DESY 21-026
IFJPAN-IV-2021-4
version 5. March 2021

TMDlib2 and TMDplotter: library and plotting tools for transverse-momentum-dependent parton distributions

N.A. Abdulov[1], A. Bacchetta[2] S. Baranov[3], A. Bermudez Martinez[4], V. Bertone[5], C. Bissolotti[2,6], L.I. Estevez Banos[4], M. Bury[7], P.L.S. Connor[4,1], L. Favart[8], F. Guzman[9], F. Hautmann[10,11], H. Jung[4], L. Keersmaekers[1]0, A. Kotikov[12], A. Kusina[13], K. Kutak[13], A. Lelek[10], J. Lidrych[4], A. Lipatov[1], G. Lykasov[12], M. Malyshev[1], M. Mendizabal[4], S. Sadeghi Barzani[12,8], S. Sapeta[11], S. Taheri Monfared[4], A. Signori[2,6], A. van Hameren[13], A.M. van Kampen[1]0, M. Vanden Bemden[8] A. Vladimirov[7], Q. Wang[4,15] H. Yang[4,15]

[1]SINP, Moscow State University, Russia
[2] Dipartimento di Fisica, Universita di Pavia and INFN, Italy
[3]Lebedev Physics Institute, Russia
[4]DESY, Hamburg, Germany
[5] RFU, CEA, Université Paris-Saclay, Gif-sur-Yvette, France
[6] EIC Center, Jefferson Lab, Newport News, USA [7] Institut für Theoretische Physik, Universität Regensburg, Germany
[8] Inter-University Institute For High Energies, Universite Libre de Bruxelles, Belgium
[9]InSTEC, Universidad de La Habana, Cuba
[10]Elementary Particle Physics, University of Antwerp, Belgium
[11]RAL and University of Oxford, UK
[12]JINR, Dubna, Russia
[13] The H. Niewodniczański Institute of Nuclear Physics, Cracow, Poland
[14]Department of Physics, Shahid Beheshti University, Iran
[15]School of Physics, Peking University, China

Abstract

A common library, TMDlib2, for Transverse-Momentum-Dependent distributions (TMDs) and unintegrated parton distributions (uPDFs) is described, which allows for easy access

[1]Now at University of Hamburg

of commonly used TMDs and uPDFs. A tool, TMDplotter, allows for web-based plotting of distributions implemented in TMDlib2, together with collinear pdfs as available in LHAPDF.

PROGRAM SUMMARY

Computer for which the program is designed and others on which it is operable: any with standard C++, tested on Linux and Mac OS systems

Programming Language used: C++

High-speed storage required: No

Separate documentation available: No

Keywords: QCD, TMD factorization, high-energy factorization, TMD PDFs, TMD FFs, unintegrated PDFs, small-x physics.

Other programs used: LHAPDF (version 6) for access to collinear parton distributions, ROOT (any version > 5.30) for plotting the results

Download of the program: `http://tmdlib.hepforge.org`

Unusual features of the program: None

Contacts: H. Jung (hannes.jung@desy.de), A. Bermudez Martinez (armando.bermudez.martinez@desy.de)

Citation policy: please cite the current version of the manual and the paper(s) related to the parameterisation(s).

1 Introduction

The calculation of processes at high energy hadron colliders is based in general on the calculation of a partonic process (matrix element) convoluted with the likelihood to find a parton of specific flavor and momentum fraction at a given scale within the hadrons. If the parton density depends only on the longitudinal momentum fraction x of the hadron's momentum carried by a parton, and the resolution scale μ, the processes are described by collinear factorization with the appropriate evolution of the parton densities (PDFs) given by the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) evolution equations [1–3]. Such descriptions are successful for sufficiently inclusive processes, like inclusive deep-inelastic lepton-hadron scattering (DIS).

In several less inclusive processes, also the transverse momentum of the involved partons plays an important role, leading to an extension of the collinear factorization theorem to include transverse degrees of freedom. Different factorization theorems or ansätze for the inclusion of transverse momenta to the parton densities have been developed in the past, leading to so-called Transverse Momentum Dependent (TMD) parton densities and unintegrated parton densities (uPDFs). For semi-inclusive processes, like semi-inclusive DIS (SIDIS), Drell-Yan (DY) production and e^+e^- scattering, TMD factorization has been formulated [4–16]. The high-energy (small-x limit) factorization was formulated for heavy flavor and heavy boson production in Refs. [17–20] using unintegrated gluon distributions [21–29]. In Refs. [30, 31] the Parton Branching (PB) method was formulated as a way to obtain TMD distributions for all flavours over a wide range of x, transverse momentum k_t, and scale μ essentially by solving the next-to-leading order (NLO) DGLAP equations keeping track of the transverse momenta during each parton branching.

