Top++: a program for the calculation of the top-pair cross-section at hadron colliders

Michał Czakon[a], Alexander Mitov[b]

[a] *Institut für Theoretische Teilchenphysik und Kosmologie, RWTH Aachen University, D-52056 Aachen, Germany*
[b] *Theory Division, CERN, CH-1211 Geneva 23, Switzerland*

Abstract

We present the program `Top++` for the numerical evaluation of the total inclusive cross-section for producing top quark pairs at hadron colliders. The program calculates the cross-section in a) fixed order approach through approximate NNLO and b) by including soft-gluon resummation for the hadronic cross-section in Mellin space with full next-to-next-to-leading logarithmic accuracy. The program offers the user significant flexibility through the large number (31) of available options. `Top++` is written in C++. It has a very simple to use interface that is intuitive and directly reflects the physics. The running of the program requires no programing experience from the user.

Program summary

Name of the program:	Top++ (ver. 1.0).
Program's homepage:	`http://www.alexandermitov.com/software`
License, Warranty:	GNU Public License. No warranty given or implied.
Compiler:	Developed and tested with GNU Compiler Collection's C++ compiler.
Operating system:	Linux; Mac OS X; can be adapted for other unix systems.
Program language:	C++.
Memory required to execute:	Typically less than 200 MB.
External libraries:	GNU Scientific Library (GSL); the Les Houches Accord PDF Interface (LHAPDF).
Keywords:	Top-quark, Resummation, QCD, Precision Physics, Hadron Colliers.
Typical running time:	Depending on the options. The program is optimized for speed.
Accuracy:	Sub per-mill accuracy achievable in realistic time (program does not employ Monte Carlo methods).

*Preprint numbers: CERN-PH-TH/2011-303, TTK-11-58

1. Introduction: what is this program for?

This program calculates the top-pair total inclusive cross-section in hadron collisions. It has built-in the most advanced theoretical results available in the literature. The program is able to calculate the total top-pair cross-section in both pure fixed order perturbation theory and by including soft-gluon resummation through next-to-next-to-leading logarithmic order (NNLL). `Top++` is the first publicly available program that can perform soft gluon resummation in top-pair production.

The program is written in C++ in a modern, modular and object-oriented way. It should be very easy to install on most Linux systems; please consult Appendix A for details. Moreover, once installed, the program is trivial to run. The program has been written with a user in mind that has no programming experience whatsoever. For that reason, the program has a very simple user interface that is the only point of contact between the program and the user. The user interface is described in Section 2. Users with average programming experience will find the program very easy to customize to their own needs. That may not be needed, however, since all options that have arisen in our own work on the subject have already been pre-programmed and are easy to access directly through the user interface.

In this manual we do not describe the physics in detail. A short description, needed to make the reading of the program's options self-consistent, can be found in Section 4. The relevant description of the physics and the options implemented in this program can be found in Ref. [1]. The results in that paper represent the most advanced results in top physics to date. Given the fast pace of developments in the subject, however, we expect that new theoretical results will be available soon. In fact, one of the main goals of this program is to seamlessly accommodate such future updates.

And an important word of caution regarding the interpretation of numerical results derived with the help of the program. Clearly a program with a number of options (like ours) is capable of correctly producing almost any number. To be able to interpret such a number, however, one needs to know exactly how it was derived. This can become a problem with the current state of top physics where the most advanced results constitute a series of approximations (see Ref. [1]). To reduce the chance of mis-communicating results derived with this program, we have implemented a default setting that corresponds to the best prediction of Ref. [1], and we urge the users to use it for publications. Parameters that are not considered part of that setting are the parton distribution functions (PDF) and the value of the top quark mass. These should always be quoted to ensure complete documentation of the results. Other settings, mainly related to the workings of the program, can affect the final numerical value (through numerical precision) but that should be at a precision level below what is physically relevant. As a rule of thumb we urge the user to produce numbers that are correct at the level of one per-mill (i.e. $\mathcal{O}(10^{-3})$). Precision better than that is a pure matter of taste on the side of the user (and, at a certain level, of the ability of the otherwise very capable integration routines).

