A hierarchical phase space generator for QCD antenna structures

André van Hameren[1] and Costas G. Papadopoulos[2]
Institute of Nuclear Physics, NCSR Δημόκϱιτος, 15310 Athens, Greece

February 1, 2008

Abstract

We present a "hierarchical" strategy for phase space generation in order to efficiently map the antenna momentum structures, typically occurring in QCD amplitudes.

1 Introduction

The reliable description of multi-jet production at the LHC [1] will be an important issue. This is not only related to the study of QCD in multi-parton final states but it is also very important in order to estimate several backgrounds for new physics effects. For instance, new unstable massive particles that decay to many partons may be discovered at the LHC only when a reliable description of these final states is established.

In this respect, apart from the problem of computing scattering matrix elements with many particles, also the efficient phase space generation is of great importance, because the scattering amplitudes in QCD exhibit strong peaking structures in phase space, which have to be taken into account by the generation algorithm. Flat phase space generators, like RAMBO [2], will not be adequate for this task. In the last years several methods to efficiently integrate the peaking structures of the scattering amplitudes have emerged, and have been used in several contexts [3]. For instance, PHEGAS [4] is an example where an efficient, automated, mapping of all possible peaking structures of a given scattering process has been established. The algorithm is based on the "natural" mappings dictated by the Feynman graphs contributing to the given process, so that the number of kinematical channels used to generate the phase space is equal to the number of Feynman graphs. Using adaptive methods, like multi-channel optimization [5] and by throwing away channels that are negligible, we may end up with a few channel generator exhibiting high efficiency, as is indeed the case in $n(+\gamma)$-fermion production in e^+e^- collisions. In contrast, the QCD scattering amplitudes point towards the opposite direction: large number of Feynman

[1] E-mail: andrevh@inp.demokritos.gr

[2] E-mail: Costas.Papadopoulos@cern.ch

graphs which means large number of kinematical channels which, moreover, contribute equally to the result.

A way out off this problem may be based on the long-standing remark that $n+2$-gluon amplitude may be described by a very compact expression when special helicities are assigned to the gluons, which, combined with the leading color approximation, results to

$$\sum_c |\mathcal{M}|^2 = 8\left(\frac{N_\mathrm{c}}{2}\right)^n (N_\mathrm{c}^2-1)\sum_{1\leq i<j}^{n+2}(p_i\cdot p_j)^4 \sum_{P(2,\ldots,n+2)} A_{n+2}(p_1,\ldots,p_{n+2}) \ , \tag{1}$$

where N_c refers to the number of colors,

$$A_{n+2}(p_1,\ldots,p_{n+2}) := [\ (p_1\cdot p_2)(p_2\cdot p_3)\cdots(p_{n+1}\cdot p_{n+2})(p_{n+2}\cdot p_1)\ ]^{-1} \ , \tag{2}$$

and the sum over all permutations of the 2^{nd} to the $(n+2)^{\mathrm{nd}}$ argument of this function is taken, with the exception of those that are equivalent under reflection $i \mapsto n+4-i$ [6].

`SARGE` [7] is the first known example of a phase space generator that deals with the momentum structures entering the above expression, namely with (2), known as *antenna structures*. The algorithm is based on the "democratic" strategy to generate the n body phase space, as is the case for `RAMBO`, and it makes use of the scale symmetry of the antenna to achieve the required goal.

In this paper, we study the "hierarchical" strategy for phase space generation in order to efficiently map the momentum antenna structures. The idea is as follows. Using the standard two-body phase space (neglecting factors of 2π)

$$d\Phi_2(P;s_1,s_2;p_1,p_2) := d^4p_1\,\delta_+(p_1^2-s_1)\,d^4p_2\,\delta_+(p_2^2-s_2)\,\delta^4(P-p_1-p_2) \ , \tag{3}$$

we decompose the phase space as

$$\begin{aligned} d\Phi_n(P;p_1\ldots,p_n) &:= \Big(\prod_{i=1}^n d^4p_i\,\delta_+(p_i^2-\sigma_i)\Big)\delta^4\Big(\sum_{i=1}^n p_i - P\Big) \\ &= ds_{n-1}\,d\Phi_2(Q_n;\sigma_n,s_{n-1};p_n,Q_{n-1}) \\ &\times ds_{n-2}\,d\Phi_2(Q_{n-1};\sigma_{n-1},s_{n-2};p_{n-1},Q_{n-2}) \\ &\qquad\vdots \\ &\times ds_2\,d\Phi_2(Q_3;\sigma_3,s_2;p_3,Q_2)\ d\Phi_2(Q_2;\sigma_2,\sigma_1;p_2,p_1)\ . \end{aligned} \tag{4}$$

The task is to express the phase space in terms of the invariants $p_i\cdot p_j$ appearing in the antenna structure (2), so that, using a suitable mapping, we can construct a density that, apart from constant and soft terms, will be identical to this antenna structure.

In the first section, we describe the basic building block of the algorithm, which is the expression of the two-body phase space in terms of the scaled invariants. In the second section, we demonstrate how this basic building block can be used in a sequential way to produce the full antenna. Finally in the third section, some details concerning the numerical implementation of the algorithm as well as comparisons with known generators is given. The Appendices present all relevant generation algorithms from which the exact functioning of the generator can be reconstructed.

2 The hierarchical antenna

2.1 The basic building block

To illustrate the idea we consider the generation of the 2-body phase space (3) when two massless antenna momenta q_1, q_2 are given. The momentum P can be decomposed as

$$P^\mu = r q_1^\mu + A_1^\mu =: A_1^\mu + B_1^\mu \quad \text{with} \quad r = P^2/(2P\cdot q_1) \ \ ,$$

so that the Sudakov parameterization of p_1 is given by

$$p_1^\mu = a_1 A_1^\mu + b_1 B_1^\mu + k_1^\mu \ \ .$$

where the variables a_1 and b_1 are given by

$$a_1 = \frac{p_1\cdot B_1}{A_1\cdot B_1} \quad , \quad b_1 = \frac{p_1\cdot A_1}{A_1\cdot B_1} \ \ .$$

The same can be done in terms of p_2 and q_2, and in the center-of-mass frame (CMF) of P, where $P = (\sqrt{s}\,, 0, 0, 0)$, $\cos(\angle(\vec{q}_1, \vec{q}_2)) = \mathrm{c}$ and $\mathrm{s} = \sqrt{1-\mathrm{c}^2}$, one can choose

$$\begin{aligned}
A_1 &= \tfrac{1}{2}\sqrt{s}(1,0,0,-1) \quad , \quad & A_2 &= \tfrac{1}{2}\sqrt{s}(1,0,-\mathrm{s},-\mathrm{c}) \ \ , \\
B_1 &= \tfrac{1}{2}\sqrt{s}(1,0,0,1) \quad , \quad & B_2 &= \tfrac{1}{2}\sqrt{s}(1,0,\mathrm{s},\mathrm{c}) \ \ , \\
k_1 &= (0,x_1,y_1,0) \quad , \quad & k_2 &= (0,x_2,y_2\mathrm{c},-y_2\mathrm{s}) \ \ .
\end{aligned}$$

The phase space can now be completely expressed in terms of a_1 and a_2, leading to

$$d\Phi_2(P;p_1,p_2) = da_1\, da_2\, \Pi^{(-1/2)}\Theta(\Pi) \ \ , \tag{5}$$

with

$$\begin{aligned}
\Pi(a_1,a_2) = {} & 4\mathrm{s}^2[(1-a_2+\bar{s}_2-\bar{s}_1)a_2-\bar{s}_2] \\
& -[(1-2a_1-\bar{s}_1+\bar{s}_2)+(1-2a_2-\bar{s}_1+\bar{s}_2)\mathrm{c}]^2 \ \ ,
\end{aligned} \tag{6}$$

where $\bar{s}_{1,2} = s_{1,2}/s$, and where Θ is the step function. In terms of Lorentz invariants, the parameters a_1 and a_2 are given by

$$a_1 = \frac{q_1\cdot p_1}{q_1\cdot P} \quad , \quad a_2 = \frac{p_2\cdot q_2}{P\cdot q_2} \ \ . \tag{7}$$

So in order to obtain a two-body phase space with a density which depends on the invariants a_1, a_2 following some given function $f(a_1, a_2)$, one has to generate a_1, a_2 following a density proportional to $f(a_1, a_2) \times \Pi^{(-1/2)}(a_1, a_2)$ in the region where $\Pi(a_1, a_2) > 0$, and construct the momenta following the Sudakov parameterization. Explicitly, the direct construction is given by

$$\begin{aligned}
&p_1^0 \leftarrow (s + s_1 - s_2)/(2\sqrt{s}) \quad , \\
&p_1^3 \leftarrow p_1^0 - \sqrt{s}\,a_1 \quad , \\
&p_1^2 \leftarrow (\ (\sqrt{s} - p_1^0 - \sqrt{s}\,a_2) + \mathrm{c}p_1^3\ )/\mathrm{s} \quad , \\
&p_1^1 \leftarrow \epsilon(\ (p_1^0)^2 - s_1 - (p_1^2)^2 - (p_1^3)^2\ )^{1/2} \quad ,
\end{aligned}$$

where ϵ should be a fair random variable which can take values $+1$ and -1. For more details about this procedure, we refer the reader to the Appendix A.

