IPPP/08/61
DCPT/08/122
SLAC-PUB-13379
MCNET/08/07
September 2008

How to calculate colourful cross sections efficiently

Tanju Gleisberg[1], Stefan Höche[2], Frank Krauss[2], Radosław Matyszkiewicz[3]

[1] Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309, USA
[2] Institute for Particle Physics Phenomenology, Durham University, Durham DH1 3LE, UK
[3] Institut für theoretische Physik, Technische Universität Dresden, 01062 Dresden, Germany

Abstract: Different methods for the calculation of cross sections with many QCD particles are compared. To this end, CSW vertex rules, Berends-Giele recursion and Feynman-diagram based techniques are implemented as well as various methods for the treatment of colours and phase space integration. We find that typically there is only a small window of jet multiplicities, where the CSW technique has efficiencies comparable or better than both of the other two methods.

1 Introduction

In the past years, a variety of string-inspired methods has been proposed for the efficient calculation of QCD scattering amplitudes with large number of external legs [1, 2, 3]. Compared with techniques based on the construction of Feynman diagrams and corresponding helicity amplitudes [4], these new methods induce a relatively mild growth in computational complexity with increasing multiplicity. A particularly interesting new technique are the CSW vertex rules, which are based on the correspondence between a weakly coupled $\mathcal{N} = 4$ Super Yang-Mills theory and a certain type of string theory. It has been shown in [1] that this method allows to build arbitrary tree-level colour-ordered amplitudes from MHV amplitudes. In the context of this publication, the CSW rules have been implemented into the SHERPA framework [5]. The corresponding code has been validated by a comprehensive comparison of cross sections with other programs for the processes $pp \to$ jets and $pp \to Z+$jets. For Monte Carlo phase space integration, the standard techniques presented in [6] in the implementation of AMEGIC++ [7] are employed. In pure QCD processes the HAAG generator [8] is used, which has been modified such that it can be employed with adaptive techniques like VEGAS [9].

The aim of this publication is to address the issue of efficiency of the CSW technique when dealing with full cross sections, including summation over colours and helicities, rather than single amplitudes. This extends previous studies presented in [10, 11]. For purely gluonic processes, it has already been pointed out in [11] that Berends-Giele type recursive relations are superior at large final state multiplicities. This is mainly due to the fact that the colour decomposition (colour ordering) of QCD amplitudes implies a large exponential growth in the number of required partial amplitudes, which can, in the case of the Berends-Giele approach, be overcome with the technique of colour dressing. It was however also noted there that colour-ordered multi-gluon amplitudes for relatively low final state multiplicities (≤ 5) are most efficiently computed using CSW vertex

Work supported in part by US Department of Energy contract DE-AC02-76SF00515

rules. This is because in this regime the number of possible colour configurations is still low. In addition, there is also only a small number of contributing non-MHV amplitudes, such that the major part of the calculation involves one MHV vertex only, maximising the impact of the compact formulae in the CSW approach. On the other hand, for low multiplicities expressions obtained from traditional Feynman-diagram based techniques are also compact and can easily be simplified. The natural question arises, whether those methods – Berends-Giele recursion and traditional Feynman-diagram based techniques – eventually perform comparably or even better than the CSW vertex rules.

In this publication we aim at quantifying the effects outlined above for cross sections with realistic cuts and for some experimentally significant processes. To this end, we compare three different numerical programs:

1. AMEGIC++ [7], which computes cross sections using helicity amplitudes [4], constructed through Feynman diagrams;
2. the new generator COMIX [12] based on colour-dressed Berends-Giele (CDBG) recursion [11];
3. and a new, optimised implementation of CSW rules in the framework of SHERPA.

Each of the methods prefers a certain type of phase space integrator. Colour-ordered amplitudes are employed in the new implementation of the CSW technique and can be constructed in COMIX, such that both codes can employ the colour sampling integrator presented in [12]. Since both AMEGIC++ and COMIX have access to internal propagator lines the techniques of [6] can naturally be employed, since they are based on the assumed pole structure of single diagrams. Using AMEGIC++ to extract corresponding integration channels from the Feynman diagrams, this method can also be ported to be used with the CSW rules.

We attempt to compare as many generator-integrator pairings as possible for a number of processes that will be relevant at the LHC. The outline is as follows. In Sec. 2 the algorithms needed for the implementation of the CSW technique are briefly summarised. The two other matrix element generators have been presented in [7,12] and we refer to the original publications for further details. We do, however, comment on some relevant details of the colour treatment and the phase space integration, which have not been published before. In Sec. 3 the programs are validated and their respective efficiency in terms of evaluation time per configuration and in terms of integration time to reach a certain precision level is discussed.

2 Implementation

In this section, we briefly introduce the basic ingredients for the numerical implementation of the CSW rules. Phase space integration algorithms are presented in detail in [7,12], therefore only the modified HAAG algorithm will be discussed here, cf. App. 2.3.

2.1 Colour factors

In our implementation we employ the CSW vertex rules for colour-ordered amplitudes; therefore the colour structure and the kinematical part of the amplitudes factorise. For QCD amplitudes containing quarks the well-known decomposition over fundamental representation matrices $T^a_{i\bar{j}}$ of $SU(3)$ [13] is used. Considering for example one quark line connecting the external quarks 1 and n with colours i_1 and $\bar{\jmath}_n$ with the intermediate gluons carrying colours a_i, this decomposition reads

$$\mathcal{A}(1,\ldots,n) = \sum_{\sigma\in S_{n-2}} \mathrm{Tr}\left(T^{a_{\sigma_2}}\ldots T^{a_{\sigma_{n-1}}}\right)_{i_1\bar{\jmath}_n} A(1,\sigma_2,\ldots,\sigma_{n-1},n)\;. \tag{1}$$

The sum runs over all $(n-2)!$ permutations of $(2,\ldots,n-1)$. For purely gluonic amplitudes the numerically more efficient representation through matrices $(F^a)_{bc} = if^{abc}$, i.e. the adjoint representation, [14] is employed. Assuming n external gluons carrying colours $a_1 \ldots a_n$ this decomposition reads

$$\mathcal{A}(1,\ldots,n) = \sum_{\sigma\in S_{n-2}} (F^{a_{\sigma_2}} F^{a_{\sigma_3}} \ldots F^{a_{\sigma_{n-1}}})_{a_1 a_n}\, A(1,\sigma_2,\ldots,\sigma_{n-1},n)\,, \tag{2}$$

where again the sum runs over all $(n-2)!$ permutations of $(2,\ldots,n-1)$.

