CERN-TH.4299/85

A NEW MONTE CARLO TREATMENT OF
MULTIPARTICLE PHASE SPACE AT HIGH ENERGIES

R. Kleiss and W.J. Stirling

CERN - Geneva

and

S.D. Ellis^{*)}

Dept. of Physics, University of Washington
Seattle, WA 98195, U.S.A.

ABSTRACT

We present a derivation of the expression for the phase-space volume for massless particles, in which we have not assumed any hierarchical or cascade-like structure. Our results lead directly to a Monte Carlo event generator algorithm for phase-space distributions which is optimally efficient for arbitrary numbers of massless particles. We show how to extend the algorithm to the case of arbitrary particle masses, and discuss the limits to the practicality of our approach.

^{*)} Work supported in part by the US Department of Energy,
Contract No. DE-AC06-81ER-40048.

1. INTRODUCTION

In this paper we discuss the problem of multiparticle phase space, in particular the derivation of a Monte Carlo (MC) procedure to generate phase-space points with a uniform probability distribution. The motivation for this study is the following. In the phenomenology of high-energy collisions, MC event generators have proved to be extremely useful, combining flexibility, unbiasedness and (given enough computing power) good accuracy. In the ideal case the MC events are *unweighted* (i.e. all events as generated by the MC are equally probable); in practice each event is usually assigned a *weight* made up of the scattering matrix element and the phase-space density. This is acceptable if the weights do not fluctuate too much: in particular, the *efficiency* of the weight distribution, defined as the average event weight divided by the maximum occurring event weight, should not be too small. This is an appropriate definition of efficiency since, in order to generate an *unweighted* sample of events, one simply keeps or discards a given generated event according to whether a random number is smaller or larger than the weight of that event divided by the maximum weight. Thus the efficiency as defined above is the fraction of all generated events which would be kept in such an unweighted sample.

Here we restrict ourselves to those situations where the scattering matrix elements do not fluctuate very much; as an example one may consider multijet production in high-energy $p\bar{p}$ or e^+e^- scattering, where all jets are separated from each other and the beams by cuts that avoid all singularities of the matrix element. Clearly, in such situations it is attractive to have a MC procedure that generates a phase-space distribution which is as uniform as possible.

The above problem has been under study for a considerable time: excellent reviews can be found in Refs. [1-3]. The most widely used approach to generating phase-space distributions is that attributed to Raubold and Lynch (see Refs. [2] and [4]), in which the multiparticle production is represented as a series of sequential two-body decays. This 'hierarchical' procedure results in non-uniform phase-space distributions and hence variable event weights. The maximum occurring weight is not known *a priori* but rather is determined empirically from the weight

distribution of a generated event sample. Especially in the case of high energy, where the masses of all produced particles are small or even negligible compared to the total energy, the efficiency (as defined above) turns out to be rather poor when n , the number of particles produced, becomes large. This is illustrated in Table 1: the efficiency becomes less than 10% for $n > 8$.

In this paper we aim at making an improvement on this situation by formulating a MC procedure which has unit efficiency in the case of massless particles (i.e. the high-energy limit). This is achieved by replacing the *hierarchical* structure of the MC procedure described above by a *democratic* one, in which all particles are treated equally. In this procedure all particle momenta are generated isotropically without a constraint on the momentum; subsequently the momenta are boosted and rescaled in such a way that the overall momentum P^μ of the event takes on the desired value $(w, 0, 0, 0)$, where w is the c.m. energy.

The outline of the paper is as follows. In Section 2 we derive an expression for the phase-space volume for massless particles. This derivation is democratic in the sense indicated above. From it we immediately derive the MC algorithm, and see that the events generated this way have a constant weight. This is discussed in Section 3. In Section 4 we describe how to take non-zero particle masses into account. This leads to the introduction of variable event weights, for which we derive the expression.

In Section 5 we discuss the weight distribution and make some numerical comparisons between the hierarchical and the democratic procedures. Finally, we have collected a number of technical points in Appendices A-D. Appendix E contains a FORTRAN listing of the corresponding program for generating events.

2. THE PHASE-SPACE VOLUME FOR MASSLESS PARTICLES

In this section we shall derive the expression for $V_n(w)$, the phase-space volume for a system of n massless particles with a total c.m. energy w :

$$V_n(w) = \int \delta^4(P - \sum_{i=1}^n p_i) \prod_{i=1}^n (d^4 p_i \delta(p_i^2) \theta(p_i^0)) , \quad (2.1)$$

where P^μ is defined in the previous section^{*}). The formula for $V_n(w)$ is actually known, and, for instance, given in Ref. [1]; in Appendix A we present a very short derivation along the more traditional lines. What interests us here, however, is not so much the result for Eq. (2.1) but rather the way we derive it, which will lead us to the MC algorithm mentioned in the Introduction.

It is useful to start by defining the quantity:

$$R_n \equiv \int \prod_{i=1}^n (d^4 q_i \delta(q_i^2) f(q_i^0) \theta(q_i^0)) , \quad (2.2)$$

from which we immediately have

$$R_n = \left[2\pi \int_0^\infty x f(x) dx \right]^n . \quad (2.3)$$

where, for the moment, $f(x)$ is an arbitrary function of a single variable. The quantity R_n is a phase-space-like object: it can be interpreted as describing a system of n massless four-momenta q_i^μ that are not constrained by momentum conservation but occur with some weight function f which keeps the total volume finite. We proceed to relate the four-vectors q_i^μ to the physical four-momenta p_i^μ by performing the conformal transformation (Lorentz boost and scaling transformation) which transforms Q^μ , the overall momentum of the set q_i^μ , into the desired P^μ :

$$\begin{aligned} p_i^0 &= x (\gamma q_i^0 + \vec{b} \cdot \vec{q}_i) , \\ \vec{p}_i &= x (\vec{q}_i + \vec{b} q_i^0 + a(\vec{b} \cdot \vec{q}_i) \vec{b}) , \end{aligned} \quad (2.4)$$

with

$$\vec{b} = -\vec{Q}/M , \quad x = w/M ,$$

^{*}) Unless indicated otherwise, every integral is understood to run from $-\infty$ to $+\infty$.

