DESY 10-091
HU-EP-10-33
SFB/CPP-10-60

arXiv:1007.1327v1 [hep-ph] 8 Jul 2010

– HATHOR –

HAdronic Top and Heavy quarks crOss section calculatoR

M. Aliev[a], H. Lacker[a], U. Langenfeld[a], S. Moch[b],

P. Uwer[a] and M. Wiedermann[a]

[a] *Humboldt-Universität zu Berlin, Institut für Physik*
Newtonstraße 15, D-12489 Berlin, Germany

[b] *Deutsches Elektronensynchrotron DESY*
Platanenallee 6, D–15735 Zeuthen, Germany

Abstract

We present a program to calculate the total cross section for top-quark pair production in hadronic collisions. The program takes into account recent theoretical developments such as approximate next-to-next-to-leading order perturbative QCD corrections and it allows for studies of the theoretical uncertainty by separate variations of the factorization and renormalization scales. In addition it offers the possibility to obtain the cross section as a function of the running top-quark mass. The program can also be applied to a hypothetical fourth quark family provided the QCD couplings are standard.

Program summary

Title of program:	Hathor
Version:	1.0
Catalogue number:	
Program summary URL:	`http://www.physik.hu-berlin.de/pep/tools` `http://www-zeuthen.desy.de/~moch/hathor`
E-mail:	`sven-olaf.moch@desy.de, peter.uwer@physik.hu-berlin.de`
License:	GNU Public License
Computers:	Standard PCs (x86, x86_64 processors)
Operating system:	Linux
Program language:	C++, fortran, Java
Memory required to execute:	256 MB
Other programs called:	None.
External files needed:	Interface to LHAPDF for the user's choice of parton distribution functions, see `http://projects.hepforge.org/lhapdf/`.
Keywords:	Top-quarks, total cross section, QCD, radiative corrections, running mass.
Nature of the physical problem:	Computation of total cross section in perturbative QCD.
Method of solution:	Numerical integration of hard parton cross section convoluted with parton distribution functions.
Restrictions on complexity of the problem:	None
Typical running time:	A few seconds to a few minutes on standard desktop PCs or notebooks, depending on the chosen options.

1 Introduction

The top-quark is the heaviest elementary particle in nature discovered so far. As a consequence of its large mass close to the scale of the electroweak symmetry breaking it has remarkable properties making it a distinct research object. For example the short lifetime does not allow the formation of hadronic bound-states. Rather, the top-quark decays before it hadronizes, a fact often referred to colloquially as the top-quark behaving like a quasi-free quark. Non-perturbative effects are thus essentially cut-off by the short lifetime and, as an important consequence, the polarization of top-quarks can be studied through the parity violating decay into a W-boson and a bottom quark. The top-quark also provides an interesting environment for precision tests of the Standard Model (SM) and possible extensions, e.g., by constraining the allowed range for the Higgs mass.

A mandatory ingredient for top-quark physics at hadron colliders are precise theoretical predictions to compare with. Current Tevatron measurements and the perspectives at LHC, i.e., a measurement of the top-quark pair cross section with an uncertainty of the order of 5% only set the target for theoretical predictions of the production process. Clearly such an accuracy needs to include quantum corrections.

Within Quantum Chromodynamics (QCD) radiative corrections were calculated some time ago to next-to-leading order (NLO) first considering unpolarized top-quark production [1, 2] and later including spin information [3]. In the former case, also completely analytical results have recently been provided [4]. Beyond the NLO accuracy in QCD various sources of possible improvements have been identified. Large logarithmic corrections due to soft gluon emission were investigated and resummed at the next-to-leading-logarithmic (NLL) accuracy [5,6] and recently improved to include also the next-to-next-to-leading-logarithmic (NNLL) corrections [7–9]. Alternatively, resummation has also provided the means to construct parts of the full next-to-next-to-leading (NNLO) fixed order results [7, 10, 11], which can be supplemented by including all Coulomb type corrections [11] and also the full scale dependence at NNLO accuracy [7, 12]. With a target precision for the total cross section at the few per cent level also bound state effects from the resummation of Coulomb type corrections [13, 14] as well as electro-weak radiative corrections at NLO [15–17] need to be considered.

The compilation of all these results is in principle straight forward given the extensive literature on the subject. However, no publicly available program exists so far which contains the latest theoretical developments. The "modus operandi" of the past was that predictions were updated by theorists from time to time taking into account new theoretical improvements and/or new sets of parton distribution functions (PDFs). The aim of the present paper is to provide a program for the computation of the top-quark pair cross section including state of the art theory. As such it can serve as a reference for future cross section calculations. The program Hathor includes perturbative QCD corrections at higher orders in the different approximations along with options allowing also a detailed study of the theoretical uncertainties. Moreover, it provides the possibility to compute the total cross section not only in the commonly adapted pole mass scheme but also in terms of the $\overline{\text{MS}}$ mass a choice recently employed for top-quark pair production in hadronic collisions for the first time [12, 18]. Finally, the aim of this publication is not only to provide a tool for cross section calculations but also to facilitate experimental analyses. To that

end the package contains in addition to the stand alone program also a small library that can be easily integrated into existing code.
The outline of the article is a follows. In the next Section we briefly discuss the theoretical foundations as well as the procedure to convert to the $\overline{\rm MS}$ mass. Sections 3 and 4 contain installation details and the program description while usage and examples are given in Section 5. We end with conclusions in Section 6. All formulae as implemented in Hathor are collected in the Appendices A and B.

2 Methods

The hadronic cross section for top-quark pair production is obtained from the convolution of the factorized partonic cross section $\hat{\sigma}_{ij}$ with the parton luminosities $\mathcal{L}_{ij}$:

$$\sigma_{h_1h_2\to t\bar{t}X}(S,m_t) = \sum_{i,j} \int_{4m_t^2}^{S} ds\, \mathcal{L}_{ij}(s,S,\mu_f)\, \hat{\sigma}_{ij}(s,m_t,\alpha_s(\mu_r),\mu_f)\,, \tag{1}$$

$$\mathcal{L}_{ij}(s,S,\mu_f) = \frac{1}{S}\int_s^S \frac{d\hat{s}}{\hat{s}} f_{i/h_1}\left(\frac{\hat{s}}{S},\mu_f\right) f_{j/h_2}\left(\frac{s}{\hat{s}},\mu_f\right). \tag{2}$$

Here S denotes the hadronic center-of-mass energy squared, μ_r, (μ_f) denotes the renormalization (factorization) scale and the functions $f_{i/h_{1,2}}(x,\mu_f)$ are the PDFs describing the probability to find a parton of type i with a momentum fraction between x and $x+dx$ in the hadron h_k. The QCD coupling constant $\alpha_s(\mu_r)$ is evaluated at the scale μ_r. In the following we use α_s in the scheme with n_f light flavors. For top-quark production, the running is thus determined by the five light flavors u,d,c,s,b which we treat as massless. The top-quark mass m_t appearing in Eq. (1) is the mass renormalized in the on-shell (pole-mass) scheme.
In perturbative QCD the partonic cross section $\hat{\sigma}_{ij}(s,m_t,\alpha_s,\mu_f)$ is expanded in the QCD coupling constant up to NNLO:

$$\hat{\sigma}_{ij} = a_s^2\,\hat{\sigma}_{ij}^{(0)}(s,m_t) + a_s^3\,\hat{\sigma}_{ij}^{(1)}(s,m_t,\mu_r,\mu_f) + a_s^4\,\hat{\sigma}_{ij}^{(2)}(s,m_t,\mu_r,\mu_f) + O(a_s^5)\,, \tag{3}$$

with $a_s = \alpha_s/\pi$.
In leading-order (LO) only the parton channels $q\bar{q}$ and gg contribute and the respective Born cross sections are given by:

$$\hat{\sigma}_{qq}^{(0)} = \frac{4\pi^3}{27}\frac{1}{s}\beta(3-\beta^2), \tag{4}$$

$$\hat{\sigma}_{gg}^{(0)} = \frac{\pi^3}{48}\frac{1}{s}\left\{(33-18\beta^2+\beta^4)\ln\left(\frac{1+\beta}{1-\beta}\right) - 59\beta + 31\beta^3\right\}, \tag{5}$$

with $\beta = \sqrt{1-\rho}$ and $\rho = 4m_t^2/s$. Starting from NLO also the gq and $g\bar{q}$ channels contribute. In Ref. [1] (and in many subsequent publications) an alternative decomposition was used in terms

of so-called scaling functions f_{ij}:

$$\hat{\sigma}_{ij} = \frac{{\alpha_s}^2}{m_t^2}\left\{f_{ij}^{(0)}(\rho) + 4\pi\alpha_s f_{ij}^{(1)}(\rho,\mu_f/m_t,\mu_r/\mu_f) + (4\pi\alpha_s)^2 f_{ij}^{(2)}(\rho,\mu_f/m_t,\mu_r/\mu_f) + O({\alpha_s}^3)\right\}. \tag{6}$$

