DESY 13-145
MCnet-13-10

RAMBO on diet

Simon Plätzer

DESY, Notkestrasse 85, D-22607 Hamburg, Germany

August 14, 2013

Abstract. We describe a phase space generator which is flat for massless particles, and approximately flat for massive particles of masses much smaller than the typical mometum scales involved in the process. The same goal is achieved by the RAMBO algorithm, contrary to which our approach does only require the minimal number of degrees of freedom and is invertible in the sense that it provides a unique mapping of random numbers to the physical phase space point and vice versa. The very motivation to seek such an algorithm is explained in detail.

PACS. 02.70.Tt Monte Carlo methods

1 Introduction

Within applying the Monte Carlo (MC) method to cross section computations, the phase space generation is an integral part to the efficiency of the implementation. By phase space generation we here refer to a particular mapping of random numbers to a physical phase space point, excluding any methods of adaptive MC integration, but typically implementing importance sampling in a way that the resulting Jacobian resembles the structure of the differential cross section as close as possible.

The very opposite to any effort of importance sampling for physical cross sections, namely a flat population of phase space, has so far only be addressed by Kleiss and collaborators, [1]. Owing to the complicated structure of the Lorentz invariant phase space manifold, this task is actually far from trivial, though the resulting RAMBO algorithm exhibits a very transparent and simple structure. It is therefore most often used to provide means of cross checks for more involved phase space generator implementations and is certainly far from being used in realistic setups other than for this particular purpose. Besides the flat phase space population, which will exhibit the full peak structure of the differential cross section to the integration methods, another show stopper in this context is certainly its exhaustive use of random numbers, requiring $4n$ instead of the $3n-4$ required degrees of freedom to generate a system of n outgoing momenta.

One may therefore question why one would seek another variant of such a flat phase space generator. We will motivate, why precisely this property amongst others has been addressed in the present note. In particular, we will derive a phase space generation algorithm which will

- perform flat[1] phase space population, while it will
- require the minimum number of degrees of freedom needed for n outgoing particles, $3n-4$, and is
- invertible in the sense that it provides a unique and invertible mapping between a physical phase space point and the random numbers used to generate that particular point.

2 Motivation

The motivation to seek such an algorithm originates from the adaptive sampling of matrix element corrections for parton showers [2,3] using algorithms such as the one implemented in the ExSample library [4], outlined in more detail in [5]. The problem we tackle here is the following: Our aim is to sample a matrix element correction splitting kernel, which is of the form $|\mathcal{M}_R(p_\perp, z, \phi; \phi_n)|^2/|\mathcal{M}_B(\phi_n)|^2$ with a Born and real emission matrix elements squared $\mathcal{M}_{B,R}$ and is a function of the Born phase space point ϕ_n and the degrees of freedom needed for the additional emission, *e.g.* $p_\perp, z, \phi$. The latter are the variables which will actually be drawn from the associated Sudakov distribution by means

[1] For massive particles, almost flat, meaning flat in the limit that all invariants are much larger than the masses present.

of standard methods, [6], while the Born phase space point is fixed by the hard process considered and constitutes an external set of parameters. Any adaptive algorithm will have to keep track of those parameters and should do so in a way as to require the minimal number of degrees of freedom needed to describe the Born phase space point. Otherwise one will make the possible adaption steps more complicated and slower convergent than needed or possible. Of course, another choice would be to directly communicate the random numbers from the hard process generation, which would provide the most simple and straightforward solution to the problem at hand. Within recent approaches to improving parton showers by fixed order matrix element corrections [7, 8], this may not be possible anymore. Finally, we note that in the bulk of the cases (*i.e.* for soft and collinear emissions), no peak structure is present in the matrix element correction as a function of the Born degrees of freedom. This completes our motivation and we will specify the resulting algorithm in the next section. A generic C++ implementation of it is available publicly.

3 The algorithm

We consider the n-body phase space measure for a total momentum Q [2],

$$\mathrm{d}\phi_n\left(\{p_1,m_1\},...,\{p_n,m_n\}|Q\right) = \delta\left(\sum_{i=1}^n p_n - Q\right)\prod_{i=1}^n \mathrm{d}^4 p_i \delta(p_i^2-m_i^2)\theta(p_i^0-m_i)\ . \tag{1}$$

