CuTe-MCFM manual

Thomas Becher[1] and Tobias Neumann[2]

[1]Universität Bern, becher@itp.unibe.ch
[2]Brookhaven National Laboratory, tneumann@bnl.gov

MCFM v10.0, CuTe-MCFM v1.0
updated March 2021

CuTe-MCFM is an extension of MCFM for q_T resummation and reaches N^{3}LL precision for all color-singlet processes included in MCFM at NNLO ($W^\pm, Z, H, \gamma\gamma, Z\gamma, ZH, W^\pm H$). We document here the resummation features and refer the reader to the the MCFM manual (manual.pdf) for information on MCFM itself.

Contents

This manual describes the installation, use and options for running CuTe-MCFM. For a description of the resummation formalism, underlying choices, and physics examples we refer to our publication *JHEP 03 (2021) 199*, arXiv:2009.11437. While the code is easy to run, some parameters must be properly chosen to obtain sensible results. These include a technical cutoff on the matching corrections at low q_T and the choice of a function which governs the transition from resummation to the fixed-order result. We will provide the definition of the parameters in this manual, and the appropriate choice for different processes is detailed in our publication.

Processes available for resummation

The processes listed below are available for N^{3}LL+NNLO computations. They can be calculated with or without decays. Any further color-singlet processes implemented in MCFM at NNLO (NLO) can easily be interfaced with N^{3}LL (N^{2}LL) q_T resummation.[1]

[1]We are happy to provide details about how to do this.

The first set of processes below have been thoroughly studied as part of our publication. We provide input files with a suitably chosen set of input parameters for all of them.

- $W^+(\to e^+\nu)$ (`nproc=1`) (`input_W+.ini`)
- $W^-(\to e^-\bar{\nu})$ (`nproc=6`) (`input_W-.ini`)
- $Z(\to e^+e^-)$ (`nproc=31`) (`input_Z.ini`)
- $H(\to \gamma\gamma)$ (`nproc=119`) (`input_Higgs.ini`)
- $\gamma\,\gamma$ (`nproc=285`) (`input_GammaGamma.ini`)
- $gg \to \gamma\,\gamma$ at N^{2}LL+NLO (`nproc=2851`)
- $Z(\to e^+e^-)\,\gamma$ (`nproc=300`) (`input_ZGamma.ini`)

The second set of processes below are ready to use, and we have checked that the fixed-order expansion of the resummed result and the fixed-order cross-section approach each other towards $q_T \to 0$. However, the transition function parameter might need adjustment. Please contact the authors if you are unsure about this.

Z production:

- $Z(\to 3\nu\bar{\nu})$ (`nproc=32`)

Higgs production:

- $H(\to \tau\bar{\tau})$ (`nproc=112`)
- $H(\to b\bar{b})$ (`nproc=111`)

$Z\gamma$ production:

- $Z(\to 3\nu\bar{\nu})\,\gamma$ (`nproc=305`)

$W^{\pm}H$ production:

- $W^+(\to e^+\nu)\,H(\to \tau\bar{\tau})$ (`nproc=91`)
- $W^+(\to e^+\nu)\,H(\to b\bar{b})$ (`nproc=92`)
- $W^-(\to e^-\bar{\nu})\,H(\to \tau\bar{\tau})$ (`nproc=96`)
- $W^-(\to e^-\bar{\nu})\,H(\to b\bar{b})$ (`nproc=97`)
- $W^+(\to e^+\nu)\,H(\to \gamma\gamma)$ (`nproc=93`)
- $W^-(\to e^-\bar{\nu})\,H(\to \gamma\gamma)$ (`nproc=98`)

ZH production:

- $Z(\to e^+e^-)\,H(\to \tau\bar{\tau})$ (`nproc=110`)
- $Z(\to e^+e^-)\,H(\to b\bar{b})$ (`nproc=101`)
- $Z(\to e^+e^-)\,H(\to \gamma\gamma)$ (`nproc=104`)

Further decay channels are available in principle, but will need some checking. Please contact the authors if you are interested in a specific process.

MCFM compilation quick start

Please refer to the full MCFM manual for details beyond this quick start guide. In the simplest case (on most Linux systems), to install CuTe-MCFM and all dependencies for execution on a single computer, execute the command `cmake ..` in the `Bin` directory.

