**[c] **and Christopher Schwan**[d]

[a] *TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano Via Celoria 16, 20133, Milano, Italy*

[b] *CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland*

[c] *Department of Physics and Astronomy, Vrije Universiteit, Amsterdam, The Netherlands Nikhef Theory Group, Amsterdam, The Netherlands*

[d] *Universität Würzburg, Institut für Theoretische Physik und Astrophysik, 97074 Würzburg, Germany*

E-mail: andrea.barontini@mi.infn.it, alessandro.candido@mi.infn.it, juan.cruz.martinez@cern.ch, felix.hekhorn@mi.infn.it, christopher.schwan@physik.uni-wuerzburg.de

Fitting PDFs requires the integration of a broad range of datasets, both from data and theory side, into a unique framework. While for data the integration mainly consists in the standardization of the data format, for the theory predictions there are multiple ingredients involved. Different providers are developed by separate groups for different processes, with a variety of inputs (runcards) and outputs (interpolation grids). Moreover, since processes are measured at different scales, DGLAP evolution has to be provided for the PDF candidate, or precomputed into the grids. We are working towards the automation of all these steps in a unique framework, that will be useful for any PDF fitting groups, and possibly also for phenomenological studies.

41st International Conference on High Energy physics - ICHEP2022
6-13 July, 2022
Bologna, Italy

*Speaker

PoS(ICHEP2022)784

1. Introduction

Performing a PDF fit requires to integrate several elements to be gathered from many different sources: data from several experiments, ranging over multiple decades and formats, and competitive theory predictions, coming from different providers. Finally, a fitting methodology has to be selected and engineered to implement theory constraints, and to limit not physically motivated bias.

While data is a *static* component in the fit, the theory predictions depend on the candidate PDF, since they are the mapping that connect the unobserved PDF space, to the observed data space. During the fit, this map will be used a large number of times (at least once for every minimization step), so it is paramount to have an efficient way to evaluate it, otherwise it can become a serious bottleneck.

For this reason, a few interfaces to PDF independent theory predictions have already been implemented [1–4], and they are used in different contexts. They propose different file formats to store the output of a Monte Carlo generator, splitting them by luminosity component, perturbative order, and observables binning. This output can be optimized as an *interpolation grid*, leveraging the fact that the PDF itself is only defined over a finite set of points, and thus including the interpolation basis in the factorized cross-section. Essentially, this recast the partonic cross sections predictions as a *theory array*, for which the Mellin convolution is replaced by a linear algebra contraction over a single or multiple PDF set. This idea can be broadened to apply to any factorized function, describing the structure of an external hadron (both incoming and outgoing).

However, this picture does not exhaust the needs of a PDF fit (or any other hadronic one), because, while the PDF dependence on flavor and x value is folded on the grid, that on factorization scale has to be fixed to the process dependent value. This dependence is not fitted, since it is only determined by perturbative QCD. In order to obtain it, it is required to solve the DGLAP equation with the border condition provided by the fit. But being DGLAP a set of integro-differential equations *linear* in the PDF, this can be converted in the application of a suitable *evolution operator*, solving the same equation. Since the evolution operator can also be computed ahead of time, it is possible to combine the two ingredients (the operator and the grid) in a single fast array interface, that will directly produce the required theory predictions once contracted on the PDF candidate. Thus, the map from PDF space to data space discussed above, is reduced to a linear algebra product (or more than one, when multiple hadrons are involved). During this operation, there is no loss of generality, since the interpolation basis used for the conversion of the analytic convolutions is already present in many PDF applications due to their non-perturbative nature. Such an interface is called a "Fast Kernel table" (shortened to *FK table*) in the context of the NNPDF collaboration.

To produce the *FK tables* an evolution operator provider is required, and needs to be interfaced with the grids. This was originally done in NNPDF by an internal tool (`FKgenerator`), and then systematized in the `APFELgrid` [5] package (leveraging the `APFEL` [6] evolution library), later reworked once more taking the name of `APFELcomb` [7].

