Preprint typeset in JHEP style - PAPER VERSION

Cavendish–HEP–10/12
CERN-TH/2010-216
IPPP/10/62
DCPT/10/124

The MC@NLO 4.0 Event Generator*

Stefano Frixione†
PH Department, TH Unit, CERN, CH-1211 Geneva 23, Switzerland
ITPP, EPFL, CH-1015 Lausanne, Switzerland
E-mail: Stefano.Frixione@cern.ch

Fabian Stoeckli
PH Department, CMG Group, CERN, CH-1211 Geneva 23, Switzerland
E-mail: Fabian.Stoeckli@cern.ch

Paolo Torrielli
ITPP, EPFL, CH-1015 Lausanne, Switzerland
E-mail: Paolo.Torrielli@epfl.ch

Bryan R. Webber
Cavendish Laboratory, J.J. Thomson Avenue, Cambridge CB3 0HE, U.K.
E-mail: webber@hep.phy.cam.ac.uk

Chris D. White
Department of Physics and Astronomy, University of Glasgow,
Glasgow G12 8QQ, Scotland, U.K.;
Institute for Particle Physics Phenomenology, Department of Physics, Durham University, Durham DH1 3LE, U.K.
E-mail: c.d.white@durham.ac.uk

Abstract: This is the user's manual of MC@NLO 4.0. This package is a practical implementation, based upon the Fortran HERWIG and Herwig++ event generators, of the MC@NLO formalism, which allows one to incorporate NLO QCD matrix elements consistently into a parton shower framework. Processes available in this version include the hadroproduction of single vector and Higgs bosons, vector boson pairs, heavy quark pairs, single top, single top in association with a W, single top in association with a charged Higgs in type I or II 2HDM models, lepton pairs, and Higgs bosons in association with a W or Z. Spin correlations are included for all processes except ZZ production. This document is self-contained, but we emphasise the main differences with respect to previous versions.

*Work supported in part by the UK Science and Technology Facilities Council, and by the Swiss National Science Foundation.

†On leave of absence from INFN, Sez. di Genova, Italy

Contents

1. Generalities **2**
- 1.1 Citation policy 2
- 1.2 Physics processes 2

2. Structure of the package **6**
- 2.1 Working environment 6
- 2.2 Source and running directories 6

3. Prior to running **8**
- 3.1 Usage with HW6 8
- 3.2 Usage with HW++ 9

4. Running **9**
- 4.1 Parton densities 10
  - 4.1.1 LHAPDF 11
  - 4.1.2 PDF uncertainties 12
- 4.2 Event file 13
- 4.3 Inclusive NLO cross sections 14
- 4.4 Wt and $H^{\pm}t$ production 16
- 4.5 $W^{\pm}Z$ production and anomalous couplings 17
- 4.6 Decays 18
- 4.7 Results 20

5. Script variables **21**

A. Version changes **25**
- A.1 From MC@NLO version 1.0 to version 2.0 25
- A.2 From MC@NLO version 2.0 to version 2.1 26
- A.3 From MC@NLO version 2.1 to version 2.2 26
- A.4 From MC@NLO version 2.2 to version 2.3 26
- A.5 From MC@NLO version 2.3 to version 3.1 27
- A.6 From MC@NLO version 3.1 to version 3.2 27
- A.7 From MC@NLO version 3.2 to version 3.3 28
- A.8 From MC@NLO version 3.3 to version 3.4 28
- A.9 From MC@NLO version 3.4 to version 4.0 30

B. Running the package without the shell scripts **30**
- B.1 Creating the executables 30
- B.2 The input files 31

1. Generalities

In this document, we briefly describe how to run the MC@NLO package, implemented according to the formalism introduced in ref. [1]. The production processes now available are listed in tables 1 and 2. The process codes `IPROC` and the variables `IV` and $\texttt{IL}_\alpha$ will be explained below. $H_{1,2}$ represent hadrons (in practice, nucleons or antinucleons). The information given in refs. [1, 2] allows the implementation in MC@NLO of any production process, provided that the formalism of refs. [3, 4] is used for the computation of cross sections to NLO accuracy. The production matrix elements have been taken from the following references: vector boson pairs [5, 6, 7], WZ production and decay with anomalous couplings [8], heavy quark pairs [9], Standard Model Higgs [10, 11], single vector boson [12], lepton pairs [13], associated Higgs [14], single-top s- and t-channel [15], single-top in association with a charged Higgs [16]; those for single-top production in association with a W have been re-derived and thoroughly compared to those of ref. [17].

This documentation refers to MC@NLO version 4.0. The major difference with respect to previous MC@NLO versions is that in the present one all processes can be simulated either with Fortran HERWIG [18, 19, 20], or with Herwig++ [21, 22] (with the exception, in the latter case, of single-top in association with a charged Higgs, and of $b\bar{b}$ production. While the latter process is fully implemented at the level of short-distance cross sections, it displays an anomalously large event-failure rate in the shower phase, which is currently under investigation). In what follows, these two event generators will be referred to as HW6 and HW++ respectively, and collectively as EvG[1]. Single-top production in association with a charged Higgs has been added since version 3.4, including spin correlations which were not yet implemented in ref. [16]. The present version includes the upgrades of sub-versions 3.41 and 3.42. For precise details of version changes, see app. A.1-A.9.

1.1 Citation policy

When using MC@NLO with HW6, please cite ref. [1]; if HW++ is adopted, please cite refs. [1, 24]. In addition to ref. [1] or to refs. [1, 24], if $t\bar{t}$ or $b\bar{b}$ events are generated, please also cite ref. [2]; if s- or t-channel single-top events are generated, please also cite ref. [25]; if Wt single-top events are generated, please also cite ref. [26]; if $H^{\pm}t$ single-top events are generated, please also cite ref. [16]; if $W^{\pm}Z$ events are generated, please also cite ref. [27]. The current user manual, or any other user manuals relevant to past versions, should not be cited unless the relevant papers mentioned above are cited too.

1.2 Physics processes

In the case of standard MC, a hard kinematic configuration is generated on a event-by-event basis, and it is subsequently showered and hadronized. In the case of MC@NLO, all of the hard kinematic configurations are generated in advance, and stored in a file (which we call the *event file* – see sect. 4.2); the event file is then read by the EvG, which showers

[1]The matching to PYTHIA has been worked out analytically in ref. [23] for the case of initial-state radiation. The production processes implemented in computer codes are still limited in number, and are not part of the present package.

and hadronizes each hard configuration. Since version 2.0, the events are handled by the "Les Houches" generic user process interface [28] (see ref. [2] for more details). Therefore, in MC@NLO the reading of a hard configuration from the event file is equivalent to the generation of such a configuration in a standard MC.

The signal to the EvG that configurations should be read from an event file using the Les Houches interface is a negative value of the process code `IPROC`; this accounts for the negative values in tables 1 and 2. In the case of heavy quark pair, Higgs, Higgs in association with a W or Z, and lepton pair (through Z/γ^* exchange) production, the codes are simply the negative of those for the corresponding standard HW6 MC processes. Where possible, this convention will be adopted for additional MC@NLO processes, regardless of the EvG actually used to shower events. Consistently with what happens in standard HW6, by subtracting 10000 from `IPROC` one generates the same processes as in tables 1 and 2, but eliminates the underlying event[2]. Note that this operation is performed within `mcatnlo_hwdriver.f` (for HW6), or by the scripts `MCatNLO_pp.Script` (for HW++). Therefore, if these files are not employed when running the EvG's, the user will be responsible for switching the underlying event off manually if so desired.

For processes with a SM Higgs (denoted by H^0) in the final state, the user may specify the identities of its decay products when using HW6, by adding `-ID` to the process code. The conventions for `ID` are the same as in HW6, namely `ID` $= 1 \dots 6$ for $u\bar{u} \dots t\bar{t}$; 7, 8, 9 for e^+e^-, $\mu^+\mu^-$, $\tau^+\tau^-$; 10, 11 for W^+W^-, ZZ; and 12 for $\gamma\gamma$. Furthermore, `ID` $= 0$ gives quarks of all flavours, and `ID` $= 99$ gives all decays. It should be stressed that the event file does not contain the Higgs decay products, and therefore is independent of the value of `ID`; the decay is dealt with by HW6.[3] In the case of HW++, the user will instead select the decay products using the HW++ input file `HWPPInput.inputs`, which we shall describe later. This means that `ID` has no effect in SM Higgs production when using HW++.

Process codes `IPROC`$=-1360-$`IL` and $-1370-$`IL` do not have an analogue in HW6; they are the same as $-1350-$`IL`, except for the fact that only a Z or a γ^* respectively is exchanged. The value of `IL` determines the lepton identities, and the same convention as in HW6 is adopted: `IL`$=1, \dots, 6$ for $l_{\rm IL} = e, \nu_e, \mu, \nu_\mu, \tau, \nu_\tau$ respectively (see also table 3). At variance with HW6, `IL` cannot be set equal to zero. Process codes `IPROC`$=-1460-$`IL` and $-1470-$`IL` are the analogue of HW6 1450+`IL`; in HW6 either W^+ or W^- can be produced, whereas MC@NLO treats the two vector bosons separately. For these processes, as in HW6, `IL`$=1, 2, 3$ for $l_{\rm IL} = e, \mu, \tau$, but again the choice `IL` $= 0$ is not allowed.

The lepton pair processes `IPROC`$=-1350-$`IL`, ..., $-1470-$`IL` include spin correlations when generating the angular distributions of the produced leptons. However, if spin correlations are not an issue, the single vector boson production processes `IPROC`$=$ -1396,-1397,-1497,-1498 can be used, in which case the vector boson decay products are distributed (by HW6, which then generates the decays) according to phase space. These processes should be considered only as a quicker alternative to lepton-pair production, with

[2]The same effect can be achieved by setting the HW6 parameter `PRSOF` $= 0$.

[3]Spin correlations between the decay products of vector boson pairs emerging from Higgs decays were neglected in HW6 versions older than the current one, 6.520. Please check the wiki at http://projects.hepforge.org/fherwig/trac/report for release reports on this and other improvements.

IPROC	IV	$\mathtt{IL}_1$	$\mathtt{IL}_2$	Spin	Process
–1350–IL				✓	$H_1H_2 \to (Z/\gamma^* \to) l_{\rm IL}\bar{l}_{\rm IL} + X$
–1360–IL				✓	$H_1H_2 \to (Z \to) l_{\rm IL}\bar{l}_{\rm IL} + X$
–1370–IL				✓	$H_1H_2 \to (\gamma^* \to) l_{\rm IL}\bar{l}_{\rm IL} + X$
–1460–IL				✓	$H_1H_2 \to (W^+ \to) l^+_{\rm IL}\nu_{\rm IL} + X$
–1470–IL				✓	$H_1H_2 \to (W^- \to) l^-_{\rm IL}\bar{\nu}_{\rm IL} + X$
–1396				×	$H_1H_2 \to \gamma^*(\to \sum_i f_i\bar{f}_i) + X$
–1397				×	$H_1H_2 \to Z^0 + X$
–1497				×	$H_1H_2 \to W^+ + X$
–1498				×	$H_1H_2 \to W^- + X$
–1600–ID					$H_1H_2 \to H^0 + X$
–1705					$H_1H_2 \to b\bar{b} + X$
–1706		7	7	×	$H_1H_2 \to t\bar{t} + X$
–2000–IC		7		×	$H_1H_2 \to t/\bar{t} + X$
–2001–IC		7		×	$H_1H_2 \to \bar{t} + X$
–2004–IC		7		×	$H_1H_2 \to t + X$
–2030		7	7	×	$H_1H_2 \to tW^-/\bar{t}W^+ + X$
–2031		7	7	×	$H_1H_2 \to \bar{t}W^+ + X$
–2034		7	7	×	$H_1H_2 \to tW^- + X$
–2040		7	7	×	$H_1H_2 \to tH^-/\bar{t}H^+ + X$
–2041		7	7	×	$H_1H_2 \to \bar{t}H^+ + X$
–2044		7	7	×	$H_1H_2 \to tH^- + X$
–2600–ID	1	7		×	$H_1H_2 \to H^0W^+ + X$
–2600–ID	1	i		✓	$H_1H_2 \to H^0(W^+ \to) l^+_i\nu_i + X$
–2600–ID	-1	7		×	$H_1H_2 \to H^0W^- + X$
–2600–ID	-1	i		✓	$H_1H_2 \to H^0(W^- \to) l^-_i\bar{\nu}_i + X$
–2700–ID	0	7		×	$H_1H_2 \to H^0Z + X$
–2700–ID	0	i		✓	$H_1H_2 \to H^0(Z \to) l_i\bar{l}_i + X$
–2850		7	7	×	$H_1H_2 \to W^+W^- + X$
–2860		7	7	×	$H_1H_2 \to Z^0Z^0 + X$
–2870		7	7	×	$H_1H_2 \to W^+Z^0 + X$
–2880		7	7	×	$H_1H_2 \to W^-Z^0 + X$

Table 1: Some of the processes implemented in MC@NLO 4.0 (see also table 2). $H_{1,2}$ represent nucleons or antinucleons. IPROC–10000 generates the same processes as IPROC, but eliminates the underlying event. A void entry indicates that the corresponding variable is unused. The 'Spin' column indicates whether spin correlations in vector boson or top decays are included (✓), neglected (×) or absent (void entry); when included, spin correlations are obtained by direct integration of the relevant NLO matrix elements. Spin correlations in Higgs decays to vector boson pairs (e.g. $H^0 \to W^+W^- \to l^+\nu l^-\bar{\nu}$) are included in HW6 versions 6.520 and higher. Processes -1705 and $-2040-$IC are not available for HW++ at present.