Since the number of available TMD densities increases very rapidly, and different groups provide different sets, it was necessary to develop a common platform to access the different TMD sets in a common form. In 2014 the first version of TMDlib (version 1) and TMDplotter was released [32], which made several TMD sets available to the community. This library has set a common standard for accessing TMD sets, similar to what was available for collinear parton densities in PDFlib [33,34] and LHAPDF [35]. TMDlib is a `C++` library which provides a framework and an interface to a collection of different uPDF and TMD parameterizations.

In this report, we describe a new version of the TMDlib library, TMDlib2, collecting different TMD sets and parameterizations in a single library, as well as the associated online plotting tool TMDplotter. TMDlib2 covers all the features present already in the previous version, and contains significant new developments, like the efficient treatment of TMD uncertainties and an easier method to include new TMD sets.

2 The TMDlib framework

The TMD parton densities are defined as momentum weighted distributions $x\mathcal{A}(x, \bar{x}, k_t, \mu)$, where $x, \bar{x}$ are the (positive and negative) light-cone longitudinal momentum fractions, k_t is the transverse momentum of the parton, and μ is the factorization scale [36–39]. In some of the applications $\bar{x}$ is set explicitly to zero, in other cases $\bar{x} = 0$ means that it is implicitly integrated over. The integral over k_t

$$x\mathcal{A}_{int}(x, \mu) = \int_{k_{t,min}}^{k_{t,max}} dk_t^2 \; x\mathcal{A}(x, k_t, \mu) \, , \tag{1}$$

can be defined. In case of the PB TMDs, this integral returns the collinear PDF (with $k_{t,min} \to 0$, $k_{t,max} \to \infty$), which can be used for calculations of cross sections in collinear factorization. An example, obtained with TMDplotter, is shown in Fig. 1 for the PB-NLO-HERAI+II-2018-set1 [40] which is identical to HERAPDF2.0 [41]. However, in general, Eq.(1) does not converge to the collinear pdf. In some cases by definition the integral is divergent for $k_{t,max} \to \infty$ corresponding to UV singularities.

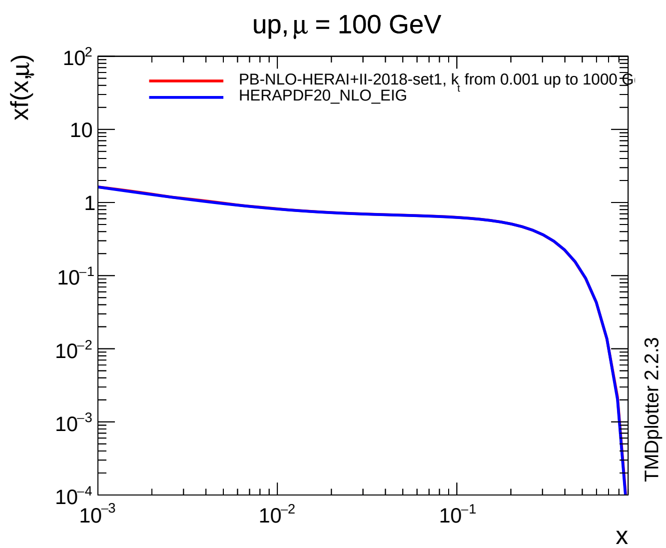

Figure 1: Comparison of up-type parton distributions, $xf(x, \mu) = x\mathcal{A}_{int}(x, \mu)$, integrated over k_t as a function of x at $\mu = 100$ GeV for the integrated distribution PB-NLO-HERAI+II-2018-set1 [40] with HERAPDF2.0 [41].

2.1 Grids and Interpolation

Since the analytic calculation of TMDs as a function of the longitudinal momentum fraction x (we neglect $\bar{x}$ in the following), the transverse momentum k_t and the scale μ is very time consuming and in some cases even not available, the TMDs are saved as grids, and TMDlib provides appropriate tools for interpolation between the grid points (where the type of evolution is indicated):

allFlavPDF	Multidimensional Linear Interpolation in x, k_t and μ is used for PB and CCFM-type TMDs.
Pavia	Interpolation based on Lagrange polynomials of degree three, performed though APFEL++ [42, 43].
InterpolationKS	Multidimensional cubic spline interpolation in x, k_t and μ, based on GSL implementation, is used for KS-type TMDs.

The parameterizations of TMDs in TMDlib are explicitly authorized for each distribution by the corresponding authors. A list of presently available TMDsets is given in Tab. 1. No explicit QCD evolution code is included: the parameterizations are as given in the corresponding references.

The grids of each selected TMD set are read into memory once (the I/O time depends on the size of the grid). Each TMD set is initialized as a separate instance of the TMD class, which is created for each different TMD set, for example for uncertainty sets, or if several different TMD sets are needed for the calculation.

It is the philosophy of TMDlib that the definition of TMD grids is left free, but a few examples are given: the grids for the PB, CCFM and KS TMD sets are stored in form of text tables, the grids of the Pavia type TMDs are stored and read via the `YAML` frame.

