TPJU-6/2004
hep-ph/0405247

The Monte Carlo Event Generator AcerMC version 2.4 with interfaces to PYTHIA 6.2, HERWIG 6.5 and ARIADNE 4.1

Borut Paul Kersevan
Faculty of Mathematics and Physics, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia.
and
Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia.

Elzbieta Richter-Wąs*
Institute of Physics, Jagellonian University,
30-059 Krakow, ul. Reymonta 4, Poland;
and
Institute of Nuclear Physics PAS, 31-342 Krakow, ul. Radzikowskiego 152, Poland.

March 21, 2005

Abstract

The **AcerMC** *Monte Carlo Event Generator is dedicated for the generation of Standard Model background processes at pp LHC collisions. The program itself provides a library of the massive matrix elements and phase space modules for generation of selected processes:* $gg, q\bar{q} \to t\bar{t}b\bar{b}$; $q\bar{q}W(\to \ell\nu)b\bar{b}$; $q\bar{q}W(\to \ell\nu)t\bar{t}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell)b\bar{b}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$; *complete electroweak* $gg, q\bar{q} \to (Z/W/\gamma^* \to)b\bar{b}t\bar{t}$; $gg, q\bar{q} \to t\bar{t}t\bar{t}$; $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}bf\bar{f}\bar{b}$; $gg, q\bar{q} \to (WWbb \to)f\bar{f}f\bar{f}b\bar{b}$. *The hard process event, generated with one of these modules, can be completed by the initial and final state radiation, hadronisation and decays, simulated with either* PYTHIA, ARIADNE *or* HERWIG *Monte Carlo event generator and (optionally) with* TAUOLA *and* PHOTOS*. Interfaces to all these packages are provided in the distribution version. The matrix element codes have been derived with the help of the* MADGRAPH *package. The phase-space generation is based on the multi-channel self-optimising approach using the modified Kajantie-Byckling formalism for phase space construction and further smoothing of the phase space was obtained by using a modified* ac-VEGAS *algorithm.*

Key Words: SM backgrounds at LHC, massive matrix elements, Monte Carlo generator, heavy flavor production, multi-channel phase-space generation
PACS: 02.70.-c , 13.38.-b , 13.90.+i

*Partly supported by Marie Curie Host Fellowship for the Transfer of Knowledge Contract No. MTKD-CT-2004-510126 and by the EC FP5 Centre of Excellence "COPIRA" under the contract No. IST-2001-37259.

Contents

1 PROGRAM SUMMARY **5**

2 Changes since AcerMC 1.0 **6**

3 Introduction **7**

4 Physics content **8**
- 4.1 The $gg, q\bar{q} \to t\bar{t}b\bar{b}$ processes . . . 9
- 4.2 The $q\bar{q} \to W(\to \ell\nu)g^*(\to b\bar{b})$ process . . . 10
- 4.3 The $q\bar{q} \to W(\to \ell\nu)g^*(\to t\bar{t})$ process . . . 12
- 4.4 The $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell)b\bar{b}$ processes . . . 12
- 4.5 The $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$ processes . . . 13
- 4.6 The electroweak $gg, q\bar{q} \to (Z/W/\gamma^* \to)b\bar{b}t\bar{t}$ process . . . 14
- 4.7 The $gg, q\bar{q} \to (WWb\bar{b} \to)f\bar{f}f\bar{f}b\bar{b}$; $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}f\bar{f}b\bar{b}$ processes . . . 14
- 4.8 The $gg, q\bar{q} \to t\bar{t}t\bar{t}$ process . . . 16
- 4.9 The *control channel* processes . . . 17

5 Monte Carlo algorithm **18**
- 5.1 The Matrix Element Calculation . . . 18
- 5.2 The Phase Space Generation Procedure . . . 19
- 5.3 Modified Kajantie-Byckling Formalism . . . 22
  - 5.3.1 The s-type Branching Algorithms . . . 22
  - 5.3.2 The t-type Branching Algorithms . . . 25
- 5.4 Propagator Sampling . . . 28
  - 5.4.1 Breit-Wigner Function with s-dependent Width . . . 28
  - 5.4.2 The Inclusion of Mass Effects in Propagator Sampling . . . 31
- 5.5 Application of the Phase Space Generation Algorithms . . . 34
  - 5.5.1 Modified VEGAS Algorithm . . . 35
  - 5.5.2 Colour Flow Information . . . 37
- 5.6 The α_{QED} and α_s calculations . . . 37

6 Structure of the package **40**
- 6.1 Main event loop and interface to PYTHIA/HERWIG, TAUOLA and PHOTOS . . . 40
- 6.2 Structure of the **AcerMC** matrix-element and phase-space code . . . 41
- 6.3 Data files for the phase-space optimisation . . . 42

7 How to use the package **45**
- 7.1 Steering switches of the overall run . . . 45
- 7.2 Steering switches of the **AcerMC** processes . . . 45
- 7.3 Steering TAUOLA and PHOTOS . . . 48
- 7.4 How to prepare data-files for the non-default setup . . . 50
- 7.5 Details on the interface to PYTHIA 6.2 . . . 51
- 7.6 Details on the interface to ARIADNE 4.1 . . . 51
- 7.7 Details on the interface to HERWIG 6.5 . . . 52
- 7.8 Definition of the energy scale . . . 53
- 7.9 Installation procedure . . . 54
- 7.10 Storing and reading events using the Les Houches accord . . . 54
- 7.11 Interface of TAUOLA to PYTHIA and HERWIG . . . 55
- 7.12 Interface of PHOTOS to PYTHIA and HERWIG . . . 55
- 7.13 Details of the TAUOLA implementation . . . 55
- 7.14 Details of the PHOTOS implementation . . . 56

