Monte-Carlo Uncertainties in Higgs + jets

Terascale Monte Carlo School 2014, DESY Hamburg

2014-03-12

1 Introduction

In this tutorial we will discuss some of the uncertainties related to the simulation of multi-jet final states, using Higgs-boson plus jets production as an example. Ideally, you should work in groups of four (or multiples thereof), where each member checks some uncertainties with the help of a different event generator. Ideally, you will have a selection of people who already know "their" generator from the tutorial of Day 1.

At the end of Day 2, you should exchange your histogrammed results within your group, so that all members can generate a full set of plots. Then, you are able to discuss the results, and judge which uncertainties are relevant for which observables.

To share your results, you can exchange emails, or, alternatively, you can use a USB flash drive. You need to make it available to the VM using the "Devices" tab, see `http://tinyurl.com/69ybgp4`, steps 4 and 5. Note that it is essential to add your user to the vboxuser group, as explained in step 5! You can then open the file manager from the task bar and mount the flash drive by clicking on the "usb" icon.

2 General considerations

In this tutorial, we will use different generators to illustrate some uncertainties in simulations of LHC processes. In particular, you will have a closer look at

- Uncertainties in the matching procedure,
- Scale uncertainties,
- PDF uncertainties,
- Non-perturbative uncertainties.

Different event generators are sometimes based on very different assumptions about the underlying physics in certain collider processes. However, every generator prediction will come with the above uncertainties, and different generator authors will stress different variations differently. Also, bear in mind that all of these variations are entangled, and that dividing them should be considered a first step in estimating uncertainties. For the current purposes, we use different generators to highlight different sources of uncertainty. Our choice was driven simply by convenience, and should not imply anything about the sophisication of the generators.

You will investigate the variation of Higgs + jets observables for three different sets of cuts on the event structure

(a) *Inclusive* events: No cuts are applied in the event selection (jet observables are defined through anti-k$_\perp$ jets with $R = 0.5$, $p_{\perp j,min} = 5$ GeV);

(b) Loose *dijet* cuts: Only loose cuts are applied to additional QCD activity (at least two anti-k$_\perp$ jets with $R = 0.5$, $p_{\perp j,min} = 20$ GeV);

(c) *VBF* cuts: Strong cuts are applied to additional QCD activity (at least two anti-k$_\perp$ jets with $R = 0.5$, $p_{\perp j,min} = 20$ GeV, $m_{j_1 j_2} > 400$ GeV, $|y_{j_1} - y_{j_2}| > 2.8$).

We will use Rivet as histogramming tool for this tutorial. Potentially interesting plots have been collected in the analysis `MC_HJETS_DESY`. Please change to the directory `~/tutorials/higgs/analysis/` and run

```
make
```

This will build a Rivet plugin library from the source files in this directory.

3 Generating and analyzing events

To visualize the results of this tutorial you can use the script `plotit.sh`, which is provided in the top level directory, `~/tutorials/higgs/`. Running `./plotit.sh --help` will show all available options. The plot script will be your most efficient means to put all results onto a single web page and compare the output of the different generators. To this end, you would run

```
./plotit.sh hw hwme py pyme pyps sh shme shps
```

You can display the results in a browser using the command `firefox plots/index.html`.

Alternatively, if you want to plot, say, the central predictions only and store the plots in a different folder, you would run

```
./plotit.sh -o central hw py sh
```

Alternatively, you can use your own `rivet-mkhtml` commands.

3.1 Herwig

With HERWIG++ you will be looking into various choices in ME+PS matching. NLO accuracy can be obtained by the Standard HERWIG++ shower with the Powheg approach. Additionally in the Framework of Matchbox we will be looking into a MC@NLO-like matching procedure with the Dipole Shower. So comparing different matching approaches and different showers – all NLO accurate – we should get an impression of uncertainties coming from matching with parton showers.