Since the scaling functions are dimensionless they depend only on ρ and the ratios μ_f/m_t and μ_r/μ_f. The full renormalization and factorization scheme dependence can be constructed using the renormalization group equation, the standard evolution equations of the PDFs and information about lower orders, i.e., up to NNLO knowledge of $f_{ij}^{(0)}(\rho)$ and $f_{ij}^{(1)}(\rho,1,1)$ is sufficient. The general structure can be written in the following form

$$\begin{aligned} f_{ij}^{(1)}(\rho,\mu_f/m_t,\mu_r/m_t) &= f_{ij}^{(10)} + L_M f_{ij}^{(11)} + 2\mathbb{0} L_R f_{ij}^{(0)}, & (7)\\ f_{ij}^{(2)}(\rho,\mu_f/m_t,\mu_r/m_t) &= f_{ij}^{(20)} + L_M f_{ij}^{(21)} + L_M^2 f_{ij}^{(22)} + 3\mathbb{0} L_R f_{ij}^{(10)} + 3\mathbb{0} L_R L_M f_{ij}^{(11)} \\ &\quad + 2\mathbb{1} L_R f_{ij}^{(0)} + 3\mathbb{0}^2 L_R^2 f_{ij}^{(0)}, & (8) \end{aligned}$$

with $ij = \{q\bar{q}, gg\}$ and we abbreviate $L_M = \ln(\mu_f^2/m_t^2)$ and $L_R = \ln(\mu_r^2/\mu_f^2)$. The scale dependence in the gq ($g\bar{q}$) channel can be easily derived from the above realizing that $f_{gq}^{(0)} = 0$ so some terms in Eqs. (7) and (8) are simplify absent. In the conventions used here the coefficients of the beta-function $\mathbb{0}, \mathbb{1}$ are given by

$$\mathbb{0} = \frac{1}{(4\pi)^2}\left(11 - \frac{2}{3}n_f\right), \qquad \mathbb{1} = \frac{1}{(4\pi)^4}\left(102 - \frac{38}{3}n_f\right). \tag{9}$$

The Born contributions have been presented in Eqs. (4), (5) and at present also the complete NLO corrections are known, i.e., the functions $f_{ij}^{(10)}$ and $f_{ij}^{(11)}$ in Eq. (7). A complete NNLO calculation for the total cross section is not yet available, since $f_{ij}^{(2)}(\rho,\mu_f/m_t,\mu_r/m_t)$ in Eq. (8) is missing the contribution $f_{ij}^{(20)}$ while $f_{ij}^{(21)}$ and $f_{ij}^{(22)}$ have been obtained from renormalization group arguments as mentioned above. However, exact expressions for $f_{ij}^{(20)}$ in the limit $\rho \to 1$ based on soft-gluon resummation have been derived and provide the foundation for approximate NNLO results of $\sigma_{h_1h_2\to t\bar{t}X}$.
The central physics questions to be addressed can be phrased as follows:

- How large is the total cross section $\sigma_{h_1h_2\to t\bar{t}X}$ at a given order in perturbation theory ?
- Given a computation of the total cross section according to Eq. (1) what is the associated theoretical uncertainty ?

In order to address these issues the package Hathor has different production models implemented which are accessible to the user as options. In the following we briefly describe these options as far as the underlying physics is concerned. To be self-consistent and for easier reference all necessary theory input, e.g., the scaling functions has been collected in Appendix B. For details of how to access these options when running Hathor we refer to the next Sections 4 and 5.

Option `LO`

The option `LO` provides a rough estimate although with large theoretical uncertainties which will receive sizable corrections at higher orders. This option uses the Born cross sections of Eqs. (4), (5) (see also Eqs. (B.1)–(B.3)).

Option `NLO`

The option `NLO` is the first instance where a meaningful theoretical uncertainty can be quoted in perturbation theory. This option employs the complete NLO QCD corrections [1, 2]. All scaling functions $f_{ij}^{(10)}$ are given as accurate fits [12] based on the recently published analytic results [4], (see also Eqs. (B.4)–(B.6)).

Option `NNLO`

The option `NNLO` is required whenever predictions with an uncertainty of better than $O(10)\%$ are needed. This option is based on the known threshold enhancement due to soft gluon emission, i.e., complete tower of Sudakov logarithms at NNLO accuracy, supplemented by all Coulomb type corrections [11] and also the full scale dependence [7, 12] (see Eqs. (B.12)–(B.17)). This ansatz provides a good approximation for the total cross section [7,10], a fact which is supported by the observation that the QCD corrections to top-quark pair production in association with an additional jet are small [19–21].
Thus, using the results of Refs. [11, 12], we have for the functions $f_{ij}^{(20)}$:

$$\begin{aligned} f_{q\bar{q}}^{(20)} &= \frac{f_{q\bar{q}}^{(0)}}{(16\pi^2)^2}\Big\{\frac{8192}{9}\ln^4\beta + (-1505.1589 + 37.925926 n_f)\ln^3\beta \\ &\quad + \Big(1046.4831 - 90.838135 n_f - 140.36771\frac{1}{\beta}\Big)\ln^2\beta \\ &\quad + \Big(249.67547 + 55.776275 n_f + (54.038454 - 4.3864908 n_f)\frac{1}{\beta}\Big)\ln\beta \\ &\quad + 3.6077441\frac{1}{\beta^2} + (-5.2728242 + 1.8447758 n_f)\frac{1}{\beta} + C_{q\bar{q}}^{(2)}\Big\}, \end{aligned} \tag{10}$$

$$f_{gq}^{(20)} = \frac{\beta^3}{(16\pi^2)^2}\frac{65\pi}{54}\ln^3(8\beta^2), \tag{11}$$

$$\begin{aligned} f_{gg}^{(20)} &= \frac{f_{gg}^{(0)}}{(16\pi^2)^2}\Big\{4608\ln^4\beta + (-2321.5810 + 85.333333 n_f)\ln^3\beta \\ &\quad + \Big(-315.57218 - 119.35529 n_f + 496.30011\frac{1}{\beta}\Big)\ln^2\beta \\ &\quad + \Big(2346.8889 + 21.969529 n_f + (286.67132 + 6.8930570 n_f)\frac{1}{\beta}\Big)\ln\beta \end{aligned} \tag{12}$$

$$+68.547138\frac{1}{\beta^2}+(-3.7910584-0.96631115n_f)\frac{1}{\beta}+C_{gg}^{(2)}\Big\}\,,$$

where the unknown functions $C_{q\bar{q}}^{(2)}$ and $C_{gg}^{(2)}$ in Eqs. (10) and (12) parametrize the contributions which are not enhanced in the threshold region, i.e., $O(\beta^0)$. The gq-channel, that is $f_{gq}^{(20)}$ in Eq. (11), is additionally suppressed near threshold with corrections of order $O(\beta^3\ln^2(\beta))$.
In summary, the option `NNLO` (which has been used e.g., for the phenomenological studies of Ref. [12]) uses all presently available information at NNLO. In this way, it attempts to construct the relevant parts of the complete NNLO corrections. Necessarily, the small associated theoretical uncertainty [7, 10] due to scale variation (μ_r and μ_f) estimates effects beyond NNLO. An additional systematical uncertainty on the quality of the approximate NNLO result can be quantified by varying the constants $C_{q\bar{q}}^{(2)}$ and $C_{gg}^{(2)}$ in a reasonable range comparable to the size of the other coefficients in Eqs. (10) and (12), i.e., $C_{ij}^{(2)}=\pm O(100)$. The default value is $C_{ij}^{(2)}=0$.

Option `LOG_ONLY`

The option `LOG_ONLY` is also motivated by the idea of soft gluon enhancement near threshold. It emerged as a conservative definition of the theoretical uncertainty in a comparison of different approaches to incorporate dominant terms beyond NLO to a certain logarithmic accuracy. In particular threshold resummation at NLL accuracy, performed as in Refs. [6, 22] which typically proceeds in Mellin-space, see Eq. (13), has been tested against an expansion in powers of $\ln^k(\beta)$ in momentum space as advocated in Ref. [7, 10]. This comparison has yielded satisfactory agreement, because resummation beyond NLL, i.e., at NNLL accuracy has only a minor effect [7].
The option `LOG_ONLY` as discussed below is based on work with CCMMMNU [23]. It is a genuine NLO approach with logarithmic improvement near threshold and scale variations in μ_r and μ_f (with a constraint on the ratio of μ_r/μ_f) estimate effects beyond NLO. Being a conservative approach the resulting theoretical uncertainty is necessarily larger than in the option `NNLO`.
Let us briefly mention the essential technical points. In Refs. [6, 22] the logarithmic enhancement is constructed from the $\ln(N)$ terms in Mellin space where the resummation is usually performed. The transformation from momentum ($\rho-$) space to Mellin (N-) space is given by

$$\sigma(N)=\int_0^1 d\rho\,\rho^{N-1}\sigma(\rho)\,, \tag{13}$$

where $\rho=4m_t^2/s$. The important feature of Eq. (13) to realize is that beyond logarithmic accuracy the momentum space and the Mellin-space expressions do differ by terms which are not enhanced in β or, respectively power-suppressed in $1/N$. Any difference could be included in a choice of $C_{ij}^{(2)}$ in Eqs. (10), (12).
Option `LOG_ONLY` has to be used with the Option `NNLO`. It applies Eqs. (10) and (12), but truncates the function $f_{ij}^{(0)}$ beyond NLO to is leading term in β (cf. Eqs. (4), (5)) and, for consistency neglects the gq-channel beyond NLO ($f_{gq}^{(20)}$ and Eqs. (B.14) and (B.15)). Also for the scale dependent part, the option `LOG_ONLY` keeps only terms which are logarithmically enhanced in the

threshold region. In this case the functions $f_{ij}^{(21)}$ and $f_{ij}^{(22)}$ in Eqs. (B.12)–(B.17)) are truncated to logarithmic accuracy. Again, one could also vary the constants $C_{q\bar{q}}^{(2)}$ and $C_{gg}^{(2)}$ in a range $C_{ij}^{(2)} = \pm O(100)$ to test for additional systematical uncertainties. The default value is $C_{ij}^{(2)} = 0$.

