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

H. Jung, et al
DESY, Hamburg, FRG

Abstract

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 OSX systems

Programming Language used: C++

High-speed storage required: No

Separate documentation available: No

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

Other programs used: LHAPDF (version 6) for access to collinear parton distributions, ROOT (version higher than 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)

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

1 Introduction

The Quantum Chromodynamics (QCD) interpretation of high-energy particle reactions requires a simultaneous treatment of processes at different energy scales. Factorisation theorems provide the mathematical framework to properly separate the physical regimes. For instance, when two protons collide in a Drell-Yan (DY) event the high-energy partonic cross section is described with a perturbative QCD expansion and the soft physics underlying the structure of the hadrons is treated with parton distribution functions (PDFs), supplemented by QCD evolution. "Evolution", in this context, refers to the scale dependence of parton distributions (and similar non-perturbative objects) that arises in a detailed treatment of factorisation in QCD perturbation theory. A classic example of a consequence of QCD evolution is the violation of Bjorken-scaling in inclusive deep-inelastic lepton-hadron scattering (DIS), predicted by the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) evolution equations [1–3].

In this paper, we describe a new tool for collecting different fits and parameterisations into a single library, TMDlib, and the online plotter tool, TMDplotter. Provided that the user takes into account all the possible differences between formalisms, collecting parameterisations for both the objects in TMDlib and TMDplotter will also make phenomenological comparisons easier.

2 TMDlib documentation

TMDlib is a `C++` library which provides a framework and an interface to a collection of different uPDF/TMD parameterisations. The parameterisations of TMDs in TMDlib are explicitly authorised for distribution in TMDlib by the authors. No explicit QCD evolution code is included: the parameterizations are as given in the corresponding references. In the present version of TMDlib no attempt is made to unify grid files and the interpolation procedure, both are those provided by the authors. However, since the grids provided for the PB-method are already well advanced and used in many applications, it is recommended to use these grid structures for further TMD parametrizations.

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 [4] and the `ROOT` data analysis framework library [5] (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.

The TMDlib library is released together with the online plotter platform TMDplotter, available at `http://tmdplotter.desy.de/`. Two snapshots from a typical usage of TMDplotter are shown in Fig. 1.

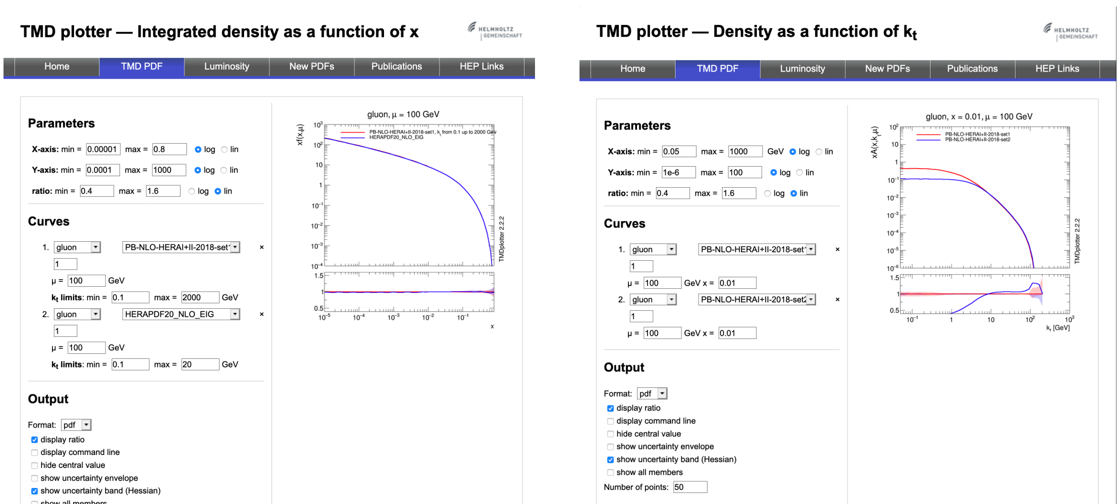

Figure 1: Two snapshots from the online portal TMDplotter for plotting: (left): comparison of the integrated distribution obtained from LHApdf with the k_t integrated distribution within TMDplotter. (right): comparsion of different PB-distributions. In all distributions uncertainty bands are included.

3 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 $0,\ldots,6$ is $\bar{t},\ldots,\bar{d}$, at index 7 is the gluon, and at index $8,\ldots,13$ 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 $0,\ldots,6$ is $\bar{t},\ldots,\bar{d}$, at index 7 is the gluon, and at index $8,\ldots,13$ 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`.
`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` .

