MC generators and future challenges
SHERPA Tutorial: W+Jets

1 The run card: things to notice

- Any setting with the exception of the process and selector definitions can be given in the (run)...(run) section of the run card.
- Tags (FSF, RSF and QSF) have been introduced for easy scale variations. Tags are replaced throughout the entire run card by their defined value.
- All scales, i.e. the factorisation, renormalisation and resummation scales are set to the above defined scale factors times the value determined by the METS-clustering algorithm.
- Tags (LJET, NJET and QCUT) have been introduced to be used in the process setup, defining the multiplicity of the MC@NLO subprocesses, the maximal number of extra jets, and the merging cut.
- The LOOPGEN<i> tag is used to name the providers of the one-loop matrix elements for the respective multiplicities. For the simplemost case here Sherpa can provide it internally.
- Enabling the EXCLUSIVE_CLUSTER_MODE for the METS-clustering algorithm ensures that only QCD-splittings are inverted, such that the production of a lepton-neutrino pair is always identified as the core process.
- Contrary to the best shower setup, for NLO merging the shower is setup in a crossing invariant way, employing the very same recoil scheme as the MC@NLO emission.
- τ leptons are set massive in order to exclude them from the massless lepton container (90).
- As both Comix and Amegic are specified as matrix element generators to be used, Amegic has to be specified to be used for all MC@NLO multiplicities using ME_Generator Amegic LJET.
- Enhance_Factors have been introduced to increase the statistical quality for processes of higher multiplicity.
- RS_Enhance_Factors have been introduced to increase the statistical quality for the non-resummed parts of the MC@NLO processes the respective multiplicity.

More information and explanations can be found in the manual (`http://sherpa.hepforge.org/doc/SHERPA-MC-2.0.0.html`).

2 Running the example set-up

This is an example for a merged sample for W (W^+ and W^- with the W decaying to a muon and muon-neutrino) production in association with up to 4 jets, where 0,1 and 2 jets are at NLO and the third and fourth jet are at LO. The loop matrix element for W is included in SHERPA, the ones for W with one or two jets are provided by BLACKHAT. The same run card can also be used to run with

GoSam as loop generator. For this the line defining LOOPGEN1:=BlackHat needs to be removed and the following block of statements (4 lines) has to be uncommented.

Sherpa and the loop generators have been installed in the tutorial directory on lxplus (`/afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2`). There is also a shell script that sets the paths:

```
source /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/env.sh
```

You should create a run directory RUNDIR and change into that directory:

```
mkdir RUNDIR
cd RUNDIR
```

If you are running with GoSam, the following link has to be set and a file needs to be copied:

```
ln -s /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/GoSam+Sherpa_Examples/
w12jets-2.0/libgolem_olp.so
cp /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/GoSam+Sherpa_Examples/
w12jets-2.0/OLE_order.olc .
```

The matrix elements for the NLO processes are generated with Amegic, which writes process libraries that have to be compiled. There are compiled libraries available on lxplus that you can use if you want to skip this step.
If you want to start generating events straight away, copy the run card to your run directory and create links to the process libraries and integration results in the central directory:

```
cp /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/setups/W+Jets/Run.dat .
ln -s /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/setups/W+Jets/Process
ln -s /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/setups/W+Jets/Results.30
```

If, on the other hand, you want to see how the process libraries are generated, you run Sherpa and compile the libraries instead of linking the compiled libraries:

```
cp /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/setups/W+Jets/Run.dat .
ln -s /afs/cern.ch/project/theory/LPCC_MC_Workshop/tutorial2/setups/W+Jets/Results.30
Sherpa
./makelibs
Sherpa
./makelibs
...
```

The last step (run Sherpa and makelibs) has to be repeated for every jet multiplicity (in this case 3) at NLO and for each process (in this case 2: $W^+ + \text{jets}$ and $W^- + \text{jets}$).
Once the libraries are compiled the event generation can begin. This is done by simply running the same run card as before asking for as many events as you require:

```
Sherpa EVENTS=100000
```

In this example the events are analysed with Rivet, which is interfaced to Sherpa so that the events can be handed over directly. The histograms are written into a file Analysis.30.aida. The plots are created and compiled into a neat html-page by running the command

```
rivet-mkhtml Analysis.30.aida
```

This creates a directory called 'plots' containing an index.html, which can be viewed with any web browser, e.g.

```
firefox plots/index.html
```

3 Proposals for variations

Now you can either run one of the other example set-ups, or try some variations of this one. As the integration of the high multiplicity processes takes rather long, it is advisable to try only variations that do not require integration the cross sections again during the tutorial. Here are a few suggestions:

- vary the number of jets that are computed at NLO by varying the LJET tag in the run card (LJET=0 corresponds to MePs@Lo, LJET=2 to MeNloPs)
- vary the scales (when generating weighted events this does not require to integrate the cross sections again)
- generate unweighted events for the MePs@Lo or MeNloPs (unweighting MePs@Nlo events works, but is still very inefficient for higher multiplicities of NLO jets)
- switch on the hadronisation and/or underlying event