$$\begin{aligned}
\gamma &= Q^0/M = \sqrt{1+\vec{b}^2}, \\
a &= 1/(1+\gamma), \\
Q^\mu &= \sum_{i=1}^n q_i^\mu, \quad M = \sqrt{Q^2}.
\end{aligned} \tag{2.5}$$

We shall denote this transformation, and its inverse, as follows:

$$\begin{aligned}
p_i^\mu &= x H_{\vec{b}}^\mu(q_i), \\
q_i^\mu &= \frac{1}{x} H_{-\vec{b}}^\mu(p_i).
\end{aligned} \tag{2.6}$$

We now implement the definition of x , $\vec{b}$, and p_i^μ into the expression for R_n , as follows:

$$\begin{aligned}
R_n &= \int \prod_{i=1}^n [d^4 q_i \delta(q_i^2) \theta(q_i^0) f(q_i^0) d^4 p_i \delta^4(p_i - x H_{\vec{b}}(q_i))] \\
&\quad \cdot d^3 \vec{b} \delta^3(\vec{b} + \sum_{i=1}^n \vec{q}_i / \sqrt{(\sum_{i=1}^n q_i)^2}) \\
&\quad \cdot d x \delta(x - w / \sqrt{(\sum_{i=1}^n q_i)^2}),
\end{aligned} \tag{2.7}$$

where the integral over x runs from zero to infinity. The next step is to manipulate the arguments of the various δ functions. Denoting the sum of the four-vectors p_i^μ by P^μ , we can make the following substitutions:

$$\delta^4(p_i - x H_{\vec{b}}(q_i)) = x^{-4} \delta^4(q_i - \frac{1}{x} H_{-\vec{b}}(p_i)), \tag{2.8a}$$

$$\theta(q_i^0) \delta(q_i^2) = x^2 \theta(p_i^0) \delta(p_i^2), \tag{2.8b}$$

$$\delta(x - w / \sqrt{(\sum_{i=1}^n q_i)^2}) = \frac{w}{x} \delta(w - \sqrt{P^2}), \tag{2.8c}$$

$$\delta^3(\vec{b} + \sum_{i=1}^n \vec{q}_i / \sqrt{(\sum_{i=1}^n q_i^2)}) = w^3 \delta^3(w\vec{b} + \vec{H}_{-\vec{b}}(P)) . \quad (2.8d)$$

As demonstrated in Appendix B, Eq. (2.8d) can be further simplified by writing

$$\delta^3(w\vec{b} + \vec{H}_{-\vec{b}}(P)) = \frac{1}{\gamma} \delta^3(\vec{P}) . \quad (2.8e)$$

We now perform the integration over $d^4 q_i$ with the result:

$$\begin{aligned} R_n = & \int \prod_{i=1}^n [d^4 p_i \delta(p_i^2) \theta(p_i^0)] \delta^3(\sum_{i=1}^n \vec{p}_i) \delta(w - \sum_{i=1}^n p_i^0) \\ & \cdot \left\{ \prod_{i=1}^n f\left(\frac{1}{x} H_{-\vec{b}}^0(p_i)\right) \right\} \frac{w^4}{x^{2n+1} \gamma} d^3 \vec{b} dx . \end{aligned} \quad (2.9)$$

We can now make a useful choice for the weight function f . Taking $f(x) = e^{-x}$, we find from Eq. (2.3) that

$$R_n = (2\pi)^n . \quad (2.10)$$

On the other hand, since $P^\mu = (w, 0, 0, 0)$, we also have

$$\prod_{i=1}^n f\left(\frac{1}{x} H_{-\vec{b}}^0(p_i)\right) = e^{-\frac{\gamma w}{x}} . \quad (2.11)$$

The only step remaining is to perform the integration over $\vec{b}$ and x . These integrations reduce the number of degrees of freedom implied in the definition (2.2) of R_n ($3n$) to the correct number of dimensions of the phase space ($3n - 4$). We can write

$$R_n = V_n(w) \cdot S_n(w) , \quad (2.12)$$

with

$$\begin{aligned}
S_n(w) &= \int d^3\vec{b} \int_0^\infty dx \frac{w^4}{\gamma x^{2n+1}} e^{-\frac{\gamma w}{x}} \\
&= 2\pi w^{4-2n} \Gamma\left(\frac{3}{2}\right) \Gamma(n-1) \Gamma(2n) / \Gamma\left(n+\frac{1}{2}\right). \quad (2.13)
\end{aligned}$$

The last identity is derived most easily by changing the integration variables from $x, |\vec{b}|$ to $w\gamma/x, 1/\gamma^2$. Combining eqs. (2.10) and (2.12), and using some properties of the Γ function [5], we finally arrive at the formula for the desired phase-space volume:

$$V_n(w) = \left(\frac{\pi}{2}\right)^{n-1} \frac{w^{2n-4}}{\Gamma(n) \Gamma(n-1)}. \quad (2.14)$$

This finishes our derivation.

3. THE MONTE CARLO ALGORITHM

In the Monte Carlo philosophy an integration over a variable can be replaced by a number of random choices of the value of that variable. This implies that we can directly translate the discussion of the previous section into a description of a MC algorithm which generates massless four-momenta p_i^μ according to the phase-space distribution (2.1). This algorithm consists of two steps:

i) Generate independently n massless four-momenta q_i^μ with isotropic angular distribution and energies q_i^0 distributed according to the density $q_i^0 e^{-q_i^0} dq_i^0$. Using the symbol ϱ_i to denote a random number uniformly distributed in $(0, 1)$ we do this as follows:

$$\begin{aligned}
C_i &= 2\varrho_{i_1} - 1, & \varphi_i &= 2\pi\varrho_{i_2}, \\
q_i^0 &= -\ln(\varrho_{i_3}\varrho_{i_4}), \quad (3.1)
\end{aligned}$$

(cont.)

$$\begin{aligned} q_i^x &= q_i^0 \sqrt{1-C_i^2} \cos \varphi_i, & q_i^y &= q_i^0 \sqrt{1-C_i^2} \sin \varphi_i, \\ q_i^z &= q_i^0 C_i. \end{aligned} \quad (3.1)$$

A proof that the algorithm for q_1^0 indeed generates the desired distribution is given in Appendix C.

ii) Transform the four-vectors q_1^μ into the four-momenta p_1^μ , using the transformations defined in Eqs. (2.4); (2.5).