2. How to use the program

In the following we assume that the program has been installed and is running correctly; see Appendix A for more details. Next, we assume that the program is installed in a directory called `top++` and we are already there, i.e. the command line reads: [1]

```
~ top++$
```

Next one needs to open the file *top++.cfg* in a text editor. Using, for example, *pico* one types:

```
~ top++$ pico top++.cfg
```

Once the user has set all options at their desired values, the file needs to be saved. A number of examples are supplied with the program; see Appendix B. Then the user needs to execute the program by typing:

[1] Please note that the precise text on the left of the cursor depends on the particular terminal.

```
~ top++$ ./top++
```

The programs starts running, displays its step-by-step progress, timing pre-defined milestones. Once the run is completed, the final result is displayed and the program exits. The output can be found on the screen and in the file *top++.res* located in the program's directory. If a new run is desired, one has to simply repeat all the steps described above.

In the following we describe all 31 options that are available to the user through the file *top++.cfg*. The options are grouped into five subgroups. Please note that since all options have predefined default values, the user needs to only specify the values for options that differ from their default values. In particular, if all options are at their default values the file *top++.cfg* can be empty. It might be convenient to keep certain options typed in the file, but when not in use the user can comment them out by putting the symbol / at the beginning of a line.

1. **General Setup** (type of collider, PDF set, pure fixed order calculation versus one with resummation).

   (a) `Collider:` Takes two values: `TEV` (default) or `LHC` as labels for $\bar{p}p$ or pp colliders.

   (b) `WithResummation:` If set to `YES` (default), the program will compute the observable at $N^nLO + N^kLL$ depending on the values for n and k specified in the subgroup **Resummation** below. In this case all settings in the subgroup **Fixed Order** become irrelevant.

   (c) `PDFuncertainty:` If set to `YES` the program will compute and display the PDF uncertainty. Two methods are used, depending on the PDF set. The prescription for the NNPDF family of sets [2] is selected automatically for any NNPDF set, while for all other PDF sets the program uses the asymmetric prescription of Ref. [3] (see also Ref. [4]). In particular, the latter prescription is the most appropriate one for the MSTW2008 family of sets [5]. When the PDF uncertainty is being calculated, central scales choice $\mu_F = \mu_R = m_t$ is set automatically, independently of the values for these scales set by the user (as described below). If `PDFuncertainty` is set to `NO` (default) then the program computes and displays the scale variation for the choice of μ_F, μ_R specified by the user (explained below) for a single pdf member (specified by the user; see option 1f below). Please note that the definition of a "central" set depends on the PDF set.

   (d) `RestrictedScaleVariation:` Takes either `NO` or a number greater than or equal to 1.0. This option allows the user to set any restriction (or no restriction at all if `RestrictedScaleVariation NO`) on the allowed ratio for the renormalization and factorization scales. For more information see the description of the group of options **Top quark mass and renormalization/factorization scales**. The default value is `2.0` which corresponds to the restricted scale variation of Ref. [1].

   (e) `PDFset:` The PDF set. The program uses the LHAPDF library and follows its nomenclature. The default set is `MSTW2008nnlo68cl`.

   (f) `PDFmember:` The specific member of the PDF set `PDFset` the user wishes to use for the calculation of the cross-section. Please note that the definition of a"central" set depends on the PDF set. Default value is `0`.

2. **Top quark mass and renormalization/factorization scales** Please note that the ranges of `muR` and `muF` (defined in the following) that the user may provide need not be equal in length. The range of their ratio is unrestricted if `RestrictedScaleVariation NO` is chosen. When the option `RestrictedScaleVariation` takes a numerical value (with the constraint `RestrictedScaleVariation` $\geq$ 1.0), then the ratio of the two scales is restricted between:

$$\frac{1}{\texttt{RestrictedScaleVariation}} \leq \mu_F/\mu_R \leq \texttt{RestrictedScaleVariation}\,.$$

(a) `Mtop:` The value of the on-shell top mass (in GeV). If the user would like to loop over a range of values for the top mass (see the description of the following two options) `Mtop` represents the lower end of that range. Default value is `173.3`.

(b) `MtopLimit:` The upper limit of the range of values for the top mass that the user wants to loop over. If `MtopLimit` is not specified, then it is set automatically equal to the top mass and therefore the calculation is performed for a single value of the top mass.

(c) `MtopStep:` A positive number (need not be an integer) that specifies the step with which the value of the top mass is incremented in a loop. Its default value is `1`.