IPROC	IV	IL_1	IL_2	Spin	Process
–1706		i	j	✓	$H_1H_2 \to (t \to)b_k f_i f'_i(\bar{t} \to)\bar{b}_l f_j f'_j + X$
–2000–IC		i		✓	$H_1H_2 \to (t \to)b_k f_i f'_i/(\bar{t} \to)\bar{b}_k f_i f'_i + X$
–2001–IC		i		✓	$H_1H_2 \to (\bar{t} \to)\bar{b}_k f_i f'_i + X$
–2004–IC		i		✓	$H_1H_2 \to (t \to)b_k f_i f'_i + X$
–2030		i	j	✓	$H_1H_2 \to (t \to)b_k f_i f'_i(W^- \to)f_j f'_j/$ $(\bar{t} \to)\bar{b}_k f_i f'_i(W^+ \to)f_j f'_j + X$
–2031		i	j	✓	$H_1H_2 \to (\bar{t} \to)\bar{b}_k f_i f'_i(W^+ \to)f_j f'_j + X$
–2034		i	j	✓	$H_1H_2 \to (t \to)b_k f_i f'_i(W^- \to)f_j f'_j + X$
–2040		i		✓	$H_1H_2 \to (t \to)b_k f_i f'_i H^-/$ $(\bar{t} \to)\bar{b}_k f_i f'_i H^+ + X$
–2041		i		✓	$H_1H_2 \to (\bar{t} \to)\bar{b}_k f_i f'_i H^+ + X$
–2044		i		✓	$H_1H_2 \to (t \to)b_k f_i f'_i H^- + X$
–2850		i	j	✓	$H_1H_2 \to (W^+ \to)l_i^+\nu_i(W^- \to)l_j^-\bar{\nu}_j + X$
–2870		i	j	✓	$H_1H_2 \to (W^+ \to)l_i^+\nu_i(Z^0 \to)l'_j\bar{l}'_j + X$
–2880		i	j	✓	$H_1H_2 \to (W^+ \to)l_i^-\bar{\nu}_i(Z^0 \to)l'_j\bar{l}'_j + X$

Table 2: Some of the processes implemented in MC@NLO 4.0 (see also table 1). $H_{1,2}$ represent nucleons or antinucleons. For more details on Wt and $H^{\pm}t$ production, see sect. 4.4. Spin correlations for the processes in this table are implemented according to the method presented in ref. [29]. b_α ($\bar{b}_\alpha$) can either denote a b (anti)quark or a generic down-type (anti)quark. f_α and f'_α can denote a (anti)lepton or an (anti)quark. See sects. 4.3 and 4.6 for fuller details. Process $-2040-$IC is not available for HW++ at present.

a more limited physics content; for this reason, they have not been interfaced to HW++. There are a number of other differences between the lepton pair and single vector boson processes. The latter do not feature the γ–Z interference terms. Also, their cross sections are fully inclusive in the final-state fermions resulting from γ^*, Z or $W^{\pm}$. The user can still select a definite decay mode using the HW6 variable MODBOS (see sect. 4.6), but the relevant branching ratio will *not* be included by MC@NLO. As stated previously, these processes are not available for running with HW++.

In NLO computations for single-top production in the SM, it is customary to distinguish between three production mechanisms, conventionally denoted as the s channel, t channel, and Wt mode. Starting from version 3.4, all three mechanisms are implemented in MC@NLO; s- and t-channel single top production correspond to setting IC=10 and IC=20 respectively. For example, according to tables 1 and 2, t-channel single-$\bar{t}$ events will be generated by entering IPROC=-2021. These two channels can also be simulated simultaneously (by setting IC=0). We point out that the Wt cross section is ill-defined beyond the leading order in QCD, which is also the case for $H^{\pm}t$ production. See sect. 4.4 for more details.

In the case of vector boson pair production, the process codes are the negative of those adopted in MC@NLO 1.0 (for which the Les Houches interface was not yet available), rather than those of standard HW6.

Furthermore, in the case of $t\bar{t}$, single-t, $H^0W^\pm$, H^0Z, W^+W^-, and $W^\pm Z$ production, the value of `IPROC` alone may not be sufficient to fully determine the process type (including decay products), and variables `IV`, $\texttt{IL}_1$, and $\texttt{IL}_2$ are also needed (see tables 1 and 2). In the case of top decays (and of the decay of the hard W in Wt production), the variables $\texttt{IL}_1$ and $\texttt{IL}_2$ have a more extended range of values than that of the variable `IL`, which is relevant to lepton pair production and to which they are analogous (notice, however, that in the latter case `IL` is not an independent variable, and its value is included via `IPROC`). In addition, $\texttt{IL}_\alpha$=7 implies that spin correlations for the decay products of the corresponding particle are not taken into account, as indicated in table 1. More details are given in sect. 4.6.

Apart from the above differences, MC@NLO and HW6 or HW++ *behave in exactly the same way*. Thus, the available user's analysis routines can be used in the case of MC@NLO. One should recall, however, that MC@NLO always generates some events with negative weights (see ref. [1]); therefore, the correct distributions are obtained by summing weights with their signs (i.e., the absolute values of the weights must *NOT* be used when filling the histograms).

With such a structure, it is natural to create two separate executables, which we improperly denote as NLO and MC. The former has the sole scope of creating the event file; the latter is just the EvG "executable". In the case of HW6, this file coincides with the actual HW6 executable. In the case of HW++, it is a script that calls the relevant HW++ executables.

2. Structure of the package

2.1 Working environment

We have written shell scripts and `Makefile`'s which will simplify the use of the package considerably. In order to use them, the computing system must support `bash` shell, and `gmake`[4]. Should they be unavailable on the user's computing system, the compilation and running of MC@NLO requires more detailed instructions; in this case, we refer the reader to app. B. This appendix will serve also as a reference for a more advanced use of the package.

2.2 Source and running directories

The package can be downloaded as a tarball from the web page:

`http://www.hep.phy.cam.ac.uk/theory/webber/MCatNLO`

The structure of the directories that contain the source codes has become more involved starting from MC@NLO version 4.0, in order to deal with the possibility of interfacing to more than one EvG. The directory tree will be created automatically when unpacking the tarball. We describe it here briefly.

The directory in which the tarball will be unpacked will be called the *source directory*. The source directory contains the following files:

[4]For Macs running under OSX v10 or higher, `make` can be used instead of `gmake`.

`MCatNLO.inputs`
`HWPPInput.inputs`
`MCatNLO.Script`
`MCatNLO_pp.Script`
`Makefile`
`Makefile_pp`
`MCatNLO_rb.inputs`

and the following subdirectories

`HW6Analyzer`
`HWppAnalyzer`
`srcCommon`
`srcHerwig6`
`srcHerwigpp`
`include`

The user will be primarily interested in `MCatNLO.inputs` (or `MCatNLO_rb.input`, see later), which contains all input parameters common to HW6 and to HW++. Further input parameters, specific to HW++, can be found in `HWPPInput.inputs`. The user will have to write his/her own analysis routines, and place them into the `HW6Analyzer` or the `HWppAnalyzer` directories for HW6 or HW++ runs respectively. Sample analysis files are provided in order to give the user a ready-to-run package. The other files of the package must not be modified, with the possible exception of the HW6 driver (`mcatnlo_hwdriver.f`), and Les Houches interface (`mcatnlo_hwlhin.f`), to be found in `srcHerwig6` – see sect. 3 for further details.

When creating the executable, our shell scripts determine the type of operating system, and create a subdirectory of the source directory, which we call the *running directory*, whose name depends on the operating system. For Linux, the name of the running directory will be `Linux` when using HW6, and `LinuxPP` when using HW++. On other operating systems, possible names of running directories are `AlphaXX`, `SunXX`, or `RunXX`, with `XX` either the empty string or `PP`. Tests on operating systems other than Linux have been performed sporadically on Mac's, and in the case of HW6 only; we recommend using Linux whenever possible.

The running directory contains all the object files and executable files, and in general all the files produced by the MC@NLO while running. It must also contain the relevant grid files (see sect. 4.1), or links to them, if the library of parton densities provided with the MC@NLO package is used.

In the subdirectory `HW6Analyzer` of the source directory, the user will find the files `mcatnlo_hwan`*xxx*`.f` (which use a version of HBOOK written by M. Mangano that outputs plots in TopDrawer format) and `mcatnlo_hwan`*xxx*`_rb.f` (which use front-end Fortran routines written by W. Verkerke [30] for filling histograms in Root format). These are sample HW6 analysis routines, one for each of the processes implemented in this package. They are provided here to give the user a ready-to-run package, but they should be replaced with appropriate codes according to the user's needs. Examples of how to use these analysis files in

MC@NLO are given in the (otherwise identical) `MCatNLO.inputs` and `MCatNLO_rb.inputs` files (see sect. 4 for more details on input cards). In the subdirectory `HWppAnalyzer` the user will find analogous codes, to be used when running with HW++. We do not provide sample analyses for all processes in this case, and only the TopDrawer-format version is given, since Root is a native-C++ code that can be much more easily used with HW++ than with HW6.

In addition to the files listed above, the user will need a version of the EvG code(s) to be used – HW6 [18, 19, 20] or HW++[21, 22]. As stressed in ref. [1], for MC@NLO we do not modify the existing (LL) shower algorithm. However, since MC@NLO versions 2.0 and higher make use of the Les Houches interface, first implemented in HERWIG 6.5, the version of HW6 must be 6.500 or higher. When using HW6, on most operating systems users will need to delete the dummy subroutines UPEVNT, UPINIT, PDFSET and STRUCTM from the standard HW6 package, to permit linkage of the corresponding routines from the MC@NLO package. As a general rule, the user is strongly advised to use the most recent versions of HERWIG, which were the ones used in the testing phase of MC@NLO.

3. Prior to running

3.1 Usage with HW6

When using HW6, the user must be aware of the fact that the files:

`mcatnlo_hwdriver.f`

`mcatnlo_hwlhin.f`

which can be found in `srcHerwig6`, and the files:

`mcatnlo_hwan`*xxx*`.f`

`mcatnlo_hwan`*xxx*`_rb.f`

which can be found in `HW6Analyzer`, contain the statement `INCLUDE HERWIG65.INC`, which indicates that the code will link to version 6.500 or higher, for the reasons explained above. In the current MC@NLO release, the file `HERWIG65.INC` contains the statement

```
INCLUDE 'herwig6520.inc'
```

We do not assume that the user will adopt version 6.520, which is the latest release; for this reason, the user may need to edit the file `HERWIG65.INC`, and change the statement above into

```
INCLUDE 'herwig65nn.inc'
```

with `65nn` the version chosen by the user (this must be consistent with the value of the input parameter `HERWIGVER`, see sects. 4 and 5).

The file `mcatnlo_hwdriver.f` contains a set of read statements, which are necessary for HW6 to get the input parameters (see sect. 4 for the input procedure); these read statements must not be modified or eliminated. Also, `mcatnlo_hwdriver.f` calls the HW6 routines which perform showering, hadronization, decays (see sect. 4.6 for more details on this issue), and so forth; the user can freely modify this part, as is customary in HW6 runs. Finally, the sample codes `mcatnlo_hwan`*xxx*`.f` and `mcatnlo_hwan`*xxx*`_rb.f` contain analysis-related routines: these files must be replaced by files which contain the user's analysis routines. We point out that, since version 2.0, the `Makefile` need not be edited

any longer, since the corresponding operations are now performed by setting script variables (see sect. 5).

3.2 Usage with HW++

When using HW++, no editing of the source codes is required, except for those related to analysis, to be found in the directory `HWppAnalyzer`. The analogues of the read statements of HW6 `mcatnlo_hwdriver.f` are, in the case of HW++, the settings of parameters in `HWPPInput.inputs`, which the user can modify at will. However, the user will have to provide installed versions of HW++ and ThePEG [31]. We advise the user to adopt versions 2.4.2 for HW++ and 1.6.1 for ThePEG or later. We stress that the options `ReconstructionOption=General` and `InitialInitialBoostOption=LongTransBoost` must be used when running HW++. These are automatically set by our script, and the user must not change them. The latter option is presently available only in the trunk version of HW++ 2.4.2.

