The ALPGEN Generator

M.L. Mangano, M. Moretti, F. Piccinini, R. P., A.D. Polosa, JHEP07(2003)001

- Collection of MC codes for many processes relevant in h–h collisions
- Exact LO matrix element calculations based on the ALPHA algorithm
- Parton-level event generation (weighted and unweighted)
- Interface to Herwig/Pythia for the evolution of the partonic final state through parton shower

Up to now available processes

- $W + N$ jets, $Z/\gamma + N$ jets, $N \leq 6$
- $WQ\bar{Q} + N$ jets, $Z/\gamma Q\bar{Q} + N$ jets $(Q = b, t)$, $N \leq 4$
- $W + c + N$ jets, $N \leq 5$
- $n\ W + m\ Z + p$ Higgs $+ N$ jets, $n + m + p \leq 8$, $N \leq 3$
- $Q\bar{Q} + N$ jets, $(Q = b, t)$, $N \leq 6$
- $Q\bar{Q}Q'\bar{Q}' + N$ jets, $(Q, Q' = b, t)$, $N \leq 4$
- N jets, $N \leq 6$
- $Q\bar{Q}H + N$ jets, $(Q = b, t)$, $N \leq 4$

The directory structure

```
2Qlib        History.txt  QQhwork      readme.txt   wcjetwork    zjetlib
2Qwork       Makefile     VF90         v122.tgz     wjetlib      zjetwork
4Qlib        Njetslib     alplib       vbjetslib    wjetwork     zqqlib
4Qwork       Njetswork    herlib       vbjetswork   wqqlib       zqqwork
DOCS         QQhlib       pylib        wcjetlib     wqqwork
```

The common driver is in `alplib/alpgen.f`:

```
C-----------------------------------------------------------------
      program alpgen
C-------------------------------------------------------------------------
C
C driver for multi-parton matrix element generator based on ALPHA
C
C--------------------------------------------------------------------------
C
C setup running defaults:
c
      call setdef
c
c     Routine location: alplib/alpgen.f
```

```
c     Purpose:
c        -  initialise the default generation parameters (e.g. beam
c           type, energy, PDF set)
c        -  initialise default mass and couplings for particles
c
c
C input user-set running parameters and generation cuts:
c
      call setusr
c
c     Routine location: XXXwork/XXXusr.f (e.g. wqqwork/wqqusr.f)
c     Purpose:
c         -change the defaults set in setdef
c   -initialise the parameters
c           specific to the desired process (XXX)
c
c
c
C setup  event generation options, bookkeeping, etc
      call setgen
c
c     Routine location: XXXlib/XXX.f
```

```
c     Purpose:
c        - setup  event generation options, bookkeeping, etc, for
c              the specific process XXX
c        - evaluate process-dependent quantities
c          (e.g. Higgs width)
c  - fill the parameters required by ALPHA
c
c
C setup histograms
      call sethis
c
c     Routine location: XXXwork/XXXusr.f
c     Purpose: initialise histograms
c
c
C setup integration grids, including optimization if required
c
      call setgrid
c
c     Routine location: XXXlib/XXX.f
c     Purpose: setup integration grid variables
      call inigrid
```

```
c
c     Routine location: alplib/alpgen.f
c     Purpose: initialise grid with warm-up iterations, if required
c
c
C generate events
c
      call evtgen
c
c     Routine location: alplib/alpgen.f
c     Purpose: generates events, calling in a standad format the
c          the process-dependent phase-space and flavour-selection
c          routines, contained in XXXlib/XXX.f

C finalise histograms
c
      call finhis
c
c     Routine location: XXXwork/XXXusr.f
c     Purpose: finalize analysis and histograms
c
      end
```

How to run a specific process (e.g. Wqq+jets)?

```
> cd wqqwork
> make wqqgen
```

The input file:

```
0             ! imode (0,1,2)
'output'      ! label for files
1             ! initial state (1=pp, -1=ppbar)
7000d0 2      ! beam energy in CM frame and PDF set
4.75d0        ! Q mass
4             ! total number of final-state QCD partons (included Q and Qbar)
30 2.5 0.7    ! ptmin(j) etamax(j), deltaRmin(j-bbar)
30 2.5 0.7    ! ptmin(b) etamax(b), deltaRmin(b-bbar)
0d0 10d0 0d0 0d0 ! ptmin(lept), etamax(lept), deltaRmin(lep-j), etmiss
1 1d0         ! iqopt, qfac
0  ! start with: 0=new grid, 1=previous warmup grid, 2=previous generation grid
500000 4   ! Nevents/iteration,  N(warm-up iterations)
1000000    ! Nevents generated after warm-up
1          ! 1: change default random number seed, 0: keep default seed
54345
65455
```

The Matrix Element computation

Multiparticle amplitudes involve the evaluation of large numbers of Feynamn diagrams. *e.g.*

Process	$n = 7$	$n = 8$	$n = 9$	$n = 10$
$g\ g \rightarrow n\ g$	559,405	10,525,900	224,449,225	5,348,843,500
$q\bar{q} \rightarrow n\ g$	231,280	4,016,775	79,603,720	1,773,172,275

Table 1: Number of Feynman diagrams corresponding to amplitudes with different numbers of quarks and gluons.

