MONTE CARLO GENERATORS*

Leif Lönnblad

Department of Theoretical Physics, Lund University
22362 Lund, Sweden

(Received October 12, 2004)

I discuss some recent development of Monte Carlo event generators and point out some problems. Starting with the issue of combining matrix element generators and parton shower algorithms, I will continue with discussing the problems associated with describing small-x final states and discuss some recent model developments for describing multiple partonic scatterings and underlying events in hadron collisions. Finally I will briefly present the ThePEG project of creating a general platform in `C++` for implementing Monte Carlo event generators.

PACS numbers: 12.38.Aw, 12.38.Bx, 13.85.Hd, 13.87.Ce

1. Introduction

Monte Carlo Event Generators (EGs) have developed into essential tools in High Energy Physics. Without them it is questionable if it at all would be possible to embark on large scale experiments such as the LHC. Although the current EGs work satisfactorily, the next generation of experiments will substantially increase the demands both on the physics models implemented in the EGs and on the underlying software technology. In this talk I will discuss some recent developments addressing both of these aspects.

LHC is, of course, mainly a machine for discovering new physics. But irrespectively of what new phenomena may exist, we know for sure that the LHC events will contain huge numbers of hadrons, and that a large fraction of these events will have many hard jets produced by standard QCD processes. Such events are interesting in their own right, but they are also important backgrounds for almost any signal of new physics. Unfortunately the standard Parton Shower (PS) based EGs of today are not well suited to describe events with more than a couple of hard jets. The alternative

* Presented at the XXXIV International Symposium on Multiparticle Dynamics, Sonoma County, California, USA, July 26–August 1, 2004.

is to use fixed-order matrix element (ME) generators, which typically can generate up to six hard partons. But these generators are not well suited for describing the conversion of these hard partons into jets. Several attempts to combine PS and ME generators have been presented, and below I will describe how the PS generator Ariadne [1] is interfaced to the ME generator MadGraph [2] to simulate W production with several associated hard jets in hadronic collisions, possibly the single most important background to signals of new physics at LHC.

It is worth noting that W production at LHC is a small-x process, with typical $x \sim m_W/\sqrt{S} \sim 0.005$, and one of the most important lessons from HERA is that we do not understand event structures at small-x. As described below, none of the available EGs can satisfactorily describe small-x events, especially in the forward region. This is, of course, a problem in general for LHC where the bulk of events are at small scales as compared to the total energy, *ie.* small-x. Also events containing a large scale process will be contaminated by the so-called underlying event, consisting of several much softer sub-processes with correspondingly smaller x. Some models for describing these secondary scatterings will be discussed below.

The current EGs are typically written in Fortran and their basic structure was designed almost two decades ago. Meanwhile there has been a change in programming paradigm, towards object oriented methodology in general and `C++` in particular. This applies to almost all areas of high-energy physics, but in particular for the LHC experiments, where all detector simulation and analysis is based on `C++`. When designing the next generation of EGs it is, therefore, natural to use `C++`. Below I will briefly describe the ThePEG [3] project for designing a general framework in `C++` for implementing EG models, and Pythia7 which is a complete re-implementation of the Lund family of EGs using that framework.

2. Matching parton showers and matrix elements

With ME generators it is possible to generate a handful (typically up to six) hard partons according to exact tree-level matrix elements. But to get properly generated events it is important to interface these to realistic hadronization models, which requires that also soft and collinear partons are generated according PS models to get reliable predictions for the intra- and inter-jet structure. To add on PS to an event from a ME generator, it is important to avoid double counting. Hence the PS must be *vetoed* to avoid generating parton emissions above the cutoff needed to avoid divergencies in the ME generator. In addition the PS assumes that the emissions are ordered in some evolution variable (scale) and uses Sudakov form factors to ensure that there was no additional emissions with a scale between two generated emissions. The ME generator, of course, have no such ordering since all diagrams are added coherently. However, there is still a need for a cutoff in some scale to regulate soft and collinear divergencies, and to naively add a PS to events from a ME generator will, therefore, give a strong dependence on this cutoff.