MCFM requires the GNU compiler gcc/g++ and gfortran version 7 or greater. Please type `gfortran --version` to verify the compiler version. On some systems these commands are linked against a different compiler. For example on Mac OS X systems this is typically the case. To set the correct compiler commands please add the flags `-DCMAKE_Fortran_COMPILER=mygfortran`, `-DCMAKE_C_COMPILER=mygcc` and `-DCMAKE_CXX_COMPILER=myg++`, where `mygfortran`, `mygcc` and `myg++` are the commands for the Fortran, C, and C++ compilers of the GNU gcc suite of at least version 7. That is, in the `Bin` directory run cmake with the specified compiler commands as in `cmake .. -DCMAKE_Fortran_Compiler=mygfortran`.

By default a bundled LHAPDF library is compiled and linked against. If you prefer to use a system installation please add the cmake options `-Duse_internal_lhapdf=Off -Duse_external_lhapdf=On`.[2] If the library is in a non-standard location another option like `-DCMAKE_PREFIX_PATH=/usr/local`, which adds the path to the cmake library search path, might be necessary.

If CMake does not report any problems you can start the compilation of MCFM with `make -j4`, where 4 (or more) is the number of compilation threads.

Upon successful compilation, the executable `mcfm` is produced and can be called with an input file as argument, for example `./mcfm input_Z.ini`.

To prepare MCFM with MPI support add the argument `-Duse_mpi=On` to the cmake call before running make. At the same time custom compiler command names must be specified with `-DCMAKE_Fortran_COMPILER=mpifort` and `-DCMAKE_CXX_COMPILER=mpic++`. The commands `mpifort` and `mpic++` must be used when compiling with MPI support. In this case, please ensure again that `mpifort ----version` and `mpic++ ----version` report the GNU compiler and a version greater than 7.

It can happen that the CMake cache gets corrupted with wrong configuration options. If you change options and errors occur, please try to delete the file `CMakeCache.txt` and directory `CMakeFiles` and restart `cmake ..` with the appropriate arguments.

Using CuTe-MCFM resummation

While CuTe-MCFM can calculate q_T-resummed results without using pregenerated beam functions grids, we recommend that LHAPDF grid files are generated for the beam functions beforehand for a choice of a PDF set. This *significantly* accelerates the evaluation of the beam functions and the integration.

CuTe-MCFM ships with pregenerated beamfunction grids for the central values of `CT14nnlo` and `NNPDF31_nnlo_as_0118`, which are included in the `Bin/PDFs` directory. This path is automatically used as the preferred path for LHAPDF grid files. With these pregenerated grids the example input files work out of the box. For other PDF sets or when using PDF errors, the first run of CuTe-MCFM should be with the setting `makegrid=.true.`. Additionally the input and output directories for the PDF grids have

[2] Do not use an external installation of version LHAPDF-6.3.0 or newer, since this has a critical multi-threading bug.

to be specified. For example the input directory is typically `/usr/local/share/LHAPDF/` (or the `PDFs/` directory relative to the mcfm executable in `Bin`) and the output directory should be a user-writeable directory like `/home/user/gridout/` (or `PDFs/`). Note the trailing slashes.

When calling mcfm with `makegrid=.true.` only the beam function grids are written during that run, and mcfm exits afterwards. We recommend to use `PDFs/` as the gridout path, since this path is automatically added to the LHAPDF search paths, and you won't have to copy the generated grid directories to your LHAPDF grid directory or set the `LHAPDF_DATA_PATH` environment variable to the gridout path.

For example for the set CT14nnlo the grid directories `CT14nnlo_B00`, `CT14nnlo_B10`, `CT14nnlo_B11`, `CT14nnlo_B20`, `CT14nnlo_B21`, `CT14nnlo_B22` and `CT14nnlo_G10` are written and have to be copied to the directory where LHAPDF searches for the grid files. When the gridout path is chosen as `PDFs/` no further action is necessary. The LHAPDF grid file search path can be modified by setting the shell environment variable `LHAPDF_DATA_PATH` to the desired directory, but the `PDFs` directory is always used as the preferred directory.

The next run of mcfm should be done with `makegrid=.false.` and `usegrid=.true.`.

Other important parameters for the resummation are `res_range`, determining the integration range of the purely resummed part, `resexp_range`, determining the integration range of the fixed-order expanded resummed part, and `fo_cutoff` which sets the lower q_T cutoff for the fixed-order part. Typically this cutoff should agree with the lower range of `resexp_range`. For example for Z production one can integrate up to m_Z with a cutoff of 1 GeV: `res_range = 0.0 90.0`, `resexp_range = 1.0 90.0`, `qt_cutoff = 1.0`.

For details regarding these parameters see the next section. The transition function is also discussed below.