2.2 Antenna generation

In the hierarchical/sequential approach, the generation strategy proceeds through a sequence of two-body phase space generations following the decomposition (4). At each two-body generation, one final-state momentum p_k is generated, together with the sum Q_{k-1} of the remaining final-state momenta to be generated. This suggest to label the momenta in a way opposite to the order of generation, so first p_n, Q_{n-1} are generated, then p_{n-1}, Q_{n-2} and so on. The starting point is the CMF of the initial momenta q_1 and q_2 with $Q_n = q_1 + q_2$ being the overall momentum. The CMF of momentum Q_k we denote by CMF$_k$. The pair p_k, Q_{k-1} is generated by generating variables $a_1^{(k)}, a_2^{(k)}$ and constructing the momenta as described before. These variables are now equal to

$$a_1^{(k)} = \frac{p_{k+1} \cdot p_k}{p_{k+1} \cdot Q_k} \quad \text{and} \quad a_2^{(k)} = \frac{q_2 \cdot Q_{k-1}}{q_2 \cdot Q_k} \quad .$$

This happens in CMF$_k$, so in order to obtain p_k, Q_{k-1}, the constructed momenta have to be boosted such that $(\sqrt{Q_k^2}, 0, 0, 0)$ is transformed to Q_k.

We would like to generate the momenta following a density that is proportional to

$$A_{n+2}(q_1, p_n \ldots, p_1, q_2) = [\ (q_1 \cdot p_n)\,(p_n \cdot p_{n-1})\ \ldots (p_1 \cdot q_2)\ ]^{-1} \quad . \tag{8}$$

Since the integrand is infra-red singular, a cutoff on the invariants is necessary. Therefore, we define a symmetric matrix σ_{ij} which encodes the restrictions on the momenta through

$$\sigma_{ii} = \sigma_i = p_i^2 \quad , \quad \sqrt{\sigma_i \sigma_j} \leq \sigma_{ij} \leq p_i \cdot p_j \quad \text{and} \quad \varSigma_k := \sum_{i=1}^{k} \sigma_i \quad . \tag{9}$$

Before we proceed, we do three observations. Firstly, we have

$$p_{k+1} \cdot Q_k = (Q_{k+1}^2 - Q_k^2 - p_{k+1}^2)/2 \quad .$$

Secondly, we have

$$\frac{s_{k+1}-\Sigma_{k+1}}{(s_{k+1}-\sigma_{k+1}-s_k)\,(s_k-\Sigma_k)} = \frac{d}{ds_k}\log\left(\frac{s_k-\Sigma_k}{s_{k+1}-\sigma_{k+1}-s_k}\right) =: g_{k+1}(s_k) \ , \tag{10}$$

and thirdly, we can write

$$A_{n+2}(q_1,p_n\dots,p_1,q_2) = \frac{1}{2^{n-1}}(s_n-\Sigma_n)(q_1\cdot Q_n)(q_2\cdot Q_n)\left(\prod_{k=n}^{3} g_k(s_{k-1})\frac{1}{a_1^{(k)}a_2^{(k)}}\right)\frac{1}{a_1^{(2)}a_2^{(2)}} \ ,$$

with $s_n = Q_n^2$, $p_{n+1} = q_1$ and $Q_1 = p_1$. These observations suggest that the phase space generation

$$\begin{array}{llll}
ds_{n-1}\,g_n(s_{n-1}) & da_1^{(n)}\,\dfrac{1}{a_1^{(n)}} & da_2^{(n)}\,\dfrac{1}{a_2^{(n)}} & \Pi_{(n)}^{(-1/2)}\Theta(\Pi_{(n)}) \\
ds_{n-2}\,g_{n-1}(s_{n-2}) & da_1^{(n-1)}\,\dfrac{1}{a_1^{(n-1)}} & da_2^{(n-1)}\,\dfrac{1}{a_2^{(n-1)}} & \Pi_{(n-1)}^{(-1/2)}\Theta(\Pi_{(n-1)}) \\
 & \vdots & & \\
ds_2\,g_3(s_2) & da_1^{(3)}\,\dfrac{1}{a_1^{(3)}} & da_2^{(3)}\,\dfrac{1}{a_2^{(3)}} & \Pi_{(3)}^{(-1/2)}\Theta(\Pi_{(3)}) \\
 & da_1^{(2)}\,\dfrac{1}{a_1^{(2)}} & da_2^{(2)}\,\dfrac{1}{a_2^{(2)}} & \Pi_{(2)}^{(-1/2)}\Theta(\Pi_{(2)}) \ ,
\end{array} \tag{11}$$

will lead to a density for the momenta that is proportional to A_{n+2}. Three variables are generated in each CMF$_k$, namely s_{k-1}, $a_1^{(k)}$ and $a_2^{(k)}$. Just as the integration of s_{k-1} (10), also the integration of $a_1^{(k)}, a_2^{(k)}$ results in a volume factor that depends on the corresponding variables generated in CMF$_{k+1}$. As we will show in Appendix A, however, these factors are logarithmic functions of their arguments and exhibit a non-singular behavior, and we call them *soft* factors. The total actual density will therefore be the product of $n-1$ soft factors times the antenna structure under consideration.

In the end, we want to generate all permutations in the momenta of (8). Those for which q_1 and q_2 each appear in two factors (none of which is $q_1\cdot q_2$) cannot be obtained by simple re-labeling. In order to obtain these, we observe that they can be decomposed into two antennas, namely

$$A_{m+2}(q_1,p_m,p_{m-1},\dots,p_2,p_1,q_2)\times A_{n-m+2}(q_2,p_n,p_{n-1},\dots,p_{m+2},p_{m+1},q_1) \tag{12}$$

and each of these can be generated after the decomposition,

$$\begin{aligned} d\Phi_n(P;p_1\dots,p_n) &= ds_m\,ds_{n-m}\,d\Phi_2(Q_n;s_m,s_{n-m};Q_m,Q_{n-m}) \\ &\quad\times d\Phi_m(Q_m;p_1,\dots,p_m)\,d\Phi_{n-m}(Q_{n-m};p_{m+1},\dots,p_n) \ . \end{aligned} \tag{13}$$

In order to combine the two sub-antennas to the required antenna structure, we have to take into account in the first decomposition a density that is proportional to

$$\frac{\Theta(\sqrt{s_n}-\sqrt{s_m}-\sqrt{s_{n-m}}\,)}{(q_1\cdot Q_m)(q_1\cdot Q_{n-m})(q_2\cdot Q_m)(q_2\cdot Q_{n-m})\,s_m\,s_{n-m}} \ . \tag{14}$$

The case of $m=1$ is still special. Then, the first step in (11) should be replaced by

$$ds_{n-1}g'(s_{n-1})\ \ da_1^{(n)}\ \ \ da_2^{(n)}\ \ \Pi_{(n)}^{(-1/2)}\Theta(\Pi_{(n)})\,\frac{1}{a_1^{(n)}\,a_2^{(n)}\,(1-a_1^{(n)})(1-a_2^{(n)})}\ \ , \qquad (15)$$

and the rest of the sequence should go on with the replacement of p_n by p_{n+1} at the second step. For the density g' we refer to Appendix A. There, we have collected all integrals and generation algorithms related to the antenna generation described so far.

2.3 Open antennas

As it will be clear from the numerical analysis presented in the next section, the soft factors appearing in the description of the QCD antenna contribute to a certain extent to the variance of the Monte Carlo integration. There is an alternative approach, that still follow the hierarchical/sequential generation strategy, and give better results. It based on the observation that the production of an 'open' antenna structure, namely one where the last product $q_2 \cdot p_1$ is missing, is simpler, since it can be constructed without using the variables $a_2^{(k)}$. They can be replaced by flatly generated azimuthal angles, to that $da_2^{(k)}\Pi_{(k)}^{(-1/2)}\Theta(\Pi_{(k)}) \to d\varphi^{(k)}$. The basic decomposition therefore becomes

$$ds_{n-1}g_n(s_{n-1})\,d\varphi^{(n)}\,da_1^{(n)}\frac{1}{a_1^{(n)}}\times ds_{n-2}g_{n-1}(s_{n-2})\,d\varphi^{(n-1)}\,da_1^{(n-1)}\frac{1}{a_1^{(n-1)}}\times\cdots$$
$$\cdots\times ds_2g_3(s_2)\,d\varphi^{(3)}\,da_1^{(3)}\frac{1}{a_1^{(3)}}\times d\varphi^{(2)}\,da_1^{(2)}\frac{1}{a_1^{(2)}}\ \ . \qquad (16)$$

This way, we will get the antenna density (8) without the factor $p_1 \cdot q_2$ in the denominator, which is the reason why we call this an 'open' antenna. A 'closed' antenna can be obtained using the fact that, by combining two open antennas, one can choose for another factor from the antenna string to be missing. Then, a multi-channeling procedure can be performed with these different choices, leading to a density that is, roughly speaking, proportional to

$$\frac{(q_1\cdot p_n)+(p_n\cdot p_{n-1})+\cdots+(p_3\cdot p_2)+(p_2\cdot p_1)+(p_1\cdot q_2)}{(q_1\cdot p_n)(p_n\cdot p_{n-1})\cdots(p_3\cdot p_2)(p_2\cdot p_1)(p_1\cdot q_2)}\ \ .$$

To get the different choices, a first splitting of Q_n into Q_m and Q_{n-m} has to be performed, after which open antennas are generated from each of these, one with q_1 and the other with q_2 as initial momentum. For details, we refer to Appendix B.