2.2 Amplitude evaluation

The first non-vanishing helicity configurations for tree amplitudes are the MHV or Parke-Taylor amplitudes [15]. They contain $n-2$ partons with like-sign helicity and 2 with opposite sign helicity. Amplitudes with all or all but one like-sign helicity vanish at tree-level, cf. [13]. The MHV ($\overline{\text{MHV}}$) amplitudes can be written as simple holomorphic (anti-holomorphic) functions. In the notation of [16] the n-gluon MHV-amplitudes read

$$A_n\left(1^+,...,k^-,...,l^-,...,n^+\right) = i\,\frac{\langle k\,l\rangle^4}{\langle 1\,2\rangle\langle 2\,3\rangle\ldots\langle n{-}1\,n\rangle\langle n\,1\rangle}\,. \tag{3}$$

MHV amplitudes for one external quark pair are easily obtained using supersymmetric Ward identities [17]. They are given by

$$\begin{aligned} A_n\left(q^+,2^+,\ldots,k^-,\ldots,n{-}1^+,\bar{q}^-\right) &= i\,\frac{\langle k\,q\rangle\langle k\,\bar{q}\rangle^3}{\langle 1\,2\rangle\ldots\langle n{-}1\,n\rangle\langle n\,1\rangle} \\ A_n\left(q^-,2^+,\ldots,k^-,\ldots,n{-}1^+,\bar{q}^+\right) &= -\,i\,\frac{\langle k\,q\rangle^3\langle k\,\bar{q}\rangle}{\langle 1\,2\rangle\ldots\langle n{-}1\,n\rangle\langle n\,1\rangle} \end{aligned} \tag{4}$$

Corresponding $\overline{\text{MHV}}$ amplitudes are obtained by complex conjugation, amounting to the replacement $\langle\ \rangle \to [\ ]$. Amplitudes for two external quark pairs and for the scattering into a Drell-Yan lepton pair have been discussed in [18].
The CSW vertex rules to build full amplitudes from MHV ($\overline{\text{MHV}}$) amplitudes read [1]:

- For an amplitude $A_n\left(1^{h_1},\ldots,n^{h_n}\right)$ with $2 < m < n-2$ negative helicity external legs, draw all possible diagrams connecting $m-1$ MHV ($\overline{\text{MHV}}$) amplitudes. Their free legs connect to the external particles.
- Construct spinors for internal lines from the corresponding momenta p via $\lambda_a = p_{a\dot{b}}\eta^{\dot{b}}$, where η is an arbitrary but fixed reference spinor employed in all CSW diagrams.
- Associate a scalar propagator $1/p^2$ with each internal line connecting two MHV vertices.

Using this algorithm, amplitudes with an arbitrary number of negative (positive) helicity partons can be computed.

2.3 Phase space integration using HAAG

Several approaches have been presented in the past to efficiently sample multi-particle phase spaces [6,19,8]. For multi-jet phase space integration in pure QCD processes, the most advanced algorithm so far is the HAAG generator presented in [8]. It is designed to produce momenta approximately following a QCD antenna function, cf. Sec. A

Symmetrisation of antennae

In analogy to the antenna function in Eq. (5), each single HAAG channel can be labeled by a specific permutation of the momenta. Since the algorithm always starts from incoming momenta, the channels are invariant with respect to cyclic permutations. However, unlike the underlying antenna function itself, different HAAG channels are obtained, if the order of the momenta is reversed. This is due to the fact that the open antenna algorithm is employed. In order to recover symmetry, pairs of channels given by a permutation and its reverse are combined into one, i.e. one of the two configurations is chosen with equal probability and the weight is given by the average of the two. Furthermore, all antennae can be classified into different types depending on the relative position of the incoming momenta (p_0 and p_1) within a permutation of the momenta. All channels of the same type, i.e. with the same number of final state momenta between p_0 and p_1, are in principle equivalent. They can be obtained from each other by simply relabeling the final state momenta.

Improvements of the algorithm

To generate an adequate phase space integrator for realistic n-particle QCD processes, different HAAG channels can be combined using the multichannel method [20], cf. [8].

The efficiency of the integrator is improved, if additionally the VEGAS algorithm [9] is applied to optimise individual single channels VEGAS is very efficient in adapting to functions, whose peaking behaviour is not too extreme and which factorise into a product of one-dimensional functions. Although this is not necessarily the case for a fully differential cross section, VEGAS can be used to better adapt the antenna-like structures in single HAAG-channels to the corresponding substructures in the matrix elements, including phase space cuts.

The equivalence of HAAG-channels of the same type can also be used such that all of them employ the same VEGAS map. This alleviates the adaptation significantly, since only very few maps survive, with a number that grows only linearly with the number of particles. This can easily be understood from the construction of the HAAG channels from a factorially growing number of equivalent mappings.

3 Results

Various approaches can be used to judge the efficiency of methods to evaluate amplitudes, to sum them over colours and helicities and to integrate them over phase space. Finally, however it is important that the numerical code which implements the method yields a cross section with the desired error as quickly as possible. Theoretically appealing forms of the amplitude are not guaranteed to be of maximal help in this respect, see for example [11].

Therefore, the following strategy to judge the various methods is adopted:

1. The evaluation times for helicity summed amplitudes are compared. Two different sets are generated, which correspond to a colour-summed and a colour-sampled integration, respectively. In this publication, helicity sampling is not considered, since it introduces additional degrees of freedom which in most cases significantly slow down the integration. For low multiplicities, this effect is not compensated by the correspondingly lower number of amplitudes that have to be evaluated, cf. [12].

