Published by IOP Publishing for SISSA

Received: *September 2, 2008*
Revised: *November 17, 2008*
Accepted: *November 18, 2008*
Published: *December 10, 2008*

JHEP12(2008)039

Comix, a new matrix element generator

Tanju Gleisberg

Stanford Linear Accelerator Center, Stanford University,
Stanford, CA 94309, U.S.A.
E-mail: tanju@slac.stanford.edu

Stefan Höche

Institute for Particle Physics Phenomenology, Durham University,
Durham DH1 3LE, U.K.
E-mail: stefan.hoeche@durham.ac.uk

Abstract: We present a new tree-level matrix element generator, based on the colour dressed Berends-Giele recursive relations. We discuss two new algorithms for phase space integration, dedicated to be used with large multiplicities and colour sampling.

Keywords: Jets, QCD, Standard Model.

© SISSA 2008

Contents

1. Introduction **1**

2. Recursive relations for tree-level amplitudes in the Standard Model **2**
- 2.1 The cost of computing a tree amplitude 3
- 2.2 General form of the recursive relations 4
- 2.3 Colour dressed Berends-Giele recursive relations in QCD 5
- 2.4 Decomposition of electroweak four-particle vertices 7
- 2.5 Prefactors of diagrams with external fermions 9

3. Matrix element generation in Comix **9**

4. Integration techniques in Comix **12**
- 4.1 Recursive algorithm for phase space integration 12
  - 4.1.1 Brief review of phase space factorisation 12
  - 4.1.2 A simple example 14
  - 4.1.3 Formulation of the recursive algorithm 15
  - 4.1.4 Implementation details 18
- 4.2 Colour sampling 18
  - 4.2.1 Determination of colour flows from colour assignments 18
  - 4.2.2 Selection of colour assignments 19
  - 4.2.3 A simple example 19
- 4.3 Combined colour-momentum integration techniques 20
  - 4.3.1 Integration of partial amplitudes and colour configurations 20
  - 4.3.2 Optimisation techniques 21

5. Results **22**
- 5.1 Helicity summation vs. helicity sampling 22
- 5.2 Performance of the CSI and $2 \to n$ gluon benchmarks 22
- 5.3 Performance of the RPG and comparison with other generators 23

6. Conclusions **25**

1. Introduction

In recent years considerable progress has been made in the calculation of full matrix elements (ME) for higher order perturbative corrections to Standard Model (SM) processes, QCD and QCD associated processes in particular. Automatic computation of NLO virtual corrections to arbitrary processes finally seems within reach due to newly emerging

JHEP12(2008)039

numerical techniques [1 – 8]. On-shell recursive methods proved to yield compact expressions for multi-leg tree-level amplitudes with massless [9, 10] and massive [11 – 13] external particles and are now widely used. The CSW vertex rules [14 – 16] as off-shell techniques are employed in many analytical and numerical approaches [17 – 20].

Apart from major developments in the computation of loop amplitudes, many attempts have been made to tackle the task of numerically evaluating tree-level amplitudes with large numbers of external legs. They led to the construction of several programs, capable of evaluating general tree-level processes [21 – 27]. In this context it turned out, that with increasing number of particles involved in the scattering one of the the most efficient methods to compute colour-ordered amplitudes is the Berends-Giele recursion [28 – 33]. Correspondingly the fastest methods available for the computation of full scattering amplitudes are the colour dressed Berends-Giele relations [34], which are essentially equivalent to the Dyson-Schwinger methods employed in refs. [35], with the ALPHA algorithm of ref. [36] being comparable in efficiency. In refs. [35] and [34] it was pointed out that a vertex decomposition of four-gluon vertices in QCD is clearly advantageous if the speed of numerical implementations is concerned. These findings raise the question, whether it is possible to construct a full set of SM Feynman rules with no four vertices present in the theory, such that recursive techniques analogous to the colour dressed Berends-Giele relations can be employed in numerical programs. In section 2 we demonstrate that this is feasible. We discuss the numerical implementation of the results in the new ME generator Comix in section 3 and present code-related aspects, such as a multi-threading concept.

A very important part of computing cross sections for tree-level processes is, to find an efficient algorithm for phase space generation. If colours are sampled over, similar problems arise for colour space. An effective general technique for phase space generation has been presented in ref. [37]. We observe in section 4.1, that it is possible to formulate the rules presented ibidem in a truly recursive fashion, i.e. on the same footing as the matrix element computation. This implies in particular, that point by point the same calculational effort is spent for computing matrix element and phase space weight. We introduce effective colour sampling techniques in section 4.2. Having these techniques at hand, we elaborate on how to eventually couple colour and phase space integration and propose a new type of integrator based on the Haag generator [38] in section 4.3.

We present a comprehensive comparison of results generated with Comix to those generated with the two other multi-leg tree-level matrix element generators Amegic++ [24] and Alpgen [27] in section 5. Section 6 contains our conclusions.

2. Recursive relations for tree-level amplitudes in the Standard Model

It has been pointed out, for example in refs. [33 – 35], that the calculation of multi-parton amplitudes is substantially simplified when employing Berends-Giele type recursive relations. One main reason for the simplification is that these relations allow to reuse basic building blocks of an amplitude, which are the m-particle internal off-shell currents. Another reason is that they can be easily rewritten to include three-particle vertices only. In the following we will briefly illuminate, why this is a major advantage.

JHEP12(2008)039

2.1 The cost of computing a tree amplitude

As an example, we try to estimate the total computational cost for tree amplitudes, given a certain type of vertices in the underlying theory. We assume that only one particle type exists and the internal n-particle currents obey a recursion, which is of the functional form

$$J_n(\pi) = P_n(\pi) \sum_{N=1}^{n} \sum_{\mathcal{P}_N(\pi)} V_N(\pi_1, \ldots, \pi_N) J_{i_1}(\pi_1) \ldots J_{i_N}(\pi_N) \ . \tag{2.1}$$

Here J_m denote unordered m-particle currents, while V_N are $N+1$-point vertices and P_n is a propagator term. The two sums run over all possible vertex types V_N and all (unordered) partitions $\mathcal{P}_N(\pi)$ of the set of particles π into N (unordered) subsets, respectively [34]. The full $n+1$-particle scattering amplitude can be constructed by putting an arbitrary n-particle internal off-shell current on-shell and contracting the remaining quantity with the corresponding external one-particle current.

$$A_{n+1}(\pi) = J_1(i) \frac{1}{P_n(\pi \setminus i)} J_n(\pi \setminus i) \ . \tag{2.2}$$

We now deal only with vertices of $N+1$ external legs and we consider their contribution to the computation of an n-particle off-shell current. The number of vertices to evaluate per m-particle subcurrent is the Stirling number of the second kind $S(m, N)$, corresponding to the number of partitions of a set π of m integers into N subsets. The total number $V(n, N)$ of $N+1$-particle vertices to be calculated thus becomes

$$V(n, N) = \sum_{m=N}^{n} \binom{n}{m} S(m, N) \ . \tag{2.3}$$

Since the Stirling numbers $S(m, N)$ are zero for $m < N$, we can extend the sum down to zero, leading to

$$\begin{aligned} V(n, N) &= \sum_{m=0}^{n} \binom{n}{m} \frac{1}{N!} \sum_{i=0}^{N} (-1)^i \binom{N}{i} (N-i)^m \\ &= \frac{1}{(N+1)!} \sum_{i=0}^{N} (-1)^i \binom{N+1}{i} (N+1-i)^{n+1} = S(n+1, N+1) \ . \end{aligned} \tag{2.4}$$

The question is, whether we can obtain a milder growth in computational complexity, if all $N+1$-particle vertices occuring in eq. (2.1) are decomposed in terms of two or more vertices with fewer number of external legs. When doing so, we must introduce additional pseudoparticles reflecting the structure of the decomposed vertex. Hence we have to consider the contribution arising from the presence of these pseudoparticles, too. The problem can be simplified by assuming that there is only one additional pseudoparticle, which obeys a completely independent recursion relation. Then the full contribution of an $N+1$-particle vertex, now being decomposed into a $M+1$- and a $N-M+1$-particle vertex becomes

$$S(n+1, N+1) \to S(n+1, M+1) + S(n+1, N-M+1) \ , \tag{2.5}$$

JHEP12(2008)039

which can be either bigger or smaller than $S(n+1, N+1)$, depending on n, N and M. With increasing n, however the right hand side is always smaller such that the vertex decomposition becomes clearly advantageous. Similar arguments hold when introducing more than one pseudoparticle.

From this simple but general consideration we see that the aim of any recursive formulation of interaction models should be, to reduce the number of external lines at interaction vertices to the lowest possible. In this section we will show that within the Standard Model it is possible to reduce $N_{\max}$ to two, which is the lowest possible number in general. For QCD interactions we employ the results of ref. [34], where this task has already been performed and the original Berends-Giele recursive relations have been reformulated to incorporate colour.

2.2 General form of the recursive relations

In the following we will denote by $\mathcal{J}_\alpha(\pi)$ an unordered SM current of type α, which receives contributions from all Feynman graphs having as external particles the on-shell SM particles in the set π and one internal particle, described by this current. The index α is a multi-index, carrying information on all quantum numbers and eventually on the pseudoparticle character of the particle. Special currents are given by the external particle currents. They correspond to external scalars, spinors and polarisation vectors, see section 3. For them there is only one multi-index $\alpha = \alpha_i$ associated with the external particle i, whereas in the general case multiple multi-indices may lead to non-vanishing internal currents. This corresponds to multiple particle types being possible as intermediate states. Assuming that only three-point vertices exist, any internal SM particle and pseudoparticle off-shell current can be written as

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{\mathcal{V}_\alpha^{\alpha_1,\alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi_1, \pi_2)\ \mathcal{V}_\alpha^{\alpha_1,\alpha_2}(\pi_1, \pi_2)\ \mathcal{J}_{\alpha_1}(\pi_1)\, \mathcal{J}_{\alpha_2}(\pi_2)\ . \tag{2.6}$$

Here $P_\alpha(\pi)$ denotes a propagator term depending on the particle type α and the set π. The term $\mathcal{V}_\alpha^{\alpha_1,\alpha_2}(\pi_1, \pi_2)$ is a vertex depending on the particle types α, α_1 and α_2 and the decomposition of the set π into disjoint subsets π_1 and π_2. The quantity $\mathcal{S}(\pi_1, \pi_2)$ is the symmetry factor associated with the decomposition of π into π_1 and π_2 and will be discussed in section 2.5. Superscripts in this context refer to incoming particles, subscripts to outgoing particles. The sums run over all vertices in the reformulated Standard Model and all unordered partitions $\mathcal{P}_2$ of the set π into two disjoint subsets, respectively. A full unordered n-particle scattering amplitude is then given by

$$\mathcal{A}(\pi) = \mathcal{J}_{\alpha_n}(n)\ \frac{1}{P_{\bar{\alpha}_n}(\pi \setminus n)}\, \mathcal{J}_{\bar{\alpha}_n}(\pi \setminus n)\ , \tag{2.7}$$

where $\bar{\alpha}$ denotes a set of reversed particle properties, i.e. opposite helicity, colour, momentum and particle type. It has been proved in ref. [34] that the above form is correct for pure gluonic scattering amplitudes once the four gluon vertex is suitably decomposed into two vertices involving an internal antisymmetric tensor pseudoparticle. We briefly recall this

JHEP12(2008)039

proof before continuing with the decomposition of four particle vertices in electroweak interactions. Once this decomposition is achieved, no further complications arise and eq. (2.6) can be employed to compute arbitrary scattering amplitudes in the Standard Model.