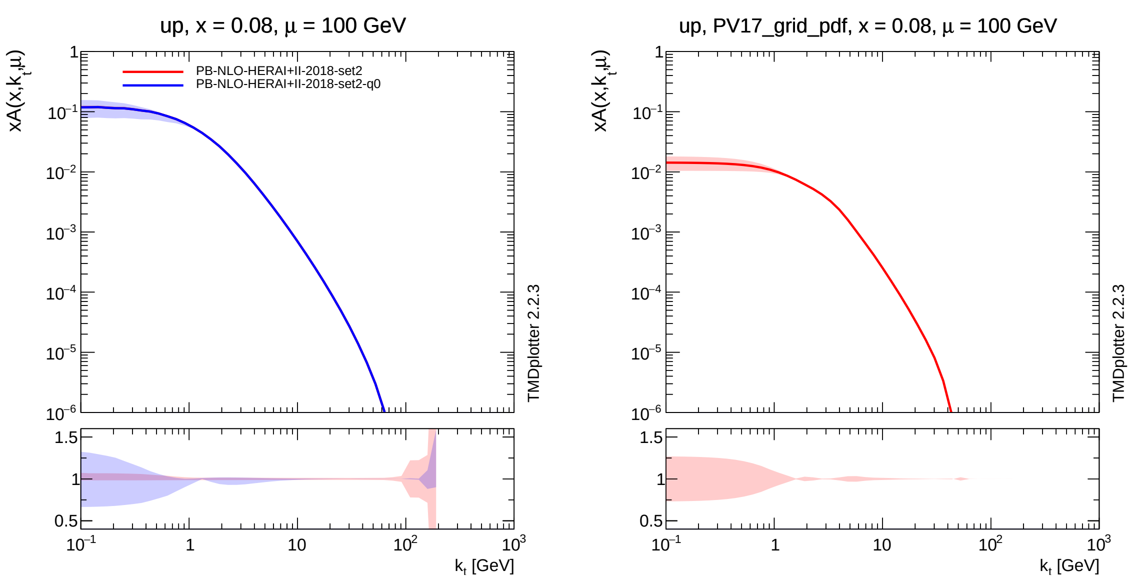

Figure 2: Transverse momentum distribution $x\mathcal{A}(x, k_t, \mu)$ at $x = 0.08$ and $\mu = 100$ GeV obtained with PB-NLO-HERAI+II-2018-set2 [40] (the `q0` set contains a variation of the intrinsic k_t-distribution) (left) and PV17 [46] (right).

2.2 Uncertainty TMD sets

The estimation of theoretical uncertainties is an important ingredient for phenomenological applications, and uncertainties from PDFs and TMDs play a central role. The uncertainties of TMDs are estimated usually from the uncertainties of the input parameters or parameterization. There are two different methods commonly used: the Hessian method [47] which is applied if the parameter variations are orthogonal or the Monte Carlo method providing Monte Carlo replicas [48,49]. The specific prescriptions on how to calculate the uncertainties for a given TMD set should be found in the original publication describing the TMDs.

An example of TMDs with uncertainty band is shown in Fig. 2 for the PB sets as well as for the PV-set.

2.3 TMDplotter

TMDlib provides also a web-based application for plotting TMD distributions – TMDplotter. In Fig. 3 (left) a comparison of the transverse momentum distributions of different TMD sets is shown, and in Fig. 3 (right) the gluon-gluon luminosity calculation for the integrated TMD sets PB-NLO-HERAI+II-2018-set1 [40] at $\mu = 100$ GeV compared with the one obtained from `HERAPDF2.0` is shown (the curves obtained from PB-NLO-HERAI+II-2018-set1 and `HERAPDF2.0` overlap).

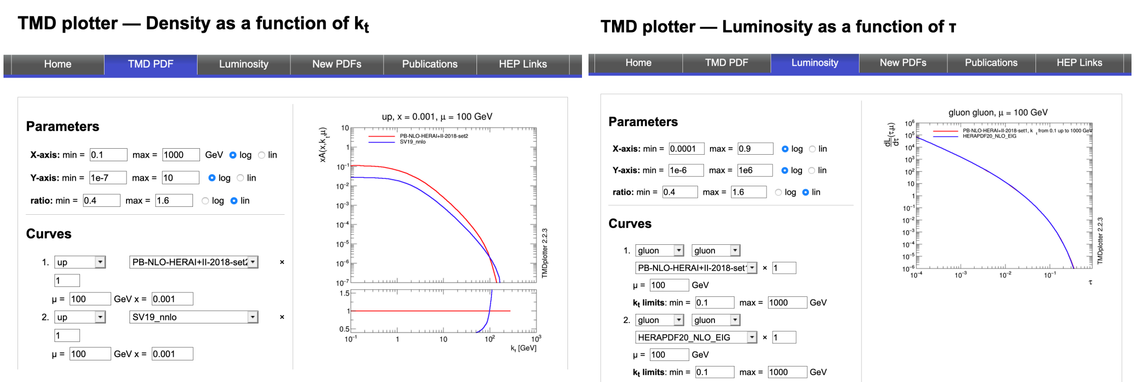

Figure 3: TMDplotter examples: (left) comparison of the transverse momentum distributions of different TMD sets, (right) gluon-gluon luminosity calculation using integrated TMD sets (the curves from PB-NLO-HERAI+II-2018-set1 and `HERAPDF2.0` overlap).