2. A number of cuts is imposed on the final state to yield physical cross sections for various processes. In each case, the integration is terminated once a certain precision of the result has

been reached. Note that this effectively tests not only the method for evaluating amplitudes, but also the phase space sampling and optimisation. This study corresponds exactly to the problem outlined above and can therefore provide vital information about a preference for which technique/code to use for the evaluation of cross sections in different regimes of particle multiplicity.

In the following, results obtained with the new implementation of the CSW technique are labeled "CSW", the helicity methods implemented in AMEGIC++ are labeled by "AMEGIC" and the colour-dressed Berends-Giele recursion implemented in COMIX is denoted by "COMIX".

3.1 Amplitude evaluation times

Colour-summed amplitudes are compared in Tab. 1. The colour-dressed Berends-Giele recursion is not included in this comparison, since it is optimised to generate minimal subsets of colour-interfering amplitudes; it would thus be very inefficient for colour-summation. The results compiled here clearly show the superiority of the CSW method over the evaluation of Feynman diagrams for pure QCD process with no or only one quark-line. In these two cases the expressions obtained from CSW rules are very compact, leading to an increasingly better behaviour for larger numbers of final state particles. Additionally, the colour-decomposition in the adjoint representation [14] used for the CSW amplitudes seems to significantly simplify the computation for purely gluonic amplitudes. For two quark lines the two methods are comparable in processes with lower multiplicities. For higher multiplicities the standard helicity method, however, quickly leads to unmanageable expressions. For processes involving electroweak interactions there is no obvious advantage in using CSW rules.

To compare evaluation times for colour-ordered amplitudes, matrix elements generated by traditional methods in AMEGIC++ are not included, since the projection of the colour structures coming from the direct evaluation of full QCD Feynman rules onto colour-ordered amplitudes would not lead to a significant simplification compared to the full colour-summed expression. Therefore the results for colour-ordered amplitudes shown in Tab. 2 contain only CSW and Berends-Giele recursion techniques. Obviously, the former is preferred over the latter for multiplicities that require the evaluation of MHV contributions only, i.e. processes with at most five particles involved. Once next-to-MHV contributions kick in (six and seven particles involved) both methods exhibit a similar performance, beyond that the Berends-Giele technique is clearly the method of choice. This corresponds to what has been found in [11] for helicity-sampled matrix elements. It should be noted, however, that when dressing amplitudes with colour and calculating corresponding matrix elements which include subleading colour contributions the colour-dressed Berends-Giele method gains additional performance compared to CSW rules, cf. [11].

3.2 Integration times

To investigate the integration behaviour of the various codes, the time needed to compute cross sections is compared for example processes at the LHC with a centre-of-mass energy of 14 TeV. The setup is essentially identical to the one employed in [21]. Respective settings are listed in Tab. 3.

A variety of phase space generators is used for the integration, which all roughly reproduce the peaking behaviour of the matrix elements and which are to some extend adaptive. For colour-summed matrix elements we use the improved version of the HAAG generator ("HAAG") or an integrator using the standard multi-channel integration technique with channels constructed according to [6], (labeled "MC"). Their usage with colour-sampled matrix elements, however, turns out to be quite inefficient since they do not take into account that different colour assignments give rise to enhanced contributions in different phase space regions. For colour-sampled multi-gluon

scattering a special integrator has been constructed which is denoted by "CSI" (Colour Sampling Integrator). It is based on the channels given by the HAAG algorithm, which are selected and weighted specific to a corresponding colour assignment. For matrix elements generated by COMIX, a recursive phase space generator (denoted "RPG") is used which adapts the recursive structure of the Berends-Giele formalism and is thus applicable for very high particle multiplicities. The latter two integrators are described in detail in [12].

The first class of processes is pure jet production with up to 6 jets in the final state, for different maximal numbers of quarks. Tab. 4 lists the results for gluon scattering processes. For up to four final state particles colour-summed CSW matrix elements combined with HAAG are most efficient. Beyond that the number of colour configurations becomes large enough to render the computation of colour-summed matrix elements a cumbersome, time-consuming exercise. In this region CDBG matrix elements paired with the CSI integrator give the best performance. In Tab. 5 multi-jet cross sections with one and two quark lines are studied, the conclusions are essentially the same as in the purely gluonic case.

The second class of processes is given by the production of a lepton pair accompanied by additional jets, with results shown in Tab. 6. In this case, only the "MC" and "RPG" integrators have been employed. Here, the best performance is achieved with colour-summed matrix elements for up to three jets. As stated before the CSW rules do not lead to any improvement.

4 Conclusions

In this publication a comparison between different matrix element generators, paired with different integration techniques has been presented. All of them are at present implemented in the framework of the event generator SHERPA.

Considering the evaluation of matrix elements, it has been shown that for low multiplicities traditional techniques to evaluate Feynman diagrams in the helicity formalism perform surprisingly well. However, with growing numbers of external legs, these methods quickly lead to unmanageably large expressions. The compact formulae for MHV amplitudes result in a significant gain with the CSW technique, once QCD amplitudes predominantly containing gluons are concerned. Otherwise, the method performs comparably or sometimes even worse than techniques based on Feynman diagrams. With growing numbers of external legs, the colour dressed Berends-Giele recursion is the candidate best-suited to quickly evaluate matrix elements.

Considering the various phase space integrators, it has been shown that traditional methods based on the expected pole structure of the integrand as guessed from Feynman diagrams or similar are not only versatile enough to yield an appreciable convergence of results for QCD processes. There, dedicated algorithms like HAAG and CSI have better integration behaviour, but they are limited in their applicability.

To summarise, a wide range of matrix element evaluation and integration techniques has been made available within the SHERPA framework. The different techniques have been tested and compared, such that for every given jet multiplicity an optimal performance of the overall package can be achieved.