2.3 Colour dressed Berends-Giele recursive relations in QCD

Any perturbative QCD scattering amplitude $\mathcal{A}$ can be written as a sum of terms, which factorise into two components, one only depending on the gauge structure and one only depending on the kinematics. Such a decomposition is called colour decomposition. Considering for example tree-level n-gluon amplitudes, several colour decompositions exist. A very intuitive one based on the fundamental representation of the gauge group is given by [39]

$$\mathcal{A}(1,\ldots,n) = \sum_{\vec{\sigma}\in S_{n-1}} \mathrm{Tr}\left(T^{a_1}T^{a_{\sigma_2}}\ldots T^{a_{\sigma_n}}\right)\; A(1,\sigma_2,\ldots,\sigma_n)\ . \tag{2.8}$$

Here $\vec{\sigma}$ runs over all permutations S_{n-1} of the $n-1$ indices $2\ldots n$. The functions A depend on the Lorentz-structure of the process only and are called colour-ordered amplitudes. A more suitable colour decomposition for n-gluon amplitudes has been introduced in refs. [40, 41]. It employs the adjoint representation matrices $(F^a)_{bc}$ of SU(3) and reads

$$\mathcal{A}(1,\ldots,n) = \sum_{\vec{\sigma}\in S_{n-2}} \left(F^{a_{\sigma_2}}\ldots F^{a_{\sigma_{n-1}}}\right)_{a_1 a_n}\; A(1,\sigma_2,\ldots,\sigma_{n-1},n)\ . \tag{2.9}$$

Note that in this case the sum runs over the permutations of the $n-2$ indices $2\ldots n-1$ only, whereas the first and the last index remain fixed. Another colour decomposition, suited especially for Monte Carlo event generation is the colour flow decomposition [42]. In this prescription the SU(3) gluon field is treated as a 3×3 matrix $(A_\mu)^{i\bar{\jmath}}$ rather than a one index field A^a_μ. The corresponding decomposition reads

$$\mathcal{A}(1,\ldots,n) = \sum_{\vec{\sigma}\in S_{n-1}} \delta^{i_1\bar{\jmath}_{\sigma_2}}\delta^{i_{\sigma_2}\bar{\jmath}_{\sigma_3}}\ldots\delta^{i_{\sigma_n}\bar{\jmath}_1}\; A(1,\sigma_2,\ldots,\sigma_n)\ . \tag{2.10}$$

The remaining task is now, to compute the colour-ordered amplitudes. In ref. [28] Berends and Giele proposed a method to do so in a recursive fashion. The basic idea is that, according to the Feynman rules of QCD, an internal n-gluon current is defined by all contributing Feynman graphs with n external on-shell gluons and one off-shell gluon.

$$\begin{aligned} J_\mu(1,2,\ldots,n) = \frac{-ig_{\mu\nu}}{p^2_{1,n}} \Bigg\{ & \sum_{k=1}^{n-1} V_3^{\nu\kappa\lambda}(p_{1,k},p_{k+1,n})\, J_\kappa(1,\ldots,k)\, J_\lambda(k+1,\ldots,n) \\ & + \sum_{j=1}^{n-2}\sum_{k=j+1}^{n-1} V_4^{\nu\rho\kappa\lambda} J_\rho(1,\ldots,j)\, J_\kappa(j+1,\ldots,k)\, J_\lambda(k+1,\ldots,n)\Bigg\}. \end{aligned} \tag{2.11}$$

Here p_i denote the momenta of the gluons, $p_{i,j} = p_i + \cdots + p_j$ and $V_3^{\nu\kappa\lambda}(p_{1,k},p_{k+1,n})$ and $V_4^{\nu\rho\kappa\lambda}$ are the colour-ordered three and four-gluon vertices defined according to ref. [43],

$$\begin{aligned} V_3^{\nu\kappa\lambda}(p,q) &= i\,\frac{g_s}{\sqrt{2}}\left(g^{\kappa\lambda}(p-q)^\nu + g^{\lambda\nu}(2q+p)^\kappa - g^{\nu\kappa}(2p+q)^\lambda\right)\ , \\ V_4^{\nu\rho\kappa\lambda} &= i\,\frac{g_s^2}{2}\left(2g^{\nu\kappa}g^{\rho\lambda} - g^{\nu\rho}g^{\kappa\lambda} - g^{\nu\lambda}g^{\rho\kappa}\right)\ . \end{aligned} \tag{2.12}$$

JHEP12(2008)039

The full colour-ordered n-gluon amplitude $A(1,\ldots,n)$ is then obtained by putting the $n-1$-particle off-shell current $J_{n-1}(1,\ldots,n-1)$ on-shell and contracting it with the external polarisation $J_\mu(n)$. Employing the tensor-gluon vertex

$$V_T^{\mu\nu\kappa\lambda} = \frac{i}{2}\frac{g_s}{\sqrt{2}}\left(g^{\mu\kappa}g^{\nu\lambda} - g^{\mu\lambda}g^{\nu\kappa}\right) , \tag{2.13}$$

and the tensor "propagator"

$$-iD_{\mu\nu}^{\kappa\lambda} = -i\left(g_\mu^\kappa g_\nu^\lambda - g_\mu^\lambda g_\nu^\kappa\right) , \tag{2.14}$$

the recursion can be reformulated to give

$$\begin{aligned} J_\mu(1,2,\ldots,n) = \frac{-ig_{\mu\nu}}{p_{1,n}^2}\sum_{k=1}^{n-1}\Big\{ & V_3^{\nu\kappa\lambda}(p_{1,k},p_{k+1,n})\,J_\kappa(1,\ldots,k)\,J_\lambda(k+1,\ldots,n) \\ & + V_T^{\nu\kappa\alpha\beta}J_\kappa(1,\ldots,k)\,J_{\alpha\beta}(k+1,\ldots,n) \\ & + V_T^{\lambda\nu\alpha\beta}J_{\alpha\beta}(1,\ldots,k)\,J_\lambda(k+1,\ldots,n)\Big\} \end{aligned} \tag{2.15}$$

and

$$J^{\alpha\beta}(1,2,\ldots,n) = -iD_{\gamma\delta}^{\alpha\beta}\sum_{k=1}^{n-1}V_T^{\gamma\delta\kappa\lambda}J_\kappa(1,\ldots,k)\,J_\lambda(k+1,\ldots,n) , \tag{2.16}$$

for the gluon and tensor pseudoparticle currents, respectively. Since no external tensor currents exist, all tensor currents with one particle index only are defined as zero. The advantage of the above formulation including a tensor current, as discussed in section 2.1, is the elimination of the four-gluon vertex. Correspondingly we introduce a "pseudogluon", which, from here on, we denote by g_4.

Following ref. [34], one can introduce colour dressed gluon and tensor pseudoparticle currents $\mathcal{J}_{\mu\,I\bar{J}}$ and $\mathcal{J}_{\alpha\beta\,I\bar{J}}$, defined by

$$\begin{aligned} \mathcal{J}_{\mu\,I\bar{J}}(1,\ldots,n) &= \sum_{\vec{\sigma}\in S_n}\delta_{I\bar{\jmath}_{\sigma_1}}\delta_{i_{\sigma_1}\bar{\jmath}_{\sigma_2}}\ldots\delta_{i_{\sigma_n}\bar{J}}\,J_\mu(\sigma_1,\ldots,\sigma_n) , \\ \mathcal{J}_{\alpha\beta\,I\bar{J}}(1,\ldots,n) &= \sum_{\vec{\sigma}\in S_n}\delta_{I\bar{\jmath}_{\sigma_1}}\delta_{i_{\sigma_1}\bar{\jmath}_{\sigma_2}}\ldots\delta_{i_{\sigma_n}\bar{J}}\,J_{\alpha\beta}(\sigma_1,\ldots,\sigma_n) . \end{aligned} \tag{2.17}$$

Denoting by π the set $(1,\ldots,n)$ of n particles, the following recursive relations for these currents are obtained:

$$\begin{aligned} \mathcal{J}_{\mu\,I\bar{J}}(\pi) = D_{\mu\,I\bar{J}}^{\nu\,H\bar{G}}(\pi)\Bigg\{ & \sum_{\mathcal{P}_2(\pi)}\mathcal{V}_{\nu\,H\bar{G}}^{\kappa\,K\bar{L},\,\lambda\,M\bar{N}}(\pi_1,\pi_2)\;\mathcal{J}_{\kappa\,K\bar{L}}(\pi_1)\,\mathcal{J}_{\lambda\,M\bar{N}}(\pi_2) \\ & + \sum_{\mathcal{OP}_2(\pi)}\mathcal{V}_{\nu\,H\bar{G}}^{\kappa\,K\bar{L},\,\alpha\beta\,M\bar{N}}\,\mathcal{J}_{\kappa\,K\bar{L}}(\pi_1)\,\mathcal{J}_{\alpha\beta\,M\bar{N}}(\pi_2)\Bigg\} , \\ \mathcal{J}_{\alpha\beta\,I\bar{J}}(\pi) = D_{\alpha\beta\,I\bar{J}}^{\gamma\delta\,H\bar{G}} & \sum_{\mathcal{P}_2(\pi)}\mathcal{V}_{\gamma\delta\,H\bar{G}}^{\kappa\,K\bar{L},\,\lambda\,M\bar{N}}\,\mathcal{J}_{\kappa\,K\bar{L}}(\pi_1)\,\mathcal{J}_{\lambda\,M\bar{N}}(\pi_2) . \end{aligned} \tag{2.18}$$

JHEP12(2008)039

Here we have defined the colour dressed gluon and tensor pseudoparticle vertices

$$\mathcal{V}_{\nu\,H\bar{G}}^{\kappa\,K\bar{L},\,\lambda\,M\bar{N}}\,(\pi_1,\pi_2) = \delta_{\bar{G}}^{\bar{L}}\delta^{K\bar{N}}\delta_H^M\,V_{3\,\nu}^{\;\;\kappa\lambda}\,(\pi_1,\pi_2) + \delta_H^K\delta^{M\bar{L}}\delta_{\bar{G}}^{\bar{N}}\,V_{3\,\nu}^{\;\;\lambda\kappa}\,(\pi_2,\pi_1)\ , \tag{2.19}$$

and

$$\mathcal{V}_{\gamma\delta\,H\bar{G}}^{\kappa\,K\bar{L},\,\lambda\,M\bar{N}} = \delta_{\bar{G}}^{\bar{L}}\delta^{K\bar{N}}\delta_H^M\,V_{T\,\gamma\delta}^{\;\;\kappa\lambda} + \delta_H^K\delta^{M\bar{L}}\delta_{\bar{G}}^{\bar{N}}\,V_{T\,\gamma\delta}^{\;\;\lambda\kappa}\ . \tag{2.20}$$

The second sum runs over the set of ordered partitions of the set π into two disjoint subsets, $\mathcal{OP}_2(\pi)$. A complete proof of these relations can be found in ref. [34]. The above procedure of colour dressing can easily be generalised to QCD processes including quarks. Since no further elementary QCD four-point interactions exists, no further vertex decomposition has to be performed and therefore no new current types are introduced. For amplitudes including quarks care must be taken of using the proper colour space gluon propagator when coupling to $q\bar{q}g$ vertices, i.e.

$$P_{g\,I\bar{J}}^{\;\;H\bar{G}}\ \propto\ \delta_I^H\,\delta_{\bar{J}}^{\bar{G}} - \frac{1}{N_C}\delta_{I\bar{J}}\,\delta^{H\bar{G}}\ , \tag{2.21}$$

as described in ref. [42].

2.4 Decomposition of electroweak four-particle vertices

The above procedure can be generalised to describe all Standard Model interactions, once a suitable replacement of the corresponding four particle vertices has been found.

We start by proposing a decomposition of four particle vertices with W-bosons only[1]

$$\begin{aligned}\mathcal{V}_{W^-\mu}^{W^-\kappa,\,W^+\nu,\,W^-\lambda} \rightarrow\quad & \mathcal{V}_{W^-\mu}^{W^-\kappa,\,Z_4\gamma\delta}\cdot P_{Z_4\,\gamma\delta}^{\;\;\alpha\beta}\cdot\mathcal{V}_{Z_4\alpha\beta}^{W^+\nu,\,W^-\lambda}\\ & +\ \mathcal{V}_{W^-\mu}^{W^-\lambda,\,Z_4\gamma\delta}\cdot P_{Z_4\,\gamma\delta}^{\;\;\alpha\beta}\cdot\mathcal{V}_{Z_4\alpha\beta}^{W^+\nu,\,W^-\kappa}\ .\end{aligned} \tag{2.22}$$

Here Z_4 denotes a new antisymmetric tensor pseudoparticle introduced for the vertex decomposition. Its interaction vertex reads

$$\begin{aligned}\mathcal{V}_{W^-\mu}^{W^-\kappa,\,Z_4\gamma\delta} &= \frac{i}{2}\,g_w\,\left(g_\mu^\gamma g^{\kappa\delta} - g_\mu^\delta g^{\kappa\gamma}\right)\ ,\\ \mathcal{V}_{Z_4\alpha\beta}^{W^+\kappa,\,W^-\lambda} &= \frac{i}{2}\,g_w\,\left(g_\alpha^\kappa g_\beta^\lambda - g_\alpha^\lambda g_\beta^\kappa\right)\ .\end{aligned} \tag{2.23}$$

To obtain correct signs of four-particle vertices, we define the tensor pseudoparticle "propagators" as

$$P_{\alpha\,\mu\nu}^{\;\;\kappa\lambda} = \kappa_\alpha D_{\mu\nu}^{\kappa\lambda} \qquad \text{where} \qquad \kappa_\alpha = \begin{cases} -i \text{ if} & \alpha\ =\ Z_4 \\ i & \text{else}\end{cases}\ , \tag{2.24}$$

and where $D_{\mu\nu}^{\kappa\lambda}$ is given by eq. (2.14). Note that the Z_4 pseudoparticle is not self-conjugate. This definition prevents double counting four-particle vertices involving the W boson and

[1]Note that this decomposition of vertices is not unique and other choices may exist.

