

High Multiplicity Processes with BlackHat and Sherpa

Z. Bern

Department of Physics and Astronomy, UCLA, Los Angeles, CA 90095-1547, USA

E-mail: bern@physics.ucla.edu

F. Febres Cordero

Physikalisches Institut, Albert-Ludwigs-Universität Freiburg, D-79104 Freiburg, Germany

E-mail: ffebres@physik.uni-freiburg.de

D. A. Kosower

Institut de Physique Théorique, CEA-Saclay, F-91191 Gif-sur-Yvette cedex, France

E-mail: david.kosower@cea.fr

K. Hamilton

Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT

E-mail: keith.hamilton@ucl.ac.uk

Daniel Maître*

IPPP, Durham University

E-mail: daniel.maitre@durham.ac.uk

High multiplicity processes are ubiquitous in studies of high energy collider collision processes. While the ingredients and procedure for the calculation of next-to-leading order corrections to such processes is well-known, their practical implementation is still challenging and much effort is dedicated to making these calculations more tractable. In this contribution we explain the rationale for NLO event files, explain the limitations and trade-offs they entail and illustrate new applications they offer.

12th International Symposium on Radiative Corrections (Radcor 2015) and LoopFest XIV (Radiative Corrections for the LHC and Future Colliders)

15-19 June, 2015

UCLA Department of Physics & Astronomy Los Angeles, USA

*Speaker.

1. High multiplicity processes

High multiplicity processes play a very important role in current BSM searches. Indeed many such searches investigate signals resulting from the decay cascade of heavy new particles. Standard Model high-multiplicity processes are the background for such searches and a detailed understanding is often an important building block in this type of search.

2. n -tuple files

The calculation of fixed order high-multiplicity processes is computationally intensive. The most time-consuming part of the simulation is the calculation of the matrix element, while the evaluation of the PDF, the jet clustering and the calculation of the observables is relatively quick. The prohibitive CPU cost of the matrix element calculation can be amortised by storing them along with kinematical information so that one can reuse them in a different calculation. Such an approach is described in ref. [1] where matrix elements from NLO calculations from BLACKHAT+SHERPA [2, 3, 4, 5, 6] and coefficients of scale logarithms are stored in a ROOT [7] n -tuple file. Storing the scale logarithms allows for the dynamical change of factorisation and renormalisation scales, and it also allows for a change of PDF set. This makes the calculation of scale and PDF uncertainties tractable. As an additional advantage these event files can facilitate the communication of high multiplicity processes between theorists and experimenters. While the generation of the theoretical predictions is typically fairly automated, the process of verifying the results requires more knowledge of the tool used to produce the result and experience identifying potential problems. This know-how is “factored out” when using event files as the verification process can be done once and for all by the authors of the tool. The event files can then be safely used in different contexts.

All these advantages come at a cost: a trade-off in storage space versus generality. In order to use a set of events one has to make sure that they have been generated with cuts more inclusive than those of the analysis one aims at performing. These cuts include the minimum jet transverse momentum for the jet clustering, the jet clustering parameters or some leptonic quantities if leptons are part of the final state. One is tempted to generate the event files with a low threshold on the jet transverse momenta but this then means that the fraction of higher jet transverse momenta is lower and the analysis would require much higher statistics if the generation and actual analysis cuts are very different. In many cases it can be more practical to provide event files with different cuts on the jet momenta rather than trying to provide enough statistics for all analyses with a single set.

A complication with the generation of NLO event files is that a jet algorithm (or a set of them) has to be fixed at generation level. The reason for this is that at NLO one cannot be inclusive for all jet algorithms. To illustrate the problem we can start from the configuration in the middle of Fig.1. For the chosen jet algorithm parameters we have three jets above the transverse momentum threshold represented by the horizontal line. If one decreases the jet radius one jet composed of two partons can be resolved and the two resolved jets might drop below the jet momentum threshold, and as a consequence the event would not be counted in the sample with the original number of jets. On the other hand if one increases the radius of the jet definition we might end up clustering two of the original jets together and so lose a jet, with the event dropping out of the sample. It is therefore

clear that if either of the larger or smaller radius setting had been used for event generation, the event would have been rejected.

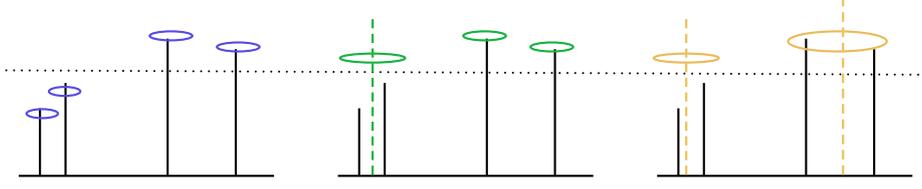


Figure 1: Two situations where jets get lost. The vertical lines represent the transverse momentum of the partons, the ellipses represent the group of partons clustered together by the jet algorithm. The dotted horizontal line represents the jet transverse momentum cut and when several partons are combined into one jet its combined transverse momentum is represented by a dashed vertical line. The three picture show the same partons clustered with three different jet radii, leading to different number of jets above the transverse momentum cut: two for the smallest (left) and largest (right) radii, and three for the intermediate jet radius (middle).