Option `MS_MASS`

The option `MS_MASS` allows for the computation of the total cross sections as a functions of the running mass in the $\overline{\rm MS}$ scheme. In a nut-shell this is based on the replacement $m_t \to \overline{m}(\mu_r)$ (see Eq. (A.1)) in the expression for $\sigma_{h_1h_2\to t\bar{t}X}$ in Eq. (1). The option `MS_MASS` can be applied together with options `LO`, `NLO` and `NNLO`.
Let us briefly discuss the main motivation for this option. So far the mass used in all formulae is given as the so-called on-shell or pole-mass which is defined as the location of pole of the quark propagator calculated order-by-order in perturbation theory. It is well known that the pole-mass is not a well defined concept in QCD since quarks do not appear as asymptotic states in the quantum field theoretical description of the strong interaction owing to confinement. In other words, the quark propagator does not have a pole in full QCD. A more quantitative analysis leads to the conclusion that the pole-mass is intrinsically uncertain of the order of Λ_{QCD} (see e.g. [24, 25]). Since in perturbation theory the pole-mass can be expressed in terms of a short distance mass like for example the running mass which is free from non-perturbative effects it is advantageous to translate the cross section predictions from the on-shell scheme to the $\overline{\rm MS}$ mass scheme. As a benefit, the convergence of the perturbative series is significantly improved when the running mass is used and the extracted numerical value of the top-quark mass is very stable under higher order corrections. These observations were first pointed out in Ref. [12].
In the Hathor program the conversion $\sigma_{h_1h_2\to t\bar{t}X}(S, m_t) \to \sigma_{h_1h_2\to t\bar{t}X}(S, \overline{m}(\mu_r))$ has been realized now in an easy way allowing a direct evaluation of the cross section using the running mass (see also [18]). All details are deferred to Appendix A.

Option `PDF_SCAN`

The option `PDF_SCAN` allows for the automated computation of PDF uncertainties. In the default setting of the Hathor package the PDFs can be accessed with the LHAPDF library [26, 27]. A prerequisite for this option is, of course, that the respective PDF provides a set of error PDFs. There are, however, different conventions with respect to PDF uncertainties.
For instance, there exists the convention of asymmetric uncertainties, a choice adopted by e.g. MSTW [28] and CTEQ [29]. Here the error PDFs come in n_{PDF} pairs $(\sigma_{k,+}, \sigma_{k,-})$, where the first element of the pair describes the error of the corresponding parameter in the '$+$'-direction, the second the one in the '$-$'-direction. Then, for a given PDF set with a central fit resulting in a cross section σ_0 the systematic uncertainty $\Delta\sigma_\pm$ is estimated by (see e.g. [30]),

$$\Delta\sigma_\pm = \frac{1}{2}\sqrt{\sum_{k=1,n_{PDF}} (\max(0, \pm\sigma_{k,+} \mp \sigma_0, \pm\sigma_{k,-} \mp \sigma_0))^2}\,. \tag{14}$$

Eq. (14) is the default of the Hathor package when using the option `PDF_SCAN`. Following standard conventions the PDF uncertainty should be linearly added to the theoretical uncertainty from scale variations (parameterizing uncalculated higher orders).
Other PDF sets, e.g. ABKM [31], employ the convention of symmetric uncertainties, where the n_{PDF} elements each describe the (symmetric) '$\pm$'-variation. In this case, the quadratic uncertainty $\Delta\sigma_{\pm}$ is obtained by adding the individual errors quadratically in the standard manner,

$$\Delta\sigma_{\pm} = \sqrt{\sum_{k=1,n_{PDF}} (\sigma_k - \sigma_0)^2}\,, \tag{15}$$

and the option `PDF_SCAN` has to be combined with the additional option `PDF_SYM_ERR`.
Finally, there exist PDF sets, e.g. [32, 33] which simply return a number n_{PDF} of best fits, where typically $n_{PDF} \simeq O(100)$. Then, the PDF uncertainty of the cross section σ is estimated by computing it with each of the best fits and taking the standard statistical average. In such cases, the option `PDF_SCAN` cannot be used for an automated computation of the PDF uncertainty. Within Hathor, it of course, always possible for the user to provide own code for the evaluation of the PDF uncertainy.
If, however, a PDF set provides different values of the strong coupling α_s for different error PDFs, this is automatically taken into account.

Before continuing with the description of the Hathor program, let us mention that the package offers the possibility for several extensions in the future. With an experimental precision of 5% as envisaged at LHC the electro-weak radiative corrections at NLO [15–17] have to be taken into account. While for the Tevatron they turn out to be small (less than 1%) they are of the order of 2% at LHC with a slight dependence on the Higgs mass. Electro-weak NLO corrections can be included in a similar manner as the higher order QCD corrections, i.e., with the help of accurate fits.
Also the treatment of QCD radiative corrections leaves room for improvement, e.g. by incorporating bound state effects from the resummation of Coulomb type corrections [13, 14]. Finally, the design of the Hathor package can in principle also accommodate related approaches for the computation of the total top-quark pair cross section beyond NLO, for instance those based on soft gluon enhancement in differential kinematics [34, 35] (see Ref. [36] for earlier work). Such extrapolations of large logarithms from soft gluon emission in a differential variable (e.g. the top-quark pair-invariant mass) to the full partonic phase space introduce systematic uncertainties and require a detailed comparison to an inclusive approach such as in Eqs. (10)–(12).

3 Installation

In the default setting the Hathor package is based on the LHAPDF library [26] to access the PDFs. The Hathor package has been tested with the most recent version lhapdf-5.8.3 which can be obtained from `http://projects.hepforge.org/lhapdf`. Please follow the instructions in the LHAPDF package to install the LHAPDF library. To build and use Hathor first unpack

the package using `tar xvfz Hathor.tar.gz` at the location where one wants to install the package. This will create a directory Hathor-1.0. Please create a symbolic link `lhapdf` inside this directory to the location of ones LHAPDF installation. The contents of `lhapdf` should contain the LHAPDF installation with the directories: `bin include lib lib64 share`. Then use make to build the Hathor package. Make will build the Hathor (static) library `libHathor.a` which can be used in other applications. In addition an example program `main.exe` is built. For details concerning the example program we refer to Section 5. Note that the compilation is done using the GNU compilers gcc, gfortran and g++. The package is known to work also with the Intel compiler. In fact we recommend the usage of the Intel compiler since this results in a much better performance. However, given that it is not everywhere available Hathor uses the GNU compiler by default. [1]
To run the example one has to tell the system where the dynamic libraries for LHAPDF can be found. This is conveniently done using the environment variable `LD_LIBRARY_PATH`. Using the C-shell the statement would be:

```
setenv LD_LIBRARY_PATH <path_to_lhapdf_installation>/lib/
```

In addition, one probably needs to specify the location where the grid files for the LHAPDF library are stored. Again using the C-shell, the statement would be of the form:

```
setenv LHAPATH <path_to_lhapdf_installation>/lhapdf/share/lhapdf/PDFsets
```

For a detailed description concerning the paths required by the LHAPDF library, we refer to the LHAPDF manual. As concluding remarks with respect to the PDF libraries we would like to point out that Hathor does not try to handle errors from the LHAPDF library. This is not possible since LHAPDF does not provide a detailed error handling. Also note that since LHAPDF is not thread safe the same is true for Hathor.

Hathor is also equipped with a graphical user interface (GUI) written in Java. The GUI makes use of dynamic libraries to access the Hathor library within Java. [2] The dynamic library is built with the command `CreateJavaGui.csh` which is included in the Hathor package. The command `CreateJavaGui.csh` creates the dynamic library `libHathor4Java.so` and writes the executable file `xhathor` which is used to start the GUI. Please note that `xhathor` sets up various paths: i.e. the environment variable `LHAPATH` is set to `./lhapdf/share/lhapdf` if it has not been set yet. In addition the dynamic libraries `libLHAPDF.so` and `libgfortran.so` on which the Hathor library relies are preloaded. This step is platform dependent and not alway easy to achieve in a universal way. If the graphical interface does not start with `xhathor` the correct setting of the paths is a likely source of potential errors. In that case we recommend to set the necessary paths directly in `xhathor` or to consult the authors for support.