An array interface is extremely useful, since it allows to treat the theory map in the context of many software frameworks, just relying on the data structure of an array. Especially relevant for machine learning software frameworks, but not limited to them, e.g. it allows to create a Bayesian inference-based methodology, without the need of the treatment of further complex functions.

2. Architecture

As it has been explained in the previous section, a theory map, i.e. an FK table, is made of two main components: a PDF independent interpolation *grid* and an *evolution operator*.

For the second one, we just need a single provider, able to compute the DGLAP solving operator for a variety of theory settings (corresponding to different PDF fits, e.g. NLO and NNLO QCD evolution), able to perform the operator computation as efficient as possible, and to smoothly interface with the grid for convolution. Multiple evolution codes are already available for the purpose [6, 8–11], but when the authors decided to begin the full rework of the architecture, it has been clear that a dedicated tool, with this exact goal in mind, would have been an ideal solution. For this reason, the software package `EKO` [12] has been created, providing a framework to solve DGLAP equations in Mellin space, similarly to [8], but generating an output in x-space, i.e. the one usually employed by the fit. Moreover, while not being the first tool to provide evolution operators [6, 9], it produces them as the default output, optimizing the process as much as possible, storing them in a dedicated file format, designed to be easily available from different programming languages.

`EKO` is very different from `APFEL`, the tool on which the NNPDF framework has until now relied. For instance `APFELgrid` (then `APFELcomb`), the tool which generates `APFEL`-based FK table, introduces an explicit dependency on `APFEL` itself (and thus its internals). `EKO` instead not only exposes a restricted public API (making all the dependent projects decoupled from its very internals), but the dependency is not required at all to consume the `EKO` output, consisting of float arrays stored in a very common `tar` archive, and standard `YAML` metadata. On the other side, the observable grids have to be produced by different generators, in order to cover the full variety of available processes. For this reason, we need an interface to them, with the following targets: standardizing the output and making it reproducible.

The solution we propose is thus based on the concept of interpolation grid, and specifically on `PineAPPL` as an interface. In particular, `PineAPPL` exposes APIs to different languages: it is natively written in Rust, but has an API to C/C++, that can be consumed also by a Fortran application (examples provided for all of them), and a Python API, mostly dedicated to scripting and integration with the rest of the pipeline, but there are providers (essentially `yadism` [13], used for DIS at NNLO) already using it to fill grids.

Since different generators require different inputs, we are trying to standardize them into a common format for which other cards can be generated, called *pinecard*. This is still work in progress, nevertheless, it is useful to speak of pinecards, since they are used as inputs for `pinefarm`, that is the unique Python package working as a front-end for the various generators. Essentially, each generator needs dedicated code to run, but this interface has to be written once, and then is part of `pinefarm`, standardizing the input for that generator, and part of the input across all of them (e.g. metadata, like references and observable details, or theory parameters). In fig. 1 we summarize our architecture: the generators are directly interfaced with the `PineAPPL` library, and the output is thus standardized to an interpolation grid (for one or two colliding hadrons), the input instead consists of a *pinecard*.

Once the grid is available, `pineko` (a package dedicated to the final construction of FK tables) can extract the details of the operator needed for the FK table generation from the grid, generate the