JHEP12(2008)039

constructing fake $WWWW$ vertices with all W's having the same charge. The four-particle vertices involving W bosons, photons and Z-bosons are decomposed as follows

$$\begin{aligned}
\mathcal{V}_{W^-\mu}^{A\kappa,\,W^-\nu,\,A\lambda} \rightarrow \quad & \mathcal{V}_{W^-\mu}^{A\kappa,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,A\lambda} \\
& + \mathcal{V}_{W^-\mu}^{A\lambda,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,A\kappa} \,, \\
\mathcal{V}_{W^-\mu}^{A\kappa,\,W^-\nu,\,Z\lambda} \rightarrow \quad & \mathcal{V}_{W^-\mu}^{A\kappa,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,Z\lambda} \\
& + \mathcal{V}_{W^-\mu}^{Z\lambda,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,A\kappa} \,, \\
\mathcal{V}_{W^-\mu}^{Z\kappa,\,W^-\nu,\,Z\lambda} \rightarrow \quad & \mathcal{V}_{W^-\mu}^{Z\kappa,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,Z\lambda} \\
& + \mathcal{V}_{W^-\mu}^{Z\lambda,\,W_4^-\gamma\delta} \cdot P_{W_4^-\,\gamma\delta}^{\quad\alpha\beta} \cdot \mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,Z\kappa} \,.
\end{aligned} \tag{2.25}$$

We introduced a new tensor pseudoparticle, W_4^-, whose interaction vertices are defined as

$$\begin{aligned}
\mathcal{V}_{W^-\mu}^{A\kappa,\,W_4^-\gamma\delta} &= \frac{i}{2}\, g_w \sin\theta_W \, \left(g_\mu^\gamma g^{\kappa\delta} - g_\mu^\delta g^{\kappa\gamma}\right) \,, \\
\mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,A\kappa} &= \frac{i}{2}\, g_w \sin\theta_W \, \left(g_\alpha^\nu g_\beta^\kappa - g_\alpha^\kappa g_\beta^\nu\right) \,, \\
\mathcal{V}_{W^-\mu}^{Z\kappa,\,W_4^-\gamma\delta} &= \frac{i}{2}\, g_w \cos\theta_W \, \left(g_\mu^\gamma g^{\kappa\delta} - g_\mu^\delta g^{\kappa\gamma}\right) \,, \\
\mathcal{V}_{W_4^-\alpha\beta}^{W^-\nu,\,Z\kappa} &= \frac{i}{2}\, g_w \cos\theta_W \, \left(g_\alpha^\nu g_\beta^\kappa - g_\alpha^\kappa g_\beta^\nu\right) \,.
\end{aligned} \tag{2.26}$$

Corresponding vertices exist for W^+ / W^- bosons. The decomposition of four particle vertices involving the Higgs boson introduces a new scalar pseudoparticle, which we denote by h_4. In order not to generate fake four particle vertices we define it not to be self-conjugate. The corresponding vertices read

$$\begin{aligned}
\mathcal{V}_h^{h,\,h,\,h} &\rightarrow \mathcal{V}_h^{h,\,h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{h,\,h} \,, \\
\mathcal{V}_h^{h,\,Z\mu,\,Z\nu} &\rightarrow \mathcal{V}_h^{h,\,h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{Z\mu,\,Z\nu} \,, \\
\mathcal{V}_h^{h,\,W^+\mu,\,W^-\nu} &\rightarrow \mathcal{V}_h^{h,\,h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{W^+\mu,\,W^-\nu} \,.
\end{aligned} \tag{2.27}$$

where the interactions of the h_4 pseudoparticle are defined by

$$\begin{aligned}
\mathcal{V}_h^{h,\,h_4} &= i \,, \\
\mathcal{V}_{h_4}^{h,\,h} &= i\, \frac{m_h^2}{v^2} \,, \\
\mathcal{V}_{h_4}^{Z\mu,\,Z\nu} &= -i\, \frac{g_w^2}{2\,\cos^2\theta_W}\, g^{\mu\nu} \,, \\
\mathcal{V}_{h_4}^{W^+\mu,\,W^-\nu} &= -i\, \frac{g_w^2}{2}\, g^{\mu\nu} \,,
\end{aligned} \tag{2.28}$$

and where we have introduced the scalar "propagator" of the h_4 pseudoparticle

$$P_{h_4} = i \,. \tag{2.29}$$

JHEP12(2008)039

$\mu\nu, I\bar{J} \overset{g_4}{\cdots\cdots} \kappa\lambda, H\bar{G}$	$=$	$-i\,\delta_I^H \delta_{\bar{J}}^{\bar{G}}\, D_{\mu\nu}^{\kappa\lambda}$	$\mu\nu \overset{Z_4}{\cdots\cdots} \kappa\lambda$	$=$	$-i\, D_{\mu\nu}^{\kappa\lambda}$
$\overset{h_4}{\text{---------}}$	$=$	i	$\mu\nu \overset{W_4^{\pm}}{\cdots\cdots} \kappa\lambda$	$=$	$i\, D_{\mu\nu}^{\kappa\lambda}$

Table 1: Standard Model propagators for auxiliary particles introduced in the vertex decomposition. Note that the $1/N_C$-term arising in eq. (2.21) is obsolete for the g_4 pseudogluon propagator because the pseudogluon does not couple to quarks.

Since all remaining vertices in the Standard Model are three point vertices, the vertex decomposition is hereby complete. The additional Standard Model propagators and vertices arising from this decomposition are summarised in tables 1 and 2, respectively.

2.5 Prefactors of diagrams with external fermions

When calculating currents with an arbitrary number of possibly indistinguishable external fermions, we have to take into account, that each Feynman diagram contains a prefactor

$$\mathcal{S} = (-1)^{\,\mathrm{S}_f(\sigma_1,\ldots,\sigma_n)}\,, \tag{2.30}$$

according to the number of fermion permutations S_f in the external particle assignment $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$. To be used in the context of a recursive computation, this prefactor must be defined on a local basis in order to avoid the proliferation of information on different $\vec{\sigma}$. It is then sufficient to note that eq. (2.30) holds at the level of interaction vertices. More precisely we can define the local prefactor $\mathcal{S}\,(\pi_1, \pi_2)$ of eq. (2.6) as

$$\mathcal{S}\,(\pi_1, \pi_2) = (-1)^{\,\mathrm{S}_f(\pi_1,\pi_2)}\,. \tag{2.31}$$

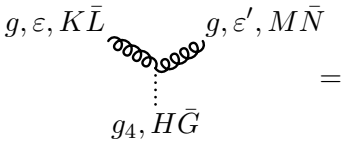
Here $\mathrm{S}_f\,(\pi_1, \pi_2)$ counts the number of fermion permutations that is needed to restore a predefined, for example ascending index ordering when combining the sets π_1 and π_2 into the set $\pi = \pi_1 \oplus \pi_2$. Upon iterating this procedure, we obtain the correct relative prefactors $\mathcal{S}$ for each diagram.

3. Matrix element generation in Comix

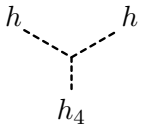
The general formulae to recursively compute tree-level amplitudes have been stated in section 2. Here we briefly explain, which conventions are used to define the external particle currents and internal Lorentz structures. We also elaborate on how to organise the computation and how to reduce the effective computation time per phase space point by a multi-threaded structure of the implementation.

We employ the spinor basis introduced in ref. [44]. Accordingly, the γ-matrices are taken in the Weyl representation. The main advantage of this representation is that spinors

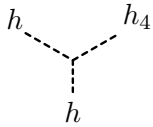
JHEP12(2008)039



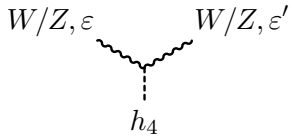
$$= \quad i\,\frac{g_s}{\sqrt{2}}\left[\,\delta^K_H\delta^{\bar{L}M}\delta^{\bar{N}}_{\bar{G}} - \delta^M_H\delta^{K\bar{N}}\delta^{\bar{L}}_{\bar{G}}\,\right]\,\mathrm{VVT}\left(\varepsilon,\varepsilon'\right)$$



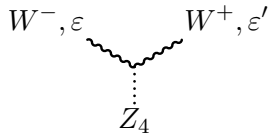
$$= \quad i\,\frac{m_h^2}{v^2}$$



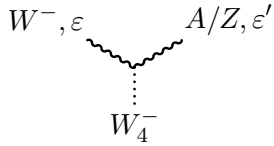
$$= \quad i$$



$$= \quad -i\,\frac{g_w^2}{2\,\lambda^2_{W/Z}}\,\mathrm{VVS}\left(\varepsilon,\varepsilon'\right) \qquad \text{where} \qquad \begin{aligned}\lambda_W &= 1\\ \lambda_Z &= \cos\theta_W\end{aligned}$$



$$= \quad i\,g_w\,\mathrm{VVT}\left(\varepsilon,\varepsilon'\right)$$



$$= \quad i\,g_w\,\kappa_{A/Z}\,\mathrm{VVT}\left(\varepsilon,\varepsilon'\right) \qquad \text{where} \qquad \begin{aligned}\kappa_A &= \sin\theta_W\\ \kappa_Z &= \cos\theta_W\end{aligned}$$

$$\mathrm{VVS}\left(\varepsilon,\varepsilon'\right) = \varepsilon^\mu\varepsilon'_\mu\ , \qquad \mathrm{VVT}^{\,\mu\nu}\left(\varepsilon,\varepsilon'\right) = \tfrac{1}{2}\left(g^{\mu\lambda}g^{\nu\kappa} - g^{\mu\kappa}g^{\nu\lambda}\right)\varepsilon_\lambda\varepsilon'_\kappa$$

Table 2: Standard Model vertices arising from the vertex decomposition and replacing the four particle vertices. In this context ε and ε' denote arbitrary incoming vector currents. Note that due to the antisymmetry of $\mathrm{VVT}^{\,\mu\nu}$, we can make the replacement $D^{\mu\nu}_{\alpha\beta}\,\mathrm{VVT}^{\,\alpha\beta} = 2\,\mathrm{VVT}^{\,\mu\nu}$, which leads to a slight decrease in computation time.

for massless particles are described through two nonzero components only. This fact greatly alleviates their construction as well as the evaluation of vertices. Polarisation vectors for external vector bosons are constructed according to ref. [45]. As pointed out in section 2, within the Standard Model tensor particles never occur as external states, such that there is no need to explicitly construct polarisation tensors.

The algorithms presented in this paper are intended to be used for large multiplicity matrix element calculations. In this context, it is often useful to sample over helicities of external particles in a Monte Carlo fashion. However, this introduces additional degrees of freedom and leads to a slower convergence of the integral. Furthermore when taking eq. (2.6) serious, we note that for helicity-summed ME's, it is possible to reuse currents to compute amplitudes with different configurations. Namely if the helicities of external particles assigned to a particular current do not change, it does not need to be recomputed. This leads to a significant decrease in evaluation time for the helicity summed ME's compared

JHEP12(2008)039

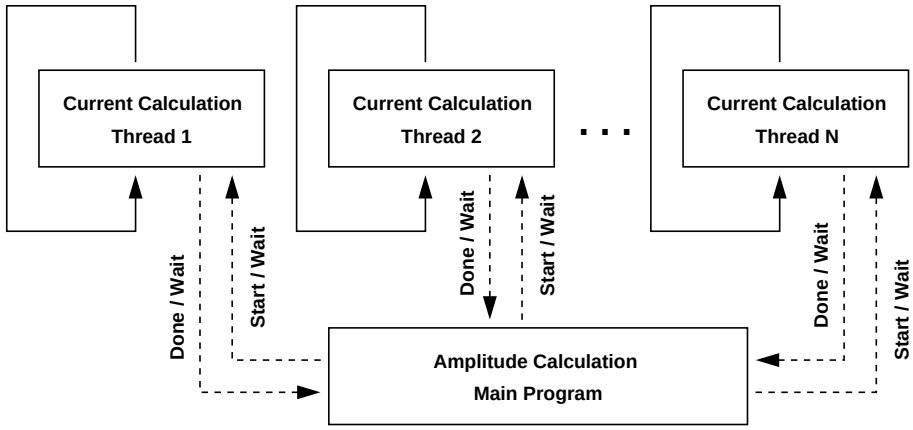


Figure 1: Structure of the multi-threaded implementation for matrix element computation in Comix. The number of threads N is variable and depends on the number of available processors. The main program communicates start and wait signals to the calculator threads, while those communicate done and wait signals to the main program. Details are explained in the text.

to the naive method of computing the full amplitude afresh for different configurations. A corresponding comparison can be found in section 5. The default choice in Comix is helicity summation. To allow computations for very large multiplicities, however, helicity sampling can be enabled as an option.

The effective computation time per phase space point can be further reduced by a multi-threaded implementation of eq. (2.6). Figure 1 shows the basic structure of this algorithm. The main advantage of eq. (2.6) is, that in order to compute a current that depends on n external particles, it is sufficient to know all subcurrents that depend on $m < n$ external particles. This leads to a straightforward multi-threading algorithm.

- Create N threads at program startup with the following properties
  1. The thread waits for the main program to signal the start of a computation. It then signals the main program to wait.
  2. It takes a number n and computes a block of currents depending on n external particles using subcurrents depending on $m < n$ external particles. If $n = 1$, it computes external polarisation vectors and spinors.
  3. It signals the main program that the calculation is done and returns to step 1.
- For each phase space point, employ the following algorithm in the main program
  1. Start with $n = 1$.
  2. Split the number of currents that depend on n external particles into N blocks. Communicate n and one block to each calculator thread.
  3. Signal the threads to start their computation. Wait for all threads to signal completion.