A solution to this problem was presented by Catani *et al.*, [4]. This so-called CKKW procedure is based on using a jet reconstruction algorithm on the ME generated event to define an ordering of the emissions and then re-weight the event according to the Sudakov form factors obtained from the reconstructed scales. In this way it was shown that the dependence on the ME cutoff cancels to NNLL accuracy. However, the dependence on the cutoff was still quite visible.

In [5] I presented an improved version of the CKKW procedure implemented in the ARIADNE program. In stead of using a standard jet algorithm to define an ordering, the dipole cascade of ARIADNE is used *backwards*, so that for each ME generated event a likely dipole emission history is found, where complete intermediate states, as well as evolution scales, are reconstructed as if the event had been generated by ARIADNE . These intermediate states are then used to make trial emissions to accept or reject the event, using the fact that the probability of not emitting a parton above the next reconstructed scale is exactly the Sudakov form factor that would have been used, had ARIADNE generated the event. Together with a special treatment of events with the highest parton multiplicity from the ME generator, the dependence on the ME cutoff was in this way much reduced for the e^+e^- annihilation case.

Recently the ARIADNE version of CKKW was implemented also for W production in hadronic collision [6, 7]. Here the situation is complicated by the fact that there are also initial-state PS involved. The procedure is still similar, however. The $W + n$ jet events generated by MADGRAPH are reconstructed to find possible intermediate states corresponding to those which would arise if ARIADNE had generated them, including the possibility to have initial state emissions. The *Sudakov veto* algorithm then works in the same way as above, with trial (initial- and final-state) emissions to determine if an event should be rejected or accepted.

Preliminary results for this procedure is shown in Fig. 1. Here the $W{-}p_\perp$ spectrum is shown for the case when ME generated events with up to only one jet is combined with the ARIADNE dipole shower. For comparison the corresponding result from standard ARIADNE is shown. The latter also contains a ME correction for the first emission [8] (in a way which, however, is not extensible to higher orders), so the curves should in principle agree exactly. Although the curves agree fairly well, there are some differences. The new procedure overshoots the old one near the ME cutoff at 12 GeV,

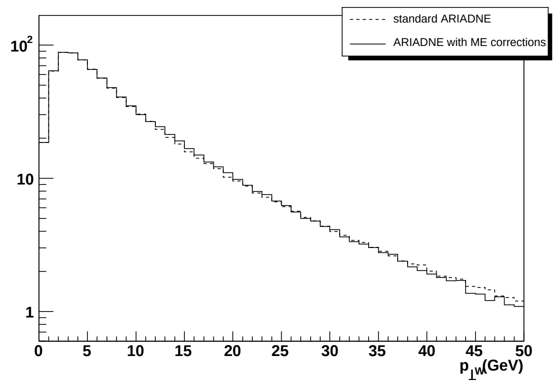

Fig. 1. The $p_\perp$ distribution of W's at the Tevatron. Full line is $W+0$ jet and $W+1$ jet MadGraph events combined with Ariadne. Dashed line is standard Ariadne.

and undershoots a bit at large $p_\perp$. However, we now believe that we understand these differences and in a future publication [7] we will present a more detailed description and investigations using also higher order MEs.

It should be noted that although similar to the CKKW based ME+PS correction of $W+n$jets presented in [9,10], the Sudakov form factors are different. While the standard CKKW uses analytical form factors corresponding to standard DGLAP [11–14] resummation, Ariadne strictly uses the no-emission probability interpretation, which means that some terms proportional to $\log(1/x)$ are also resummed. Especially at the LHC, such terms are expected to be important as $x \sim m_W/\sqrt{S}$ becomes small, and it will be interesting to compare the different procedures to see how large the differences are and if they are diminished by using higher order MEs.