Sample input files can be found in

```
~/tutorials/higgs/herwig
```

All runs will be analyzed with the Rivet analysis `MC_HJETS_DESY`. Rivet's histogram output is written to `.aida` files. In this part of the tutorial we mainly take at scale variations. In the input files provided MPI (multi parton interactions) and hadronisation are switched off for time reasons. Try to find the switches and turn them on. Then make a comparison between the run times for one or more runs.

3.1.1 Matching uncertainties with standard Herwig++[7] (Powheg)

At the beginning start without showering. In the folder `shower-off` you find .in-files for Higgs and Higgs plus jet production. Try to generate 1000 Higgs events at LO and 5000 events with an additional jet. Why do we need a cut on the additional jet? What can we see in the histograms? Can you explain the LO Higgs results?

Now we switch the the shower on. In the folder `shower-on` you find files for LO and NLO Higgs production matched to the standard angular ordered HERWIG++ shower. Generate $\mathcal{O}(10k)$ events for each approach. For comparison and estimating an uncertainty try to vary the renormalisation and factorisation scale by a factor 2 for the Powheg NLO case. Are the uncertainties driven by statistic or can you see the scale variations? Try to rebin the histograms if necessary!

To plot the results using the `plotit.sh` as described above there schould be in the end the three files:

```
herwig/Shower-on/H-Powheg/{H-Powheg.aida, H-Powheg-05.aida, H-Powheg-2.aida}.
```

`H-Powheg.aida` should contain the central value and the the others scale variations.

3.1.2 Matching uncertainties with Matchbox/Herwig++ and Dipoleshower[8][6] (MC@NLO-like)

In the folder `Matchbox` you again find files for LO and NLO Higgs production. This time in the framework called Matchbox, which is a specialized tool for NLO calculations and matching. Here we will match NLO matrix elements to a p_T ordered parton shower based on Catani Seymour dipoles[5]. If time allows try again a comparison between LO and NLO calculations. If not go directly to the NLO case and try different scale choices. You will find in the in-files various possibilities to vary not only renormalisation/factorisation scales but also the scales of the shower. What are the effects of the variations?

As above you should provide:

```
herwig/Matchbox/NLO-DipolMatching/{LHC.aida, LHC-05.aida, LHC-2.aida}
```

3.2 Pythia

The most recent development in combining fixed-order calculations and parton shower resummation is next-to-leading order merging. This is different from NLO matching (e.g. POWHEG and MC@NLO) because it allows to describe different parton multiplicities simultaneously at NLO accuracy. NLO merging schemes are the successor of tree-level merging schemes.

In this tutorial, we will assess the uncertainties of the unitarised NLO+PS merging method (UNLOPS, [1]). For this, we will depart from our usual practise of simply printing histograms to the terminal. Instead, we will use a sample main program that uses HepMC output, which can then be analysed with Rivet. We will briefly describe the main program used for this study, and come back to the actual uncertainty estimates below. The Pythia 8 sources are installed in `/opt/Pythia8`, but you will not need to know their explicit location, as the relevant environment variables have been set for you in `~/.bashrc`.

3.2.1 main88.cc

UNLOPS merging is the direct extention of UMEPS (see last session) to NLO accuracy. For this school, we would like our simulation to describe $\mathrm{H}+0$ and $\mathrm{H}+1$ jet simultaneously with NLO accuracy, describe $\mathrm{H}+2$ jets with tree-level accuracy, and take all further jets from the PS approximation. This means we have to consider two types of input calculations: Tree-level and NLO inputs. For this school, we supply tree-level event samples produced with MadGraph, and NLO event samples produced with POWHEG-BOX. UNLOPS then processes these samples in the following way:

(1) Use tree-level matrix elements for n partons as "seeds" for higher-order corrections (of $\mathcal{O}\left(\alpha_{\mathrm{s}}^{n+2}\right)$ and beyond), making sure that no $\mathcal{O}\left(\alpha_{\mathrm{s}}^{n+1}\right)$ are produced.

(2) Add the NLO samples, making sure that no higher orders (of $\mathcal{O}\left(\alpha_{\mathrm{s}}^{n+2}\right)$ and beyond) are produced.