3 Results

In this section, we present results obtained by SARGE and HAAG[3], the program that implements the hierarchical algorithm described before. In order to be as general as possible,

[3] HAAG stands for: Hierarchical AntennA Generation.

the only cut we apply is

$$(p_i + p_j)^2 \geq s_0 \quad ,$$

where $i, j (i \neq j)$ runs from 1 to $n+2$ where n is the number of final-state particles. Unless explicitly mentioned differently, we use $s_0 = 900\,\mathrm{GeV}^2$ and the total energy $\sqrt{s} = 1000\,\mathrm{GeV}$. Moreover, all particles are assumed to be massless in order to compare with `SARGE`, with which only massless particles can be treated.

As it was mentioned in the introduction, we are interested in integrating sums of QCD antenna structures (2). We start by considering the simplest case, namely integrating the function

$$s^2[\,(p_1 \cdot p_3)(p_3 \cdot p_4)(p_4 \cdot p_2)(p_2 \cdot p_5) \ldots (p_{n+2} \cdot p_1)\,]^{-1} \tag{17}$$

that corresponds to a given permutation of the momenta, namely $(1, 3, 4, 2, 5, \ldots, n+2)$. In Table 1 we give the results for `SARGE`, `HAAG` with open antenna generation, and `HAAG(C)` with closed antenna generation. In all three codes the same single channel, corresponding to (17), has been used in the generation. N_{gen} and N_{acc} are the number of generated and accepted events, and by f we define

$$f := \frac{V_2}{I^2} \quad ,$$

where V_2 is the quadratic variance and I is the estimated integral. f is clearly a measure of the efficiency of the generator. Moreover ε, defined as

$$\varepsilon := \frac{< w >}{w_{\max}} \quad ,$$

is the usual generation efficiency related for instance to 'unweighted' events in a realistic simulation. The results agree well, and exhibit the fact that the generated densities of the generators the hierarchical type are much closer to the integrand. Moreover, the closed

jets	algorithm	N_{gen}	N_{acc}	I	ΔI	f	$\varepsilon(\%)$
4	`SARGE`	1×10^5	34853	$.251 \times 10^{-9}$	$.734 \times 10^{-11}$	85.9	0.34
	`HAAG`	5×10^4	31193	$.260 \times 10^{-9}$	$.280 \times 10^{-11}$	5.75	1.77
	`HAAG(C)`	5×10^4	28366	$.256 \times 10^{-9}$	$.252 \times 10^{-11}$	4.84	4.22
5	`SARGE`	2.5×10^5	30960	$.438 \times 10^{-10}$	$.153 \times 10^{-11}$	307	0.23
	`HAAG`	6.5×10^4	29855	$.442 \times 10^{-10}$	$.640 \times 10^{-12}$	13.6	1.02
	`HAAG(C)`	6.5×10^4	24345	$.441 \times 10^{-10}$	$.706 \times 10^{-12}$	16.7	1.04
6	`SARGE`	1×10^6	28383	$.487 \times 10^{-11}$	$.164 \times 10^{-12}$	1141	0.21
	`HAAG`	1.2×10^5	32070	$.487 \times 10^{-11}$	$.658 \times 10^{-13}$	21.9	1.48
	`HAAG(C)`	1.2×10^5	25040	$.485 \times 10^{-11}$	$.886 \times 10^{-13}$	40.1	0.69

Table 1: Results for the single-channel integration/generation.

antenna algorithm HAAG(C) becomes less efficient compared to the open one as the number of particles increases. The same picture is reproduced for an arbitrary permutation.

For a realistic QCD calculation, the integrated function may be approximated by a sum over permutations. Therefore, an efficient generator has to include all possible channels, where each channel corresponds to a given permutation of the momenta. In that case, a multi-channeling optimization procedure can be applied, which is incorporated in HAAG. In order to study the efficiency of this optimization we consider the same integration as before, but with all channels contributing to the generation and allowing for optimization. In this optimization procedure, we discard channels that contribute less than a certain pre-determined fraction to the set of available channels. It is expected, of course, that in end the right permutation will be 'chosen' by the optimization. This is indeed the case and the results are presented in Table 2. We see that the optimization results to a picture close to the one obtained with the single channel generation, with some noticeable improvement in the case of SARGE. We also include results with SARGE.n, a slightly different version, described in Appendix C.

jets	algorithm	N_{gen}	N_{acc}	I	ΔI	f	$\varepsilon(\%)$
4	SARGE	1×10^5	52516	$.262 \times 10^{-9}$	$.294 \times 10^{-11}$	12.6	1.29
	SARGE.n	1×10^5	46529	$.260 \times 10^{-9}$	$.298 \times 10^{-11}$	13.2	1.55
	HAAG	5×10^4	34293	$.257 \times 10^{-9}$	$.210 \times 10^{-11}$	3.36	4.28
	HAAG(C)	5×10^4	29736	$.259 \times 10^{-9}$	$.227 \times 10^{-11}$	3.84	3.91
5	SARGE	2.5×10^5	32315	$.422 \times 10^{-10}$	$.106 \times 10^{-11}$	159	0.44
	SARGE.n	$2. \times 10^5$	30994	$.440 \times 10^{-10}$	$.807 \times 10^{-12}$	67.2	0.83
	HAAG	6.5×10^4	31063	$.444 \times 10^{-10}$	$.503 \times 10^{-12}$	8.32	1.17
	HAAG(C)	6.5×10^4	24179	$.436 \times 10^{-10}$	$.593 \times 10^{-12}$	12.03	1.84
6	SARGE	1×10^6	29138	$.476 \times 10^{-11}$	$.145 \times 10^{-12}$	933	0.45
	SARGE.n	1×10^6	35445	$.492 \times 10^{-11}$	$.109 \times 10^{-12}$	492	0.25
	HAAG	1.2×10^5	33278	$.483 \times 10^{-11}$	$.595 \times 10^{-13}$	18.2	1.19
	HAAG(C)	1.2×10^5	24126	$.471 \times 10^{-11}$	$.749 \times 10^{-13}$	30.3	1.21

Table 2: Results for the all-channel generation with optimization.

As is the case for any multi-channel generator, a computational complexity problem arises when the number of channels increases. For instance, in our case we are facing a number of $\frac{1}{2}(n+1)!$ channels! On the other hand, it is also clear that the channels we are considering have a large overlap in most of the available phase space. It is therefore worth to investigate the dependence of the integration efficiency on the number of channels used. This is presented in Table 3, where the full antenna

$$s^2 \sum_{P(2,\ldots,n+2)} [\ (p_1 \cdot p_3)(p_3 \cdot p_4)(p_4 \cdot p_2)(p_2 \cdot p_5) \ldots (p_{n+2} \cdot p_1)\ ]^{-1} \tag{18}$$

is integrated, using a number of channels that has been selected on a random basis. We see the rather interesting phenomenon that a decent description can be achieved with a much smaller number of channels. Variations of this technique of using only subsets of channels, for example choosing another subset after each step of multi-channel optimization, lead to the same picture.

# channels	2520	1500	1000	500	200	50	10
f	5.33	5.37	5.48	5.72	6.14	11.6	84.7
$N_{\rm acc}$	26630	26521	26437	26676	27009	27190	27205
$\varepsilon(\%)$	11.2	13.1	11.6	7.1	7.5	1.7	0.28

Table 3: All-channel integration with subsets of channels for generation.