[1] For the Intel compiler the user has to adapt the `makefile`.

[2] The same technology can be used to access the Hathor library from Mathematica or Maple. The authors may provide the respective interfaces in a future update on demand.

4 Description

The entire cross section is calculated inside the class Hathor. This is done in order to avoid any possible problem with names used in already existing codes. Inside this class, a two dimensional numerical integration is performed in which the PDFs are convoluted with the hard scattering cross section. As a numerical integration procedure we use the Vegas algorithm [37]. [3] Since Vegas is a Monte Carlo integration we need to provide random numbers. Those are obtained by using the Ranlux algorithm [38] and we use the implementation available from Ref. [39].
The Hathor class takes as constructor argument a reference to the PDF which should be used in the current cross section calculation. In the following we list the publically available function calls and option choices together with a short description.

- `Hathor(Pdf & pdf)`
  Constructor to build one instance of the Hathor class. The argument is an instance of the PDF. In case that LHAPDF is used the corresponding definition would be:
  `Lhapdf pdf("MSTW2008nnlo68cl");`
  to use the `MSTW2008nnlo68cl` set.
  At first sight it might appear strange that we use an additional "wrapper class" as interface to LHAPDF. The idea behind this is to give the user the possibility to become independent from LHAPDF. This is achieved by inherting the class `Lhapdf` from the base class `Pdf`. By inhering its own class from the base class the user can thus easily implement its own wrapper to whatever PDF set he wants to use. As an example the `class MSTW` has been supplied, which gives direct access to the MSTW set [28]. (Note that in the MSTW case the α_s value is set to 1 since the library does not provide a function to evaluate it. The user has thus provide its own α_s.) We have observed that in some cases the original code provided with the PDF sets is faster than what is provided by LHAPDF. Since the evaluation of the PDFs represents a significant part of the calculation the usage of the original PDFs may speed up the entire calculation significantly.

- `void printOptions()`
  Use this function to print the options currently selected via the routine `setScheme();`

- `void setScheme(unsigned int newscheme)`
  Sets the specific scheme in particular perturbative order and renormalization schemes. Possible options are:

  - `Hathor::LO` to switch on the leading-order contribution.
  - `Hathor::NLO` to switch on the individual NLO contribution.
  - `Hathor::NNLO` to switch on the individual NNLO contribution (see Section 2).

  Please also note that for the computation of the total cross section up to e.g. NNLO accuracy, it is necessary to combine the options as in

[3] Hathor uses Vegas code which is a C port of the original fortran version [37].

`setScheme(Hathor::LO | Hathor::NLO | Hathor::NNLO);`
Other possible options are:

- `Hathor::MS_MASS` to use the mass renormalized in the $\overline{\text{MS}}$ scheme (see Section 2).
- `Hathor::LOG_ONLY` to keep only the logarithmically enhanced terms (see Section 2). Please note that this option requires also `Hathor::NNLO` to be set.
- `Hathor::PDF_SCAN` to switch on the evaluation of the PDF uncertainties. That is to say, the error PDFs are also integrated along with the central value. To save computing time one may set the accuracy with `XS.setPrecision(Hathor::LOW)` to `LOW` in this case. Care has to be taken, though, by the user to ensure that the default PDF uncertainty estimate as implemented in Hathor (asymmetric error) complies with the conventions of the respective PDF set, as e.g. some PDF sets provide only a symmetric error. In this case, the additional option `Hathor::PDF_SYM_ERR` needs to be set. See also the discussion in Section 2.
- `Hathor::PDF_SYM_ERR` invokes symmetric PDF error, if foreseen by the convention of the PDF set.

Please note, that these options can again be combined, e.g. as in
`setScheme(Hathor::LO | Hathor::NLO | Hathor::MS_MASS);`

- `void setColliderType( COLLIDERTYPE type)`
Sets the hadronic initial state. Use `Hathor::PP` to select proton–proton collisions and `Hathor::PPBAR` for proton–anti-proton collisions. The collider energies are set to the default values: 7 TeV in case of proton–proton collisions and 1960 GeV in case of proton–anti-proton collisions. If this is inappropriate the values can be changed using the command `void setSqrtShad(double ecms)`, where the center of mass energy is provided in GeV.

- `void setSqrtShad(double ecms)`
Sets the center-of-mass energy in GeV.

- `void setNf(int nf)`
Sets the number of massless flavors to `nf`. For top-quark physics the default setting is $n_f = 5$ and should not be changed. This function may be used when the cross section for a hypothetical heavy quark of a fourth family is computed, as Hathor includes the full n_f dependence of the hard scattering cross section, i.e. it features the formulae for general n_f. However, please note that the PDFs usually provide α_s in the $n_f = 5$ flavor scheme. So the user should be careful with this option (and the interpretation of the results).

- `void setCqq(double tmp)`
Can be used to set the constant defined in Eq. (10) to a specific value (see Section 2). The default is $C_{qq} = 0$.

- `void setCgg(double tmp)`
  Can be used to set the constant defined in Eq. (12) to a specific value (see Section 2). The default is $C_{gg} = 0$

- `void setPrecision(int n)`
  Can be used to define the accuracy of the numerical integration performed by the Hathor package. It sets the number of function evaluations used in the Monte Carlo integration. In principle, the user can provide any reasonable integer value.

  Pre-defined values are: `Hathor::LOW`, `Hathor::MEDIUM`, `Hathor::HIGH`. We recommend `LOW` for the PDF scan and `MEDIUM` for the central value. This should be sufficient for most applications. Please note that `Hathor::LOW` should give already an accuracy at the percent level. For detailed comparisons of theoretical results `HIGH` may be used.

- `double getAlphas(double mur)`
  Returns the QCD coupling constant at the renormalization scale `mur` as provided by the (central) PDF.

- `double getXsection(double m, double mur, double muf)`
  This function starts the cross section calculation for a given top-quark mass and the factorization/renormalization scales provided as arguments. Unless a specific scheme is set through `setScheme` the default setting is used:
  `Hathor::LO` | `Hathor::NLO` | `Hathor::NNLO`
  The cross section for the central PDF is returned. More information can be obtained through `getResult`.

- `void getResult(int pdfset, double & integral, double & err)`
  This function is used to obtain additional information after the cross section has been calculated for a specific setting of the renormalization/factorization scale using `getXsection`. The integer value `pdfset` specifies the respective PDF: 0 for the central value, and 1 to `getPdfNumber()` for the respective error PDF. The result for the central value and the numerical error from the numerical integration are returned through `integral` and `err`. Note that `err` should always be negligible. If this is not the case the precision of the numerical integration should be increased through `setPrecision`.

- `void getPdfErr(double & up, double \& down)`
  This function returns the PDF uncertainty, if the option `Hathor::PDF_SCAN` has been used. By default, Hathor assumes an asymmetric PDF error convention. In case of a symmetric one, the option `PDF_SCAN` has to be combined with the option `PDF_SYM_ERR` (see Section 2 and the discussion above).

- `int getPdfNumber()`
  Returns the number of error PDFs currently in use. If 0 is returned the option `PDF_SCAN` is not switched on or the PDF set does not support error PDFs. E.g., in case of the PDF set `mstw2008nnlo.68cl` the return value would be 40.

- `void sethc2(double)`
  Can be used to change the value for $(hc)^2$ which is used by Hathor to convert the cross sections from GeV^{-2} to pico barn. The default used by Hathor is:
  `0.389379323e+9`.

5 Usage and examples

A concrete instance of the Hathor class is built using:

```
Lhapdf pdf("MSTW2008nnlo68cl");
Hathor XS(pdf);
```

The evaluation of the cross section (using the default setting) is then done using

```
XS.getXsection(171.,171.*2,171./2);
```

where the mass has been set to 171 GeV and $\mu_r = 2 \times 171$ GeV and $\mu_f = 171/2$ GeV. The result of the evaluation is obtained through

```
XS.getResult(0,val,err,chi2a);
```

Note that `XS.getXsection(171.,171.*2,171./2);` triggers the numerical integration of the cross section. It has to be called first before `XS.getResult(0,val,err,chi2a);` can be used. In a typical application we may want to use a lower statistic in the Monte Carlo integration for the evaluation of the PDF uncertainty. This is achieved by the following code:

```
unsigned int scheme = Hathor::LO | Hathor::NLO | Hathor::NNLO;
double mt = 171., muf=171.,mur=171.;
double val,err,chi2a,up,down;

Lhapdf pdf("MSTW2008nnlo68cl");
Hathor XS(pdf);

XS.setPrecision(Hathor::MEDIUM);
XS.getXsection(mt,mur,muf);
XS.getResult(0,val,err,chi2a);

XS.setScheme(scheme | Hathor::PDF_SCAN);
XS.setPrecision(Hathor::LOW);
XS.getXsection(mt,mur,muf);
XS.getPdfErr(up,down);
```

The central value is calculated with the precision set to `MEDIUM`. The estimate of the PDF uncertainty is then obtained with a lower accuracy.
An example of the usage of Hathor illustrates the calculation with running a mass. It reproduces the central curve (NNLO) of the right plot in Figure 5 of Ref. [12].

```
double val,err,chi;

Lhapdf lhapdf("MSTW2008nnlo68cl");
Hathor XS(lhapdf);

XS.setColliderType(Hathor::PPBAR);
XS.setScheme(Hathor::LO | Hathor::NLO | Hathor::NNLO | Hathor::MS_MASS );
XS.setPrecision(Hathor::LOW);

for(double mt = 140; mt < 181.; mt++ ){
  XS.getXsection(mt,mt,mt);
  XS.getResult(0,val,err,chi);
  cout << mt << " " << XS.getAlphas(mt) <<" "<< val << " " << err << endl;
}
```

The typical runtimes for these examples range between seconds and a few minutes and also depend on the chosen compilers. E.g. on an Intel 3.00GHz QuadCore PC we have obtained with the options `NNLO`, `PDF_SCAN` (PDF set `MSTW2008nnlo68cl`) and `XS.setPrecision(Hathor::LOW)` the result for the cross section after 64 seconds using the gfortran compiler, and, 41 seconds respectively, using Intel's ifort compiler.