We factor the measure by the well-known approach of iterative 1 → 2 decays,

$$\begin{aligned}\mathrm{d}\phi_n\left(\{p_1,m_1\},...,\{p_n,m_n\}|Q\right) = \mathrm{d}\phi_n\left(\{p_1,m_1\},...,\{p_{n-2},m_{n-2}\},\{Q_{n-1},M_{n-1}\}|Q\right)\ \times \\ \mathrm{d}M_{n-1}^2 \times \mathrm{d}\phi_2\left(\{p_{n-1},m_{n-1}\},\{p_n,m_n\}|Q_{n-1}\right)\ ,\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathrm{d}\phi_n\left(\{p_1,m_1\},...,\{p_n,m_n\}|Q\right) = \\ &\left(\prod_{i=2}^n \mathrm{d}\phi_2\left(\{p_{i-1},m_{i-1}\},\{Q_i,M_i\}|Q_{i-1}\right)\right)\times\left(\prod_{i=2}^{n-1}\theta\left(M_{i-1}-m_{i-1}-M_i\right)\theta\left(M_i-\sum_{k=i}^n m_k\right)\mathrm{d}M_i^2\right)\end{aligned} \tag{3}$$

where we have made explicit the phase space limits present in the 1 → 2 decays, and we have identified $Q_1 = Q$ as well as $\{Q_n, M_n\} = \{p_n, m_n\}$. We express the two-body phase space measures in the respective rest frame of the parents Q_{i-1},

$$\mathrm{d}\phi_2\left(\{p_{i-1},m_{i-1}\},\{Q_i,M_i\}|Q_{i-1}\right) = \rho(M_{i-1},M_i,m_{i-1})\mathrm{d}\cos\theta_{i-1}\mathrm{d}\phi_{i-1} \tag{4}$$

where

$$\rho(M_{i-1},M_i,m_{i-1}) = \frac{1}{8M_{i-1}^2}\sqrt{(M_{i-1}^2-(M_i+m_{i-1})^2)(M_{i-1}^2-(M_i-m_{i-1})^2)} \tag{5}$$

$$\mathbf{p}_{i-1} = -\mathbf{Q}_i = 4M_{i-1}\rho(M_{i-1},M_i,m_{i-1})\left(\cos\phi_{i-1}\sin\theta_{i-1},\sin\phi_{i-1}\sin\theta_{i-1},\cos\theta_{i-1}\right)^T \tag{6}$$

and the zero components are determined by the respective mass shell conditions. We can thus perform flat phase space population, if we generate the intermediate masses from the measure

$$\begin{aligned}&\mathrm{d}M_n(M_2,...,M_{n-1}|M_1;m_1,...,m_n) = \\ &\left(\prod_{i=2}^{n-1}\rho(M_{i-1},M_i,m_{i-1})\theta\left(M_{i-1}-m_{i-1}-M_i\right)\theta\left(M_i-\sum_{k=i}^n m_k\right)\mathrm{d}M_i^2\right)\rho(M_{n-1},m_n,m_{n-1})\end{aligned} \tag{7}$$

in such a way as to arrive at constant weight.

3.1 The massless case

We shall now limit ourself to the discussion of external particles with zero masses; non-zero masses do not pose a problem, and the (invertible) procedure of how to obtain a set of massive momenta from a set of massless momenta

[2] Note that we have suppressed the overall factor $1/(2\pi)^{3n-4}$ for the sake of readability.

of identical total momentum is detailed in [1], or an alternative procedure outlined in the next section may be used. If all external masses vanish, $m_i = 0$, the measure for the intermediate masses is given by

$$\mathrm{d}M_n(M_2, ..., M_{n-1}|M_1; 0, ..., 0) = \frac{1}{8^{n-1}} \prod_{i=2}^{n-1} \frac{M_{i-1}^2 - M_i^2}{M_{i-1}^2} \theta(M_{i-1}^2 - M_i^2)\theta(M_i^2)\mathrm{d}M_i^2 \ . \tag{8}$$

Expressing $M_i = u_2 \cdots u_i \ M_1$ we have

$$\mathrm{d}M_n(M_2, ..., M_{n-1}|M_1; 0, ..., 0) = \frac{1}{8^{n-1}} M_1^{2n-4} \prod_{i=2}^{n-1} u_i^{n-1-i}(1-u_i)\theta(1-u_i)\theta(u_i)\mathrm{d}u_i \ . \tag{9}$$

Finally, substituting $v_i = (n+1-i) \ u_i^{n-i} - (n-i) \ u_i^{n+1-i}$ we arrive at the desired flat result,

$$\mathrm{d}M_n(M_2, ..., M_{n-1}|M_1; 0, ..., 0) = \frac{1}{8^{n-1}} M_1^{2n-4} \frac{1}{(n-1)! \ (n-2)!} \prod_{i=2}^{n-1} \theta(1-v_i)\theta(v_i)\mathrm{d}v_i \ . \tag{10}$$

Performing the mass and angular integrations, we find the total phase space volume to be given by

$$V_n = \int \mathrm{d}\phi_n \left(\{p_1, 0\}, ..., \{p_n, 0\}|Q\right) = \left(\frac{\pi}{2}\right)^{n-1} \frac{(Q^2)^{n-2}}{(n-1)! \ (n-2)!} \tag{11}$$

recovering precisely the RAMBO result. The complete algorithm for generating flat phase space with total momentum Q from $3n-4$ random numbers $r_1, ..., r_{3n-4}$ is specified in algorithm 1, and its inverse, *i.e.* solving for the random numbers r_i as a function of a given set of momenta, is given in algorithm 2.

