

Manual for slepton pair plus jet production in the POWHEG BOX V2

This document describes the settings and input parameters that are specific to the implementation of slepton pair production in association with a jet at the LHC within the POWHEG BOX V2 framework. The parameters that are common to all POWHEG BOX V2 implementations are given in the documents `manual-BOX.pdf` and `Manyseeds.pdf` in the `POWHEG-BOX-V2/Docs` directory. If you use our program, please quote Refs. [1–4].

Model parameters

Note that the current implementation has been tailored to left-handed slepton pair production in association with a jet. Right-handed sleptons are not yet supported. The masses of squarks that appear as virtual particles in the loop corrections are implicitly assumed to be degenerate. Mixing effects between squarks of different generations are entirely disregarded. Please contact the authors if you want to apply the code to scenarios that do not meet these limitations. Model parameters are read from the file `param.card.dat` in the subdirectory `MODEL` which is formatted according to the SUSY Les Houches Accord (SLHA) [5,6] and can be taken from standard SUSY spectrum generators.

The treatment of the virtual corrections is specified by the flag `qcdonly` in the `powheg.input` file. If the switch is enabled, only Standard-Model type virtual corrections with a Drell-Yan-type structure are computed. Contributions with supersymmetric loop corrections and non-Drell-Yan type diagrams are neglected. While the numerical impact of the neglected terms is very small for current SUSY mass limits, the execution time of the code is halved when this switch is enabled.

Download and running of the code

Download the POWHEG BOX V2 following the instructions at the website

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

and subsequently retrieve the subdirectory `dislepton-jet` into the directory `POWHEG-BOX-V2`. Before compiling the code make sure that:

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

Go to the project directory

```
$ cd POWHEG-BOX-V2/dislepton-jet
```

and compile the code with

```
$ make
```

which may take some time. All three generated programs `pwhg_main`, `lhef_analysis` and `main-PYTHIA-lhef` are to be executed in a separate working directory. This directory has to contain a file `powheg.input` that specifies several technical parameters and, for parallel running, a file `pwgseeds.dat` with different seeds for each parallel computation. Samples of these files are provided in the subdirectory `runs`. For your own runs, create a subdirectory, e.g. named `testrun`, in `dislepton-jet`:

```
$ mkdir testrun
```

enter the `testrun` directory

```
$ cd testrun
```

and perform all your runs there. We recommend running the program in a parallel mode in several consecutive steps, with the following common settings in `powheg.input`:

```
foldcsi 2
foldy 2
foldphi 2
manyseeds 1
```

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 timings given in the following refer to the program compiled with `gfortran` and run on Intel Xeon E5-2670 CPUs with 2.6 GHz and 32 GB RAM each.

Step 1

In this first step, the importance sampling grids are generated. To speed up this step, set

```
fakevirt 1
```

in order to use fake virtuals that are proportional to the Born rather than the full virtual contributions for the preparation of the importance sampling grids. Make sure that the relevant technical parameters in `powheg.input` are set to the following values:

```
xgriditeration 1
parallelstage 1
```

For precision results, we recommend running about 100 parallel jobs with the number of calls set by

```
ncall1 200000
```

for each. In order to run the parallel jobs, use

```
$ ../pwhg_main
```

When prompted

```
enter which seed
```

enter an index for each run (from 1 to 100). The `pwgseeds.dat` must contain at least 100 lines, each with a different seed.

The completion of the first iteration of the grid production takes roughly one hour of CPU per job. Once all jobs of the first iteration are completed, the grids are refined in further iterations. We recommend to perform at least a second iteration, setting

```
xgriditeration 2  
parallelstage 1
```

and rerunning the program as in the first iteration. If more iterations are needed, the value of `xgriditeration` has to be adapted accordingly. Each iteration takes roughly the same amount of CPU time as the first one.

Step 2

In the directory where the previously generated grids are located, the generation of the NLO results can be performed, again in the parallel mode. Make sure to enable the full virtual amplitudes by deactivating the `fakevirt` switch. Set

```
ncall2 300000  
xgriditeration 1  
parallelstage 2
```

in `powheg.input` and run jobs in parallel, in the same way as explained for step 1 above. Upon the completion of step 2 (after about eight hours), for each parallel run with number `[index]` a file `pwg-[index]-NLO.top` is generated. 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, for instance with the help of auxiliary files contained in the `plot-aux` directory of the POWHEG BOX V2 implementation VBF.Z.Z as described in the corresponding manual.

Step 3

Also this step can be run in parallel. With the settings

```
nubound 100000  
parallelstage 3
```

each jobs takes roughly five minutes. The parallel execution of the program is performed as in the previous step.

Step 4

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

```
numevts 100000
```

and run in parallel with

```
parallelstage 4
```

Generating the specified number of events takes about one day. At this point, files of the form `pwgevents-[index].lhe` are present in the run directory. The events can be merged into a single event file `pwgevents.lhe` by

```
$ cat pwgevents-*.lhe | grep -v "/LesHouchesEvents" > pwgevents.lhe
$ echo "</LesHouchesEvents>" >> pwgevents.lhe
```

Analyzing the events

It is straightforward to feed the combined event file `pwgevents.lhe` 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 `gnuplot` output files. Doing

```
$ ../lhef_analysis
```

analyzes the bare POWHEG BOX output and writes histograms to the file `pwgLHEF_analysis.top`.

Parton showers, slepton decays, and further effects can be simulated when the events are handed over to PYTHIA [7] by

```
$ ../main-PYTHIA-lhef
```

The corresponding histograms can be found in `pwgPOWHEG+PYTHIA-output.top`.

Various settings of PYTHIA can be modified by editing the file `setup-PYTHIA-lhef.F` prior to compilation. We modified PYTHIA in such a way that by setting the flag `mstp(41)=1` slepton decays are activated. PYTHIA generates the decays according to the masses provided in `param_card.dat`. For the case of each slepton decaying into a lepton and the lightest neutralino, our sample analysis provides various histograms for the decay products.

References

- [1] B. Jäger, A. von Manteuffel, and S. Thier, *Slepton pair production in association with a jet: NLO-QCD corrections and parton-shower effects*, arXiv:1410.3802.
- [2] 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]].

- [3] R. K. Ellis and G. Zanderighi, *Scalar one-loop integrals for QCD*, JHEP **0802** (2008) 002 [arXiv:0712.1851 [hep-ph]].
- [4] G. J. van Oldenborgh, *FF: A Package to evaluate one loop Feynman diagrams*, Comput. Phys. Commun. **66** (1991) 1.
- [5] P. Z. Skands *et al.*, *SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators*, JHEP **0407** (2004) 036 [hep-ph/0311123].
- [6] B. C. Allanach *et al.*, *SUSY Les Houches Accord 2*, Comput. Phys. Commun. **180** (2009) 8 [arXiv:0801.0045 [hep-ph]].
- [7] T. Sjostrand, S. Mrenna, P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, JHEP **0605** (2006) 026 [hep-ph/0603175].