Another complication arises when the NLO corrections are calculated using a subtraction method. By construction the real events and the associated subtraction counter-events have weights which are very anti-correlated as infrared limits are approached. Since the absolute value of the weights get very large in this limit and only their difference has a physical meaning, we will get an over-estimation of the statistical integration error if we use the traditional formula for the variance,

$$\begin{aligned}\sigma^2 &= \langle (\omega - \langle \omega \rangle)^2 \rangle = \langle \omega^2 \rangle - (\langle \omega \rangle)^2 \\ &\simeq \frac{\sum \omega_i^2}{N} - \left(\frac{\sum \omega_i}{N} \right)^2 = \frac{1}{N} \left(\sum \omega_i^2 - \frac{(\sum \omega_i)^2}{N} \right).\end{aligned}$$

In order to understand the problem, one can look at a situation where all real matrix elements and subtractions come in pairs of approximate equal size $\omega_+ \simeq -\omega_- \simeq L$ and that their difference (which is physically relevant) is of order s . The error estimates is then

$$\sigma^2 \simeq \frac{1}{N} \left(NL^2 + \frac{(Ns)^2}{N} \right) = L^2 - s^2 \simeq L^2 \quad (2.1)$$

This overestimates the variance. We get a more realistic estimate by summing the two anti-correlated contributions before squaring

$$\sigma^2 \simeq \frac{1}{N} \left(\sum (\omega_{i,+} + \omega_{i,-})^2 - \frac{(\sum \omega_i)^2}{N} \right) \simeq \langle s^2 \rangle - \langle s \rangle^2 \simeq s^2. \quad (2.2)$$

A long-standing limitation of the RIVET [8] framework for experimental analysis and Monte-Carlo validation was the lack of support for this procedure to estimate the statistical uncertainties in the presence of correlated events. This limitation will be lifted in a next release of RIVET. With the support of this feature n -tuple event files can be used directly to perform a RIVET analysis. In the next release of the `nTupleReader` library this facility will be available and a C++ program as short as that listed in Fig. 2 can be run directly on n -tuple files, greatly simplifying the use of n -tuple files.

```

int main(int argc, char *argv[]){

Rivet::AnalysisHandler rivet;
Rivet.addAnalysis(argv[1]);

nTupleReader r;

for (int ii=2;ii<argc;ii++){
  r.addFile(argv[ii]);
}

r.setPDF("CT10nlo.LHgrid");
r.setCMSEnergy(7000);
r.setColliderType(pp);

while(r.nextEntry()){
  HepMC::GenEvent evt;
  r.setHepMC(evt);
  rivet.analyze(evt);
}

rivet.setCrossSection(r.getCrossSection());
rivet.finalize();
rivet.writeData(std::string(argv[1])+".yoda");
}

```

Figure 2: Simple C++ program to run a generic RIVET analysis using n -tuple files.

While n -tuple files are quite versatile, the time required for an analysis might still be too long. This is the case for example when the analysis is to be used in a PDF fitting procedure. In cases where a fixed set of histograms is considered, it is possible to use n -tuple files to produce a fastNLO [9] table¹, and use this table to evaluate the same histograms with different PDFs or different scales. The three ways of calculating a NLO observable (the direct use, n -tuple, fastNLO table) represent different trade-offs between generality and speed. We show them in the diagram of Fig. 3.

3. MiNLO scale setting

The availability of n -tuple files gives us the opportunity to experiment with different schemes for scale setting. One such scheme recently proposed is the MiNLO procedure described in Ref. [10]. The basic idea in this scheme is to re-cluster the final state of an event using a clustering algorithm, and use the nodal scales at each clustering as the scale for one power of the strong

¹or any similar technique.

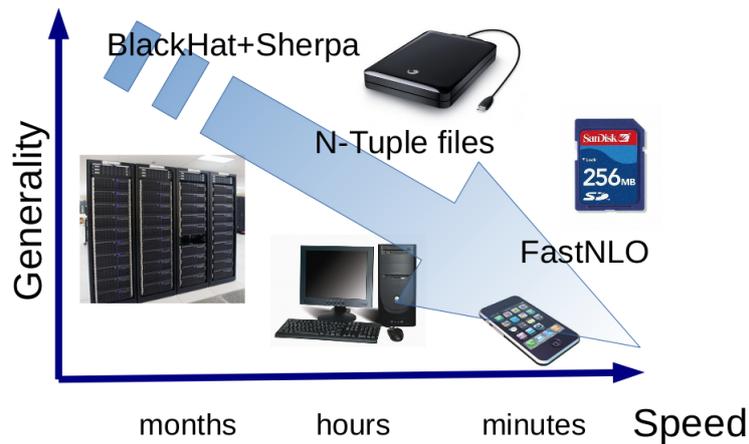


Figure 3: Graphical representation of the speed-generalty trade-off for the different ways of using a NLO calculation.

coupling constant. One also needs to introduce Sudakov form factors to account for the non-emission probability from one nodal scale to the other. The advantage of this method is that it provides scales for the coupling constant that are more physically motivated than a global scale set at the level of the complete final state. We can use the n -tuple files to see the effect of the procedure on observable predictions.