4. Running

It is straightforward to run MC@NLO. First, edit[5]

```
MCatNLO.inputs
```

and write there all the input parameters (for the complete list of the input parameters, see sect. 5). Further parameters specific to HW++ that control the behaviour of this EvG can be set in `HWPPInput.inputs`. As the last line of the file `MCatNLO.inputs`, write

```
runMCatNLO
```

Finally, execute `MCatNLO.inputs` from the `bash` shell. This procedure will create the NLO and MC executables, and run them using the inputs given in `MCatNLO.inputs`, which guarantees that the parameters used in the NLO and MC runs are consistent. Should the user only need to create the executables without running them, or to run the NLO or the MC only, he/she should replace the call to `runMCatNLO` in the last line of `MCatNLO.inputs` by calls to

```
compileNLO
compileMC
runNLO
runMC
```

which have obvious meanings. Take note that in the case of HW++, the `compileMC` command actually does not compile the HW++ executable. This has already been done in the installation of the HW++ package. The `compileMC` command rather compiles the chosen analyzer, creates all necessary soft links in the running directory, and creates the executable MC script.

We point out that if using HW6 the command `runMC` may be used with `IPROC`=1350+`IL`, 1450+`IL`, 1600+`ID`, 1699, 1705, 1706, 2000–2008, 2600+`ID`, 2699, 2700+`ID`, 2799, 2800, 2810, 2815, 2820, 2825 to generate Z/γ^*, $W^\pm$, Higgs, $b\bar{b}$, $t\bar{t}$, single top, H^0W, H^0Z, and

[5] See below for comments on `MCatNLO_rb.inputs`

vector boson pair events with standard HW6 (see the HW6 manual for more details). Note that the events thus produced are weighted events (except for single-top production), since we have set `NOWGT=.FALSE.` in `mcatnlo_hwdriver.f`; however, the user can freely change this setting. For obvious reasons this does not work with HW++. In order to use standalone HW++ please consult the dedicated manual.

We stress that the input parameters are not solely related to physics (masses, CM energy, and so on); there are several of them which control other things, such as the number of events generated. These must also be set by the user, according to his/her needs: see sect. 5.

Two such variables are `HERWIGVER` and `HWUTI`, which were moved in version 2.0 from the `Makefile` to `MCatNLO.inputs`, and which are relevant to runs with HW6. The former variable must be set equal to the object file name of the version of HW6 currently adopted (matching the one whose common blocks are included in the files mentioned in sect. 3). The variable `HWUTI` must be set equal to the list of object files that the user needs in the analysis routines. In the case of HW++, the analogue of `HWUTI` is `HWPPANALYZER`, which must be set equal to the name of the C++ analysis code the user means to use. The variable `HERWIGVER` does not have an analogue when using HW++; its role is played by the variables `HWPPPATH` and `THEPEGPATH`, which must be set equal to the physical address of the base directories of the HW++, and of the ThePEG installations, respectively.

The sample input file `MCatNLO.inputs` provided in this package is relevant to $t\bar{t}$ production and subsequent t and $\bar{t}$ leptonic decays. Similar sample inputs are given in the file `MCatNLO_rb.inputs`, which is identical to the former, except that at the end of the MC run an output file in Root format will be produced (as opposed to the output file in TopDrawer format produced by `MCatNLO.inputs`); for this to happen, the user will have to edit `MCatNLO_rb.inputs` in order to insert the path to the Root libraries for the machine on which the run is performed (shell variables `EXTRAPATHS` and `INCLUDEPATHS`). We stress that, apart from the differences in the output formats, `MCatNLO.inputs` and `MCatNLO_rb.inputs` have exactly the same meaning. Thus, although for the sake of brevity we shall often refer only to `MCatNLO.inputs` in this manual, all the issues concerning the inputs apply to `MCatNLO_rb.inputs` as well. Furthermore, as stressed above, an explicit example of how to obtain results in Root format is only necessary in the case of HW6 runs, and therefore `MCatNLO_rb.inputs` provided here should be used only in conjunction with that EvG.

If the shell scripts are not used to run the codes, the inputs are given to the NLO or MC codes during an interactive talk-to phase; the complete sets of inputs for our codes are reported in app. B.2 for vector boson pair production.

4.1 Parton densities

Since knowledge of the parton densities (PDFs) is necessary in order to get the physical cross section, a PDF library must be linked. For the NLO runs, the possibility exists to link the (now obsolete) CERNLIB PDF library (PDFLIB), or its replacement LHAPDF [32]; however, we also provide a self-contained PDF library with this package, which is faster than PDFLIB, and contains PDF sets released after the last and final PDFLIB version (8.04; most of these sets are now included in LHAPDF). The three PDF libraries mentioned

above can also be used for HW6 runs. As far as HW++ is concerned, the only possibility is that of linking to LHAPDF, or that of using the default HW++ PDF set. If this is desired, the variable `HERPDF` must be set equal to `DEFAULT` (see below for further comments). A complete list of the PDFs available in our PDF library can be downloaded from the MC@NLO web page. The user may link one of the three PDF libraries; all that is necessary is to set the variable `PDFLIBRARY` (in the file `MCatNLO.inputs`) equal to `THISLIB` if one wants to link to our PDF library, and equal to `PDFLIB` or to `LHAPDF` if one wants to link to PDFLIB or to LHAPDF. Our PDF library collects the original codes, written by the authors of the PDF fits; as such, for most of the densities it needs to read the files which contain the grids that initialize the PDFs. These files, which can also be downloaded from the MC@NLO web page, must either be copied into the running directory, or defined in the running directory as logical links to the physical files (by using `ln -sn`). We stress that if the user runs MC@NLO with the shell scripts, the logical links will be created automatically at run time. Starting from MC@NLO version 3.4, the reading of parameters associated with the selected PDF set from LHAPDF has been made fully robust (see sect. 4.1.1). For this reason, recent PDF sets are not being added to our PDF library, and when adopting these (and/or when running HW++) LHAPDF must be used.

As stressed before, consistent inputs must be given to the NLO and MC codes. However, in ref. [1] we found that the dependence upon the PDFs used by HW6 is rather weak. So one may want to run the NLO and HW6 adopting a regular NLL-evolved set in the former case, and the default HW6 set in the latter (the advantage is that this option reduces the amount of running time of HW6). In order to do so, the user must set the variable `HERPDF` equal to `DEFAULT` in the file `MCatNLO.inputs`; setting `HERPDF=EXTPDF` will force HW6 to use the same PDF set as the NLO code.

On the other hand, we found that the use of the HW++ default PDF set can be problematic in versions where the default set is a so-called LO* PDF set. We observed unexpected features, especially in $p_{\rm T}$ spectra. These features are manifest in both MC@NLO and HW++ standalone runs, and disappear when other sets are used. We did not investigate further the origin of the problem, but we deprecate the use of the HW++ default PDF set with MC@NLO and recommend using the same set as in the NLO run.

When using HW6, regardless of the PDFs used in the MC run users must delete the dummy PDFLIB routines PDFSET and STRUCTM from the HW6 source code, as explained earlier.

4.1.1 LHAPDF

As mentioned above, by setting `PDFLIBRARY=LHAPDF` in the input file the code is linked to the LHAPDF library. The user may choose whether to link to the static or to the dynamic LHAPDF library (the latter will produce a smaller executable but otherwise results are identical to those obtained with the former). This has obviously no effect for the MC step when using HW++, since the way LHAPDF is linked to the HW++ executable is defined in the HW++ installation. Starting from MC@NLO version 4.0[6] this choice is made

[6] In version 3.4 or earlier, linking to the static library was the default, and linking to the dynamic one

by assigning to the variable LHALINK in MCatNLO.inputs the values STATIC or DYNAMIC respectively. In order for the Makefile's to be able to find the LHAPDF library, the variable LHAPATH in MCatNLO.inputs should be set equal to the name of the directory where the local version of LHAPDF is installed. This is typically the name of the directory where one finds the files libLHAPDF.a and libLHAPDF.so, except for the final /lib in the directory name.

As is well known, a given PDF set has a preferred value of $\Lambda_{\rm QCD}$, which should be used in the computation of short-distance cross sections. Upon setting LAMBDAFIVE in MCatNLO.inputs equal to a negative value, this choice is made automatically. However, when linking to PDFLIB or LHAPDF, the code has to rely on the value $\Lambda_{\rm QCD}$ stored (by the PDF libraries) in a common block. This is far from ideal, since $\Lambda_{\rm QCD}$ is not a physical parameter, and in particular is dependent upon the form adopted for $\alpha_{\rm s}$, which may not be the same as that used in MC@NLO. Starting from version 3.4, the above automatic choice has been rendered more solid in the case of a linkage to LHAPDF; the code now reads the value of $\alpha_{\rm s}(M_Z)$ (i.e., of a physical quantity) from the PDF library, and converts it into a value for $\Lambda_{\rm QCD}$ using the form of $\alpha_{\rm s}(Q^2)$ used internally in MC@NLO. MC@NLO will print out on the standard output when running the NLO code (FPREFIXNLO.log if using the scripts) the value of $\Lambda_{\rm QCD}$ used in the computation. Such a value is now expected to be quite close to that listed under the column labeled with $\Lambda^{(5)}_{\rm QCD}$(MeV) on our PDF library manual (which can be found on the MC@NLO web page).

Version 4.0 of MC@NLO has been tested to link and run with several versions of LHAPDF. In particular, the user is not supposed to edit the Makefile's if linking with LHAPDF version 5.0 or higher. If one is interested into linking with earlier versions of LHAPDF (which is strongly deprecated), then one must replace the string mcatnlo_uti.o in the variable LUTIFILES in the Makefile, with the string mcatnlo_utilhav4.o. Again, the version of LHAPDF used in the MC step when using HW++ is defined when installing HW++. We advise however to try to always use the same LHAPDF package, i.e. link MC@NLO with the same library as HW++.

4.1.2 PDF uncertainties

The use of error sets to estimate the uncertainties due to PDFs which affect cross sections implies one computation for each of the members of the given error set – when performing this procedure with MC@NLO, we recommend to set HERPDF=EXTPDF.

The procedure is straightforward but computing intensive, and an approximate solution is that of reweighting the results obtained with the default PDF set by ratios of PDFs. In order to do so, information is necessary on the values of the fractional momenta of the incoming partons x_1 and x_2, and on the scale squared Q^2 used in the computation of the PDFs. Starting from MC@NLO version 4.0, this information is made available to the user on an event-by-event basis, without the necessity of kinematical reconstructions on the EvG event record. In the case of HW6, the values of these quantities can be found in the

required the use of the scripts in MCatNLO_dyn.Script. These scripts are now obsolete, and have been removed from the package, with their companion Makefile_dyn.

common block

```
DOUBLE PRECISION UX1,UX2,UQ2
COMMON/CPDFRWGT/UX1,UX2,UQ2
```

with UX1= x_1, UX2= x_2, and UQ2= Q^2 (in GeV^2). In the case of HW++, the information can be found right after the compulsory event information, in the form:

```
#pdf    x1 x2 Q^2
```

We point out that reweighting with different PDFs is never exact in the context of an NLO computation, and this is especially true when such computation is interfaced to an event generator, as in MC@NLO, since PDF effects in Sudakovs cannot possibly be taken into account in this way. We therefore recommend performing PDF reweighting with utmost care.

4.2 Event file

The NLO code creates the event file. In order to do so, it goes through two steps; first it integrates the cross sections (integration step), and then, using the information gathered in the integration step, produces a set of hard events (event generation step). Integration and event generation are performed with a modified version of the SPRING-BASES package [33].

We stress that the events stored in the event file contain only the partons involved in the hard subprocesses. Owing to the modified subtraction introduced in the MC@NLO formalism (see ref. [1]) they do not correspond to pure NLO configurations, and should not be used to plot physical observables. Parton-level observables must be reconstructed using the fully-showered events.

The event generation step necessarily follows the integration step; however, for each integration step one can have an arbitrary number of event generation steps, i.e., an arbitrary number of event files. This is useful in the case in which the statistics accumulated with a given event file is not sufficient.

Suppose the user wants to create an event file; editing `MCatNLO.inputs`, the user sets `BASES=ON`, to enable the integration step, sets the parameter `NEVENTS` equal to the number of events wanted on tape, and runs the code; the information on the integration step (unreadable to the user, but needed by the code in the event generation step) is written on files whose name begin with `FPREFIX`, a string the user sets in `MCatNLO.inputs`; these files (which we denote as *data files*) have extensions `.data`. The name of the event file is `EVPREFIX.events`, where `EVPREFIX` is again a string set by the user.