Input file parameters

The `[resummation]` section has been added to the input file to control the resummation. The following keys are available:

Key	Description
`usegrid`	`.true.` or `.false.` determines whether pregenerated LHAPDF interpolation grids should be used for the resummation beam functions.
`makegrid`	If `.true.`, then MCFM runs in grid generation mode. This generates LHAPDF grid files in the directory `gridoutpath` from LHAPDF grids in the directory `gridinpath`. After the grid generation MCFM stops and should be run subsequently with `makegrid = .false.` and `usegrid = .true.`. When `lhapdf%dopdferrors=.true.` then also grids for the error sets are generated.
`gridoutpath`	Output directory for LHAPDF grid files, for example `/home/tobias/local/share/LHAPDF/`
`gridinpath`	Input directory for LHAPDF grid files, for example `/home/tobias/local/share/LHAPDF/`
`res_range`	Integration range of purely resummed part, for example `0.0 80.0` for q_T integration between 0 and 80 GeV.

Key	Description
`resexp_range`	Integration range of fixed-order expanded resummed part, for example `1.0 80.0` for q_T integration between 1 and 80 GeV.
`fo_cutoff`	Lower q_T cutoff q_0 for the fixed-order part, see eq. (1) below. Typically the value should agree with the lower range of `resexp_range`.
`transitionswitch`	Parameter passed to the plotting routine to modify the transition function, see text.

We strongly recommend to calculate resummed results with pregenerated grids, see the previous section. The integration range for the purely resummed part can be controlled with the key `res_range` and should typically be between 0 and some upper value. For example for $W^{\pm}, Z$ or H production this can just be the boson mass. For other processes there can be thresholds and this number must be selected more carefully to not run into numerical issues, see arXiv:2009.11437.

The setting `resexp_range` and `fo_cutoff` are relevant for the matching corrections. The values of the `resexp_range` determine the integration range for the fixed-order expansion of the resummed part. The minimum should typically be at least one GeV for numerical stability. For smaller values significantly more time goes into the integration, and the minimum number of Vegas calls might need to increased. For single boson processes the maximum value can again be the boson mass, although it can be set to a value where the implemented transition function fully switches to zero. The fixed-order cutoff `fo_cutoff` determines the minimum q_T for the fixed-order calculation. This should typically agree with the lower range of the `resexp_range`.

Lastly, the parameter `transitionswitch` is passed for convenience to the plotting routines where the transition function is implemented. It can be used for for an easy control of the transition region as described in the following.

Plotting routine and transition function

The following transition function is implemented for the example input files. For more details we refer to our publication. The fully matched cross-section is described in general by

$$\left.\frac{\mathrm{d}\sigma^{\mathrm{N^3LL}}}{\mathrm{d}q_T}\right|_{\text{matched to NNLO}} = t(x)\left(\frac{\mathrm{d}\sigma^{\mathrm{N^3LL}}}{\mathrm{d}q_T} + \Delta\sigma|_{q_T>q_0}\right) + (1-t(x))\frac{\mathrm{d}\sigma^{\mathrm{NNLO}}}{\mathrm{d}q_T} \qquad (1)$$

using a transition function $t(x)$. We have implemented a transition function t with $x = q_T^2/Q^2$ that smoothly switches between 1 and 0 like a sigmoid function.

Following a choice in CuTe, we first define

$$s(x;l,r,u) = \left(1+\exp\left(\log\left(\frac{1-u}{u}\right)\frac{x-m}{w}\right)\right)^{-1}, \quad m=(r+l)/2\,, \quad w=(r-l)/2\,.$$

The function $s(x)$, parametrized by l, r, u, is defined to be $s(l) = 1-u$ and $s(r) = u$. In terms of this sigmoid, our transition function $t(x; x^{\min}, x^{\max}, u)$, where $x = q_T^2/Q^2$, is then defined by

$$t(x;x^{\min},x^{\max},u) = \left\{ \begin{array}{ll} 1, & \text{for } x < x^{\min} \\ \frac{s(x;x^{\min},x^{\max},u)}{s(x^{\min};x^{\min},x^{\max},u)}, & \text{otherwise} \end{array} \right\}. \qquad (2)$$