The complete results of the integration of the full antenna are presented in Table 4. We see that `HAAG` has a much better f factor than `SARGE`. On the other hand the ε exhibits a less dramatic effect. This is related to the fact that `SARGE` generates a phase space that is much larger than the one defined by the cut on s_0. In that sense, if the main time consumption in a given computation is spent over the evaluation of the integrand (matrix element squared), it is more fair to compare the square of the estimated expected error, normalized by the number of accepted events $N_{\rm acc}$. In that case we see that `HAAG` is still 2-3 times more efficient, and if we consider a smaller cut, namely $\sqrt{s_0} = 10$ GeV, this gain goes up to an order of magnitude (Table 5).

jets	algorithm	$N_{\rm gen}$	$N_{\rm acc}$	I	ΔI	$\varepsilon(\%)$	f
4	`SARGE`	1×10^5	47483	$.166 \times 10^{-7}$	$.115 \times 10^{-9}$	4.21	4.8
	`HAAG`	6×10^4	42019	$.167 \times 10^{-7}$	$.810 \times 10^{-10}$	12.01	1.4
5	`SARGE`	3×10^5	39095	$.176 \times 10^{-7}$	$.162 \times 10^{-9}$	3.27	25.6
	`HAAG`	1.2×10^5	55234	$.177 \times 10^{-7}$	$.856 \times 10^{-10}$	7.53	2.7
6	`SARGE`	1.5×10^6	44529	$.157 \times 10^{-7}$	$.135 \times 10^{-9}$	2.95	109
	`HAAG`	1.8×10^5	47911	$.161 \times 10^{-7}$	$.905 \times 10^{-10}$	7.15	5.7
7	`SARGE`	1×10^7	47766	$.123 \times 10^{-7}$	$.988 \times 10^{-10}$	3.02	642
	`HAAG`	3.6×10^5	45599	$.123 \times 10^{-7}$	$.241 \times 10^{-10}$	5.11	13
8	`SARGE`	1×10^8	53560	$.784 \times 10^{-8}$	$.554 \times 10^{-10}$	3.29	4998
	`HAAG`	1×10^6	49206	$.789 \times 10^{-8}$	$.496 \times 10^{-10}$	6.30	39

Table 4: Results for the all-channel integration.

For a realistic calculation of the cross section of a QCD process, one may assume that the time it takes to perform one evaluation of the integrand is much larger than the time it takes to generate one accepted event and to calculate the weight. This means that the computing time is completely determined by the number of accepted events $N_{\rm acc}$. We introduce

$$\frac{N_{\rm acc} f}{N_{\rm gen}}$$

jets	algorithm	$N_{\rm gen}$	$N_{\rm acc}$	I	ΔI	$\varepsilon(\%)$	f
4	SARGE	1×10^5	60986	$.364 \times 10^{-6}$	$.548 \times 10^{-8}$	0.631	22.7
	HAAG	6×10^4	46763	$.366 \times 10^{-6}$	$.235 \times 10^{-8}$	4.34	2.47
5	SARGE	2×10^5	43150	$.619 \times 10^{-6}$	$.165 \times 10^{-7}$	0.29	142
	HAAG	1×10^5	56034	$.643 \times 10^{-6}$	$.465 \times 10^{-8}$	1.84	5.23
6	SARGE	1×10^6	67811	$.114 \times 10^{-5}$	$.257 \times 10^{-7}$	0.28	502
	HAAG	1.4×10^5	51983	$.111 \times 10^{-5}$	$.883 \times 10^{-8}$	2.50	8.83
7	SARGE	5×10^6	84391	$.186 \times 10^{-5}$	$.346 \times 10^{-7}$	0.176	1723
	HAAG	2×10^5	44015	$.192 \times 10^{-5}$	$.177 \times 10^{-7}$	2.24	16
8	SARGE	5×10^7	175541	$.354 \times 10^{-5}$	$.517 \times 10^{-7}$	.119	10618
	HAAG	5×10^5	58874	$.350 \times 10^{-5}$	$.289 \times 10^{-7}$	1.65	34

Table 5: Results for the all-channel integrationwith $s_0 = 100\,\mathrm{GeV}^2$.

as a measure of the computing time. For a realistic calculation, one has to multiply this number by the evaluation time of the integrand, and devide by the square of the relative error one wants to reach. Figure 1 shows this quantity as function of the number of produced partons using the data of Table 5. According to this graph, a calculation with SARGE would take 10 times longer than the calculation with HAAG.

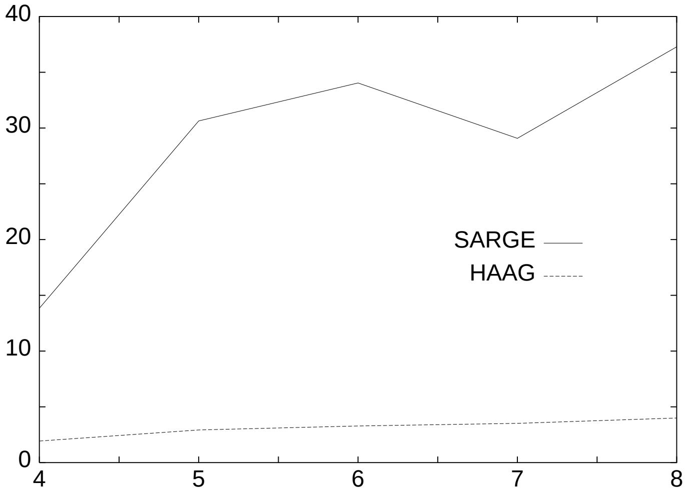

Figure 1: $N_{\rm acc} f / N_{\rm gen}$ (a measure of computing time) as function of the number of produced partons.

Finally, Figure 2 shows the dependence of the result on the value of the infrared cutoff $\sqrt{s_0}$ and the number of produced partons. The function clearly exhibits a negative power behavior for s_0. Moreover, the curve becomes steeper with increasing number of jets, suggesting that at least the leading power is related to the final-state multiplicity.

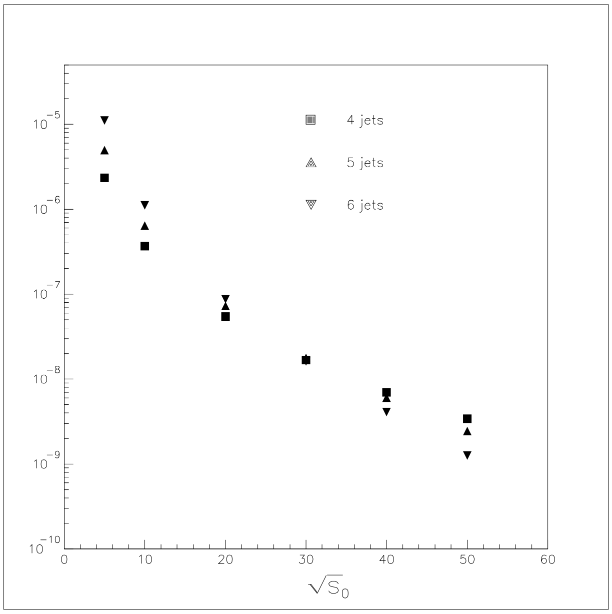

Figure 2: The integral of the full antenna as function of $\sqrt{s_0}$ for different values of the number of produced partons.

4 Conclusions

HAAG exhibits an improved efficiency compared to SARGE for multi-parton calculations. It is also more powerful in describing densities where a partial symmetrization over the permutation space is considered. Finally, HAAG makes no fundamental distinction among massless and massive particles, so it can be used for an arbitrary multi-partonic process.

Acknowledgments

The authors would like to thank Ronald Kleiss for proof reading this paper. The research has been financially supported by the European Union under contract number HPRN-CT-2000-00149.

References

[1] `http://lhc.web.cern.ch/lhc/`.

[2] W.J. Stirling, R. Kleiss and S.D. Ellis, *A new Monte Carlo treatment of multiparticle phase space at high energy*, Comp. Phys. Comm. **40** (1986) 359.

[3] D. Bardin *et al.*, *Event generators for W W physics,* `hep-ph/9709270`.

[4] C.G. Papadopoulos, *PHEGAS: A phase space generator for automatic cross-section computation*, Comp. Phys. Comm. **137** (2001) 247-254, `hep-ph/0007335`.

[5] R. Kleiss and R. Pittau, *Weight optimization in multichannel Monte Carlo*, Comp. Phys. Comm. **83** (1994) 141-146.

[6] J.G.M. Kuijf, *Multiparton production at hadron colliders*, PhD thesis, University of Leiden, 1991.

[7] A. van Hameren, R. Kleiss and P. Draggiotis, *SARGE: An Algorithm for generating QCD antennas*, Phys. Lett. **B483** (2000) 124-130, `hep-ph/0004047`.
A. van Hameren and R. Kleiss, *Generating QCD-antennas*, Eur. Phys. J. **C17** (2000) 611-621, `hep-ph/0008068`.

[8] A. van Hameren, *Loaded Dice in Monte Carlo*, PhD thesis, University of Nijmegen, 2001, `hep-ph/0101094`.