JHEP12(2008)039

4. Let $n \to n+1$ and return to step 2 if further currents need to be computed.

The efficiency of this algorithm solely depends on an efficient thread library. The overhead with a modern POSIX threading is about 10% of the total computational cost. This, however, is not of any concern considering that the employment of multiple CPU's reduces the computation time roughly proportional to the increase in processor usage.

4. Integration techniques in Comix

In this section we present two new methods for integrating over the phase space. Both of them are designed to cope especially with large numbers of outgoing particles. The first method is a fully general approach and makes use of the standard multi-channel technique [46] in a recursive fashion, i.e. the phase space sampling fits the method of generating the corresponding matrix element. The second method is designed for QCD and QCD-associated processes and employs the phase space generator Haag [38] in conjunction with a new prescription for coupling colour and momentum sampling and the multi-channel technique.

4.1 Recursive algorithm for phase space integration

One of the most efficient general approaches to sample the phase space of multi-particle processes is, to employ a multi-channel method according to ref. [46] with each of the single channels corresponding to the pole structure of a certain Feynman diagram. However, for large numbers of diagrams this is clearly not the method of choice. In the following we will therefore focus on the recursive relations for phase space generation proposed in ref. [37]. We construct a separate multi-channel for each possible subamplitude on the flight according to the propagator structure and use Vegas [47] to optimise the integration over propagator masses and polar angles in decays. The obvious drawback of this procedure is evident: It relies heavily on the assumption that the matrix element factorises according to its propagator structure. However, it is a generalisable way to tame the rather factorial growth in the number of phase space channels encountered in conventional approaches [21 – 26]. If we take the prescription serious, we can factorise the full phase space weight such that it can be computed in a recursive fashion corresponding to how the matrix element is evaluated. It turns out that this procedure gives an excellent performance, cf. section 5.

4.1.1 Brief review of phase space factorisation

In the following we consider a $2 \to n$ scattering process and denote incoming particles by a and b and outgoing particles by $1 \ldots n$. The corresponding n-particle differential phase space element reads

$$\mathrm{d}\Phi_n\left(a,b;1,\ldots,n\right) = \left[\prod_{i=1}^{n} \frac{\mathrm{d}^4 p_i}{(2\pi)^3}\,\delta\left(p_i^2 - m_i^2\right)\Theta\left(p_{i0}\right)\right]\,(2\pi)^4\,\delta^{(4)}\left(p_a + p_b - \sum_{i=1}^{n} p_i\right)\,, \tag{4.1}$$

JHEP12(2008)039

where m_i are the on-shell masses of outgoing particles. Following ref. [48], the full phase space may be factorised according to

$$\mathrm{d}\Phi_n\left(a,b;1,\ldots,n\right) = \mathrm{d}\Phi_{n-m+1}\left(a,b;\pi,m+1,\ldots,n\right)\,\frac{\mathrm{d}s_\pi}{2\pi}\,\mathrm{d}\Phi_m\left(\pi;1,\ldots,m\right)\ , \tag{4.2}$$

where $\pi = \{1,\ldots,m\}$ corresponds to a set of particle indices, similar to section 2. Generally in this section, Greek letters denote a subset of all possible indices. Overlined letters denote the missing subset, i.e. $\overline{\alpha} = \{a,b,1,\ldots,n\}\setminus\alpha$ for all $\alpha\subset\{a,b,1,\ldots,n\}$. Similarly, $\overline{\pi_1\pi_2} = (\{a,b,1,\ldots,n\}\setminus\pi_1)\setminus\pi_2$, etc. Equation (4.2) allows to decompose the complete phase space into building blocks corresponding to the t- and s-channel decay processes $T_{\alpha,b}^{\pi,\overline{\alpha b\pi}} = \mathrm{d}\Phi_2\left(\alpha,b;\pi,\overline{\alpha b\pi}\right)$ and $S_\pi^{\rho,\pi\setminus\rho} = \mathrm{d}\Phi_2\left(\pi;\rho,\pi\setminus\rho\right)$ and the s-channel production process $D_{\alpha,b}$, cf. figure 2. We refer to these objects as phase space vertices, while the integral $P_\pi = \mathrm{d}s_\pi/2\pi$, introduced in eq. (4.2), will be called a phase space propagator. We use the same notation as for the propagators in section 2 to highlight the close correspondence between matrix element computation and phase space generation. In the algorithm presented here, only timelike propagators are employed.

The phase space vertices are used differently in the case of weight calculation and phase space generation. Consider the t-channel decay. If a phase space point is to be generated, the new final state momenta p_π and $p_{\overline{\alpha b\pi}}$ are determined from the known initial state momenta p_α and p_b. If a weight needs to be computed, the new weight $w_\alpha^{(b)}$ is determined from the vertex weight and the input weights w_π and $w_{\overline{\alpha b\pi}}$. The corresponding situations are depicted in figures 2 and 3, respectively. The basic building blocks of phase space integration are summarised as follows

$$\begin{aligned}
P_\pi &= \begin{cases} 1 & \text{if } \pi \text{ or } \bar{\pi} \text{ external} \\ \dfrac{\mathrm{d}s_\pi}{2\pi} & \text{else} \end{cases}\ , \\
S_\pi^{\rho,\pi\setminus\rho} &= \frac{\lambda\left(s_\pi,s_\rho,s_{\pi\setminus\rho}\right)}{16\pi^2\,2\,s_\pi}\,\mathrm{d}\cos\theta_\rho\,\mathrm{d}\phi_\rho\ , \\
T_{\alpha,b}^{\pi,\overline{\alpha b\pi}} &= \frac{\lambda\left(s_{\alpha b},s_\pi,s_{\overline{\alpha b\pi}}\right)}{16\pi^2\,2s_{\alpha b}}\,\mathrm{d}\cos\theta_\pi\,\mathrm{d}\phi_\pi\ , \\
D_{\alpha,b} &= (2\pi)^4\,\mathrm{d}^4p_{\overline{\alpha b}}\,\delta^{(4)}\left(p_\alpha+p_b-p_{\overline{\alpha b}}\right)\ .
\end{aligned} \tag{4.3}$$

Here we have introduced the triangular function

$$\lambda\left(s_a,s_b,s_c\right) = \sqrt{\left(s_a-s_b-s_c\right)^2-4s_bs_c} \tag{4.4}$$

Note that even since α might correspond to an off-shell internal particle, b always indicates a fixed external incoming particle. This is essential in all further considerations and allows reusing weight factors in the Monte Carlo integration, just as currents are reused in the matrix element computation. The functions corresponding to $S_\pi^{\rho,\pi\setminus\rho}$ and $T_\alpha^{\pi,\overline{\alpha b\pi}}$ are in fact identical, since they represent a solid angle integration. In practice however we choose the different sampling strategies proposed in ref. [37]. The s-channel production vertex $D_{\alpha,b}$ is only needed for bookkeeping, since it corresponds to overall momentum conservation and the associated overall weight factor $(2\pi)^4$.

JHEP12(2008)039

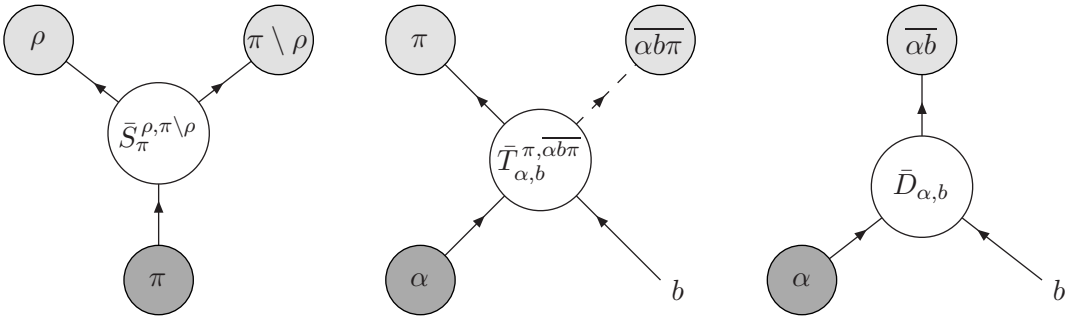


Figure 2: Basic vertices for phase space generation. Grey blobs correspond to eventually off mass-shell particles. Dark blobs denote known momenta, light blobs unknown momenta. Arrows indicate the momentum flow, i.e. the order in which unknown momenta are determined from known ones. The $\bar{D}$-vertex corresponds to overall momentum conservation.

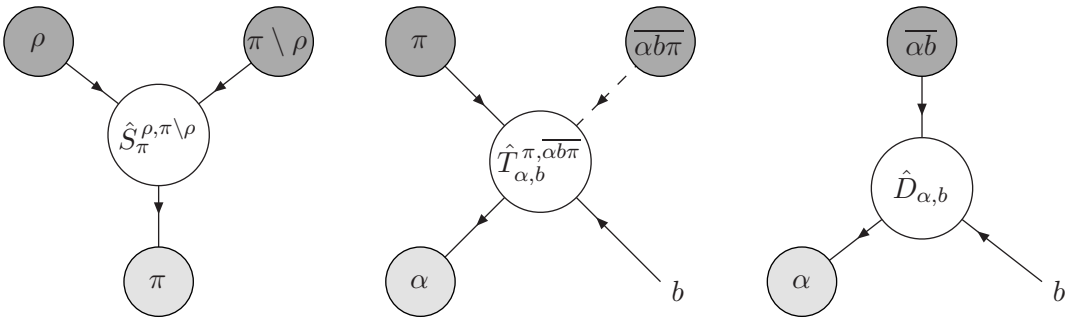


Figure 3: Basic decay vertices for weight calculation. Dark blobs denote potentially nontrivial known weights, light blobs weights to be determined. Arrows indicate the weight flow, i.e. the order in which unknown weights are determined from known ones. The $\hat{D}$-vertex corresponds to overall momentum conservation.

4.1.2 A simple example

We illustrate in this section how a recursive phase space generator for the process $q\bar{q} \to e^+e^-g$ can be constructed, based on the diagrammatic structure of the integrand. Figure 4 depicts the translation of the corresponding Feynman diagrams into related building blocks of the phase space. The standard procedure to define an integrator consists of constructing one integration channel per line of figure 4 and joining these channels in a multi-channel. Because it is based on full diagrams, this strategy cannot be implemented in a recursive fashion and we have to modify it. Consider first, which tasks have to be performed for each phase space point.

To generate momenta, one starts with the s-channel propagator P_{23}. Then, depending on what yields the better performance, the t-channel decay $T_{a,b}^{1,23}$ or $T_{a,b}^{23,1}$ and finally the s-channel decay $S_{23}^{2,3}$ are constructed, leading to the final state momenta $p_1 \dots p_3$. Note again that D-type vertices are just for bookkeeping at this point. They simply imply overall momentum conservation. When computing the phase space weight, the order of

JHEP12(2008)039

treating vertices can be reversed because all corresponding momenta are known. Therefore the weights for the decay $S_{23}^{2,3}$ and the propagator P_{23} are computed first, followed by the weights for $T_{a,b}^{1,23}$ and $T_{a,b}^{23,1}$. It is obvious that the weights $\hat{P}_{23}$ and $\hat{S}_{23}^{2,3}$ are unique and therefore have to be computed only once, although arising in both lines of figure 4. We refer to this feature as the "weight flow", which is directed from the final state particles and the right beam, particle b, towards the left beam, particle a. This generalises to arbitrary processes, provided that the right beam particle is kept fix, i.e. t-channel indices always combine only a and external indices, as indicated in figures 2 and 3. It allows to compute the full phase space weight recursively, in a manner similar to eq. (2.6), which implies in particular, that at most the same growth is induced in the matrix element and the phase space weight computation.

Let us illustrate this new procedure using the above example. The difference with respect to the standard approach is how multi-channels are defined. Following the weight flow, in the first step of the recursion we construct a multi-channel for the phase space element $\mathrm{d}\Phi_2\left(\{23\};2,3\right)$. Of course, since particles 2 and 3 are external, this multi-channel consists of one single channel only, which is the s-channel decay $S_{23}^{2,3}$. It has therefore no additional parameters. In the second step we construct a multi-channel for the full phase space $\mathrm{d}\Phi_3\left(a,b;1,2,3\right)$, which receives contributions from the two t-channels $T_{a,b}^{1,23}$ and $T_{a,b}^{23,1}$. Each of them can be assigned a multi-channel weight w, which eventually yields the overall weight

$$(2\pi)^4\, F^{-1}\left[\frac{w_{a,b}^{1,23} F\left[\hat{T}_{a,b}^{1,23}\hat{P}_{23}\hat{S}_{23}^{2,3}\right] + w_{a,b}^{23,1} F\left[\hat{T}_{a,b}^{23,1}\hat{P}_{23}\hat{S}_{23}^{2,3}\right]}{w_{a,b}^{1,23} + w_{a,b}^{23,1}}\right], \tag{4.5}$$

Here $\hat{P}$ denotes the propagator weight, and F is a generalised mean function, see next section. The overall factor $(2\pi)^4$ arises from the D-type vertices.