Now suppose the user wants to create another event file, to increase the statistics. The user simply sets `BASES=OFF`, since the integration step is not necessary any longer (however, the data files must not be removed: the information stored there is still used by the NLO code); changes the string `EVPREFIX` (failure to do so overwrites the existing event file), while keeping `FPREFIX` at the same value as before; and changes the value of `RNDEVSEED` (the random number seed used in the event generation step; failure to do so results in an event file identical to the previous one); the number `NEVENTS` generated may or may not be equal to the one chosen in generating the former event file(s).

We point out that data and event files may be very large. If the user wants to store them in a scratch area, this can be done by setting the script variable `SCRTCH` equal to the

physical address of the scratch area (see sect. 4.7).

For historical reasons, the formats of the event files to be used by HW6 and HW++ are not identical; the former uses an internal MC@NLO format, whereas the latter uses the Les Houches format. It is crucial to realize that event files meant to be showered by, say, HW++ must not be showered by HW6 (and the other way round). This has nothing to do with the format of the event files (which one may consider changing) and is instead due to the fact that the short-distance cross sections in MC@NLO do depend on the EvG used for the shower.

It has to be noted that in the case of HW++ the command `runMC` will typically not process the entire event file. The reason for this is the following. By using the Les Houches interface, a run will not stop automatically when reaching the end of the event file, which will then be oversampled. This happens since an EvG counts as events processed only those which are successfully showered, which are typically slightly less than those read from the event file. In order to avoid oversampling, we therefore give in input to HW++ a number of events equal to 98% times `NEVENTS` (the latter variable should coincide with the number of events in the event file). The safety margin of 2% is amply sufficient to avoid oversampling. The user may change such a safety margin by manipulating the MC executable script by hand, by changing the `-N` parameter passed there to HW++.

4.3 Inclusive NLO cross sections

MC@NLO integrates NLO matrix elements in order to produce the event file, and thus computes (as a by-product) the inclusive NLO cross section. This cross section (whose value is given in *pb*) can be obtained from an MC@NLO run in three different ways when running HW6[7]:

a) It is printed out at the end of the NLO run (search for `Total for fully inclusive` in the standard output).

b) It is printed by HW6 at the end of the MC run (search for `CROSS SECTION (PB)` in the standard output).

c) It is equal to the integral of any differential distribution which covers the whole kinematically-accessible range (e.g. $0 \leq p_{\rm T} \leq \infty$) and on which no cuts are applied.

These three numbers are the same (*up to statistics*, which here means the number of generated events – see the bottom of this section for further comments) for the processes listed in table 1. For the processes listed in table 2, on the other hand, the results of *b)* and *c)* are equal to that of *a)*, times the branching ratio(s) for the selected decay channel(s), times (in the case of top decays) other factors due to kinematic cuts specified in input (see below). This is so because for the processes of table 2 spin correlations are obtained as described in ref. [29]. For these processes, we shall denote in what follows the cross section obtained in *a)* as the undecayed cross section, and those obtained in *b)* or *c)* as the decayed cross sections. We note that, both for the processes in table 1 and for those in table 2, the

[7] This is the case for items *b)* and *c)* only if `WGTTYPE`=1.

results of *b)* and *c)* are equal to the sum of the weights of all events stored in the event file (possibly up to the contributions of those few events which HW6 is unable to shower and hadronize, and which are therefore discarded with error messages in the MC run).

The situation is basically identical in the case of HW++, except for the fact that the result of *b)* is to be found in the `*.out` file (in the running directory) rather than in the standard output, and it is given in *nb* rather than in *pb*.

For the processes of table 2, the branching ratios used in the computation are determined by the values of the branching ratios for individual decay channels. The following variables are relevant to top decays:

$$\texttt{BRTOPTOLEP} = \frac{\Gamma\left(\sum_j t \to l\nu_l b_j\right)}{\Gamma_t}\,, \qquad \texttt{BRTOPTOHAD} = \frac{\Gamma\left(\sum_{ij} t \to u\bar{d}_i b_j\right)}{\Gamma_t}\,, \tag{4.1}$$

with b_j and $\bar{d}_i$ any down-type quark and antiquark respectively, u an up-type quark, and l a charged lepton; lepton and flavour universality are assumed. In the case of W decays, one has the analogous variables

$$\texttt{BRWTOLEP} = \frac{\Gamma\left(W \to l\nu_l\right)}{\Gamma_W}\,, \qquad \texttt{BRWTOHAD} = \frac{\Gamma\left(\sum_i W \to u\bar{d}_i\right)}{\Gamma_W}\,. \tag{4.2}$$

Finally, in the case of Z decays one has

$$\texttt{BRZTOEE} = \frac{\Gamma\left(Z \to l\bar{l}\right)}{\Gamma_Z}\,. \tag{4.3}$$

The variables in eqs. (4.1)–(4.3) can either be given a numerical value in input, or computed at the LO in the SM by the code – see sect. 4.6 for details. The numerical values of these variables are then combined to obtain the overall branching ratio for the decay channels selected, which is done by setting the variables $\texttt{IL}_\alpha$ and `TOPDECAY` as explained in sect. 4.6 (see in particular table 3). For example, for a top decaying into a W and any down-type quarks, with the W decaying to an electron, muon, or any quarks, one sets $\texttt{IL}_\alpha$=6, `TOPDECAY=ALL`, and the resulting branching ratio will be $2 \times$ `BRTOPTOLEP`$+2 \times$ `BRTOPTOHAD`.

As mentioned above, in the case of top decays (as opposed to hard W decays in Wt or W^+W^- or $W^\pm Z$ production) the decayed cross section will include kinematic factors in addition to the branching ratios. These factors are due to the fact that in general the range for the invariant mass of the pair of particles emerging from the W decay (i.e. the virtuality of the W) does not coincide with the maximum that is kinematically allowed. For each top that decays, the following kinematic factor will therefore be included in the decayed cross section

$$\frac{\Gamma\left(t \to f f' b \,|\, q_{\rm W}(\inf), q_{\rm W}(\sup)\right)}{\Gamma\left(t \to f f' b \,|\, 0, m_t\right)}\,, \tag{4.4}$$

with

$$\Gamma\left(t \to f f' b \,|\, m, M\right) = \int_{m^2}^{M^2} dq^2_{\rm W} \frac{d\Gamma\left(t \to f f' b\right)}{dq^2_{\rm W}}\,, \tag{4.5}$$

and $q_{\rm W}(\rm inf)$, $q_{\rm W}(\rm sup)$ the lower and upper limits of the W virtuality, which can be chosen in input. In particular, if `V1GAMMAX`> 0, one will have

$$q_{\rm W}(\rm inf) = \texttt{WMASS} - \texttt{V1GAMMAX} \times \texttt{WWIDTH}\,, \qquad q_{\rm W}(\rm sup) = \texttt{WMASS} + \texttt{V1GAMMAX} \times \texttt{WWIDTH}\,. \tag{4.6}$$

On the other hand, if `V1GAMMAX`< 0, one has

$$q_{\rm W}(\rm inf) = \texttt{V1MASSINF}\,, \qquad q_{\rm W}(\rm sup) = \texttt{V1MASSSUP}\,. \tag{4.7}$$

The ranges in eqs. (4.6) or (4.7) apply to the W emerging from the decay of the top quark in $t\bar{t}$ production, and of the top or antitop in single-top production (all channels). The corresponding ranges for the W emerging from the decay of the antitop quark in $t\bar{t}$ production are identical to those above, except for the replacement of `V1` with `V2`.

The user is also allowed to generate events by fixing the virtuality of the W emerging from top/antitop decays equal to the W pole mass, by setting `xGAMMAX`$= 0$, with `x=V1,V2`. In such a case, the decayed cross section will be equal to the undecayed cross section, times the branching ratios, times a factor

$$\left.\frac{d\Gamma\left(t \to f f' b\right)}{dq^2_{\rm W}}\right|_{q^2_{\rm W}=M^2_{\rm W}}\,, \tag{4.8}$$

for each decaying top quark. The decayed cross section will have therefore to be interpreted as differential in the W virtuality squared (doubly differential in the case of $t\bar{t}$ production), and will be expressed in pb GeV^{-2} (or pb GeV^{-4} for $t\bar{t}$ production) units.

The branching ratios and kinematics factors for each decaying particles are multiplied to give a single number (always less than or equal to one), which is by definition the ratio of the decayed over the undecayed cross section. This number is printed out at the end of the NLO run (search for `Normalization factor due to decays` in the standard output).

We conclude this section by stressing that, while the result of *a)* is always computed with a typical relative precision of 10^{-4}, those of *b)* and *c)* depend on the number of events generated. Although it has been checked that, upon increasing the number of events generated, the results of *b)* and *c)* do approach that of *a)* (possibly times the branching ratios and kinematic factors), option *a)* is clearly preferred. As mentioned above, the decayed cross section of *b)* or *c)* can be obtained without any loss of accuracy by multiplying the undecayed cross section of *a)* by the normalization factor printed out by the code at the end of the NLO run.

4.4 Wt and $H^{\pm}t$ production

Owing to the interference with $t\bar{t}$ production, which occurs in the gg and $q\bar{q}$ partonic channels starting at the NLO, the Wt cross section is ill-defined beyond the leading order in QCD. One can still give an operative meaning to NLO Wt production, but must always be aware of the potential biases introduced in this way. This issue and its potential physics implications are discussed at length in ref. [26], which the reader is strongly advised to consult before generating Wt events.

Starting from MC@NLO version 3.4, we have implemented two different definitions of the Wt cross section, which we denoted by *diagram removal* and *diagram subtraction* in ref. [26]. The former computation is carried out by setting `WTTYPE=REMOVAL` in `MCatNLO.inputs`, while the latter corresponds to `WTTYPE=SUBTRACTION`. A practical application of these ideas to a phenomenological analysis is presented in ref. [34].

In Wt production, the factorization (renormalization) scale is assigned the value of the variable `PTVETO` (whose units are GeV) if `FFACT`< 0 (`FREN`< 0). This option should be used for testing purposes only; it is not recommended in the generation of event samples for experimental studies.

In MC@NLO version 4.0 we have implemented $H^{\pm}t$ production in a generic 2HDM model (see ref. [16]). When $m_{H^{\pm}} < m_t$ this process interferes with $t\bar{t}$ production, and the same considerations as for Wt production apply here. In order to avoid the proliferation of input variables, `WTTYPE` also controls which definition of the $H^{\pm}t$ cross section is used. When $m_{H^{\pm}} > m_t$ there is no interference with $t\bar{t}$ production, and therefore $H^{\pm}t$ production is well defined and does not require any special treatment at the level of matrix elements. In this mass range, the user must set `WTTYPE=REMOVAL` to run the code. Finally, we point out that there are four input variables that are specific to $H^{\pm}t$ production: `TYPEIORII`, `TANBETA`, `ACPL`, and `BCPL` – see sect. 5 for more details.

4.5 $W^{\pm}Z$ production and anomalous couplings

Starting from MC@NLO version 4.0, spin correlations have been added to $W^{\pm}Z$ production. Furthermore, the user can now generate this process either in the Standard Model, or with a Lagrangian with anomalous (i.e., non-SM) couplings, according to ref. [8]. The most general amplitude for this process can be written as follows:

$$\mathcal{A} = \mathcal{A}_0 + \Delta g_1^Z \mathcal{A}_{\Delta g_1^Z} + \Delta\kappa^Z \mathcal{A}_{\Delta\kappa^Z} + \lambda^Z \mathcal{A}_{\lambda^Z} \,, \tag{4.9}$$

with $\mathcal{A}_0$ the SM result. An event weight (i.e., the cross section) will therefore be:

$$\begin{aligned} w_{\rm TOT} \propto\; & w_0 + \left(\Delta g_1^Z\right)^2 w_1 + \left(\Delta\kappa^Z\right)^2 w_2 + \left(\lambda^Z\right)^2 w_3 \\ & + 2\Delta g_1^Z w_4 + 2\Delta\kappa^Z w_5 + 2\lambda^Z w_6 \\ & + 2\Delta g_1^Z \Delta\kappa^Z w_7 + 2\Delta g_1^Z \lambda^Z w_8 + 2\Delta\kappa^Z \lambda^Z w_9 \,. \end{aligned} \tag{4.10}$$

The values of Δg_1^Z, $\Delta\kappa^Z$, and λ^Z can be given in input using the script variables `DELG1Z`, `DELKAPZ`, and `LAMANZ` respectively. By setting these three variables equal to zero one recovers the SM result. A fourth script variable, `LAMFFAN`, corresponds to the quantity Λ introduced in eq. (8) of ref. [8], and serves the purpose of avoiding violations of unitarity; if it is set equal to zero in input, the program will re-set it to the default value of 2 TeV.