Acknowledgements

We thank Phil Roffe, Graeme Stewart and the ScotGrid [22] Tier 2 sites Durham and Glasgow for technical support. TG's research was supported by the US Department of Energy, contract DE-AC02-76SF00515. SH thanks the HEPTOOLS Marie Curie Research Training Network (contract

number MRTN-CT-2006-035505) for an Early Stage Researcher position. Support from MCnet (contract number MRTN-CT-2006-035606) is gratefully acknowledged.

A The VEGAS-improved HAAG algorithm

The HAAG phase space generator [8] is designed to produce momenta distributed approximately according to a QCD antenna function for an n-particle process

$$A_n(p_0, p_1, ..., p_{n-1}) = \frac{1}{(p_0p_1)(p_1p_2)...(p_{n-2}p_{n-1})(p_{n-1}p_0)}. \tag{5}$$

Different antennae can be obtained from permutations of the momenta p_i. Cyclic permutation and reversion of the order will however lead to the same structure. Generally HAAG relies on phase space factorisation over time-like intermediate momenta. In Ref. [8] two algorithms are proposed which are referred to as closed and open antenna and which differ in the decomposition of the 2-particle phase space $d\Phi_2$. Only the closed antenna contains all factors in Eq. (5), while in the open antenna one factor (p_ip_{i+1}) is missing. Although the closed antenna seems to be more symmetric, in practice it turns out that the open antenna is more efficient. This is mainly due to its simpler structure and less additional weight factors that appear within the algorithm.[1] In the following we will therefore focus on open antennas. The algorithm is reviewed for the case of massless external particles, however it can easily be generalised to the massive case.

A.1 Antenna Generation

In the following we use a classification of antenna types by the position of the incoming momenta, p_0 and p_1, within the antenna, see Fig. 1. The type is then given by $\mathrm{Min}(m-1, n-m-1)$.

The basic building block for antenna generation is the split of a massive momentum according to the phase space element $\mathrm{d}s\,\mathrm{d}\Phi_2(Q;p,P;q)$, where $P^2 = s$ and the last argument, q, defines an axis for the momentum generation. We further decompose

$$\mathrm{d}\Phi_2(Q;p,P;q) = \mathrm{d}a\,\mathrm{d}\phi\ , \quad \text{where} \quad a = \frac{q\cdot p}{q\cdot P} \tag{6}$$

and ϕ is an azimuthal angle around q.

The phase space for a single split, now defined through the variables s, a, ϕ, is constructed as follows[2]:

1. Dice s according to $1/s$ in $[s_{\min}, s_{\max}]$.
2. Dice a according to $1/a$ in $[a_{\min}, a_{\max}]$.
3. Dice ϕ according to a flat distribution in $[0, 2\pi]$.
4. The momenta are given by

$$\begin{aligned} p &= \left(\frac{Q^2-s}{2\sqrt{Q^2}}, \vec{p}\right)\ , \\ P &= \left(\frac{Q^2+s}{2\sqrt{Q^2}}, -\vec{p}\right)\ , \\ \vec{p} &= \left(h\cos\phi, h\sin\phi, \frac{Q^2(1-2a)-s}{2\sqrt{Q^2}}\right)\ , \quad \text{where} \quad h = \sqrt{Q^2a(1-a)-as}. \end{aligned} \tag{7}$$

[1] These weights are nonsingular in any of the products (p_ip_j).

[2] Frame dependent quantities are defined in the CM frame of Q with the z-Axis along q

Process	Time per ME AMEGIC++ [s]	Time per ME CSW [s]	AMEGIC++/CSW
$gg \to 2g$	6.35×10^{-6}	1.78×10^{-6}	3.6
$gg \to 3g$	1.68×10^{-4}	1.73×10^{-5}	9.7
$gg \to 4g$	3.63×10^{-2}	1.18×10^{-3}	31
$gg \to 5g$	-	3.29×10^{-2}	
$gg \to 6g$	-	4.56	
$gg \to 7g$	-	280	
$q\bar{q} \to 2g$	3.25×10^{-6}	1.45×10^{-6}	2.2
$q\bar{q} \to 3g$	3.40×10^{-5}	1.21×10^{-5}	2.8
$q\bar{q} \to 4g$	2.06×10^{-3}	8.80×10^{-4}	2.3
$q\bar{q} \to 5g$	0.614	2.65×10^{-2}	23
$q\bar{q} \to 6g$	-	2.96	
$q\bar{q} \to 7g$	-	170	
$q\bar{q} \to q\bar{q}$	1.36×10^{-6}	2.76×10^{-6}	0.49
$q\bar{q} \to q\bar{q}\ g$	1.11×10^{-5}	1.15×10^{-5}	1.0
$q\bar{q} \to q\bar{q}\ 2g$	4.48×10^{-4}	5.30×10^{-4}	0.85
$q\bar{q} \to q\bar{q}\ 3g$	8.98×10^{-2}	1.12×10^{-2}	8.0
$q\bar{q} \to q\bar{q}\ 4g$	-	0.934	
$q\bar{q} \to q\bar{q}\ 5g$	-	42.0	
$q\bar{q} \to q'\bar{q}'$	7.99×10^{-7}	1.52×10^{-6}	0.53
$q\bar{q} \to q'\bar{q}'\ g$	5.74×10^{-6}	5.65×10^{-6}	1.0
$q\bar{q} \to q'\bar{q}'\ 2g$	1.07×10^{-4}	2.72×10^{-4}	0.39
$q\bar{q} \to q'\bar{q}'\ 3g$	1.34×10^{-2}	5.80×10^{-3}	2.3
$q\bar{q} \to q'\bar{q}'\ 4g$	-	0.470	
$q\bar{q} \to q'\bar{q}'\ 5g$	-	20.94	
$q\bar{q} \to Z(\to e^-e^+)$	1.56×10^{-6}	3.88×10^{-6}	0.40
$q\bar{q} \to Z(\to e^-e^+)\ g$	3.99×10^{-6}	6.85×10^{-6}	0.58
$q\bar{q} \to Z(\to e^-e^+)\ 2g$	2.16×10^{-5}	1.07×10^{-4}	0.20
$q\bar{q} \to Z(\to e^-e^+)\ 3g$	2.34×10^{-4}	1.31×10^{-3}	0.18
$q\bar{q} \to Z(\to e^-e^+)\ 4g$	1.44×10^{-2}	8.20×10^{-2}	0.18

Tab. 1 Computation times for full matrix elements summed over colour and helicity. Displayed are averages for a single evaluation, employing Feynman diagrams computed in the helicity formalism (using AMEGIC++) and the Cachazo-Svrček-Witten (CSW) vertex rules. Numbers were generated on a 2.53 GHz Intel® Core™2 Duo T9400 CPU.