This completes the description of the algorithm, which is extremely simple to implement on a computer.

The above MC procedure has, of course, to be supplemented with a prescription for the weight of a generated event. In our case this is particularly simple: from the fact that $S_n(w)$, defined in Eqs. (2.12), (2.13), only depends on n and w and not on any particular momentum (or number of momenta) p_1^μ , we see that the event weights are constant. To be precise, every MC event generated in this way has a weight

$$W_0 = V_n(w) . \quad (3.2)$$

From the point of view of this paper this is the optimal result: the efficiency E of the weight distribution for our democratic procedure (DP) is unity:

$$E_{DP} = 1 . \quad (3.3)$$

This is to be contrasted with the efficiency from the hierarchical procedure (HP) as discussed in Refs. [1, 2, 4], E_{HP} , which is given for massless particles in Table 1. These results will be discussed in more detail below.

If we are only interested in the zero mass case, we are now finished. However, in practice non-zero particle masses occur and we have to devise a

procedure for generating the corresponding momenta. This is the subject of the next section.

Before we finish this section one remark is in order. There are many different transformations which, acting on the vectors q_i^μ , would give vectors p_i^μ with the correct overall momentum (for instance, we could add a constant vector to all the $\vec{q}_i$'s instead of boosting them); similarly, different weight functions $f(x)$ are possible. In general, however, these would not lead to the above simple form for $S_n(w)$ and hence for the weight W . If, for instance, we had chosen the same transformations (2.4) but had used a weight function $f(x) = \exp(-x^2)$, the form of $S_n(w)$ would have become

$$S'_n(w; p_1, \dots, p_n) = \int dx d^3b \frac{w^4}{\gamma x^{2n+1}} \exp\left(\frac{-1}{x^2}(A + \vec{b} \cdot \vec{B} + \vec{b} \vec{b} : \vec{\vec{C}})\right) \quad (3.4)$$

with

$$\begin{aligned} A &= \gamma^2 \sum_{i=1}^n p_i^0{}^2, \\ \vec{B} &= -2\gamma \sum_{i=1}^n p_i^0 \vec{p}_i, \\ \vec{\vec{C}} &= \sum_{i=1}^n \vec{p}_i \vec{p}_i, \end{aligned} \quad (3.5)$$

which depends on the particulars of the generated event. This means that the events would have a variable weight

$$W' = \pi^n / S'_n(w; p_1, \dots, p_n), \quad (3.6)$$

which is, of course, not as attractive as a constant weight. The transformations and weight function chosen in this section seem to be the only reasonably simple ones that yield a constant weight.

4. MASSIVE PARTICLES

Having discussed the case where all produced particles are massless, we shall now turn to the problem of finite mass. In the usual hierarchical procedures there is no fundamental difference between generating massless or massive particles, but the democratic approach described above cannot be applied in a straightforward manner: introducing non-zero masses for the four-vectors q_i^μ in Eq. (2.2) makes it impossible to scale them and at the same time keep the masses fixed. Instead, we shall discuss how, starting with a set of zero-mass momenta satisfying the conditions of phase space, we can transform the momentum components so as to introduce a set of given non-zero masses while still satisfying the phase-space conditions. For the sake of generality we shall start with a set of momenta p_i^μ having a mass μ_i . The desired phase-space integral is then

$$V(\{p\}) = \int \delta^3\left(\sum_{i=1}^n \vec{p}_i\right) \delta\left(w - \sum_{i=1}^n p_i^0\right) \prod_{i=1}^n (d^4 p_i \delta(p_i^2 - \mu_i^2) \theta(p_i^0)), \quad (4.1)$$

where the symbol $\{p\}$ denotes the set of possible p_i^μ . We now transform the p_i^μ into four-momenta k_i^μ with mass m_i as follows:

$$\begin{aligned} \vec{k}_i &= \xi \vec{p}_i, \\ k_i^0 &= \sqrt{m_i^2 + \xi^2(p_i^{02} - \mu_i^2)}, \end{aligned} \quad (4.2)$$

where ξ is the solution of the equation

$$w = \sum_{i=1}^n \sqrt{m_i^2 + \xi^2(p_i^{02} - \mu_i^2)}. \quad (4.3)$$

The inverse transformation is, of course,

$$\begin{aligned} \vec{p}_i &= \vec{k}_i / \xi, \\ p_i^0 &= \sqrt{\mu_i^2 + (k_i^{02} - m_i^2) / \xi^2}, \end{aligned} \quad (4.4)$$

with ξ satisfying

$$w = \sum_{i=1}^n \sqrt{\mu_i^2 + (k_i^0{}^2 - m_i^2)/\xi^2} . \quad (4.5)$$

It should be noted that Eqs. (4.3) and (4.5) give the same value for ξ .

In a way similar to the treatment in Section 2 we implement these transformations in Eq. (4.1), as follows:

$$\begin{aligned} V(\{p\}) = & \int \prod_{i=1}^n \left\{ d^4 p_i \delta(p_i^2 - \mu_i^2) \theta(p_i^0) d^4 k_i \delta^3(k_i - \xi \vec{p}_i) \right. \\ & \left. \cdot \delta(k_i^0 - \sqrt{m_i^2 + \xi^2(p_i^0{}^2 - \mu_i^2)}) \right\} \\ & \cdot \delta^3\left(\sum_{i=1}^n \vec{p}_i\right) \delta\left(w - \sum_{i=1}^n p_i^0\right) \\ & \cdot d\xi \delta\left(w - \sum_{i=1}^n \sqrt{m_i^2 + \xi^2(p_i^0{}^2 - \mu_i^2)}\right) \sum_{i=1}^n \frac{\xi(p_i^0{}^2 - \mu_i^2)}{\sqrt{m_i^2 + \xi^2(p_i^0{}^2 - \mu_i^2)}} , \quad (4.6) \end{aligned}$$

where the last factor compensates for the Jacobian which is implied in the δ -function defining ξ . Proceeding as before we eliminate the p_i^μ from the above expression. Since in going from the p_i^μ to the k_i^μ the number of degrees of freedom remains the same, no integration over ξ has to be performed. Instead $d\xi$ cancels against the δ function for the sum of the p_i^0 , resulting in a Jacobian factor:

$$\begin{aligned} d\xi \delta\left(w - \sum_{i=1}^n p_i^0\right) &= d\xi \delta\left(w - \sum_{i=1}^n \sqrt{\mu_i^2 + (k_i^0{}^2 - m_i^2)/\xi^2}\right) \\ &= \xi \left[\sum_{i=1}^n (\vec{p}_i^2 / p_i^0) \right]^{-1} . \quad (4.7) \end{aligned}$$