Regardless of the values of Δg_1^Z, $\Delta\kappa^Z$ and λ^Z given in input[8], MC@NLO can save the values of the weights w_i of eq. (4.10) in the event file; in order to do so, the user must set `CPLWGT=YES` in the inputs. The quantities w_i change event-by-event, and can be used to re-weight the cross section and to obtain predictions simultaneously for any number

[8] We stress that this includes the SM case, $\Delta g_1^Z = 0$, $\Delta\kappa^Z = 0$ and $\lambda^Z = 0$.

of combinations of anomalous couplings. In the case of HW6, the information on w_i is available in the MC run through the common block

```
DOUBLE PRECISION WGTACP(10)
COMMON/CWGTACP/WGTACP
```

with `WGTACP(I+1)`$= w_I/w_{\rm TOT}$. In the case of HW++, the information is available in the event file right before the `</event>` tag, in the following form:

`#an_cpl_wgt` $w_0/w_{\rm TOT}$ ... $w_9/w_{\rm TOT}$

4.6 Decays

MC@NLO is intended primarily for the study of NLO corrections to production cross sections and distributions; NLO corrections to the decays of produced particles are not included. As for spin correlations, the situation in version 4.0 is summarized in tables 1 and 2: they are included for all processes except ZZ production[9]. For the latter processes, quantities sensitive to the polarization of produced particles are not given correctly even to leading order. For such quantities, it may be preferable to use the standard HERWIG MCs, which do include leading-order spin correlations.

Following HW6 conventions, spin correlations in single-vector-boson processes are automatically included using the process codes (`IPROC`) relevant to lepton pair production (in other words, if one is interested in including spin correlations in e.g. W^+ production and subsequent decays into $\mu^+\nu_\mu$, one needs to use `IPROC`$=-1461$ rather than `IPROC`$=-1497$ and `MODBOS`$(1) = 3$). In order to avoid an unnecessary proliferation of `IPROC` values, this strategy has not been adopted in other cases ($t\bar{t}$, single-t, $H^0W^\pm$, H^0Z, W^+W^-, $W^\pm Z$), in which spin correlations are included if the variables $\texttt{IL}_1$ and $\texttt{IL}_2$ (the latter is used only in $t\bar{t}$, Wt, W^+W^-, and $W^\pm Z$ production) are assigned certain values. In the case of individual lepton decays, these range from 1 to 3 if the decaying particle is a W or a top, or from 1 to 6 if the decaying particle is a Z. For these cases, the value of $\texttt{IL}_\alpha$ fully determines the identity of the leptons emerging from the decay, and the same convention as in HW6 is adopted (see the HW6 manual and sect. 1.2).

In $t\bar{t}$ and single-top production, i.e. for all processes listed in table 2 which involve t and/or $\bar{t}$, the top quark and/or antiquark, and the hard W in the case of Wt production, can also decay hadronically. In such cases, therefore, the variables $\texttt{IL}_\alpha$ can be assigned more values than for the other processes; the situation is summarized in table 3. When generating the decays, lepton and flavour universalities are assumed. The relative probabilities of individual hadronic decays (e.g. $W^+ \to u\bar{d}$ vs $W^+ \to u\bar{s}$) are determined using the CKM matrix elements entered by the user (variables `Vud` in `MCatNLO.inputs`). The relative probabilities of leptonic vs hadronic decays are on the other hand determined using the values of the corresponding branching ratios entered by the user: variables `BRTOPTOLEP` and `BRTOPTOHAD` for top/antitop decays, and `BRWTOLEP` and `BRWTOHAD` for the decays of the hard W emerging from the hard process in Wt production[10] – see eqs. (4.1) and (4.2) for

[9] Non-factorizable spin correlations of virtual origin are not included in W^+W^-, $t\bar{t}$, and single-t production. See ref. [29].

[10] `BRWTOLEP` is also used in W^+W^- production. W hadronic decays are not implemented in this process, hence the branching ratio is only used as a rescaling factor for event weights.

the definitions of these variables.

$\texttt{IL}_\alpha$	W decays	Z decays
0	$e+\mu+\tau+q$	$\times$
1	e	e^+e^-
2	μ	$\nu_e\bar{\nu}_e$
3	τ	$\mu^+\mu^-$
4	$e+\mu$	$\nu_\mu\bar{\nu}_\mu$
5	q	$\tau^+\tau^-$
6	$e+\mu+q$	$\nu_\tau\bar{\nu}_\tau$
7	no decay	no decay

Table 3: Decays of the W's (produced in the hard process, or originating from top/antitop decays), and of the Z's. The symbol q denotes all hadronic W decays. Values different from 1, 2, or 3 for W decays are only allowed in $t\bar{t}$ and single-top production (all channels).

In the case of top/antitop decays, it is also possible to generate events in which the top decays into a W and any down-type quark (hence the notations b_α and $\bar{b}_\alpha$ in table 2). The identity of the latter is determined according to the CKM matrix values. For this to happen, one needs to set `TOPDECAY=ALL` in `MCatNLO.inputs`. If, on the other hand, one wants to always generate $t \to Wb$ decays, one needs to set `TOPDECAY=Wb`; in such a case, event weights (and thus the decayed cross section, as defined in sect. 4.3) will be multiplied by a factor $V_{tb}^2/(V_{td}^2+V_{ts}^2+V_{tb}^2)$.

In MC@NLO version 4.0, spin correlations in leptonic processes involving intermediate Z bosons are always included[11] except in the case of ZZ production. These processes are: dilepton production (`IPROC`$= -1350-$`IL`), H^0Z production (`IPROC`$= -2700-$`ID`), W^+Z production (`IPROC`$= -2870$), and W^-Z production (`IPROC`$= -2880$). In the case of dilepton and of H^0Z production, the Z may decay into charged-lepton or neutrino pairs. The identities of the decay products are determined by setting `IL` (for dilepton production) or $\texttt{IL}_1$ (for H^0Z production) according to the values given in table 3. In the case of $W^\pm Z$ production, only Z decays into charged-lepton pairs are implemented, with the identities of the leptons determined by setting the variable $\texttt{IL}_2$ equal to 1, 3, or 5 (see table 3).

For the processes in table 2 it is also possible to force the code to use the LO values of the relevant leptonic and hadronic branching ratios, by entering negative values for the top, W, and Z widths (variables `TWIDTH`, `WWIDTH`, and `ZWIDTH` in `MCatNLO.inputs`). In such a case, the values of `BRTOPTOLEP`, `BRTOPTOHAD`, `BRWTOLEP`, `BRWTOHAD`, `BRZTOEE` given in the input file will be ignored, and replaced by their LO values (which are equal to 1/9 for top and W leptonic decays, and equal to 1/3 for top and W hadronic decays). Likewise, the top, W, and Z widths will be computed using the LO SM formulae.

Spin correlations are implemented in the processes in table 2 according to the method of ref. [29], which is based on a zero-width approximation for the decaying particles. Neverthe-

[11] In other words, hadronic Z decays may be simulated by the EvG, but spin correlations are not included in such cases.

less, the top quark and antiquark in $t\bar{t}$ production (IPROC$=-1706$), and the vector bosons in W^+W^- (IPROC$=-2850$) and $W^\pm Z$ (IPROC$=-2870/2880$) production can be given masses different from the pole masses. These off-shell effects are modeled by re-weighting the cross section with skewed Breit-Wigner functions (in order to take into account the fact that by changing the invariant mass of the system produced one probes different values of Bjorken x's). This re-weighting is unitary, i.e. it does not change the inclusive cross section. For $t\bar{t}$ production, the ranges of top and antitop masses are controlled by the parameters TiGAMMAX, TiMASSINF, and TiMASSSUP (with i=1,2 for top and antitop respectively). For W^+W^- and $W^\pm Z$ production, one needs to use instead ViGAMMAX, ViMASSINF, and ViMASSSUP, with i=1,2 for W^+ and W^-, and for $W^\pm$ and Z respectively. In both cases, the mass ranges will be defined by formulae formally identical to those of eqs. (4.6) and (4.7). In version 4.0, off-shell effects are not implemented in the other processes in table 2, i.e. all channels of single-top production.

Finally, we point out that since spin correlations for the processes in table 2 are implemented according to the method of ref. [29], tree-level matrix elements for leptonic final states are needed. The codes for these have been generated with MadGraph/MadEvent [35, 36], and embedded into the MC@NLO package.

When IL$_\alpha$=7, the corresponding particle is left undecayed by the NLO code, and is passed as such to the MC code; the information on spin correlations is lost. However, the user can still force particular decay modes during the MC run. When the chosen EvG is HW6, in the case of vector bosons, one proceeds in the same way as in standard HW6, using the MODBOS variables – see sect. 3.4 of ref. [19]. However, top decays cannot be forced in this way because the decay is treated as a three-body process: the $W^\pm$ boson entry in HEPEVT is for information only. Instead, the top branching ratios can be altered using the HWMODK subroutine – see sect. 7 of ref. [19]. This is done separately for the t and $\bar{t}$. For example, `CALL HWMODK(6,1.D0,100,12,-11,5,0,0)` forces the decay $t \to \nu_e e^+ b$, while leaving $\bar{t}$ decays unaffected. Note that the order of the decay products is important for the decay matrix element (NME = 100) to be applied correctly. The relevant statements should be inserted in the HERWIG main program (corresponding to mcatnlo_hwdriver.f in this package) after the statement CALL HWUINC and before the loop over events. A separate run with `CALL HWMODK(-6,1.D0,100,-12,11,-5,0,0)` should be performed if one wishes to symmetrize the forcing of t and $\bar{t}$ decays, since calls to HWMODK from within the event loop do not produce the desired result. On the other hand, when HW++ is used, the relevant decay channels may be selected with suitable set instructions, that may be conveniently included in the HWPPInput.inputs file. In the case a specific decay channel should be switched on/off, a set /Herwig/Particles/ command should be added, e.g. set t->b,bbar,c;:OnOff Off to switch off the decay of top quarks into $b, \bar{b}, c$. For more details please consult the HW++ manual.

4.7 Results

As in the case of standard HERWIG the form of the results will be determined by the user's analysis routines. However, in addition to any files written by the user's analysis routines, the MC@NLO writes the following files:

♦ FPREFIXNLOinput: the input file for the NLO executable, created according to the set of input parameters defined in MCatNLO.inputs (where the user also sets the string FPREFIX). See table 4.
♦ FPREFIXNLO.log: the log file relevant to the NLO run.
♦ FPREFIXxxx.data: xxx can assume several different values. These are the data files created by the NLO code. They can be removed only if no further event generation step is foreseen with the current choice of parameters.
♦ FPREFIXMCinput: analogous to FPREFIXNLOinput, but for the MC executable. See table 6.
♦ FPREFIXMC.log: analogous to FPREFIXNLO.log, but for the MC run.
♦ FPREFIXMC.out: produced only in HW++ runs. Contains the result for the inclusive NLO rate (see sect. 4.3)
♦ EVPREFIX.events: the event file, where EVPREFIX is the string set by the user in MCatNLO.inputs.
♦ EVPREFIXxxx.events: xxx can assume several different values. These files are temporary event files, which are used by the NLO code, and eventually removed by the shell scripts. They MUST NOT be removed by the user during the run (the program will crash or give meaningless results).

In addition to the above one will get the following files when running with HW++:
♦ FPREFIXMC.run: This file is produced by the HW++ executable and is internally needed for the run. It is in simplified words the compiled FPREFIXMCinput file, resulting from the Herwig++ read FPREFIXMCinput command, which is executed in the executable script.
♦ FPREFIXMC.tex: Information, including references, describing the settings used in the HW++ run in LaTeX format.

By default, all the files produced by the MC@NLO are written in the running directory. However, if the variable SCRTCH (to be set in MCatNLO.inputs) is *not* blank, the data and event files will be written in the directory whose address is stored in SCRTCH (such a directory is not created by the scripts, and must already exist at run time).

5. Script variables

In the following, we list all the variables appearing in MCatNLO.inputs; these can be changed by the user to suit his/her needs. This must be done by editing MCatNLO.inputs. For fuller details see the comments in MCatNLO.inputs.

ECM The CM energy (in GeV) of the colliding particles.

FREN The ratio between the renormalization scale, and a reference mass scale.

FFACT As FREN, for the factorization scale.

HVQMASS The mass (in GeV) of the top quark, except when IPROC = −(1)1705, when it is the mass of the bottom quark. In this case, HVQMASS must coincide with BMASS.

xMASS The mass (in GeV) of the particle x, with x=HGG,W,Z,U,D,S,C,B,G.

`xWIDTH` The physical (Breit-Wigner) width (in GeV) of the particle `x`, with `x=HGG,W,Z,T` for H^0, $W^\pm$, Z, and t respectively.

`BRTOPTOx` Branching ratio for top decay channels $\sum_j t \to l\nu_l b_j$ (when `x=LEP`) and $\sum_{ij} t \to u\bar{d}_i b_j$ (when `x=HAD`). Lepton and flavour universality is assumed.