Appendices

The following appendices contain details about the generation of the various random variables necessary to build phase space with the desired density. The techniques used to achieve this are *inversion*, *rejection* and *multi-channeling*. For details about these techniques, we refer to [8]. We only want to mention that inversion is applicable if one has an analytic expression (with reasonable complexity) of the inverse of the indefinite integral of the density. Rejection can be applied if one knows a function which is strictly larger than the density and is proportional to a density one *is* able to generate. The efficiency is given by one divided by the integral of that function. Multi-channeling can be used if the density can be written as the weighted sum of densities, each of which one *is* able to generate, and if the weights are positive.

A Closed antenna

In this appendix we present all relevant algorithms for the generation of the closed antenna. More specifically, in the following we describe the generation of the three variables s_{k-1}, $a_1^{(k)}$ and $a_2^{(k)}$ that are needed to describe the antenna at the $k-$th CMF. Most of the time the superscript (k) is omitted for convenience.

A1 Generation of s_{k-1}

Since the indefinite integral is given in (10), we see that s_{k-1} can be generated with inversion. The limits are given by

$$s_k - \Delta_{k-1} \geq s_{k-1} \geq \Lambda_{k-1} \quad \text{with} \quad \Lambda_k = \mathit{\Sigma}_k + \sum_{i\neq j}^{k} \sigma_{ij} \ , \quad \Delta_k = \sigma_{k+1} + 2\sum_{i=1}^{k} \sigma_{k+1,i} \ , \tag{19}$$

and the weight factor is

$$\log\left(\frac{s_k - \mathit{\Sigma}_{k-1} - \Delta_{k-1}}{\Delta_{k-1} - \sigma_k}\right) + \log\left(\frac{s_k - \sigma_k - \Lambda_{k-1}}{\Lambda_{k-1} - \mathit{\Sigma}_{k-1}}\right) \ .$$

In the case of $m = 1$ (15) we simply have

$$\int ds_{k-1} \frac{1}{s_{k-1} - \mathit{\Sigma}_{k-1}} = \log(s_{k-1} - \mathit{\Sigma}_{k-1}) \ , \tag{20}$$

so that the weight factor is

$$\log\left(s_k - \Delta_{k-1} - \mathit{\Sigma}_{k-1}\right) - \log\left(\Lambda_{k-1} - \mathit{\Sigma}_{k-1}\right) \ .$$

A2 Generation of a_1, a_2

The general integral corresponding to the generation of a_1, a_2 is given by

$$\int da_1 da_2 \frac{\Theta(\,\Pi(a_1,a_2)\,)\,\Theta(\,a_1 - a_1^{(0)}\,)\,\Theta(\,a_2 - a_2^{(0)}\,)}{a_1 a_2 \sqrt{\Pi(a_1,a_2)}} \ , \tag{21}$$

where Π is defined in (6), and $a_1^{(0)}, a_2^{(0)}$ are possibly necessary infra-red cut-offs. We shall analyze this integral by first integrating over the a_2-variable, and then over the a_1-variable. The generation has then to be performed in the opposite order: first a_1 and then a_2. As we shall see, the inclusion of the cut-off on a_1 does not lead to complications, but the inclusion of *both* cut-offs *does*. For that case, we see two solutions. Firstly, we can replace the integral by

$$\int da_1 da_2 \frac{\Theta(\,\Pi(a_1,a_2)\,)\,\Theta(\,a_1 - a_1^{(0)}\,)}{a_1(a_2+h)\sqrt{\Pi(a_1,a_2)}} \ ,$$

where h is related to $a_2^{(0)}$. This, of course, changes the actual density with which a_1 and a_2 are generated, but may, for small h, still be considered suitable for the desired antenna structure. Also, the final antenna generator will cover phase space less efficiently (will generate 'to much' phase space), simply because less cuts are included analytically. In the second solution, we write the integrand as the sum

$$\frac{\Theta(\,\Pi(a_1,a_2)\,)\,\Theta(\,a_1 - a_1^{(0)}\,)}{a_1(a_2+a_1)\sqrt{\Pi(a_1,a_2)}} + \frac{\Theta(\,\Pi(a_1,a_2)\,)\,\Theta(\,a_2 - a_2^{(0)}\,)}{(a_1+a_2)a_2\sqrt{\Pi(a_1,a_2)}} \ ,$$

which is exactly equal to the original integrand on the phase space for which *both* cuts are included. Both integrands can be integrated analytically, so that the multi-channeling procedure can be applied to generate their sum. For this solution, the only problem is that, again, 'to much' phase space is generated.

A2.1 The $\boldsymbol{a_2}$-variable In order to integrate over the a_2-variable, $\Pi(a_1,a_2)$ is more conveniently written as

$$\Pi(a_1,a_2) = 4(\,a_2^+(a_1) - a_2\,)(\,a_2 - a_2^-(a_1)\,) \ \ , \tag{22}$$

with

$$a_2^\pm(a_1) := \frac{1}{2}\Big(1 + \frac{s_{k-1}-\sigma_k}{s_k} + \Big(1 - 2a_1 - \frac{s_{k-1}-\sigma_k}{s_k}\Big)\mathrm{c}\Big) \\ \pm \mathrm{s}\Big(a_1\Big(1 - a_1 - \frac{s_{k-1}-\sigma_k}{s_k}\Big) - \frac{\sigma_k}{s_k}\Big)^{1/2} \ \ . \tag{23}$$

Then, the general a_2-integral is given by

$$\int_{a_2^-}^{a_2^+} da_2\,\frac{1}{(a_2+h)\sqrt{(a_2^+ - a_2)(a_2 - a_2^-)}} \ \ ,$$

with $h = 0$ if no cut-off on a_2 is desirable, and h related to $a_2^{(0)}$ or $h = a_1$, depending on the solution mentioned above if a cut-off *is* desirable. By substituting $a_2 \leftarrow a_2 - h$, this integral can be written as

$$\int_{\bar{a}_2^-}^{\bar{a}_2^+} da_2\,\frac{1}{a_2\sqrt{(\bar{a}_2^+ - a_2)(a_2 - \bar{a}_2^-)}} = \frac{-2}{\sqrt{\bar{a}_2^+\bar{a}_2^-}}\left[\arctan\left(\frac{\bar{a}_2^-(\bar{a}_2^+ - a_2)}{\bar{a}_2^+(a_2 - \bar{a}_2^-)}\right)\right]_{\bar{a}_2^-}^{\bar{a}_2^+} = \frac{\pi}{\sqrt{\bar{a}_2^+\bar{a}_2^-}} \ \ ,$$

with $\bar{a}_2^\pm = a_2^\pm + h$. The explicit indefinite integral shows that the variable a_2 can be generated by inversion.

A2.2 The $\boldsymbol{a_1}$-variable We start this section by mentioning that, in the case $p_{k+1}^2 = \sigma_{k+1} \neq 0$, the variables

$$a_1^{(k)} = \frac{p_k\cdot p_{k+1}}{Q_k\cdot p_{k+1}}$$

can be expressed in terms of the 'massless' $\tilde{a}_1^{(k)}$ defined in terms of the 'long' component $p_{k+1}^{(L)}$ of p_{k+1} in the CMF$_k$: if $p_{k+1}^{(L)} = (p_{k+1}^0, \beta_k^{-1}\vec{p}_{k+1})$ and $p_{k+1}^{(S)} = (p_{k+1}^0, -\beta_k^{-1}\vec{p}_{k+1})$ with

$$\beta_k := \frac{|\vec{p}_{k+1}|}{p_{k+1}^0} = \frac{\sqrt{\lambda(s_{k+1},\sigma_{k+1},s_k)}}{s_{k+1}-\sigma_{k+1}-s_k} \ \ , \tag{24}$$

then $p_{k+1} = \frac{1+\beta_k}{2}\, p_{k+1}^{(L)} + \frac{1-\beta_k}{2}\, p_{k+1}^{(S)}$, and

$$\tilde{a}_1^{(k)} := \frac{p_k \cdot p_{k+1}^{(L)}}{Q_k \cdot p_{k+1}^{(L)}} = \beta_k^{-1} a_1^{(k)} - h_1 \quad \text{with} \quad h_1 := \frac{1-\beta_k}{2\beta_k}\left(1 + \frac{\sigma_k - s_{k-1}}{s_k}\right) \quad .$$

Since the relation between $a_1^{(k)}$ and $\tilde{a}_1^{(k)}$ is linear, the two-body phase space is still expressible in terms of $a_1^{(k)}$ scaled by β_k. The generic a_1-integral is given by

$$\int_{a_1^{\min}}^{a_1^{\max}} da_1 \, \frac{1}{(a_1 + h_1)\sqrt{(\, a_2^+(a_1) + h(a_1)\,)(\, a_2^-(a_1) + h(a_1)\,)}} \quad , \tag{25}$$

with $a_2^\pm$ as defined in (23), h is a constant (for this integral) or $h(a_1) = a_1$. The *kinematical* integration limits, coming from the requirement that $a_2^\pm(a_1)$ are real, are given by

$$a_1^\pm = \frac{1}{2}\left(1 + \frac{\sigma_k - s_{k-1}}{s_k} \pm \sqrt{\lambda\left(1, \frac{\sigma_k}{s_k}, \frac{s_{k-1}}{s_k}\right)}\,\right) \quad .$$