We are left with the task to determine the necessary building blocks of the phase space using information from the matrix element. This turns out to be extremely simple since we now in fact have a phase space weight recursion of the form of eq. (2.6) (cf. eqs. (4.7) and (4.8)), where multi-channels are associated with intermediate s/t-channel propagators. Therefore each intermediate current in the matrix element implies a separate multi-channel in the corresponding integrator and each vertex implies a decay vertex associated with a single channel. Because of this correspondence the default sampling strategy for s-channel propagator masses can be chosen according to the type of intermediate particle, i.e. a Breit-Wigner-like distribution for massive, unstable particles and a $1/s^{\alpha}$-type distribution for massless particles. These distributions as well as the polar angle distributions in decays (cf. eq. (4.3)) are further refined during the integration using Vegas.

4.1.3 Formulation of the recursive algorithm

In this subsection we derive the general algorithm for the recursive phase space integrator. We employ the notation of section 4.1.1. Recursive relations for phase space integration in

JHEP12(2008)039

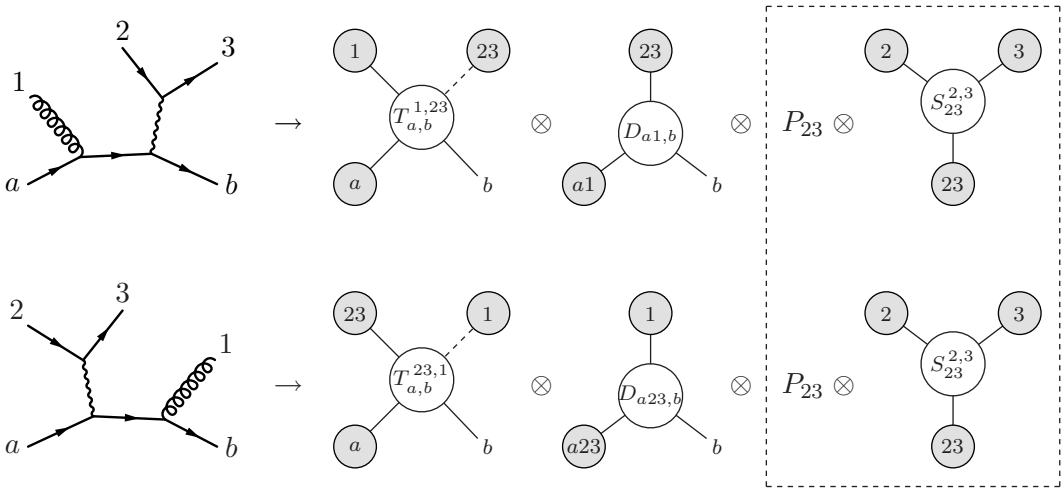


Figure 4: Correspondence between Feynman diagrams and building blocks of the phase space for the process $q\bar{q} \to e^+e^-g$. The terms in the dashed box arise from both diagrams and have to be evaluated only once when computing the phase space weight.

terms of the quantities introduced ibidem can be defined through

$$\begin{aligned} \mathrm{d}\Phi_S(\pi) &= S^{\pi_1,\pi_2}_{\pi}\, P_{\pi_1}\,\mathrm{d}\Phi_S(\pi_1)\, P_{\pi_2}\,\mathrm{d}\Phi_S(\pi_2)\,\Big|_{(\pi_1,\pi_2)\in\mathcal{OP}(\pi)}\,, \\ \mathrm{d}\Phi^{(b)}_T(\alpha) &= T^{\pi_1,\pi_2}_{\alpha,b}\, P_{\pi_1}\,\mathrm{d}\Phi_S(\pi_1)\, P_{\pi_2}\,\mathrm{d}\Phi^{(b)}_T(\alpha\pi_1)\,\Big|_{(\pi_1,\pi_2)\in\mathcal{OP}(\overline{\alpha b})} + D_{\alpha,b}\,\mathrm{d}\Phi_S\left(\overline{\alpha b}\right)\,. \end{aligned} \tag{4.6}$$

The above equations correspond to selecting one possible splitting of the multi-index π or $\overline{\alpha b}$ per phase space point. We can improve the integration procedure by forming an average over all possible splittings in the spirit of a multi-channel. Let F be a generalised mean function. We can then use the F-mean to define

$$\begin{aligned} \mathrm{d}\Phi_S(\pi) = F^{-1}\Bigg[&\left(\sum_{(\pi_1,\pi_2)\in\mathcal{OP}(\pi)} \omega^{\pi_1,\pi_2}_{\pi}\right)^{-1} \\ &\times \sum_{(\pi_1,\pi_2)\in\mathcal{OP}(\pi)} \omega^{\pi_1,\pi_2}_{\pi}\, F\Big[\, S^{\pi_1,\pi_2}_{\pi}\, P_{\pi_1}\,\mathrm{d}\Phi_S(\pi_1)\, P_{\pi_2}\,\mathrm{d}\Phi_S(\pi_2)\,\Big]\Bigg]\,, \end{aligned} \tag{4.7}$$

$$\begin{aligned} \mathrm{d}\Phi^{(b)}_T(\alpha) = F^{-1}\Bigg[&\left(\omega_{\alpha,b} + \sum_{(\pi_1,\pi_2)\in\mathcal{OP}(\overline{\alpha b})} \omega^{\pi_1,\alpha\pi_1}_{\alpha}\right)^{-1} \\ &\Bigg(\omega_{\alpha,b}\, F\Big[\, D_{\alpha,b}\,\mathrm{d}\Phi_S(\overline{\alpha b})\,\Big] + \sum_{(\pi_1,\pi_2)\in\mathcal{OP}(\overline{\alpha b})} \omega^{\pi_1,\alpha\pi_1}_{\alpha} \\ &\times F\Big[\, T^{\pi_1,\pi_2}_{\alpha,b}\, P_{\pi_1}\,\mathrm{d}\Phi_S(\pi_1)\, P_{\pi_2}\,\mathrm{d}\Phi^{(b)}_T(\alpha\pi_1)\,\Big]\Bigg)\Bigg]\,. \end{aligned} \tag{4.8}$$

JHEP12(2008)039

In this context we define the one- and no-particle phase space

$$\begin{aligned} \mathrm{d}\Phi\left(i\right) &= 1\ , \\ \mathrm{d}\Phi\left(\emptyset\right) &= 0\ . \end{aligned} \tag{4.9}$$

The function ω corresponds to a vertex-specific weight which may be adapted to optimise the integration procedure, see ref. [46]. The second sums run over all possible S- and T-type vertices which have a correspondence in the matrix element. The full differential phase space element is given by

$$\mathrm{d}\Phi_n\left(a,b;1,\ldots,n\right) = \mathrm{d}\Phi_T\left(a\right)\ . \tag{4.10}$$

Note that eqs. (4.7) and (4.8) in the form stated above are *not* suited to generate the sequence of final state momenta. To do so one rather has to employ the following algorithm, which corresponds to a reversion of the recursion and respects the weight factors w introduced above.

- From the set of possible vertices connecting currents in the matrix element, choose a sequence connecting all external particles in the following way:

  1. Start with the set of indices $\pi = \{b, 1, \ldots, n\}$, corresponding to the unique external current of index a.
  2. From the set of possible phase space vertices connecting to π select one according to an on the flight constructed multi-channel employing the weights w.[2] If π is a single index, stop the recursion.
  3. According to the selected vertex, split π into the subsets π_1 and π_2. Repeat step 2 for these subsets.

- Fore each vertex, make use of the fact that π is equivalent to $\overline{\pi}$ and adjust the indices in an appropriate way for momentum generation. That is if any π contains b and other indices, replace π by $\overline{\pi}$.

- Order $\bar{T}$-type vertices ascending and $\bar{S}$-type vertices descending in the number of external indices connected to initial states.

- Generate the corresponding momenta starting with $\bar{T}$-type vertices.

Even though T-type vertices depend on b, since b is fixed throughout the computation of one phase space point we obtain no expressions depending on more than two particle indices. This induces the same growth of computational complexity in both the hard matrix elements and the phase space and makes the above algorithm well suited for integration of processes with large final state multiplicity. In the following we refer to it as the Recursive Phase space Generator (RPG).

[2] Note that in this context weights have to be normalised to unity on the flight.

JHEP12(2008)039

4.1.4 Implementation details

Since the phase space weight computation, eq. (4.7) obeys a recursion similar to those of the matrix element calculation, eq. (2.6), it is straightforward to implement this weight computation into a numerical program along the lines of section 3. The same techniques described for the multi-threading of matrix element calculations can be implemented for the phase space weight. In the multi-threaded version of Comix, this weight is computed in parallel to the matrix element, which further reduces the net computation time if enough resources are available.

4.2 Colour sampling

For QCD and QCD associated processes with a large number of external legs, it becomes unfeasible to compute colour-summed scattering amplitudes. Instead the better strategy is to sample over external colour assignments in a given representation of SU(3). According to eqs. (2.8)–(2.10), this selects a set of colour-ordered amplitudes which contribute to the corresponding point in colour space. This set is typically strongly reduced compared to the full set of partial amplitudes. The issue has been studied in ref. [42] for the fundamental representation decomposition, the adjoint representation decomposition and the colour flow decomposition, which has been presented therein. The conclusion is that the colour flow decomposition is the method best suited for sampling over colour assignments if the number of external partons is large, i.e. it provides the slowest growth in the average number of partial amplitudes per non-vanishing colour assignment. Also it has been exemplified for recursive calculations in ref. [34], that the colour flow decomposition is advantageous, since no computational intensive matrix multiplications have to be performed. We therefore employ this prescription throughout Comix.

In the following we focus on an n-gluon scattering process. However, the presented ideas and algorithms are straightforward to generalise for arbitrary sets of colour octet objects, such as e.g. quark-antiquark pairs. In the colour flow decomposition each external gluon is labeled by a colour index i and an anti-colour index $\bar{\jmath}$. The colour assignment for an n-gluon scattering is thus given by selecting each index $i_1, \dots i_n$ and $\bar{\jmath}_1, \dots \bar{\jmath}_n$ out of three values (R, G, B) and $\left(\bar{R}, \bar{G}, \bar{B}\right)$.

4.2.1 Determination of colour flows from colour assignments

A specific colour flow, and thus an ordering in the sense of a colour-ordered amplitude, is specified by a permutation

$$\vec{\sigma} = (1, \sigma_2, \sigma_3, \dots, \sigma_n) \in S_{n-1} \tag{4.11}$$

of external gluon indices. This colour flow contributes to a colour assignment, if

$$\delta^{i_1 \bar{\jmath}_{\sigma_2}} \delta^{i_{\sigma_2} \bar{\jmath}_{\sigma_3}} \dots \delta^{i_{\sigma_n} \bar{\jmath}_1} = 1 \; . \tag{4.12}$$

It is thus easy to construct an algorithm which determines all valid colour flows from a given colour assignment.

JHEP12(2008)039

1. Set the first gluon index to $\sigma_1 = 1$. Let $k = 2$.
2. Select one of the remaining gluon indices to be σ_k, such that $i_{\sigma_{k-1}} = \bar{\jmath}_{\sigma_k}$. If this is possible, let $k \to k+1$. Otherwise let $k \to k-1$, then repeat this step selecting a different σ_k.
3. If $k = n+1$ and $i_{\sigma_n} = \bar{\jmath}_{\sigma_1}$, a valid flow has been found.
   Otherwise continue with step 2.

By systematically selecting through all possible σ_k in step 2 all valid colour flows are determined.

4.2.2 Selection of colour assignments

The simplest way of choosing a colour assignment is accomplished by randomly selecting the $2n$ colours for the i- and $\bar{\jmath}$-indices. Each colour is chosen with an equal probability, leading to a weight of 3^{2n}. However, only a small fraction of those assignments will have at least one valid colour flow. A trivial (but not sufficient) condition for non-vanishing amplitudes is, that the number of i-indices carrying the colour R (G,B) must be equal to the number of $\bar{\jmath}$-indices carrying the corresponding anticolour.

We thus propose a more efficient way to determine colour configurations.

1. The n i-indices are selected randomly in (R, G, B).
2. A permutation $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ of n particles is selected randomly with a uniform weight. The anticolours of the $\bar{\jmath}$-indices are then given by $\bar{\jmath}_k = \overline{i_{\sigma_k}}$, for $k = 1, \ldots, n$.
3. Each colour assignment is weighted by
$$w \;=\; 3^n \, \frac{n!}{n_R!\, n_G!\, n_B!} \;, \tag{4.13}$$
   where n_R, n_G and n_B are the multiplicities of i-indices carrying the colours R, G and B, respectively.

Clearly, assignments generated by this algorithm will always fulfil the trivial condition mentioned above. Moreover, the weight is roughly proportional to the number of possible colour flows and thus already corresponds to some extent to the expected cross section for this colour configuration.

4.2.3 A simple example

To illustrate the colour sampling in the colour flow decomposition we consider a five gluon scattering process. The random selection of a colour configuration using the improved algorithm may return the following i-indices:

$$i_1 = R\,, \quad i_2 = R\,, \quad i_3 = G\,, \quad i_4 = G\,, \quad i_5 = B\,. \tag{4.14}$$

The $\bar{\jmath}$-indices are fixed by a randomly chosen permutation, say $\vec{\sigma} = (4, 1, 2, 5, 3)$:

$$\bar{\jmath}_1 = \bar{G}\,, \quad \bar{\jmath}_2 = \bar{R}\,, \quad \bar{\jmath}_3 = \bar{R}\,, \quad \bar{\jmath}_4 = \bar{B}\,, \quad \bar{\jmath}_1 = \bar{G}\,. \tag{4.15}$$

For this assignment the only colour flow that satisfies eq. (4.12) is given by the permutation $\vec{\sigma} = (1, 4, 5, 3, 2)$.