TMDplotter is available at `http://tmdplotter.desy.de/`.

3 New features

- TMDlib2 makes use of C++ classes, and the different sets corresponding to uncertainty sets or sets corresponding to different parameterizations are read once and initialized as different instances, allowing to load many sets into memory.

- information about TMDsets is read via YAML from the TMD info files, containing all metadata
- including new TMDsets is simplified with the new structure of the input sets
- TMDsets are no longer part of the TMDlib distribution, but can be downloaded via `TMDlib-getdata`, distributed with TMDlib2.

4 TMDlib documentation

TMDlib is written in C++ , with an interface for access from FORTRAN code. The source code of TMDlib is available from http://tmdlib.hepforge.org/ and can be installed using the *standard* `autotools` sequence `configure`, `make`, `make install`, with options to specify the installation path and the location of the LHAPDF PDF library [35], and the ROOT data analysis framework library [50] (which is used optionally for plotting). If ROOT is not found via `root-config`, the plotting option is disabled. After installation, `TMDlib-config` gives access to necessary environment variables.

4.1 Description of the program components

Initialisation in C++

`TMDinit(name)`	To initialise the dataset specified by its name `name`. A complete list of datasets available in the current version of TMDlib with the corresponding name is provided in Tab. 1.
`TMDinit(name,irep)`	To initialise a given `irep` replica of the dataset `name`.
`TMDinit(iset)`	To initialise the dataset specified by its identifier `iset`.

Initialisation in Fortran

`TMDinit(iset)`	To initialise the dataset specified by its identifier `iset`.
`TMDset(iset)`	To switch to the dataset `iset`.

Access to TMDs in C++

`TMDpdf(x, xbar, kt, mu)`	Vector double-type function returning an array of 13 variables for QCD parton densities with the values of $x\mathcal{A}(x, \bar{x}, k_t, \mu)$: at index $0, \ldots, 5$ is $\bar{t}, \ldots, \bar{d}$, at index 6 is the gluon, and at index $7, \ldots, 12$ is $d, \ldots, t$ densities.
`TMDpdf(x, xbar, kt, mu, xpq)`	Void-type function filling an array of 13 variables, `xpq`, with the values of $x\mathcal{A}(x, \bar{x}, k_t, \mu)$: at index $0, \ldots, 5$ is $\bar{t}, \ldots, \bar{d}$, at index 6 is the gluon, and at index $7, \ldots, 12$ is $d, \ldots, t$ densities.
`TMDpdf(x, xbar, kt, mu, uval, dval, sea, charm, bottom, gluon, photon)`	Void-type function to return $x\mathcal{A}(x, \bar{x}, k_t, \mu)$ at $x, \bar{x}, k_t, \mu$ for valence u-quarks `uval`, valence d-quarks `dval`, light sea-quarks `s`, charm-quarks `c`, bottom-quarks `b`, gluons `glu` and gauge boson `photon`.
`TMDpdf(x, xbar, kt, mu, up, ubar, down, dbar, strange, sbar, charm, cbar, bottom, bbar, gluon, photon)`	To return $x\mathcal{A}(x, \bar{x}, k_t, \mu)$ at $x, \bar{x}, k_t, \mu$ for the partons `up`, `ubar`, `down`, `dbar`, `strange`, `sbar`, `charm`, `cbar`, `bottom`, `bbar`, `gluon` and gauge boson `photon` (if available).
`TMDpdf(x, xbar, kt, mu, up, ubar, down, dbar, strange, sbar, charm, cbar, bottom, bbar, gluon, photon, Z0, W+,W-,higgs)`	To return $x\mathcal{A}(x, \bar{x}, k_t, \mu)$ at $x, \bar{x}, k_t, \mu$ for the partons `up`, `ubar`, `down`, `dbar`, `strange`, `sbar`, `charm`, `cbar`, `bottom`, `bbar`, `gluon`, the gauge bosons `photon`, `Z0`, `W+`, `W-` and `higgs` (if available).