Process	Time per ME BG [s]	Time per ME CSW [s]	BG/CSW
$gg \to 2g$	8.42×10^{-6}	1.18×10^{-6}	7.1
$gg \to 3g$	3.19×10^{-5}	3.31×10^{-6}	9.6
$gg \to 4g$	1.13×10^{-4}	6.09×10^{-5}	2.0
$gg \to 5g$	3.58×10^{-4}	2.91×10^{-4}	1.3
$gg \to 6g$	1.17×10^{-3}	6.38×10^{-3}	0.20
$gg \to 7g$	3.99×10^{-3}	5.66×10^{-2}	0.079
$q\bar{q} \to 2g$	6.20×10^{-6}	1.02×10^{-6}	6.1
$q\bar{q} \to 3g$	2.18×10^{-5}	2.46×10^{-6}	8.9
$q\bar{q} \to 4g$	6.91×10^{-5}	4.59×10^{-5}	1.5
$q\bar{q} \to 5g$	2.15×10^{-4}	2.34×10^{-4}	0.92
$q\bar{q} \to 6g$	6.53×10^{-4}	4.00×10^{-3}	0.16
$q\bar{q} \to 7g$	2.03×10^{-3}	3.11×10^{-2}	0.065
$q\bar{q} \to q\bar{q}$	2.86×10^{-6}	1.56×10^{-6}	1.8
$q\bar{q} \to q\bar{q}\ g$	1.17×10^{-5}	3.26×10^{-6}	3.6
$q\bar{q} \to q\bar{q}\ 2g$	4.99×10^{-5}	5.92×10^{-5}	0.84
$q\bar{q} \to q\bar{q}\ 3g$	1.94×10^{-4}	2.90×10^{-4}	0.67
$q\bar{q} \to q\bar{q}\ 4g$	7.16×10^{-4}	4.93×10^{-3}	0.15
$q\bar{q} \to q\bar{q}\ 5g$	2.86×10^{-3}	3.69×10^{-2}	0.076
$q\bar{q} \to q'\bar{q}'$	2.24×10^{-6}	1.06×10^{-6}	2.1
$q\bar{q} \to q'\bar{q}'\ g$	8.97×10^{-6}	1.96×10^{-6}	4.6
$q\bar{q} \to q'\bar{q}'\ 2g$	2.87×10^{-5}	3.39×10^{-5}	0.85
$q\bar{q} \to q'\bar{q}'\ 3g$	8.18×10^{-5}	1.55×10^{-4}	0.59
$q\bar{q} \to q'\bar{q}'\ 4g$	2.70×10^{-4}	2.48×10^{-3}	0.11
$q\bar{q} \to q'\bar{q}'\ 5g$	8.13×10^{-4}	1.84×10^{-2}	0.044
$q\bar{q} \to Z(\to e^-e^+)$	3.84×10^{-6}	3.88×10^{-6}	0.99
$q\bar{q} \to Z(\to e^-e^+)\ g$	1.02×10^{-5}	6.85×10^{-6}	1.5
$q\bar{q} \to Z(\to e^-e^+)\ 2g$	2.57×10^{-5}	6.90×10^{-5}	0.37
$q\bar{q} \to Z(\to e^-e^+)\ 3g$	7.06×10^{-5}	2.95×10^{-4}	0.24
$q\bar{q} \to Z(\to e^-e^+)\ 4g$	1.95×10^{-4}	3.72×10^{-3}	0.052

Tab. 2 Average computation time of partial amplitudes in multi-gluon scattering, summed over all helicity configurations. Displayed are averages for a single evaluation, employing colour dressed Berends-Giele (BG) recursion and the Cachazo-Svrček-Witten (CSW) vertex rules. Numbers were generated on a 2.53 GHz Intel® Core™2 Duo T9400 CPU.

Parameter	Value
EW parameters in the G_μ scheme	
G_F	1.16639×10^{-5}
α_{QED}	1/132.51
$\sin^2\theta_W$	0.2222
M_Z	91.188 GeV
m_H	120 GeV
QCD parameters	
PDF set	CTEQ6L1
α_s	0.130
μ_F, μ_R	M_Z
jet, initial parton	g, u, d, s, c

Parameter	Value
Widths (fixed width scheme)	
Γ_Z	2.446 GeV
Cuts	
$p_{\perp,i}$	> 30 GeV
$\lvert\eta_i\rvert$	< 5
66 GeV $< m_{ll} <$ 116 GeV	
CDF Run II k_T algorithm [23]	
with $k_T > 30$ GeV and D=0.7	

Tab. 3 Parameters for the integration time comparison.

$pp \to n$ jets gluons only	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
MC cross section [pb]	$8.915 \cdot 10^7$	$5.454 \cdot 10^6$	$1.150 \cdot 10^6$	$2.757 \cdot 10^5$	$7.95 \cdot 10^4$
stat. error	0.1%	0.1%	0.2%	0.5%	1%
	integration time for given stat. error [s]				
CSW (HAAG)	4	165	1681	12800	$2 \cdot 10^6$
CSW (CSI)	-	480	6500	11900	197000
AMEGIC (HAAG)	6	492	41400	-	-
COMIX (RPG)	159	5050	33000	38000	74000
COMIX (CSI)	-	780	6930	6800	12400

Tab. 4 Cross section and evaluation times for different matrix element (phase space) generation methods for multi-gluon scattering at the LHC, given in pb. Numbers were generated on a 2.53 GHz Intel® Core™2 Duo T9400 CPU. For cuts and parameter settings, cf. Tab. 3.