JHEP12(2008)039

4.3 Combined colour-momentum integration techniques

Generally the peaking behaviour of the colour-sampled differential cross section is rather complex within the phase space and strongly different for different colour assignments. The idea must thus be to construct integrators specific for a given colour assignment, based on the knowledge of contributing partial amplitudes. One can for example think of a variant of the algorithm described in section 4.1, where the basic building blocks of the phase space are either available or not, depending whether there is a corresponding non-vanishing coloured current present in the matrix element. However, in practice this choice does not lead to any significant improvement of the integration behaviour of the RPG and we thus refrain from promoting this method.[3] Instead we present a second type of integrator, dedicated to be used with QCD and QCD associated processes, which is based on the HAAG algorithm [38]. As before we concentrate on purely gluonic processes.

4.3.1 Integration of partial amplitudes and colour configurations

As a basic building block we use the HAAG-integrator, which generates momenta distributed proportional to a QCD antenna function [38],

$$A_n(p_1, p_2, \ldots, p_n) = \frac{1}{(p_1p_2)(p_2p_3)\ldots(p_{n-1}p_n)(p_np_1)}. \tag{4.16}$$

Details on our implementation of the algorithm and improvements to the original version are given in ref. [20]. Single HAAG-channels provide efficient integrators for squared partial amplitudes associated with a given colour flow, both labeled by the same permutation $\vec{\sigma}$, eq. (4.11). For the HAAG-channel the permutation corresponds to the order of momenta in the antenna function. As for the RPG we again obtain a close correspondence between the matrix element and the phase space generation, now at the level of partial amplitudes.

The cross section for a single colour assignment is given by the squared sum of partial amplitudes associated with all valid colour flows. Ignoring the interferences between the amplitudes in the context of the phase space setup, a dedicated integrator can be constructed by combing the corresponding HAAG-channels for each flow in a multi-channel integrator. With growing number of external particles, however, one faces the following problem: Although the average number of contributing colour flows per colour assignment is relatively low in the colour flow decomposition, the maximal number grows factorially. Thus it quickly becomes impossible to store all data associated with the multi-channels defined above, i.e. the contributing HAAG-channels and the internal weights. The situation gets even worse if the sampling over all colour assignments is taken into account, because the number of possible assignments grows exponentially with the number of external particles. The solution to this is not to store any multi-channel parameters, but to generate the complete multi-channel on the flight.

A fast algorithm, as presented in section 4.2.1 to provide all colour flows from a colour assignment is essential for this approach: for a single phase space point one has to loop

[3] Note that this is not a statement about the integration behaviour of the RPG itself, but only about a possible coupling of colour and momentum sampling using the RPG.

JHEP12(2008)039

three times over the list of all colour flows (which cannot be stored as well due to the factorially growing maximal number of flows).

1. To determine the normalisation of weights within the multi-channel integrator.
2. To select a channel for generating a phase space point with a probability given by the relative weight α_k, and
3. To compute the multi-channel weight corresponding to this phase space point.

4.3.2 Optimisation techniques

The proposed integrator contains a number of parameters which can be adjusted or adapted to reduce the variance during integration. A multi-channel integrator dedicated to a specific colour assignment has the following degrees of freedom for optimisation:

- VEGAS grids to refine individual HAAG-channels,
- Relative weights α_k in the multi-channel generator,

The sheer multiplicity of different channels and on-the-flight construction of the integrator forbids an individual adaptation of all parameters. However, their number can be greatly reduced by making use of the symmetry among different HAAG-channels w.r.t. to permutations of the final state. All channels with the same relative positions of the initial state momenta within the antenna can be determined from each other by a permutation of final state momenta. This prevents the number of structurally different phase space channels from growing factorially with the number of particles and induces a linear growth only. Taking into account that the same symmetry holds for the partial amplitudes justifies to reuse the optimisation parameters among all channels of one kind. For later reference we label different types of HAAG-channels (and respective partial amplitudes) by the number of final state momenta between the first and the second incoming momentum within a certain antenna.

We achieve the best integration efficiency by performing the optimisation of the free parameters prior to the actual integration: The VEGAS grids of the HAAG-channels are adapted individually by integrating corresponding single squared partial amplitudes over the allowed phase space. Using the above mentioned symmetry this has to be done only for one channel of each kind.[4] This technique not only speeds up the optimisation, it also provides a much cleaner environment for the adaptation of the VEGAS grids. At this stage a summation over helicities is performed. Cross sections σ_t, given by the integration result from the channel of type t, are stored.

In the actual integration run no further optimisation is performed. The channels are used as they emerged from the above procedure, including the VEGAS-grid and a parameter α_k, proportional to the cross section, σ_t, of the corresponding squared partial amplitude.

[4]During this step the full result can not be determined since potential interferences between partial amplitudes are ignored. However, it is sufficient for computing the leading $1/N_C$ limit for n gluon processes, using the fact that in the colour flow decomposition (as well as in the fundamental representation decomposition) interferences are always subleading.

JHEP12(2008)039

Best performance is achieved, if the colour assignment is selected with a probability proportional to the sum of cross sections of contributing squared partial amplitudes (as determined during the optimisation step), instead of the weight given by eq. (4.13). To do so, the total normalisation for the new weight must be determined summing over all colour assignments. For n-gluon processes this number is given by the following simple formula:

$$N = (n-2)!\, 3^n \sum_{i=0}^{n-2} \sigma_{\min(i,n-i-2)} \,, \tag{4.17}$$

where the $\sigma_{\min(i,n-i-2)}$ is the cross section of a squared partial amplitude of the type "$\min(i, n-i-2)$". The reweighting can be done by a simple hit-or-miss method.

For the integration run it is a matter of choice whether to sum or sample over helicities. All practical tests for up to the 11-gluon process favoured summation. Beyond that, however, it seems to become too costly to compute summed matrix elements, thus a sampling should be considered.

In the context of this work, we refer to the above algorithm as the Colour Sampling Integrator (CSI).

5. Results

In this section we present selected results generated with Comix. We focus on the special feature of this new generator, to be suitable in particular for computation of large multiplicity matrix elements. A detailed comparison of integration times, compared to a dedicated code using CSW vertex rules and the generator Amegic++ can be found in ref. [20].

5.1 Helicity summation vs. helicity sampling

Firstly we illustrate the effect of suitable matrix element generation in the helicity summed mode of Comix, see section 3. Computation times for helicity summed and helicity sampled matrix elements in pure gluonic processes are compared in table 3. The naive ratio between the two is the number of possible helicity assignments of the respective amplitude, $2^n - 2(n+1)$, with n the number of external gluons. This naive ratio corresponds to computing the amplitude afresh for each of the different helicity assignments. Employing the ideas presented in section 3, however we find that this value overestimates the real computational cost by up to a factor of ≈ 7. Obviously this statement is process dependent. The general feature, however is that there is a gain when computing helicity summed matrix elements. For the computation of cross sections this type of calculation might be preferred over the helicity sampled mode, especially when using the phase space integration methods of the previous section, which are not designed for helicity sampling.

5.2 Performance of the CSI and $2 \to n$ gluon benchmarks

In this subsection we present a comparison of gluon production cross sections to illustrate both the performance of the CSI and the efficiency of the matrix element generation. We

JHEP12(2008)039

start with a fixed centre-of-mass energy. The parameters are those of refs. [49, 42], i.e. $\alpha_S = 0.12$ and

$$p_{Ti} > 60 \text{ GeV} , \qquad |\eta_i| < 2 , \qquad \Delta R_{ij} > 0.7 , \tag{5.1}$$

for all final state gluons i and pairs of gluons i, j. Integration results are summarised in table 4. We find perfect agreement with the results in the literature and give new predictions for the processes $gg \to 11g$ and $gg \to 12g$. Results have been generated with the CSI, except for the $2 \to 11$ and $2 \to 12$ process, where RAMBO [50] has been employed. In order to examine the performance of the new phase space generator in a more realistic scenario, we investigate the same partonic processes at the LHC and employ the Tevatron Run II k_T algorithm [51][5] to define a cut on the multi-particle phase space. The respective results are summarised in table 5. We find that the CSI performs very well in both cases, even for large multiplicities, such that the respective cross sections can be computed with good precision.

Figures 5 and 6 show the convergence behaviour of the CSI for various gluon multiplicities. Since the computation of $2 \to 8$ and $2 \to 9$ gluon processes is quite cumbersome, it is worthwhile to switch to the helicity sampled mode in that case. Correspondingly we compare the performance of the CSI in helicity summed and helicity sampled mode in figure 6.

5.3 Performance of the RPG and comparison with other generators

We finally compare the performance of Comix with those of other programs. All results presented in this section were obtained with the RPG described in section 4.1. As references we use Amegic++ [24] and Alpgen [27]. The original setup for the comparison was established during the MC4LHC workshop [52]. For a comprehensive listing of results from all participating projects, see ibidem. Input parameters are given in table 6. Cross sections are summarised in tables 7, 8 and 10.

As pointed out in section 4.1, a drawback of the RPG is that it might not be able to adapt to certain peaks of the matrix element which correspond to specific diagrams. No significant disadvantage compared to other generators can however be observed. A measure for the efficiency of a phase space generator is given by the ratio of the average over the maximal weight $\langle w \rangle / w_{\max}$, i.e. the efficiency for generating events of unit weight using a hit-or-miss method. As discussed in ref. [53], the maximum weight and thus this ratio is a numerically rather unstable quantity, often determined by very rare events in the high tail of the weight distribution. In table 9 we therefore list the more stable quantity $\langle w \rangle / w^{\varepsilon}_{\max}$, where the reduced maximum weight $w^{\varepsilon}_{\max}$ is defined such that $1 - \langle \min(w, w^{\varepsilon}_{\max}) \rangle / \langle w \rangle = \varepsilon \ll 1$. It turns out that we achieve a reasonably good efficiency using the RPG, even for very large multiplicities. It can therefore be concluded that this phase space generator is an excellent approach to tame the factorial growth of phase space channels while still maintaining an a priori adaptation to the assumed peak structure of the integrand.

[5] Note that we replace $\Delta R^2_{ij} \to \cosh \Delta\eta_{ij} - \cos \Delta\phi_{ij}$ in order to match the Durham measure for final state clusterings.

JHEP12(2008)039

Process	Time [ms / pt]		Ratio	Gain
	sum	sample		
$gg \to 2g$	0.073	0.025	2.9	2.1
$gg \to 3g$	0.339	0.060	5.7	3.5
$gg \to 4g$	1.67	0.149	11	4.5
$gg \to 5g$	8.98	0.427	21	5.3
$gg \to 6g$	49.6	1.39	36	6.6
$gg \to 7g$	298	4.32	69	7.1
$gg \to 8g$	1990	13.6	146	6.9
$gg \to 9g$	13100	43.7	300	6.7
$gg \to 10g$	96000	138	695	5.9

Table 3: Computation time for multi-gluon scattering matrix elements sampled over colour configurations. Displayed times are averages for a single evaluation of the colour dressed BG recursion relation, when summing and sampling over helicity configurations, respectively. Additionally in the last column, labeled 'Gain' we give the inverse ratio of evaluation times multiplied by the naive ratio $2^n - 2(n+1)$, where n is the number of external gluons. Numbers were generated on a 2.80 GHz Pentium® 4 CPU.

gg → ng	Cross section [pb]				
n	8	9	10	11	12
$\sqrt{s}$ [GeV]	1500	2000	2500	3500	5000
Comix	0.755(3)	0.305(2)	0.101(7)	0.057(5)	0.026(1)
ref. [42]	0.70(4)	0.30(2)	0.097(6)		
ref. [49]	0.719(19)				

Table 4: Cross sections for multi-gluon scattering at the centre-of-mass energy $\sqrt{s}$, using the phase space cuts specified in eq. (5.1), compared to literature results. In parentheses the statistical error is stated in units of the last digit of the cross section.

gg → ng	Cross section [pb]			
n	7	8	9	10
Comix	2703(14)	407.0(36)	66.5(13)	15.2(26)

Table 5: Multi-gluon cross sections at the LHC with $\sqrt{d} \geq 20$ GeV and d defined as in ref. [51], except that $\Delta R_{ij}^2 \to \cosh \Delta\eta_{ij} - \cos \Delta\phi_{ij}$. In parentheses the statistical error is stated in units of the last digit of the cross section.