The Java GUI is invoked by the command `xhathor` (see the discussion in Sec. 3 for the installation). A screenshot is displayed in Fig. 1 and the input is self-explanatory with the help of Sec. 4 for the description of all options.

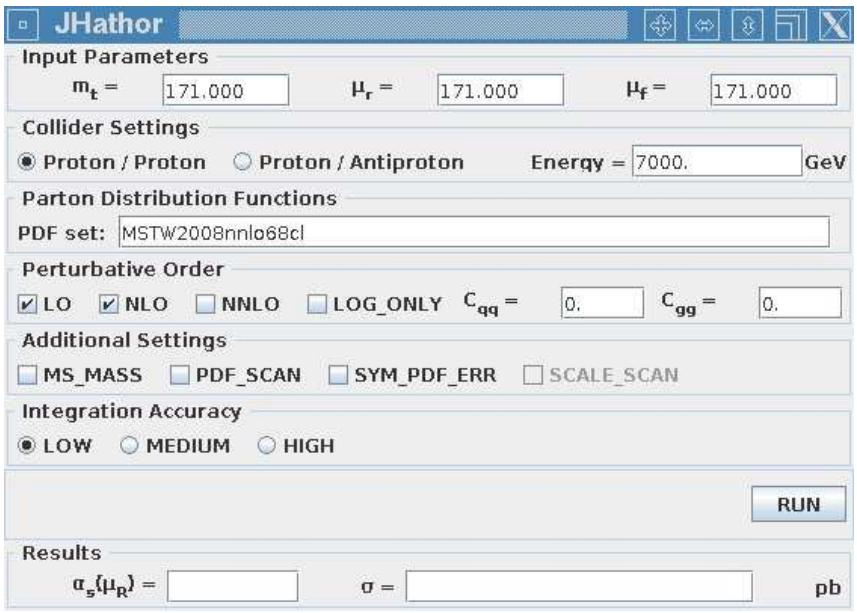

Figure 1: Screen shot of the Java graphical user interface for Hathor.

6 Conclusions

Top-quark production at hadron colliders is at the edge of becoming a precision measurement requiring accurate theory predictions. Hathor is a fast and flexible program for the computation of the total cross section of hadronic heavy-quark pair-production. It takes into account the latest theoretical developments through a variety of options, it allows for separate variations of all scales and can be used with a large number of PDFs through the LHAPDF interface. As a novelty, Hathor offers predictions in different renormalization schemes for the heavy-quark mass (pole and $\overline{\rm MS}$) and it can also be applied to a hypothetical fourth quark family assuming standard QCD couplings. Thus, with these functionalities Hathor can serve as a reference for future cross section calculations.
Hathor typically runs in a few seconds up to a few minutes (depending on the chosen options, e.g. extensive PDF scans) on standard desktop or notebook PCs. The Hathor package can either be used as a stand alone program or, as a small library, it can be easily integrated into existing code, e.g. for experimental analyses.

Hathor is publicly available for download from [40] or from the authors upon request.

Acknowledgments

We acknowledge useful discussions with S. Alekhin and U. Husemann. This work is supported in part by the Deutsche Forschungsgemeinschaft in Sonderforschungsbereich/Transregio 9 and the Helmholtz Gemeinschaft under contract VH-NG-105 and contract VH-HA-101 (Alliance *"Physics at the Terascale"*).

A Total cross section with a running mass

The starting point is the relation between the on-shell mass and the $\overline{\rm MS}$ mass:

$$m_t = \overline{m}(\mu_r)\left(1 + a_s \dot{1} + a_s^2 \dot{2}\right), \tag{A.1}$$

with $a_s = \alpha_s/\pi$. If the decoupling ${\alpha_s}^{n_f=6} \to {\alpha_s}^{n_f=5}$ is performed at $\overline{m}(\mu_r)$ the coefficients are given by

$$\dot{1} = \frac{4}{3} + \ell, \tag{A.2}$$

$$\begin{aligned}\dot{2} = {} & \frac{307}{32} + 2\zeta(2) + \frac{2}{3}\zeta(2)\ln 2 - \frac{1}{6}\zeta(3) + \frac{509}{72}\ell + \frac{47}{24}\ell^2 \\ & - \left(\frac{71}{144} + \frac{1}{3}\zeta(2) + \frac{13}{36}\ell + \frac{1}{12}\ell^2\right) n_f + \frac{4}{3}\sum_l \Delta(m_l/m_t),\end{aligned} \tag{A.3}$$

which are known from Refs. [41–43] and $\ell = \ln({\mu_r}^2/\overline{m}(\mu_r)^2)$. Note that the coefficients d_i depend in general on the renormalization scale. Using $\overline{m}(\overline{m})$ instead of $\overline{m}(\mu_r)$ the formulae simplify significantly. The full renormalization scale dependence can be restored at the end using

renormalization group arguments. $\Delta(m_i/m_t)$ accounts for all massive quarks m_i lighter than the top-quark. For all light quarks we set $m_l = 0$ so the sum in Eq. (A.3) vanishes.
To convert the cross section to the $\overline{\rm MS}$ mass scheme we start from the hadronic cross section expanded in α_s:

$$\sigma_{h_1h_2\to t\bar{t}X}(S,m_t) = a_s^2\sigma^{(0)}(S,m_t) + a_s^3\sigma^{(1)}(S,m_t) + a_s^4\sigma^{(2)}(S,m_t) + O(a_s^5)\,. \tag{A.4}$$

Expressing m_t in terms of $\overline{m}(\overline{m})$ and expanding in α_s we obtain

$$\begin{aligned}\sigma_{h_1h_2\to t\bar{t}X} &= a_s^2\sigma^{(0)}(S,\overline{m}(\overline{m})) + a_s^3\sigma^{(1)}(S,\overline{m}(\overline{m})) + a_s^4\sigma^{(2)}(S,\overline{m}(\overline{m})) \\ &\quad + a_s^3 1\dot{\overline{m}}(\overline{m})\left.\frac{d\sigma^{(0)}(m_t)}{dm_t}\right|_{m_t=\overline{m}(\overline{m})} + a_s^4\left\{\overline{m}(\overline{m})2\dot{}\left.\frac{d\sigma^{(0)}(m_t)}{dm_t}\right|_{m_t=\overline{m}(\overline{m})}\right. \\ &\quad \left. + \overline{m}(\overline{m})1\dot{}\left.\frac{d\sigma^{(1)}(m_t)}{dm_t}\right|_{m_t=\overline{m}(\overline{m})} + \frac{1}{2}\overline{m}(\overline{m})^2 1^2\left.\frac{d^2\sigma^{(0)}(m_t)}{dm_t{}^2}\right|_{m_t=\overline{m}(\overline{m})}\right\}.\end{aligned} \tag{A.5}$$

The derivatives of the LO cross sections with respect to the mass can be written in the following form:

$$\left.\frac{d\sigma^{(0)}(m_t)}{dm_t}\right|_{m_t=\overline{m}(\overline{m})} = \frac{2}{\overline{m}(\overline{m})}\int_{4\overline{m}(\overline{m})^2}^{S} ds\left(s\frac{d}{ds}\mathcal{L}_{ij}(s,S,\mu_f)\right)\hat{\sigma}_{ij}^{(0)}(s,\overline{m}(\overline{m}))\,, \tag{A.6}$$

and

$$\begin{aligned}\left.\frac{d^2\sigma^{(0)}}{dm_t^2}\right|_{m_t=\overline{m}(\overline{m})} &= -\frac{2}{\overline{m}(\overline{m})^2}\int_{4\overline{m}(\overline{m})^2}^{S} ds\left(s\frac{d}{ds}\mathcal{L}_{ij}(s,S,\mu_f)\right)\hat{\sigma}_{ij}^{(0)}(s,\overline{m}(\overline{m})) \\ &\quad + \frac{2}{\overline{m}(\overline{m})}\int_{4\overline{m}(\overline{m})^2}^{S} ds\left(s\frac{d}{ds}\mathcal{L}_{ij}(s,S,\mu_f)\right)\left.\frac{d\hat{\sigma}_{ij}^{(0)}}{dm_t}\right|_{m_t=\overline{m}(\overline{m})},\end{aligned} \tag{A.7}$$

where a summation over the contributing parton channels is understood. Note that in Eqs. (A.5), (A.6) and (A.7) the renormalization scale is set to $\mu_r = \overline{m}(\overline{m})$. The required derivatives of the LO partonic cross sections with respect to the mass are easily obtained from Eqs. (4), (5):

$$m_t\frac{d\sigma_{qq}}{dm_t} = -\frac{1}{m_t^2}\frac{1}{9}\pi{\alpha_s}^2\frac{\rho^3}{\beta}\,, \tag{A.8}$$

$$m_t\frac{d\sigma_{gg}}{dm_t} = \frac{1}{192}\pi{\alpha_s}^2\frac{1}{m_t^2}\frac{\rho}{\beta}\left(\beta(36-40\beta^2+4\beta^4)\ln\left(\frac{1+\beta}{1-\beta}\right) - 7 - 116\beta^2 + 91\beta^4\right). \tag{A.9}$$

