Squark Production and Decay in the POWHEG-BOX

This manual summarizes the parameters which are specific to the implementation of squark antisquark and squark pair production and decays in the POWHEG-BOX framework. More details on the implementation of these processes can be found in [1] and [2]. For a summary of the parameters which are common to all processes included in the POWHEG-BOX see the main manual.

Compiling the program

The squark antisquark and squark pair production processes are implemented as two independent programs. The respective process-dependent routines can be found in the subfolders `squark_squark` and `squark_antisquark`, which contain each a sample `makefile`. In order to generate the executables change into the respective folder and run `make`. Per default the programs require `fastjet` and `lhapdf`, so make sure that `fastjet-config` and `lhapdf-config` are set correctly in your path, otherwise you have to adapt the `makefile`.

In order to run the generated executables you have to provide a `powheg.input` file, which contains the settings for different technical parameters, and an SLHA file containing the masses of the sparticles. Some sample files can be found in the subfolder `testrun-lhc`. Copy these files to a directory of your choice and adapt the parameters as desired. Then either execute `pwhg_main_sqsq` or `pwhg_main_sqantisq` from the folder which contains the input files.

Input parameters

The input parameters introduced in the course of the implementation of the processes are summarized in the following, the individual settings can be changed in the `powheg.input` file.

- **Spectrum file:** The user has to provide a file containing the masses, mixing matrices etc. of the considered scenario according to the SUSY Les Houches accord:

  ```
  SLHA   'input.slha' !  name of the SLHA input-file
  ```

- **Process choice:** There are three different ways to operate the code. The first option consists in specifying one specific final-state configuration by setting

  ```
  part1      'uL'  ! first squark: e.g. uL for sup-left and so on
  part2      'uL'  ! second squark (for antisquarks: uLbar)
  withcc     1     ! for squark pair production: consider also antisquarks
  ```

  where the flag `withcc` is only used for squark pair production.

  In the second option a list of Born flavour structures, which shall be considered in one run, is read from an external file:

```
partialsumup 0        ! if set to 1: read in a user-provided file
SUMFLST  'flst.input' ! input file containing the flavour structures
```

In this case the masses in all given combinations have to be identical, otherwise the program stops. The flag `smartsig` is set automatically in order to benefit from the considerable speed up. For a sample file see `testrun-lhc/flst.input`.

In the third operation mode all squarks are treated as mass degenerate and all channels are summed up. This mode is invoked by setting

```
debug_sumup  0  ! sum up all contributions using degenerate mass-spectrum
avgslhamass  0  ! if set: read SLHA-file, but form average of sq-masses
msquark  500.0  ! mass of the degenerate squarks
mgluino  500.0  ! gluino-mass
mtop     175.0  ! top-mass
```

If the flag `avgslhamass` is set, the common squark mass is calculated by averaging the masses of all squarks occurring in the final state, using the masses provided in the SLHA file. In this case the top and gluino mass are read from this file, too. Otherwise the specified masses are used. Again, the flag `smartsig` is set.

- **Scale settings:** The renormalization and factorization scales, which are in any case identical, can be either set to a fixed value or a dynamic scale can be used.

```
fixedscale      1     ! fixed (1) or dynamic (0) scale
setscale        500.0 ! set mur=muf to a specific value
dynscalechoice  1     ! 1: average of m_T, 2: m_sq1sq2
```

The fixed scale can be set to any value. If the `setscale` keyword is not present in the input file, the average mass of the two final-state squarks is used as default value. Possible choices for dynamic scales are the average of the transverse masses of the two squarks or their invariant mass.

