Introduction to Sherpa

Tutorial for summer schools

1 Prerequisites

For this tutorial, we will be using Docker. To download the image for Sherpa and Rivet, simply type

```
docker pull hepstore/rivet-sherpa:3.1.6
```

We have provided a little setup.sh script which will let you use the image software outside of the container You can source the script like so

```
source setup.sh
```

which should give you access to all relevant command-line tools. To check that it worked, try e.g.

```
rivet --version
```

which should return `rivet v3.1.6`. You can pass other commands to the Docker environment using the dockerdo alias, e.g.

```
dockerdo Sherpa
```

Alternatively, you can run the container interactively, e.g. like so

```
docker container run -it hepstore/rivet-sherpa:3.1.6 /bin/bash
```

If you would like to have access to your current directory also from within the container, you can also use the following variation of the above command:

```
docker container run -it -v $PWD:$PWD -w $PWD hepstore/rivet-sherpa:3.1.6 /bin/bash
```

For more details/troubleshooting on docker also have a look at the main tutorial website and the common project page!

You can download the example run cards used in this tutorial from gitlab.

2 A First Sherpa Run

Sherpa is a complete Monte-Carlo event generator for particle production at lepton-lepton, lepton-hadron, and hadron-hadron colliders [1]. The simulation of higher-order perturbative QCD effects, including NLO corrections to hard processes and resummation as encoded in the parton shower, is emphasized in Sherpa. QED radiation, underlying events, hadronization and hadron decays can also be simulated. Alternatively, Sherpa can be used for pure parton-level NLO QCD calculations with massless or massive partons.

Many reactions at the LHC suffer from large higher-order QCD corrections. The correct simulation of Bremsstrahlung in these processes is essential. It can be tackled either in the parton-shower approach, or using fixed-order calculations. Sherpa combines both these methods using a technique known as Matrix Element + Parton Shower merging (ME+PS). Details are described in Ref. [2] and have been discussed in the lectures. This tutorial will show you how to use the method in Sherpa.

2.1 The Input File

Sherpa is steered using input files, which consist of several sections. A comprehensive list of all input parameters for Sherpa is given in the Sherpa manual [3]. For the purpose of this tutorial, we will focus on the most relevant ones.

Download the first example run card from gitlab (since Sherpa writes out data during its runs, it might be advisable to create separate directories for each run):

```
mkdir TTbar && cd TTbar\\
wget https://gitlab.com/hepcedar/mcnet-schools/zakopane-2022/-/raw/main/sherpa/TTbar/Run.dat
```

Open the `Run.dat` file in an editor. Have a look at the section which is delimited by the tags `(run){` and `}(run)` (We will call this section the `(run)` section in the following). You will find the specification of the collider, i.e. its beam type and center-of-mass energy, as well as a couple of other parameters, which will be explained later.

The `(processes)` section specifies, which reactions are going to be simulated. Particles are identified by their PDG codes, i.e. 1 stands for a down-quark, -1 stands for an anti-down, 2 for an up-quark, etc. The special code 93 represents a "container", which comprises all light quarks, b-quarks, and the gluon. It is also called the "jet" container.

2.2 Running Sherpa

Have you found out which physics process is going to be generated? You can verify your guess by running the command line (Note that all commands in this section must be prefixed by `docker-run-sherpa` or run inside a shell started by `docker-run-sherpa bash`. Running inside a shell is recommended. See the introduction for details.)

```
Sherpa -e1 -o3
```

When run for the first time, Sherpa will produce diagram information for the calculation of hard scattering processes. It will also compute that hard scattering cross sections, which are stored, together with the diagram information, for subsequent runs.

The option `-e1` used above instructs Sherpa to produce one event, and the option `-o3` will print the result of event generation on the screen. You will see Sherpa's internal event record. Search for `Signal Process` inside the output and check incoming and outgoing particles. What happens to the unstable particles after they have been produced in the hard scattering? Is this result physically meaningful?

Have a look at the Feynman diagrams, which contribute to the simulation of the hard process:

```
plot_graphs.sh graphs/
firefox graphs/index.html
```

Are these the diagrams you expect to find? If not, which ones are missing? Can you find the setting in the runcard which restricts the set of diagrams?

2.3 Unstable particles

Open the input file again. Add the setting `HARD_DECAYS On;` (note the underscore, and be aware that Sherpa run cards are case sensitive) to the `(run)` section, which instructs Sherpa to automatically decay unstable particles. Verify that the particles you wish to decay are flagged unstable by checking the screen output of Sherpa during runtime. Search for the 'List of Particle Data'. Note that you can set particles stable individually using the switch `STABLE[<PDG ID>]=1`.