The final result of these manipulations is

$$\begin{aligned}
V(\{p\}) = & \int \prod_{i=1}^n (d^4 k_i \delta(k_i^2 - m_i^2) \theta(k_i^0)) \\
& \cdot \delta^3(\sum_{i=1}^n \vec{k}_i) \delta(w - \sum_{i=1}^n k_i^0) \\
& \cdot \left\{ \xi^{3(n-1)} \left[\prod_{i=1}^n \frac{k_i^0}{p_i^0} \right] \left[\sum_{i=1}^n \frac{|\vec{k}_i|^2}{k_i^0} \right] \left[\sum_{i=1}^n \frac{|\vec{p}_i|^2}{p_i^0} \right]^{-1} \right\} \quad (4.8)
\end{aligned}$$

The right-hand side of this equation consists of the phase-space integral over the set $\{k\}$ of the expression in brackets. Again reformulating this result in terms of a MC algorithm we arrive at the following result. Starting with a MC event, i.e. a set of p_i^μ , we can find ξ as the solution of Eq. (4.3) and make the transformation (4.2) from the p_i^μ to the k_i^μ . We then have a set of four-momenta with the required masses. The MC weight corresponding to this procedure is given by

$$W(\{p\}; \{k\}) = \xi^{3(n-1)} \left[\prod_{i=1}^n \frac{p_i^0}{k_i^0} \right] \left[\sum_{i=1}^n \frac{|\vec{p}_i|^2}{p_i^0} \right] \left[\sum_{i=1}^n \frac{|\vec{k}_i|^2}{k_i^0} \right]^{-1}. \quad (4.9)$$

It should be noted that for general masses μ_i , m_i , and energies p_i^0 , no analytic expression for ξ exists. However, for given m_i , μ_i , and p_i^0 , ξ can be computed numerically to high accuracy very quickly. Clearly, upon interchanging the sets $\{p\}$ and $\{k\}$, the weight is inverted.

In the special case under consideration here, all the μ_i are zero. We can then express ξ in terms of the four-momenta k_i^μ :

$$\xi = \sum_{i=1}^n |\vec{k}_i| / w, \quad (4.10)$$

and the weight becomes

$$W_m = \left[\frac{1}{w} \sum_{i=1}^n |\vec{k}_i| \right]^{2n-3} \left[\prod_{i=1}^n \frac{|\vec{k}_i|}{k_i^0} \right] \left[\sum_{i=1}^n \frac{|\vec{k}_i|^2}{k_i^0} \right]^{-1}. \quad (4.11)$$

This concludes our discussion. We now have at our disposal:

- i) an algorithm for generating MC events with massless momenta, with weight W_0 as given in Eq. (3.2);
- ii) an algorithm for transforming the massless momenta into massive ones, with weight W_m as given in Eq. (4.11).

By subsequently applying these two MC algorithms we are able to generate any type of final state. The weight assigned to the resulting phase-space events is given by

$$W = W_0 \cdot W_m \quad (4.12)$$

In contrast to W_0 , the weight factor W_m is not a constant but varies over phase space. Consequently, the efficiency E_{DP} of our democratic procedure is less than 100%. Since the efficiency determines how well we can use the algorithm to generate unweighted events (by applying a rejection criterion on the events as discussed above), it is a quantity of actual computational interest. In the next section we discuss it in more detail.

5. NUMERICAL RESULTS

As a conclusion to our discussion we shall present some comparisons between our democratic MC approach and the standard one--the HP as described earlier. As an example of this latter procedure we have taken a version of the phase-space event generator GENBOD by James [4].

The DP method is represented by our MC program RAMBO, which is listed in Appendix E together with the random number generator that it uses.

Our comparison concentrates on the efficiency of the weight distributions as defined in Section 1. This quantity is one of the most important parameters for describing the quality of a weight distribution: from the point of view of generating phase-space distributions, the efficiency should be as close as possible to 100%. We shall now discuss how this is satisfied in practical cases.

In determining the efficiency corresponding to the weight distribution generated by a given MC algorithm one of the problems is to determine the maximum weight which can actually occur. We have studied this issue for the case of the DP and give some results in Appendix D, which are particularly useful for specific configurations of masses. Unfortunately, no simple, exact result for the maximum weight is known for the general case, either for the DP or HP. For the following discussion we shall adopt the usual procedure which involves using not the true maximum weight but rather the actual maximum value which is realized in the event sample under study. It is important to note that in many situations, especially for large n values, the weight distribution has a long (low area) tail leading up to the true maximum. In this case the maximum weight observed in a given MC event sample is often appreciably smaller than the true maximum possible weight. This can be viewed as an advantage of the 'practical' efficiency employed here (and in actual calculations) as it tends to be more optimistic than the 'theoretical' efficiency. On the other hand, this practical definition can lead to efficiencies which fluctuate considerably between two different MC event samples, depending on the random numbers that occurred. The results presented here are therefore to be taken as indications rather than as absolutely valid results. All our results here are based on samples of 10,000 MC events each.

Since we know that for massless particles RAMBO is perfectly efficient ($E_{DP} = 100\%$), it is instructive to examine the efficiency E_{HP} for the standard method in this case. The results are summarized in Table 1. It is seen that E_{HP} decreases with increasing n (this is a general feature), becoming less than 10% for $n > 8$.