3. Small-x final states and underlying events

Besides the striking small-x rise of F_2, maybe the most interesting measurements to have come from HERA have to do with the hadronic final states at small-x, especially in the forward (proton) direction. In a large fraction of events this region of phase space is completely empty, which has sparked a revitalization of the study of diffraction in general and hard diffraction in particular. On the other hand event without such *rapidity gaps* have much more activity in the forward region than would be expected from standard models of multi-particle production. In fact, none of the standard DGLAP-based PS generators come even close to describe in particular the rate of forward jets. The reason is, of course, that these PS models corresponds to a resummation of large logs of Q^2, while at small-x we need also to worry about large logs if $1/x$.

As mentioned in the previous section, the colour–dipole model [15] in ARIADNE does resume some logs of $1/x$ and it indeed gives a much better description of data. However, it is very difficult to translate this into a formal evolution scheme such as BFKL [16–18] or CCFM [19–22]. Especially CCFM is believed to the appropriate model for describing small-x final states, as it correctly reproduces BFKL in the asymptotic limit but also is similar to DGLAP at larger x and Q^2. In addition it takes special care of gluon coherence by introducing angular ordering.

There are now two different EGs which implement CCFM evolution: CASCADE [23] which is standard CCFM and LDCMC [24] which implements the Linked Dipole Chain (LDC) model [25,26], a reformulation and generalization of CCFM. They both give mutually consistent results and are indeed able to reproduce forward jet rates as measured at HERA. However, they can only do so if non-singular terms in the gluon splitting function are left out, *ie.* using $P_g(z) = 1/z+1/(1-z)$ rather than $P_g(z) = 1/z+1/(1-z)-2+z(1-z)$. So far no satisfactory explanation has been found for these results (see *e.g.* [27] for a review of the subject). The bottom line is that we do not have a good understanding of the structure of small-x events.

Due to the large energy at LHC, the vast majority of events will be at small $x \sim \sqrt{Q^2/S}$, since the typical hard scale, Q^2, is fairly low. Of course, the triggers in the experiments will mainly select events with a rather large scale, since this is where we expect signs of new physics. However, these events will not only be accompanied by overlayed events of minimum-bias type, due to several pp collisions in each bunch crossing, but also in a given collision, a hard partonic sub-process will be accompanied by several softer partonic scatterings, which we need to understand in order to analyze the results in detail.

The simplest way of seeing that there has to be more than one partonic scattering per event is to look at the simple parton–parton cross section above some given cutoff $k_{\perp 0}$

$$\sigma_{\rm hard}(k_{\perp 0}^2) = \int_{k_{\perp 0}^2} dk_\perp^2 \frac{d\sigma_{\rm hard}}{dk_\perp^2}\,, \tag{1}$$

which diverges as $k_{\perp 0} \to 0$ and will exceed the total non-diffractive cross section, $\sigma_{\rm ND}$, even for $k_{\perp 0}$ in the perturbative region of a couple of GeV already at the Tevatron.

The standard interpretation of this, which is used in the standard underlying event models, is that there are several scatterings in each collision with an average number of scatterings given by

$$\langle n\rangle = \frac{\sigma_{\rm hard}(k_{\perp 0}^2)}{\sigma_{\rm ND}}\,. \tag{2}$$

Maybe the most used model for such multiple scatterings is implemented in Pythia [28, 29]. Here also non-perturbative scatterings are included, although they are described by standard partonic $2 \to 2$ matrix elements using a simple soft regularization:

$$\frac{d\sigma_{\rm hard}}{dk_\perp^2} \to \frac{d\sigma_{\rm hard}}{dk_\perp^2} \times \frac{p_{\perp 0}^4}{(p_{\perp 0}^2 + k_\perp^2)^2} \quad \text{and} \quad \alpha_{\rm s}(k_\perp^2) \to \alpha_{\rm s}(k_\perp^2 + p_{\perp 0}^2)\,. \tag{3}$$

Also a double-Gaussian impact-parameter dependence is introduced giving rise to large non-trivial fluctuations in the number of scatterings per collision. Although this is a seemingly simple model (there is a lot of tricky details in the algorithm which is beyond the scope of this talk), it does a very good job of describing particle production at *e.g.* the Tevatron (see *e.g.* [30]).