For the first derivative of $\sigma^{(1)}$ we obtain a similar result:

$$\begin{aligned}\left.\frac{d\sigma^{(1)}}{dm_t}\right|_{m_t=\overline{m}(\overline{m})} &= -\int_{4\overline{m}(\overline{m})^2}^{S} ds\mathcal{L}_{ij}(s,S,\mu_f)\frac{1}{\overline{m}(\overline{m})}\tilde{\sigma}_{ij}^{(1)}(s,\overline{m}(\overline{m})) \\ &\quad + \frac{2}{\overline{m}(\overline{m})}\int_{4\overline{m}(\overline{m})^2}^{S} ds\left(s\frac{d}{ds}\mathcal{L}_{ij}(s,S,\mu_f)\right)\sigma^{(i)}\left(s,\overline{m}(\overline{m}),\frac{\mu_f}{\overline{m}(\overline{m})},1\right),\end{aligned} \tag{A.10}$$

with

$$\tilde{\sigma}_{ij}^{(1)}(s,\overline{m}(\overline{m})) = \mu_f \frac{\partial}{\partial \mu_f} \sigma_{ij}^{(1)} \left(s, \overline{m}(\overline{m}), \frac{\mu_f}{\overline{m}(\overline{m})}, 1 \right) . \tag{A.11}$$

Using Eq. (6) the contribution $\tilde{\sigma}_{ij}^{(1)}(s,\overline{m}(\overline{m}))$ can be written as

$$a_s^2 \tilde{\sigma}_{ij}^{(1)} = 8 \frac{{\alpha_s}^2(\overline{m}(\overline{m}))}{\overline{m}(\overline{m})^2} f_{ij}^{(11)} . \tag{A.12}$$

Since the luminosities are only known numerically the derivatives are evaluated using

$$\frac{d}{ds}\mathcal{L}_{ij}(s,S,\mu_f) = \frac{1}{2\delta}\left(\mathcal{L}_{ij}(s+\delta,S,\mu_f) - \mathcal{L}_{ij}(s-\delta,S,\mu_f)\right) + O(\delta^2) . \tag{A.13}$$

The results presented so far are only valid for $\mu_r = \overline{m}(\overline{m})$. Using

$$a_s(\overline{m}(\overline{m})) = a_s(\mu_r)\left(1 + 4\pi^2 a_s(\mu_r) L_{\bar{R}} \underline{0} + (4\pi^2)^2 a_s(\mu_r)^2 (1 L_{\bar{R}} + \underline{0}^2 L_{\bar{R}}^2)\right) , \tag{A.14}$$

with $L_{\bar{R}} = \ln({\mu_r}^2/\overline{m}(\overline{m})^2)$ it is easy to restore the complete renormalization scale dependence.

B Scaling functions

Here we present the expressions for the scaling functions as implemented in the program Hathor. At Born level,

$$f_{q\bar{q}}^{(0)} = \frac{\pi\beta\rho}{27}[2+\rho] , \tag{B.1}$$

$$f_{gq}^{(0)} = 0 , \tag{B.2}$$

$$f_{gg}^{(0)} = \frac{\pi\beta\rho}{192}\left[\frac{1}{\beta}(\rho^2+16\rho+16)\ln\left(\frac{1+\beta}{1-\beta}\right) - 28 - 31\rho\right] , \tag{B.3}$$

where $\beta = \sqrt{1-\rho}$ and $\rho = 4m_t^2/s$. At NLO the functions $f_{ij}^{(10)}$ have been described through precise parametrizations with per mil accuracy and the following ansatz [12]:

$$f_{q\bar{q}}^{(10)} = \frac{\rho\beta}{36\pi}\left[\frac{32}{3}\ln^2\beta + \left(32\ln 2 - \frac{82}{3}\right)\ln\beta - \frac{1}{12}\frac{\pi^2}{\beta}\right] + \beta\rho a_0^{qq} + h(\beta, a_1, \ldots, a_{17}) \tag{B.4}$$
$$+ \frac{1}{8\pi^2}(n_f-4) f_{q\bar{q}}^{(0)}\left[\frac{4}{3}\ln 2 - \frac{2}{3}\ln\rho - \frac{10}{9}\right] ,$$

$$f_{gq}^{(10)} = \frac{1}{16\pi}\beta^3\left[\frac{5}{9}\ln\beta + \frac{5}{6}\ln 2 - \frac{73}{108}\right] + h_{gq}^{(a)}(\beta, a_1, \ldots, a_{15}) , \tag{B.5}$$

$$f_{gg}^{(10)} = \frac{7\beta}{768\pi}\left[24\ln^2\beta + \left(72\ln 2 - \frac{366}{7}\right)\ln\beta + \frac{11}{84}\frac{\pi^2}{\beta}\right] + \beta a_0^{gg} + h(\beta, a_1, \ldots, a_{17}) \tag{B.6}$$
$$+ (n_f-4)\frac{\rho^2}{1024\pi}\left[\ln\left(\frac{1+\beta}{1-\beta}\right) - 2\beta\right] ,$$

where n_f denotes the number of light quarks and the complete n_f-dependence has been kept manifest. The constants a_0^{ij} read $a_0^{qq} = 0.03294734$ and $a_0^{gg} = 0.01875287$ and the fit functions $h(\beta, a_1, \ldots, a_{17})$ and $h_{gq}^{(a)}(\beta, a_1, \ldots, a_{15})$ are given in Eqs. (B.18), (B.19) together with a list of all parameters in Tabs. 1–3. The exact expressions for scale dependent functions $f_{ij}^{(11)}$ have already compact analytical form containing at most dilogarithms. They read [1]

$$f_{q\bar{q}}^{(11)} = \frac{1}{8\pi^2}\left[\frac{16\pi}{81}\rho\ln\left(\frac{1+\beta}{1-\beta}\right) + \frac{1}{9}f_{q\bar{q}}^{(0)}(\rho)\left(127 - 6n_f + 48\ln\left(\frac{\rho}{4\beta^2}\right)\right)\right], \tag{B.7}$$

$$f_{gq}^{(11)} = \frac{1}{8\pi^2}\frac{\pi}{192}\left[\frac{4}{9}\rho\left(14\rho^2 + 27\rho - 136\right)\ln\left(\frac{1+\beta}{1-\beta}\right)\right. \tag{B.8}$$
$$\left. - \frac{32}{3}\rho\left(2-\rho\right)h_1(\beta) - \frac{8}{135}\beta\left(1319\rho^2 - 3468\rho + 724\right)\right],$$

$$f_{gg}^{(11)} = \frac{1}{8\pi^2}\left[\frac{\pi}{192}\left\{2\rho\left(59\rho^2 + 198\rho - 288\right)\ln\left(\frac{1+\beta}{1-\beta}\right)\right.\right. \tag{B.9}$$
$$+12\rho\left(\rho^2 + 16\rho + 16\right)h_2(\beta) - 6\rho\left(\rho^2 - 16\rho + 32\right)h_1(\beta)$$
$$\left.\left. - \frac{4}{15}\beta\left(7449\rho^2 - 3328\rho + 724\right)\right\} + 12 f_{gg}^{(0)}(\rho)\ln\left(\frac{\rho}{4\beta^2}\right)\right],$$

with the auxiliary functions containing the standard dilogarithm $\mathrm{Li}_2(x) = -\int_0^x \frac{\mathrm{d}t}{t}\ln(1-t)$,

$$h_1(\beta) = \ln^2\left(\frac{1+\beta}{2}\right) - \ln^2\left(\frac{1-\beta}{2}\right) + 2\mathrm{Li}_2\left(\frac{1+\beta}{2}\right) - 2\mathrm{Li}_2\left(\frac{1-\beta}{2}\right), \tag{B.10}$$

$$h_2(\beta) = \mathrm{Li}_2\left(\frac{2\beta}{1+\beta}\right) - \mathrm{Li}_2\left(\frac{-2\beta}{1-\beta}\right). \tag{B.11}$$