Access to TMDs in Fortran

`TMDpdf(kf,x, xbar, kt, mu, up, ubar, down, dbar, strange, sbar, charm, cbar, bottom, bbar, gluon)`	To return $x\mathcal{A}(x, \bar{x}, k_t, \mu)$ at $x, \bar{x}, k_t, \mu$ for the partons `up`, `ubar`, `down`, `dbar`, `strange`, `sbar`, `charm`, `cbar`, `bottom`, `bbar`, `gluon` for the hadron flavor `kf`. (`kf` is no longer used, only kept for backward compatibility with TMDlib1)
`TMDpdfEW(x, xbar, kt, mu, up, ubar, down, dbar, strange, sbar, charm, cbar, bottom, bbar, gluon, photon, Z0, W+,W-, higgs)`	To return $x\mathcal{A}(x, \bar{x}, k_t, \mu)$ at $x, \bar{x}, k_t, \mu$ for the partons `up`, `ubar`, `down`, `dbar`, `strange`, `sbar`, `charm`, `cbar`, `bottom`, `bbar`, `gluon`, the gauge bosons `photon`, `Z0`, `W+`, `W-` and `higgs` (if available).

Callable program components

The program components listed in this section are accessible with the same name in C++ as well as in Fortran.

`TMDinfo(dataset)`	Accesses information from the `info` file.
`TMDgetDesc()`	Returns data set description from `info` file.
`TMDgetIndex()`	Returns index number as a string of data set from `info` file.
`TMDgetNumMembers()`	Returns number of members of data sets from `info` file.
`TMDgetScheme()`	Returns evolution scheme of dataset from `info` file.
`TMDgetNf()`	Returns the number of flavours, N_f, used for the computation of Λ_{QCD}.
`TMDgetOrderAlphaS()`	Returns the perturbative order of α_s used in the evolution of the dataset.
`TMDgetOrderPDF()`	Returns the perturbative order of the evolution of the dataset.
`TMDgetXmin()`	Returns the minimum value of the momentum fraction x for which the dataset initialised by `TMDinit(name)` was determined.
`TMDgetXmax()`	Returns the maximum value of the momentum fraction x for which the dataset initialised by `TMDinit(name)` was determined.
`TMDgetQmin()` (`TMDgetQ2min()`)	Returns the minimum value of the energy scale μ (in GeV), (μ^2 (in GeV^2)) for dataset.
`TMDgetQmax()` (`TMDgetQ2max()`)	Returns the maximum value of the energy scale μ (in GeV) , (μ^2 (in GeV^2)) for dataset.
`TMDgetExtrapolation_Q2()`	Returns the method of extrapolation in scale outside the grid definition as specified in `info` file.
`TMDgetExtrapolation_kt()`	Returns the method of extrapolation in k_t outside the grid definition as specified in `info` file.
`TMDgetExtrapolation_x()`	Returns the method of extrapolation in x outside the grid definition as specified in `info` file.
`TMDnumberPDF(name)`	Returns the identifier as a value of the associated `name` of the dataset.
`TMDstringPDF(index)`	Returns the name associated with `index` of the dataset.

4.2 TMDlib calling sequence

In the following simple examples are given to demonstrate how information from the TMD parton densities can be obtained in C++ and Fortran.

- in C++

```
string name ="PB-NLO-HERAI+II-2018-set2";
double x=0.01, xbar=0, kt=10., mu=100.;
```

```
        TMD TMDtest;
        int irep=0;
        TMDtest.TMDinit(name,irep);
        cout << "TMDSet Description: " << TMDtest.TMDgetDesc() << endl;
        cout << "number      = " << TMDtest.TMDnumberPDF(name) << endl;
        TMDtest.TMDpdf(x,xbar,kt,mu, up, ubar, down, dbar, strange, sbar,
                 charm, cbar, bottom, bbar, gluon, photon);
```

- in Fortran (using multiple replicas of the TMD)

```
      x = 0.01
      xbar = 0
      kt = 10.
      mu = 100.
      iset = 102200
      call TMDinit(iset)
      write(6,*) ’ iset = ’, iset
      call TMDinit(iset)
      nmem=TMDgetNumMembers()
      write(6,*) ’ Nr of members ’, nmem,’ in Iset = ’, iset
      do i=0,nmem
        isetTMDlib = iset+i
        write(6,*) ’ isetTMDlib = ’, isetTMDlib
        call TMDinit(isetTMDlib)
        call TMDset(isetTMDlib)
        call TMDpdf(kf,x,xbar,kt,mu,up,ubar,dn,dbar,strange,sbar,
     &    charm,cbar,bottom,bbar,glu)
        call TMDpdfew(kf,x,xbar,kt,mu,up,ubar,dn,dbar,strange,sbar,
     &    charm,cbar,bottom,bbar,glu,photon,z0,wplus,wminus,higgs)
      end do
```

4.3 Installation of TMD grids

The TMD grid files are no longer automatically distributed with the code package, but have to be installed separately. A list of available TMD parameterizations is given in Tab. 1.

```
# get help
bin/TMDlib-getdata --help

# install all data sets
bin/TMDlib-getdata all

# install a single data (for example: SV19_nnlo)
bin/TMDlib-getdata SV19_nnlo
```

