DESY 11-146
KA-TP-22-2011
MCnet-11-19

The Sudakov Veto Algorithm Reloaded

Simon Plätzer[1] and Malin Sjödahl[2]

[1] DESY, Notkestrasse 85, D-22607 Hamburg, Germany
[2] Institut für Theoretische Physik, KIT, D-76128 Karlsruhe, Germany

March 20, 2012

Abstract. We perform a careful analysis of the main Monte Carlo algorithm used in parton shower simulations, the Sudakov veto algorithm. We prove a general version of the algorithm, directly including the dependence on the infrared cutoff. Taking this as a starting point, we then consider non-positive definite splitting kernels, as encountered when dealing with sub-leading colour correlations or splitting kernels beyond leading order. New algorithms suited for these situations are developed.

PACS. 02.70.Tt Monte Carlo methods – 12.38.Cy Summation of QCD perturbation theory

1 Introduction

Parton shower Monte Carlo simulations as implemented in for example [1–3], are indispensable tools for analyzing and predicting realistic final states encountered in collider experiments. Matrix element corrections, as discussed in [4–8], the technically similar matching to NLO calculations employing the POWHEG method [9], or schemes to combine parton showers and multijet tree-level matrix elements [10–14], all rely directly or indirectly on the same method for generating subsequent parton shower emissions in Monte Carlo simulations.

With the notable exception of the FORTRAN version of HERWIG, nowadays most parton shower implementations use the Sudakov veto algorithm to facilitate this task, as the splitting kernels normally are too complicated to allow efficient integration.

A justification of the Sudakov veto algorithm is given in [4], stating that for upper bounds R on splitting kernels P, $R(q) \geq P(q)$ for all q, algorithm (1) will draw random variables with density

$$\mathrm{d}T_P(q|Q) = \theta(Q-q)P(q)\Delta_P(q|Q)\mathrm{d}q \ , \qquad (1)$$

where the Sudakov form factor is given by

$$\Delta_P(q|Q) = \exp\left(-\int_q^Q P(t)\mathrm{d}t\right) \ . \qquad (2)$$

We note here, however, that the algorithm has to be more carefully defined. Most obviously if, in algorithm (1), $P(q) = 0$ but $R(q) \neq 0$ for all $q \leq q_c$ and some q_c, the algorithm will potentially enter an infinite loop. We shall therefore assume that $R(q)$ is suitably restricted to avoid this situation, making the algorithm well-defined in the sense that it will never hit a state in which it will not terminate with probability one.

Algorithm 1 The Sudakov veto algorithm as quoted in the literature.

```
Q' ← Q
loop
  Draw q with density
        θ(Q' − q)R(q)Δ_R(q|Q')dq .
  return  q with probability P(q)/R(q)
  Q' ← q
end loop
```

Literally implementing the algorithm as presented above will not generate the desired density owing to the fact that $\mathrm{d}T_P$ is not a probability density,

$$\int_q^Q \frac{\mathrm{d}T_P(t|Q)}{\mathrm{d}t}\mathrm{d}t = 1 - \Delta_P(q|Q) \neq 1 \ . \qquad (3)$$

At best the algorithm will approximate the target density if, for the lowest possible q, $\Delta_P(q|Q) \ll 1$. In practice, however, a vanishing Δ_P will never be encountered in parton shower simulations, due to the fact that an infrared cutoff $\mu \geq 0$ is always present. Thus the typically divergent part of the splitting kernel at $q = 0$ is never reached, and the no-emission probability remains, $\Delta_P(\mu|Q) > 0$.

Similarly, the competing processes algorithm

```
Draw {q_i, ..., q_n} from dT_{P_i}(q_i|Q), i = 1, ..., n
return  max({q_i, ..., q_n})
```

targeting at drawing random variables with density $\mathrm{d}T_P$, $P = \sum_i P_i$, will not produce the desired result for the same reason.

2 The Complete Algorithm

The failure of the simple algorithm presented in the previous section has been argued to originate from the fact that the density considered is not a probability density.