F. Caravaglios et al., NPB 539 (1999) 215

A pure numerical approach to the calculations of transition amplitudes is welcome. This can be done with the ALPHA algorithm

F. Caravaglios and M. Moretti, PLB 358 (1995) 332

The Idea: The Matrix Element 'is' the Legendre Transform Z of the (effective) lagrangian Γ (1PI Green Functions generator) → the problem can be recasted as a *minimum problem*, more suitable for a *numerical approach* (DS equation).

Subprocess selection

The calculation of the cross section for multiparton final states involves typically the sum over a large set of subprocesses and flavour configurations

e.g. for the $Wb\bar{b}$ final state

`jp`	subprocess	`jp`	subprocess	`jp`	subprocess
1	$q\bar{q}' \to WQ\overline{Q}$	2	$qg \to q'WQ\overline{Q}$	3	$gq \to q'WQ\overline{Q}$
4	$gg \to q\bar{q}'WQ\overline{Q}$	5	$q\bar{q}' \to WQ\overline{Q}q''\bar{q}''$	6	$qq'' \to WQ\overline{Q}q'q''$
7	$q''q \to WQ\overline{Q}q'q''$	8	$q\bar{q} \to WQ\overline{Q}q'\bar{q}''$	9	$q\bar{q}' \to WQ\overline{Q}q\bar{q}$
10	$\bar{q}'q \to WQ\overline{Q}q\bar{q}$	11	$q\bar{q} \to WQ\overline{Q}q\bar{q}'$	12	$q\bar{q} \to WQ\overline{Q}q'\bar{q}$
13	$qq \to WQ\overline{Q}qq'$	14	$qq' \to WQ\overline{Q}qq$	15	$qq' \to WQ\overline{Q}q'q'$
16	$qg \to WQ\overline{Q}q'q''\bar{q}''$	17	$gq \to WQ\overline{Q}q'q''\bar{q}''$	18	$qg \to WQ\overline{Q}qq\bar{q}'$
19	$qg \to WQ\overline{Q}q'q\bar{q}$	20	$gq \to WQ\overline{Q}qq\bar{q}'$	21	$gq \to WQ\overline{Q}q'q\bar{q}$
22	$gg \to WQ\overline{Q}q\bar{q}'q''\bar{q}''$	23	$gg \to WQ\overline{Q}q\bar{q}q\bar{q}'$		

Each of these subprocesses receives contributions from several possible flavour configurations (e.g. $u\bar{d} \to WQ\overline{Q}gg$, $u\bar{s} \to WQ\overline{Q}gg$, etc.).

Our subdivison in subprocesses is designed to allow to sum the contribution of different flavour configurations by simply adding trivial factors such as parton densities or CKM factors, which factorize out of a single, flavour independent, matrix element

For example the overall contribution from the 1^{st} process in the list is given by

$$\left[u_1\bar{d}_2\cos^2\theta_c + u_1\bar{s}_2\sin^2\theta_c + c_1\bar{s}_2\cos^2\theta_c + c_1\bar{d}_2\sin^2\theta_c\right] \times |M(q\bar{q}' \to WQ\overline{Q}gg)|^2$$

where $q_i = f(x_i)$, $i = 1, 2$, are the parton densities for the quark flavour q. Contributions from charge-conjugate or isospin-rotated states can also be summed up, after trivial momentum exchanges. For example, the same matrix element calculation is used to describe the four events:

$$\begin{array}{rcl}
u(p_1)\bar{d}(p_2) & \to & b(p_3)\bar{b}(p_4)g(p_5)g(p_6)e^+(p_5)\nu(p_6) \\
\bar{u}(p_1)d(p_2) & \to & \bar{b}(p_3)b(p_4)g(p_5)g(p_6)e^-(p_5)\bar{\nu}(p_6) \\
\bar{d}(p_1)u(p_2) & \to & \bar{b}(p_3)b(p_4)g(p_5)g(p_6)\nu(p_5)e^+(p_6) \\
d(p_1)\bar{u}(p_2) & \to & b(p_3)\bar{b}(p_4)g(p_5)g(p_6)\bar{\nu}(p_5)e^-(p_6)
\end{array}$$

Event by event, the flavour configuration for the assigned subprocess is then selected with a probability proportional to the relative size of the individual contributions to the luminosity, weighted by the Cabibbo angles

Phase Space generation

Two quantities must be kept maximal, namely the Monte Carlo generation efficiency ϵ_g, that is the ratio between the maximum and the average weight, and the Phase Space generation efficiency ϵ_{PS}, defined as the ratio between the number of events that fall inside the cuts and the total number of generated events.