(d) `muR:` Set of values for the renormalization scale (in units of the top mass `Mtop`). By default it takes the value 1.0. It can take an arbitrary number of values. In particular, to determine the scale variation of a cross-section, one may use the set:

`muR 0.5 1.0 2.0 .`

A fine scan can be achieved with the help of the following set:

`muR 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 .`

The larger set takes much longer to compute and typically returns the same result as the restricted set consisting of three elements. We have encountered, however, exceptional situations that in our experience lead to differences of up to 0.5%.

(e) `muF:` Set of values for the factorization scale (in units of the top mass `Mtop`). It is set and used independently of the renormalization scale described above. Its default value is 1.0 and can be set analogously to `muR`.

3. **Resummation** (all options in this group are irrelevant when `WithResummation NO` is chosen).

(a) `OrderFO:` Takes values `LO, NLO` (default) or `NNLO`. It specifies the order n of the fixed order cross-section that should be added to the resummed result, i.e. $N^nLO + N^kLL$. For example `LO` implies $n = 0$, `NLO` implies $n = 1$, etc. The option `NNLO` includes at present the approximately known NNLO cross-section [6] described in the group of options **Fixed Order** below. We advise against using this option until the exact NNLO result becomes available, see Ref. [1] for details.

(b) `OrderRES:` Takes the values `LL, NLL` and `NNLL` (default). It represents the logarithmic order k of the resummation, i.e. $N^nLO + N^kLL$. For example `LL` implies $k = 0$, `NLL` implies $k = 1$, etc. Full $NNLL$ resummation is implemented matched to a hard function whose coefficients are still unknown and their values are set to zero by default (however see the description of the options `H2qq, H2gg1, H2gg8` below).

(c) `A:` The value of the parameter A introduced in Ref. [7]. Typically used with the values `A 2` (default) and `A 0`.

(d) `RESonORoff:` Takes the values `ON` or `OFF` (default); specifies how the unknown two-loop hard function is treated. We have implemented this option consistently both for the fixed order (relevant if `OrderFO` is set to `NNLO`) and the resummed components. Moreover, the unknown constants C_{qq} and C_{gg} in the fixed order NNLO approximation are automatically expressed through the constant in the N-space hard function (described in option 3f).

(e) `TwoLoopCoulombs:` Takes the values `YES` (default) and `NO`. It includes/excludes the two loop Coulombic terms in the function $\sigma^{(\mathrm{Coul})}$, see Eq. (2) below.

(f) `H2qq, H2gg1, H2gg8:` The three unknown constants in the two-loop hard function. They are defined in a normalization α_S/π. They are irrelevant when `RESonORoff` is set to `OFF`. When `RESonORoff` is set to `ON`, one needs to vary them to determine the full theoretical uncertainty. The default value for all three constants is `0.0`.

4. **Fixed Order** (all options in this group are irrelevant when `WithResummation YES` is chosen).

   (a) `LO:` takes values `YES` (default) or `NO`.

   (b) `NLO:` takes values `YES` (default) or `NO`. For example, to perform a calculation in NLO QCD one needs to set `LO YES` and `NLO YES`. This way the user has direct access to the individual orders in the perturbative expansion. Same applies to the option `NNLO` below.

   (c) `NNLO:` takes values `YES` (default) or `NO`. Please note that this option currently implements the threshold approximation to the NNLO cross-section [6], and not the exact result which is still unknown.

   (d) `NNLOonORoff:` Takes the values `ON` or `OFF` (default). Specifies how the $\ln(\mu)$ terms in the approximate NNLO result are treated; see Ref. [1].

   (e) `Cbarqq, Cbargg:` The values for the yet-unknown two-loop constants C_{qq} and C_{gg} in the threshold expansion of the NNLO cross-section [6] (defined in a normalization $\alpha_S/(4\pi)$). As explained in Ref. [1] the default values for these constants (corresponding to `H2qq` = `H2gg1` = `H2gg8` = 0) are `Cbarqq` $= -489.168$, `Cbargg` $= -1136.81$.

5. **Setup parameters** (parameters related to the working of the program and other, less frequently modifiable parameters).

   (a) `ECMLHC:` The c.m. energy of the pp collider (in GeV). The default value is `7000`.