Recently, the multiple scattering model in Pythia reached another level of sophistication [31,32], introducing among other things so-called junction-strings in the case more than one valens quark in a proton has participated in scatterings, and a new way of connecting the colours in the scatterings to form strings to the remnants. Also a new multiple scattering procedure based on similar underlying assumptions, Jimmy [33], has been introduced to the herwig [34] generator. All such models have one big problem, namely the strong dependence on the cutoff $k_{\perp 0}$ (or the soft regularization parameter $p_{\perp 0}$ in Pythia). In addition, this cutoff seems to depend on energy in a basically unknown way, making the predictions for LHC fairly uncertain.

Another more theoretical problem is the fact that the scatterings are treated with collinear factorization, even though they are very soft and hence have very small-x, where we know we cannot describe the final state with standard DGLAP-based EGs. There are no working multiple scattering programs based on CCFM evolution yet, but there is a suggestion for treating the exchange of multiple chains within the framework of the LDC model [35]. Preliminary results for this model show a remarkable stability w.r.t. the soft cutoff.

4. ThePEG /Pythia7 /herwig++

ThePEG is a general platform written in `C++` for implementing models for event generation. It is made up from the basic model-independent parts of Pythia7 [36, 37], the project of rewriting the Lund family of EGs in `C++`. When the corresponding rewrite of the herwig program [34] started it was decided to use the same basic infrastructure as PYTHIA7 and, therefore, the THEPEG was factorized out of PYTHIA7 and is now the base of both PYTHIA7 and HERWIG++ [38].

THEPEG uses CLHEP [39] and adds on a number of general utilities such as smart pointers, extended type information, persistent I/O, dynamic loading and some extra utilities for kinematics, phase space generation *etc.*

The actual event generation is then performed by calling different *handler* classes for hard partonic sub-processes, parton densities, QCD cascades, hadronization *etc.* To implement a new model to be used by THEPEG , the procedure is then to write a new `C++` class inheriting from a corresponding handler class and implement a number of pre-defined virtual functions. *E.g.* a class for implementing a new hadronization model would inherit from the abstract `HandronizationHandler` class, and a class for a new parton density would inherit from the `PDFBase` class.

To generate events with THEPEG one first runs a setup program where an *EventGenerator* object is set up to use different models for different steps of the generation procedure. All objects to be chosen from are stored in a *repository*, within which it is also possible to modify switches and parameters of the implemented models in a standardized fashion, using so called *interface* objects. Typically the user would choose from a number of pre-defined `EventGenerator` objects and only make minor changes for the specific simulation to be made. When an `EventGenerator` is properly set up it is saved persistently to a file which can then be read into a special run program to perform the generation, in which case special `AnalysisHandler` objects may be specified to analyze the resulting events. Alternatively it can be read into *e.g.* a detector simulation program where it can be used to generate events.

Currently, THEPEG and PYTHIA7 are available through their respective web pages [3, 37] and include some basic $2 \to 2$ matrix elements, a couple of PDF parameterizations, remnant handling, initial- and final-state parton showers, Lund string fragmentation and particle decays. Also HERWIG++ is available [38].

The future development for THEPEG is mainly to improve the documentation. For PYTHIA7 the work continues to re-implement models from the old Fortran PYTHIA version. In addition the ARIADNE program will be re-implemented using the THEPEG platform.

REFERENCES

[1] L. Lönnblad, *Comput. Phys. Commun.* **71**, 15 (1992).