Run again

```
Sherpa -e1 -o3
```

and observe how the event record changes.

For experts: What happens when you change or remove the setting `WIDTH[6] 0;` (Hint: Search for 'Performing tests' in the screen output of Sherpa). Can you guess what the problem might be?

2.4 ME+PS merging

The current runcard lets Sherpa generate events at lowest order in the strong coupling. To improve the description of real radiative corrections, we can include higher-multiplicity tree-level contributions in the simulation. This is done by changing the process specification

```
Process 93 93 -> 6 -6;
```

to

```
Process 93 93 -> 6 -6 93{1};
```

The last entry instructs Sherpa to produce up to one additional "jet" using hard matrix elements and combine the respective process with the leading-order process. This is known as Matrix Element + Parton Shower merging (ME+PS), or the CKKW method. The essence of the method is a separation of hard from soft radiative corrections, achieved using phase-space slicing by means of a variable called the jet criterion. The slicing parameter is called the merging cut.

Let us assume we want to classify jets of more than 20 GeV transverse momentum as hard. In Sherpa, the corresponding merging cut would be specified as

```
CKKW sqr(20/E_CMS);
```

Therefore, the complete (processes) section for the merged event sample reads:

```
(processes){
  Process 93 93 -> 6 -6 93{1};
  CKKW sqr(20/E_CMS);
  Order (*,0);
  End process;
}(processes);
```

If you like, you can have a look at the Feynman graphs again, which contribute to the ME+PS merged event sample. In this case, you should not remove the line `Print_Graphs graphs;` from the (processes) section, and rerun the plot command from Sec. 2.2. Run the new setup. Why does Sherpa compute another cross section?

2.5 Analyses

By default, Sherpa does not store events. Run Sherpa with the following command to write out weighted event files in HepMC format, which can subsequently be analyzed

```
Sherpa EVENT_OUTPUT=HepMC3_GenEvent[events_1j]
```

Sherpa will produce a gzipped file called `events_1j.hepmc`, which can be processed with Rivet. Alternatively, you can use Sherpa's interface to rivet, to analyse event directly without writing out large amounts of data. To do that, add the following line to the (run) section of your run card

```
ANALYSIS Rivet
```

To tell rivet what analysis should be used, you have to add a new section to the run card

```
(analysis){
  BEGIN_RIVET {
    -a MC_TTBAR;
  } END_RIVET;
}(analysis);
```

The option `-a MC_TTBar` instructs Rivet to run a Monte-Carlo analysis of semileptonic $t\bar{t}$ events, which will provide us with a few observables that can be used for diagnostic purposes. You can change the name of the produced yoda file by adding

```
ANALYSIS_OUTPUT=Analysis.yoda
```

in the (**run**) section or on the command line

```
Sherpa -A Analysis.yoda
```

You can view the results of this analysis by running the command

```
rivet-mkhtml --mc-errs Analysis.yoda
```

and opening `rivet-plots/index.html` in a browser.

Now it is your turn: Generate a separate file without ME+PS merging and analyze it with Rivet. Compare the results to your previous analysis using `rivet-mkhtml`. Do you observe any differences? Why?

2.6 Other hard scattering processes

For the project, we want to simulate the production of a lepton and a neutrino. Can you figure out how to change the (**process**) section of your run card? (Hint: Similar to 93 for partons, Sherpa also defines containers 90 for leptons and 91 for neutrinos.)

Note that, by default, the lepton container also includes τ-leptons. You can exclude them by adding

```
MASSIVE[15] 1
```

to the (**run**) section (15 is the PDG code for τs). You will also need a new section

```
(selector){
  Mass 11 -12 1. E_CMS
  Mass 13 -14 1. E_CMS
  Mass -11 12 1. E_CMS
  Mass -13 14 1. E_CMS
}(selector)
```

Why is this section needed? You can also find a ready made example run card for this process in `sherpa/W/Run.dat` on the tutorial gitlab page.

3 Advanced options

At this point you will probably want to have a look at the project description on the gitlab page and setup Sherpa to study W production at the LHC and analyse the results. You can use the suggestions below to study systematic uncertainties of this calculation.