   (b) `ECMTEV:` The c.m. energy of the $\bar{p}p$ collider (in GeV). The default value is `1960`.

   (c) `Precision:` Defines the required relative precision of the integration routines:

   $$\text{Relative precision} = 10^{-\texttt{Precision}}\,.$$

   The default value `precision 2` tends to produce fast and accurate (at the per-mill level) results.

   (d) `NPdfGrid:` Defines the size of the grid on which the PDF fluxes are being discretized. The default value `NPdfGrid 100` tends to produce fast and sufficiently accurate results.

   (e) `ETA:` Parameter that controls the subtraction flux implemented in the resummed calculation. The default value `ETA 1e-5` is optimal. [2] The user normally will not need to be concerned with this parameter.

   (f) `CMP:` The value where the contour for the Mellin inverse transform crosses the real line. The default value `CMP 2.7` is optimal. The user normally need not be concerned with this parameter.

[2] We use the usual notation: $1e\text{-}p \equiv 10^{-p}$.

3. Once the program is running: some fine tuning

3.1. Numerical precision and speed

As every program for numerical calculations, `Top++` has its limits, too. In the following we discuss this, as well as ways to improve the accuracy and length of the runs.

There are two places where speed (and therefore accuracy) can be controlled. The first one is through the option `Precision` of the integration routines. In our own experience the value `Precision 2` is more than adequate to calculate the $t\bar{t}$ cross-section to per-mill accuracy. Increasing the value of `Precision` slows the calculation down. The program is set in such a way, that if the integration routines cannot reach the accuracy in a point, a warning message is displayed specifying the relative error returned by the integration routines. The presence of these messages, as such, is harmless. They can be ignored if the displayed relative precision is high enough. Rarely, the output might contain the symbol NaN, which stands for Not-a-Number. NaNs are returned due to invalid numerical operations such as division by zero, or operations, which produce numbers out of the range of double precision. This is a sign of numerical instability. We have taken the pragmatic approach of not handling such exceptions in any specific way (which is very difficult in practice). In case of a NaN result, the user should rerun the calculation with increased requested accuracy.

We recommend that at least once the user does the calculation with a larger value for `Precision` and verifies that the change in the result is beyond the required accuracy.

A second (and independent) source of numerical uncertainty is the size `NPdfGrid` of the grid over which the partonic fluxes are approximated. We have implemented a second order finite difference scheme. The relative precision scales as $\sim 1/\texttt{NPdfGrid}^2$. In practice we have noticed that a value `NPdfGrid 100` is more than adequate in terms of accuracy and produces very fast calculations. We recommend that at least once the user does the calculation with a larger value, say `NPdfGrid 500`, and verifies that the change in the result is beyond the required accuracy.

Finally, the user should keep in mind that the overall numerical accuracy is a combination of the settings for `Precision` and `NPdfGrid`, i.e. increasing only one of them to an extreme may not have a net positive effect on the overall uncertainty but might lead to a significant slowdown.

3.2. Outputting the results

The program outputs on the screen the information about the timing of each step and all results. The specifics depend on the requested options. At the end of the calculation, a summary of the final result is displayed, including the central value (i.e. the value corresponding to central scale choice, if requested) and the scale or PDF variation's absolute and relative values. The program also outputs the final result in a file `top++.res` which is ready for plotting with the program *Gnuplot*. This is particularly useful for the case when the user requests a loop over a range of values of the top quark mass. In such case the result for each value is conveniently recorded in the file.

3.3. Modifying the default values of the parameters

The default values for all 31 parameters available to the user are set inside the function `main()` located in the file `top++.cpp`. All default settings can be modified by the user, although this is not recommended and should not be necessary.

3.4. Additional

The strong coupling constant is calculated at a scale μ_R through the LHAPDF interface [8]. The calculation of the cross-section is performed in a scheme with $N_F = 5$ active flavors. For consistency only PDF sets with $N_F = 5$ active flavors should be used for scales above m_{top}.

4. Contact with physics

The program `Top++` comes with settings that reproduce the "best" numbers from Ref. [1]. In our view they represent the best current theoretical predictions with realistic theoretical uncertainties. All physics-related inputs are defined and explained in Ref. [1], which should be consulted if more detailed questions arise.