JHEP12(2008)039

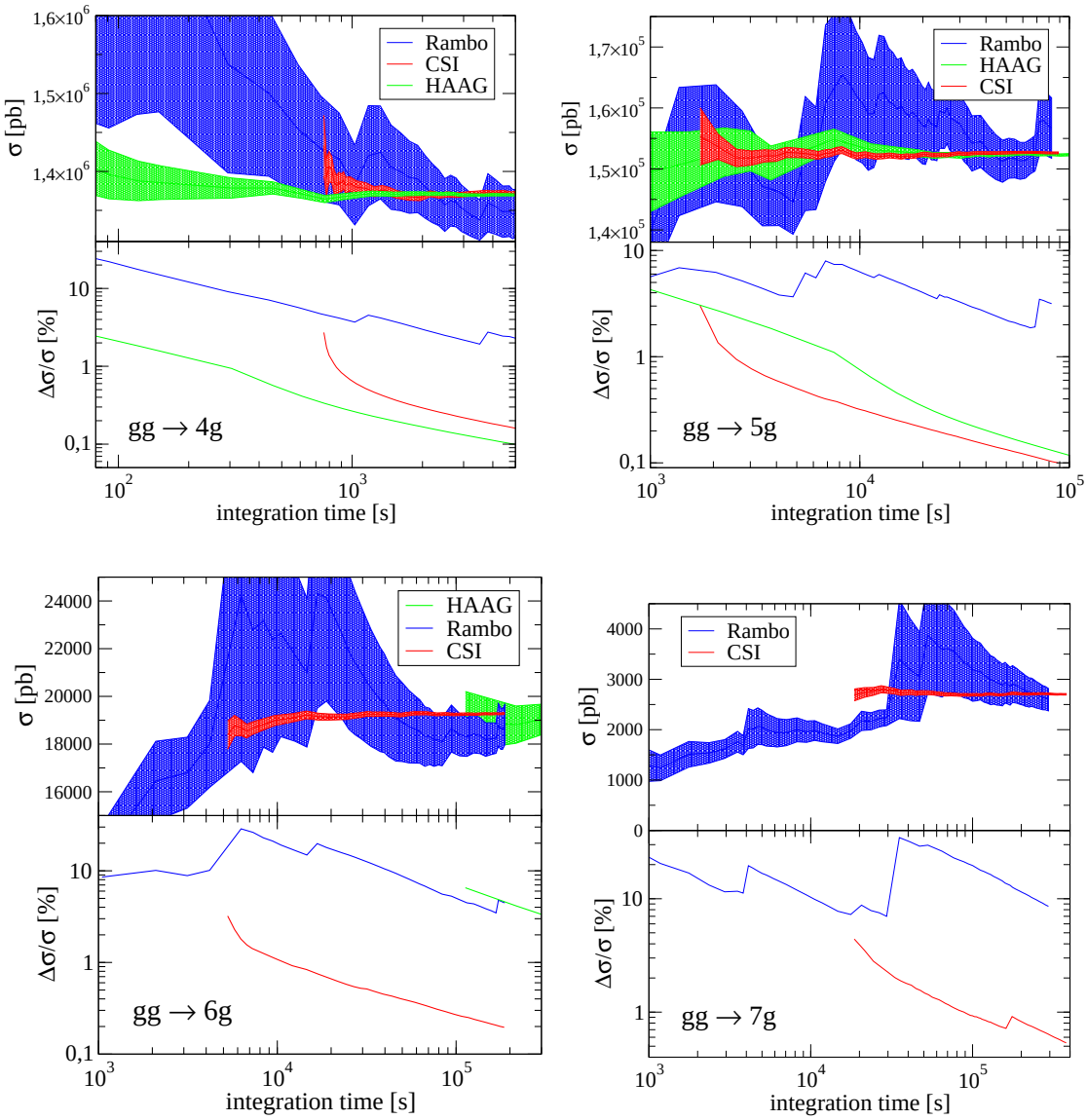


Figure 5: Overall integration performance for multi-gluon scattering. Upper panels display the Monte Carlo estimate of the cross section with the corresponding 1σ statistical error band as a function of the total integration time. Lower panels show the relative statistical error. HAAG denotes the phase space integrator described in ref. [20], applied on colour- and helicity-summed ME, generated using the CSW vertex rules. CSI denotes the integrator discussed in section 4.3.1, applied on colour-sampled and helicity-summed ME's, generated using the CDBG recursion. Results for RAMBO were generated using colour- and helicity-sampled ME's form the CDBG recursion. Calculations have been performed on a 2.66 GHz Xeon^{TM} CPU

6. Conclusions

We have presented the new matrix element generator COMIX, based on the colour dressed

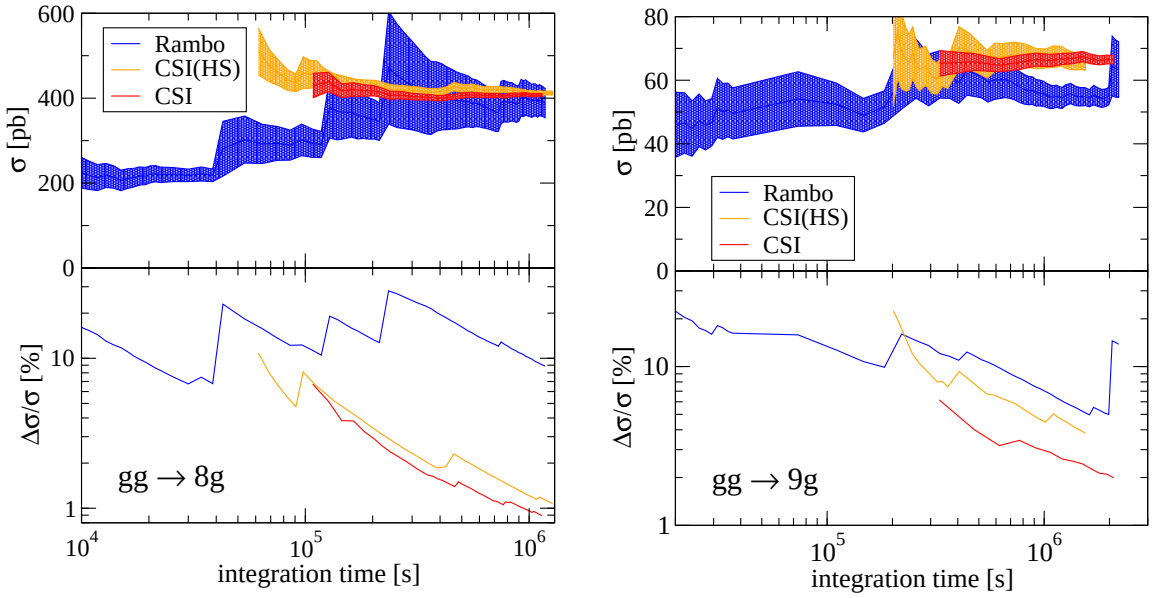


Figure 6: Overall integration performance for multi-gluon scattering, continued from figure 5. Additionally, for the CSI a sampling over helicity is considered, denoted by CSI(HS).

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_W	80.419 GeV
M_Z	91.188 GeV
m_H	120 GeV
CKM matrix	
V_{ud}, V_{cs}	0.975
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
Non-zero fermion masses (no evolution)	
m_b	4.7 GeV
m_t	174.3 GeV
m_τ	1.777 GeV
Widths (fixed width scheme)	
Γ_W	2.048 GeV
Γ_Z	2.446 GeV
Γ_H	3.7×10^{-3} GeV
Γ_t	1.508 GeV
Γ_τ	2.36×10^{-12} GeV
Cuts	
$p_{\perp,i}$	> 20 GeV
$\lvert\eta_i\rvert$	< 2.5
ΔR_{ij}	> 0.4
no cuts on particles of $m > 3$ GeV and ν_l	

Table 6: Parameters for the MC4LHC comparison setup.

Berends-Giele recursive relations and two new methods for phase space generation. We have analysed the performance of the new generator and compared the respective results to other ME generators. We find that the new algorithms perform very well and we obtain promising results for large multiplicity processes. COMIX can therefore be considered an

2(2008)039

σ [μb]	Number of jets						
jets	2	3	4	5	6	7	8
Comix	331.0(4)	22.72(6)	4.95(2)	1.232(4)	0.352(1)	0.1133(5)	0.0369(3)
ALPGEN	331.7(3)	22.49(7)	4.81(1)	1.176(9)	0.330(1)		
AMEGIC	331.0(4)	22.78(6)	4.98(1)	1.238(4)			
σ [μb]	Number of jets						
$b\bar{b}+$jets	0	1	2	3	4	5	6
Comix	471.2(5)	8.83(2)	1.813(8)	0.459(2)	0.150(1)	0.0531(5)	0.0205(4)
ALPGEN	470.6(6)	8.83(1)	1.822(9)	0.459(2)	0.150(2)	0.053(1)	0.0215(8)
AMEGIC	470.3(4)	8.84(2)	1.817(6)				
σ [pb]	Number of jets						
$t\bar{t}+$jets	0	1	2	3	4	5	6
Comix	754.8(8)	745(1)	518(1)	309.8(8)	170.4(7)	89.2(4)	44.4(4)
ALPGEN	755.4(8)	748(2)	518(2)	310.9(8)	170.9(5)	87.6(3)	45.1(8)
AMEGIC	754.4(3)	747(1)	520(1)				
σ [pb]	Number of jets						
$e^+\nu_e+$jets	0	1	2	3	4	5	6
Comix	5434(5)	1274(2)	465(1)	183.0(6)	77.5(3)	33.8(1)	14.7(1)
ALPGEN	5423(9)	1291(13)	465(2)	182.8(8)	75.7(8)	32.5(2)	13.9(2)
AMEGIC	5432(5)	1279(2)	466(2)	185.2(5)	77.3(4)		
σ [pb]	Number of jets						
$e^-\bar{\nu}_e+$jets	0	1	2	3	4	5	6
Comix	3911(4)	1011(2)	362(1)	137.1(3)	54.9(2)	22.4(1)	9.26(4)
ALPGEN	3904(6)	1013(2)	364(2)	136(1)	53.6(6)	21.6(2)	8.7(1)
AMEGIC	3903(4)	1012(2)	363(1)	137.6(3)	54.8(6)		
σ [pb]	Number of jets						
e^-e^++jets	0	1	2	3	4	5	6
Comix	723.5(4)	187.9(3)	69.7(2)	27.14(7)	11.09(4)	4.68(2)	2.02(2)
ALPGEN	723.4(9)	188.3(3)	69.9(3)	27.2(1)	10.95(5)	4.6(1)	1.85(1)
AMEGIC	723.0(8)	188.2(3)	69.6(2)	27.21(6)	11.1(1)		
σ [pb]	Number of jets						
$\nu_e\bar{\nu}_e+$jets	0	1	2	3	4	5	6
Comix	3266(3)	715.9(8)	266.6(7)	105.0(3)	44.4(2)	19.11(7)	8.30(7)
ALPGEN	3271(1)	717.4(5)	267.4(4)	105.4(2)	43.7(2)	18.68(8)	7.88(5)
AMEGIC	3270(1)	717.3(7)	266.3(6)	105.4(3)	44.3(5)		

Table 7: Cross sections σ in the MC4LHC comparison [52] setup. In parentheses the statistical error is stated in units of the last digit of the cross section. Note that for Amegic++ and Comix all subprocesses are considered, while Alpgen is restricted to up to four quarks.

JHEP12(2008)039

σ [pb]	Number of jets						
$\gamma\gamma$ + jets	0	1	2	3	4	5	6
Comix	45.64(5)	25.23(6)	18.57(6)	9.64(4)	4.65(2)	2.07(2)	0.88(3)
AMEGIC	45.66(3)	25.41(6)	18.81(7)	9.82(3)			

σ [nb]	Number of jets					
γ + jets	1	2	3	4	5	6
Comix	89.5(2)	19.65(6)	7.52(3)	2.664(8)	1.000(5)	0.387(2)
AMEGIC	89.6(1)	19.60(5)	7.59(2)	2.64(2)		

σ [pb]	Number of jets					
$e^-\bar{\nu}_e + b\bar{b}$ + jets	0	1	2	3	4	5
Comix	9.40(2)	9.81(3)	6.82(5)	4.32(4)	2.47(2)	1.28(2)
ALPGEN	9.34(4)	9.85(6)	6.82(6)	4.18(7)	2.39(5)	
AMEGIC	9.37(1)	9.86(2)	6.98(3)	4.31(6)		

σ [pb]	Number of jets					
$e^-e^+ + b\bar{b}$ + jets	0	1	2	3	4	5
Comix	18.90(3)	6.81(2)	3.07(3)	1.536(9)	0.763(6)	0.37(1)
ALPGEN	18.95(8)	6.80(3)	2.97(2)	1.501(9)	0.78(1)	
AMEGIC	18.90(2)	6.82(2)	3.06(4)			

Table 8: Cross sections σ in the MC4LHC comparison [52] setup. In parentheses the statistical error is stated in units of the last digit of the cross section. Note that for Amegic++ and Comix all subprocesses are considered, while Alpgen is restricted to up to four quarks.