(3) Unitarise everything, making sure that no unwanted $\mathcal{O}\left(\alpha_{\mathrm{s}}^{n+1}\right)$ terms are produced, i.e. ensure that the inclusive cross section is given by the $\mathrm{H}+0$ NLO result.

This is done internally in `main88.cc`, the sample main program we will use to generate NLO merged results. In the event generation phase, `main88.cc` uses the tree-level input file twice (once for Step (1) and once for Step (3)), as well as using the NLO (POWHEG) inputs twice (once for Step (2) and once for Step (3)). `main88.cc` is described in detail in the online manual under **Link to other programs → NLO merging**.

`main88.cc` reads a settings file (e.g. `main88.cmnd`) for the necessary settings. It is important to set the switches

```
// Definition core process for merging
Merging:Process                = ?
Merging:mayRemoveDecayProducts  = ?
// Maximal number of additional LO jets.
Merging:nJetMax                = ?
// Maximal number of additional NLO jets.
Merging:nJetMaxNLO             = ?
// Merging scale value.
```

```
Merging:TMS                    = ?
// Values of (fixed) scales in the matrix element calculation.
Merging:muFacInME              = ?
Merging:muRenInME              = ?
// Values of (fixed) scales for the PS lowest multiplicity process.
Merging:muFac                  = ?
Merging:muRen                  = ?
```

in such an input file. The process definition, maximal number of additional tree-level jets and merging scale value have already been discussed in the CKKW-L section. In addition, you now need to set the maximal number of additional jets for which an NLO event sample is available, and the renormalisation and factorisation scales with which the event samples have been produced. Finally, it is also necessary to set te scales which would be used in default Pythia 8 to evaluate the core scattering (i.e. for H production, the mass $m_{\rm H}$). In the case of wimpy showers, the value of `Merging:muFac` further sets the shower starting scale in UNLOPS.

In general, you also need to generate tree-level- and POWHEG event samples as input for `main88.cc`. For this school, we have linked some pre-calculated samples in the directory

```
~/tutorials/higgs/pythia .
```

These need to be mounted by executing

```
~/tutorial/mountSamples.sh
```

The main program `main88.cc` assumes, to allow for a streamlined file parsing, a particular naming convention. All POWHEG event samples should be called

```
myLHEF_powheg_njets.lhe.gz
```

where `myLHEF` is a free process idetifier, which is assumed to be identical for all samples belonging to one particular process. `njets` should give the number of additional partons that are described at NLO accuracy (i.e. *not* counting real-emission partons). POWHEG events for $\mathrm{H}+j$@NLO could for example be called `higgs_powheg_1.lhe.gz`. Tree-level inputs are called

```
myLHEF_tree_njets.lhe.gz
```

A legitimate name for a tree-level sample with two additional jets could e.g. be `higgs_tree_2.lhe.gz`.

To use `main88.cc`, go into the `examples` directory and compile the main program

```
cd ∼/tutorials/higgs/pythia .
make main88
```

and run by issuing a command of the form

```
./main88.exe myInputFile myLHEF myHEPMCoutput
```

`main88.cc` will then, consecutively, read tree-level and POWHEG event samples, and produce HepMC events.

To get used to the program, produce only small number of events, and run

```
./main88.exe main88.cmnd ggh_cc myHEPMCoutput
```

Can you identify the Steps (1)-(3) through the terminal output? Why is step (3) applied both to tree-level and POWHEG samples?

We use HepMC events to parse the Pythia 8 output to Rivet. However, HepMC event files quickly become prohibitively large. This is why Rivet allows the usage of `fifo` pipes to pass the events at generator run time, without having to store large intermediate files. Pythia 8 will write a single event to the pipe, and Rivet will read and analyse this event. To make this as simple as possible, you can use the BASH script `run.sh`. This script allows to run both Pythia 8 and Rivet in the background of one common terminal. Simply run

```
./run.sh
```

`main88.cc` needs to be compiled before running the script. You will have to change the name of the log-files and of the `aida` output files in the script, and change the settings file `main88.cmnd` if you want to generate histograms with different settings.

