June 28, 2019
POWHEG-BOX-RES

Same-sign W-boson scattering at EW NLO accuracy in the POWHEG-BOX-RES: user manual

Mauro Chiesa

Universität Würzburg, Institut für Theoretische Physik und Astrophysik
Emil-Hilb-Weg 22, 97074 Würzburg, Germany
E-mail: Mauro.Chiesa@physik.uni-wuerzburg.de

Ansgar Denner

Universität Würzburg, Institut für Theoretische Physik und Astrophysik
Emil-Hilb-Weg 22, 97074 Würzburg, Germany
E-mail: Ansgar.Denner@physik.uni-wuerzburg.de

Jean-Nicolas Lang

Universität Zürich, Physik-Institut,
CH-8057 Zürich, Switzerland
E-mail: jlang@physik.uzh.ch

Mathieu Pellen

University of Cambridge, Cavendish Laboratory,
Cambridge CB3 0HE, United Kingdom
E-mail: mpellen@hep.phy.cam.ac.uk

Abstract: This note documents the use of the package `POWHEG-BOX-RES/vbs-ssww-nloew` for same-sign W-boson scattering processes at the LHC including NLO electroweak corrections. The generated LH-events can be easily interfaced to shower Monte Carlo programs, in such a way that both NLO and shower accuracy are maintained.

Keywords: POWHEG, Shower Monte Carlo, NLO, Electroweak.

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, 4]. The code is distributed according to the "MCNET GUIDELINES for Event Generator Authors and Users" and can be found at the web page

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

This program is an implementation of the NLO EW corrections to the same-sign W-boson scattering process at the LHC presented in Ref. [5]. Please cite the paper when you use the program. Though the calculation uses the full LO and NLO matrix elements for the process pp $\to l\nu l'\nu' jj$ with same-sign l and l' (at order $\mathcal{O}\left(\alpha^6\right)$ and $\mathcal{O}\left(\alpha^7\right)$, respectively), the code should only be used for typical VBS selection cuts. This limitation is related to the interface to shower Monte Carlo programs, which perform the shower evolution of the LH-events preserving the virtuality of the resonances that appear explicitly in the LH-event files. As the resonance structure of the events written on disk for s-channel processes corresponds to the one in Fig. 1 of Ref. [5], the event selection should suppress the other possible event topologies as in the case of VBS selection cuts. In order to run the `POWHEG BOX` program, we recommend 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 same-sign W-boson scattering implementation.

2. External libraries

The code relies on the matrix element provider `RECOLA` [6, 7], that makes use of the `COLLIER` [8] library for the evaluation of tensor and scalar one-loop integrals. The user should download the `recola2-collier-X.Y.Z` package from the web page `https://recola.hepforge.org/` and follow the installation instructions therein.

In addition, we recommend to use `LHAPDF` (`https://lhapdf.hepforge.org/`) [9] as well as `FastJet` (`http://fastjet.fr/`) [10]. These are by default used in our implementation but can be switched off if needed.

We provide an interface to `PYTHIA8` [11] (version 8.2XY) for the parton shower evolution and subsequent hadronization of the events.

In order to build the `pwhg_main` executable in `POWHEG-BOX-RES/vbs-ssww-nloew`, the following variables should be set in the `Makefile`:

- `RECOLALOCATION=PATH_TO_RECOLA2COLLIER/recola2-collier-X.Y.Z/recola2-X.Y.Z`
- `LHAPDF_CONFIG=PATH_TO_LHAPDF/bin/lhapdf-config`
- `FASTJET_CONFIG=PATH_TO_FASTJET/bin/fastjet-config`

For the compilation of the `main-PYTHIA82-lhef` executable the following variables should be defined in the `Makefile`:

- `HEPMCLOCATION=PATH_TO_HEPMC`
- `PYTHIA8LOCATION=PATH_TO_PYTHIA8`

3. Generation of events

Build the executable

```
$ cd POWHEG-BOX-RES/vbs-ssww-nloew
$ make pwhg_main
```