efficiency	Number of jets						
jets	2	3	4	5	6	7	8
$\varepsilon = 10^{-3}$	$9.3{\cdot}10^{-2}$	$7.8{\cdot}10^{-3}$	$2.1{\cdot}10^{-3}$	$7.0{\cdot}10^{-4}$	$3.6{\cdot}10^{-4}$	$1.3{\cdot}10^{-4}$	$6.1{\cdot}10^{-5}$
$\varepsilon = 10^{-6}$	$3.1{\cdot}10^{-2}$	$3.8{\cdot}10^{-3}$	$1.5{\cdot}10^{-3}$	$4.3{\cdot}10^{-4}$	$2.4{\cdot}10^{-4}$	$9.9{\cdot}10^{-5}$	$5.8{\cdot}10^{-5}$

efficiency	Number of jets						
$e^+\nu_e$ + jets	0	1	2	3	4	5	6
$\varepsilon = 10^{-3}$	$1.5{\cdot}10^{-1}$	$2.4{\cdot}10^{-2}$	$9.1{\cdot}10^{-3}$	$2.0{\cdot}10^{-3}$	$6.7{\cdot}10^{-4}$	$1.9{\cdot}10^{-4}$	$3.1{\cdot}10^{-5}$
$\varepsilon = 10^{-6}$	$1.6{\cdot}10^{-2}$	$4.5{\cdot}10^{-3}$	$3.3{\cdot}10^{-3}$	$1.2{\cdot}10^{-3}$	$4.3{\cdot}10^{-4}$	$1.3{\cdot}10^{-4}$	$2.8{\cdot}10^{-5}$

Table 9: Efficiencies for processes in the MC4LHC comparison [52] setup.

excellent supplementary generator for large multiplicities, which is especially helpful in the context of a matrix element - parton shower merging. The treatment of colour in Comix makes the algorithm well suited for such an interface, since the colour structure of the

JHEP12(2008)039

σ [nb]	Number of jets n	
QCD jets	7	8
$gg \to ng$	49.1(4)	14.2(3)
$gg \to (n-2)g\,2q$	17.0(1)	6.0(1)
$gg \to (n-4)g\,4q$	1.69(1)	0.74(5)
$gg \to (n-6)g\,6q$	0.0401(5)	0.0297(8)
$gg \to 8q$	-	0.000158(5)
$gq \to (n-1)g\,1q$	30.5(2)	9.9(2)
$gq \to (n-3)g\,3q$	8.46(6)	3.38(6)
$gq \to (n-5)g\,5q$	0.565(7)	0.332(8)
$gq \to (n-7)g\,7q$	0.00501(6)	0.0067(2)
$qq \to ng$	0.0209(1)	0.0067(1)
$qq \to (n-2)g\,2q$	4.97(4)	1.84(3)
$qq \to (n-4)g\,4q$	1.044(9)	0.477(9)
$qq \to (n-6)g\,6q$	0.0374(3)	0.0291(5)
$qq \to 8q$	-	0.000223(4)

σ [pb]	Number of jets n	
$e^+\nu_e$ + QCD jets	5	6
$qq \to e^+\nu_e\,ng$	0.256(2)	0.0768(6)
$qq \to e^+\nu_e\,(n-2)g\,2q$	6.49(3)	2.92(3)
$qq \to e^+\nu_e\,(n-4)g\,4q$	0.591(3)	0.449(8)
$qq \to e^+\nu_e\,6q$	-	0.00640(7)
$gq \to e^+\nu_e\,(n-1)g\,1q$	20.0(1)	8.21(8)
$gq \to e^+\nu_e\,(n-3)g\,3q$	4.03(2)	2.14(2)
$gq \to e^+\nu_e\,(n-5)g\,5q$	0.0741(4)	0.094(1)
$gg \to e^+\nu_e\,(n-2)g\,2q$	2.13(1)	0.775(5)
$gg \to e^+\nu_e\,(n-4)g\,4q$	0.1817(9)	0.1058(7)
$gg \to e^+\nu_e\,6q$	-	0.001403(7)

Table 10: Subprocess cross sections σ in the MC4LHC comparison [52] setup. In parentheses the statistical error is stated in units of the last digit of the cross section.

matrix element does not need to be guessed from the kinematics, it is rather fixed on a point by point basis. A corresponding publication is forthcoming [54].

Acknowledgments

We like to thank Claude Duhr, Frank Krauss and Fabio Maltoni for fruitful discussions and their comments on the manuscript. Special thanks for technical support go to Jonathan Ferland, Phil Roffe, Graeme Stewart and the ScotGrid [55] Tier 2 sites Durham and Glasgow. We thank Steffen Schumann for providing comparison results from Amegic++ and

JHEP12(2008)039

Michelangelo Mangano for results from Alpgen. 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 acknowledged.

References

[1] G. Ossola, C.G. Papadopoulos and R. Pittau, *Reducing full one-loop amplitudes to scalar integrals at the integrand level*, *Nucl. Phys.* **B 763** (2007) 147 [hep-ph/0609007].

[2] R.K. Ellis, W.T. Giele and Z. Kunszt, *A numerical unitarity formalism for evaluating one-loop amplitudes*, *JHEP* **03** (2008) 003 [arXiv:0708.2398].

[3] W.T. Giele, Z. Kunszt and K. Melnikov, *Full one-loop amplitudes from tree amplitudes*, *JHEP* **04** (2008) 049 [arXiv:0801.2237].

[4] G. Ossola, C.G. Papadopoulos and R. Pittau, *On the rational terms of the one-loop amplitudes*, *JHEP* **05** (2008) 004 [arXiv:0802.1876].

[5] S. Catani, T. Gleisberg, F. Krauss, G. Rodrigo and J.-C. Winter, *From loops to trees by-passing Feynman's theorem*, *JHEP* **09** (2008) 065 [arXiv:0804.3170].

[6] G. Ossola, C.G. Papadopoulos and R. Pittau, *CutTools: a program implementing the OPP reduction method to compute one-loop amplitudes*, *JHEP* **03** (2008) 042 [arXiv:0711.3596].

[7] C.F. Berger et al., *An automated implementation of on-shell methods for one-loop amplitudes*, *Phys. Rev.* **D 78** (2008) 036003 [arXiv:0803.4180].

[8] W.T. Giele and G. Zanderighi, *On the numerical evaluation of one-loop amplitudes: the gluonic case*, arXiv:0805.2152.

[9] R. Britto, F. Cachazo and B. Feng, *New recursion relations for tree amplitudes of gluons*, *Nucl. Phys.* **B 715** (2005) 499 [hep-th/0412308].

[10] 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].

[11] S.D. Badger, E.W.N. Glover, V.V. Khoze and P. Svrček, *Recursion relations for gauge theory amplitudes with massive particles*, *JHEP* **07** (2005) 025 [hep-th/0504159].

[12] S.D. Badger, E.W.N. Glover and V.V. Khoze, *Recursion relations for gauge theory amplitudes with massive vector bosons and fermions*, *JHEP* **01** (2006) 066 [hep-th/0507161].

[13] K.J. Ozeren and W.J. Stirling, *Scattering amplitudes with massive fermions using BCFW recursion*, *Eur. Phys. J.* **C 48** (2006) 159 [hep-ph/0603071].

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

[15] K. Risager, *A direct proof of the CSW rules*, *JHEP* **12** (2005) 003 [hep-th/0508206].

[16] P. Mansfield, *The Lagrangian origin of MHV rules*, *JHEP* **03** (2006) 037 [hep-th/0511264].

[17] S.D. Badger, E.W.N. Glover and V.V. Khoze, *MHV rules for Higgs plus multi-parton amplitudes*, *JHEP* **03** (2005) 023 [hep-th/0412275].

JHEP12(2008)039

[18] T.G. Birthwright, E.W.N. Glover, V.V. Khoze and P. Marquard, *Collinear Limits in QCD from MHV rules*, *JHEP* **07** (2005) 068 [hep-ph/0505219].

[19] C. Duhr and F. Maltoni, *Antenna functions from MHV rules*, *JHEP* **11** (2008) 002 [arXiv:0808.3319].

[20] T. Gleisberg, S. Hoeche and F. Krauss, *How to calculate colourful cross sections efficiently*, arXiv:0808.3672.

[21] A. Kanaki and C.G. Papadopoulos, *HELAC: a package to compute electroweak helicity amplitudes*, *Comput. Phys. Commun.* **132** (2000) 306 [hep-ph/0002082].

[22] C.G. Papadopoulos, *PHEGAS: a phase space generator for automatic cross-section computation*, *Comput. Phys. Commun.* **137** (2001) 247 [hep-ph/0007335].

[23] A. Cafarella, C.G. Papadopoulos and M. Worek, *Helac-Phegas: a generator for all parton level processes*, arXiv:0710.2427.

[24] F. Krauss, R. Kuhn and G. Soff, *AMEGIC++* 1.0*: a matrix element generator in C++*, *JHEP* **02** (2002) 044 [hep-ph/0109036].

[25] F. Maltoni and T. Stelzer, *MadEvent: automatic event generation with MadGraph*, *JHEP* **02** (2003) 027 [hep-ph/0208156].

[26] J. Alwall et al., *MadGraph/MadEvent v4: the new web generation*, *JHEP* **09** (2007) 028 [arXiv:0706.2334].

[27] M.L. Mangano, M. Moretti, F. Piccinini, R. Pittau and A.D. Polosa, *ALPGEN, a generator for hard multiparton processes in hadronic collisions*, *JHEP* **07** (2003) 001 [hep-ph/0206293].

[28] F.A. Berends and W.T. Giele, *Recursive calculations for processes with n gluons*, *Nucl. Phys.* **B 306** (1988) 759.

[29] F.A. Berends and W. Giele, *The six gluon process as an example of Weyl-Van Der Waerden spinor calculus*, *Nucl. Phys.* **B 294** (1987) 700.

[30] R. Kleiss and H. Kuijf, *Multi-gluon cross-sections and five jet production at hadron colliders*, *Nucl. Phys.* **B 312** (1989) 616.

[31] F.A. Berends, W.T. Giele and H. Kuijf, *Exact expressions for processes involving a vector boson and up to five partons*, *Nucl. Phys.* **B 321** (1989) 39.

[32] F.A. Berends, W.T. Giele and H. Kuijf, *On six jet production at hadron colliders*, *Phys. Lett.* **B 232** (1989) 266.

[33] 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].

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

[35] P.D. Draggiotis, R.H.P. Kleiss and C.G. Papadopoulos, *Multi-jet production in hadron collisions*, *Eur. Phys. J.* **C 24** (2002) 447 [hep-ph/0202201].

[36] F. Caravaglios and M. Moretti, *An algorithm to compute Born scattering amplitudes without Feynman graphs*, *Phys. Lett.* **B 358** (1995) 332 [hep-ph/9507237].

[37] E. Byckling and K. Kajantie, *N-particle phase space in terms of invariant momentum transfers*, *Nucl. Phys.* **B 9** (1969) 568.

JHEP12(2008)039

[38] A. van Hameren and C.G. Papadopoulos, *A hierarchical phase space generator for QCD antenna structures*, *Eur. Phys. J.* **C 25** (2002) 563 [hep-ph/0204055].

[39] M.L. Mangano, S.J. Parke and Z. Xu, *Duality and multi-gluon scattering*, *Nucl. Phys.* **B 298** (1988) 653.

[40] V. Del Duca, L.J. Dixon and F. Maltoni, *New color decompositions for gauge amplitudes at tree and loop level*, *Nucl. Phys.* **B 571** (2000) 51 [hep-ph/9910563].

[41] V. Del Duca, A. Frizzo and F. Maltoni, *Factorization of tree QCD amplitudes in the high-energy limit and in the collinear limit*, *Nucl. Phys.* **B 568** (2000) 211 [hep-ph/9909464].

[42] F. Maltoni, K. Paul, T. Stelzer and S. Willenbrock, *Color-flow decomposition of QCD amplitudes*, *Phys. Rev.* **D 67** (2003) 014026 [hep-ph/0209271].

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

[44] K. Hagiwara and D. Zeppenfeld, *Helicity amplitudes for heavy lepton production in e^+e^- annihilation*, *Nucl. Phys.* **B 274** (1986) 1.

[45] S. Dittmaier, *Weyl-van-der-Waerden formalism for helicity amplitudes of massive particles*, *Phys. Rev.* **D 59** (1999) 016007 [hep-ph/9805445].

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

[47] G.P. Lepage, *VEGAS — an adaptive multi-dimensional integration program*, CLNS-80-447.

[48] F. James, *Monte Carlo phase space*, CERN-68-15.

[49] F. Caravaglios, M.L. Mangano, M. Moretti and R. Pittau, *A new approach to multi-jet calculations in hadron collisions*, *Nucl. Phys.* **B 539** (1999) 215 [hep-ph/9807570].

[50] 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.

[51] G.C. Blazey et al., *Run II jet physics*, hep-ex/0005012.

[52] *CERN Workshop on Monte Carlo tools for the LHC homepage*, http://mlm.home.cern.ch/mlm/mcwshop03/mcwshop.html.

[53] S. Jadach, *Foam: multi-dimensional general purpose Monte Carlo generator with self-adapting symplectic grid*, *Comput. Phys. Commun.* **130** (2000) 244 [physics/9910004].

[54] S. Höche, F. Krauss, S. Schumann and F. Siegert, *A comprehensive approach to CKKW merging*, in preparation.

[55] *ScotGrid Scottish grid service homepage*, http://www.scotgrid.ac.uk.

JHEP12(2008)039