The program calculates the $t\bar{t}$ total inclusive cross-section, schematically given as:

$$\sigma_{\rm tot} = \sigma^{(n)}_{\rm F.O.} + \left[\sigma^{\rm res} - \sigma^{\rm res}|_{\alpha_S^n}\right] . \tag{1}$$

The power n is set by the option `OrderFO` if resummation is desired (i.e. when `WithResummation YES`). The convention is that $n = 2$ corresponds to LO, $n = 3$ to NLO and $n = 4$ corresponds to NNLO. If `WithResummation NO`, i.e. one wishes to calculate in a fixed order approach, then the terms in the square bracket in Eq. (1) are absent and n is set through the options `LO, NLO` and `NNLO`.

In Mellin N-space, the resummed partonic cross-section reads

$$\sigma^{\rm res\ part}_{N,\mathbf{I}} = \sum_{\mathbf{I}=\mathbf{1},\mathbf{8}} \sigma^{(\rm Coul)}_{N,\mathbf{I}} \times \sigma^{(\rm Hard)}_{N,\mathbf{I}} \times \Delta_{N,\mathbf{I}} . \tag{2}$$

The function $\sigma^{(\rm Coul)}$ contains the Coulombic effects and has a known perturbative expansion through NNLO. The function $\sigma^{(\rm Hard)}$ is an N-independent hard function. Its NNLO constant terms are currently unknown. These constants, in an expansion in α_S/π, correspond to the options `H2qq, H2gg1, H2gg8`. The $\ln(\mu)$ terms in the two-loop correction to the function $\sigma^{(\rm Hard)}$ are included if `RESonORoff ON` and excluded when `RESonORoff OFF`. The latter option gives a more conservative (i.e. realistic) determination of the theoretical uncertainties through scale variation alone.

The function $\Delta_{N,\mathbf{I}}$ contains the towers of LL, NLL and NNLL soft $\ln(N)$ logs.

By setting the option `OrderRES LL` the program included only the LL corrections from $\Delta_{N,\mathbf{I}}$, as well as the LO terms in the functions $\sigma^{(\rm Coul)}$ and $\sigma^{(\rm Hard)}$. If `OrderRES NLL` then $\Delta_{N,\mathbf{I}}$ includes LL and NLL logs, while the functions $\sigma^{(\rm Coul)}$ and $\sigma^{(\rm Hard)}$ are given by their NLO expansions. Similarly, for `OrderRES NNLL` $\Delta_{N,\mathbf{I}}$ includes LL, NLL and NNLL logs, and the functions $\sigma^{(\rm Coul)}$ and $\sigma^{(\rm Hard)}$ are given by their expansions through NNLO. The expressions for these functions can be found in Ref. [1].

For pure fixed order calculations one also has a variety of options for the treatment of the currently approximately known NNLO corrections. Setting `NNLOonORoff ON` one includes the two-loop $\ln(\mu)$ terms unaccompanied by a power of β or $\ln\beta$. By setting `NNLOonORoff OFF` one excludes these two-loop logarithmic terms which corresponds to the standard setting adopted in Ref. [1].

Through the options `Cbarqq, Cbargg` one has a direct access to the currently unknown two-loop constants $C^{(2)}_{qq}$ and $C^{(2)}_{gg}$. This is helpful for determining the sensitivity of the approximate NNLO cross-section to these unknown parameters.

5. Summary

We present the C++ program `Top++` for the calculation of the total top-pair cross-section at hadron colliders. This is the first publicly available program capable of performing soft-gluon resummation for this collider observable. The program uses all currently available theoretical results. The user has access to 31 options which results in a great deal of flexibility and control over the calculation on the side of the user. In this manual we have given only a very short introduction to the physics behind our program. For further details the user should consult Ref. [1].

The program is organized in a modular, object oriented way. It can seamlessly accommodate future results, notably the complete NNLO results, by combining them with the currently available NNLL resummation.

The program is optimized for speed given the user's requirements for accuracy. Our experience shows that, depending on the chosen options, the run times can vary significantly. In practice the speed can be an

issue only for resummed calculations, due to the integration of rapidly oscillating functions in the complex plane. For fixed order calculations the run times are very short.