We provide a sample run folder (`testrun`) to test the event generation at very low statistics:

```
$ cd testrun
$ ./runpar.sh
```

The `runpar.sh` script should be edited to export the path to `Recola2`:

```
export LD_LIBRARY_PATH=PATH_TO_RECOLA2COLLIER/recola2-collier-X.Y.Z/recola2-X.Y.Z
```

At the end of the run there should be six event files (`pwgevents-000X.lhe`) containing ten events each. As many upper bound violations occur when running the code at

NLO EW accuracy, a correction factor has to be applied to the weights of the btilde and remnant events (ub_btilde_corr and ub_remn_corr, respectively): the runpar.sh script takes care of computing these correction factors after the event generation by running the phyton script POWHEG-BOX-RES/Scripts/FindReweightFromCounters.py on the pwgcounters-st4-XXXX.dat files. The correction factors are then written in the input files XXXX-powheg.input that should be used for the parton shower evolution of the events (see the description of the main-PYTHIA82-lhef executable below). In order to run the parton shower, compile the main-PYTHIA82-lhef executable and edit the runpar.sh script to set the variable DOPYTHIAPS to 1.

The second run folder (run) contains the files powheg.input-save and runpar.sh with all the parameters and settings used to produce the results presented in Refs. [5, 12].[1] As can be seen from the runpar.sh script, 25 cores are used for the preparation of the grids in stages 1–3 (we actually generated the grids on a single machine with 5 cores), while 2000 jobs are used for the event generation in stage 4 (this stage of the calculation was performed on a computing cluster). Since the 2000 instances of pwhg_main used in stage 4 are independent, the parallelization of the event generation can be easily adapted by the user.

We provide the executable main-PYTHIA82-lhef which processes the events in the file pwgevents.lhe and passes them as input to parton shower programs, performing the required vetoes and setting the required flags in a manner consistent with the aimed (NLO+PS) accuracy of the input events. The shower Monte Carlo program PYTHIA is used to perform the QCD and QED shower. The recommended PYTHIA options can be found in the function pythia_init in the file pythia82F77.cc: we refer to the online PYTHIA documentation for a description of the above mentioned options. To shower an event file proceed as follows:

- build the main-PYTHIA82-lhef executable
```
$ cd POWHEG-BOX-RES/vbs-ssww-nloew
$ make main-PYTHIA82-lhef
```
- export the relevant environmental variables
```
$ export PYTHIA8DATA=PATH_TO_pythia82XY/share/Pythia8/xmldoc/
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:PATH_TO_HEPMC/lib:PATH_TO_PYTHIA/lib/lib
```
- run the parton shower on the event file pwgevents-XXXX.lhe using the options in the file XXXX-powheg.input
```
$ cd testrun
```

[1] There is one difference: in the original article, the PDF NNPDF3.0QED has been used. This PDF does not have a lhaid identifier. Therefore we have put the lhapdf id of NNPDF3.0.

```
$ echo $I > input
$ echo pwgevents-XXXX.lhe >> input
$ ../main-PYTHIA82-lhef < input
```

where `$I` corresponds to index `XXXX` of the event file once the starting `0` characters have been removed. If the event file is generated at NLO EW accuracy, the input file `XXXX-powheg.input` must include the input parameters `ub_btilde_corr` and `ub_remn_corr` mentioned above.