As a first step in the investigation one can look at the ratio between the original weight and the weight according to the MiNLO procedure in a set of events. Here we only consider a leading-order sample for a W boson with two jets. The ratio between the two different scale setting schemes is displayed in the left-hand panel of Fig. 4. The seemingly erratic structure can be explained by separating the sample into sub-samples according to how many clustering happened before the MiNLO procedure stopped. This separation displayed in the right-hand panel of Fig 4 clearly explains the peak structure in the first panel.

4. Conclusion

This contribution describes the advantages of NLO event files and the associated trade-offs. Some new opportunities offered by n -tuple files were presented, showing their potential as a scale setting laboratory.

References

- [1] Z. Bern, L. Dixon, F. Febres Cordero, S. Höche, H. Ita, *et al.*, *Ntuples for NLO Events at Hadron Colliders*, *Comput.Phys.Commun.* **185** (2014) 1443–1460, [[1310.7439](#)].
- [2] C. Berger, Z. Bern, L. J. Dixon, F. Febres Cordero, D. Forde, H. Ita, D. A. Kosower, and D. Maître, *An automated implementation of on-shell methods for one-loop amplitudes*, *Phys. Rev.* **D78** (2008) 036003, [[0803.4180](#)].
- [3] T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, and J. Winter, *SHERPA 1.α: A proof of concept version*, *JHEP* **0402** (2004) 056, [[hep-ph/0311263](#)].

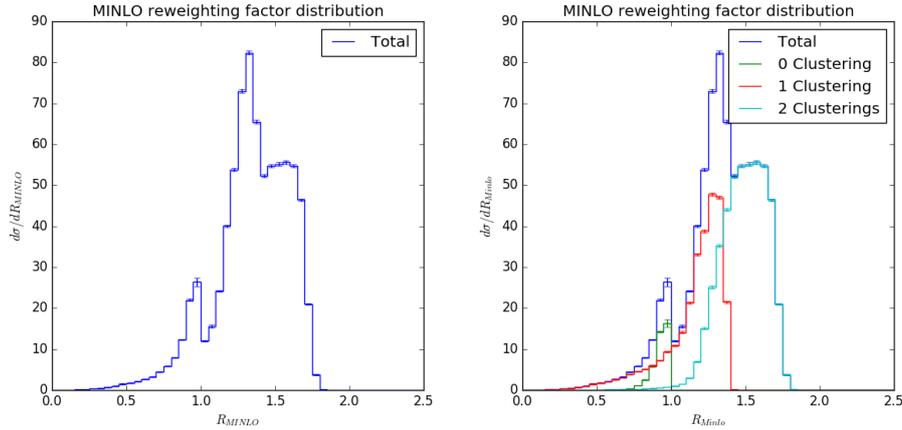


Figure 4: Born contribution to the cross section for $W + 2$ jets as a function of the ratio R_{MINLO} of the original weight and the weight calculated with the MiNLO procedure.

- [4] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, *Event generation with SHERPA 1.1*, *JHEP* **0902** (2009) 007, [[0811.4622](#)].
- [5] F. Krauss, R. Kuhn, and G. Soff, *AMEGIC++ 1.0: A matrix element generator in C++*, *JHEP* **0202** (2002) 044, [[hep-ph/0109036](#)].
- [6] T. Gleisberg and F. Krauss, *Automating dipole subtraction for QCD NLO calculations*, *Eur. Phys. J.* **C53** (2008) 501–523, [[0709.2881](#)].
- [7] R. Brun and F. Rademakers, *ROOT: An object-oriented data analysis framework*, *Nucl. Instrum. Meth.* **A389** (1997) 81–86.
- [8] A. Buckley, J. Butterworth, L. Lönnblad, H. Hoeth, J. Monk, H. Schulz, J. E. von Seggern, F. Siegert, and L. Sonnenschein, *Rivet user manual*, (2010) [[1003.0694](#)].
- [9] **fastNLO** Collaboration, D. Britzger, K. Rabbertz, F. Stober, and M. Wobisch, *New features in version 2 of the fastNLO project*, in *Proceedings, 20th International Workshop on Deep-Inelastic Scattering and Related Subjects (DIS 2012)*, pp. 217–221, 2012. [[1208.3641](#)].
- [10] K. Hamilton, P. Nason, and G. Zanderighi, *MINLO: Multi-Scale Improved NLO*, *JHEP* **10** (2012) 155, [[1206.3572](#)].