However, the density considered in the previous section is also not what is typically aimed at in a parton shower implementation. (See *e.g.* [15] for a concise treatment.) This can be seen by the fact that a lower cutoff scale μ has not been specified, nor is a virtual no-emission contribution present. Owing to the fact that a parton shower is to preserve the total inclusive cross section, the combined density, including both emission and no-emission, has to be a probability density. As the probability to not emit between two scales is determined by the Sudakov form factor, the *probability* density which we are interested in is

$$\frac{\mathrm{d}S_P(\mu,q|Q)}{\mathrm{d}q} = \Delta_P(\mu|Q)\delta(q-\mu)+ \theta(Q-q)\theta(q-\mu)P(q)\Delta_P(q|Q) \ , \quad (4)$$

which relates to the previously introduced density as

$$\frac{\mathrm{d}S_P(\mu,q|Q)}{\mathrm{d}q} = \Delta_P(\mu|Q)\delta(q-\mu) + \theta(q-\mu)\mathrm{d}T_P(q|Q) \ . \quad (5)$$

Using sampling by inversion, [1]

$$\int_0^q \frac{\mathrm{d}S_P(\mu,t|Q)}{\mathrm{d}t}\mathrm{d}t = \Delta_P(q|Q)\theta(q-\mu) = \mathbf{rnd} \ , \quad (6)$$

we find the equation to be solved for the next scale q. This is similar to what one would expect by viewing the Sudakov form factor $\Delta_P(q|Q)$ as a no-emission probability between two scales Q and q, but now explicitly taking into account the dependence on the infrared cutoff μ.

As the splitting kernel, P, is not normally easily integrated, what is used in actual implementations is instead typically a version of the Sudakov veto algorithm where $\Delta_R(q|Q) = \mathbf{rnd}$ is solved for some easily integrable function $R(q) > P(q)$, and the radiation is kept only with a probability of $P(q)/R(q)$. The issue of how to deal with the fact that the Sudakov factor $\Delta_R(\mu|Q) \neq 0$, however, remains.

In the typically encountered case that $P(q)$ is divergent at an absolute lower bound (which we take to be $q = 0$), the problem with the non-vanishing Sudakov factor at the lowest physically considered bound ($q = \mu$) can be circumvented by integrating down to $q = 0$. Events which only have emissions below the lowest physical bound (μ) are then regarded as no-emission events [16,17]. This is guaranteed to work as for such splitting kernels $\Delta_P(0|Q) = 0$.

However, for splittings of massive particles it is the case, that - even if the splitting kernel is integrated down to 0 - the corresponding Sudakov factor is not vanishing, $\Delta_P(0|Q) \neq 0$. This situation can be dealt with by using an overestimation function $R(q)$ which does correspond to $\Delta_R(0|Q) = 0$. The approximation of a non-divergent splitting kernel with a divergent one is, however, likely to lead to a severe overestimate, i.e. $R(q) \gg P(q)$, which significantly influences the efficiency of the algorithm. Alternatively, we here suggest that algorithm (2) can be used, both for divergent and non-divergent splitting kernels.

[1] In this paper **rnd** denotes a source of uniformly distributed random numbers on $[0, 1)$.

Algorithm 2 The alternative Sudakov veto algorithm.

```
Q' ← Q
loop
  solve rnd= Δ_R(q|Q')θ(q − μ) for q
  if q = μ then
    return μ
  else
    return q with probability P(q)/R(q)
  end if
  Q' ← q
end loop
```

We claim that this algorithm will correctly produce $\mathrm{d}S_P(\mu,q|Q)$ for all chosen boundaries $\mu \leq q < Q$. To prove it, we first prove theorem (1).

Theorem 1 *The q-density produced by the Sudakov veto algorithm after n rejection steps and a final acceptance step is given by*

$$\frac{\mathrm{d}S^{(n)}_{veto}(\mu,q|Q)}{\mathrm{d}q} = \Delta_R(\mu|Q)\delta(q-\mu)\Delta^{(n)}_{P-R}(\mu|Q)+ \theta(Q-q)\theta(q-\mu)P(q)\Delta_R(q|Q)\Delta^{(n)}_{P-R}(q|Q) \quad (7)$$

where

$$\Delta^{(n)}_{P-R}(q|Q) = \frac{1}{n!}\left(-\int_q^Q (P(k)-R(k))\,\mathrm{d}k\right)^n \ . \quad (8)$$

From this the correctness of the algorithm follows upon summing over any number of rejection steps $n = 0$ to ∞, and the usage of $\Delta_R(q|Q)\Delta_{P-R}(q|Q) = \Delta_P(q|Q)$. Note that theorem (1) does include the density of non-radiating events, and that each time the loop in algorithm (2) is entered, an event q is drawn from $\mathrm{d}S_R$ by virtue of eq. (6).