4.4 Structure of TMD grids

In TMDlib2 the TMDgrids are stored in directories with the name of a given TMD set which is located in `installation_prefix/share/tmdlib/TMDsetName`. Every such directory contains info file and grid file(s), for example for a TMD set called `test`:

```
~/local/share/tmdlib> ls test
test.info     test_0000.dat
```

The `info` file contains general information on the TMDset (inspired by LHAPDF), as described below, and the file(s) `test_0000.dat` contains the TMDgrid. If further replicas are available (for example for uncertainties), the files are numbered as `test_0000.dat, test_0001.dat,...,` with the number of files given by `NumMembers` as described below.
The `info` file must contain all the information to initialize and use the TMDgrid:

```
SetDesc: "Description of the dataset "
SetIndex: XXXXX
Authors: XXXX
Reference: XXXX
Particle: 2212
NumMembers: 34
NumFlavors: 6
TMDScheme: PB TMD
Flavors: [-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 21]
AlphaS_MZ: 0.118
AlphaS_OrderQCD: 1
OrderQCD: 1
XMin: 9.9e-07
XMax: 1.
KtMin: 0.01
KtMax: 13300.
QMin: 1.3784
QMax: 13300
MZ: 91.1876
MUp: 0.
MDown: 0.
MStrange: 0.
MCharm: 1.47
MBottom: 4.5
MTop: 173
```

The meaning of most entries is obvious from their name, with `TMDScheme` different structures for the TMDgrids can be selected:

`PB TMD`	used for the PB TMD series
`PB TMD-EW`	used for the PB TMD series including electroweak particles
`Pavia TMDs`	used for the PaviaTMD (or similar TMD) series

5 Summary

The authors of this manual set up a collaboration to develop and maintain TMDlib and TMDplotter, respectively a `C++` library for handling different parameterizations of uPDFs/ TMDs and a corresponding online plotting tool. The aim is to update these tools with more uPDF/TMD parton sets and new features, as they become available and are developed.

Acknowledgments

C. Bissolotti is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 647981, 3DSPIN) and by the U.S. Department of Energy contract DE-AC05-06OR23177 under which Jefferson Science Associates operates the Thomas Jefferson National Accelerator Facility. V. Bertone is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 824093. A. van Hameren acknowledges support from the Polish National Science Centre grant no. 2019/35/ST2/03531. K. Kutak acknowledges the support by Polish National Science Centre grant no. DEC-2017/27/B/ST2/01985 A. Signori acknowledges support from the European Commission through the Marie Skłodowska-Curie Action SQuHadron (grant agreement ID: 795475).

References

[1] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 438 (1972). [Yad. Fiz.15,781(1972)].

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

[3] Y. L. Dokshitzer, Sov. Phys. JETP **46**, 641 (1977). [Zh. Eksp. Teor. Fiz.73,1216(1977)].

[4] J. C. Collins and D. E. Soper, Nucl. Phys. B **193**, 381 (1981). [Erratum: Nucl. Phys.B213,545(1983)].

[5] J. C. Collins and D. E. Soper, Nucl. Phys. B **194**, 445 (1982).

[6] J. C. Collins, D. E. Soper, and G. F. Sterman, Nucl. Phys. B **223**, 381 (1983).

[7] J. C. Collins, D. E. Soper, and G. F. Sterman, Phys. Lett. B **109**, 388 (1982).

[8] J. C. Collins, D. E. Soper, and G. F. Sterman, Nucl. Phys. B **250**, 199 (1985).

[9] J. Collins, *Foundations of perturbative QCD*, Vol. 32. Cambridge monographs on particle physics, nuclear physics and cosmology., 2011.

[10] R. Meng, F. I. Olness, and D. E. Soper, Phys. Rev. D **54**, 1919 (1996). `hep-ph/9511311`.

[11] P. M. Nadolsky, D. R. Stump, and C. P. Yuan, Phys. Rev. D **61**, 014003 (2000). [Erratum: Phys.Rev.D 64, 059903 (2001)], `hep-ph/9906280`.

[12] P. M. Nadolsky, D. R. Stump, and C. P. Yuan, Phys. Rev. D **64**, 114011 (2001). `hep-ph/0012261`.

[13] X.-D. Ji, J.-P. Ma, and F. Yuan, Phys. Rev. D **71**, 034005 (2005). `hep-ph/0404183`.

[14] X.-D. Ji, J.-P. Ma, and F. Yuan, Phys. Lett. B **597**, 299 (2004). `hep-ph/0405085`.

[15] M. G. Echevarria, A. Idilbi, and I. Scimemi, JHEP **07**, 002 (2012). `1111.4996`.

[16] J.-Y. Chiu, A. Jain, D. Neill, and I. Z. Rothstein, Phys. Rev. Lett. **108**, 151601 (2012). `1104.0881`.