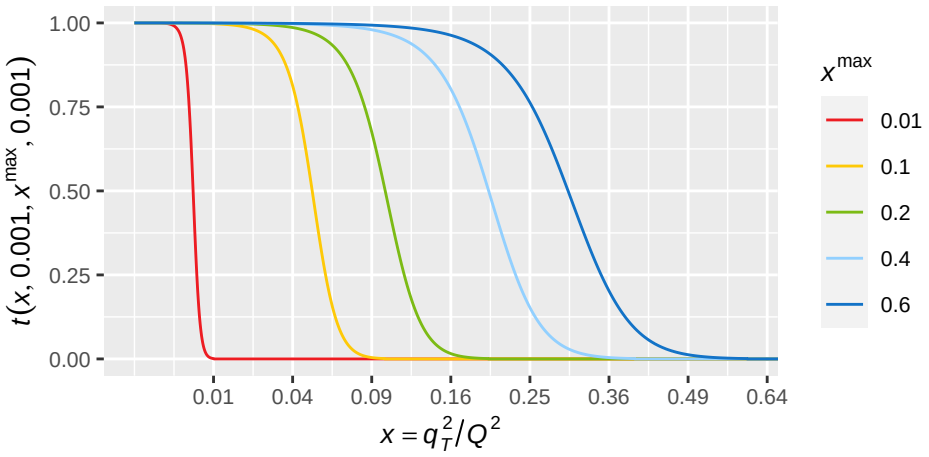

Figure 1: The transition function defined in eq. (2) for different values of the parameter $x^{\max}$ which determines the position of the transition. The x-axis is displayed on a square-root scale to guide the eye on the quadratic q_T-dependence.

This ensures that below $x^{\min} = (q_T^{\min}/Q)^2$ only the naively matched result is used, and at $x^{\max}$ for small $u \ll 1$ the transition function is approximately u. In practice it makes sense to set the transition function to zero below a small threshold like 10^{-3} without a noticeable discontinuity. This has the advantage that the deteriorating resummation and matching corrections do not impact the region of large q_T at all. Our example plotting routines use $x^{\min} = 0.001$, and $u = 0.001$, and the parameter $x^{\max}$ corresponds to the value of `transitionswitch` set in the input file. The transition function can be changed or completely replaced by just modifying the plotting routines. The following figure illustrates this transition function.

Modifying the plotting routines and transition function. The plotting infrastructure has been completely rewritten in this version of MCFM, and we recommended to only use the new infrastructure from this point on by setting `histogram%newstyle = .true.` in the input file. This is the default for the CuTe-MCFM example input files.

For the processes $W^{\pm}, Z, H$, $\gamma\gamma$, $Z\gamma$, ZH and $W^{\pm}H$ we include predefined plotting routines that can be adjusted. For example for Z production the plotting routine is in the file `src/User/nplotter_Z_new.f90`, and similarly for the other processes. The routine `setup` defines all histograms with custom or uniform binning and names. The number of used histograms needs to be allocated in this routine. The routine `book` is called for each phase space point. Through the boolean variable `abovecut` it is known whether the routine is called for "boosted $q_T = 0$" (resummed part and fixed-order expansion of resummed part) or for $q_T > 0$ (fixed-order). All provided example input files use the transition function as defined above, see also arXiv:2009.11437.

The plotting routine returns the calculated observables in the `vals` array, and Vegas weights in `wts`. The transition function is implemented by reweighting the original Vegas weights with the output of the transition function. To disable the transition function, one sets `trans` to 1 before filling the `wts` array.

Apart from modifying a default set of kinematical cuts in the input file, cuts can also be set in the file `src/User/gencuts_user.f90` in a fully flexible way based on the event's four momenta. Some commented out examples are included there.

Development details

We briefly document information for modifying and extending the resummation code.

Relevant files for q_T resummation

- `src/Mods/mod_Beamfunctions.f90`: Implementation of beam functions. `getbeam` returns beam function components for a specified power of α_s and $L_\perp$ for $q\bar{q}$ initial states, while `getbeam2` is for gg initial states.
- `src/Mods/mod_ResummationGrid.f90`: Implementation of LHAPDF grid generation for beam functions, with implementations for OpenMP, MPI and Fortran Coarrays.
- `src/Mods/mod_Resummation_params.f90`: Saves integration range variables from input file.
- `src/Mods/mod_ResummationFourier.f90`: Implementation of Fourier integral for both $q\bar{q}$ and gg initial states.
- `src/Mods/mod_Resummation.f90` Ties together all components above in subroutine `resummation`; determines scale q^*; proper evolution of α_s over quark mass thresholds; includes procedure for recoil boost;
- `src/Procdep/resint.f90`: Overall integration routine that generates phase-space; calls boost routine; evaluates matrix elements; calls resummation and fixed-order expansion of resummation