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 with exact next-to-next-to leading order (NNLO) accuracy and b) by including soft-gluon resummation for the hadronic cross-section in Mellin space with full next-to-next-to-leading logarithmic (NNLL) accuracy. The program offers the user significant flexibility through the large number (29) 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. 2.0).
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).

1. Introduction: what is this program for?

The program `Top++` calculates the total inclusive top pair production cross-section in hadronic collisions. The program can be used in both pure fixed order perturbation theory through exact next-to-next-to leading order (NNLO) [1, 2, 3, 4] and by including soft-gluon resummation through next-to-next-to-leading logarithmic order (NNLL) [5, 6] matched through NNLO. The implementation of the soft gluon resummation is as in Ref. [7]. `Top++` is the first publicly available program that can perform soft gluon resummation in top-pair production. [1]

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 5. The relevant description of the physics and the options implemented in this program can be found in Refs. [1, 2, 3, 4, 7]. The results in these papers represent the most advanced results in top physics to date.

And a word of caution regarding the numerical results derived with the help of the program. 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 recommend producing 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: [2]

```
~ 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:

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

The program 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 29 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,

---

[1] After `Top++` was released, the program *TOPIXS* [8] appeared. It performs NNLL resummed calculations in x-space. The program *HATHOR* [9] performs calculations in fixed order perturbation theory.

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

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:` Takes two values: `YES` (default) and `NO`. If set to `YES` the program will compute the observable by including soft-gluon resummation with $N^nLO + N^kLL$ accuracy. The values of n, k are set through the parameters in the subgroup **Resummation**, below. In this case all settings in the subgroup **Fixed Order** become irrelevant. If set to `NO` then no soft gluon resummation is performed, i.e. the calculation is done at fixed order. In such a case the calculation is controlled through the options in the group **Fixed Order**, and all options in the group **Resummation** become irrelevant.

   (c) `PDFuncertainty:` If set to `YES` the program will compute and display the pdf uncertainty. 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). Detailed information about computing pdf uncertainties can be found in section 3.

   (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 Refs. [1, 2, 3, 4, 7].

   (e) `PDFset:` The pdf set. The program uses the LHAPDF library [10] and follows its nomenclature. The default set is `MSTW2008nnlo68cl` [11]. Both ".LHgrid" and ".LHpdf" files can be used, see the option `PdfFileType` below.

   (f) `PDFmember:` The specific member of the pdf set `PDFset` that the user wishes to use for the calculation of the cross-section. We assume that the counting of pdf members starts from "0". 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) 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` ≥ 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` < `Mtop` then the program automatically sets `MtopLimit` = `Mtop` and issues a warning message. The default is the unattainable value `MtopLimit` = −1. The value −1 is converted internally to `MtopLimit` = `Mtop` too, but is special, because only for it the (annoying) warning message is suppressed.

(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:` A set of values for the renormalization scale (in units of the top mass `Mtop`). An arbitrary number of values is allowed. By default `muR` takes the set of values 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:` A 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 0.5 1.0 2.0 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` or `NNLO` (default). It specifies the fixed order accuracy of the resummed result as well as the order through which the resummed exponent is matched; see section 5 for more details.

   (b) `OrderRES:` Takes the values `LL, NLL` and `NNLL` (default). It specifies the logarithmic accuracy of the resummation; see section 5 for more details.

   (c) `A:` The value of the parameter A introduced in Ref. [12]. Typically used with the value `A 0` (default) or `A 2`.

   (d) `TwoLoopCoulombs:` Takes the values `YES` (default) and `NO`. It includes/excludes the two loop Coulombic terms in the function $\sigma^{(\mathrm{Coul})}$, see Eq. (3) below.

   (e) `H2qq, H2gg1, H2gg8:` The three constants in the two-loop hard function defined in a normalization of α_S/π. Implemented is the two-loop matching derived in Refs. [1, 4] (see also the description of the options below):

   $$\mathtt{H2qq} = 84.81 \ , \ \mathtt{H2gg1} = 53.17 \ , \ \mathtt{H2gg8} = 96.34 \,. \tag{1}$$

   Please note that in addition to `H2qq` only the color averaged combination of the gg-initial state matching coefficients `H2gg1, H2gg8` is presently known [1].