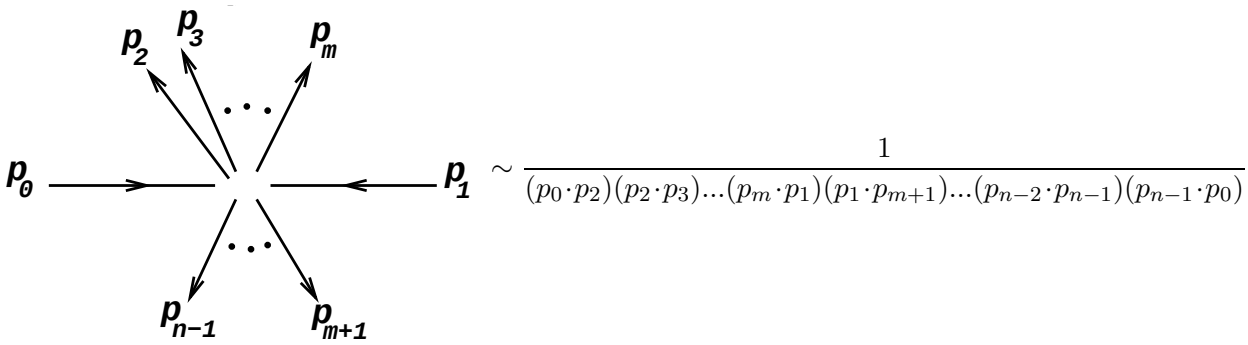
$pp \to n$ jets ≤ 1 quark line	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
MC cross section [pb]	$1.456 \cdot 10^8$	$1.051 \cdot 10^7$	$2.490 \cdot 10^6$	$6.75 \cdot 10^5$	$2.14 \cdot 10^5$
stat. error	0.1%	0.1%	0.2%	0.5%	1%
	integration time for given stat. error [s]				
CSW (HAAG)	10	354	6980	60000	$9 \cdot 10^6$
AMEGIC (HAAG)	13	930	73000	-	-
COMIX (RPG)	254	5370	15900	36800	64100

$pp \to n$ jets ≤ 2 quark lines	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
MC cross section [pb]	$1.5129 \cdot 10^8$	$1.1198 \cdot 10^7$	$2.831 \cdot 10^6$	$8.12 \cdot 10^5$	$2.71 \cdot 10^5$
stat. error	0.1%	0.1%	0.2%	0.5%	1%
	integration time for given stat. error [s]				
CSW (HAAG)	16	730	12300	120000	$2 \cdot 10^7$
AMEGIC (HAAG)	19	1530	78000	-	-
COMIX (RPG)	525	10800	25600	59000	113000

Tab. 5 Cross section and evaluation times for different matrix element (phase space) generation methods for multi-jet production at the LHC, given in pb. Numbers were generated on a 2.53 GHz Intel® Core™2 Duo T9400 CPU. For cuts and parameter settings, cf. Tab. 3.

$pp \to Z+$ jets	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
MC cross section [pb]	1080.8	121.67	54.67	23.59	11.22
stat. error	0.1%	0.1%	0.1%	0.2%	0.5%
	integration time for given stat. error [s]				
CSW (MC)	12	210	4100	57000	1500000
AMEGIC (MC)	7	98	1060	10400	310000
COMIX (RPG)	15	364	6400	16400	54000

Tab. 6 Cross section and evaluation times for different matrix element (phase space) generation methods for Z+jet production at the LHC, given in pb. Numbers were generated on a 2.53 GHz Intel® Core™2 Duo T9400 CPU. For cuts and parameter settings, cf. Tab. 3.



$$\sim \frac{1}{(p_0\cdot p_2)(p_2\cdot p_3)...(p_m\cdot p_1)(p_1\cdot p_{m+1})...(p_{n-2}\cdot p_{n-1})(p_{n-1}\cdot p_0)}$$

Fig. 1 Antenna configuration.

5. The weight is given by

$$\frac{g(s_{\min},s_{\max})}{s}\frac{g(a_{\min},a_{\max})}{a}\frac{1}{2\pi}\,, \quad \text{where} \quad g(x_{\min},x_{\max})=\log\frac{x_{\max}}{x_{\min}}\,. \tag{8}$$

Type 0 antennae

The phase space for type 0 antenna configurations can be obtained by a direct multiple application of the basic building block:

$$\begin{aligned}
\mathrm{d}\Phi_n(p_0,p_1;p_2,...,p_{n-1}) = \quad & \mathrm{d}s_2\;\mathrm{d}\Phi_2(p_0+p_1;p_2,Q_2;p_1)\\
&\times \mathrm{d}s_3\;\mathrm{d}\Phi_2(Q_2;p_3,Q_3;p_2)\\
&\quad\vdots\\
&\times \mathrm{d}s_{n-3}\;\mathrm{d}\Phi_2(Q_{n-4};p_{n-3},Q_{n-3};p_{n-4})\\
&\times \mathrm{d}\Phi_2(Q_{n-3};p_{n-1},p_{n-2};p_{n-3})\,.
\end{aligned} \tag{9}$$

The corresponding total weight is given by

$$w\sim\frac{\prod_{j=1}^{n-3}p_j\left(\sum_{i=j+1}^{n-1}p_i\right)}{\prod_{j=3}^{n-3}\left(\sum_{i=j}^{n-1}p_i\right)^2}\,\frac{1}{(p_1\cdot p_2)(p_2\cdot p_3)\cdots(p_{n-2}\cdot p_{n-1})}\,, \tag{10}$$

where the contributions from boundary dependent functions g have been omitted.

