Preprint typeset in JHEP style - PAPER VERSION

CERN–PH–TH/2006–012
Cavendish–HEP–06/03
GEF–TH–1/2006

The MC@NLO 3.2 Event Generator*

Stefano Frixione

INFN, Sezione di Genova, Via Dodecaneso 33, 16146 Genova, Italy, and
PH Department, TH Unit, CERN, 1211 Geneva 23, Switzerland
E-mail: `Stefano.Frixione@cern.ch`

Bryan R. Webber

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

Abstract: This is the user's manual of MC@NLO 3.2. This package is a practical implementation, based upon the HERWIG event generator, 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, lepton pairs, and Higgs bosons in association with a W or Z. Spin correlations in decays are included for all processes except $t\bar{t}$, single-t, ZZ, and WZ production. This document is self-contained, but we emphasise the main differences with respect to previous versions.

Keywords: QCD, Monte Carlo, NLO Computations, Resummation, Hadronic Colliders.

*Work supported in part by the UK Particle Physics and Astronomy Research Council and by the EU Fourth Framework Programme 'Training and Mobility of Researchers', Network 'Quantum Chromodynamics and the Deep Structure of Elementary Particles', contract FMRX-CT98-0194 (DG 12 - MIHT).

Contents

1. Generalities

In this documentation file, we briefly describe how to run the MC@NLO package, implemented according to the formalism introduced in ref. [1]. When using MC@NLO, please cite refs. [1, 2]; if single-top events are generated, please also cite ref. [3]. The production processes now available are listed in table 1. The process codes `IPROC` will be explained below. $H_{1,2}$ represent hadrons (in practice, p or $\bar{p}$). The treatment of (undecayed) vector boson pair production within MC@NLO has been described in ref. [1], that of heavy quark pair production in ref. [2]. The NLO matrix elements for these processes have been taken from refs. [4, 5, 6, 7]. The information given in refs. [1, 2] allows the implementation in MC@NLO of any production process, provided that the formalism of refs. [8, 9] is used for the computation of cross sections to NLO accuracy. For specific information on single-top production, see ref. [3]. The matrix elements for Standard Model Higgs, single vector boson, lepton pair, associated Higgs, and single-top production have been taken from refs. [10, 11], ref. [12], ref. [13], ref. [14], and ref. [15] respectively.

This documentation refers to MC@NLO version 3.2 (previous versions 1.0, 2.0, 2.2, 2.3, and 3.1 are described in refs. [16, 17, 18, 19, 20] respectively). The new process implemented since version 3.1 is single-top production; furthermore, the use of parton density library LHAPDF [21, 22] is now supported. For precise details of version changes, see app. A.1-A.6.

1.1 Mode of operation

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 *event file* – see sect. 3.1); the event file is then read by HERWIG, which showers and hadronizes each hard configuration. Since version 2.0, the events are handled by the "Les Houches" generic user process interface [23] (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 the standard MC.

The signal to HERWIG 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 table 1. 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 HERWIG MC processes. Where possible, this convention will be adopted for additional MC@NLO processes. Consistently with what happens in standard HERWIG, by subtracting 10000 from `IPROC` one generates the same processes as in table 1, but eliminates the underlying event[1].

Higgs decays are controlled in the same way as in HERWIG, that is by adding `-ID` to the process code. The conventions for `ID` are the same as in HERWIG, namely $\texttt{ID} = 1\ldots6$ for $u\bar{u}\ldots t\bar{t}$; $7\ldots9$ for $e^+e^-\ldots\tau^+\tau^-$; $10, 11$ for W^+W^-, ZZ; and 12 for $\gamma\gamma$. Furthermore, $\texttt{ID} = 0$ gives quarks of all flavours, and $\texttt{ID} = 99$ gives all decays.

Process codes $\texttt{IPROC}{=}{-1360}{-}\texttt{IL}$ and $-1370{-}\texttt{IL}$ do not have an analogue in HERWIG; they are the same as $-1350{-}\texttt{IL}$, 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 HERWIG is adopted: $\texttt{IL}{=}1,\ldots,6$ for $l_{\rm IL} = e, \nu_e, \mu, \nu_\mu, \tau, \nu_\tau$ respectively. At variance with HERWIG, `IL` cannot be set equal to zero. Process codes $\texttt{IPROC}{=}{-1460}{-}\texttt{IL}$ and $-1470{-}\texttt{IL}$ are the analogue of HERWIG $1450{+}\texttt{IL}$; in HERWIG either W^+ or W^- can be produced, whereas MC@NLO treats the two vector bosons separately. For these processes, as in HERWIG, $\texttt{IL}{=}1, 2, 3$ for $l_{\rm IL} = e, \mu, \tau$, but again the choice $\texttt{IL} = 0$ is not allowed.