We will show theorem (1) using induction and therefore start by noting that the probability to accept an event, starting at an intermediate scale k, is given by

$$\mathrm{d}S^{\mathrm{accept}}(\mu,q|k) = \Delta_R(\mu|k)\delta(q-\mu)+ \theta(k-q)\theta(q-\mu)P(q)\Delta_R(q|k) \ , \quad (9)$$

where the first term reflects the fact that proposal events at the infrared cutoff are always accepted, while the second term accounts for proposal events above the cutoff being accepted with probability $P(q)/R(q)$. For $n = 0$ the intermediate scale k equals the starting scale Q, *i.e.*

$$\mathrm{d}S^{(0)}_{\mathrm{veto}}(\mu,q|Q) = \mathrm{d}S^{\mathrm{accept}}(\mu,q|Q) \ , \quad (10)$$

proving eq. (7) for $n = 0$. If the algorithm had performed one rejection step, events could only have been proposed above the infrared cutoff (otherwise the algorithm would have terminated), and we have

$$\mathrm{d}S^{(1)}_{\mathrm{veto}}(\mu, q|Q) = \int_\mu^Q \mathrm{d}S^{\mathrm{accept}}(\mu, q|k)\,(R(k) - P(k))\,\Delta_R(k|Q)\mathrm{d}k = \int_\mu^Q \mathrm{d}S^{(0)}_{\mathrm{veto}}(\mu, q|k)\,(R(k) - P(k))\,\Delta_R(k|Q)\mathrm{d}k \ . \quad (11)$$

Here, the factor of $R(k) - P(k)$ originates from the veto probability, $1 - P(q)/R(q)$, times the kernel $R(q)$ which had been used for the proposed event.

To arrive at the desired density in eq. (7) we use the 'chain' property of the Sudakov form factors,

$$\Delta_R(q|k)\Delta_R(k|Q) = \Delta_R(q|Q) \ , \quad (12)$$

and the relation

$$\Delta^{(1)}_{P-R}(q|Q) = \int_q^Q (R(k) - P(k))\,\mathrm{d}k \ . \quad (13)$$

This proves eq. (7) for $n = 1$. In general,

$$\mathrm{d}S^{(n+1)}_{\mathrm{veto}}(\mu, q|Q) = \int_\mu^Q \mathrm{d}S^{(n)}_{\mathrm{veto}}(\mu, q|k)\,(R(k) - P(k))\,\Delta_R(k|Q)\mathrm{d}k \quad (14)$$

reflecting an initially proposed event k below Q, which initiated a sequence of n veto steps and a final acceptance step. Thus, if the theorem was correct for some $n > 0$, we readily obtain the claimed result for $n + 1$ upon using

$$\frac{1}{n!}\int_q^Q \left(\int_q^k f(k')\mathrm{d}k'\right)^n f(k)\mathrm{d}k = \frac{1}{(n+1)!}\left(\int_q^Q f(k)\mathrm{d}k\right)^{n+1} \ . \quad (15)$$

The competing processes algorithm in turn reads

Draw $\{q_i, ..., q_n\}$ from $\mathrm{d}S_{P_i}(q_i|Q)$, $i = 1, ..., n$
return $\max(\{q_i, ..., q_n\})$

---

which is easily proven as $\mathrm{d}S_{P_i}(q_i|Q)$ now is a true probability density.

3 Towards Splitting Kernels of Indefinite Sign

For the remainder of this note we shall be concerned with seeking solutions to the case of non-positive definite splitting kernels. For potentially negative-valued 'densities' $D(x)$, a Monte Carlo implementation is still sensible by sampling events x according to $|D(x)|$ and afterwards assigning weights $+1$ or -1, depending on whether $D(x) > 0$ or $D(x) < 0$, however, the generalization of the Sudakov veto algorithm is not obvious.

In this section we will outline an algorithm, algorithm (3), which is able to deal with the general case of non-positive definite splitting kernels, but is limited to considering distributions at fixed starting scale Q. That this is a limitation can be seen from the fact the the generated density will multiply a Q-dependent normalization factor smaller than one. As long as only one starting scale is considered, this scale dependence can trivially be normalized away. However, in the case of varying scales, one would have to introduce scale dependent event weights larger than one. For the case of an unlimited number of emissions driven by subsequently sampling the density at varying scales, there is clearly no upper bound for the combined size of these weights. The algorithm presented here could, however, be of practical interest for cases where splitting kernels P of indefinite sign are present only for a limited number of emissions. Such scenarios would indeed give rise to an upper bound on the expected event weight; particularly one could consider matrix element corrections incorporating higher order corrections with the need for appropriate subtractions to regularize infrared divergences.

