The POWHEG BOX user manual:
Single-top Wt-channel process (DR and DS)

Emanuele Re

Institute for Particle Physics Phenomenology, Department of Physics
University of Durham, Durham, DH1 3LE, UK
E-mail: `emanuele.re@durham.ac.uk`

ABSTRACT: This note documents the use of the package `POWHEG BOX` for the single-top Wt-channel production process. Results can be easily interfaced to shower Monte Carlo programs, in such a way that both NLO and shower accuracy are maintained. Two alternative prescriptions to quantify interference effects with $t\bar{t}$ are available. Please read carefully the manual before using this code.

KEYWORDS: POWHEG, Shower Monte Carlo, NLO.

Contents

1. Introduction

The `POWHEG BOX` program is a framework for implementing NLO calculations in Shower Monte Carlo programs according to the `POWHEG` method. An explanation of the method and a discussion of how the code is organized can be found in refs. [1, 2, 3]. The code is distributed according to the "MCNET GUIDELINES for Event Generator Authors and Users" and can be found at the web page

<div align="center">

`http://powhegbox.mib.infn.it`.

</div>

In the following we will focus on the implementation of single-top Wt-channel production, whose source files can be found in the `POWHEG-BOX/ST_wtch_DR` and `POWHEG-BOX/ST_wtch_DS` subdirectories.

This program is an implementation of the NLO cross section calculated in [4] in the `POWHEG` formalism of refs. [1, 2]. A detailed description of the implementation can be found in ref. [5]. Spin correlations of the top-quark decay products are included with a method analogous to the one described in [6], and the relevant matrix elements for the full decayed amplitudes were obtained using MadGraph [7]. To evaluate scalar integrals, the program needs to be linked with QCDloop [8], that can be downloaded from

<div align="center">

`http://qcdloop.fnal.gov`.

</div>

In this note we give all the necessary information to run the program. A quick description on how to install and properly link QCDloop is also given in the following.

2. Installation

Before describing the installation procedure, we want to recall that single-top Wt-channel production is not well defined beyond LO, since there are interference effects with $t\bar{t}$ production. To deal with this problem, we used an approach very similar to the one described in ref. [4]. A description of the problem and of the POWHEG implementation used in this code can be found in [5]. As described in the aforementioned references, the size of interference effects with $t\bar{t}$ (and all the related issues) can be quantified essentially by comparing results obtained with two different prescriptions: DR (*Diagram Removal*) and DS (*Diagram Subtraction*). Hence, the two directories.

In order to run the POWHEG BOX program, we recommend the reader to start from the POWHEG BOX user manual, which contains all the information and settings that are common between all subprocesses. In this note we focus on the settings and parameters specific to the single-top Wt-channel implementation.

In the following, we will describe how to run the DR code. Same considerations hold also for DS: in fact, the structure of the two codes (and the relevant files) is similar.[1] Explicit instructions will be given when there are relevant differences.

As anticipated before, the program needs to be linked with QCDloop: to this end, the Makefile variables VPATH and LIBSQCDLOOP have to be set properly.

2.1 Installation of QCDloop

The procedure to install QCDloop can be found in its webpage. Here we summarize it:

1. Download and decompress the QCDloop tarball.

2. Edit the QCDloop-1.9/ff/ffinit_mine.f file by setting the variable 'path' from its current value to the absolute path of the folder where ff will be located. Therefore, the user needs to change the line 786 from

   ```
   path = '/Users/ellis/QCDLoop-1.9/ff/'
   ```

   to something like

   ```
   path = '/home/username/path_to_QCDLoop-1.9/QCDLoop-1.9/ff/'
   ```

3. Compile QCDloop by typing make in the QCDLoop-1.9 directory. If the files libqcdloop.a and libff.a are now present in the directories ql and ff, the QCDloop library has been compiled and installed properly. Compiling QCDloop with the same compiler used for POWHEG may be necessary in order to run the two programs together successfully. In this case, remember to change the compiler line in the QCDloop Makefile (the default is g77).

---

[1]We recall that in the DS case a slightly modified version of the file sigreal.f has to be used with respect to the same file present in the main folder. The Makefile is build such that the proper file will be compiled and linked.