[17] S. Catani, M. Ciafaloni, and F. Hautmann, Phys. Lett. B **242**, 97 (1990).

[18] E. M. Levin, M. G. Ryskin, Y. M. Shabelski, and A. G. Shuvaev, Sov. J. Nucl. Phys. **53**, 657 (1991).

[19] J. C. Collins and R. K. Ellis, Nucl. Phys. B **360**, 3 (1991).

[20] F. Hautmann, Phys. Lett. B **535**, 159 (2002). `hep-ph/0203140`.

[21] E. Avsar (2012). `1203.1916`.

[22] E. Avsar, Int. J. Mod. Phys. Conf. Ser. **04**, 74 (2011). `1108.1181`.

[23] S. Jadach and M. Skrzypek, Acta Phys. Polon. B **40**, 2071 (2009). `0905.1399`.

[24] F. Dominguez, *Unintegrated Gluon Distributions at Small-x*. Ph.D. Thesis, Columbia U., 2011.

[25] F. Dominguez, J.-W. Qiu, B.-W. Xiao, and F. Yuan, Phys. Rev. D **85**, 045003 (2012). `1109.6293`.

[26] F. Dominguez, A. Mueller, S. Munier, and B.-W. Xiao, Phys. Lett. B **705**, 106 (2011). `1108.1752`.

[27] F. Hautmann, Acta Phys.Polon. B **40**, 2139 (2009).

[28] F. Hautmann, M. Hentschinski, and H. Jung (2012). `1205.6358`.

[29] F. Hautmann and H. Jung, Nucl. Phys. Proc. Suppl. **184**, 64 (2008). `0712.0568`.

[30] F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, JHEP **01**, 070 (2018). `1708.03279`.

[31] F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, Phys. Lett. B **772**, 446 (2017). 1704.01757.

[32] F. Hautmann, H. Jung, M. Krämer, P. Mulders, E. Nocera, *et al.*, Eur. Phys. J. C **74**, 3220 (2014). 1408.3015.

[33] H. Plothow-Besch, Comput. Phys. Commun. **75**, 396 (1993).

[34] H. Plothow-Besch, Int. J. Mod. Phys. A **10**, 2901 (1995).

[35] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, Eur. Phys. J. C **75**, 132 (2015). 1412.7420.

[36] G. Watt, A. D. Martin, and M. G. Ryskin, Phys. Rev. **D70**, 014012 (2004). hep-ph/0309096.

[37] G. Watt, A. D. Martin, and M. G. Ryskin, Eur. Phys. J. **C31**, 73 (2003). hep-ph/0306169.

[38] J. Collins and H. Jung (2005). hep-ph/0508280.

[39] J. C. Collins, T. C. Rogers, and A. M. Stasto, Phys. Rev. D **77**, 085009 (2008). 0708.2833.

[40] A. Bermudez Martinez, P. Connor, F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, Phys. Rev. D **99**, 074008 (2019). 1804.11152.

[41] ZEUS, H1 Collaboration, H. Abramowicz *et al.*, Eur. Phys. J. C **75**, 580 (2015). 1506.06042.

[42] V. Bertone, PoS **DIS2017**, 201 (2018). 1708.00911.

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

[44] A. Bacchetta, F. Delcarro, C. Pisano, M. Radici, and A. Signori, JHEP **06**, 081 (2017). 1703.10157.

[45] A. Bacchetta, V. Bertone, C. Bissolotti, G. Bozzi, F. Delcarro, F. Piacenza, and M. Radici (2019). 1912.07550.

[46] I. Scimemi and A. Vladimirov (2019). 1912.06532.

[47] J. Pumplin, D. Stump, J. Huston, H. Lai, P. M. Nadolsky, *et al.*, JHEP **0207**, 012 (2002). hep-ph/0201195.

[48] W. T. Giele and S. Keller, Phys. Rev. D **58**, 094023 (1998). hep-ph/9803393.

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

[50] R. Brun and F. Rademakers, Nucl. Instrum. Meth. A **389**, 81 (1997).

[51] H. Jung (2004). hep-ph/0411287.

[52] M. Hansson and H. Jung (2003). hep-ph/0309009.

[53] F. Hautmann and H. Jung, Nuclear Physics B **883**, 1 (2014). 1312.7875.

[54] N. A. Abdulov, H. Jung, A. V. Lipatov, G. I. Lykasov, and M. A. Malyshev, Phys. Rev. D **98**, 054010 (2018). 1806.06739.

[55] A. V. Kotikov, A. V. Lipatov, B. G. Shaikhatdenov, and P. Zhang, JHEP **02**, 028 (2020). 1911.01445.

[56] H. Jung, S. T. Monfared, and T. Wening (2021). 2102.01494.

[57] E. Blanco, A. van Hameren, H. Jung, A. Kusina, and K. Kutak, Phys. Rev. D **100**, 054023 (2019). 1905.07331.

