Manual for electroweak W^+W^+jj production in the POWHEG-BOX-V2

The VBF_Wp_Wp program is an implementation of the electroweak W^+W^+jj production cross section within the POWHEG BOX framework. It complements the Wp_Wp_j_j program, which provides the QCD-induced contributions to W^+W^+jj production in hadronic collisions.

This document describes the input parameters that are specific to the implementation of the EW channel in the POWHEG-BOX-V2. Note that the recommended settings are very similar to those for the QCD production mode. The parameters that are common to all POWHEG-BOX-V2 implementations are described in the POWHEG-BOX-V2/Docs directory.

If you use this program, please quote Refs. [1–3].

Running the program

Download the POWHEG-BOX-V2, following the instructions at the web site

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

In addition, retrieve the VBF_Wp_Wp directory as explained, and go to the new directory via

```
$ cd POWHEG-BOX-V2/User-Processes-V2/VBF_Wp_Wp
```

Running is most conveniently done in a separate directory, for instance do

```
$ mkdir testrun
```

The directory must contain the powheg.input file and, for parallel running, a pwgseeds.dat file (see manual-BOX.pdf and Manyseeds.pdf).

Before compiling make sure that:

- fastjet is installed and fastjet-config is in the path,
- lhapdf is installed and lhapdf-config is in the path,
- gfortran, ifort or g77 is in the path, and the appropriate libraries are in the environment variable LD_LIBRARY_PATH.

If LHAPDF or fastjet are not installed, the code can still be run using a dummy analysis routine and built-in PDFs, see the Makefile in VBF_Wp_Wp.

After compiling, enter the testrun directory:

```
$ cd testrun
```

When executing

```
$../pwhg_main
```

the program will ask you to

```
enter which seed
```

The program requires you to enter an index that specifies the line number in the `pwgseeds.dat` file where the seed of the random number generator to be used for the run is stored. All results generated by the run will be stored in files named `*-[index].*`. When running on parallel CPUs, make sure that each parallel run has a different index.

The program can be run in several steps. Each new step requires the completion of the previous step.

Step 1

At this point the user has to decide whether the weak bosons are to be generated on-shell (`zerowidth = 1`) or off-shell, distributed according to a Breit-Wigner distribution (`zerowidth = 0`). By default on-shell weak bosons are generated. The user can also select the decay modes of the weak bosons by assigning appropriate PDG code to the parameters `vdecaymodew1` and `vdecaymodew2` in `powheg.input`. Note that only leptonic decays are supported. The parameters `vdecaymodew1` and `vdecaymodew2` are to be set to -11, -13 or -15 for decays to positrons, anti-muons, or anti-taus, respectively. Is it also possible to generate events with combinations of positron and anti-muon decays, in which case `vdecaymodew1` and `vdecaymodew2` at the same time should be set to -113. A combination of all leptons (positron, anti-muons, and anti-taus) can be obtained by setting `vdecaymodew1` and `vdecaymodew2` to -135. The template analysis file `pwhg_analysis.f` needed in subsequent steps of the analysis is designed for the $e^+\nu_e\,\mu^+\nu_\mu$ mode. In order to obtain predictions for $pp \to W^-W^-jj$, one can do a calculation for $\bar{p}\bar{p} \to W^+W^+jj$, treat the decay leptons of the W^+ bosons as if they were negatively charged and reverse the momentum directions for parity-odd distributions. In the `POWHEG BOX`, predictions for anti-protons can be obtained by setting the input parameters `ih1` and `ih2` to `-1` in the `powheg.input` file.

We recommend to generate the grid with the option `fakevirt 1` in `powheg.input`. When using this option, the virtual contribution is replaced by a fake one proportional to the Born term. This speeds up the generation of the grid.

In the `POWHEG-BOX-V2` grids can be generated in the parallel mode. To that end, set

```
manyseeds 1
xgriditeration 1
parallelstage 1
```

For a default setup one needs about 50–100 jobs with the number of calls set by

```
ncall1 10000
```

for each. Run the program via

```
$../pwhg_main
```

When prompted

```
enter which seed
```

enter 1 or any other valid seed number.

The program stops automatically after the compilation of this step. We recommend performing at least 3 iterations. After completion of the first iteration, change the value of `xgriditeration` according to the respective iteration, and re-do step 1.