4. Process specific input parameters

All the parameters and flags are set in the input file `powheg.input`. The mandatory parameters are those needed to select the final-state leptons originating from the vector bosons:

```
idvecbos 24 !  24 pp->W+W+jj (-24 pp->W+W+jj)

vdecaymode 1 !  1 W(-> e nu_e)W(-> mu nu_mu)
             !  2 W(-> e nu_e)W(-> e nu_e )
             !  3 W(-> mu nu_mu)W(-> mu nu_mu)
```

In addition to the mandatory parameters, the `POWHEG BOX` input allows for an easy setting of EW and run parameters, by explicitly adding the relevant lines to the input card. If one of the following entries is not present in the input card the reported default value is assumed. The values below correspond to the current default.

```
Wmass   80.357973609878d0          !  W pole mass in GeV
Wwidth  2.0842989982782d0          !  W pole width in GeV
Zmass   91.1534806191827d0         !  Z pole mass in GeV
Zwidth  2.4942663787728d0          !  Z pole width in GeV
gmu     0.11663787d-4              !  Fermi constant in GeV^-2
Hmass   125d0                      !  Higgs mass in GeV
Tmass   173.2d0                    !  top mass in GeV

runningscale 0     !  choice for ren and fac scales in Bbar integration
                   !  0:  fixed scale M_W
                   !  1:  running scale sqrt(ptj1*ptj2)
```

The user can implement different functional forms for the running scale editing the subroutine `set_fac_ren_scales` in the file `Born_phsp.f`.

```
fakevirt 0         !  0=off,1=on (default 0)
```

As the calculation of the virtual matrix elements is highly time-consuming, the user can replace them with the Born ones multiplied by $\alpha/4\pi$ by setting the flag `fakevirt` to 1 in the input file. This should only be done in stage 1 (iterations `xg1` and `xg2` in `runpar.sh`) and stage 3. Using `fakevirt` for stages 2 or 4 spoils the NLO accuracy of the calculation.

```
bornzerodamp 1     !  0=off,1=on (default 1)
```

Flag to split the event generation in btilde and remnant generation with improved efficiency (see POWHEG manual).

```
rad_ptsqmin_em 0.001d0**2     !  minimum pt of the POWHEG radiation
                              !  (default 0.001d0**2)
```

```
cmass_lhe 0d0     !  charm mass written in the LH events (default 0)
```

Note that quarks are massless in the calculation. The c mass can be used to reshuffle the event kinematics before the LH event is written on disk.

```
ubexcess_correct 1     !  store ub-correction factors in
                       !  pwg-counters-st4-XXXX.dat files (default 0)
```

Due to the large number on upper-bound violations at NLO, the flag `ubexcess_correct` should always be set to 1 when generating events at NLO EW accuracy. As described above, the correction factors `ub_btilde_corr` and `ub_remn_corr` are computed by the script `FindReweightFromCounters.py` and should be added to the input file when running the PYTHIA interface `main-PYTHIA82-lhef`.

4.1 General input flags

The flags below are common to all the POWHEG process-folders. We list them for completeness.

```
ih1 1           !  hadron 1 (1 for protons, -1 for antiprotons)
ih2 1           !  hadron 2 (1 for protons, -1 for antiprotons)
ndns1 131       !  pdf set for hadron 1 (native mlm numbering)
ndns2 131       !  pdf set for hadron 2 (native mlm numbering)
lhans1 260000   !  pdf set for hadron 1 (LHAPDF LHA numbering)
lhans2 260000   !  pdf set for hadron 2 (LHAPDF LHA numbering)
ebeam1 6500d0   !  energy of beam 1 (GeV)
ebeam2 6500d0   !  energy of beam 2 (GeV)
ncall1 10000    !st1 number of calls for initializing the integration grid
itmx1 2         !st1 number of iterations for initializing the integration grid
ncall2 1000     !st2 number of calls for computing the Bbar integral
itmx2 2         !st2 number of iterations for computing the Bbar integral
nubound 1000    !st3 number of bbarra calls to setup norm of upper bounding function
numevts 10      !st4 number of events to be generated
```

4.2 Flags used by the shower interfaces

The flags below do not affect the event generation (stages 1–4). They are only used by the `PYTHIA` interface `main-PYTHIA82-lhef`.

```
ub_btilde_corr 1      !  ub-corr factor for btilde events (default 1)
ub_remn_corr 1        !  ub-corr factor for remnant events (default 1)
```