The main idea is generating events directly in terms of quantities measured in the Laboratory Frame, namely pseudo-rapidities η_i, transverse momenta p_{Ti} and azimuthal angles ϕ_i. This gives the possibility of generating kinematical variables, with given probability densities, directly within the cuts, increasing ϵ_{PS}.

After integrating over the momentum fractions x_1 and x_2 of the initial state partons, the n-body Phase Space integral

$$\int D\Phi_n = \int (\prod_i^n \frac{d^3 p_i}{2\,E_i}) \delta^4(P - \sum_i^n p_i) \tag{1}$$

gives the expression

$$I = \int dx_1\, dx_2\, D\Phi_n = \frac{2^{(1-n)}}{s} \int (\prod_i^n d\eta_i \beta_i)(\prod_i^{n-1} dp_{Ti}\ p_{Ti})(\prod_i^{n-1} d\phi_i)\,. \tag{2}$$

In the previous equations

$$\begin{aligned} P &= \frac{\sqrt{s}}{2}(x_1 + x_2, 0, 0, x_1 - x_2) \\ p_i &= (E_i, p_{Ti}\ \sin(\phi_i), p_{Ti}\ \cos(\phi_i), p_{Li})\,, \\ \beta_i &= \sqrt{1 - \frac{m_i^2}{E_i^2}}\,, \end{aligned} \tag{3}$$

where $E_i = \sqrt{p_{Ti}^2\ \cosh^2(\eta_i) + m_i^2}$ and $p_{Li} = p_{Ti}\ \sinh(\eta_i)$ are the energies and the longitudinal momenta of the outgoing partons. Notice the presence of non vanishing masses m_i for the outgoing particles.

With a change of variable, one can generate x_1 and x_2 instead of η_n and η_{n-1}.

Further Improving the efficiencies

We are left with the choice of the probability distributions for all the remaining integration variables. While it is easy to argue leading behaviors such as, for example, $1/p_T^\ell$ for the transverse momenta, the best value of ℓ to be used is in general difficult, if not impossible, to establish a-priori; without mentioning that it may also depend on the generation range of p_T. Therefore, to increase the Monte Carlo generation efficiency ϵ_g, we have to rely on self-adapting procedures.

Our strategy is best illustrated with a 1-dimensional example. Imagine we have to compute numerically the integral

$$I = \int_0^1 dx f(x)\,. \tag{4}$$

We can split the integration interval into N bins, multiply and divide by a-priori weights α_i, normalized such that $\sum_{i=1}^N \alpha_i = 1$, and rewrite

$$I = \sum_{i=1}^{N} \alpha_i\, J_i\,, \quad J_i = \frac{1}{\alpha_i} \int_{(i-1)/N}^{i/N} dx f(x) \equiv \frac{1}{\alpha_i} I_i\,. \tag{5}$$

The original integral is then re-expressed as a sum of terms, that we can perform by Monte Carlo. More explicitly, we start with $\alpha_i = 1/N$ and choose to generate the variable x between $(j-1)/N$ and j/N if a uniformly generated random number $0 < \rho < 1$ falls between $\sum_{i=1}^{j-1} \alpha_i$ and $\sum_{i=1}^{j} \alpha_i$. While performing the Monte Carlo integration, we bookkeep the estimated values for the N quantities I_i, and, after a few thousand points, define a new set of weights as follows:

$$\alpha_i' = \frac{I_i}{\sum_{j=1}^{N} I_j} \, . \tag{6}$$

After few iterations, the probability of choosing the i^{th} generation interval for x, becomes proportional to I_i, namely to the contribution of the i^{th} interval to the whole result, improving ϵ_g.

The extension to m-dimensional integrations is straightforward. Instead of equation (4) we have

$$I = \prod_{i=1}^{m} \int_0^1 \; dx_i f(\vec{x}) \, , \tag{7}$$

so that each of the m integrations can be separately treated as described.