`BRWTOx` Branching ratio for W decay channels $W \to l\nu_l$ (when `x=LEP`) and $\sum_i W \to u\bar{d}_i$ (when `x=HAD`). Lepton and flavour universality is assumed.

`BRZTOEE` Branching ratio for Z decay channels $Z \to l\bar{l}$. Lepton universality is assumed.

`IBORNHGG` Valid entries are 1 and 2. If set to 1, the exact top mass dependence is retained *at the Born level* in Higgs production. If set to 2, the $m_t \to \infty$ limit is used.

`xGAMMAX` If `xGAMMAX` > 0, controls the width of the mass range for Higgs (`x=H`), vector bosons (`x=V1,V2`), and top (`x=T1,T2`): the range is `MASS`$\pm$(`GAMMAX`$\times$`WIDTH`). Off-shell effects for top are only implemented in $t\bar{t}$ production.

`xMASSINF` Lower limit of the Higgs (`x=H`), vector boson (`x=V1,V2`), and top (`x=T1,T2`) mass range; used only when `xGAMMAX` < 0.

`xMASSSUP` Upper limit of the Higgs (`x=H`), vector boson (`x=V1,V2`), and top (`x=T1,T2`) mass range; used only when `xGAMMAX` < 0.

`Vud` CKM matrix elements, with `u`=`U,C,T` and `d`=`D,S,B`. Set `VUD=VUS=VUB`=0 to use values of PDG2003.

`AEMRUN` Set it to `YES` to use running α_{em} in lepton pair and single vector boson production, set it to `NO` to use $\alpha_{em} = 1/137.0359895$.

`TYPEIORII` Set this variable equal to 1 or 2 in order to use a type-I or a type-II 2HDM model for the computation of $H^\pm t$ production.

`TANBETA` Set this variables equal to $\tan\beta$; effective only for $H^\pm t$ production in the context of a type-II 2HDM model.

`xCPL` Here, `x=A,B`. Set these variables equal to the A (scalar) and B (pseudoscalar) coefficients of the tHb vertex, in a type-I 2HDM model. Effective only for $H^\pm t$ production.

`IPROC` Process number that identifies the hard subprocess: see tables 1 and 2 for valid entries.

`IVCODE` Identifies the nature of the vector boson in associated Higgs production. It corresponds to variable `IV` of table 1.

`ILxCODE` Identify the nature of the particles emerging from vector boson or top decays. They correspond to variables $\texttt{IL}_1$ and $\texttt{IL}_2$ (for $\texttt{x} = 1, 2$ respectively) of tables 1, 2 and 3.

`TOPDECAY` Valid entries are `ALL` and `Wb`. Controls the type of top decay. See sect. 4.6.

`WTTYPE` Valid entries are `REMOVAL` and `SUBTRACTION`. Determines the definition of the Wt and $H^{\pm}t$ cross sections at the NLO. See sect. 4.4.

`PTVETO` Used in conjunction with `FFACT` and/or `FREN` to set mass scales in Wt production. See sect. 4.4.

`PARTn` The type of the incoming particle #`n`, with `n`=1,2. HW6 naming conventions are used (`P, PBAR, N, NBAR`).

`PDFGROUP` The name of the group fitting the parton densities used; the labeling conventions of PDFLIB are adopted. Unused when linked to LHAPDF.

`PDFSET` The number of the parton density set; according to PFDLIB conventions, the pair (`PDFGROUP`, `PDFSET`) identifies the densities for a given particle type. When linked to LHAPDF, use the numbering conventions of LHAGLUE [32].

`LAMBDAFIVE` The value of $\Lambda_{\rm QCD}$, for five flavours and in the $\overline{\rm MS}$ scheme, used in the computation of NLO cross sections. A negative entry sets $\Lambda_{\rm QCD}$ equal to that associated with the PDF set being used.

`LAMBDAHERW` The value of $\Lambda_{\rm QCD}$ used in MC runs; this parameter has the same meaning as $\Lambda_{\rm QCD}$ in HERWIG.

`SCHEMEOFPDF` The subtraction scheme in which the parton densities are defined.

`FPREFIX` Our integration routine creates files with name beginning by the string `FPREFIX`. Most of these files are not directly accessed by the user. See sects. 4.2 and 4.7.

`EVPREFIX` The name of the event file begins with this string. See sects. 4.2 and 4.7.

`EXEPREFIX` The names of the NLO and MC executables begin with this string; this is useful in the case of simultaneous runs.

`NEVENTS` The number of events stored in the event file, eventually processed by the EvG. See sect. 4.2 for comments relevant to HW++.

`MCMODE` Valid entries are `HW6` and `HWPP`, for using HW6 or HW++ in the MC run respectively.

`WGTTYPE` Valid entries are 0 and 1. When set to 0, the weights in the event file are ± 1. When set to 1, they are $\pm w$, with w a constant such that the sum of the weights gives the total inclusive NLO cross section (see sect. 4.3 for more details). Note that these weights are redefined by the EvG at MC run time according to its own convention (see HW6 or HW++ manual).

`RNDEVSEED` The seed for the random number generation in the event generation step; must be changed in order to obtain statistically-equivalent but different event files.

`BASES` Controls the integration step; valid entries are `ON` and `OFF`. At least one run with `BASES=ON` must be performed (see sect. 4.2).

`PDFLIBRARY` Valid entries are `PDFLIB`, `LHAPDF`, and `THISLIB`. In the former two cases, PDFLIB or LHAPDF is used to compute the parton densities, whereas in the latter case the densities are obtained from our self-contained PDF library.

`HERPDF` If set to `DEFAULT`, the EvG uses its internal PDF set (controlled by `NSTRU` in the case of HW6), regardless of the densities adopted at the NLO level. If set to `EXTPDF`, the EvG uses the same PDFs as the NLO code (see sect. 4.1).

`HWPATH` The physical address of the directory where the user's preferred version of HW6 is stored.

`HWPPPATH` The physical address of the directory where the user's preferred version of the HW++ tool is installed. Point to the base directory that holds the `lib, bin, include` and `share` directories. This is typically the directory one has specified in the `configure` step of the HW++ installation with the `--prefix` parameter.

`THEPEGPATH` The analogue of `HWPPPATH`, for the ThePEG library. Point to the base directory that holds the `lib, bin, include` and `share` directories. This is typically the directory you have specified in the `configure` step of the ThePEG installation with the `--prefix` parameter.

`SCRTCH` The physical address of the directory where the user wants to store the data and event files. If left blank, these files are stored in the running directory.

`HWUTI` This variables must be set equal to a list of object files, needed by the HW6 analysis routines of the user (for example, `HWUTI=''obj1.o obj2.o obj3.o''` is a valid assignment).

`HWPPANALYZER` This variables must be set equal to the name of the C++ analysis file relevant to HW++ runs (for example, `HWPPANALYZER=''TopAnalysis''` is a valid assignment, with `TopAnalysis.cc` and `TopAnalysis.h` being files in the `HWppAnalyzer` directory).

`HERWIGVER` This variable must to be set equal to the name of the object file corresponding to the version of HW6 linked to the package (for example, `HERWIGVER=herwig6520.o` is a valid assignment).

`PDFPATH` The physical address of the directory where the PDF grids are stored. Effective only if `PDFLIBRARY=THISLIB`.

`LHALINK` Set this variable equal to `STATIC` or `DYNAMIC` for linking with the static or dynamic LHAPDF library with HW6. This variable has no effect on how the HW++ executable is linked with LHAPDF.

`LHAPATH` Set this variable equal to the name of the directory where the local version of LHAPDF is installed. See sect. 4.1.1. This has no effect on which version of LHAPDF is used in the MC step when running HW++.

`LHAOFL` Set `LHAOFL=FREEZE` to freeze PDFs from LHAPDF at the boundaries, or equal to `EXTRAPOLATE` otherwise. See LHAPDF manual for details.

`EXTRALIBS` Set this variable equal to the names of the libraries which need be linked. LHAPDF is a special case, and must not be included in this list.

`EXTRAPATHS` Set this variable equal to the names of the directories where the libraries which need be linked are installed.

`INCLUDEPATHS` Set this variable equal to the names of the directories which contain header files possibly needed by C++ files provided by the user (also in HW6 runs, via `HWUTI`).

Acknowledgments

It is a pleasure to thank the co-authors of the MC@NLO papers, E. Laenen, P. Motylinski, and P. Nason, for having contributed so much to many different aspects of the MC@NLO project, and for stimulating discussions. We thank W. Verkerke for having provided us with a Fortran interface to C++ Root-calling routines. We thank F. Filthaut for having uncovered and fixed a bug in version 3.4. B.R. W. thanks the CERN theory group for frequent hospitality. The work of P. T. is supported by the Swiss National Science Foundation. C.D. W. is supported by the STFC Postdoctoral Fellowship "Collider Physics at the LHC". Finally, we are indebted with all the members of experimental collaborations, unfortunately too numerous to be explicitly mentioned here, who used the code and gave us precious suggestions and feedback.

Appendices

A. Version changes

A.1 From MC@NLO version 1.0 to version 2.0

In this appendix we list the changes that occurred in the package from version 1.0 to version 2.0.

• The Les Houches generic user process interface has been adopted.

• As a result, the convention for process codes has been changed: MC@NLO process codes `IPROC` are negative.

• The code `mcatnlo_hwhvvj.f`, which was specific to vector boson pair production in version 1.0, has been replaced by `mcatnlo_hwlhin.f`, which reads the event file according to the Les Houches prescription, and works for all the production processes implemented.

• The `Makefile` need not be edited, since the variables `HERWIGVER` and `HWUTI` have been moved to `MCatNLO.inputs` (where they must be set by the user).

• A code `mcatnlo_hbook.f` has been added to the list of utility codes. It contains a simplified version (written by M. Mangano) of HBOOK, and it is only used by the sample analysis routines `mcatnlo_hwan`*xxx*`.f`. As such, the user will not need it when linking to a self-contained analysis code.

We also remind the reader that the HW6 version must be 6.5 or higher since the Les Houches interface is used.

A.2 From MC@NLO version 2.0 to version 2.1

In this appendix we list the changes that occurred in the package from version 2.0 to version 2.1.

• Higgs production has been added, which implies new process-specific files (`mcatnlo_hgmain.f`, `mcatnlo_hgxsec.f`, `hgscblks.h`, `mcatnlo_hwanhgg.f`), and a modification to `mcatnlo_hwlhin.f`.

• Post-1999 PDF sets have been added to the MC@NLO PDF library.

• Script variables have been added to `MCatNLO.inputs`. Most of them are only relevant to Higgs production, and don't affect processes implemented in version 2.0. One of them (`LAMBDAHERW`) may affect all processes: in version 2.1, the variables `LAMBDAFIVE` and `LAMBDAHERW` are used to set the value of $\Lambda_{\rm QCD}$ in NLO and MC runs respectively, whereas in version 2.0 `LAMBDAFIVE` controlled both. The new setup is necessary since modern PDF sets have $\Lambda_{\rm QCD}$ values which are too large to be supported by HERWIG. (Recall that the effect of using `LAMBDAHERW` different from `LAMBDAFIVE` is beyond NLO.)

• The new script variable `PDFPATH` should be set equal to the name of the directory where the PDF grid files (which can be downloaded from the MC@NLO web page) are stored. At run time, when executing `runNLO`, or `runMC`, or `runMCatNLO`, logical links to these files will be created in the running directory (in version 2.0, this operation had to be performed by the user manually).

• Minor bugs corrected in `mcatnlo_hbook.f` and sample analysis routines.

A.3 From MC@NLO version 2.1 to version 2.2

In this appendix we list the changes that occurred in the package from version 2.1 to version 2.2.

• Single vector boson production has been added, which implies new process-specific files (`mcatnlo_sbmain.f`, `mcatnlo_sbxsec.f`, `svbcblks.h`, `mcatnlo_hwansvb.f`), and a modification to `mcatnlo_hwlhin.f`.

• The script variables `WWIDTH` and `ZWIDTH` have been added to `MCatNLO.inputs`. These denote the physical widths of the W and Z^0 bosons, used to generate the mass distributions of the vector bosons according to the Breit–Wigner function, in the case of single vector boson production (vector boson pair production is still implemented only in the zero-width approximation).

A.4 From MC@NLO version 2.2 to version 2.3

In this appendix we list the changes that occurred in the package from version 2.2 to version 2.3.

• Lepton pair production has been added, which implies new process-specific files (`mcatnlo_llmain.f`, `mcatnlo_llxsec.f`, `llpcblks.h`, `mcatnlo_hwanllp.f`), and modifications to `mcatnlo_hwlhin.f` and `mcatnlo_hwdriver.f`.

• The script variable `AEMRUN` has been added, since the computation of single vector boson and lepton pair cross sections is performed in the $\overline{\rm MS}$ scheme (the on-shell scheme was previously used for single vector boson production).

• The script variables `FRENMC` and `FFACTMC` have been eliminated.