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`.

   (c) `NNLO:` takes values `YES` (default) or `NO`. This option implements the exact NNLO result for $t\bar{t}$ production at hadron colliders [1, 2, 3, 4].

The options in this group allow access to each individual order in perturbation theory. For example, the option `NLO` controls only the term $\sim \alpha_s^3$, i.e. for a calculation with NLO accuracy one has to set `LO YES, NLO YES` and `NNLO NO`.

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 `8000`.

   (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 introduced in Ref. [13] that controls the subtraction flux implemented in the resummed calculation. The default value `ETA 1e-5` is optimal. [3] The user normally will not need to be concerned with this parameter.

   (f) `CMP:` Within the Minimal Prescription of Ref. [13], this option corresponds to the point where the contour for the inverse Mellin transform crosses the real line. The default value `CMP 2.7` is optimal. The user normally need not be concerned with this parameter.

   (g) `PdfFileType:` Option for switching between ".LHgrid" and ".LHpdf" files in the LHAPDF interface [10]. The default value is `LHgrid`. The alternative value is `LHpdf`.

   (h) `PartonChannel:` This option allows the user to compute the contribution of a single partonic channel to the total cross-section. It can be used with both fixed order and resummed calculations. To select a particular channel one has to select one of the following six values: `qqbar`, `gg`, `qg`, `qq`, `qqprime`, `qqbarprime`. The default value is `ALL` which represents the only phenomenologically relevant case when all partonic channels are included (i.e. `ALL` is equivalent to the sum of the six partonic channels). Any other value, different from the six partonic channels is equivalent to `ALL`. Needless to say, results derived from a single partonic channel have to be interpreted with care.

3. Computation of pdf uncertainties

Starting from ver.2.0, `Top++` implements a new approach for the computation of pdf uncertainties. This approach is meant to guarantee compatibility of the program with respect to future modifications in the *LHAPDF* interface [10] or in existing families of pdf sets. The basic logic is as follows:

1. Four pdf computation prescriptions (called *Asymmetric*, *NNPDF*, *Symmetric* and *HERA_VAR*) are pre-built into the program. If needed, more prescriptions can be added by the user (see section 3.1 for details).
2. In order for the program to automatically compute pdf uncertainty for a specific pdf set, the full name of this particular pdf set has to be added to a library which associates it with the appropriate prescription for computing pdf uncertainty. Note that no default naming conventions are allowed, i.e. each pdf set has to be specified by its full name (without the extension). The program comes with a database of pairs (*pdf set*; *prescription*) contained in the file `pdf.cfg`. The default version of the file `pdf.cfg` contains a number of five-flavor pdf sets, and more sets can be freely added by the user (in no particular order). Note that if this file is modified, the program need not be recompiled.

---

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

If no known pdf prescription exists (in the file `pdf.cfg`) for the pdf set requested by the user (in the file `top++.cfg`) then the program will not compute pdf uncertainty at all and a warning message will be displayed. Still, the results for all individual pdf members are computed and displayed, i.e. if no prescription for computing pdf uncertainty with the current pdf set is known, the displayed individual results can still be used by the user for further manual processing.

The four available methods for computing pdf uncertainties are:

1. The asymmetric prescription *Asymmetric* of Ref. [14] (see also Ref. [15]). It is typically used for the MSTW [11] (and earlier sets) as well as the CTEQ family of sets [16].
2. The *NNPDF* prescription [17] for the NNPDF family of sets.
3. The symmetric prescription *Symmetric* (as defined in [18]). It is typically used for the ABM11 [18], ABKM09 [19] and earlier Alekhin [20] family of pdf sets.
4. The prescription *HERA_VAR* for computing pdf variation of the *_VAR*-type of sets from the HERA family of pdf sets (see [10]). Note that the complete pdf variation is a combination of the corresponding *_VAR* and *_EIG* sets; the pdf variation of the latter set can be computed with the *Asymmetric* prescription.