Step 2

To produce true NLO results comment out the `fakevirt` token from `powheg.input`. The runs must be performed in the directory where the previously generated grids are stored.

The integration and upper bound for the generation of btilde can be performed with 50-100 runs with 5000-10000 calls each. Set for instance

```
ncall2 5000
itmx2 1
```

in `powheg.input`.

Folding numbers that are appropriate for runs at LHC energy are:

```
foldcsi 5 !  number of folds on csi integration
foldy 5 !  number of folds on y integration
foldphi 10 !  number of folds on phi integration
```

In addition, adapt the values for steering the parallel mode of the program to

```
xgriditeration 1
parallelstage 2
```

Run jobs in parallel, in the same way as explained for step 1 above.

Upon the completion of this step, for each parallel run a file `pwg-*-NLO.top` is generated (where the * denotes the integer identifier of the run). These files contain the histograms defined in `pwhg_analysis.f` at NLO-QCD accuracy, if the variable `bornonly` is set to zero in `powheg.input`. Setting `bornonly` to 1 yields the respective LO results. In either case, the individual results of the parallel runs can be combined with the help of the `mergedata.f` file contained in the `plot-aux` directory. To this end, just compile the file by typing, e.g.,

```
$ gfortran -o mergedata mergedata.f
```

and run the resulting executable. in your run directory `mergedata 1 *NLO.top`. The program expects a number between 1 and 5 and a list of files to merge. If no number or no list is specified, the user will be prompted by the program to enter them. Running mergedata without any arguments will also explain what the 5 different options are. `mergedata` will combine the histograms into a file called `fort.12`. We provide a file `plot-aux/genplots.sh` and `plot-aux/gnuplotsplit.gp` which can be used to plot the resulting histograms. First run `genplots.sh`

```
./genplots.sh file nameoutput
```

or if a comparison between two different runs is wanted

```
./genplots.sh file1 file2 nameoutput
```

This will result in a file called `genplots.gp`. In the later case the file `plot-aux/pastegnudata.f` has to be compiled and either put in the working directory or in the users path. After `genplots.sh` has been run, the user can produce a set of .eps files with gnuplot running

```
./gnuplotsplit.gp genplots.gp
```

Step 3

Also this step can be run in parallel. We recommend to set

```
ncall1 500000
itmx1 4
ncall2 500000
itmx2 4
nubound 100000
parallelstage 3
```

The program will stop after completion of this step. The parallel execution of the program is performed as in the previous steps.

Step 4

Set `numevts` to the number of events you want to generate per process, for example

```
numevts 5000
```

and run in parallel.

At this point, files of the form `pwgevents-[index].lhe` are present in the run directory.

Count the events:

```
$ grep '/event' pwgevents-*.lhe | wc
```

The events can be merged into a single event file by

```
cat pwgevents-*.lhe | grep -v '/LesHouchesEvents' > pwgevents.lhe
```

Analyzing the events

It is straightforward to feed the `*.lhe` events into a generic shower Monte Carlo program, within the analysis framework of each experiment. We also provide a sample analysis that computes several histograms and stores them in topdrawer output files.

Doing (from the `VBF_Wp_Wp` directory)

```
$ make lhef_analysis
$ cd testrun
```

```
../lhef_analysis
```

analyses the bare POWHEG BOX output, creating the topdrawer file LHEF_analysis.top. The targets main-HERWIG-lhef and main-PYTHIA-lhef are instead used to perform the analysis on events fully showered using HERWIG or PYTHIA. Various setting of the Monte Carlo can be modified by editing the files setup-PYTHIA-lhef.f and setup-HERWIG-lhef.f respectively.

References

[1] B. Jäger, C. Oleari, D. Zeppenfeld, *Next-to-leading order QCD corrections to W^+W^+jj and W^-W^-jj production via weak-boson fusion*, Phys. Rev. **D80** (2009) 034022. [arXiv:0907.0580 [hep-ph]].

[2] B. Jäger, G. Zanderighi, *NLO corrections to electroweak and QCD production of W^+W^+ plus two jets in the POWHEG BOX*, JHEP **1111** (2011) 055. [arXiv:1108.0864 [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** (2010) 043. [arXiv:1002.2581 [hep-ph]].