In the massless case we get $a_1^- = 0, a_1^+ = 1 - s_{k-1}/s_k$ and we have to impose a lower bound on a_1 given by $a_1^{(0)} = \sigma_{k+1,k}/p_{k+1} \cdot Q_k$. In all cases for the form of $h(a_1)$, the a_1-integral is of the type

$$\int_{a_1^{\min}}^{a_1^{\max}} da_1 \, \frac{1}{(a_1 + h_1) f(a_1)} \quad \text{with} \quad f(a_1) := \sqrt{a_1^2 + 2va_1 + w^2} \quad , \quad v^2 < w^2 \quad , \tag{26}$$

and the indefinite integral is given by

$$\int da_1 \, \frac{1}{(a_1 + h_1) f(a_1)} = \frac{1}{f(-h_1)} \, \log\left(\frac{a_1 + h_1 + f(a_1) - f(-h_1)}{a_1 + h_1 + f(a_1) + f(-h_1)}\right) \quad ,$$

which is analytically invertible. The definite integral (times $\pi/4$ from the a_2-integral and a factor $1/\beta_k$ if p_{k+1} is massive) gives the weight factors in the generation of a_1, a_2 for the closed antenna.

A2.3 The $\boldsymbol{a_1}$-variable in the case $\boldsymbol{m = 1}$ This refers to (15), where we have c $= -1$ and $a_2 = a_1 + \mu$ with

$$\mu := \frac{s_{k-1} - \sigma_k}{s_k} \quad , \quad \text{so that} \quad \int_{a_1^{\min}}^{a_1^{\max}} da_1 \, \frac{1}{a_1(1-a_1)(a_1+\mu)(1-\mu-a_1)} \tag{27}$$

is the integral to be performed. Since the integrand is equal to

$$\frac{1}{a_1(a_1+\mu)} + \frac{1}{a_1(1-\mu-a_1)} + \frac{1}{(a_1+\mu)(1-a_1)} + \frac{1}{(1-a_1)(1-\mu-a_1)} \quad ,$$

we see that the generation of a_1 can be done easily using the multi-channel technique with four channels with weight

$$w_i = \frac{1}{g_i^{(2)} e_i - g_i^{(1)} d_i}\left(\log\left(\frac{g_i^{(1)} a_1^{\max} + d_i}{g_i^{(2)} a_1^{\max} + e_i}\right) - \log\left(\frac{g_i^{(1)} a_1^{\min} + d_i}{g_i^{(2)} a_1^{\min} + e_i}\right)\right) \quad ,$$

where

i	d_i	e_i	$g_i^{(1)}$	$g_i^{(2)}$
1	0	μ	$+$	$+$
2	0	$1-\mu$	$+$	$-$
3	μ	1	$+$	$-$
4	1	$1-\mu$	$-$	$-$

The soft factor will simply be equal to $\sum_{i=1}^4 w_i$. The integration over the azimuthal angle, replacing the a_2-integration, gives an extra factor of 2π.

A3 Antenna split

With $q_1 \propto (1,0,01)$ and $q_2 \propto (1,0,0,-1)$, the two-body phase space integral (14) assumes the form

$$\int ds_1 ds_2\, dQ_z\, \frac{\Theta(\sqrt{s}-\sqrt{s_1}-\sqrt{s_2}\,)\,\Theta(\,s_1-s_1^{(0)}\,)\,\Theta(\,s_2-s_2^{(0)}\,)}{s_1 s_2\,(\,E_1(s_1,s_2)^2-Q_z^2\,)(\,E_2(s_1,s_2)^2-Q_z^2\,)} \quad , \tag{28}$$

with the energies

$$E_1(s_1,s_2) := \frac{s+s_1-s_2}{2\sqrt{s}} \quad , \quad E_2(s_1,s_2) := \sqrt{s}-E_1(s_1,s_2) \;\; ,$$

and where $s_{1,2}^{(0)}$ are the sums of the matrix elements σ_{ij} (9) corresponding with the momenta in two antennas to be generated. Q_z is integrated between the kinematical limits $\pm\sqrt{E_1^2-s_1^2}$, and can be treated in a way similar to the one described in the previous paragraph, by multi-channeling over four channels with

i	d_i	e_i	$g_i^{(1)}$	$g_i^{(2)}$
1	E_1	E_2	$+$	$+$
2	E_1	E_2	$+$	$-$
3	E_1	E_2	$+$	$-$
4	E_1	E_2	$-$	$-$

The final weight is the sum of the channel-weights, divided by the (dimensionful) factor $4E_1E_2$. The generation of s_1, s_2 is a specific case of a more general problem described in Appendix B2.2.

B Open antenna

The three variables needed to describe the open antenna are s_{k-1}, $a_1^{(k)}$ and $\varphi^{(k)}$. In each CMF$_k$, the $\varphi^{(k)}$-variable should be generated with uniform distribution between 0 and 2π, and the $a_1^{(k)}$-variable should be distributed following $1/a_1^{(k)}$ between

$$a_{1,+}^{(k)} = \frac{s_k+\sigma_k-s_{k-1}+\beta_k\sqrt{\lambda(s,\sigma_k,s_{k-1})}}{2s_k}$$
$$a_{1,-}^{(k)} = \max\left[\frac{s_k+\sigma_k-s_{k-1}-\beta_k\sqrt{\lambda(s,\sigma_k,s_{k-1})}}{2s_k}\,,\,\frac{\sigma_{k+1,k}}{p_{k+1}\cdot Q_k}\right] \quad ,$$

with β_k as defined in (24). The normalized density for these generations is equal to

$$\frac{2\beta_k}{\pi L_k\, a_1^{(k)}} = \frac{\sqrt{\lambda(s_{k+1},\sigma_{k+1},s_k)}}{\pi L_k\,(p_{k+1}\cdot p_k)} \quad \text{with} \quad L_k = \log(a_{1,+}^{(k)}/a_{1,-}^{(k)}) \ .$$

This suggests to use

$$\begin{aligned} g_{k+1}(s_k) &:= \frac{\sqrt{\lambda(s_{k+1},\sigma_{k+1},\Sigma_k)}}{(s_k-\Sigma_k)\sqrt{\lambda(s_{k+1},\sigma_{k+1},s_k)}} \\ &= \frac{d}{ds_k}\log\left(-\frac{s_k-\Sigma_k+\sqrt{\lambda(s_{k+1},\sigma_{k+1},s_k)}-\sqrt{\lambda(s_{k+1},\sigma_{k+1},\Sigma_k)}}{s_k-\Sigma_k+\sqrt{\lambda(s_{k+1},\sigma_{k+1},s_k)}+\sqrt{\lambda(s_{k+1},\sigma_{k+1},\Sigma_k)}}\right) \end{aligned}$$

instead of (10) for the case that $\beta_k \neq 1$. The logarithm contributes to the weight as a soft factor again. For small values of the squared masses σ_k, the factor in the numerator will be canceled by $s_{k+1}-\Sigma_{k+1}$ in the denominator of $g_{k+2}(s_{k+1})$. Just as in the case of the closed antenna, we end up with one remaining, and desirable, factor $s_2-\Sigma_2 = 2(p_2\cdot p_1)$ in the denominator of the open antenna density, which cannot be achieved by the generation of the $a_1^{(k)}$-variables.

Let us denote the soft factor coming from the s_k-generation by

$$G_{k+1} := \int_{\Lambda_k}^{s_{k+1}-\Delta_k} ds_k\, g_{k+1}(s_k) \ ,$$

then the complete density resulting from the decomposition (16) is

$$\frac{dD_0(Q_n;p_n,\dots,p_1)}{d\Phi_n(Q_n;p_n,\dots,p_1)} = \frac{(q_1\cdot Q_n)\sqrt{\lambda(s_n,\sigma_n,\Sigma_{n-1})}\;B_n}{(q_1\cdot p_n)(p_n\cdot p_{n-1})\cdots(p_3\cdot p_2)(p_2\cdot p_1)} \quad ,$$

with soft factor

$$B_n := \frac{1}{(4\pi)^{n-1}G_3L_3L_2}\prod_{k=n}^{4}\frac{1}{G_kL_k}\sqrt{1-\frac{4\sigma_{k-1}\Sigma_{k-2}}{(s_{k-1}-\Sigma_{k-1})^2}} \ .$$

The factor $p_1\cdot q_2$ is missing in the denominator of the density, which is the reason why we call this an *open* antenna. Other open antennas can be obtained by starting with a decomposition of Q_n into two momenta, from each of which open antennas of the above type are generated, with initial momentum q_1 for the one, and q_2 for the other.

