Status of Monte-Carlo event generators

Stefan Hoeche
SLAC National Accelerator Laboratory
Menlo Park, CA 94025, USA

SLAC-PUB-15266

A dominant part in the analysis of early LHC data is played by general-purpose Monte-Carlo (MC) event generators [1], which are employed by both experimentalists and theorists to obtain particle level predictions for collider experiments. Hundreds of final-state particles are typically produced in LHC collisions, and the reactions involve both large and small momentum transfer. The high-dimensional phase space and the non-abelian, nonlinear nature of Quantum Chromodynamics (QCD) make an exact solution of the problem impossible. Instead, MC event generators resort to factorization, which allows to split events into different stages, ordered descending in invariant momentum transfer. In this picture, a hard interaction, described through fixed-order perturbation theory, is followed by multiple Bremsstrahlung emissions off initial- and final-state particles and, eventually, by the hadronization process. Each step is simulated independently.

Three general-purpose Monte-Carlo event generators are currently available which implement this paradigm: HERWIG [2], Pythia [3] and Sherpa [4]. A comprehensive description of the physics models implemented in these programs can be found in [1].

Traditionally, multi-purpose event generators compute hard processes at the lowest order in the perturbative expansion. This approximation leads to serious deficiencies in the description of final states with large jet multiplicity. Tree-level matrix element generators have therefore been constructed, which can cope with arbitrary final-states. The most widely used programs nowadays are ALPGEN, AMEGIC, Comix, HELAC and MadGraph [5]. Parton-level events produced by these tools are processed by general-purpose event generators to implement parton showers and hadronization. Although independent programs in principle, matrix-element generators like the above should thus be viewed as an integral part of the simulation chain in general-purpose programs. Their extension to new physics scenarios is handled by FeynRules [6], a Mathematica package, which automatically derives interaction vertices from virtually arbitrary Lagrangians.

Predictions for observables in multi-jet final states involve high powers of the strong coupling, and thus, they have large associated uncertainties. It is therefore desirable to improve the description of high-multiplicity events through next-to-leading order (NLO) calculations. Real and virtual NLO corrections can be combined in an automated way using universal infrared subtraction algorithms [7], which are implemented in various tree-level matrix-element generators [8]. The computation of many

no. jets	ATLAS	LO	ME+PS	NLO	NLO+NP
≥ 2	$620 \pm 1.3^{+110}_{-66} \pm 24$	$958(1)^{+316}_{-221}$	$559(5)$	$1193(3)^{+130}_{-135}$	$1130(19)^{+124}_{-129}$
≥ 3	$43 \pm 0.13^{+12}_{-6.2} \pm 1.7$	$93.4(0.1)^{+50.4}_{-30.3}$	$39.7(0.9)$	$54.5(0.5)^{+2.2}_{-19.9}$	$50.2(2.1)^{+2.0}_{-18.3}$
≥ 4	$4.3 \pm 0.04^{+1.4}_{-0.79} \pm 0.24$	$9.98(0.01)^{+7.40}_{-3.95}$	$3.97(0.08)$	$5.54(0.12)^{+0.08}_{-2.44}$	$5.11(0.29)^{+0.08}_{-2.32}$

Table 1: Total cross sections in nb for jet production at the LHC, using the anti-k_T jet algorithm with $R = 0.4$. ATLAS are compared against LO, ME+PS and NLO theoretical predictions. Numerical integration uncertainties are given in parentheses, the scale dependence is quoted as super- and subscripts. The last column gives NLO results including non-perturbative corrections computed with Sherpa. Uncertainties shown with the ATLAS data are statistical, jet energy scale, and detector unfolding. Table taken from [9].

challenging background processes at the LHC was accomplished with the help of these tools. Prominent examples include $pp \to W/Z$+4 jets, $pp \to$4jets and $pp \to t\bar{t}b\bar{b}$ [9].

A variety of processes can now be computed in a fully automated fashion by linking the matrix element generators described above with dedicated programs for one-loop virtual matrix elements through a standardized interface [10]. Computing virtual corrections often poses the greatest challenge, both because of complexity and numerical stability. Tremendous progress was made in this field, leading to new computational algorithms based on generalized unitarity. Automated calculations of one-loop corrections have since become available in the BlackHat, GoSam, HelacNLO, MadLoop, OpenLoops and Rocket [11,12] programs, as well as several others [13]. Additionally, more traditional, Feynman-diagram based techniques have been extended and applied for example to the process $pp \to W^+W^-b\bar{b}$ [14]. They are also used in the program OpenLoops [12]. Table 1 shows an example of next-to-leading order results for 4 jet production. The calculation was performed with BlackHat and Sherpa, which exemplifies the possible synergy between programs for one-loop calculations and leading order event generators.