Notice that each variable is re-weighted independently from the others, so that non factorizable peaking structures may cause problems during the self-adaptation. In this respect, our procedure is quite similar to the classical VEGAS algorithm, the only difference being that, instead of changing the size of the bins, we re-weight them.

With the help of the above procedure, one improves not only ϵ_g, but also the Phase Space generation efficiency ϵ_{PS} of the Monte Carlo. The reason is that when, due to the cuts, a particular bin gives a small contribution, the corresponding resulting weight, computed with equation (6), is also small.

To further contribute to the efficiency of the phase space sampling, independent grids are employed to sample different subprocesses. In particular, one phase space grid is associated to each of the following initial states:

1. $q\bar{q}$, $q\bar{q}'$ and charge conjugates
2. qg and $\bar{q}g$
3. gq and $g\bar{q}$
4. gg
5. qq, qq' and charge conjugates

Reconstruction of colour flows

The emission of soft gluon radiation in shower MC programs accounts for quantum coherence, which is implemented via the prescription of angular ordering in the parton cascade. The colour flow is the set of colour connections among the partons which defines the set of dipoles for a given event

In order to reliably evolve a multiparton state into a multijet configuration, it is necessary to associate a specific colour-flow pattern to each generated parton-level event

Consider for example the case of multigluon processes. The scattering amplitude for n gluons with momenta p_i^μ, helicities ϵ_i^μ and colours a_i (with $i = 1, \ldots, n$), can be written as

F.A. Berends and W. Giele, NPB 294 (1987) 700

M. Mangano, S. Parke and Z. Xu, NPB 298 (1988) 653

$$M(\{p_i\}, \{\epsilon_i\}, \{a_i\}) = \sum_{P(2,3,\ldots,n)} \mathrm{tr}(\lambda^{a_{i_1}} \lambda^{a_{i_2}} \ldots \lambda^{a_{i_n}}) A(\{p_{i_1}\}, \{\epsilon_{i_1}\}; \ldots \{p_{i_n}\}, \{\epsilon_{i_n}\})$$

The functions $A(\{P_i\})$ (known as *dual* or *colour-ordered* amplitudes) are gauge-invariant, cyclically-symmetric functions of the gluons' momenta and helicities. Each dual amplitude $A(\{P_i\})$ corresponds to a set of diagrams where colour flows from one gluon to the next, according to the ordering specified by the permutation of indices

When summing over colours the amplitude squared, different orderings of dual amplitudes are orthogonal at the leading order in $1/N^2$

$$\sum_{\mathrm{col's}} |M(\{p_i\},\{\epsilon_i\},\{a_i\})|^2 \;=\; N^{n-2}(N^2-1)\sum_{P_i}\left[|A(\{P_i\})|^2 + \frac{1}{N^2}(\mathrm{interf.})\right]$$

At the leading order in $1/N^2$, therefore, the square of each dual amplitude is proportional to the relative probability of the corresponding colour flow. Each flow defines, in a gauge invariant way, the set of colour currents which are necessary and sufficient to implement the colour ordering prescription necessary for the coherent evolution of the gluon shower. Because of the incoherence of different colour flows, each event can be assigned a specific colour configuration by comparing the relative size of $|A(\{P_i\})|^2$ for all possible flows

When working at the physical value of $N_c = 3$, the interferences among different flows cannot be neglected in the evaluation of the square of the matrix element. As a result, the basis of colour flows does not provide an orthogonal set of colour states $\Rightarrow$

Our solution for and efficient color generation including $1/N$ corrections in the Matrix Element evaluation:

- choose a standard $SU(3)$ orthonormal basis (Gell-Mann matrices for example)
- randomly select a non-vanishing colour assignement for the exernal gluons
- *if the event is accepted* choose randomly among the *contributing* dual amplitudes a color flow on the basis of their relative weight