In order to digress about this procedure, let us extend the labeling a bit. With a set $\{I_1,I_2,\dots,I_n\}$ of n non-equal labels, we write

$$Q_{I_k} := \sum_{m=1}^{k} p_{I_m} \quad , \quad s_{I_k} := Q_{I_k}^2$$

and so on. Now take $n = M+N$ and let

$$\{I_1,\dots,I_N\} := \{M+1,M+2,\dots,M+N\} \quad , \quad \{J_1,\dots,J_M\} := \{M,M-1,\dots,2,1\} \ .$$

We introduce $p_{I_{N+1}} = q_1$, $p_{J_{M+1}} = q_2$, and the decomposition

$$\begin{aligned}&ds_{J_M} ds_{I_N}\, d\Phi_2(Q_n; s_{I_N}, s_{J_M}; Q_{I_N}, Q_{J_M})\\ &\times dD_0(Q_{I_N}; p_{I_N}, p_{I_{N-1}}, \ldots, p_{I_1})\, dD_0(Q_{J_M}; p_{J_M}, p_{J_{M-1}}, \ldots, p_{J_1}) \;\;,\end{aligned}$$

which produces the density

$$\begin{aligned}\frac{dD_M(Q_n; p_n, \ldots, p_1)}{d\Phi_n(Q_n; p_n, \ldots, p_1)} &= \frac{2Q_n^2\, B_{J_M} B_{I_N}}{\pi\sqrt{\lambda(Q_n^2, s_{J_M}, s_{I_N})}} \times \frac{(p_{M+1}\cdot p_M)}{(q_1\cdot p_n)(p_n\cdot p_{n-1})\cdots(p_2\cdot p_1)(p_1\cdot q_2)}\\ &\times (q_1\cdot Q_{I_N})(q_2\cdot Q_{J_M})\sqrt{\lambda(s_{I_N}, \sigma_{I_N}, \Sigma_{I_{N-1}})\lambda(s_{J_M}, \sigma_{J_M}, \Sigma_{J_{M-1}})}\;. \quad (29)\end{aligned}$$

In order to cancel the 'undesirable' factors on the second line, we need to take care of the generation of $s_{I_N}, s_{J_M}, Q_{I_N}, Q_{J_M}$.

B1 Generation of Q_{I_N}, Q_{J_M}

Since $q_1 \propto (1,0,0,1)$ and $q_2 \propto (1,0,0,-1)$, we can write

$$a_1 = \frac{q_1\cdot Q_{I_N}}{q_1\cdot Q_n} \quad \text{and} \quad \frac{Q_{J_M}\cdot q_2}{Q_n\cdot q_2} = a_1 + \mu \quad \text{with} \quad \mu = \frac{s_{J_M} - s_{I_N}}{s_n} \;\;,$$

so that the generation of an azimuthal angle φ between 0 and 2π with the uniform distribution, and the generation of a_1 with a density proportional to

$$\frac{1}{a_1(a_1+\mu)} \quad \text{between} \quad a_1^{\pm} = \frac{s_n + s_{I_N} - s_{J_M} \pm \sqrt{\lambda(s_n, s_{I_N}, s_{J_M})}}{2s_n}$$

leads to the total density

$$d\Phi_2(Q_n; s_{I_N}, s_{J_M}; Q_{I_N}, Q_{J_M})\,\frac{2\mu}{\pi\log\frac{a_1^+(a_1^-+\mu)}{a_1^-(a_1^++\mu)}}\,\frac{(q_1\cdot Q_n)(Q_n\cdot q_2)}{(q_1\cdot Q_{I_N})(Q_{J_M}\cdot q_2)} \;\;.$$

B2 Generation of s_{I_N}, s_{J_M}

B2.1 If $M = 1$, then $s_{J_M} = \sigma_{J_M}$, and we only need to generate s_{I_N} with a density proportional to

$$\frac{1}{\sqrt{\lambda(s_{I_N}, \sigma_{I_N}, \Sigma_{I_{N-1}})}} = \frac{d}{ds_{I_N}}\log\left(s_{I_N} - \sigma_{I_N} - \Sigma_{I_{N-1}} + \sqrt{\lambda(s_{I_N}, \sigma_{I_N}, \Sigma_{I_{N-1}})}\right) \;\;.$$

This density cancels the corresponding factor in the total antenna density. Something similar can be done if $N = 1$.

B2.2 If $N > 1$ and $M > 1$, then both s_{I_N} and s_{J_M} have to be generated, in the region where $\sqrt{s_n} - \sqrt{s_{I_N}} - \sqrt{s_{J_M}} > 0$. This is far more complicated than the previous case, and we restrict ourselves to a density with a denominator proportional to $(s_{I_N} - \varSigma_{I_N})(s_{J_M} - \varSigma_{J_M})$. Because $\sigma_{I_N}\varSigma_{I_{N-1}}$ and $\sigma_{J_M}\varSigma_{J_{M-1}}$ may be considered small, it still cancels the factor

$$\sqrt{\lambda(s_{I_N}, \sigma_{I_N}, \varSigma_{I_{N-1}})\lambda(s_{J_M}, \sigma_{J_M}, \varSigma_{J_{M-1}})}$$

in (29). We shall write s_1, s_2 instead of s_{I_N}, s_{J_M} from now on, and denote $m := \sqrt{s_n}$,

$$m_1^2 := \varSigma_{I_N} \quad , \quad c_1^2 := \sum_{K,L=1}^{N} \sigma_{I_K,I_L} \quad \text{and} \quad m_2^2 := \varSigma_{J_M} \quad , \quad c_2^2 := \sum_{K,L=1}^{M} \sigma_{J_K,J_L} \quad .$$

We choose first to generate s_1, and then s_2, so that the integral, corresponding with the generation, becomes

$$\int_{c_1^2}^{(m-m_2)^2} \frac{ds_1}{s_1 - m_1^2} \int_{c_2^2}^{(m-\sqrt{s_1})^2} \frac{ds_2}{s_2 - m_2^2} \quad .$$

The s_2-integral is simple, and shows that s_2 can easily be obtained by inversion. After integration over s_2, the s_1-integral becomes

$$\int_{c_1^2}^{(m-m_2)^2} \frac{ds_1}{s_1 - m_1^2} \left[\log\left((m - \sqrt{s_1})^2 - m_2^2\right) - \log\left(c_2^2 - m_2^2\right)\right] \quad .$$

The s_1-variable distributed following this integrand can be obtained with high efficiency by rejection from the density proportional to $1/(s_1 - m_1^2)$. The total integral can be calculated, and is given by

$$\sum_{\rho,\epsilon=\pm1} \left[\mathrm{Li}\left(\frac{x_1 + m_1}{m + \rho m_2 + \epsilon m_1}\right) + \log\left(\frac{m + \rho m_2 - x_1}{m}\right) \log\left(\frac{x_1 + m_1}{m + \rho m_2 + \epsilon m_1}\right)\right]_{x_1=c_1}^{m-m_2}$$
$$- \log\left(\frac{c_2^2 - m_2^2}{m^2}\right) \log\left(\frac{(m - m_2)^2 - m_1^2}{c_1^2 - m_1^2}\right) \quad ,$$

where

$$\mathrm{Li}(x) := \sum_{n=1}^{\infty} \frac{(1 - x)^n}{n^2} \quad .$$

C SARGE

In this Appendix, we give a short review about `SARGE` and present the adaptations applied in `SARGE.n`. We need to start with the establishment of some notation.