At NNLO the functions $f_{ij}^{(21)}$ and $f_{ij}^{(22)}$ are known exactly [12]. The fits to the scaling functions generally have per mil accuracy with exceptions in regions close to zero, where an accuracy better than one percent is kept.

$$f_{q\bar{q}}^{(21)} = \frac{1}{(16\pi^2)^2}f_{q\bar{q}}^{(0)}\left[-\frac{8192}{9}\ln^3\beta + \left(\frac{12928}{3} - \frac{32768}{9}\ln 2\right)\ln^2\beta\right. \tag{B.12}$$
$$\left. + \left(-840.51065 + 70.183854\frac{1}{\beta}\right)\ln\beta - 82.246703\frac{1}{\beta} + 467.90402\right]$$
$$+ \frac{n_f}{(16\pi^2)^2}f_{q\bar{q}}^{(0)}\left[-\frac{256}{3}\ln^2\beta + \left(\frac{2608}{9} - \frac{2816}{9}\ln 2\right)\ln\beta + 6.5797363\frac{1}{\beta} - 64.614276\right]$$
$$+ h(\beta, b_i + n_f c_i) - \frac{4{n_f}^2}{(16\pi^2)^2}f_{q\bar{q}}^{(0)}\left[\frac{4}{3}\ln 2 - \frac{2}{3}\ln\rho - \frac{10}{9}\right],$$

$$f_{q\bar{q}}^{(22)} = \frac{1}{(16\pi^2)^2}f_{q\bar{q}}^{(0)}\left[\frac{2048}{9}\ln^2\beta + \left(-\frac{7840}{9} + \frac{4096}{9}\ln 2\right)\ln\beta + 270.89724\right] \tag{B.13}$$
$$+ \frac{n_f}{(16\pi^2)^2}f_{q\bar{q}}^{(0)}\left[\frac{320}{9}\ln\beta - \frac{596}{9} + \frac{320}{9}\ln 2\right] + h(\beta, b_i + n_f c_i) + \frac{4{n_f}^2}{3(16\pi^2)^2}f_{q\bar{q}}^{(0)},$$

$$
\begin{aligned}
f_{gq}^{(21)} &= -\frac{\pi}{(16\pi^2)^2}\beta^3\Big[\frac{770}{27}\ln^2\beta+\Big(-\frac{6805}{81}+\frac{6160}{81}\ln 2\Big)\ln\beta+0.13707784\frac{1}{\beta} \qquad \text{(B.14)}\\
&\quad +0.22068868\Big]-\frac{\pi n_f}{81(16\pi^2)^2}\beta^3\Big[46\ln\beta-\frac{163}{3}+76\ln 2\Big]+h_{gq}^{(b)}(\beta,b_i+n_f c_i)\\
f_{gq}^{(22)} &= \frac{\pi}{(16\pi^2)^2}\beta^3\Big[\frac{385}{81}\ln\beta-\frac{1540}{243}+\frac{385}{81}\ln 2\Big]+h_{gq}^{(b)}(\beta,b_i+n_f c_i)\,, \qquad \text{(B.15)}\\
f_{gg}^{(21)} &= \frac{1}{(16\pi^2)^2}f_{gg}^{(0)}\Big[-4608\ln^3\beta+\Big(\frac{109920}{7}-18432\ln 2\Big)\ln^2\beta \qquad \text{(B.16)}\\
&\quad +\Big(69.647185-248.15005\frac{1}{\beta}\Big)\ln\beta+56.867721\frac{1}{\beta}+17.010070\Big]\\
&\quad +\frac{n_f}{(16\pi^2)^2}f_{gg}^{(0)}\Big[-64\ln^2\beta+\Big(\frac{4048}{21}-192\ln 2\Big)\ln\beta-3.4465285\frac{1}{\beta}-37.602004\Big]\\
&\quad +h(\beta,b_i+n_f c_i)\,,\\
f_{gg}^{(22)} &= \frac{1}{(16\pi^2)^2}f_{gg}^{(0)}\Big[1152\ln^2\beta+(-2568+2304\ln 2)\ln\beta-79.74312140\Big] \qquad \text{(B.17)}\\
&\quad +\frac{n_f}{(16\pi^2)^2}f_{gg}^{(0)}\Big[16\ln\beta-16+16\ln 2\Big]+h(\beta,b_i+n_f c_i)\,,
\end{aligned}
$$

with the fit functions $h(\beta,a_1,\ldots,a_{17})$ and $h_{gq}^{(b)}(\beta,a_1,\ldots,a_{18})$ (see Tabs. 1–3 for a list of all parameters),

$$
\begin{aligned}
h(\beta,a_1,\ldots,a_{17}) &= a_1\beta^2+a_2\beta^3+a_3\beta^4+a_4\beta^5 \qquad \text{(B.18)}\\
&\quad +a_5\beta^2\ln\beta+a_6\beta^3\ln\beta+a_7\beta^4\ln\beta+a_8\beta^5\ln\beta\\
&\quad +a_9\beta^2\ln^2\beta+a_{10}\beta^3\ln^2\beta+a_{11}\beta\ln\rho+a_{12}\beta\ln^2\rho+a_{13}\beta^2\ln\rho\\
&\quad +a_{14}\beta^2\ln^2\rho+a_{15}\beta^3\ln\rho+a_{16}\beta^3\ln^2\rho+a_{17}\beta^4\ln\rho\,,\\
h_{gq}^{(a)}(\beta,a_1,\ldots,a_{15}) &= a_1\beta^4+a_2\beta^5+a_3\beta^6 \qquad \text{(B.19)}\\
&\quad +a_4\beta^4\ln\beta+a_5\beta^5\ln\beta+a_6\beta^6\ln\beta\\
&\quad +a_7\beta^2\rho\ln\rho+a_8\beta^2\rho\ln^2\rho+a_9\beta^3\rho\ln\rho\\
&\quad +a_{10}\beta^3\rho\ln^2\rho+a_{11}\beta^4\rho\ln\rho\\
&\quad +a_{12}\beta^4\rho\ln^2\rho+a_{13}\beta^2\rho\ln^3\rho+a_{14}\beta^2\rho\ln^4\rho+a_{15}\beta^2\rho\ln^5\rho\,,\\
h_{gq}^{(b)}(\beta,a_1,\ldots,a_{18}) &= a_1\beta^3+a_2\beta^4+a_3\beta^5+a_4\beta^6+a_5\beta^7 \qquad \text{(B.20)}\\
&\quad +a_6\beta^4\ln\beta+a_7\beta^5\ln\beta+a_8\beta^6\ln\beta+a_9\beta^7\ln\beta\\
&\quad +a_{10}\beta^3\ln\rho+a_{11}\beta^3\ln^2\rho+a_{12}\beta^4\ln\rho+a_{13}\beta^4\ln^2\rho\\
&\quad +a_{14}\beta^5\ln\rho+a_{15}\beta^5\ln^2\rho+a_{16}\beta^6\ln\rho+a_{17}\beta^6\ln^2\rho+a_{18}\beta^7\ln\rho\,.
\end{aligned}
$$

	$f_{q\bar{q}}^{(10)}$	$f_{q\bar{q}}^{(21)}$		$f_{q\bar{q}}^{(22)}$	
i	a_i	b_i	c_i	b_i	c_i
1	0.07120603	−0.15388765	−0.07960658	0.37947056	−0.00224114
2	−1.27169999	4.85226571	0.50111294	−4.25138041	0.02685576
3	1.24099536	−7.06602840	−0.09496432	2.91716094	−0.01777126
4	−0.04050443	2.36935255	−0.32590203	0.94994470	−0.00626121
5	0.02053737	−0.03634651	−0.02229012	0.10537529	−0.00062062
6	−0.31763337	1.25860837	0.23397666	−1.69689874	0.00980999
7	−0.71439686	2.75441901	0.30223487	−2.60977181	0.01631175
8	0.01170002	−1.26571709	0.13113818	−0.27215567	0.00182500
9	0.00148918	−0.00230536	−0.00162603	0.00787855	−0.00004627
10	−0.14451497	0.15633927	0.08378465	−0.47933827	0.00286176
11	−0.13906364	1.79535231	−0.09147804	−0.18217132	0.00111459
12	0.01076756	0.36960437	−0.01581518	−0.04067972	0.00017425
13	0.49397845	−5.45794874	0.26834309	0.54147194	−0.00359593
14	−0.00567381	−0.76651636	0.03251642	0.08404406	−0.00035339
15	−0.53741901	5.35350436	−0.25679483	−0.51918414	0.00363300
16	−0.00509378	0.39690927	−0.01670122	−0.04336452	0.00017915
17	0.18250366	−1.68935685	0.07993054	0.15957988	−0.00115164

Table 1: Coefficients for fits of the $q\bar{q}$ scaling functions.