[2] F. Maltoni, T. Stelzer, *J. High Energy Phys.* **02**, 027 (2003).

[3] L. Lönnblad *et al.*, “THEPEG program”, `http://www.thep.lu.se/ThePEG`.

[4] S. Catani, *et al. J. High Energy Phys.* **11**, 063 (2001).
[5] L. Lönnblad ,*J. High Energy Phys.* **05**, 046 (2002),
[6] N. Lavesson, Masters thesis, Theoretical Physics, Lund University.
[7] N. Lavesson, L. Lönnblad, preprint in preparation.
[8] L. Lönnblad, *Nucl. Phys.* **B458**, 215 (1996).
[9] F. Krauss, *J. High Energy Phys.* **08**, 015 (2002).
[10] S. Mrenna, P. Richardson, *J. High Energy Phys.* **05**, 040 (2004),
[11] V.N. Gribov, L.N. Lipatov, *Yad. Fiz.* **15**, 781 (1972).
[12] L.N. Lipatov, *Sov. J. Nucl. Phys.* **20**, 94 (1975).
[13] G. Altarelli, G. Parisi, *Nucl. Phys.* **B126**, 298 (1977).
[14] Y.L. Dokshitzer, *Sov. Phys. JETP* **46**, 641 (1977).
[15] B. Andersson *et al.*, *Z. Phys.* **C43**, 625 (1989).
[16] E.A. Kuraev *et al.*, *Sov. Phys. JETP* **44**, 443 (1976).
[17] E.A. Kuraev *et al.*, *Sov. Phys. JETP* **45**, 199 (1977).
[18] I.I. Balitsky, L.N. Lipatov, *Sov. J. Nucl. Phys.* **28**, 822 (1978).
[19] M. Ciafaloni, *Nucl. Phys.* **B296**, 49 (1988).
[20] S. Catani, F. Fiorani, G. Marchesini, *Phys. Lett.* **B234**, 339 (1990).
[21] S. Catani, F. Fiorani, G. Marchesini, *Nucl. Phys.* **B336**, 18 (1990).
[22] G. Marchesini, *Nucl. Phys.* **B445**, 49 (1995).
[23] H. Jung, *Comput. Phys. Commun.* **143**, 100 (2002).
[24] H. Kharraziha, L. Lönnblad, *J. High Energy Phys.* **03**, 006 (1998).
[25] B. Andersson *et al.*, *Nucl. Phys.* **B467**, 443 (1996).
[26] B. Andersson *et al.*, *Phys. Rev.* **D57**, 5543 (1998).
[27] B. Andersson *et al.*, *Eur. Phys. J.* **C25**, 77 (2002).
[28] T. Sjöstrand, M. van Zijl, *Phys. Rev.* **D36**, 2019 (1987).
[29] T. Sjöstrand *et al.*, *Comput. Phys. Commun.* **135**, 238 (2001).
[30] R. Field, *Acta Phys. Pol. B* **36**, 167 (2005).
[31] T. Sjöstrand, P. Z. Skands, *J. High Energy Phys.* **03**, 053 (2004).
[32] T. Sjöstrand, P. Z. Skands, `hep-ph/0408302`.
[33] J. Butterworth *et al.*, `http://jetweb.hep.ucl.ac.uk/JIMMY`.
[34] G. Corcella *et al.*, *J. High Energy Phys.* **01**, 010 (2001).
[35] G. Gustafson *et al.*, *Phys. Rev.* **D67**, 034020 (2003).
[36] M. Bertini *et al.*, *Comput. Phys. Commun.* **134**, 365 (2001).
[37] L. Lönnblad *et al.*, "Pythia7 program", `http://www.thep.lu.se/Pythia7`.
[38] S. Gieseke *et al.*, *J. High Energy Phys.* **02**, 005 (2004).
[39] L. Lönnblad, *Comput. Phys. Commun.* **84**, 307 (1994).