- **Decays:** The flags specifying the decays of the produced squarks are the following:

```
decay       0  ! perform decays of squarks (1) or not (0)
decchan1    1  ! decay-channel for first squark, (1): sq->q chi^0_1
decchan2    1
NLOwhich    3  ! NLO only in prod.(1), only in decay (2) or in both (3)
CalcGatot   1  ! calculate Gamma^sq_tot (1) or read from SLHA file (0)
NWAapproach 2  ! 1: do not expand NLO width in denominator, 2: expand all
```

At the moment only the decay mode $\tilde{q} \to q\tilde{\chi}_1^0$ is implemented, hence `decchan1` and `decchan2` have to be set to 1. The NLO corrections can be calculated for either only the production process, only the decay processes, or both. The total width used in the combination formulae for production and decay can be either calculated consistently (see [2] for details) or read from the specified SLHA file. Setting the flag `NWAapproach` to 1 or 2 invokes the use of either

$$\begin{aligned} d\sigma_{\text{tot}} = \frac{1}{\Gamma_{\text{tot}}^{\tilde{q}_1}\Gamma_{\text{tot}}^{\tilde{q}_2}} \Bigg[& d\sigma_0\, d\Gamma_0^{\tilde{q}_1\to\tilde{\chi}_1^0 q} d\Gamma_0^{\tilde{q}_2\to\tilde{\chi}_1^0 q} + \alpha_s \big(d\sigma_0\, d\Gamma_1^{\tilde{q}_1\to\tilde{\chi}_1^0 q} d\Gamma_0^{\tilde{q}_2\to\tilde{\chi}_1^0 q} \\ & + d\sigma_0\, d\Gamma_0^{\tilde{q}_1\to\tilde{\chi}_1^0 q} d\Gamma_1^{\tilde{q}_2\to\tilde{\chi}_1^0 q} + d\sigma_1\, d\Gamma_0^{\tilde{q}_1\to\tilde{\chi}_1^0 q} d\Gamma_0^{\tilde{q}_2\to\tilde{\chi}_1^0 q} \big) \Bigg] \end{aligned} \quad (1)$$

or

$$\mathrm{d}\sigma_{\mathrm{tot}} = \frac{1}{\Gamma^{\tilde{q}_1}_{\mathrm{tot},0}\Gamma^{\tilde{q}_2}_{\mathrm{tot},0}}\Bigg[\mathrm{d}\sigma_0\,\mathrm{d}\Gamma_0^{\tilde{q}_1\to\tilde{\chi}^0_1 q}\,\mathrm{d}\Gamma_0^{\tilde{q}_2\to\tilde{\chi}^0_1 q}\left(1-\frac{\alpha_s\Gamma^{\tilde{q}_1}_{\mathrm{tot},1}}{\Gamma^{\tilde{q}_1}_{\mathrm{tot},0}}-\frac{\alpha_s\Gamma^{\tilde{q}_2}_{\mathrm{tot},1}}{\Gamma^{\tilde{q}_2}_{\mathrm{tot},0}}\right) + \alpha_s\left(\mathrm{d}\sigma_0\,\mathrm{d}\Gamma_1^{\tilde{q}_1\to\tilde{\chi}^0_1 q}\mathrm{d}\Gamma_0^{\tilde{q}_2\to\tilde{\chi}^0_1 q}+\mathrm{d}\sigma_0\,\mathrm{d}\Gamma_0^{\tilde{q}_1\to\tilde{\chi}^0_1 q}\mathrm{d}\Gamma_1^{\tilde{q}_2\to\tilde{\chi}^0_1 q}+\mathrm{d}\sigma_1\,\mathrm{d}\Gamma_0^{\tilde{q}_1\to\tilde{\chi}^0_1 q}\mathrm{d}\Gamma_0^{\tilde{q}_2\to\tilde{\chi}^0_1 q}\right)\Bigg] \tag{2}$$

for the combination of production and decay, see [2] for details.

- **Cuts and jet parameters:** In the NLO analysis performed during the integration of the $\overline{B}$ function some cuts on the jets, $\not{E}_T$ etc. can be applied, which have to be specified in a separate file:

```
cuts      0      ! apply cuts in NLO-plots (1) or not (0)
CUTS  'cuts.dat' ! name of the file where the cuts are specified
```

A sample file can be found in `testrun-lhc/powheg.cuts`. In order to implement different cuts the corresponding analysis routine and the function `readcuts` have to be adapted. Moreover, if the decays of the squarks are taken into account a jet algorithm has to be applied to cluster the resulting partons. To this end the FASTJET package is used. For the choice of the jet algorithm, the jet radius parameter and the minimal transverse momentum of the jets the following parameters have to be set:

```
jetalgo  2     ! 1:kt, 2: antikt, 3: Cambridge-Aachen
Rpara    0.7   ! jet radius parameter for the (anti)kt-algorithm
ptjmin   1.0   ! minimal pt requested for the jets
```