	$f_{gq}^{(10)}$	$f_{gq}^{(21)}$		$f_{gq}^{(22)}$	
i	a_i	b_i	c_i	b_i	c_i
1	−0.26103970	−0.00120532	0.00003257	−0.00022247	0.00001789
2	0.30192672	−0.04906353	0.00014276	0.00050422	0.00000071
3	−0.01505487	−0.20885725	−0.00402017	−0.02945504	−0.00020581
4	−0.00142150	−13.73137224	0.06329831	0.34340412	0.00108759
5	−0.04660699	14.01818840	−0.05952825	−0.31894917	−0.00086284
6	−0.15089038	−0.00930488	0.00002694	0.00009213	0.00000010
7	−0.25397761	−0.52223668	0.00159804	0.00690402	0.00001638
8	−0.00999129	−4.68440515	0.01522672	0.07847233	0.00022730
9	0.39878717	−7.61046166	0.02869438	0.16042051	0.00045698
10	−0.02444172	1.36687743	−0.00875589	−0.05186974	−0.00025620
11	−0.14178346	1.84698291	−0.00800271	−0.03861021	−0.00016026
12	0.01867287	−7.26265988	0.04043479	0.21650362	0.00070713
13	0.00238656	−4.89364026	0.01965878	0.10137656	0.00034937
14	−0.00003399	11.04566784	−0.05262293	−0.28056264	−0.00072547
15	−0.00000089	4.13660190	−0.01457395	−0.08090469	−0.00025525
16	0.00000000	−6.33477051	0.02314616	0.13077889	0.00034015
17	0.00000000	−1.08995440	0.00291792	0.01813862	0.00006613
18	0.00000000	1.19010561	−0.00220115	−0.01585757	−0.00006562

Table 2: Coefficients for fits of the gq scaling functions.

	$f_{gg}^{(10)}$	$f_{gg}^{(21)}$		$f_{gg}^{(22)}$	
i	a_i	b_i	c_i	b_i	c_i
1	−8.92563222	−4.18931464	0.12306772	0.01222783	−0.00380386
2	149.90572830	82.35066406	−2.75808806	−0.77856184	0.08757766
3	−140.55601420	−87.87311969	3.19739272	1.33955698	−0.10742267
4	−0.34115615	9.80259328	−0.56233045	−0.59108409	0.02382706
5	−2.41049833	−1.12268550	0.03240048	0.00248333	−0.00099760
6	54.73381889	29.51830225	−0.92541788	−0.23827213	0.02932941
7	90.91548015	48.36110694	−1.57154712	−0.38868910	0.04906147
8	−4.88401008	−7.06261770	0.35109760	0.28342153	−0.01373734
9	−0.17466779	−0.08025226	0.00227936	0.00010876	−0.00006986
10	13.47033628	7.01493779	−0.21030153	−0.03383862	0.00658371
11	22.66482710	15.00588140	−0.63688407	−0.29071016	0.02089321
12	4.60726682	3.84142441	−0.12959776	−0.11473654	0.00495414
13	−67.62342328	−47.02161789	1.91690216	0.98929369	−0.06553459
14	−9.70391427	−8.05583379	0.26755747	0.24899069	−0.01046635
15	65.08050888	47.02740535	−1.86154423	−1.06096321	0.06559130
16	5.09663260	4.21438052	−0.13795865	−0.13425338	0.00551218
17	−20.12225341	−14.99599732	0.58155056	0.35935660	−0.02095059

Table 3: Coefficients for fits of the *gg* scaling functions.

References

[1] P. Nason, S. Dawson, and R. K. Ellis, Nucl. Phys. **B303**, 607 (1988).

[2] W. Beenakker, H. Kuijf, W. L. van Neerven, and J. Smith, Phys. Rev. **D40**, 54 (1989).

[3] W. Bernreuther, A. Brandenburg, Z. G. Si, and P. Uwer, Nucl. Phys. **B690**, 81 (2004), arXiv:hep-ph/0403035.

[4] M. Czakon and A. Mitov, Nucl. Phys. **B824**, 111 (2010), arXiv:0811.4119.

[5] N. Kidonakis and G. Sterman, Nucl. Phys. **B505**, 321 (1997), arXiv:hep-ph/9705234.

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

[7] S. Moch and P. Uwer, Phys. Rev. **D78**, 034003 (2008), arXiv:0804.1476.

[8] M. Czakon, A. Mitov, and G. Sterman, Phys. Rev. **D80**, 074017 (2009), arXiv:0907.1790.

[9] M. Beneke, P. Falgari, and C. Schwinn, Nucl. Phys. **B828**, 69 (2010), arXiv:0907.1443.

[10] S. Moch and P. Uwer, Nucl. Phys. Proc. Suppl. **183**, 75 (2008), arXiv:0807.2794.

[11] M. Beneke, M. Czakon, P. Falgari, A. Mitov, and C. Schwinn, (2009), arXiv:0911.5166.

[12] U. Langenfeld, S. Moch, and P. Uwer, Phys. Rev. **D80**, 054009 (2009), arXiv:0906.5273.

[13] K. Hagiwara, Y. Sumino, and H. Yokoya, Phys. Lett. **B666**, 71 (2008), arXiv:0804.1014.

[14] Y. Kiyo, J. H. Kühn, S. Moch, M. Steinhauser, and P. Uwer, Eur. Phys. J. **C60**, 375 (2009), arXiv:0812.0919.

[15] W. Beenakker, A. Denner, W. Hollik, R. Mertig, T. Sack, and D. Wackeroth, Nucl. Phys. **B411**, 343 (1994).

[16] W. Bernreuther, M. Fücker, and Z.-G. Si, Phys. Rev. **D74**, 113005 (2006), arXiv:hep-ph/0610334.

[17] J. H. Kühn, A. Scharf, and P. Uwer, Eur. Phys. J. **C51**, 37 (2007), arXiv:hep-ph/0610335.

[18] S. Moch, U. Langenfeld, and P. Uwer, (2010), arXiv:1001.3987.

[19] S. Dittmaier, P. Uwer, and S. Weinzierl, Phys. Rev. Lett. **98**, 262002 (2007), hep-ph/0703120.

[20] S. Dittmaier, P. Uwer, and S. Weinzierl, Eur. Phys. J. **C59**, 625 (2009), arXiv:0810.0452.

[21] K. Melnikov and M. Schulze, JHEP **0908**, 049 (2009), arXiv:arXiv:0907.3090.

[22] M. Cacciari, S. Frixione, M. L. Mangano, P. Nason, and G. Ridolfi, JHEP **09**, 127 (2008), arXiv:0804.2800.

[23] M. Cacciari, M. Czakon, M. L. Mangano, A. Mitov, S. Moch, P. Nason, and P. Uwer, (2010), to appear.

[24] I. I. Bigi, M. A. Shifman, N. Uraltsev, and A. Vainshtein, Phys.Rev. **D50**, 2234 (1994), arXiv:hep-ph/9402360.

[25] M. C. Smith and S. S. Willenbrock, Phys.Rev.Lett. **79**, 3825 (1997), arXiv:hep-ph/9612329.

[26] M. R. Whalley, D. Bourilkov, and R. C. Group, (2005), hep-ph/0508110.

[27] CEDAR HepForge, LHAPDF data base, `http://projects.hepforge.org/lhapdf/`.

[28] A. D. Martin, W. J. Stirling, R. S. Thorne, and G. Watt, Eur. Phys. J. **C63**, 189 (2009), arXiv:0901.0002.

[29] P. M. Nadolsky *et al.*, Phys. Rev. **D78**, 013004 (2008), arXiv:0802.0007.

[30] J. M. Campbell, J. Huston, and W. Stirling, Rept.Prog.Phys. **70**, 89 (2007), arXiv:hep-ph/0611148.

[31] S. Alekhin, J. Blümlein, S. Klein, and S. Moch, Phys. Rev. **D81**, 014032 (2010), arXiv:0908.2766.

[32] S. I. Alekhin, Phys.Rev. **D63**, 094022 (2001), arXiv:hep-ph/0011002.

[33] R. D. Ball *et al.*, (2010), arXiv:arXiv:1002.4407.

[34] N. Kidonakis and R. Vogt, Phys. Rev. **D78**, 074005 (2008), arXiv:0805.3844.

[35] V. Ahrens, A. Ferroglia, M. Neubert, B. D. Pecjak, and L. L. Yang, (2010), arXiv:1003.5827.

[36] N. Kidonakis, E. Laenen, S. Moch, and R. Vogt, Phys. Rev. **D64**, 114001 (2001), arXiv:hep-ph/0105041.

[37] G. P. Lepage, J. Comp. Phys. **27**, 192 (1978).

[38] M. Lüscher, Comput. Phys. Commun. **79**, 100 (1994), arXiv:hep-lat/9309020.

[39] M. Lüscher, Ranlux, `http://luscher.web.cern.ch/luscher/ranlux/index.html` (GNU license).

[40] Hathor, `http://www.physik.hu-berlin.de/pep/tools/` or `http://www-zeuthen.desy.de/~moch/hathor` .

[41] N. Gray, D. J. Broadhurst, W. Gräfe, and K. Schilcher, Z.Phys. **C48**, 673 (1990).

[42] K. Chetyrkin and M. Steinhauser, Nucl.Phys. **B573**, 617 (2000), arXiv:hep-ph/9911434.

[43] K. Melnikov and T. v. Ritbergen, Phys.Lett. **B482**, 99 (2000), arXiv:hep-ph/9912391.