3.1. Adding new prescriptions for computing pdf uncertainties

New prescriptions might be needed for certain pdf sets, or might be developed in the future. In such cases the program in its current form will not compute the pdf uncertainty but will display the results for all individual pdf members. It will be up to the user then to derive the pdf uncertainty from these results.

To avoid the repetition of this time consuming work, the user might wish to add his/her own prescription inside `Top++` and automate the task. The addition of a new prescription for pdf uncertainty requires a rather straightforward three-step modification of the code: [4]

1. Add a function that defines the prescription itself. It is placed at the end of the file `Utilities.cpp`; the file `Utilities.h` has to be modified accordingly. As a guidance, one could use the four existing functions for computing pdf uncertainties.
2. The name of the new pdf prescription must be accounted for inside the file `top++.cpp` in the two lines following the comment *Setup for the computation of pdf uncertainties*.
3. In the file `top++.cpp`, pair the new pdf uncertainty prescription (defined in step 2) with the new function that computes it (defined in step 1), following the comment *Compute scale/pdf variation for a fixed mtop*.

4. Once the program is running: some fine tuning

4.1. 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 shorten the 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

---

[4]The authors will be happy to assist users with this task and possibly incorporate new pdf prescriptions that might be of interest to a broader audience into future official versions of the program.

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.

4.2. Outputting the results

The program outputs on the screen all results as well as information about the timing of each step. 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 (when prescription for pdf uncertainty is known or when central value can be computed). 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. Please note that the file `top++.res` is overwritten after each run.

4.3. Modifying the default values of the parameters

The default values for all 29 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.

4.4. Parallelizing calculations

Practice shows that oftentimes one has to perform a large number of calculations with different parameters. Given that the run times vary significantly, it is natural *and very beneficial* to parallelize these calculations. To achieve such parallelization the user basically needs to have in a common directory only the executable `top++` (common for all calculations), together with the two configurations files `pdf.cfg` (also common for all calculations) and `top++.cfg` (specific to each calculation).

4.5. Additional

The strong coupling constant is calculated at a scale μ_R through the LHAPDF interface [10]. 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}.

5. Contact with physics

The program can be used to compute the $t\bar{t}$ total inclusive cross-section σ_{tot} either in a pure fixed order perturbation theory through exact NNLO [1, 2, 3, 4] or including soft gluon resummation performed in Mellin space through NNLL [5, 6] and matched through NNLO:

$$\sigma_{\text{tot}}^{(n,k)} = \sigma_{\text{F.O.}}^{(n)} + \left[\sigma_{\text{res}}^{(n,k)} - \sigma_{\text{res}}^{(n,k)}|_{\alpha_S^n} \right] , \qquad (2)$$

where the labels n (k) implicitly denote the fixed (logarithmic) order accuracy of the result.

Pure fixed order calculation can be achieved by setting the option `WithResummation NO`. In this case the terms in the square bracket in Eq. (2) are absent and n is controlled through the appropriate combination of the options `LO, NLO` and `NNLO` described in section 2.4. We remind the reader that these three options control separately the corrections at orders $O(\alpha_S^n)$, $n = 2, 3, 4$ in the fixed order result. This way the user has individual access to each one of the three known orders in the perturbative expansion of $\sigma_{\text{F.O.}}$. For example, for calculations with NNLO accuracy, one has to set all three options `LO, NLO, NNLO` to `YES`.

To perform soft-gluon resummation one has to include the terms in the square bracket in Eq. (2) by setting the option `WithResummation YES`. The power n has the same meaning as in the case of pure fixed order calculation, but is now controlled through the option `OrderFO`. This option takes the value `LO` (or `NLO`, or `NNLO`) which, unlike the case of pure fixed order calculations described above, includes all terms through order $O(\alpha_S^n)$, $n = 2$ (or 3, or 4).