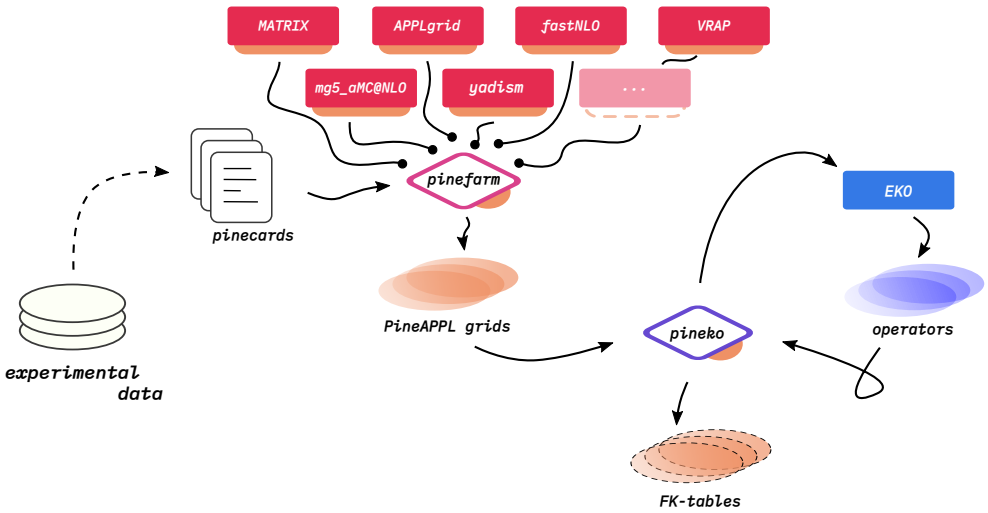

Figure 1: Updated version of the flow diagram already appeared in [14], showing the overall pipeline architecture. Arrows in the picture indicate the flow of information (together with the execution order), and the orange insets on other elements indicate an interface to `PineAPPL` (notice `EKO` not having it). In particular, magenta blocks above `pinefarm` are the providers [1, 2, 13, 15–17].

`EKO` input, and then combine the grid and the operator into the final FK table.

All the components of the pipeline are open source and the code is available in the NNPDF GitHub organization:

- `PineAPPL`: https://github.com/NNPDF/pineappl
- `EKO`: https://github.com/NNPDF/eko
- `pineko`: https://github.com/NNPDF/pineko
- `pinefarm`: https://github.com/NNPDF/runcards (including the relevant pinecards, required for NNPDF fits)

The set of tools does not depend on the NNPDF fitting methodology and can be used in general for any hadronic function fitting[1].

3. Applications

Components have applications on their own, and part of them have already been used (or are being used) to support other works. Even though here it appears incidental, this is an important design feature: we are building a framework, not just a pipeline application. The various components should be focused on dedicated tasks and easy to integrate in different architecture (or, more realistically, stand-alone projects), for similar but different goals.

A first example is the study on evidence for an intrinsic charm component in the proton [18], based on the NNPDF 4.0 PDF set, latest release of the NNPDF family, and `EKO` [12], the evolution code described in the previous section 2. The role of `EKO` has been to unfold the intrinsic component

[1]Generalization of `PineAPPL` to support fragmentation functions and polarized PDFs is work in progress.

from the so-called fitted charm, in the 4 flavor number scheme (default scheme at fitting scale for NNPDF), by backward evolving with DGLAP equation in a 3 flavor number scheme PDF set at a lower scale. On top of the required backward evolution, and the proper treatment of intrinsic components, `EKO` implemented the N^3LO matching conditions between the 4 and 3 flavor schemes, that have been relevant to estimate the perturbative stability of the result obtained.

Another application is the study of the forward backward asymmetry in the Drell–Yan process with a high cut in the invariant mass of the lepton pair [19]. In particular, the work focuses on the comparison between results obtained with the NNPDF 4.0 PDF set and other contemporary PDF sets from different collaborations. We find that a certain shape in the high cut setting is related to the specific shape of the PDFs in the large-x extrapolation region, and so very sensitive to the possible bias of extending behaviors typical of the central data region. In this context, it has been crucial to have `PineAPPL` [4] [20] grids pre-computed to reproduce the results, iterating on the PDF set to investigate different features of the PDF, and trying to trace back the distribution behavior to PDF features.

Finally, a study of the low energy neutrino structure functions is ongoing, where the low Q^2 experimental data is reconciled to the known perturbative calculation at higher energies, based on the PDFs. Here, we use `yadism`, a general inclusive DIS provider interfaced with `PineAPPL`, to produce perturbative QCD calculation for the structure functions that get matched to experimental data.

4. Conclusions

In this proceeding we presented the new theory predictions framework and described its main application in the context of PDF fitting. The main features are the standardization, maintainability, modularity, and reproducibility. We aim for a good support for multiple users and external contributions. Finally, we discussed some early stand-alone applications for the individual components introduced in the framework.