Type 1 antennae

For this configuration the following phase space decomposition is considered:

$$\begin{aligned}
\mathrm{d}\Phi_n(p_0,p_1;p_2,...,p_{n-1}) = \quad & \mathrm{d}s_2\;\mathrm{d}\Phi_2(p_0+p_1;p_2,Q_2;p_0)\\
&\times \mathrm{d}s_3\;\mathrm{d}\Phi_2(Q_2;p_3,Q_3;p_1)\\
&\times \mathrm{d}s_4\;\mathrm{d}\Phi_2(Q_3;p_4,Q_4;p_3)\\
&\quad\vdots\\
&\times \mathrm{d}s_{n-3}\;\mathrm{d}\Phi_2(Q_{n-4};p_{n-3},Q_{n-3};p_{n-4})\\
&\times \mathrm{d}\Phi_2(Q_{n-3};p_{n-1},p_{n-2};p_{n-3})\,.
\end{aligned} \tag{11}$$

In the first momentum split, $\mathrm{d}\Phi_2(p_0+p_1;p_2,Q_2;p_0)$, the variable a is now diced according to $\frac{1}{a(1-a)}$. All following splits are generated according to the basic building block. The corresponding total weight is given by

$$w \sim p_0{\cdot}(p_0+p_1-p_2)\,p_1{\cdot}(p_0+p_1-p_2)\frac{\prod_{j=3}^{n-3} p_j\left(\sum_{i=j+1}^{n-1} p_i\right)}{\prod_{j=3}^{n-3}\left(\sum_{i=j}^{n-1} p_i\right)^2}\,\frac{1}{(p_0{\cdot}p_2)(p_1{\cdot}p_3)\cdots(p_{n-2}{\cdot}p_{n-1})}\,. \quad (12)$$

Type k (≥ 2) antennae

In this case we have the following decomposition:

$$\begin{aligned}\mathrm{d}\Phi_n(p_0,p_1;p_2,...,p_{n-1}) = \quad & \mathrm{d}s_2\mathrm{d}s_k\;\mathrm{d}\Phi_2(p_0+p_1;p_2,Q_k;p_0)\\ & \times \mathrm{d}s_3\;\mathrm{d}\Phi_2(Q_2;p_3,Q_3;p_0)\\ & \quad\vdots\\ & \times \mathrm{d}s_{k-2}\;\mathrm{d}\Phi_2(Q_{k-3};p_{k-2},Q_{k-2};p_{k-3})\\ & \times \mathrm{d}\Phi_2(Q_{k-2};p_{k-1},p_k;p_{k-2})\\ & \times \mathrm{d}s_{k+1}\;\mathrm{d}\Phi_2(Q_k;p_{k+1},Q_{k+1};p_1)\\ & \quad\vdots\\ & \times \mathrm{d}s_{n-3}\;\mathrm{d}\Phi_2(Q_{n-4};p_{n-3},Q_{n-3};p_{n-4})\\ & \times \mathrm{d}\Phi_2(Q_{n-3};p_{n-1},p_{n-2};p_{n-3})\;.\end{aligned} \quad (13)$$

All splittings are generated according to the basic building block. The corresponding total weight is given by

$$\begin{aligned}w \sim \quad & p_1{\cdot}(p_{k+1}+...+p_{n-1})\frac{\prod_{j=2}^{k-2} p_j\left(\sum_{i=j+1}^{k} p_i\right)}{\prod_{j=2}^{k-2}\left(\sum_{i=j}^{k} p_i\right)^2}\,\frac{\prod_{j=k+1}^{n-3} p_j\left(\sum_{i=j+1}^{n-1} p_i\right)}{\prod_{j=k+1}^{n-3}\left(\sum_{i=j}^{n-1} p_i\right)^2}\\ & \times\frac{1}{(p_0{\cdot}p_2)(p_2{\cdot}p_3)\cdots(p_{k-1}{\cdot}p_k)(p_1{\cdot}p_{k+1})\cdots(p_{n-2}{\cdot}p_{n-1})}\,.\end{aligned}$$

A.2 HAAG and variance reducing techniques

To generate an adequate phase space integrator for realistic n-particle QCD processes, different HAAG channels can be combined using the multi-channel method [20]. Symbolically we can write a single channel as a map X from uniformly distributed random numbers $\vec{a}\in[0,1]^{3n-4}$ to the four-momenta $\vec{p}=(p_1,\ldots,p_n)$ of external particles, The corresponding phase space weight g is given by

$$\frac{1}{g}=\frac{\mathrm{d}\Phi_n(X(\vec{a}))}{\mathrm{d}\vec{a}}\,. \quad (14)$$

The multi-channel method now combines several maps X_i to a new map as follows:

$$\mathbf{X}(\vec{a},\tilde{\alpha})=X_k(\vec{a})\;,\quad \text{for}\quad \sum_{l=1}^{k-1}\alpha_l<\tilde{\alpha}<\sum_{l=1}^{k}\alpha_l\;, \quad (15)$$

requiring an additional random number $\tilde{\alpha}$ and arbitrary coefficients α_k with $\alpha_k>0$ and $\sum_k\alpha_k=1$. The corresponding phase space weight is given by

$$G=\sum_k \alpha_k\;g_k\;. \quad (16)$$

The coefficients α_k can be adapted such that the variance of the phase space integral is minimised. The efficiency of the integrator is improved if additionally the VEGAS algorithm [9] is applied to the single channels. VEGAS is very efficient in the numerical adaptation to functions, whose peaking behaviour is not too extreme and which are factorisable to a product of one-dimensional functions. Although this is usually not the case for a full differential cross section, it can be used to better adapt the antenna-like structures in a single HAAG-channel to the corresponding structures in the matrix elements, including phase space cuts.