In Mellin N-space, the resummed partonic cross-section reads (see Ref. [7] for complete description)

$$\sigma_{\text{res part};N,\mathbf{I}}^{(n,k)} = \sum_{\mathbf{I}=1,8} \sigma_{N,\mathbf{I}}^{(\text{Coul}),(n)} \times \sigma_{N,\mathbf{I}}^{(\text{Hard}),(n)} \times \Delta_{N,\mathbf{I}}^{(k)}. \tag{3}$$

The function $\sigma^{(\text{Coul}),(n)}$ in Eq. (3) contains the Coulombic effects and has a known perturbative expansion through NNLO. The depth of the perturbative expansion of this function is set by the option `OrderFO`. The user can also turn on or off the NNLO correction to this function through the option `TwoLoopCoulombs` (normally it should not be modified). The expression for the function $\sigma^{(\text{Coul})}$ can be found in Ref. [7].

The function $\sigma^{(\text{Hard}),(n)}$ in Eq. (3) is an N-independent function. The depth of its perturbative expansion, in an expansion in α_S/π, is set by the option `OrderFO`. The one-loop corrections are known exactly [21]. The two-loop corrections `H2gg1, H2gg8, H2qq` are accessible through the options `H2gg1, H2gg8, H2qq`: the $q\bar{q}$ reaction term `H2qq` is known with sufficiently high accuracy [4], while only the color averaged combination of `H2gg1, H2gg8` is currently known [1]. The default values for `H2gg1, H2gg8` from Ref. [1] are given in Eq. (1).

The function $\Delta^{(k)}$ in Eq. (3), the so-called Sudakov exponent, contains the towers of LL, NLL and NNLL soft $\ln(N)$ logs. The user can request LL, NLL or NNLL logarithmic accuracy by setting the option `OrderRES` to `LL, NLL`, or `NNLL` respectively. For example, for resummation with full NNLL accuracy one sets `OrderRES NNLL`.

As follows from the description of the functions $\sigma^{(\text{Coul})}$ and $\sigma^{(\text{Hard})}$ above, for a given fixed order accuracy `OrderFO` the resummed cross-section $\sigma_{\text{res}}^{(n,k)}$ is automatically matched through the same order `OrderFO` and, to avoid double counting, its perturbative expansion $\sigma_{\text{res}}^{(n,k)}|_{\alpha_S^n}$ is subtracted also through order `OrderFO`. [5]

No options related to approximate NNLO calculations are available in the program, since they are now obsolete and superseded by the exact NNLO results. If the user would like to use them anyway, then he/she will have to use the appropriate earlier version of `Top++`.

6. Summary

We present the C++ program `Top++` for the calculation of the total inclusive 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 incorporates all currently available theoretical results: fixed order calculations through NNLO [1, 2, 3, 4] and soft gluon resummation through NNLL [5, 6]. The user has access to 29 options which results in a great deal of flexibility and control over the calculation. 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. [7] as well as Refs. [1, 2, 3, 4].

---

[5]We note that starting from ver. 2.0, the matching of the resummed cross-section σ_{res} is internally controlled through the option `OrderFO` and not through the option `OrderRES` as was the case in the earlier versions of `Top++`. The new setting corresponds to the standard matching conventions and is the natural choice when working with the exact NNLO result (unlike working with approximate NNLO as was the case in earlier versions). Of course, for typical applications like LO+LL, NLO+NLL and NNLO+NNLL, the two implementations are equivalent.

The program is organized in a modular, object oriented way. It 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.

Acknowledgments

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

The paths to the directories LHADIR and GSLDIR are determined automatically. The paths can also be set explicitly, the most standard case being `GSLDIR = /usr/local` and `LHADIR = /usr/local`.

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 (updated for ver.2.0)

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.

- The file `top++complete-set.cfg` contains all options offered by the program. Most of these options are not needed in typical phenomenological studies and therefore will rarely be used.

- The file `top++pheno.cfg` contains all parameters needed in typical phenomenological studies.

- The file `top++best-precision.cfg` is a subset of `top++pheno.cfg` and contains a set of options needed for typical applications, i.e. calculations with the "best" available NNLO+NNLL precision.

In the following, we give few examples based on the file `top++pheno.cfg`.