Upon introducing non-zero particle masses a problem arises: the number of parameters of our comparison becomes very large and a systematic comparison of E_{DP} and E_{HP} for all possible mass configurations becomes impractical. One might require all masses m_i to be equal, thus reducing the set of parameters to simply two, n and the total mass fraction f , defined as

$$f = \sum_{i=1}^n m_i / w , \quad (5.1)$$

but this choice typically does not correspond to any physically interesting situation. Instead, we prefer to make a MC sampling of the set of all mass configurations: before generating an event we first choose n random numbers ρ_i ($i = 1, \dots, n$) and fix the masses by

$$m_i = w \cdot f \cdot \rho_i / \left(\sum_{j=1}^n \rho_j \right) , \quad (5.2)$$

so that every MC event has a different mass configuration. In this way, we hope to describe a property of the whole set of mass configurations while keeping only f and n as parameters. In Table 2 and 3 we present the efficiencies E_{HP} and E_{DP} , respectively, for different values of f and n . As mentioned above, E decreases if n increases; E_{HP} is seen to increase with increasing f , while E_{DP} decreases with increasing f . In general, E_{DP} is larger than E_{HP} as long as $f < 0.5$, for all n . This is due to the fact that, for $f = 0$, E_{DP} is 100% for all n , in contrast to E_{HP} . For small values of E the effect of the above-mentioned fluctuations becomes visible in the fact that the values along rows and columns are not behaving in a monotonic way; this gives one an idea of the uncertainty in our results. We have checked that the tables remain essentially unchanged if we replace the 'random mass' prescription of Eq. (5.2) with equal mass for all particles; the efficiencies are then slightly higher in all cases.

One final practical feature, which is not clear from the tables, is the CPU time involved. Clearly, for RAMBO the computing time necessary to generate an event is proportional to n . Interestingly enough, for the HP the computing time has a faster increase with n . We have traced this back to the fact that for HP one has to pick values for the masses of the subsequent stages in the cascade. These masses have to be picked at random, and then put in decreasing order. It turns out that for large n this seemingly innocent procedure takes up a sizeable

amount of the computing time, owing to the many comparisons of floating-point numbers involved.

Thus we are led to conclude, both on the grounds of formal efficiency and CPU efficiency, that the DP can be expected to work better than the HP for the types of situations now experimentally relevant, i.e. where sizeable numbers of light particles are produced with the sum of their masses less than half the total energy.

REFERENCES

- [1] E. Byckling and K. Kajantie, Particle kinematics (John Wiley, NY, 1973).
- [2] F. James, CERN 68-15 (1968).
- [3] F. James, Rep. Progr. Phys. 43 (1980) 1145.
- [4] F. James, program GENBOD, CERN Program Library.
- [5] M. Abramowitz and I. Stegun, Handbook of mathematical functions (US National Bureau of Standards, Washington, 1984).

Table 1

Efficiency of the hierarchical procedure

for final states of n massless particles.

Each entry is based on a sample of 10,000 events.

n	E_{HP} (%)	n	E_{HP} (%)
3	64.3	11	4.3
4	43.7	12	2.8
5	28.9	13	2.3
6	20.0	14	1.9
7	14.3	15	1.9
8	10.2	16	0.8
9	7.1	17	1.0
10	5.5	18	0.9

Table 2

E_{HP} in % for random-mass samples of 10,000 events
for different values of the parameters f and n defined in the text

		f									
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
n	2	100	100	100	100	100	100	100	100	100	100
	3	64	69	71	73	75	76	76	77	78	78
	4	44	49	51	55	58	58	59	61	62	62
	5	29	35	38	41	44	45	47	48	49	49
	6	20	25	28	32	33	35	37	38	38	39
	7	14	19	22	25	26	28	29	29	27	30
	8	10	13	17	19	22	23	23	20	24	25
	9	7.2	10	14	16	15	17	19	17	20	21
	10	5.5	7.6	9.3	12	13	15	14	16	17	17
	11	4.3	5.9	9.2	9.4	11	11	10	14	14	15
	12	2.8	5.0	8.6	8.0	10	9.1	11	12	13	11
	13	2.3	4.2	6.2	5.3	8.5	6.5	9.3	8.9	10	10
	14	1.9	3.7	5.1	5.2	5.2	5.7	8.5	9.4	7.9	8.6
	15	1.9	3.1	3.8	5.4	4.4	5.4	7.4	5.8	5.8	8.3
	16	0.8	3.1	3.0	4.4	4.7	4.9	5.8	7.0	6.1	5.4
	17	1.1	2.5	2.7	3.6	3.6	4.5	4.5	5.0	5.0	6.2
	18	0.9	1.4	2.4	3.2	2.7	4.5	4.5	4.0	4.4	5.5

Table 3

E_{DP} in % for random-mass samples of 10,000 events
for different values of the parameters f and n defined in the text

		f									
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
n	2	100	100	100	100	100	100	100	100	100	100
	3	100	96	93	82	80	76	73	71	45	54
	4	100	93	85	68	66	53	52	26	36	18
	5	100	89	71	65	45	45	38	12	17	15
	6	100	85	63	56	39	22	14	9.9	15	9.5
	7	100	82	63	42	29	23	13	12	5.3	3.0
	8	100	76	55	37	21	17	13	6.8	3.8	4.6
	9	100	74	52	33	15	11	4.9	2.0	4.1	1.2
	10	100	72	50	24	14	9.5	5.4	2.1	2.9	1.9
	11	100	69	44	25	10	5.9	3.5	1.2	1.7	0.1
	12	100	67	40	18	10	6.6	2.6	1.2	1.9	0.6
	13	100	64	36	16	11	3.8	2.6	3.4	6.5	0.1
	14	100	64	32	16	9.8	3.5	2.9	1.6	3.3	0.4
	15	100	62	31	14	5.9	3.1	1.9	0.8	0.4	0.3
	16	100	59	29	15	5.0	4.2	1.4	0.8	0.4	0.3
	17	100	57	26	13	4.6	2.5	0.8	0.2	0.4	0.2
	18	100	55	22	8.9	4.1	1.3	0.9	0.5	0.4	0.1

APPENDIX A

In this appendix we present an alternative derivation of the result (2.14). This proceeds by induction in the number of particles using a hierarchical picture. First we notice that since w is the only dimensionful parameter in the problem (all masses being zero), we necessarily have from dimensional arguments:

$$V_n(w) = a_n w^{2n-4}, \quad (\text{A.1})$$

so that we only have to find the coefficient a_n . We can relate a_n to a_{n-1} by integrating over $n-1$ momenta in the production of n particles in the following way:

$$\begin{aligned} V_n(w) &= \int \delta^4(P - \sum_{i=1}^n p_i) \prod_{i=1}^n (d^4 p_i \delta(p_i^2) \theta(p_i^0)) \\ &\quad \cdot d^4 q \delta^4(q - \sum_{i=1}^{n-1} p_i) \delta(q^2 - w'^2) d w'^2 \\ &= \int \delta^4(P - p_n - q) d^4 p_n \delta(p_n^2) \theta(p_n^0) \\ &\quad d^4 q \theta(q^0) \delta(q^2 - w'^2) a_{n-1} w'^{2n-6} d w'^2. \end{aligned} \quad (\text{A.2})$$