3.2.2 Soft-physics uncertainties with PYTHIA 8

In this part, we will try to illustrate the impact of soft physics modelling on Higgs + jets observables. For this we will investigate the impact of changing a few parameters in Pythia.

The soft-phyics models in General Purpose event generators are very complex, and it is of course not possible to perform a systematic uncertainty of soft physics modelling in the course of a tutorial. Nevertheless, it seems reasonable to show-case some effects.

We will continue using `main88.cc` as before. Now we will try to look into soft physics modelling. Start by running with and without multiparton interactions (MPI). The necessary input is `PartonLevel:MPI`, which takes can take values `on` or `off`. For which observables does MPI matter most?

MPI are modelled as additional (QCD) $2 \to 2$ scatterings. One of the main parameters govering the MPI activity is the α_s^{MPI} value associated to such splittings. Vary α_s^{MPI} by 5% around its default value (given by `MultipartonInteractions:alphaSvalue = 0.135`). For later comparisons, call the results `mpilow.aida`, `mpicentral.aida` and `mpihigh.aida`.

In which corners of phase space do you see the largest dependence?

Another source of non-perturbative modelling is primordial transverse momentum: At the interface between pertubative radiation and hadronisation, incoming partons are not exactly aligned with the proton momentum. Rather, they have some transverse momentum. Event generators include thsi intrinsic transverse momentum spectrum as modelling parameters. Try to find out where primordial kT makes most difference by varying by 50% around the default value (`BeamRemnants:primordialKThard = 2.0`). For later comparisons, call the results `pktlow.aida`, `pktcentral.aida` and `pkthigh.aida`.

Finally, compare the UNLOPS results to the results of the other Event Generators. This is most conveniently done if you display the variations as an uncertainty band, using the `plotit.sh` script described above. The Pythia results by themselves can be plotted using

```
cd ~/tutorials/higgs/
./plotit.sh -o MyPythiaPlots py pympi pyktp
```

Don't hesitate to contact us [2] if you have further questions.

3.3 Sherpa

To assess the Monte-Carlo uncertainty in Sherpa, we will produce events with the S-MC@NLO technique, and with ME+PS merging at NLO (MEPS@NLO). This is an extension of the leading order CKKW merging method[3]. The NLO merged simulation contains up to 2 jets computed with hard matrix elements, where the 0 and 1 jet process have NLO accuracy and the second jet is simulated at LO. We will vary renormalization and factorization scales as well as the merging scale.

Change to the tutorial directory `tutorials/higgs/sherpa`. In the first run, Sherpa will generate process-specific source code necessary for the computation of the tree-level matrix elements and for the NLO subtraction terms (Note that this is different from what you have experienced during the previous tutorial, because we are using a different matrix element generator). To create the process libraries execute the following commands

```
/opt/SHERPA/bin/Sherpa
./makelibs
```

The individual cross sections have been pre-computed and the results are stored in the tarred file `Res.tar.gz`. Untar the file to get the results file `Results.db` with

```
tar xzf Res.tar.gz
```

You can launch Sherpa and it will start generating events immediately:

```
/opt/SHERPA/bin/Sherpa
```

While Sherpa runs and generates events, have a look at the runcard. The most important part is the `(processes)` section:

```
(processes){
  Process 93 93 -> 25 93{NJET};
  Order_EW 1; CKKW sqr(QCUT/E_CMS);
  NLO_QCD_Mode MC@NLO {LJET};
  Loop_Generator Internal;
  End process;
}(processes);
```

Note that we produce a stable Higgs boson with up to two additional "jets". This is because the tag `NJET` is defined as `NJET:=2` in the `(run)` section. Such a tag can be used anywhere in the runcard, and its definition can also be changed on the command line.

Another tag is `LJET`, which is defined as `LJET:=1,2`. When it is used in the line

```
  NLO_QCD_Mode MC@NLO {LJET};
```

it steers the implementation of NLO corrections to the hard processes. All processes with final state multiplicity 1 and 2 are then computed at NLO accuracy using the S-MC@NLO method[4].