3. Generation of events and showering

After having downloaded and placed the QCDloop library properly, the executable is built with the following commands

```
$ cd POWHEG-BOX/ST_wtch_DR
$ make pwhg_main
```

In the `testrun` folder, there are several examples of input files. For example, you can start a run doing

```
$ cd testrun
$ ../pwhg_main
```

The input file read in this case is `powheg.input` and at the end a file named `pwgevents.lhe` will contain 100000 events for tW^- production at the LHC, in the Les Houches format. To shower them with `PYTHIA` do

```
$ cd POWHEG-BOX/ST_wtch_DR
$ make main-PYTHIA-lhef
$ cd testrun
$ ../main-PYTHIA-lhef
```

Similar commands will run the `HERWIG` shower.

4. Process specific input parameters

```
facscfact 1 !  factorization scale factor:  mufact=muref*facscfact
renscfact 1 !  renormalization scale factor:  muren=muref*renscfact
```

Factorization and renormalization scale factors appearing here have to do with the computation of the inclusive cross section (i.e. the $\bar{B}$ function [1, 2, 3]), and can be varied by a factor of order 1 to study scale dependence. The natural choice for this process is the mass of the top-quark. We choose to perform the NLO calculation keeping these scales fixed. The experienced user can change this setting modifying the `set_fac_ren_scales` routine.

It follows a description of parameters which are relevant for this production process:

- As discussed in ref. [5] when the DS procedure is used, negative weights can appear. To keep track of them properly, in the input files for DS we decided to activate the `withnegweights` flag. The fraction of negative-weighted events can be reduced by setting the *folding* variables properly, as discussed in the next item.

- If the fraction of negative weights is large, one may increase `foldcsi`, `foldy`, `foldphi`. Allowed values are 1, 2, 5, 10, 25, 50. The speed of the program is inversely propor-

tional to the product of these numbers, so that a reasonable compromise should be found. Our experiences tell us that, even at LHC energies, the fraction of negative weights in $\bar{B}$ calculation is such that the numbers provided in the examples need not to be changed. In particular, in the DS case, it is recommended to leave the default foldings on the csi and y variables, in order to generate an event sample with a low number of negative-weighted events.

- In the Wt-channel case, it is not needed to activate the `withdamp` option. Therefore, this token is set to zero. The same setting can be obtained by commenting or deleting the corresponding line, which was left as a reminder.

Other parameters are those specifically related to the single-top Wt-channel processes: from revision 1.0, some of these parameters are mandatory (the program stops if they are missing), other are optional (default values are assigned in `init_couplings.f`, but are overwritten if the token is found uncommented in the input file, as in previous versions).

For the production step, the relevant parameters are:

```
!  mandatory production parameters
ttype 1         !  1 for t, -1 for tbar
topmass 175.0   !  top mass
```

where the value of `ttype` is used to decide if top or antitop quarks will be produced and `topmass` set the top-quark mass.

In the current released version, top-quark decay products are always generated by POWHEG, accordingly to a procedure very similar to the one of ref. [6]. Therefore, the following parameters are mandatory too:

```
!  mandatory parameters used in decay generation
topdecaymode 10000 !  decay mode:  the 5 digits correspond to the following
                   !  top-decay channels (l,mu,tau,u,c)
                   !  0 means close, 1 open
wdecaymode 10000   !  decay mode:  the 5 digits correspond to the following
                   !  primary-w-decay channels (l,mu,tau,u,c)
                   !  0 means close, 1 open
tdec/elbranching 0.108 !  W electronic branching fraction
```

where the value of the `topdecaymode` token is formed by five digits, each representing the maximum number of the following particles at the (parton level) decay of the t ($\bar{t}$) quark: $e^\pm$, $\mu^\pm$, $\tau^\pm$, $\overset{(-)}{u}$, $\overset{(-)}{c}$. Thus, for example, 10000 means $t \to e^+ \nu_e b$, 11100 means all semileptonic decays, 00011 means fully hadronic. The same syntax has to be used to set the `wdecaymode` token, which controls the decay of the W-boson produced in the hard process (i.e. not the W-boson present in the decay chain of the top quark).