Inserting the well-known result for the two-particle phase space we are led to

$$\begin{aligned} a_n w^{2n-4} &= \frac{\pi}{2} a_{n-1} \int_0^{w^2} (1 - w'^2/w^2) w'^{2n-6} d w'^2 \\ &= \left(\frac{\pi}{2}\right) a_{n-1} (n-1)^{-1} (n-2)^{-1} w^{2n-4} \end{aligned} \quad (\text{A.3})$$

Together with the fact that $a_2 = \pi/2$ this results in the form (2.14) for the phase-space volume.

APPENDIX B

Here we give a proof of Eq. (2.8e). The δ function on the left-hand side of this equation implies the following relation

$$w\vec{b} + \vec{H}_{-\vec{b}}(P) = w\vec{b} + \vec{P} - bP^0 + a(\vec{b} \cdot \vec{P})\vec{P} = 0, \quad (\text{B.1})$$

which we have to solve under the condition $P^2 = w^2$ as implied by Eq. (2.8c). It is clear that $\vec{P}$ must be proportional to $\vec{b}$ so that we only have to consider the lengths of the vectors:

$$wb + (1 + ab^2)P - bP^0 = \gamma P - b(P^0 - w) = 0. \quad (\text{B.2})$$

By squaring we immediately see that the only physical solution is $\vec{P} = 0$, hence $P^0 = w$.

The only thing left is to find the inverse Jacobian related to the change of argument in the δ function. The derivative matrix is

$$\Delta^{ij} = \frac{\partial}{\partial P^j} (w b^i + H_{-\vec{b}}^i(P)) \Big|_{\vec{P}=0} = \delta^{ij} + a b^i b^j, \quad (\text{B.3})$$

and the Jacobian is its determinant:

$$\begin{aligned} \det(\Delta) &= \frac{1}{3!} \epsilon^{ijk} \epsilon^{\ell mn} \Delta^{i\ell} \Delta^{jm} \Delta^{kn} \\ &= \frac{1}{6} (\epsilon^{ijk} \epsilon^{ijk} + 3a \epsilon^{ijk} \epsilon^{ljk} b^i b^l) \\ &= 1 + ab^2 = \gamma, \end{aligned} \quad (\text{B.4})$$

where summation over repeated indices is implied. This finishes the proof.

APPENDIX C

We present a proof of the following

Theorem:

Let ϱ_i be random numbers with a uniform distribution in $(0, 1)$. The number x , defined as

$$x = -\ln\left(\prod_{i=1}^n \varrho_i\right), \quad (\text{C.1})$$

is then distributed in the interval $(0, \infty)$ with the probability distribution

$$P_n(x) = x^{n-1} e^{-x} \theta(x) / \Gamma(n). \quad (\text{C.2})$$

Proof:

It is clear that if $x = f(\varrho_1, \varrho_2, \dots, \varrho_n)$, the distribution for x is given by

$$P(x) = \int_0^1 \prod_{i=1}^n d\varrho_i \delta(x - f(\varrho_1, \dots, \varrho_n)). \quad (\text{C.3})$$

We use this to prove Eq. (C.2) by induction. First we take $n = 1$:

$$\begin{aligned} P_1(x) &= \int_0^1 d\varrho_1 \delta(x + \ln \varrho_1) \\ &= \int_0^1 d\varrho_1 \varrho_1 \delta(\varrho_1 - e^{-x}) \\ &= e^{-x} \theta(x), \end{aligned} \quad (\text{C.4})$$

which is of the desired form. Then, using the induction assumption:

$$\begin{aligned}
P_{n+1}(x) &= \int_0^1 d\xi \, P_n(x + \ln \xi) \\
&= \frac{e^{-x}}{\Gamma(n)} \int_0^1 \frac{d\xi}{\xi} (x + \ln \xi)^{n-1} \theta(x + \ln \xi) \\
&= \frac{e^{-x}}{\Gamma(n)} \int_{-x}^0 d(\ln \xi) (x + \ln \xi)^{n-1} \theta(x) \\
&= x^n e^{-x} \theta(x) / (n \Gamma(n)) \, ,
\end{aligned} \tag{C.5}$$

which completes the proof.

APPENDIX D

In this Appendix we make some additional observations about the weight distribution introduced in Eq. (4.11):

$$\begin{aligned}
 W_m &= W_1 \cdot W_2 , \\
 W_1 &= \left[\sum_{i=1}^n y_i \right]^{2h-3} \left[\sum_{i=1}^n y_i^2/x_i \right]^{-1} , \\
 W_2 &= \prod_{i=1}^n y_i/x_i , \tag{D.1}
 \end{aligned}$$

where we have used dimensionless quantities:

$$\begin{aligned}
 x_i &= k_i^0/w , \\
 y_i &= \sqrt{x_i^2 - v_i^2} , \\
 v_i &= m_i/w . \tag{D.2}
 \end{aligned}$$

As stated before, the maximum of W_m is a quantity of computational interest.

Unfortunately, the set of equations which define this extremum:

$$\begin{aligned}
 \sum_{k=1}^n \frac{\partial W_m}{\partial x_k} \cdot \delta x_k &= 0 , \\
 \sum_{k=1}^n \delta x_k &= 0 , \tag{D.3}
 \end{aligned}$$

is very complicated and no known solution for the general case exists.

We can obtain some insight, however, from the following reasoning. The weight W_m corresponds to the compression of the multiparticle phase space induced by the transformation (4.2). Therefore, we can expect the weight to be closest to 1 for those regions in phase space that are compressed by the least

amount, i.e. regions where the resultant vectors k_1^μ differ from the original vectors p_1^μ by the minimal amount. This leads us to the following

Conjecture

The maximum value of W_m is realized for those configurations of the vectors k_1^μ in which the velocities of the massive particles are maximal.