Algorithm 1 The massless phase space generation algorithm.

```
$Q_1 \leftarrow Q$, $M_1 \leftarrow \sqrt{Q^2}$, $M_n \leftarrow 0$
for $i = 2, ..., n-1$ do
    solve $r_{i-1} = (n+1-i) \ u_i^{n-i} - (n-i) \ u_i^{n+1-i}$ for $u_i$
    $M_i \leftarrow u_2 \cdots u_i \sqrt{Q^2}$
    $\cos\theta_i \leftarrow 2 \ r_{n-5+2i} - 1$, $\phi_i = 2\pi \ r_{n-4+2i}$
    $q_i \leftarrow 4M_{i-1}\rho(M_{i-1}, M_i, 0)$
    $\mathbf{p}_{i-1} \leftarrow q_i(\cos\phi_i\sqrt{1-\cos^2\theta_i}, \sin\phi_i\sqrt{1-\cos^2\theta_i}, \cos\theta_i)$
    $p_{i-1} \leftarrow (q_i, \mathbf{p}_{i-1})$, $Q_i \leftarrow (\sqrt{q_i^2 + M_i^2}, -\mathbf{p}_{i-1})$
    boost $p_{i-1}$ and $Q_i$ by $\mathbf{Q}_{i-1}/Q_{i-1}^0$
end for
$p_n \leftarrow Q_n$
return $\{p_1, ..., p_n\}$ with weight $V_n$
```

Algorithm 2 The inverse of the massless phase space generation algorithm.

```
$M_1 \leftarrow \sqrt{(p_1 + ... + p_n)^2}$
$Q_n \leftarrow p_n$
for $i = n, ..., 2$ do
    $M_i \leftarrow \sqrt{(p_i + ... + p_n)^2}$, $u_i \leftarrow M_i/M_{i-1}$
    $r_{i-1} \leftarrow (n+1-i) \ u_i^{n-i} - (n-i) \ u_i^{n+1-i}$
    $Q_{i-1} \leftarrow Q_i + p_{i-1}$
    boost $p_{i-1}$ by $-\mathbf{Q}_{i-1}/Q_{i-1}^0$
    $r_{n-5+2i} \leftarrow (\mathbf{p}_{i-1}^z/|\mathbf{p}_{i-1}| + 1)/2$, $\phi \leftarrow \tan^{-1}(\mathbf{p}_{i-1}^y/\mathbf{p}_{i-1}^x)$, $r_{n-4+2i} = (2\pi\theta(-\phi) + \phi)/(2\pi)$
end for
```

3.2 The massive case

The massive case can in principle be dealt with by the reshuffling procedure given in the original RAMBO paper [1], including its inverse. We here outline an alternative method, which does not involve solving the reshuffling condition,