While the production of jets in high-energy collisions is typically described very well by fixed-order calculations, the modeling of inner jet structure in this approach is poor. The composition of jets in terms of several partons should therefore be simulated by parton showers, which employ collinear factorization properties of scattering amplitudes to sum leading and certain subleading logarithmic corrections to hard scattering processes. The difference between existing parton-shower implementations in HERWIG [15], Pythia [16] and Sherpa [17] lies in the parametrization of the radiative phase space, the splitting functions which are employed and, in particular, the splitting kinematics.

Sherpa implements a dipole-like parton shower [17], which is based on the Catani-Seymour dipole subtraction method in the large-N_c approximation. The advantage compared to traditional parton showers is an improved description of soft-collinear re-

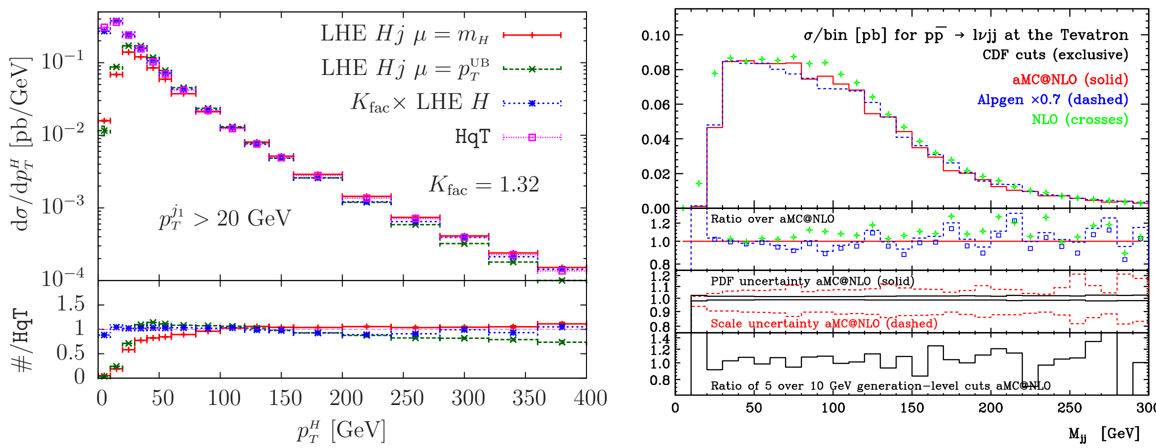

Figure 1: Left: Transverse momentum of the Higgs boson in h+jet events at the LHC (7 TeV). Results from POWHEG simulations with different scale choice are compared against each other and against predictions from HqT [20]. The simulation was performed by combining MadGraph with MCFM and the POWHEG Box. Right: Di-jet mass in W+2jet events at the Tevatron (1.96 TeV). Results from aMC@NLO are compared against predictions from ALPGEN and against an NLO calculation. Figures taken from [21].

gions, which arises as a consequence of the dependence of the splitting function on the kinematics of the spectator parton. Similar ideas are implemented in HERWIG [18]. Within Pythia, recent development focused on improved matching to hard processes at next-to-leading order and on incorporating multiple scattering and rescattering effects into parton shower simulations [16]. First attempts have been made in HERWIG to include all possible color correlations into the parton shower [19].

Higher-order tree-level calculations and parton showers, as introduced above, are two essentially complementary approaches to simulating perturbative QCD interactions in general-purpose Monte-Carlo. It is desirable to combine both, in order to obtain the best possible description of jet production and evolution. To this end, two different strategies have been exploited, which are known as matching and merging.

Matching algorithms either aim at replacing parton-shower splitting operators with the ratio of complete higher-order matrix elements divided by the Born, or they provide means to correct for the difference between the two. The main problem to be solved is that parton showers alter the kinematics of partonic final states. If the underlying parton-level calculation is performed at NLO, this implies that the Born contribution times the parton shower leads to spurious terms of order α_s, which must be subtracted to avoid double counting that would spoil the NLO accuracy. Two universally applicable methods to accomplish this task were suggested in the past, which are dubbed MC@NLO and POWHEG [22]. Both methods were applied to a