Let us denote the number of massive particles by $\ell (\leq n)$, and assign to them indices 1 through ℓ . If $\ell = 1$, our conjecture assigns the maximum weight to those situations in which all massless momenta are collinear, pointing away from the massive momentum; for $\ell \geq 2$, it requires the massless particles to have zero energy. In that case, we can restrict the product and sums in Eq. (D.1) to the ℓ massive particles only (keeping, of course, the exponent $2n - 3$ unchanged).

From this we arrive at the following important observation. The phase-space regions where the largest weights occur are very small. Often, therefore, a MC-generator weight distribution has an extremely low tail leading up to the theoretical maximum, as mentioned in Section 5. Moreover, the typical signature of a large-weight event is likely to be rejected in a study for which our event generator is appropriate.

These considerations lead us to advocate the use of an effective maximum rather than the absolute maximum, as discussed above.

However, for the sake of completeness, we give expressions valid for some special cases.

$\ell = 2$

Under the above conjecture (all massless particles at rest) the problem reduces to one of two-particle phase space, and we find

$$W_m^{\max}(\ell=2) = \left[(1+v_1^2-v_2^2)^2 - 4v_1^2 \right]^{n-\frac{3}{2}}. \quad (\text{D.4})$$

$\ell = 1$

Since Eq. (D.4) tells us that $W_m^{\max}(\ell = 2)$ is a decreasing function of v_2 , we immediately have

$$W_m^{\max}(\ell=1) = (1 - v_1^2)^{2n-3}. \quad (D.5)$$

Incidentally, the case $\ell = 1$ can be solved without using the conjecture, giving the result (D.5).

$\ell > 2$, all non-zero v_i equal

On symmetry grounds, we can assume the maximum to be reached if all x_i ($i = 1, \dots, \ell$) are equal: $x_i = 1/\ell$ ($x_i = 0$ for $i = \ell+1, \dots, n$). This gives

$$W_m^{\max}(\ell > 2, \text{equal masses}) = \beta^{2n+\ell-5}, \quad \beta = \sqrt{1 - (\ell v_i)^2}. \quad (D.6)$$

In the case where the ℓ masses v_i are not equal, no result is known to us. It turns out, however, that quite a good choice for the effective maximum is obtained by using Eq. (D.6), replacing β by $(1 - M^2)^{1/2}$, where M is the sum of the v_i . This effective maximum behaves in the same way as the observed weight distribution under changes of the parameters, and we expect it to be sufficient except for MC runs with extremely high statistics. Of course, once a MC event occurs with a weight larger than the above estimate, the estimate should be adjusted accordingly.

APPENDIX E

```

SUBROUTINE RAMBO(N,ET,XM,P,WT,LW)
C-----
C
C          RAMBO
C
C          RA(NDOM)  M(OMENTA)  BO(OOSTER)
C
C  A DEMOCRATIC MULTI-PARTICLE PHASE SPACE GENERATOR
C  AUTHORS:  S.D. ELLIS,  R. KLEISS,  J.W. STIRLING
C
C  N  = NUMBER OF PARTICLES (>1, IN THIS VERSION <101)
C  ET = TOTAL CENTRE-OF-MASS ENERGY
C  XM = PARTICLE MASSES ( DIM=100 )
C  P  = PARTICLE MOMENTA ( DIM=(4,100) )
C  WT = WEIGHT OF THE EVENT
C  LW = FLAG FOR EVENT WEIGHTING:
C      LW = 0 WEIGHTED EVENTS
C      LW = 1 UNWEIGHTED EVENTS ( FLAT PHASE SPACE )
C-----
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XM(100),P(4,100),Q(4,100),Z(100),R(4),
      B(3),P2(100),XM2(100),E(100),V(100),IWARN(5)
      DATA ACC/1.D-14/,ITMAX/6/,IBEGIN/0/,IWARN/5*0/
C
C  INITIALIZATION STEP: FACTORIALS FOR THE PHASE SPACE WEIGHT
      IF(IBEGIN.NE.0) GOTO 103
      IBEGIN=1
      TWOPI=8.*DATAN(1.D0)
      PO2LOG=DLOG(TWOPI/4.)
      Z(2)=PO2LOG
      DO 101 K=3,100
101  Z(K)=Z(K-1)+PO2LOG-2.*DLOG(DFLOAT(K-2))
      DO 102 K=3,100
102  Z(K)=(Z(K)-DLOG(DFLOAT(K-1)))
C
C  CHECK ON THE NUMBER OF PARTICLES
103 IF(N.GT.1.AND.N.LT.101) GOTO 104
      PRINT 1001,N
      STOP
C
C  CHECK WHETHER TOTAL ENERGY IS SUFFICIENT; COUNT NONZERO MASSES
104 XMT=0.

```

```

      NM=0
      DO 105 I=1,N
        IF(XM(I).NE.0.D0) NM=NM+1
105  XMT=XMT+DABS(XM(I))
      IF(XMT.LE.ET) GOTO 106
      PRINT 1002,XMT,ET
      STOP

C
C CHECK ON THE WEIGHTING OPTION
106 IF(LW.EQ.1.OR.LW.EQ.0) GOTO 201
      PRINT 1003,LW
      STOP

C
C THE PARAMETER VALUES ARE NOW ACCEPTED
C
C GENERATE N MASSLESS MOMENTA IN INFINITE PHASE SPACE
201 DO 202 I=1,N
      C=2.*RN(1)-1.
      S=DSQRT(1.-C*C)
      F=TWOPI*RN(2)
      Q(4,I)=-DLOG(RN(3)*RN(4))
      Q(3,I)=Q(4,I)*C
      Q(2,I)=Q(4,I)*S*DCOS(F)
202  Q(1,I)=Q(4,I)*S*DSIN(F)

C
C CALCULATE THE PARAMETERS OF THE CONFORMAL TRANSFORMATION
      DO 203 I=1,4
203  R(I)=0.
      DO 204 I=1,N
        DO 204 K=1,4
204  R(K)=R(K)+Q(K,I)
      RMAS=DSQRT(R(4)**2-R(3)**2-R(2)**2-R(1)**2)
      DO 205 K=1,3
205  B(K)=-R(K)/RMAS
      G=R(4)/RMAS
      A=1./(1.+G)
      X=ET/RMAS

C
C TRANSFORM THE Q'S CONFORMALLY INTO THE P'S
      DO 207 I=1,N
        BQ=B(1)*Q(1,I)+B(2)*Q(2,I)+B(3)*Q(3,I)
        DO 206 K=1,3
206  P(K,I)=X*(Q(K,I)+B(K)*(Q(4,I)+A*BQ))
207  P(4,I)=X*(G*Q(4,I)+BQ)