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` .

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 of data set from `info` file.
`TMDgetNumMembers()`	Returns number of members of data set description 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 $\alpha_{\rm s}$ 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 associated with the `name` of dataset.
`TMDstringPDF(index)`	Returns the name associated with `index` of the dataset.

4 TMDlib Calling sequence

In the following simple examples are given 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
```

5 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 parametrizations is given in Tab. **??**.

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

```
# install all data sets
bin/TMDlib-getdata all
```

6 Structure of TMD grids

In TMDlib2 the TMDgrids are stored on a directory with the name of the TMD as subdirectory, for example a TMD named `test`:

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

The `info` file contains general information on the TMDset, 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` 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 Pavia etc TMD series

7 Summary

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

Acknowledgments

Many thanks to the authors of TMD calculations, especially M. Bury, E. Bianco, K. Kutak, A. Kusina and a very special thank you to M. Rohrmoser for his help with C++ issues.

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. **B126**, 298 (1977).

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

[4] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, Eur. Phys. J. **C75**, 132 (2015). 1412.7420.

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

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

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

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

[9] A. Bermudez Martinez, P. Connor, F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, Phys. Rev. **D99**, 074008 (2019). 1804.11152.

[10] E. Blanco, A. van Hameren, H. Jung, A. Kusina, and K. Kutak, Phys. Rev. **D100**, 054023 (2019). 1905.07331.

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

[12] 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.

[13] K. Kutak and S. Sapeta, Phys.Rev. **D86**, 094043 (2012). 1205.5035.

[14] K. Kutak, Phys. Rev. **D91**, 034021 (2015). 1409.3822.

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

[16] M. Bury, A. van Hameren, H. Jung, K. Kutak, S. Sapeta, and M. Serino, Eur. Phys. J. **C78**, 137 (2018). 1712.05932.

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

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

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

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

Identifier	uPDF/TMD set	Subsets	Ref.
101000	ccfm-JS-2001	1	[6]
101010	ccfm-setA0	4	[6]
101020	ccfm-setB0	4	[6]
101001	ccfm-JH-set1	1	[7]
101002	ccfm-JH-set2	1	[7]
101003	ccfm-JH-set3	1	[7]
101201	ccfm-JH-2013-set1	13	[8]
101301	ccfm-JH-2013-set2	13	[8]
101800	PB-LO-HERAI+II-2020-set1	1	
101900	PB-LO-HERAI+II-2020-set2	1	
102100	PB-NLO-HERAI+II-2018-set1	35	[9]
102200	PB-NLO-HERAI+II-2018-set2	37	[9]
102139	PB-NLO-HERAI+II-2018-set1-q0	3	[9]
102239	PB-NLO-HERAI+II-2018-set2-q0	3	[9]
10904300	PB-NLO_ptoPb208-set1	1	[10]
10904400	PB-NLO_ptoPb208-set2	1	[10]
10901300	PB-EPPS16nlo_CT14nlo_Pb208-set1	1	[10]
10901400	PB-EPPS16nlo_CT14nlo_Pb208-set2	1	[10]
10902300	PB-nCTEQ15FullNuc_208_82-set1	33	[10]
10902400	PB-nCTEQ15FullNuc_208_82-set2	33	[10]
200001	GBWlight	1	[11]
200002	GBWcharm	1	[11]
210001	Blueml	1	[12]
400001	KS-2013-linear	1	[13]
400002	KS-2013-non-linear	1	[13]
400003	KS-hardscale-linear	1	[14]
400004	KS-hardscale-non-linear	1	[14]
500001	EKMP	1	[15]
410001	BHKS	1	[16]
300001	SBRS-2013-TMDPDFs	1	[17]
300002	SBRS-2013-TMDPDFs-par	1	[17]
601000	PV17_grid_pdf	201	[18]
602000	PV17_grid_ff_Pim	201	[18]
603000	PV17_grid_ff_Pip	201	[18]
604000	PV17_grid_FUUT_Pim	100	[18]
605000	PV17_grid_FUUT_Pip	100	[18]
606000	PV19_grid_pdf	216	[19]
607000	PV20_grid_FUTTsin_P_Pim	101	
608000	PV20_grid_FUTTsin_P_Pip	101	
701000	SV19_nnlo_all=0	301	[20]
702000	SV19_nnlo	301	[20]

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