$\mathcal{H}_p$ is a Lorentz transformation that boosts momentum p to $(\sqrt{p^2}, 0, 0, 0)$. $\mathcal{R}_p$ is a Lorentz transformation that rotates momentum p to $(p^0, 0, 0, |\vec{p}|)$. The standard representation of a unit vector in terms of parameters $z \in [-1, 1]$ and $\varphi \in [0, 2\pi]$ is denoted by

$$\hat{n}(z, \varphi) := (\sqrt{1-z^2}\,\sin\varphi, \sqrt{1-z^2}\,\cos\varphi, z) \quad .$$

$\mathbf{P}_n \subset [-1, 1]^n$ is the n-dimensional polytope, which consists of the support of the indicator function

$$\vartheta_{\mathbf{P}_n}(x_1, x_2, \dots, x_n) := \prod_{i=1}^{n} \Theta(1 - |x_i|) \prod_{j,k=1}^{n} \Theta(1 - |x_j - x_k|) \quad .$$

The algorithm for the generation of n final-state momenta with a single antenna structure without initial-state momenta as given in [7] is

Algorithm C.1

1. generate two massless momenta q_1, q_n, back-to-back;

2. generate $(x_1, \dots, x_{2n-4}) \in \mathbf{P}_{2n-4}$ and $(\varphi_2, \dots, \varphi_{n-1}) \in [0, 2\pi]^{n-2}$, all uniformly distributed;

3. for $i = 2, \dots, n-2$ construct q_i following (with $x_0 := 0$)
$$\begin{gathered} b_1 \leftarrow q_{i-1} + q_n \quad , \quad b_2 \leftarrow \mathcal{H}_{b_1} q_{i-1} \quad , \\ \xi_1 \leftarrow e^{(x_{2i-3} - x_{2i-4}) \log \xi_{\mathrm{m}}} \quad , \quad \xi_2 \leftarrow e^{(x_{2i-2} - x_{2i-4}) \log \xi_{\mathrm{m}}} \quad , \\ v^0 \leftarrow \sqrt{\frac{(q_{i-1} \cdot q_n)}{2}}\,(\xi_2 + \xi_1) \quad , \quad z \leftarrow \frac{\xi_2 - \xi_1}{\xi_2 + \xi_1} \quad , \quad \vec{v} \leftarrow q^0 \mathcal{R}_{b_2}^{-1} \hat{n}(z, \varphi_i) \quad , \\ q_i \leftarrow \mathcal{H}_{b_1}^{-1} v \quad ; \end{gathered}$$

4. put $p_i \leftarrow u \mathcal{H}_Q q_i$ for $i = 1, \dots, n$, where $Q = \sum_{i=1}^{n} q_i$ and $u = \sqrt{s/Q^2}$.

The density with which the generated momenta are distributed is then given by

$$d\Phi_n(P; p_1, \dots, p_n)\, \frac{s^2}{2\pi^{n-2}}\, g_n(p_1, p_2, \dots, p_n) A_n(p_1, p_2, \dots, p_n) \quad , \tag{30}$$

with $P = (\sqrt{s}, 0, 0, 0)$, where $A_n(p_1, p_2, \dots, p_n)$ is defined in (2), and where

$$g_n(p_1, p_2, \dots, p_n) := \frac{\vartheta_{\mathbf{P}_{2n-4}}(x_1, x_2, \dots, x_{2n-4})}{(2n-3)(\log \xi_{\mathrm{m}})^{2n-4}} \quad ,$$

with

$$x_{2i-3} = \frac{1}{\log \xi_{\mathrm{m}}} \log \frac{(p_{i-1} \cdot p_i)}{(p_1 \cdot p_n)} \quad \text{and} \quad x_{2i-4} = \frac{1}{\log \xi_{\mathrm{m}}} \log \frac{(p_i \cdot p_n)}{(p_1 \cdot p_n)} \quad , \quad i = 2, \dots, n-2 \quad .$$

The density obtained before step 4 of Algorithm C.1 is invariant under simultaneous scaling of all momenta, but the sum of the momenta is not at rest. In order to achieve this, the scaling symmetry has to be broken. In order to include the initial-state momenta in the density, more symmetries have to be broken: the symmetry under cyclic permutations of the momenta, a part of the simultaneous rotation symmetry of the momenta, and finally the symmetry under the reflection permutation $(1, 2, \dots, n) \mapsto (n, n-1, \dots, 1)$.

Let us denote $\lceil k \rceil = k \bmod n$. The cyclic symmetry is broken through

Algorithm C.2

1. choose a $k \in \{0, 2, \dots, n-1\}$ with relative probability $(p_{\lceil k \rceil} \cdot p_{k+1})$;
2. put $\{p_1, p_2, \dots, p_n\} \leftarrow \{p_{\lceil 1+k \rceil}, p_{\lceil 2+k \rceil}, \dots, p_{\lceil n+k \rceil}\}$.

As a result, the function A_n in the density (30) is replaced by

$$\frac{n}{(p_n \cdot p_1) + (p_1 \cdot p_2) + \cdots + (p_{n-1} \cdot p_n)} \times (p_n \cdot p_1) A_n(p_1, \dots, p_n) \ \ .$$

The factor $(p_n \cdot p_1)$ in the denominator of A_n is replaced by the average of all factors. Next, part of the rotation symmetry and the symmetry under the reflection permutation are broken through

Algorithm C.3

1. generate $\varphi \in [0, 2\pi]$ uniformly distributed, and $z \in [-1, 1-c]$ following $1/(1-z)$;
2. rotate all momenta such that $\vec{p}_1$ lies along $\hat{n}(z, \varphi)$.

If we denote $q_1 := \frac{1}{2}\sqrt{s}(1, 0, 0, 1)$, this leads to an extra factor

$$\frac{1}{2\pi \log(2/c)} \times \frac{\frac{1}{2}\sqrt{s}\, p_1^0}{(q_1 \cdot p_1)}$$

in the density. The cut-off c can be taken equal to $s_0/(\sqrt{s}\, p_1^0)$, where s_0 should be a cut-off on the invariant mass $(q_1 + p_1)^2$. Notice that $p_1^0 = (P \cdot p_1)/\sqrt{s}$, so that the new factor in the density becomes

$$\frac{1}{4\pi \log(\, 2(P \cdot p_1)/s_0\,)} \times \frac{(P \cdot p_1)}{(q_1 \cdot p_1)} \quad .$$

In order to include a factor proportional to $(p_n \cdot q_2)$ in the denominator of the density, where $q_2 := \frac{1}{2}\sqrt{s}(1, 0, 0, -1)$, Algorithm C.3 can be preceded by

Algorithm C.4

1. choose with equal probabilities whether to rotate all momenta such that $\vec{p}_1$ lies along $\hat{n}(z, \varphi)$, or that $\vec{p}_n$ lies along $-\hat{n}(z, \varphi)$.

The total density (30) becomes then such that A_n is replaced by

$$\frac{n}{4\pi} \times \frac{\frac{(P\cdot p_1)(p_n\cdot q_2)}{\log(\,2(P\cdot p_1)/s_0\,)} + \frac{(q_1\cdot p_1)(P\cdot p_n)}{\log(\,2(P\cdot p_n)/s_0\,)}}{(p_n\cdot p_1) + (p_1\cdot p_2) + \cdots + (p_{n-1}\cdot p_n)} \times A_{n+2}(q_1,\ p_1, p_2, \dots, p_{n-1}, p_n,\ q_2) \ .$$

In the end, we want to obtain all permutations in the momenta $\{p_1, p_2, \dots, p_n, q_2\}$ of the above density. Some of them can be obtained by relabeling, and for the others, which change the position of q_2, we can do the following. If the cyclic permutation in Algorithm C.2 is chosen with relative probability $(p_{\lceil k\rceil}\cdot p_{k+1})(p_{k+i}\cdot p_{k+1+i})$, then the factor $(p_n\cdot p_1)(p_i\cdot p_{i+1})$ in the denominator of A_n is replaced by the average over its cyclic permutations. Then we can choose with equal relative probabilities whether to rotate all final-state momenta such that $\vec{p}_1$ lies along $\hat{n}$, $\vec{p}_i$ lies along $-\hat{n}$, $\vec{p}_{i+1}$ lies along $-\hat{n}$, or $\vec{p}_n$ lies along $\hat{n}$. This will lead to density that is proportional to

$$A_{n+2}(q_1,\ p_1, p_2, \dots, p_{i-1}, p_i,\ q_2,\ p_{i+1}, p_{i+2}, \dots, p_{n-1}, p_n) \ .$$

D Permutations

For the analyses of the the multi-channel procedure over the antenna densities for the different permutations of the momenta, we need an enumeration of the permutations. In other words, we need a mapping

$$\{1, 2, \dots, n!\} \mapsto Sym_n \ .$$

During a computation, one could, of course, just store an enumeration in the memory of the computer, but this costs an amount of memory of $\mathcal{O}(n!)$. Every algorithm that delivers all permutations supplies a mapping *a priori*, but this algorithm has an *a priori* computational complexity of $\mathcal{O}(n!)$.

We propose the following algorithm as a solution. Any number $l \in \{1, 2, \dots, n!\}$ can uniquely be written in the basis of the lower factorials: $l = 1 + l_2 + l_3 2! + l_4 3! + \cdots + l_n (n-1)!$, where $l_k \in \{0, 1, \dots, k-1\}$. Let $\gamma_i^{(k)}$ denote the i-fold cyclic permutation of the first k elements of $(1, 2, \dots, n)$, for example $\gamma_2^{(3)}(1, 2, 3, 4) = (2, 3, 1, 4)$. Then any permutation of $(1, 2, \dots, n)$ can be written as $\gamma_{l_2}^{(2)}\gamma_{l_3}^{(3)}\cdots\gamma_{l_n}^{(n)}(1, 2, \dots, n)$. This leads to a mapping of the required kind, which has a complexity of $\mathcal{O}(n^2)$. In a loop delivering all permutations, it can easily be reduced to complexity $\mathcal{O}(n)$ at the cost of an amount of memory of $\mathcal{O}(n)$.