Example 1: If one executes the program with its default settings provided in the file `top++.cfg` (which is equivalent to the file `top++pheno.cfg`) one should get the best prediction of Ref. [1] for the LHC at 8 TeV (this result can also be obtained with an empty file `top++.cfg` or, alternatively, one with all its options commented out):

$$\texttt{sigma\_tot} = 245.794 + 6.23992(2.53868\%) - 8.41868(3.4251\%) \ [\texttt{pb}].$$

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

$$\texttt{sigma\_tot} = 7.1642 + 0.109671(1.53082\%) - 0.199974(2.79129\%) \ [\texttt{pb}].$$

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

$$\texttt{sigma\_tot} = 172.025 + 4.7067(2.73606\%) - 4.79784(2.78904\%) \ [\texttt{pb}].$$

Example 4: To derive the scale variation variation for the NNLO fixed order prediction for the LHC at 14 TeV one can use the file `top++NNLO-LHC14.cfg`:

$$\texttt{sigma\_tot} = 932.959 + 31.7776(3.4061\%) - 50.9738(5.46367\%) \ [\texttt{pb}].$$

Next we give few examples of less common calculations.

Example 5: To derive the contributions from the term $\sim \alpha_s^3$ (i.e. not including the contributions at $\sim \alpha_s^2$ and $\sim \alpha_s^4$ and no soft-gluon resummation) for central scale values $\mu_F = \mu_R = m_{\text{top}}$ at the LHC at 7 TeV and with NNLO pdf, we use the file `top++as3-LHC7.cfg`:

$$\texttt{sigma\_tot} = 47.6818 \ [\texttt{pb}].$$

Example 6: To compute the cross-section with NNLO+LL precision at the LHC at 8 TeV, we use the file `top++NNLO-LL-LHC8.cfg`:

$$\texttt{sigma\_tot} = 244.283 + 8.18593(3.351\%) - 14.4041(5.89649\%) \ [\texttt{pb}].$$

Example 7: To compute the cross-section at NLO, with NLO pdf, for three different values of the top mass (chosen here to be close to its current best value and uncertainty) at the Tevatron, we utilize the looping option over the top mass (see the file `top++mass-loop-TEV.cfg`):

$$\text{m}_{\text{top}} = 172.3 : \ \texttt{sigma\_tot} = 6.89431 + 0.367937(5.33682\%) - 0.776566(11.2639\%) \ [\texttt{pb}] \,,$$
$$\text{m}_{\text{top}} = 173.3 : \ \texttt{sigma\_tot} = 6.68166 + 0.356095(5.32945\%) - 0.752499(11.2622\%) \ [\texttt{pb}] \,,$$
$$\text{m}_{\text{top}} = 174.3 : \ \texttt{sigma\_tot} = 6.47632 + 0.344724(5.32283\%) - 0.72929(11.2609\%) \ [\texttt{pb}] \,.$$

Appendix C. What is new in ver.2.0: changes from ver.1.4

- Added the exact NNLO result for the reaction $gg \to t\bar{t} + X$ [1]. With this addition the program now contains the full set of NNLO QCD corrections to top pair production at hadron colliders.

- Improved formatting of the output file `top++.res`.

- New approach to computing pdf uncertainties (see section 3 for details).

- Modified the evaluation of pdf uncertainties to accommodate pdf sets with different α_S for each pdf member.

- Removed the options `NNLOonORoff`, `RESonORoff` and `Cbargg` related to approximate NNLO [22] approaches.

- Replaced the NDE [23] parameterization of the NLO partonic cross-section with a fit [9] to the exact NLO result [24].

- New default value for the option `A` (was 2, now is 0).

- New default values for the options `H2gg1` and `H2gg8` (both were 0; for current values see Eq. (1)).

- Matching of the resummed result is set by the option `OrderFO` (was set by the option `OrderRES`); see also section 5. The two are equivalent in the standard case when equal fixed order and logarithmic precision is required, i.e. LO+LL, NLO+NLL and NNLO+NNLL.

- Added the prescription *HERA_VAR* for computing pdf variation of the *_VAR*-type of sets from the HERA family of pdf sets.