See the description of the flag `ubexcess_correct`.

```
use-scalup-ptj 4      !  starting scale for QCD radiation in PS (default 4)
                      !  0=standard powheg scalup
                      !  1=hardest quark pt (ptj1)
                      !  2=softest quark pt (ptj2)
                      !  3=(ptj1+ptj2)/2
                      !  4=sqrt(ptj1*ptj2)
```

Note that the scale is computed from the kinematics of the LH event generated by `POWHEG`.

```
SI_maxshowerevents -1 !  Number of events to read from the .lhe
                      !  (default: -1, all events)

SI_pytune -1          !  PYTHIA tune in -1,50 (default -1, default pp tune)
                      !  see PYTHIA documentation for the tune numbering

SI_nohad 0            !  if 1 switch off the hadronization (default 0)

SI_savehistos 1       !  if 1 save the results in a .top file (default 0)

SI_noQEDq 0           !  if 1 switch off the QED PS from quarks (default 0)

SI_qcdps 1            !  if 0 switch off QCD PS radiation (defaulf 1)
```

Note that `PYTHIA` gives an intrinsic p_{T} to the initial-state partons even for `SI_qcdps` equal to 0.

The decays of hadronic resonances which can proceed radiatively have been suppressed. In order to allow for such decays, the user should open the file `pythia82F77.cc` and comment the relevant lines in the function `pythia_init`.

Acknowledgements

We would liked to thank Paolo Nason for his help regarding POWHEG. MC, AD, and MP acknowledge financial support by the German Federal Ministry for Education and Research (BMBF) under contracts no. 05H15WWCA1 and 05H18WWCA1 and the German Research Foundation (DFG) under reference number DE 623/6-1. J.-N. Lang acknowledges support from the Swiss National Science Foundation (SNF) under contract

BSCGI0-157722. MP is supported by the European Research Council Consolidator Grant NNLOforLHC2.

References

[1] P. Nason, *A new method for combining NLO QCD with shower Monte Carlo algorithms*, *JHEP* **11** (2004) 040, [hep-ph/0409146].

[2] S. Frixione, P. Nason, and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, *JHEP* **11** (2007) 070, [arXiv:0709.2092].

[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* **06** (2010) 043, [arXiv:1002.2581].

[4] T. Ježo and P. Nason, *On the Treatment of Resonances in Next-to-Leading Order Calculations Matched to a Parton Shower*, *JHEP* **12** (2015) 065, [arXiv:1509.09071].

[5] M. Chiesa, A. Denner, J.-N. Lang, and M. Pellen, *An event generator for same-sign W-boson scattering at the LHC including electroweak corrections*, arXiv:1906.01863.

[6] S. Actis, A. Denner, L. Hofer, A. Scharf, and S. Uccirati, *Recursive generation of one-loop amplitudes in the Standard Model*, *JHEP* **04** (2013) 037, [arXiv:1211.6316].

[7] S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf, and S. Uccirati, *RECOLA: REcursive Computation of One-Loop Amplitudes*, *Comput. Phys. Commun.* **214** (2017) 140–173, [arXiv:1605.01090].

[8] A. Denner, S. Dittmaier, and L. Hofer, COLLIER*: a fortran-based Complex One-Loop LIbrary in Extended Regularizations*, *Comput. Phys. Commun.* **212** (2017) 220–238, [arXiv:1604.06792].

[9] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, *LHAPDF6: parton density access in the LHC precision era*, *Eur. Phys. J.* **C75** (2015) 132, [arXiv:1412.7420].

[10] M. Cacciari, G. P. Salam, and G. Soyez, *FastJet User Manual*, *Eur. Phys. J.* **C72** (2012) 1896, [arXiv:1111.6097].

[11] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159–177, [arXiv:1410.3012].

[12] B. Biedermann, A. Denner, and M. Pellen, *Large electroweak corrections to vector-boson scattering at the Large Hadron Collider*, *Phys. Rev. Lett.* **118** (2017) 261801, [arXiv:1611.02951].