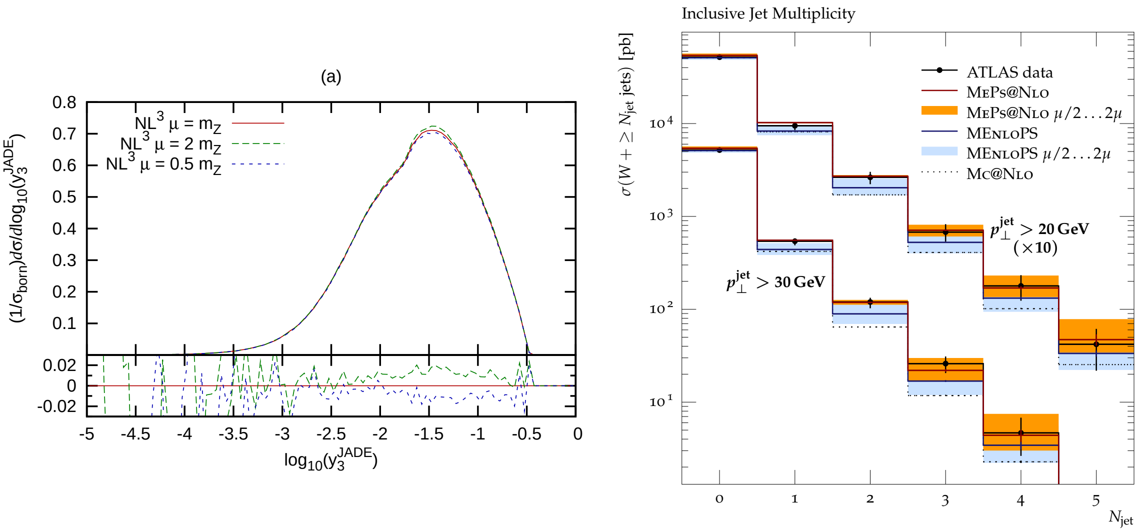

Figure 2: Left: Jade 3→2-jet rate in e^+e^- →hadrons. The renormalization scale dependence of NL^3-merging with 2 and 3-parton processes described at NLO is shown in the ratio plot. Figure taken from [26]. Right: Jet multiplicity in W+jets events at the LHC. The renormalization scale dependence of MEPS@NLO merging with up to W+2 parton processes described at NLO is shown as an orange band. Figure taken from [27].

variety of processes, using the event generation frameworks of HERWIG and Pythia. The MC@NLO method has also been automated in the aMC@NLO framework, based on MadLoop and HERWIG [23] In contrast to MC@NLO, the POWHEG technique does not depend on the parton-shower algorithm, hence, independent implementations exist [24]. Within Sherpa, the POWHEG and MC@NLO methods have been automated [25]. Figure 1 displays results for Higgs boson plus jet and W boson plus two jets production, which are some of the most challenging processes recently implemented using matching methods.

To improve the description of hard QCD radiation by general-purpose event generators, so-called merging algorithms were proposed in the context of the LEP physics program [28], and subsequently extended for hadron collisions [29]. The aim of these techniques is to replace the parton-shower approximation with fixed-order matrix elements for only those partons or parton ensembles, which can be identified with experimentally observed jets. Merging algorithms define an unambiguous way to separate the phase-space of real parton emission into a soft and a hard regime. Soft regions, where higher-order corrections must be resummed, but can be approximated, are filled by the parton shower. Hard regions, where soft and collinear approximations are unsuitable, are filled by fixed-order calculations. Since fixed-order calculations are inclusive, they must be made exclusive using the parton-shower no-branching probability, commonly referred to as the Sudakov factor. In this manner, double-counting of logarithmically enhanced terms is avoided, while sub-leading logarithms and finite corrections are correctly included in the hard domain.

An extension of the original merging approaches, which generically maintains the exact logarithmic accuracy of the parton shower while respecting the phase-space separation cut, was achieved in [30]. The first merging at the leading to next-to-leading order was been presented in [26], in a method based on explicit subtraction of the LO and NLO contributions from the parton shower. The technique introduced in [27] is based instead on an extended modified subtraction similar to MC@NLO, which is implemented using truncated vetoed parton showers in the spirit of [30]. The two existing implementations of a merging method at the NLO in [26] and [27] both indicate a substantial reduction of theoretical uncertainties, as exemplified in Fig. 2.

Monte-Carlo event generators have a variety of free parameters, which can be tuned such that predictions better match experimental data. Many of these paraemters are connected to fragmentation models and underlying-event simulation, or more general, to models for non-perturbative QCD effects. The resulting parameter space can be quite large, which makes it impossible to find an optimal solution by hand.