• The structure of pseudo-random number generation in heavy flavour production has been changed, to avoid a correlation that affected the azimuthal angle distribution for the products of the hard partonic subprocesses.

• A few minor bugs have been corrected, which affected the rapidity of the vector bosons in single vector boson production (a 2–3% effect), and the assignment of $\Lambda_{\rm QCD}$ for the LO and NLO PDF sets of Alekhin.

A.5 From MC@NLO version 2.23 to version 3.1

In this appendix we list the changes that occurred in the package from version 2.3 to version 3.1.

• Associated Higgs production has been added, which implies new process-specific files (`mcatnlo_vhmain.f`, `mcatnlo_vhxsec.f`, `vhgcblks.h`, `mcatnlo_hwanvhg.f`), and modifications to `mcatnlo_hwlhin.f` and `mcatnlo_hwdriver.f`.

• Spin correlations in W^+W^- production and leptonic decay have been added; the relevant codes (`mcatnlo_vpmain.f`, `mcatnlo_vhxsec.f`) have been modified; the sample analysis routines (`mcatnlo_hwanvbp.f`) have also been changed. Tree-level matrix elements have been computed with MadGraph/MadEvent [35, 36], which uses HELAS [37]; the relevant routines and common blocks are included in `mcatnlo_helas2.f` and `MEcoupl.inc`.

• The format of the event file has changed in several respects, the most relevant of which is that the four-momenta are now given as (p_x, p_y, p_z, m) (up to version 2.3 we had (p_x, p_y, p_z, E)). Event files generated with version 2.3 or lower *must not be used* with version 3.1 or higher (the code will prevent the user from doing so).

• The script variables `GAMMAX`, `MASSINF`, and `MASSSUP` have been replaced with `xGAMMAX`, `xMASSINF` and `xMASSSUP`, with `x=H,V1,V2`.

• New script variables `IVCODE`, `IL1CODE`, and `IL2CODE` have been introduced.

• Minor changes have been made to the routines that put the partons on the HW6 mass shell for lepton pair, heavy quark, and vector boson pair production; effects are beyond the fourth digit.

• The default electroweak parameters have been changed for vector boson pair production, in order to make them consistent with those used in other processes. The cross sections are generally smaller in version 3.1 wrt previous versions, the dominant effect being the value of $\sin\theta_{\rm W}$: we have now $\sin^2\theta_{\rm W} = 0.2311$, in lower versions $\sin^2\theta_{\rm W} = 1 - m_W^2/m_Z^2$. The cross sections are inversely proportional to $\sin^4\theta_{\rm W}$.

A.6 From MC@NLO version 3.1 to version 3.2

In this appendix we list the changes that occurred in the package from version 3.1 to version 3.2.

• Single-t production has been added, which implies new process-specific files (`mcatnlo_stmain.f`, `mcatnlo_stxsec.f`, `stpcblks.h`, `mcatnlo_hwanstp.f`), and modifications to `mcatnlo_hwlhin.f` and `mcatnlo_hwdriver.f`.

• LHAPDF library is now supported, which implies modifications to all `*main.f` files, and two new utility codes, `mcatnlo_lhauti.f` and `mcatnlo_mlmtolha.f`.

• New script variables `Vud`, `LHAPATH`, and `LHAOFL` have been introduced.

• A bug affecting Higgs production has been fixed, which implies a modification to `mcatnlo_hgxsec.f`. Cross sections change with respect to version 3.1 *only if* `FFACT`$\neq 1$ (by $\mathcal{O}(1\%)$ in the range $1/2 \leq$ `FFACT` ≤ 2).

A.7 From MC@NLO version 3.2 to version 3.3

In this appendix we list the changes that occurred in the package from version 3.2 to version 3.3.

• Spin correlations have been added to $t\bar{t}$ and single-t production processes, which imply modifications to several codes (`mcatnlo_qqmain.f`, `mcatnlo_qqxsec.f`, `mcatnlo_stmain.f`, `mcatnlo_stxsec.f`, `mcatnlo_hwlhin.f` and `mcatnlo_hwdriver.f`). Tree-level matrix elements have been computed with MadGraph/MadEvent [35, 36].

• The matching between NLO matrix elements and parton shower is now smoother in Higgs production, which helps eliminate one unphysical feature in the $p_{\rm T}$ spectra of the accompanying jets. The code `mcatnlo_hgmain.f` has been modified. Technical details on this matching procedure will be posted on the MC@NLO web page.

• The new script variable `TWIDTH` has been introduced.

• All instances of `HWWARN('s',i,*n)` have been replaced with `HWWARN('s',i)` in HW6-related codes. This is consistent with the definition of `HWWARN` in HW6 versions 6.510 and higher; the user must be careful if linking to HW6 versions, in which the former form of `HWWARN` is used. Although HW6 6.510 compiles with `g95` or `gfortran`, MC@NLO 3.3 does not.

A.8 From MC@NLO version 3.3 to version 3.4

In this appendix we list the changes that occurred in the package from version 3.3 to version 3.4.

• Wt production has been implemented, which implies new process-specific codes (`mcatnlo_wtmain_dr.f`, `mcatnlo_wtmain_ds.f`, `mcatnlo_wtxsec_dr.f` and `mcatnlo_wtxsec_ds.f`).

• Owing to the implementation of Wt production and of top hadronic decays, the Les Houches interface (`mcatnlo_hwlhin.f`) and the driver (`mcatnlo_hwdriver.f`) have been upgraded.

• New script variables (`BRTOPTOx` and `BRWTOx`, with `x=LEP,HAD`; `yGAMMAX`, `yMASSINF` and `yMASSSUP` with `y=T1,T2`; `TOPDECAY`; `WTTYPE`; `PTVETO`) have been introduced.

• The new script variables `EXTRALIBS`, `EXTRAPATHS`, and `INCLUDEPATHS` can be used to link to external libraries. Their use has only been tested on a recent Scientific Linux release, and they may be not portable to other systems.

• The ranges of variables `ILxCODE` have been extended for several processes, in order to account for the newly-implemented hadronic decays.

• `MCatNLO.inputs` and `MCatNLO.Script` have been upgraded to reflect the changes above. A new sample input file (`MCatNLO_rb.inputs`) is included, which documents the use of an analysis producing plots in Root format. Finally, the possibility is given to link to a dynamic LHAPDF library (through `MCatNLO_dyn.Script` and `Makefile_dyn`).

• Front-end Fortran routines (`rbook_fe.f`) are provided, to produce plots in Root format, using the same syntax as for calling our HBOOK-type routines. A companion C++ code is needed (`rbook_be.cc`). These codes have been written by W. Verkerke. Examples of analysis routines using Root format have been added (`mcatnlo_hwan`*xxx*`_rb.f`). A call to a release-memory routine (`RCLOS`) has been added to `mcatnlo_hwdriver.f`; this is only needed when using a Root-format output, and a dummy body of `RCLOS` has been added to HBOOK-format analysis files `mcatnlo_hwan`*xxx*`.f`.

• The linking to LHAPDF has been upgraded, assuming the use of LHAPDF version 5.0 or higher. The file `mcatnlo_lhauti.f` has been eliminated, and replaced with `mcatnlo_utilhav4.f`, which is however necessary only if the user wants to link with LHAPDF versions 4.xx (in such a case, the user will also need to edit the Makefile).

• The automatic assignment of $\Lambda_{\rm QCD}$ when using LHAPDF is now to be considered robust. This implies changes to `mcatnlo_mlmtolha.f`, the insertion of a dummy routine into `mcatnlo_mlmtopdf.f` and `mcatnlo_pdftomlm.f`, and very minor changes to all `*main*.f` files.

• Minor changes to `mcatnlo_hbook.f`, mainly affecting two-dimensional plot outputs.

• A bug has been fixed, which prevented one from choosing properly the W mass ranges in W^+W^- production and subsequent decays in the case of `ViGAMMAX`< 0 (thanks to F. Filthaut).

• A bug has been fixed, which affected the computation of branching ratios in $t\bar{t}$ and single-top production; $\alpha_{em}(q^2)$ was previously called with argument m_{top} rather than m^2_{top}. This only affects event weights (i.e. not distributions), and is numerically very small.

• A bug in HERWIG versions 6.500 – 6.510 can lead to occasional violation of momentum conservation when the HW6 parameter `PRESPL=.FALSE.` (hard subprocess rapidity preserved), as is formally assumed in MC@NLO. Therefore at present we leave this parameter at its default value, `PRESPL=.TRUE.` (hard subprocess longitudinal momentum preserved). We have checked that this formal inconsistency has negligible actual consequences. The bug has been fixed in HW6 version 6.520. With older versions, the fix may be found on the Fortran HERWIG wiki at http://projects.hepforge.org/fherwig/trac/report (ticket 33). When this fix is implemented, the statement `PRESPL=.FALSE.` must be inserted in `mcatnlo_hwdriver.f` at the place indicated by the comments therein.

• It has been found that a simpler form for the MC subtraction terms with respect to that of eq. (B.43) of ref. [2] can be adopted; this form is now implemented in version 3.4. This change is relevant only to $Q\bar{Q}$ and single-top production, since for the other processes the new form and that of eq. (B.43) (which is implemented in MC@NLO version 3.3 or earlier) coincide. The differences between the two forms are equivalent to power-suppressed terms. This has been verified by comparing results obtained with version 3.4

for $t\bar{t}$ and single-top (s- and t-channel) production at the Tevatron and the LHC, and for $b\bar{b}$ production at the Tevatron, with analogous results obtained with version 3.3. On the other hand, $b\bar{b}$ production at the LHC does display large differences, owing to the fact that the old form of MC subtraction terms has a pathology which affects this process. Starting from version 3.4 $b\bar{b}$ production at the LHC may be considered safe. Technical details on the new form of the MC subtraction terms will be posted on the MC@NLO web page.

A.9 From MC@NLO version 3.4 to version 4.0

In this appendix we list the changes that occurred in the package from version 3.4 to subversions 3.41 and 3.42 to version 4.0.

• A problem was found which affected top decays in the processes listed in table 2 (except for $H^{\pm}t$, not implemented in version 3.4). This implied that the identities of top decay products in n event samples of k events each could have been statistically not equivalent to those of one single event sample of $n \times k$ events, for $k \simeq 5000$ or smaller. Fixed in subversion 3.41.

• All processes have been interfaced to HW++(except for $H^{\pm}t$ production). This implies a new structure of the source directory, and the addition of scripts (`MCatNLO_pp.Script`) and a makefile (`Makefile_pp`) specific to HW++.

• The linking to the static or dynamic LHAPDF library is now done via a shell variable. `MCatNLO_dyn.Script` and `Makefile_dyn` are thus obsolete, and have been removed from the package.

• $H^{\pm}t$ production has been implemented, including spin correlations.

• Spin correlations and anomalous couplings have been included in $W^{\pm}Z$ production.

• A numerical inaccuracy problem which affected the large-rapidity, large-$p_{\rm T}$ region of leptons in W production has been fixed in subversion 3.42. In addition, the convention on the range of lepton-parton azimuthal angular differences has been changed from $(0, \pi)$ to $(-\pi, \pi)$ in this process.

• The values of the fractions of the longitudinal momenta of the incoming partons x_1 and x_2, and that of the mass scale squared Q^2 (in GeV^2), used in the computations of the PDFs, are now stored in the event file.

B. Running the package without the shell scripts

In this appendix, we describe the actions that the user needs to take in order to run the package without using the shell scripts, and the `Makefile`. Examples are given for vector boson pair production, but only trivial modifications are necessary in order to treat other production processes.

B.1 Creating the executables

An MC@NLO run requires the creation of two executables, for the NLO and MC codes respectively. The files to link depend on whether one uses PDFLIB, LHAPDF, or the PDF library provided with this package; we list them below:

- **NLO with private PDFs:** `mcatnlo_vbmain.o mcatnlo_vbxsec.o mcatnlo_helas2.o mcatnlo_date.o mcatnlo_int.o mcatnlo_uxdate.o mcatnlo_uti.o mcatnlo_str.o mcatnlo_pdftomlm.o mcatnlo_libofpdf.o dummies.o SYSFILE`
- **NLO with PDFLIB:** `mcatnlo_vbmain.o mcatnlo_vbxsec.o mcatnlo_helas2.o mcatnlo_date.o mcatnlo_int.o mcatnlo_uxdate.o mcatnlo_uti.o mcatnlo_str.o mcatnlo_mlmtopdf.o dummies.o SYSFILE CERNLIB`
- **NLO with LHAPDF:** `mcatnlo_vbmain.o mcatnlo_vbxsec.o mcatnlo_helas2.o mcatnlo_date.o mcatnlo_int.o mcatnlo_uxdate.o mcatnlo_lhauti.o mcatnlo_str.o mcatnlo_mlmtolha.o dummies.o SYSFILE LHAPDF`
- **MC with private PDFs:** `mcatnlo_hwdriver.o mcatnlo_hwlhin.o mcatnlo_hwanvbp.o mcatnlo_hbook.o mcatnlo_str.o mcatnlo_pdftomlm.o mcatnlo_libofpdf.o dummies.o HWUTI HERWIGVER`
- **MC with PDFLIB:** `mcatnlo_hwdriver.o mcatnlo_hwlhin.o mcatnlo_hwanvbp.o mcatnlo_hbook.o mcatnlo_str.o mcatnlo_mlmtopdf.o dummies.o HWUTI HERWIGVER CERNLIB`
- **MC with LHAPDF:** `mcatnlo_hwdriver.o mcatnlo_hwlhin.o mcatnlo_hwanvbp.o mcatnlo_hbook.o mcatnlo_str.o mcatnlo_mlmtolha.o dummies.o HWUTI HERWIGVER LHAPDF`

The process-specific codes `mcatnlo_vbmain.o` and `mcatnlo_vbxsec.o` (for the NLO executable) and `mcatnlo_hwanvbp.o` (the HW6 analysis routines in the MC executable) need to be replaced by their analogues for other production processes.