- **Subtraction of on-shell gluinos:** Several different schemes for the subtraction of contributions with intermediate on-shell gluinos have been implemented. In order to choose a specific scheme the following parameters have to be modified:

```
flagsubmethod 1 ! 1: DS, 2: DR-I, 3: DR-II
! in order to choose a specific DS scheme:
flagsplit     3 ! 0: DS(*)-I, 2: DS-II, 3: DS*-III (gauge-invariant method)
flagunexp     0 ! distinguish DS*-I (0) and DS-I (1)

flagrestrict  1 !modify Jacobian for the subtracted residuum (1) or not (0)
```

The meaning of the different schemes can be inferred from the discussion in [1] and [2]. The option `flagsplit = 1` is in principle equivalent to the DS-II approach, however, with this choice only the resonant amplitude squared, $|M_{\mathrm{r}}|^2$, is treated as regular remnant. If the flag `flagunexp` is set to 1, the analytical expansion in the poles is not performed for the DS-I scheme. In order to neglect the modification of the Jacobian for the reshuffled phase space, the flag `flagrestrict` has to be set to 0. The parameters required for the different schemes are set as follows:

```
widthgluino  0.1 ! the regularizing width for the gluino
radcut       1.0 ! radiation cut applied to avoid negative R-values
```

The number of points and iterations for the integration of the remnant terms can be chosen differently from the settings for the $\overline{\mathcal{B}}$ integration:

```
ncall1split  100000
itmx1split   8
ncall2split  300000
itmx2split   5
```

These settings are only relevant if the DS scheme is used and `flagsplit` $\neq 0$.

- **Miscellaneous options:** The colour flow for events with real emissions can be either determined according to the approximate algorithm implemented in the POWHEG-BOX, or using the colour flow decomposition for the real amplitude squared:

```
flagownRCF 0 ! 0: POWHEG-BOX algorithm to assign real colour flows
             ! 1: use colour flow decomposition
```

The real amplitudes can be either evaluated using the lightcone gauge or the Feynman gauge for the external gluons.

```
flaggauge  1 ! in the real routines: use lightcone (1) or Feynman (2) gauge
```

Both choices yield of course identical results.

Showering the generated events

As usually running the code generates a Les Houches event (LHE) file, which can be processed in turn by any parton shower program which supports this standard. We have tested this using the two Monte Carlo event generators HERWIG++ and PYTHIA. By construction the POWHEG method requires the radiation generated in the showering stage to take place at smaller p_T-values than the first radiation generated with the POWHEG-BOX. In principle this is achieved in the respective shower program by imposing a p_T-veto with the veto scale read from the LHE record of each event, which corresponds to the `SCALUP` entry. However, in case of PYTHIAa further subtlety occurs: if final-state resonances are present the mass of those has to be preserved by the reshuffling operations performed in the shower algorithm. Therefore, the showering of partons originating from the decays of these resonances is performed separately in PYTHIA. The starting scale for these shower contributions is set to the invariant mass of all decay products, hence in the case at hand to the mass of the respective squark. This scale is typically an order of magnitude larger than the `SCALUP` value, leading to much more radiation and thus to a strong bias of the results. In order to correct for this effect, the PYTHIA routines had to be adapted to use the scale specified in the LHE file as starting scale in all individual contributions to the parton shower. A modified version of PYTHIA 6.4.28 can be found in the subfolder `Tools/Pythia6`. In order to override the normal operation mode of the code the flag `MSTJ(200)` has to be set to 1.

References

[1] R. Gavin, C. Hangst, M. Krämer, M. Mühlleitner, M. Pellen, E. Popenda, and M. Spira, *Matching Squark Pair Production at NLO with Parton Showers*, JHEP **10** (2013) 187 [arXiv:hep-ph/1305.4061]

[2] C. Hangst, *Matching Squark Production and Decay at Next-to-Leading Order Accuracy with Parton Showers.* Dissertation, Karlsruher Institut für Technologie, 2014.
http://nbn-resolving.org/urn:nbn:de:swb:90-411555