To be precise, we decompose the non-positive definite kernel $P(q)$ as $P^+(q) - P^-(q)$, where

$$P^\pm(q) = \begin{cases} \pm P(q) & : \ P(q) \gtrless 0 \\ 0 & : \ \text{otherwise} \end{cases} \quad (16)$$

and utilize algorithm (3).

Algorithm 3 The algorithm for splitting kernels of indefinite sign. See text for the definition of $P_\pm$.

```
loop
  Draw q+ from dS_{P+}(μ, q|Q)
  Draw q− from dS_{P−}(μ, q|Q)
  q ← max(q+, q−)
  if q = μ then
    return μ with weight +1
  end if
  Draw t from dS_{2P−}(μ, t|q)
  if t = μ then
    if max(q+, q−) = q+ then
      return q with weight +1
    else
      return q with weight −1
    end if
  end if
end loop
```

Note that all random variables needed from Sudakov-type distributions are readily generated using the veto algorithm as outlined above. We claim that the algorithm will generate

$$\mathrm{d}S_P(\mu, q|Q) \times \Delta^2_{P^-}(\mu|Q) \ .$$

To prove this, note that if $q_+ = q_- = \mu$ we obtain a contribution

$$\delta(q-\mu)\Delta_{P^+}(\mu|Q)\Delta_{P^-}(\mu|Q) = \delta(q-\mu)\Delta_P(\mu|Q)\Delta^2_{P^-}(\mu|Q)\ .$$

The probability for $t = \mu$, *i.e.* not to re-enter the loop is clearly given by $\Delta^2_{P^-}(\mu|q)$. Then, if $q_+ > q_-$ (and hence $q_+ > \mu$), we find a contribution

$$\theta(q-\mu)\mathrm{d}T_{P^+}(q|Q)\Delta_{P^-}(q|Q)\Delta^2_{P^-}(\mu|q) = \theta(q-\mu)\mathrm{d}T_{P^+}(q|Q)\Delta_{-P^-}(q|Q)\Delta^2_{P^-}(\mu|Q)\ .$$

Finally, if $q_- > q_+$ (and hence $q_- > \mu$), while including the negative weight for these events, the last contribution is

$$-\theta(q-\mu)\mathrm{d}T_{P^-}(q|Q)\Delta_{P^+}(q|Q)\Delta^2_{P^-}(\mu|q) = \theta(q-\mu)\mathrm{d}T_{-P^-}(q|Q)\Delta_{P^+}(q|Q)\Delta^2_{P^-}(\mu|Q)\ ,$$

completing the proof. For the case of several available processes we can always decompose

$$P(q) = \sum_i P_i(q) = \sum_i P_i^+(q) - \sum_i P_i^-(q)\ , \qquad (17)$$

such that the proposal events $q_\pm$, as well as the 'control variate' t may be generated using the competing processes algorithm for the individual positive and negative contributions,

$$P^\pm(q) = \sum_i P_i^\pm(q)\ . \qquad (18)$$

4 Interleaving Vetoing and Competition

The algorithm outlined in the previous section may be used to deal with the case of non-positive definite splitting kernels in full generality provided we are interested in distributions for a single starting scale Q, or are prepared to accept potentially large weights. For practical purposes, we are, however, interested in cascades at subsequent scales $q_1 > q_2 > ... > q_n$, where q_{k-1} serves as the starting scale of the distribution for q_k. The Q-dependent normalization $\Delta^2_{P_-}$ present in the distribution generated will thus make it non-ideal in the context of cascades.