The variable `SYSFILE` must be set either equal to `alpha.o`, or to `linux.o`, or to `sun.o`, according to the architecture of the machine on which the run is performed. For any other architecture, the user should provide a file corresponding to `alpha.f` etc., which he/she will easily obtain by modifying `alpha.f`. The variables `HWUTI` and `HERWIGVER` have been described in sect. 5. In order to create the object files eventually linked, static compilation is always recommended (for example, `g77 -Wall -fno-automatic` on Linux).

B.2 The input files

Here, we describe the inputs to be given to the NLO and MC executables in the case of vector boson pair production. The case of other production processes is completely analogous. When the shell scripts are used to run the MC@NLO, two files are created, `FPREFIXNLOinput` and `FPREFIXMCinput`, which are read by the NLO and MC executable respectively. We start by considering the inputs for the NLO executable, presented in table 4. The variables whose name is in uppercase characters have been described in sect. 5. The other variables are assigned by the shell script. Their default values are given in table 5. Users who run the package without the script should use the values given in table 5. The variable `zi` controls, to a certain extent, the number of negative-weight events generated by the MC@NLO (see ref. [1]). Therefore, the user may want to tune this

'FPREFIX'	! prefix for BASES files
'EVPREFIX'	! prefix for event files
ECM FFACT FREN FFACTMC FRENMC	! energy, scalefactors
IPROC	! -2850/60/70/80=WW/ZZ/ZW+/ZW-
WMASS ZMASS	! M_W, M_Z
UMASS DMASS SMASS CMASS BMASS GMASS	! quark and gluon masses
'PART1' 'PART2'	! hadron types
'PDFGROUP' PDFSET	! PDF group and id number
LAMBDAFIVE	! Lambda_5, <0 for default
'SCHEMEOFPDF'	! scheme
NEVENTS	! number of events
WGTTYPE	! 0 => wgt=+1/-1, 1 => wgt=+w/-w
RNDEVSEED	! seed for rnd numbers
zi	! zi
$\texttt{nitn}_1$ $\texttt{nitn}_2$	! itmx1,itmx2

Table 4: Sample input file for the NLO code (for vector boson pair production). FPREFIX and EVPREFIX must be understood with SCRTCH in front (see sect. 5).

Variable	Default value
zi	0.2
$\texttt{nitn}_i$	10/0 (BASES=ON/OFF)

Table 5: Default values for script-generated variables in FPREFIXNLOinput.

parameter in order to reduce as much as possible the number of negative-weight events. We stress that the MC code will not change this number; thus, the tuning can (and must) be done only by running the NLO code. The variables $\texttt{nitn}_i$ control the integration step (see sect. 4.2), which can be skipped by setting $\texttt{nitn}_i = 0$. If one needs to perform the integration step, we suggest setting these variables as indicated in table 5.

We now turn to the inputs for the MC executable, presented in table 6. The variables whose names are in uppercase characters have been described in sect. 5. The other variables are assigned by the shell script. Their default values are given in table 7. The user can freely change the values of esctype and pdftype; on the other hand, the value of beammom must always be equal to half of the hadronic CM energy.

When LHAPDF is linked, the value of PDFSET is sufficient to identify the parton density set. In such a case, PDFGROUP must be set in input equal to LHAPDF if the user wants to freeze the PDFs at the boundaries (defined as the ranges in which the fits have been performed). If one chooses to extrapolate the PDFs across the boundaries, one should set PDFGROUP=LHAEXT in input.

'EVPREFIX.events'	! event file
NEVENTS	! number of events
pdftype	! 0->Herwig PDFs, 1 otherwise
'PART1' 'PART2'	! hadron types
beammom beammom	! beam momenta
IPROC	! –2850/60/70/80=WW/ZZ/ZW+/ZW-
'PDFGROUP'	! PDF group (1)
PDFSET	! PDF id number (1)
'PDFGROUP'	! PDF group (2)
PDFSET	! PDF id number (2)
LAMBDAHERW	! Lambda_5, <0 for default
WMASS WMASS ZMASS	! M_W+, M_W-, M_Z
UMASS DMASS SMASS CMASS BMASS GMASS	! quark and gluon masses

Table 6: Sample input file for the MC code (for vector boson pair production), resulting from setting HERPDF=EXTPDF, which implies pdftype=1. Setting HERPDF=DEFAULT results in an analogous file, with pdftype=0, and without the lines concerning PDFGROUP and PDFSET. EVPREFIX must be understood with SCRTCH in front (see sect. 5). The negative sign of IPROC tells the EvG to use Les Houches interface routines.

Variable	Default value
esctype	0
pdftype	0/1 (HERPDF=DEFAULT/EXTPDF)
beammom	EMC/2

Table 7: Default values for script-generated variables in MCinput.

In the case of γ/Z, $W^{\pm}$, Higgs or heavy quark production, the MC executable can be run with the corresponding positive input process codes IPROC = 1350, 1399, 1499, 1600+ID, 1705, 1706, 2000–2008, 2600+ID or 2700+ID, to generate a standard HW6 run for comparison purposes[12]. Then the input event file will not be read: instead, parton configurations will be generated by HW6 according to the LO matrix elements.

References

[1] S. Frixione and B. R. Webber, "Matching NLO QCD computations and parton shower simulations," JHEP **0206** (2002) 029 [hep-ph/0204244].

[2] S. Frixione, P. Nason and B. R. Webber, "Matching NLO QCD and parton showers in heavy flavour production," JHEP **0308** (2003) 007 [arXiv:hep-ph/0305252].

[12] For vector boson pair production, for historical reasons, the different process codes 2800–2825 must be used.

[3] S. Frixione, Z. Kunszt and A. Signer, "Three-jet cross sections to next-to-leading order," Nucl. Phys. B **467** (1996) 399 [arXiv:hep-ph/9512328].

[4] S. Frixione, "A general approach to jet cross sections in QCD," Nucl. Phys. B **507** (1997) 295 [arXiv:hep-ph/9706545].

[5] B. Mele, P. Nason and G. Ridolfi, "QCD Radiative Corrections To Z Boson Pair Production In Hadronic Collisions," Nucl. Phys. B **357** (1991) 409.

[6] S. Frixione, P. Nason and G. Ridolfi, "Strong corrections to W Z production at hadron colliders," Nucl. Phys. B **383** (1992) 3.

[7] S. Frixione, "A Next-to-leading order calculation of the cross-section for the production of W+ W- pairs in hadronic collisions," Nucl. Phys. B **410** (1993) 280.

[8] L. J. Dixon, Z. Kunszt and A. Signer, "Vector boson pair production in hadronic collisions at order α_s : Lepton correlations and anomalous couplings", Phys. Rev. D **60** (1999) 114037 [arXiv:hep-ph/9907305].

[9] M. L. Mangano, P. Nason and G. Ridolfi, "Heavy quark correlations in hadron collisions at next-to-leading order," Nucl. Phys. B **373** (1992) 295.

[10] S. Dawson, "Radiative Corrections To Higgs Boson Production," Nucl. Phys. B **359** (1991) 283.

[11] A. Djouadi, M. Spira and P. M. Zerwas, "Production of Higgs bosons in proton colliders: QCD corrections," Phys. Lett. B **264** (1991) 440.

[12] G. Altarelli, R. K. Ellis and G. Martinelli, "Large Perturbative Corrections To The Drell-Yan Process In QCD," Nucl. Phys. B **157** (1979) 461.

[13] P. Aurenche and J. Lindfors, "QCD Corrections To Direct Lepton Production In Hadronic Collisions," Nucl. Phys. B **185** (1981) 274.

[14] V. Del Duca, S. Frixione and B. R. Webber, "MC@NLO for Higgs Boson Production," in preparation.

[15] B. W. Harris, E. Laenen, L. Phaf, Z. Sullivan and S. Weinzierl, "The fully differential single top quark cross section in next-to-leading order QCD," Phys. Rev. D **66** (2002) 054024 [arXiv:hep-ph/0207055].

[16] C. Weydert *et al.*, Eur. Phys. J. C **67** (2010) 617 [arXiv:0912.3430 [Unknown]].

[17] W. T. Giele, S. Keller and E. Laenen, "QCD Corrections to W Boson plus Heavy Quark Production at the Tevatron," Phys. Lett. B **372** (1996) 141 [arXiv:hep-ph/9511449].

[18] G. Marchesini, B. R. Webber, G. Abbiendi, I. G. Knowles, M. H. Seymour and L. Stanco, "HERWIG: A Monte Carlo event generator for simulating hadron emission reactions with interfering gluons. Version 5.1 - April 1991," Comput. Phys. Commun. **67** (1992) 465.

[19] G. Corcella, I.G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M.H. Seymour and B.R. Webber, "HERWIG 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes)," JHEP **0101** (2001) 010 [hep-ph/0011363].

[20] G. Corcella *et al.*, "HERWIG 6.5 release note," arXiv:hep-ph/0210213.

[21] M. Bahr *et al.*, "Herwig++ Physics and Manual," Eur. Phys. J. C **58** (2008) 639 [arXiv:0803.0883 [hep-ph]].

[22] M. Bahr *et al.*, "Herwig++ 2.3 Release Note," arXiv:0812.0529 [hep-ph].

[23] P. Torrielli and S. Frixione, "Matching NLO QCD computations with PYTHIA using MC@NLO", JHEP **1004** (2010) 110 [arXiv:1002.4293 [Unknown]].

[24] S. Frixione, F. Stoeckli, P. Torrielli, B. R. Webber, "NLO QCD corrections in Herwig++ with MC@NLO", preprint Cavendish–HEP–10/14, CERN-TH/2010-212.

[25] S. Frixione, E. Laenen, P. Motylinski and B. R. Webber, "Single-top production in MC@NLO," JHEP **0603** (2006) 092 [arXiv:hep-ph/0512250].

[26] S. Frixione, E. Laenen, P. Motylinski, B. R. Webber and C. D. White, "Single-top hadroproduction in association with a W boson," JHEP **0807** (2008) 029 [arXiv:0805.3067 [hep-ph]].

[27] S. Frixione, A Oh, "Anomalous couplings in WZ hadroproduction", in preparation.

[28] E. Boos *et al.*, "Generic user process interface for event generators," arXiv:hep-ph/0109068.

[29] S. Frixione, E. Laenen, P. Motylinski and B. R. Webber, "Angular correlations of lepton pairs from vector boson and top quark decays JHEP **0704** (2007) 081 [arXiv:hep-ph/0702198].

[30] W. Verkerke, Nikhef, unpublished.

[31] L. Lonnblad, "ThePEG, Pythia7, herwig++ and Ariadne", Nucl. Instrum. Meth. A **559** (2006) 246. Web page: `http://www.thep.lu.se/ThePEG`.

[32] M. R. Whalley, D. Bourilkov and R. C. Group, "The Les Houches accord PDFs (LHAPDF) and LHAGLUE," arXiv:hep-ph/0508110.

[33] S. Kawabata, "A New version of the multidimensional integration and event generation package BASES/SPRING," Comput. Phys. Commun. **88** (1995) 309.

[34] C. D. White, S. Frixione, E. Laenen and F. Maltoni, "Isolating Wt production at the LHC", JHEP **0911** (2009) 074 [arXiv:0908.0631 [hep-ph]].

[35] T. Stelzer and W. F. Long, "Automatic generation of tree level helicity amplitudes," Comput. Phys. Commun. **81** (1994) 357 [arXiv:hep-ph/9401258].

[36] F. Maltoni and T. Stelzer, "MadEvent: Automatic event generation with MadGraph," JHEP **0302** (2003) 027 [arXiv:hep-ph/0208156].

[37] H. Murayama, I. Watanabe and K. Hagiwara, "HELAS: HELicity amplitude subroutines for Feynman diagram evaluations," KEK-91-11