For each channel k, VEGAS is used to generate a mapping ξ_k from uniformly distributed random numbers to a non-uniform distribution, still inside the interval $[0,1]$, and a corresponding weight v_k. To combine this with the multi-channel method the mapping $X(\vec{a})$ for single channels must be invertible, which is the case for HAAG channels. The full map reads

$$\mathbf{X}(\vec{a},\tilde{\alpha}) \quad = \quad X_k(\xi_k(\vec{a}))\ , \quad \text{for} \quad \sum_{l=1}^{k-1}\alpha_l < \tilde{\alpha} < \sum_{l=1}^{k}\alpha_l\ . \tag{17}$$

For a momentum configuration $\vec{p}$ the weight is therefore given by

$$G(\vec{p}) = \sum_k \alpha_k\ g_k(\vec{p})\ v_k(X_k^{-1}(\vec{p}))\ . \tag{18}$$

We can make use of the equivalence of HAAG-channels of the same type, such that all of them employ the same VEGAS map. This alleviates the adaptation significantly, since we are left with only very few maps and a linear growth with the number of particles.

References

[1] F. Cachazo, P. Svrček and E. Witten, *MHV vertices and tree amplitudes in gauge theory*, JHEP **09** (2004), 006, [hep-th/0403047].

[2] E. Witten, *Perturbative Gauge Theory as a String Theory in Twistor Space*, Commun. Math. Phys. **252** (2004), 189–258, [hep-th/0312171]. F. Cachazo and P. Svrček, *Lectures on Twistor Strings and Perturbative Yang-Mills Theory*, PoS **RTN2005** (2005), 004, [hep-th/0504194].

[3] R. Britto, F. Cachazo and B. Feng, *New Recursion Relations for Tree Amplitudes of Gluons*, Nucl. Phys. **B715** (2005), 499–522, [hep-th/0412308]. R. Britto, F. Cachazo, B. Feng and E. Witten, *Direct proof of tree-level recursion relation in Yang-Mills theory*, Phys. Rev. Lett. **94** (2005), 181602, [hep-th/0501052].

[4] R. Kleiss and W. J. Stirling, *Spinor techniques for calculating $p\bar{p} \to W^{\pm}/Z^0$ + jets*, Nucl. Phys. **B262** (1985), 235–262. A. Ballestrero and E. Maina, *A new method for helicity calculations*, Phys. Lett. **B350** (1995), 225–233, [hep-ph/9403244].

[5] T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann and J. Winter, *SHERPA 1.α, a proof-of-concept version*, JHEP **02** (2004), 056, [hep-ph/0311263].

[6] E. Byckling and K. Kajantie, *N-particle phase space in terms of invariant momentum transfers*, Nucl. Phys. **B9** (1969), 568–576.

[7] F. Krauss, R. Kuhn and G. Soff, *AMEGIC++ 1.0: A Matrix Element Generator In C++*, JHEP **02** (2002), 044, [hep-ph/0109036].

[8] A. van Hameren and C. G. Papadopoulos, *A hierarchical phase space generator for QCD antenna structures*, Eur. Phys. J. **C25** (2002), 563–574, [hep-ph/0204055].

[9] G. P. Lepage, *VEGAS: An Adaptive Multi-dimensional Integration Program*, CLNS-80/447. T. Ohl, *Vegas revisited: Adaptive Monte Carlo integration beyond factorization*, Comput. Phys. Commun. **120** (1999), 13–19, [hep-ph/9806432].

[10] M. Dinsdale, M. Ternick and S. Weinzierl, *A comparison of efficient methods for the computation of Born gluon amplitudes*, JHEP **03** (2006), 056, [hep-ph/0602204].

[11] C. Duhr, S. Höche and F. Maltoni, *Color-dressed recursive relations for multi-parton amplitudes*, JHEP **08** (2006), 062, [hep-ph/0607057].

[12] T. Gleisberg and S. Höche, *Comix, a new Matrix Element generator*, arXiv:yymm.nnnn [hep-ph].

[13] M. L. Mangano and S. J. Parke, *Multi-Parton Amplitudes in Gauge Theories*, Phys. Rept. **200** (1991), 301–367, [hep-th/0509223].

[14] V. Del Duca, L. J. Dixon and F. Maltoni, *New Color Decompositions for Gauge Amplitudes at Tree and Loop Level*, Nucl. Phys. **B571** (2000), 51–70, [hep-ph/9910563].

[15] S. J. Parke and T. R. Taylor, *Amplitude for n-Gluon Scattering*, Phys. Rev. Lett. **56** (1986), 2459. F. A. Berends and W. T. Giele, *Recursive calculations for processes with n gluons*, Nucl. Phys. **B306** (1988), 759.

[16] L. J. Dixon, *Calculating scattering amplitudes efficiently*, hep-ph/9601359.

[17] M. L. Mangano, S. J. Parke and Z. Xu, *Duality and multi-gluon scattering*, Nucl. Phys. **B298** (1988), 653. M. L. Mangano and S. J. Parke, *Quark - gluon amplitudes in the dual expansion*, Nucl. Phys. **B299** (1988), 673.

[18] M. L. Mangano, *The color Structure of gluon emission*, Nucl. Phys. **B309** (1988), 461. F. A. Berends, W. T. Giele and H. Kuijf, *Exact expressions for processes involving a vector boson and up to five partons*, Nucl. Phys. **B321** (1989), 39.

[19] R. Kleiss, W. J. Stirling and S. D. Ellis, *A new Monte Carlo treatment of multiparticle phase space at high energies*, Comput. Phys. Commun. **40** (1986), 359. P. D. Draggiotis, A. van Hameren and R. Kleiss, *SARGE: an algorithm for generating QCD-antennas*, Phys. Lett. **B483** (2000), 124–130, [hep-ph/0004047].

[20] R. Kleiss and R. Pittau, *Weight optimization in multichannel Monte Carlo*, Comput. Phys. Commun. **83** (1994), 141–146, [hep-ph/9405257].

[21] http://mlm.home.cern.ch/mlm/mcwshop03/mcwshop.html.

[22] http://www.scotgrid.ac.uk.

[23] G. C. Blazey et al., *Run II Jet Physics*, hep-ex/0005012.