[1] The same effect can be achieved by setting the HERWIG parameter $\texttt{PRSOF} = 0$.

`IPROC`	`IV`	`IL`$_1$	`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				×	$H_1H_2 \to t\bar{t} + X$
–2000–`IC`				×	$H_1H_2 \to t/\bar{t} + X$
–2001–`IC`				×	$H_1H_2 \to \bar{t} + X$
–2004–`IC`				×	$H_1H_2 \to t + X$
–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	×	$H_1H_2 \to W^+W^- + X$
–2850		i	j	✓	$H_1H_2 \to (W^+ \to) l^+_i\nu_i(W^- \to) l^-_j\bar{\nu}_j + X$
–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: Processes implemented in MC@NLO 3.2. H^0 denotes the Standard Model Higgs boson and the value of `ID` controls its decay, as described in the HERWIG manual and below. The values of `IV`, `IL`, `IL`$_1$, and `IL`$_2$ control the identities of vector bosons and leptons, as described below. In single-t production, the value of `IC` controls the production processes, as described below. `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). Spin correlations in Higgs decays are included by HERWIG (e.g. in $H^0 \to W^+W^- \to l^+\nu l^-\bar{\nu}$).

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 according to phase space.

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 variable `MODBOS` (see sect. 3.2), but the relevant branching ratio will *not* be included automatically by MC@NLO. In the case of γ^* production, the branching ratios are $C_i q_i^2/(20/3)$, q_i being the electric charge (in units of the positron charge) of the fermion i selected through `MODBOS`, and $C_i = 1$ for leptons and 3 for quarks. Notice that $20/3 = \sum_i C_i q_i^2$, the sum including all leptons and quarks except the top. Thus, the total rate predicted by MC@NLO in the case of lepton pair production can also be recovered by multiplying the corresponding single vector boson total rate by the relevant branching ratio.

In NLO computations for single-top production, it is customary to distinguish between three production mechanisms, conventionally denoted as s channel, t channel, and Wt mode. In version 3.2, only the former two are implemented. The user can generate them either separately, by setting `IC`=10 and `IC`=20 for s- and t-channel production respectively, or together, by setting `IC`=0. For example, according to table 1, t-channel single-$\bar{t}$ events will be generated by entering `IPROC`=−2021.

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 HERWIG.

Furthermore, in the case of Higgs production in association with a W or Z, as well as vector boson pair production, the value of `IPROC` alone is not sufficient to fully determine the process type, and new variables `IV`, `IL`$_1$, and `IL`$_2$ have been introduced (see table 1). The variables `IL`$_1$ and `IL`$_2$ can take the same values as `IL` relevant to lepton pair production (notice, however, that in the latter case `IL` is not an independent variable, and its value is included via `IPROC`); in addition, `IL`$_\alpha$=7 implies that lepton spin correlations for the decay products of the corresponding vector boson are not taken into account, as indicated in table 1.

Apart from the above differences, MC@NLO and HERWIG *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 refs. [1, 2]); 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 HERWIG, augmented by the capability of reading the event file.

1.2 Package files

The package consists of the following files:

- **Shell utilities**
  `MCatNLO.Script`

MCatNLO.inputs
Makefile

- **Utility codes**
  MEcoupl.inc
  alpha.f
  dummies.f
  linux.f
  mcatnlo_date.f
  mcatnlo_hbook.f
  mcatnlo_helas2.f
  mcatnlo_hwdummy.f
  mcatnlo_int.f
  mcatnlo_lhauti.f
  mcatnlo_libofpdf.f
  mcatnlo_mlmtolha.f
  mcatnlo_mlmtopdf.f
  mcatnlo_pdftomlm.f
  mcatnlo_str.f
  mcatnlo_uti.f
  mcatnlo_uxdate.c
  sun.f
  trapfpe.c

- **General HERWIG routines**
  mcatnlo_hwdriver.f
  mcatnlo_hwlhin.f

- **Process-specific codes**
  mcatnlo_hwanbtm.f
  mcatnlo_hwanhgg.f
  mcatnlo_hwanllp.f
  mcatnlo_hwantop.f
  mcatnlo_hwanstp.f
  mcatnlo_hwansvb.f
  mcatnlo_hwanvbp.f
  mcatnlo_hwanvhg.f
  mcatnlo_hgmain.f
  mcatnlo_hgxsec.f
  mcatnlo_llmain.f
  mcatnlo_llxsec.f
  mcatnlo_qqmain.f
  mcatnlo_qqxsec.f
  mcatnlo_sbmain.f

```
mcatnlo_sbxsec.f
mcatnlo_stmain.f
mcatnlo_stxsec.f
mcatnlo_vbmain.f
mcatnlo_vbxsec.f
mcatnlo_vhmain.f
mcatnlo_vhxsec.f
hgscblks.h
hvqcblks.h
llpcblks.h
stpcblks.h
svbcblks.h
vhgcblks.h
```

These files can be downloaded from the web page:

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

The files `mcatnlo_hwan`*xxx*`.f`, which appear in the list of the process-specific codes, are sample HERWIG analysis routines. 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.