As we already emphasized `Top++` is the only publicly available program capable of producing resummed calculations. Another program for the calculation of the total cross-section in fixed order perturbation theory is available in the literature [9]. While we have not performed detailed side-by-side comparisons, our experience shows that our program can be significantly faster, especially when higher accuracy is required (since we do not use Monte Carlo integration methods). We would like to also note that a direct comparison between the results of `Top++` and `HATHOR` [9] for the approximate NNLO results may not be possible. The reason is that different implementations of this approximation are contained in both programs. In `Top++` we have implemented two versions of this approximation, depending on whether the two-loop scale dependent terms are included or excluded (see Ref. [1] for details). This is controlled through the option `NNLOonORoff OFF/ON`. An approximation analogous to our `NNLOonORoff ON` is implemented in HATHOR by using the option `LOG_ONLY`. We note that while the two are compatible they are not identical, since different subleading terms $\mathcal{O}(\beta)$ are included in both implementations.

We would also like to stress (see Ref. [1]) that approximations like `NNLOonORoff ON` could significantly underestimate the theoretical uncertainty if its estimate is based on scale variation alone.

Acknowledgments

We thank Matteo Cacciari, Michelangelo Mangano and Paolo Nason for numerous cross-checks and a very fruitful collaboration that led to the creation of this software. The work of M.C. was supported by the Heisenberg and by the Gottfried Wilhelm Leibniz programmes of the Deutsche Forschungsgemeinschaft, and by the DFG Sonderforschungsbereich/Transregio 9 "Computergestützte Theoretische Teilchenphysik".

Appendix A. Installation

The program has been written in standard `C++` and has been tested to correctly compile under the GNU compiler `g++` from version 4 upwards. It requires two external libraries:

- **GNU Scientific library**, which can be downloaded free of charge from

  `http://www.gnu.org/s/gsl/`

  and is used for special functions and integration

- **the Les Houches Accord PDF Interface**, which can be downloaded free of charge from

  `http://projects.hepforge.org/lhapdf/`

  and is used for the parton distribution functions

To setup the program for installation, it is necessary to set three variables in the `Makefile` contained in the `Top++` installation directory.

- `CXX` - c++ compiler. specify the full path if necessary
- `GSLDIR` - prefix directory for the gsl library, it is assumed that the library is in GSLDIR/lib and the include files in GSLDIR/include
- `LHADIR` - prefix directory for the Les Houches PDF library, it is assumed that the library is in LHADIR/lib and the include files are in LHADIR/include/LHAPDF

In the next step, it is sufficient to compile the code using

```
~ top++$ make
```

The program should compile without any error massages or warnings, and is ready to use under the name `top++`.

Alternatively to modifying the Makefile it is possible to compile directly with

```
~ top++$ make CXX="user CXX value" GSLDIR="user GSLDIR value" LHADIR="user LHADIR value"
```

The quotation marks above are not necessary, unless the paths contain spaces or other special values (as usual under unix).

Appendix B. Examples

A number of examples can be found in the directory `examples`. The user can copy/paste their content into the file `top++.cfg` located in the program's main directory.

Example 1: If one executes the program with its default settings (provided in the file `top++default.cfg`) one should get the following result (this result can also be obtained with an empty file `top++.cfg` or, alternatively, one with all its options commented out):

$$\texttt{sigma\_tot} = 6.7217\ [\texttt{pb}].$$

Example 2: With the file `top++CCMMN-best-TEV.cfg` one can compute the scale variation of the cross-section corresponding to the best prediction of Ref. [1] for the Tevatron:

$$\texttt{sigma\_tot} = 6.7217 + 0.238285(3.54501\%) - 0.410247(6.10332\%)[\texttt{pb}].$$

Example 3: With the file `top++CCMMN-best-LHC7.cfg` one can compute the scale variation of the cross-section corresponding to the best prediction of Ref. [1] for the LHC @7 TeV:

$$\texttt{sigma\_tot} = 158.669 + 12.1745(7.67288\%) - 13.4631(8.48503\%)[\texttt{pb}].$$

Example 4: To derive the PDF variation for the same cross-section at the Tevatron one can use the file `top++pdfvar-TEV.cfg`:

$$\texttt{sigma\_tot} = 6.7217 + 0.160258(2.38418\%) - 0.1145(1.70344\%)[\texttt{pb}].$$