- Added new option `PartonChannel` which allows the user to select the contribution from a single partonic channel.

- Rewritten manual to reflect the above modifications.

Appendix D. 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 4.1.
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. [13] 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. [13].

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 four 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. Czakon, P. Fiedler and A. Mitov, arXiv:1303.6254 [hep-ph].

[2] M. Czakon and A. Mitov, JHEP **1301**, 080 (2013) [arXiv:1210.6832 [hep-ph]].

[3] M. Czakon and A. Mitov, JHEP **1212**, 054 (2012) [arXiv:1207.0236 [hep-ph]].

[4] P. Baernreuther, M. Czakon and A. Mitov, Phys. Rev. Lett. **109**, 132001 (2012) [arXiv:1204.5201 [hep-ph]].

[5] M. Beneke, P. Falgari and C. Schwinn, Nucl. Phys. B **828**, 69 (2010) [arXiv:0907.1443 [hep-ph]].

[6] M. Czakon, A. Mitov and G. F. Sterman, Phys. Rev. D **80**, 074017 (2009) [arXiv:0907.1790 [hep-ph]].

[7] M. Cacciari, M. Czakon, M. L. Mangano, A. Mitov and P. Nason, Phys. Lett. B **710**, 612 (2012) [arXiv:1111.5869 [hep-ph]].

[8] M. Beneke, P. Falgari, S. Klein, J. Piclum, C. Schwinn, M. Ubiali and F. Yan, JHEP **1207**, 194 (2012) [arXiv:1206.2454 [hep-ph]].

[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] M. R. Whalley, D. Bourilkov and R. C. Group, arXiv:hep-ph/0508110; `http://hepforge.cedar.ac.uk/lhapdf/`.

[11] A. D. Martin, W. J. Stirling, R. S. Thorne and G. Watt, Eur. Phys. J. C **63**, 189 (2009) [arXiv:0901.0002 [hep-ph]].

[12] R. Bonciani, S. Catani, M. L. Mangano, P. Nason, Nucl. Phys. **B529**, 424-450 (1998). [hep-ph/9801375].

[13] S. Catani, M. L. Mangano, P. Nason and L. Trentadue, Nucl. Phys. B **478**, 273 (1996) [arXiv:hep-ph/9604351].

[14] P. M. Nadolsky and Z. Sullivan, eConfC **010630**, P510 (2001) [hep-ph/0110378].

[15] M. Cacciari, S. Frixione, M. L. Mangano, P. Nason, G. Ridolfi, JHEP **0809**, 127 (2008). [arXiv:0804.2800 [hep-ph]].

[16] H. -L. Lai, M. Guzzi, J. Huston, Z. Li, P. M. Nadolsky, J. Pumplin and C. -P. Yuan, Phys. Rev. D **82**, 074024 (2010) [arXiv:1007.2241 [hep-ph]].

[17] R. D. Ball *et al.* [The NNPDF Collaboration], Nucl. Phys. B **855**, 153 (2012) [arXiv:1107.2652 [hep-ph]].

[18] S. Alekhin, J. Blumlein and S. Moch, Phys. Rev. D **86**, 054009 (2012) [arXiv:1202.2281 [hep-ph]].

[19] S. Alekhin, J. Blumlein, S. Klein and S. Moch, Phys. Rev. D **81**, 014032 (2010) [arXiv:0908.2766 [hep-ph]].

[20] S. Alekhin, Phys. Rev. D **68**, 014002 (2003) [hep-ph/0211096].

[21] M. Czakon and A. Mitov, Phys. Lett. B **680**, 154 (2009) [arXiv:0812.0353 [hep-ph]].

[22] M. Beneke, M. Czakon, P. Falgari, A. Mitov, C. Schwinn, Phys. Lett. **B690**, 483-490 (2010). [arXiv:0911.5166 [hep-ph]].

[23] P. Nason, S. Dawson and R. K. Ellis, Nucl. Phys. B **303**, 607 (1988).

[24] M. Czakon and A. Mitov, Nucl. Phys. B **824**, 111 (2010) [arXiv:0811.4119 [hep-ph]].