3.1 Uncertainty estimates

Using Sherpa, you can study the renormalization and factorization scale dependence of the simulation as well as the uncertainty related to the PDF fit. Thanks to reweighting techniques, this can be done on-the-fly, without running Sherpa multiple times. The reweighting is enabled in Sherpa by adding the following flags to the (**run**) section of you input file:

```
SCALE_VARIATIONS 0.25,0.25 0.25,1 1,0.25 1,4 4,1 4,4;
PDF_LIBRARY LHAPDFSherpa; PDF_SET NNPDF30_nnlo_as_0118;
PDF_VARIATIONS NNPDF30_nnlo_as_0118[all];
```

The line starting with `SCALE_VARIATIONS` instructs Sherpa to perform a six-point variation of the renormalization and factorization scales, with the pairs of numbers representing the scale factors for μ_R^2 and μ_F^2. The line starting with `PDF_VARIATIONS` instructs Sherpa to perform a variation using all PDFs in the error set NNPDF30_nnlo_as_0118, which is interfaced through LHAPDF. Finally, the default PDF is set to NNPDF30_nnlo_as_0118. Depending on the rivet version used, you might have to add

```
HEPMC_USE_NAMED_WEIGHTS 1;
SKIPWEIGHTS 0;
```

in the (`analysis`) section, after the `BEGIN_RIVET` line.

3.2   Hadronization and underlying event

So far, multiple parton interactions, which provide a model for the underlying event, have not been simulated. Also, hadronization has not been included in order to speed up event generation. You can enable both by removing or commenting the line

```
MI_HANDLER None; FRAGMENTATION Off;
```

How do the predictions for the observables change and why?

3.3   Changing model parameters

You can change model parameters like coupling strengths, particle widths and particle masses in the run card. Have a look in the manual for a full overview of the available options. To change, for example, the mass of the W boson, you would include the following line (which in fact sets the mass to its default value)

```
MASS[24] = 80.385;
```

3.3.1   Side note: Tags

A useful feature of Sherpa run cards are "tags": A line like

```
FOO:=5
```

will define a tag "FOO" to represent "5". Any occurence of "FOO" in the run card will then be replaced by "5" (be aware of unwanted consequences however, something like `ALPHA:=foo; ALPHAS(MZ) 0.1188;` will obviously cause problems). Tags, as well as any other parameter, can also be set on the command line. Adding the following to your run card

```
MW:=385
MASS[24] = 80.MW;
ANALYSIS_OUTPUT = Analysis_mWMW
RESULT_DIRECTORY = Results_mWMW
```

will for example allow you to run Sherpa with different W masses like

```
Sherpa MW:=385
```

which will recalculate the cross section and generate events with the modified W mass and write the results for different masses into different files.

3.4 Variation of the merging cut

ME+PS merging is based on phase-space slicing, and the slicing cut is called the merging scale. Is is an unphysical parameter, and no observable should depend sizeably on its precise value. Verify this by varying the merging cut by a factor of two up and down.

Note that the initial cross sections that Sherpa computes will be different for different values of the merging cut (Why?). You should therefore instruct Sherpa to use different result directories for each run in the test. The result directory is specified on the command line with option `-r`, for example

```
Sherpa -r MyResultDirectory/
```

3.5 QED Corrections

Higher order QED corrections are effected both on hard interaction and, upon their formation, on each hadron's subsequent decay. The Photons module is called in both cases for this task. It employes a YFS-type resummation of all infrared singular terms to all orders and is equipped with complete first order corrections for the most relevant cases (all other ones receive approximate real emission corrections built up by Catani-Seymour splitting kernels). Add the following flags to the (run) section

```
YFS_MODE = [0,1,2]
YFS_USE_ME = [0,1]
```

YFS_MODE, unsurprisingly, specifies the mode in which Photons will treat the resummation. YFS_MODE = 0 turns off Photons completely. YFS_MODE = 1 is a "soft only" treatment, meaning soft emissions will be treated correctly to all orders but no hard emission corrections will be included. To include hard emission corrections up to first order in α_{QED} set the mode to YFS_MODE=2. The switch YFS_USE_ME = [0,1] specifies how the hard emissions are treated. If YFS_USE_ME = 0, then Photons will use collinearly approximated real emission matrix elements. Virtual emission matrix elements of order α_{QED} are ignored. If, however, YFS_USE_ME = 1, then exact real and/or virtual emission matrix elements are used wherever possible. This treatment is completely exclusive in the photon multiplicity and the event will include additional photons. You can verify this by checking your HepMC output file. Since we now have soft emissions, how do we ensure we have an infrared and collinear safe observable?

References

[1] T. Gleisberg et al. "Event generation with SHERPA 1.1." JHEP **02** (2009) 007.

[2] A. Buckley et al. "General-purpose event generators for LHC physics." Phys. Rept. **504** (2011) 145.

[3] https://sherpa.hepforge.org/doc/SHERPA-MC-2.0.0.html.