Example 5: Fixed order calculations can be done by turning off the option for resummed calculation. For example, the scale variation of the NLO result at the Tevatron, computed for consistency with an NLO PDF set (MSTW in this case), is obtained from the file `top++NLO-tev.cfg`:

$$\texttt{sigma\_tot} = 6.68123 + 0.35577(5.32492\%) - 0.752327(11.2603\%)[\texttt{pb}].$$

Example 6: The approximate NNLO result at the LHC, with excluded two-loop scale dependent terms as explained in Ref. [1] and non-zero two-loop constants $C_{qq}^{(2)}$ and $C_{gg}^{(2)}$, is obtained from the file `top++NNLO-approx-LHC7.cfg`:

$$\texttt{sigma\_tot} = 153.97 + 11.9661(7.77166\%) - 8.57241(5.56757\%)[\texttt{pb}].$$

Appendix C. Program's structure: a brief overview

The function `main()` is located in the file `top++.cpp`. The program consists of four classes that are initialized in the function `main()`. The classes, listed in the order they are initialized, and their functionality are briefly described in the following.

1. `Class PartonicFlux.` This class precomputes the partonic fluxes on a grid of `NPdfGrid` points. One object of the class is created for each value of the factorization scale μ_F. See also Section 3.1.

2. `Class FixedOrder.` This class represents the partonic fixed order cross-section. One object of this class is created for each combination of (μ_F, μ_R).
3. `Class SubtrFlux.` This class implements a fake partonic flux that mirrors the actual partonic flux within a distance $\mathcal{O}(\texttt{ETA})$ from the partonic threshold. Our implementation follows Ref. [10] where this flux was introduced and the need for it explained. Objects of class `SubtrFlux` are created for each object of class `PartonicFlux.`
4. `Class Resummation.` One object of this class is created for each pair (μ_F, μ_R) if a resummed calculation is requested by the user. Each object is constructed analytically in Mellin N-space and then inverted numerically back to x-space with the help of the Minimal Prescription of Ref. [10].

A small number of functions can be found outside the above classes. The file `lgamma` contains the logarithm of the Euler gamma function $\ln\Gamma(z)$ for complex argument, and `psin` contains the polygamma function $\Psi(z,k),\ k = 0, 1$ for complex z. The remaining functions are located in the file `Utilities`. These are the prescriptions for calculation of PDF uncertainties (we have implemented two prescriptions; see the description of option `PDFuncertainty` in Section 2) and the actual computation of the final result as a convolution of perturbative functions and partonic fluxes.

[1] M. Cacciari, M. Czakon, M. L. Mangano, A. Mitov and P. Nason, arXiv:1111.5869 [hep-ph].
[2] R. D. Ball *et al.* [The NNPDF Collaboration], arXiv:1107.2652 [hep-ph].
[3] P. M. Nadolsky and Z. Sullivan, eConfC **010630**, P510 (2001) [hep-ph/0110378].
[4] M. Cacciari, S. Frixione, M. L. Mangano, P. Nason, G. Ridolfi, JHEP **0809**, 127 (2008). [arXiv:0804.2800 [hep-ph]].
[5] A. D. Martin, W. J. Stirling, R. S. Thorne and G. Watt, Eur. Phys. J. C **63**, 189 (2009) [arXiv:0901.0002 [hep-ph]].
[6] M. Beneke, M. Czakon, P. Falgari, A. Mitov, C. Schwinn, Phys. Lett. **B690**, 483-490 (2010). [arXiv:0911.5166 [hep-ph]].
[7] R. Bonciani, S. Catani, M. L. Mangano, P. Nason, Nucl. Phys. **B529**, 424-450 (1998). [hep-ph/9801375].
[8] M. R. Whalley, D. Bourilkov and R. C. Group, arXiv:hep-ph/0508110; `http://hepforge.cedar.ac.uk/lhapdf/`.
[9] M. Aliev, H. Lacker, U. Langenfeld, S. Moch, P. Uwer and M. Wiedermann, Comput. Phys. Commun. **182**, 1034 (2011) [arXiv:1007.1327 [hep-ph]].
[10] S. Catani, M. L. Mangano, P. Nason and L. Trentadue, Nucl. Phys. B **478**, 273 (1996) [arXiv:hep-ph/9604351].