All other options in the `(processes)` section are known from yesterday's tutorial.

The `(run)` section contains several more settings, which are new.

- We use an effective operator approach to implement the Higgs-Gluon couplings using `MODEL SM+EHC`.
- The Higgs boson mass is set to 125 GeV.
- We allow variation of factorization, renormalization (and resummation, although we will not look at this today) scales using the `SCALES` parameter.
- The merging scale can be varied using the `QCUT` tag.

3.3.1 Scale uncertainties with Sherpa

The last 2 points of the previous section deserve some more attention. Look at the precise specification of the scale:

```
  # tags and settings for scale variations
  SP_NLOCT 1; FSF:=1.0; RSF:=1.0; QSF:=1.0;
  SCALES STRICT_METS{FSF*MU_F2}{RSF*MU_R2}{QSF*MU_Q2};
```

Firstly, the scale is defined using the ME+PS merging algorithm. This is indicated by the `STRICT_METS` setting (METS stands for Matrix Elements plus Truncated Showers). Secondly, the three components of the scale definition, factorization, renormalization and resummation scale, which are indicated by `MU_F2`, `MU_R2` and `MU_Q2`, respectively, can be varied independently using the tags `FSF`, `RSF` and `QSF`.

Note that all scales in Sherpa have dimension GeV^2, hence setting `FSF:=4`, for example, means increasing the factorization scale by a factor two.

The merging scale is set by '`CKKW sqr(QCUT/E_CMS)`'. The `E_CMS` tag refers to the centre of mass energy of the event, and `QCUT` is the scale you can vary. It is set to 30 GeV to start with, and we will consider varying it by a factor of 2.

Now it is your turn! Generate the following event files:

- The central prediction, defined by the settings in the runcard.
- Simultaneous variations of the renormalization and factorization scale in the range $1/2 \ldots 2$.
- Variations of the merging scale in the range $15 \ldots 60$.

Note that, because of the setting `ANALYSIS_OUTPUT Analysis/NJET/QCUT/FSF-RSF/QSF;` your aida files will be put into a directory determined by the tags you specify, and it is not necessary to set file names on the command line.

Generate plots from the Sherpa output by using the plot script.

```
cd ~/tutorials/higgs/
./plotit.sh -o MySherpaPlots sh shme shqcut
```

Compare your results to the predictions from other generators by exchanging aida files with your peers and running the plot script as described in the introduction.

3.4 Whizard

Small intro for Whizard.

With Whizard, you will be looking into PDF uncertainties in the simulation.

Sample input files can be found in

```
~/tutorials/higgs/whizard
```

All runs will be analysed with the Rivet analysis `MC_HJETS_DESY`. Rivet's histogram output is written to `.aida` files.

3.4.1 PDF uncertainties with Whizard

Discussion of scale uncertainties and how to generate runs.

References

[1] L. Lönnblad and S. Prestel, JHEP **03** (2013) 166, `arxiv:1211.7278 [hep-ph]`

[2] For merging related questions in Pythia 8, email `stefan.prestel@thep.lu.se`
In case of general problems, contact us under `pythia8@projects.hepforge.org`

[3] S. Höche and F. Krauss and M. Schonherr and F. Siegert, `arXiv:1207.5030`
T. Gehrmann and S. Höche and F. Krauss and M. Schonherr and F. Siegert, `arXiv:1207.5031`

[4] S. Höche and F. Krauss and M. Schonherr and F. Siegert, JHEP **09** (2012) 049, `arXiv:1111.1220`

[5] S. Catani and M. H. Seymour,Nucl. Phys. B **485**, 291 (1997), `[hep-ph/9605323]`

[6] S. Platzer and S. Gieseke, Eur. Phys. J. C **72**, 2187 (2012), `arXiv:1109.6256`

[7] K. Hamilton, P. Richardson and J. Tully, JHEP **0904**, 116 (2009), `arXiv:0903.4345`

[8] S. Platzer and S. Gieseke, JHEP **1101**, 024 (2011), `arXiv:0909.5593`