Two new tools have been developed recently, which attack these problems using a generator-independent validation and tuning strategy. Rivet [31], implements analyses from the LEP, Tevatron and LHC experiments in a common framework and allows simultaneous tests of Monte-Carlo output against all available collider data. Professor [32] employs Rivet to semi-automatically find the best point in the parameter space of the event generator.

In summary, modern general-purpose event generators are highly sophisticated tools for LHC phenomenology. They often implement perturbative QCD calculations at next-to-leading order in the strong coupling and they provide parton showers to include resummation effects. Extensions of event generators allow them to become a platform for testing new physics models and improved descriptions of perturbative QCD in the same framework. Validation and tuning has been in the focus of interest during the first years of LHC running and has been simplified by dedicated tools.

We are indebted to the members of the MCnet network for discussions and input. Support from the US Department of Energy (contract DE–AC02–76SF00515), and from the US LHC Theory Initiative (NSF contract PHY-0705682) is gratefully acknowledged.

References

[1] Ann. Rev. Nucl. Part. Sci. **36** (1986), 253–286; Phys. Rept. **504** (2011), 145–233.

[2] JHEP **01** (2001), 010; Eur. Phys. J. **C58** (2008), 639–707.

[3] JHEP **05** (2006), 026; Comput. Phys. Commun. **178** (2008), 852–867.

[4] JHEP **02** (2004), 056; JHEP **02** (2009), 007.

[5] JHEP **07** (2003), 001; JHEP **02** (2002), 044; JHEP **12** (2008), 039; Eur. Phys. J. **C24** (2002), 447–458; Comput. Phys. Commun. **81** (1994), 357–371.

[6] Comput. Phys. Commun. **180** (2009), 1614–1641.

[7] Nucl. Phys. **B467** (1996), 399–442; Nucl. Phys. **B485** (1997), 291–419; Nucl. Phys. **B627** (2002), 189–265.

[8] Eur. Phys. J. **C53** (2008), 501–523; JHEP **09** (2008), 122; JHEP **10** (2009), 003; JHEP **08** (2009), 085.

[9] Phys. Rev. Lett. **106** (2011), 092001; Phys.Rev. **D85** (2012), 031501; Phys.Rev.Lett. **109** (2012), 042001; JHEP **0909** (2009), 109.

[10] Comput. Phys. Commun. **181** (2010), 1612–1622.

[11] Phys. Rev. **D78** (2008), 036003; Eur.Phys.J. **C72** (2012), 1889; JHEP **0909** (2009), 106; JHEP **1105** (2011), 044; Phys. Rev. **D80** (2009), 094002.

[12] Phys.Rev.Lett. **108** (2012), 111601.

[13] `arXiv:0812.2998` [hep-ph]; Nucl. Phys. **B840** (2010), 214–270; `arXiv:1209.2797` [hep-ph].

[14] Nucl. Phys. **B734** (2006), 62–115; Comput. Phys. Commun. **180** (2009), 2317–2330; Phys. Rev. Lett. **106** (2011), 052001.

[15] JHEP **12** (2003), 045; JHEP **02** (2007), 069.

[16] Eur. Phys. J. **C39** (2005), 129–154; JHEP **01** (2009), 035; Eur. Phys. J. **C69** (2010), 1–18.

[17] JHEP **03** (2008), 038; Phys. Rev. **D81** (2010), 034026.

[18] JHEP **01** (2011), 024.

[19] JHEP **1207** (2012), 042; `arXiv:1206.0180` [hep-ph].

[20] Nucl. Phys. **B737** (2006), 73–120.

[21] JHEP **1207** (2012), 092; JHEP **1202** (2012), 048.

[22] JHEP **06** (2002), 029; JHEP **11** (2004), 040; JHEP **11** (2007), 070.

[23] JHEP **1109** (2011), 061.

[24] JHEP **06** (2010), 043.

[25] JHEP **04** (2011), 024; JHEP **09** (2012), 049.

[26] JHEP **12** (2008), 070.

[27] `arXiv:1207.5031` [hep-ph]; `arXiv:1207.5030` [hep-ph].

[28] Phys. Rev. **D57** (1998), 5767–5772; JHEP **11** (2001), 063; JHEP **05** (2002), 046.

[29] Nucl. Phys. **B632** (2002), 343–362; JHEP **0208** (2002), 015; JHEP **04** (2008), 085; JHEP **01** (2007), 013; Eur. Phys. J. **C53** (2008), 473–500; JHEP **03** (2012), 019.

[30] JHEP **05** (2009), 053; JHEP **11** (2009), 038.

[31] `arXiv:1003.0694` [hep-ph].

[32] Eur. Phys. J. **C65** (2010), 331–357.