Here we consider the typically encountered physical setup for which we may assume that

$$P(q) = \sum_i P_i(q) > 0\ , \qquad (19)$$

still allowing for a probabilistic interpretation, though a subset of the splitting kernels are of indefinite sign. $P(q)$ can be decomposed as in eq. (17), and we can directly identify an overestimate to the desired splitting kernel,

$$P^+(q) \geq P(q) = P^+(q) - P^-(q)\ . \qquad (20)$$

Algorithm 4 The interleaved veto/competition algorithm.

```
Q' ← Q
loop
  Draw {q_i, ..., q_n} from dS_{P_i^+}(q_i|Q), i = 1, ..., n
  q ← max({q_i, ..., q_n})
  if q = μ then
    return μ
  else
    return q with probability (P^+(q) − P^−(q))/P^+(q)
  end if
  Q' ← q
end loop
```

This suggests a two-step procedure of interleaving competing processes and vetoing, formalized in algorithm (4), which we choose to call the 'interleaved veto/competition algorithm'. Here, the q_i may be generated directly, if the P_i^+ allow to. Alternatively the veto algorithm may be used with overestimates $R_i^+(q) \geq P_i^+(q)$. The correctness of the complete algorithm is seen by the fact that the first two instructions in the loop will guarantee that q is distributed according to $\mathrm{d}S_{P^+}$ by the competing process algorithm. In the following steps, the obtained density $P^+(q)$ is corrected to $P(q) = P^+(q) - P^-(q)$ by virtue of the standard veto algorithm. Note that this algorithm will neither require negative weights, or introduce a Q-dependent normalization.

5 Conclusions and Outlook

We have given a careful analysis of the main Monte Carlo algorithm entering current parton shower simulations, the Sudakov veto algorithm. Especially, we have discussed in detail the importance of the no emission probability arising as a consequence of an infrared cutoff, and suggested an alternative formulation, algorithm (2), which directly includes the dependence on the infrared cutoff. This algorithm is argued to be more efficient in the case of a non-divergent splitting kernel.

We also consider possible extensions to the case of splitting kernels of indefinite sign. Such splitting kernels are encountered when trying to extend parton showers beyond the large N_c limit or beyond leading order.

First, in algorithm (3) we develop a general algorithm for a splitting kernel of indefinite sign. Modulo a normalization dependence on the starting scale of the algorithm, this case may indeed be dealt with in full generality. The Q-dependent normalization, however, prevents efficient usage in the context of cascades using an ordered chain of scales.

For the typically encountered case, in which splitting kernels of indefinite sign are present, but the sum over all possible splitting kernels stays positive, we give, in algorithm (4), an algorithm interleaving the competing process algorithm with subsequent veto steps.

Acknowledgments

We would like to thank Stefan Gieseke, Leif Lönnblad and Torbjörn Sjöstrand for useful discussions. This work has been supported by the European Union Marie Curie Research Training Network MCnet under contract MRTN-CT-2006-035606 and the Helmholtz Alliance "Physics at the Terascale".

References

1. M. Bähr et al., Eur. Phys. J. **C58**, 639 (2008), 0803.0883.
2. T. Sjöstrand, S. Mrenna, and P. Skands, Comput. Phys. Commun. **178**, 852 (2008), 0710.3820.
3. T. Gleisberg *et al.*, JHEP **02**, 007 (2009), 0811.4622.
4. M. H. Seymour, Comp. Phys. Commun. **90**, 95 (1995), hep-ph/9410414.
5. M. Bengtsson and T. Sjostrand, Phys. Lett. **B185**, 435 (1987).
6. M. Bengtsson and T. Sjostrand, Nucl. Phys. **B289**, 810 (1987).
7. E. Norrbin and T. Sjöstrand, Nucl. Phys. **B603**, 297 (2001), hep-ph/0010012.
8. G. Miu and T. Sjostrand, Phys. Lett. **B449**, 313 (1999), hep-ph/9812455.
9. P. Nason, JHEP **11**, 040 (2004), hep-ph/0409146.
10. S. Catani, F. Krauss, R. Kuhn, and B. R. Webber, JHEP **11**, 063 (2001), hep-ph/0109231.
11. L. Lönnblad, JHEP **05**, 046 (2002), hep-ph/0112284.
12. W. T. Giele, D. A. Kosower, and P. Z. Skands, Phys. Rev. **D78**, 014026 (2008), 0707.3652.
13. S. Hoeche, F. Krauss, S. Schumann, and F. Siegert, JHEP **05**, 053 (2009), 0903.1219.
14. K. Hamilton, P. Richardson, and J. Tully, (2009), 0905.3072.
15. S. Gieseke, JHEP **01**, 058 (2005), hep-ph/0412342.
16. T. Sjöstrand, S. Mrenna, and P. Skands, JHEP **05**, 026 (2006), hep-ph/0603175.
17. A. Buckley *et al.*, Phys. Rept. **504**, 145 (2011), 1101.2599.