```

```

C
C RETURN FOR UNWEIGHTED MASSLESS MOMENTA
  IF(NM.EQ.0.AND.LW.EQ.1) RETURN
C
C CALCULATE WEIGHT AND POSSIBLE WARNINGS
  WT=PO2LOG
  IF(N.NE.2) WT=(2.*N-4.)*DLOG(ET)+Z(N)
  IF(WT.GE.-180.D0) GOTO 208
  IF(IWARN(1).LE.5) PRINT 1004,WT
  IWARN(1)=IWARN(1)+1
208 IF(WT.LE. 174.D0) GOTO 209
  IF(IWARN(2).LE.5) PRINT 1005,WT
  IWARN(2)=IWARN(2)+1
C
C RETURN FOR WEIGHTED MASSLESS MOMENTA
209 IF(NM.NE.0) GOTO 210
  WT=DEXP(WT)
  RETURN
C
C MASSIVE PARTICLES: RESCALE THE MOMENTA BY A FACTOR X
210 XMAX=DSQRT(1.-(XMT/ET)**2)
  DO 301 I=1,N
    XM2(I)=XM(I)**2
301 P2(I)=P(4,I)**2
  ITER=0
  X=XMAX
  ACCU=ET*ACC
302 F0=-ET
  G0=0.
  X2=X*X
  DO 303 I=1,N
    E(I)=DSQRT(XM2(I)+X2*P2(I))
    F0=F0+E(I)
303 G0=G0+P2(I)/E(I)
  IF(DABS(F0).LE.ACCU) GOTO 305
  ITER=ITER+1
  IF(ITER.LE.ITMAX) GOTO 304
  PRINT 1006,ITMAX
  GOTO 305
304 X=X-F0/(X*G0)
  GOTO 302
305 DO 307 I=1,N
  V(I)=X*P(4,I)
  DO 306 K=1,3

```

```

306 P(K,I)=X*P(K,I)
307 P(4,I)=E(I)
C
C CALCULATE THE MASS-EFFECT WEIGHT FACTOR
  WT2=1.
  WT3=0.
  DO 308 I=1,N
    WT2=WT2*V(I)/E(I)
308 WT3=WT3+V(I)**2/E(I)
    WTM=(2.*N-3.)*DLOG(X)+DLOG(WT2/WT3*ET)
    IF(LW.EQ.1) GOTO 401
C
C RETURN FOR WEIGHTED MASSIVE MOMENTA
  WT=WT+WTM
  IF(WT.GE.-180.D0) GOTO 309
  IF(IWARN(3).LE.5) PRINT 1004,WT
  IWARN(3)=IWARN(3)+1
309 IF(WT.LE. 174.D0) GOTO 310
  IF(IWARN(4).LE.5) PRINT 1005,WT
  IWARN(4)=IWARN(4)+1
310 WT=DEXP(WT)
  RETURN
C
C UNWEIGHTED MASSIVE MOMENTA REQUIRED: ESTIMATE MAXIMUM WEIGHT
401 WT=DEXP(WTM)
  IF(NM.GT.1) GOTO 402
C
C ONE MASSIVE PARTICLE
  WTMAX=XMAX**(4*N-6)
  GOTO 405
402 IF(NM.GT.2) GOTO 404
C
C TWO MASSIVE PARTICLES
  SM2=0.
  PM2=0.
  DO 403 I=1,N
    IF(XM(I).EQ.0.D0) GOTO 403
    SM2=SM2+XM2(I)
    PM2=PM2*XM2(I)
403 CONTINUE
  WTMAX=((1.-SM2/(ET**2))**2-4.*PM2/ET**4)**(N-1.5)
  GOTO 405
C
C MORE THAN TWO MASSIVE PARTICLES: AN ESTIMATE ONLY
404 WTMAX=XMAX**(2*N-5+NM)

```

C

C DETERMINE WHETHER OR NOT TO ACCEPT THIS EVENT

405 W=WT/WTMAX

IF(W.LE.1.D0) GOTO 9005

IF(IWARN(5).LE.5) PRINT 1007,WTMAX,W

IWARN(5)=IWARN(5)+1

9005 CONTINUE

IF(W.LT.RN(5)) GOTO 201

RETURN

1001 FORMAT(' RAMBO FAILS: # OF PARTICLES =',I5,' IS NOT ALLOWED')

1002 FORMAT(' RAMBO FAILS: TOTAL MASS =',D15.6,' IS NOT',

. ' SMALLER THAN TOTAL ENERGY =',D15.6)

1003 FORMAT(' RAMBO FAILS: LW=',I3,' IS NOT AN ALLOWED OPTION')

1004 FORMAT(' RAMBO WARNS: WEIGHT = EXP(',F20.9,') MAY UNDERFLOW')

1005 FORMAT(' RAMBO WARNS: WEIGHT = EXP(',F20.9,') MAY OVERFLOW')

1006 FORMAT(' RAMBO WARNS:',I3,' ITERATIONS DID NOT GIVE THE',

. ' DESIRED ACCURACY =',D15.6)

1007 FORMAT(' RAMBO WARNS: ESTIMATE FOR MAXIMUM WEIGHT =',D15.6,

. ' EXCEEDED BY A FACTOR ',D15.6)

END

FUNCTION RN(IDMY)

DATA 1/65539/

DATA C/Z39200000/

1 IDMY=IDMY

I=I*69069

IF(I.LE.0) I=I + 2147483647 + 1

J=I/256

RN=C*FLOAT(256*J)

IF(RN.NE.0.) RETURN

PRINT 2,I,C,J,RN

2 FORMAT(' RN WARNING: WRONG VALUE OCCURRED',

. ' I,C,J,RN :',I15,D15.6,I15,D15.6)

GOTO 1

END