Acknowledgments

AB, AC, JCM, and FH are supported by the European Research Council under the European Union's Horizon 2020 research and innovation Programme (grant agreement number 740006). C.S. is supported by the German Research Foundation (DFG) under reference number DE 623/6-2.

References

[1] T. Carli, D. Clements, A. Cooper-Sarkar, C. Gwenlan, G. P. Salam, F. Siegert et al., *A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: The APPLGRID Project*, *Eur. Phys. J. C* **66** (2010) 503 [0911.2985].

[2] FASTNLO collaboration, *New features in version 2 of the fastNLO project*, in *20th International Workshop on Deep-Inelastic Scattering and Related Subjects*, pp. 217–221, 2012, DOI [1208.3641].

[3] D. Britzger et al., *NNLO interpolation grids for jet production at the LHC*, 2207.13735.

[4] S. Carrazza, E. R. Nocera, C. Schwan and M. Zaro, *PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes*, *JHEP* **12** (2020) 108 [2008.12789].

[5] V. Bertone, S. Carrazza and N. P. Hartland, *APFELgrid: a high performance tool for parton density determinations*, *Comput. Phys. Commun.* **212** (2017) 205 [1605.02070].

[6] V. Bertone, S. Carrazza and J. Rojo, *APFEL: A PDF Evolution Library with QED corrections*, *Comput. Phys. Commun.* **185** (2014) 1647 [1310.1394].

[7] V. Bertone, S. Carrazza and N. Hartland, "https://github.com/NNPDF/apfelcomb.".

[8] A. Vogt, *Efficient evolution of unpolarized and polarized parton distributions with QCD-PEGASUS*, *Comput. Phys. Commun.* **170** (2005) 65 [hep-ph/0408244].

[9] G. Salam and J. Rojo, *The HOPPET NNLO parton evolution package*, in *16th International Workshop on Deep Inelastic Scattering and Related Subjects*, p. 42, 7, 2008, DOI [0807.0198].

[10] M. Botje, *QCDNUM: Fast QCD Evolution and Convolution*, *Comput. Phys. Commun.* **182** (2011) 490 [1005.1481].

[11] V. Bertone, *APFEL++: A new PDF evolution library in C++*, *PoS* **DIS2017** (2018) 201 [1708.00911].

[12] A. Candido, F. Hekhorn and G. Magni, *EKO: Evolution Kernel Operators*, 2202.02338.

[13] A. Candido, F. Hekhorn and G. Magni, `N3PDF/yadism: FONLL-B v0.11.0` (Feb., 2022), doi:10.5281/zenodo.6285149.

[14] S. Amoroso et al., *Snowmass 2021 whitepaper: Proton structure at the precision frontier*, 2203.13923.

[15] M. Grazzini, S. Kallweit and M. Wiesemann, *Fully differential NNLO computations with MATRIX*, *Eur. Phys. J. C* **78** (2018) 537 [1711.06631].

[16] R. Frederix, S. Frixione, V. Hirschi, D. Pagani, H. S. Shao and M. Zaro, *The automation of next-to-leading order electroweak calculations*, *JHEP* **07** (2018) 185 [1804.10017].

[17] C. Anastasiou, L. J. Dixon, K. Melnikov and F. Petriello, *High precision QCD at hadron colliders: Electroweak gauge boson rapidity distributions at NNLO*, *Phys. Rev. D* **69** (2004) 094008 [hep-ph/0312266].

[18] NNPDF collaboration, *Evidence for intrinsic charm quarks in the proton*, *Nature* **608** (2022) 483 [2208.08372].

[19] R. D. Ball, A. Candido, S. Forte, F. Hekhorn, E. R. Nocera, J. Rojo et al., *Parton Distributions and New Physics Searches: the Drell-Yan Forward-Backward Asymmetry as a Case Study*, 2209.08115.

[20] C. Schwan, A. Candido, F. Hekhorn and S. Carrazza, `NNPDF/pineappl: v0.5.7 v0.5.7` (Oct., 2022), doi:10.5281/zenodo.7145377.