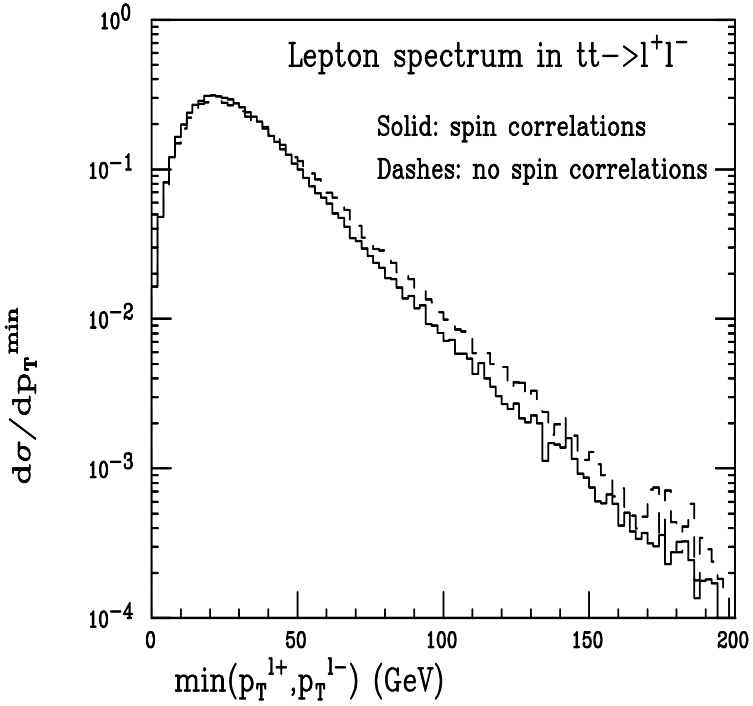
  Two advantages

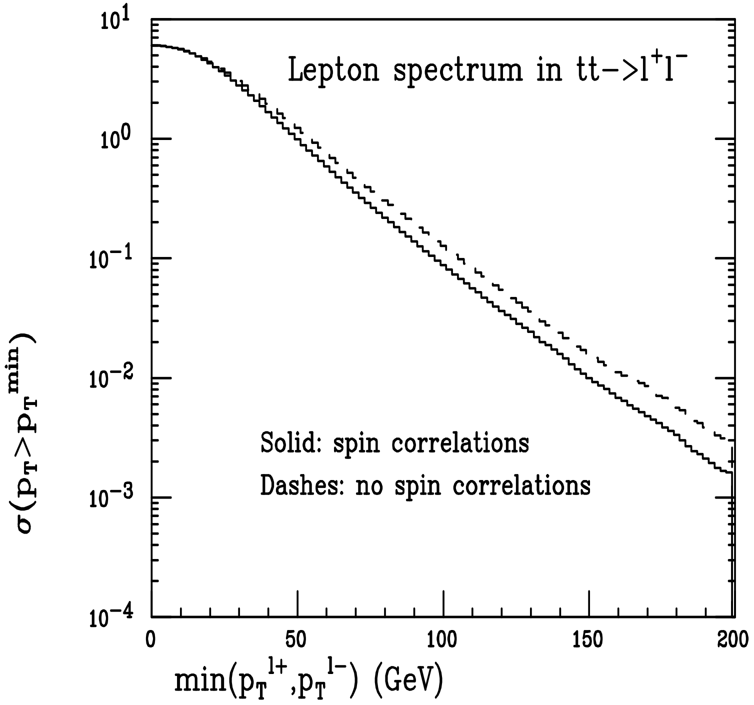
  – dual amplitudes required only for a small number of phase space points

  – contributing dual amplitudes to a given external coulor assignment $\ll$ than total number.

Top decays with full spin correlations

In $t\bar{t}+$ jets the top quarks are generated on shell, however the decay $t \to b f \bar{f}'$ is generated with exact matrix element keeping top quark and W boson on shell, in order to avoid inclusion of non-resonant diagrams while retaining gauge invariance. In so doing all spin correlations between top decay products are exactly taken into account





Multi-boson + jets production

W/Z gauge bosons produced on shell and let decay in fermionic pairs in the zero width approximation but including exact spin correlations among the decay products

Gauge invariance: the calculation of multi-boson final states requires a careful treatment of the widths in the propagators because they generally break gauge invariance, giving rise to bad high energy behaviour of the cross sections. The strategy adopted in ALPGEN is to calculate the matrix element with $\Gamma_i = 0$, by cutting away events around the mass of the instable particle M is such a way to keep the area of a Breit-Wigner distribution

$$\int_{-\infty}^{M^2-s_0} ds \frac{1}{(s-M^2)^2} = \int_{-\infty}^{M^2} ds \frac{1}{(s-M^2)^2+\Gamma^2 M^2}$$

that gives $s_0 = \frac{2\Gamma M}{\pi}$ and the condition

$$|s-M^2| \geq \frac{2\Gamma M}{\pi}$$

Higgs production processes

- $t\bar{t}H+$ jets
- $b\bar{b}H+$ jets
- $H+$ gauge bosons $+$ jets
- $gg \rightarrow H+$ jets (work in progress)

Spin correlations in top and gauge boson decays exactly taken into account At present no Higgs decay implemented (work in progress)

List of wishes

- Inclusion of `CKKW` algorithm
- Inclusion of p.d.f. with errors
- Improving the generation efficiency