At the end of the event generation, just before writing the partonic event on the LHEF, a reshuffling procedure is called, in order to put outgoing charged leptons and quarks on

mass shell. In revisions before r213, for Wt-channel these particles were left massless. To reshuffle momenta, default values are used for masses, unless the entries `lhfm/***mass` are present in the input card.

The optional parameters are listed below. Their meaning is self-explanatory. We remind that it is not allowed to set any entry of the CKM matrix exactly equal to zero.

```
!  optional production parameters
!  (defaults defined in init_couplings.f)
#wmass 80.4               !  w mass
#sthw2 0.23113            !  (sin(theta_W))**2
#alphaem_inv 127.011989   !  1/alphaem
#CKM_Vud 0.9740           !  CKM matrix entries ...
#CKM_Vus 0.2225
#CKM_Vub 0.000001
#CKM_Vcd 0.2225
#CKM_Vcs 0.9740
#CKM_Vcb 0.000001
#CKM_Vtd 0.000001
#CKM_Vts 0.000001
#CKM_Vtb 1.0


!  optional parameters used in decay generation
!  (defaults defined in init_couplings.f)
#topwidth 1.7         !  top width
#wwidth 2.141         !  w width
#lhfm/cmass 1.5       !  c mass
#lhfm/bmass 5.0       !  b mass
#lhfm/emass 0.000511  !  e mass
#lhfm/mumass 0.1056   !  mu mass
#lhfm/taumass 1.777   !  tau mass
```

5. Generation of a sample with t and $\bar{t}$ events

The user can be interested in the generation of a sample where both top and antitop events appear. To this purpose, a script and a dedicated executable have been included. The script is named `merge_ttb.sh` and can be found in the directory `testrun`. It can be run in any subfolder of `ST_wtch_DR`. Three inputs are mandatory: the first two are the prefixes of the input files used to generate t and $\bar{t}$ events. The third input has to be an integer and correspond to the total number of events that the final *merged* sample will contain. The script has to be run twice, using a positive integer value at the first call and its opposite afterward. Therefore, for example, to produce a sample of 10000 events at the LHC, starting from the input files `lhc_wt_t-powheg.input` and `lhc_wt_tb-powheg.input`,

the invocation lines should be as follows:

```
$ sh merge_ttb.sh lhc_wt_t lhc_wt_tb 10000
```

and then

```
$ sh merge_ttb.sh lhc_wt_t lhc_wt_tb -10000
```

Few remarks are needed:

- it is responsibility of the user to check that the 2 input files are equal. The `ttype` tokens have to be different, obviously.

- the two values of `numevts` are not really used: the program re-calculate the needed values as a function of the t and $\bar{t}$ cross sections and of the total number of events to be generated.

- the final event file is always named `t_tb_sample-events.lhe`. In the header section it also contains a copy of the two input files used to generate it, for cross-checking purposes

References

[1] P. Nason, "A new method for combining NLO QCD with shower Monte Carlo algorithms," JHEP **0411** (2004) 040 [arXiv:hep-ph/0409146].

[2] S. Frixione, P. Nason and C. Oleari, "Matching NLO QCD computations with Parton Shower simulations: the POWHEG method," JHEP **0711** (2007) 070 [arXiv:0709.2092 [hep-ph]].

[3] S. Alioli, P. Nason, C. Oleari and E. Re, "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX," JHEP **1006**, 043 (2010) [arXiv:1002.2581 [hep-ph]].

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

[5] E. Re, "Single-top Wt-channel production matched with parton showers using the POWHEG method," Eur. Phys. J. C **71**, 1547 (2011) [arXiv:1009.2450 [hep-ph]].

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

[7] J. Alwall *et al.*, "MadGraph/MadEvent v4: The New Web Generation," JHEP **0709**, 028 (2007) [arXiv:0706.2334 [hep-ph]].

[8] R. K. Ellis and G. Zanderighi, "Scalar one-loop integrals for QCD," JHEP **0802**, 002 (2008) [arXiv:0712.1851 [hep-ph]].