[58] K. J. Golec-Biernat and M. Wusthoff, Phys. Rev. D **59**, 014017 (1998). hep-ph/9807513.

[59] J. Blumlein, *On the k_T dependent gluon density in hadrons and in the photon*, in *'95 QCD and high-energy hadronic interactions. Proceedings, 30th Rencontres de Moriond, Moriond Particle Physics Meetings, Hadronic Session, Le Arcs, France, March 19-25, 1995*, pp. 191–197. 1995. Also in preprint hep-ph/9506446.

[60] K. Kutak and S. Sapeta, Phys. Rev. D **86**, 094043 (2012). 1205.5035.

[61] K. Kutak, Phys. Rev. D **91**, 034021 (2015). 1409.3822.

[62] P. Kotko, K. Kutak, S. Sapeta, A. M. Stasto, and M. Strikman, Eur. Phys. J. C **77**, 353 (2017). 1702.03063.

[63] M. G. Echevarria, T. Kasemets, P. J. Mulders, and C. Pisano, JHEP **07**, 158 (2015). 1502.05354.

[64] M. Bury, A. van Hameren, H. Jung, K. Kutak, S. Sapeta, and M. Serino, Eur. Phys. J. C **78**, 137 (2018). 1712.05932.

[65] A. Signori, A. Bacchetta, M. Radici, and G. Schnell, JHEP **1311**, 194 (2013). 1309.3507.

[66] A. Bacchetta, F. Delcarro, C. Pisano, and M. Radici (2020). 2004.14278.

[67] A. Vladimirov, JHEP **10**, 090 (2019). 1907.10356.

[68] M. Bury, A. Prokudin, and A. Vladimirov (2020). 2012.05135.

iset	uPDF/TMD set	Subsets	Ref.
101000	ccfm-JS-2001	1	[51]
101010	ccfm-setA0	4	[51]
101020	ccfm-setB0	4	[51]
101001	ccfm-JH-set1	1	[52]
101002	ccfm-JH-set2	1	[52]
101003	ccfm-JH-set3	1	[52]
101201	ccfm-JH-2013-set1	13	[53]
101301	ccfm-JH-2013-set2	13	[53]
101401	MD-2018	1	[54]
101410	KLSZ-2020	1	[55]
102100	PB-NLO-HERAI+II-2018-set1	35	[40]
102200	PB-NLO-HERAI+II-2018-set2	37	[40]
102139	PB-NLO-HERAI+II-2018-set1-q0	3	[40]
102239	PB-NLO-HERAI+II-2018-set2-q0	3	[40]
103100	PB-NLO+QED-set1-HERAI+II	1	[56]
103200	PB-NLO+QED-set2-HERAI+II	1	[56]
10904300	PB-NLO_ptoPb208-set1	1	[57]
10904400	PB-NLO_ptoPb208-set2	1	[57]
10901300	PB-EPPS16nlo_CT14nlo_Pb208-set1	1	[57]
10901400	PB-EPPS16nlo_CT14nlo_Pb208-set2	1	[57]
10902300	PB-nCTEQ15FullNuc_208_82-set1	33	[57]
10902400	PB-nCTEQ15FullNuc_208_82-set2	33	[57]
200001	GBWlight	1	[58]
200002	GBWcharm	1	[58]
210001	Blueml	1	[59]
400001	KS-2013-linear	1	[60]
400002	KS-2013-non-linear	1	[60]
400003	KS-hardscale-linear	1	[61]
400004	KS-hardscale-non-linear	1	[61]
400101	KS-WeizWill-2017	1	[62]
500001	EKMP	1	[63]
410001	BHKS	1	[64]
300001	SBRS-2013-TMDPDFs	1	[65]
300002	SBRS-2013-TMDPDFs-par	1	[65]
601000	PV17_grid_pdf	201	[44]
602000	PV17_grid_ff_Pim	201	[44]
603000	PV17_grid_ff_Pip	201	[44]
604000	PV17_grid_FUUT_Pim	100	[44]
605000	PV17_grid_FUUT_Pip	100	[44]
606000	PV19_grid_pdf	216	[45]
607000	PV20_grid_FUTTsin_P_Pim	101	[66]
608000	PV20_grid_FUTTsin_P_Pip	101	[66]
701000	SV19_nnlo	23	[46]
702000	SV19_nnlo_all=0	21	[46]
703000	SV19_n3lo	23	[46]
704000	SV19_n3lo_all=0	21	[46]
705000	SV19_ff_pi_n3lo	23	[46]
706000	SV19_ff_pi_n3lo_all=0	21	[46]
707000	SV19_ff_K_n3lo	23	[46]
708000	SV19_ff_K_n3lo_all=0	21	[46]
709000	SV19_pion	7	[67]
710000	SV19_pion_all=0	7	[67]
711000	BPV20_Sivers	25	[68]

Table 1: Available uPDF/TMD parton sets in TMDlib.