In addition to the files listed above, the user will need a version of the HERWIG code [24, 25, 26]. As stressed in ref. [1], for the 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 must be 6.500 or higher. On most systems, users will need to delete the dummy subroutines UPEVNT, UPINIT, PDFSET and STRUCTM from the standard HERWIG 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 version of HERWIG (currently 6.510 – with versions lower than 6.504 problems can be found in attempting to specify the decay modes of single vector bosons through the variable `MODBOS`. Also, crashes in the shower phase have been reported when using HERWIG 6.505, and we therefore recommend not to use that version).

1.3 Working environment

We have written a number of shell scripts and a `Makefile` (all listed under **Shell utilities** above) which will simplify the use of the package considerably. In order to use them, the computing system must support `bash` shell, and `gmake`[2]. 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.

1.4 Source and running directories

We assume that all the files of the package sit in the same directory, which we call the *source directory.* When creating the executable, our shell scripts determine the type of

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

operating system, and create a subdirectory of the source directory, which we call the *running directory*, whose name is `Alpha`, `Sun`, `Linux`, or `Darwin`, depending on the operating system. If the operating system is not known by our scripts, the name of the working directory is `Run`. 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. 2.1), or links to them, if the library of parton densities provided with the MC@NLO package is used.

2. Prior to running

Before running the code, the user needs to edit the following files:

`mcatnlo_hwan`*xxx*`.f`
`mcatnlo_hwdriver.f`
`mcatnlo_hwlhin.f`

We do not assume that the user will adopt the latest release of HERWIG (although, as explained above, it must be version 6.500 or higher). For this reason, the files `mcatnlo_hwdriver.f` and `mcatnlo_hwlhin.f` must be edited, in order to modify the `INCLUDE HERWIGXX.INC` command to correspond to the version of HERWIG the user is going to adopt. `mcatnlo_hwdriver.f` contains a set of read statements, which are necessary for the MC to get the input parameters (see sect. 3 for the input procedure); these read statements must not be modified or eliminated. Also, `mcatnlo_hwdriver.f` calls the HERWIG routines which perform showering, hadronization, decays (see sect. 3.2 for more details on this issue), and so forth; the user can freely modify this part, as customary in MC runs. Finally, the sample codes `mcatnlo_hwan`*xxx*`.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. 4).

2.1 Parton densities

Since the knowledge of the parton densities (PDF) is necessary in order to get the physical cross section, a PDF library must be linked. The possibility exists to link the (now obsolete) CERNLIB PDF library (PDFLIB), or its replacement LHAPDF; 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). 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 be also 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.

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 the MC is rather weak. So one may want to run the NLO and MC adopting a regular NLL-evolved set in the former case, and the default HERWIG set in the latter (the advantage is that this option reduces the amount of running time of the MC). 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 the MC to use the same PDF set as the NLO code.

Regardless of the PDFs used in the MC run, users must delete the dummy PDFLIB routines PDFSET and STRUCTM from HERWIG, as explained earlier.

3. Running

It is straightforward to run the MC@NLO. First, edit

```
MCatNLO.inputs
```

and write there all the input parameters (for the complete list of the input parameters, see sect. 4). 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. We point out that 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 to generate Z/γ^*, $W^\pm$, Higgs, $b\bar{b}$, $t\bar{t}$, single top, H^0W or H^0Z events with standard HERWIG (see the HERWIG manual for more details).

We stress that the input parameters are not solely related to physics (masses, CM energy, and so on); there are a few 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. 4.

Two such variables are `HERWIGVER` and `HWUTI`, which were moved in version 2.0 from the `Makefile` to `MCatNLO.inputs`. The former variable must be set equal to the object file name of the version of HERWIG currently adopted (matching the one whose common blocks are included in the files mentioned in sect. 2). The variable `HWUTI` must be set equal to the list of object files that the user needs in the analysis routines.

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.

3.1 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 [27].

We stress that the events stored in the event file just contain the partons involved in the hard suprocesses. 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 denotes 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. 3.3).

3.2 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 in decays, the situation in version 3.2 is summarized in table 1: they are included for all processes except $t\bar{t}$, single-t, ZZ, and WZ production[3].

[3]Non-factorizable spin correlations of virtual origin are not included in W^+W^- production.