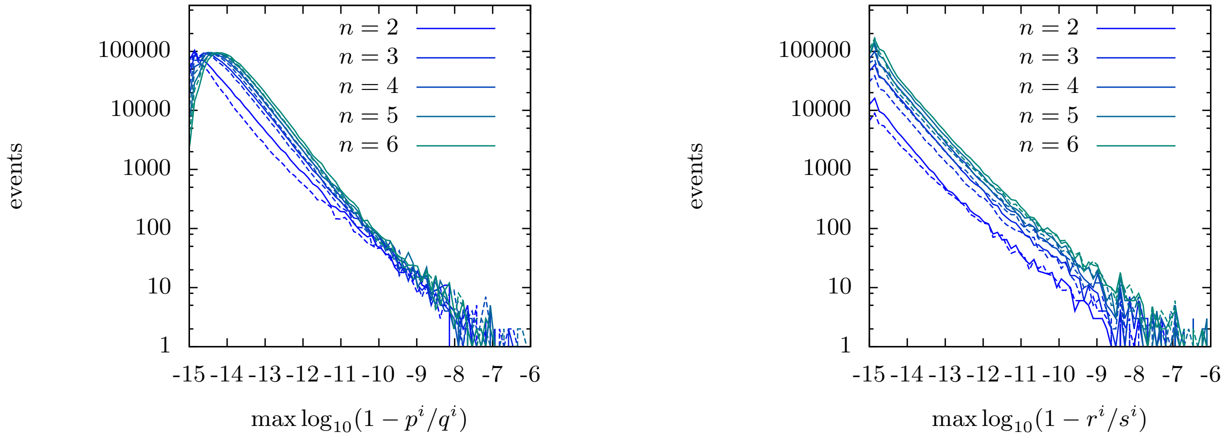

Fig. 1. Left: Accuracy of a sequence of phase space hashing and generation from the hash, $\phi(q) \to \boldsymbol{r} \to \phi(p)$ for two and up to six external, massless legs. The initial phase space momenta q have been generated with RAMBO. Right: Accuracy of a sequence of random number generation, phase space generation and phase space hashing, $\boldsymbol{r} \to \phi \to \boldsymbol{s}$ for the same settings. The solid lines represent a setting for a hadronic collider with varying center of mass energy, $\sqrt{\hat{s}}$, which has been sampled flat. The dashed lines correspond to a fixed energy collision.

$\sum_i \sqrt{\xi \mathbf{p}_i^2 + m_i^2} = \sum_i |\mathbf{p}_i|$, numerically. We start by rewriting the measure for the intermediate masses as

$$\mathrm{d}M_n(M_2, ..., M_{n-1}|M_1; m_1, ..., m_n) = \frac{1}{8} \prod_{i=2}^{n} \frac{\rho(M_{i-1}, M_i, m_{i-1})}{\rho(K_{i-1}, K_i, 0)} \times \prod_{i=2}^{n-1} \frac{M_i}{K_i} \times \mathrm{d}M_n(K_2, ..., K_{n-1}|K_1; 0, ..., 0) \qquad (12)$$

where $K_i = M_i - \sum_{k=i}^{n} m_k$ for $i = 1, ..., n-1$ and $K_n = 0$. Up to an additional weight factor the massive case is thus simply related to the massless one upon taking care of the conversion between K_i and M_i. The additional weight factor will be close to unity, if all invariants are forced into regions where they are much larger than the masses of the external legs – the same observation holds for the massive variant of RAMBO.

3.3 Numerical stability

Having at hand the possibility to invert the phase space generation back to the original random numbers, it is of course desirable to test the numerical stability of the implementation. To this extent, we test two possible sequences of both directions of the algorithm which need to preserve their respective input values: a phase space point ϕ is used to calculate the random numbers associated to it which in turn serve as input for the phase space generation step; and a given set of random numbers is used to generate a phase space point which in turn is inverted back to the respective random numbers. We have tested those sequences for two and up to six massless as well as massive outgoing particles. The typical accuracy, shown in figure 1, with per mille level number of events with less than ten significant digits, is certainly not matching expectations from double precision, though we argue that it is absolutely sufficient for the purposes we have in mind. The most probable source of the broad tails to less accurate points are the implementation of Lorentz boosts we have used and improving their numerical stability will be subject to future development.

4 Summary

We have presented a phase space generation algorithm which preforms (almost) flat phase space generation while requiring the minimal number of degrees present in a physical phase space point and being invertible in the sense that it represents a unique and invertible mapping between random numbers and momentum components. The main application of this algorithm is in providing a measure of hashing a phase space point to provide minimal information for adaptive algorithms dealing with functions which are parametric in an externally fixed set of momenta. We anticipate further application within the context of improving the convergence of next-to-leading order calculations carried out within the subtraction formalism.

Acknowledgments

This work was supported by the Helmholtz Alliance “Physics at the Terascale”.

A Availability

An implementation of the algorithm outlined in this note is available as a phase space generator class within the Matchbox module of Herwig++. A stand-alone C++ implementation is vailable upon request from the author.

References

1. R. Kleiss, W. J. Stirling, and S. D. Ellis, Comput. Phys. Commun. **40**, 359 (1986).
2. M. H. Seymour, Comp. Phys. Commun. **90**, 95 (1995), hep-ph/9410414.
3. E. Norrbin and T. Sjöstrand, Nucl. Phys. **B603**, 297 (2001), hep-ph/0010012.
4. S. Platzer, Eur.Phys.J. **C72**, 1929 (2012), 1108.6182.
5. S. Platzer and S. Gieseke, Eur.Phys.J. **C72**, 2187 (2012), 1109.6256.
6. S. Platzer and M. Sjodahl, Eur.Phys.J.Plus **127**, 26 (2012), 1108.6180.
7. S. Platzer, (2012), 1211.5467.
8. S. Platzer, in preparation (2013).