For the latter processes, quantities sensitive to the polarisation of produced particles are not given correctly even to leading order. For such quantities, it may be preferable to use the standard HERWIG MC, which does include leading-order spin correlations.

Particular decay modes of vector bosons may be forced in MC@NLO in the same way as in standard HERWIG, using the `MODBOS` variables – see sect. 3.4 of ref. [25]. 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. [25]. 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.

3.3 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 2.

♦ `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 4.

♦ `FPREFIXMC.log`: analogous to `FPREFIXNLO.log`, but for the MC run.

♦ `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).

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).

4. 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 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`.

`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`) and vector bosons (`x=V1,V2`): the range is `MASS` $\pm$ (`GAMMAX` $\times$ `WIDTH`).

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

`xMASSSUP` Upper limit of the Higgs (`x=H`) or vector boson (`x=V1,V2`) 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. `Vud` is only used in single-t production.

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

`IPROC` Process number that identifies the hard subprocess: see table 1 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 leptons emerging from vector boson decays ($\mathtt{x} = 1, 2$). They correspond to variables $\mathtt{IL}_1$ and $\mathtt{IL}_2$ of table 1.

`PARTn` The type of the incoming particle #`n`, with `n`=1,2. HERWIG 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 [22].

`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.

`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`. These files are not directly accessed by the user; for more details, see sect. 3.1.

`EVPREFIX` The name of the event file begins with this string; for more details, see sect. 3.1.

`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 HERWIG .

`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 NLO cross section. N.B. These weights are redefined by HERWIG at MC run time according to its own convention (see HERWIG 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. 3.1).

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

`HERPDF` If set to `DEFAULT`, HERWIG uses its internal PDF set (controlled by `NSTRU`), regardless of the densities adopted at the NLO level. If set to `EXTPDF`, HERWIG uses the same PDFs as the NLO code (see sect. 2.1).

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

`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 analysis routines of the user (for example, `HWUTI=obj1.o obj2.o obj3.o` is a valid assignment).

HERWIGVER This variable must to be set equal to the name of the object file corresponding to the version of HERWIG linked to the package (for example, `HERWIGVER=herwig65.o` is a valid assignment).

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

LHAPATH Set this variable equal to the name of the directory where the local version of LHAPDF is installed.

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

Acknowledgement

Many thanks to P. Nason for contributions to the heavy quark code and valuable discussions on all aspects of the MC@NLO project, and to E. Laenen and P. Motylinski for the work done on single-t production. We also thank V. Drollinger and B. Quayle for testing a preliminary version of the W^+W^- code with spin correlations.

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 HERWIG 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 2.3 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 [28, 29], which uses HELAS [30]; 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 HERWIG 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).

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 HERWIG analysis routines in the MC executable) need to be replaced by their analogues for other production processes, which can be easily read from the list given in sect. 1.2.

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. 4. 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

In this appendix, 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 2. The variables whose name is in uppercase characters have been described in

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

sect. 4. The other variables are assigned by the shell script. Their default values are given in table 3. Users who run the package without the script should use the values given

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

Table 3: Default values for script-generated variables in `FPREFIXNLOinput`.

in table 3. 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 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. 3.1), 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 3.

`'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 4: 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. 4). The negative sign of `IPROC` tells HERWIG to use Les Houches interface routines.

We now turn to the inputs for the MC executable, presented in table 4. The variables whose names are in uppercase characters have been described in sect. 4. The other variables are assigned by the shell script. Their default values are given in table 5. The user can

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

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

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.

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 HERWIG run for comparison purposes[4]. Then the input event file will not be read: instead, parton configurations will be generated by HERWIG 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].

[3] S. Frixione, E. Laenen, P. Motylinski and B. R. Webber, "Single-top production in MC@NLO," arXiv:hep-ph/0512250.

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

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

[6] 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.

[7] 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.

[8] 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].

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

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

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

[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] S. Frixione and B. R. Webber, "The MC@NLO event generator," arXiv:hep-ph/0207182.

[17] S. Frixione and B. R. Webber, "The MC@NLO 2.0 event generator," arXiv:hep-ph/0307146.

[18] S. Frixione and B. R. Webber, "The MC@NLO 2.2 event generator," arXiv:hep-ph/0309186.

[19] S. Frixione and B. R. Webber, "The MC@NLO 2.3 event generator," arXiv:hep-ph/0402116.

[20] S. Frixione and B. R. Webber, "The MC@NLO 3.1 event generator," arXiv:hep-ph/0506182.

[21] W. Giele *et al.*, "The QCD/SM working group: Summary report," arXiv:hep-ph/0204316.

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

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

[24] 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.

[25] 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].

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

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

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

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

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