8 Outlook and conclusions **57**

A Feynman Diagrams **60**

B Example input files **69**
B.1 File run.card . . . 69
B.2 File acermc.card . . . 71
B.3 File tauola.card . . . 73
B.4 File photos.card . . . 75

C Example output files **76**
C.1 File acermc.out . . . 76
C.2 File pythia.out . . . 78
C.3 File herwig.out . . . 80

```
 -------------------------------------------------------------------------------
                                   ._
                                 .j%3]:,
                               ~!%%%%%%% ,._.
                            _|xx%xxxx%%%%%+`
                             :~]%xxxx]xx%x_,+_x_%`
                       -__||x||xx]+]]]+]]]]x|xxx]`
                        -+%%xxxx]]]]+]]++]+]x]|>- .;..;.:_/`
                           -+x]]]]|+]+]+]++]++]]+|+|]+|]]+-
               ,.   .  ., |x]+]+||=++]+]=+++++++=|++]=+]|~-
               -|%x]]];x]]||=+++++++++|];|+++++++++++=+]=];   ..    ..   :..,;
                 -/]||]+]|]+++++++++||||==|:;:=|==++==+;|,    :;;. :;=;;===|`
                  _|]|+>+++]+]]|+|||=|=;|;;:|;=:===|==;::.;;:;,;:;;;;;;;:-
                      -++]+++:+|x+::||=||:=:::::;;:::::::::;:;:;:;:;;;=::
                    .|x+|+|]++:,-..:::|=;=:-.:.:-::.:.:.::-:::::::::;:-
          .,   .:||_   --||;:|:::.-.:.-:|;||::.:...-.-.-....::::-:::;::.
      __._;++;;;|=|;. -:::::::---:.--;;|==|:;:.:-:-:.-.-.:::.--:::;:;--
      -+++=+=======;==:::::-:.::...:.|+=;;===:-..:.:.:::::-...--:  -
      :|:-:|===;:;:;::::::--::.:-::::.|||===:::.:.:::::.-......--:: .
         -;|===;;:;;;;;::;::.:-:: -:=;;|+||=;==|=:::::::::...-.:..-.::..
          ---|;:,::::.::::=;=:=::  -++===|======:--:   ...-...-:-:.-
              ---;:::::;:--:===;;;:|-|+=+|+|==;:`    ...-....-..
              .::;;:::;:;;:--- -- --.|=||=+|=:=,      ..... .
             .;;=;;:-:::;:::.        -+;,|]| ;;=`
               - -:- --:;;:- -         - :|; -
                        -                  :`
                                            :
                                            :
                                            .

  40000L,                                          |0000i   j000&  .a00000L#0
  --?##0L        .aaaa aa     .aaaa;     aaaa, _aaa, -000A  _0001- _d0!`  -400
    d0 40,     _W0#V9N0#&    d0#V*N#0,   0##0LW0@4#@' 00j#; J0|01  d0'      40
   J0l -#W     #0'    ?#W   ##~    -#0;    j##9       00 4#|01|01  00
  _00yyyW0L   :0f      ^-  :000###00001    j#1        00 ?#0@`|01  #0
  ##!!!!!#0;  -0A       _  -0A             j#1        00  HH< |01  j0L       _
ad0La,  aj0Aa  4#Aaa_aj#0`  ?0Laa_aaa0L  aaJ0Laaa,  _a00aa  _aj0La  *0Aaa_aad
HHHRHl  HHHHH   `9##009!     `9NW00@!!`  HHHHRHHHl  :HHHHH  ?HHHRH   ?!##00P!`

 -------------------------------------------------------------------------------

       AcerMC 2.4 (March 2005),  B. P. Kersevan, E. Richter-Was

 -------------------------------------------------------------------------------
```

Available from web page: **http://cern.ch/Borut.Kersevan**

Contact author email: **Borut.Kersevan@cern.ch**

1 PROGRAM SUMMARY

Title of the program: **AcerMC version 2.4**
Operating system: Linux
Programming language: FORTRAN 77 with popular extensions.
External libraries: CERNLIB.
Size of the compressed distribution directory: about 10.7 MB. The distribution includes modified versions of PYTHIA 6.2, HERWIG 6.5, HELAS libraries, TAUOLA and PHOTOS packages.
Key words: Standard Model backgrounds at LHC, massive matrix elements, Monte Carlo generator, heavy flavor production, multi-channel phase-space generation.

Nature of physical problem: Despite a large repertoire of processes implemented for generation in event generators like PYTHIA [1] or HERWIG [2] a number of background processes, crucial for studying expected physics potential of the LHC experiments is missing. For some of these processes the matrix element expressions are rather lengthly and/or to achieve a reasonable generation efficiency it is necessary to tailor the phase-space selection procedure to the dynamics of the process. That is why it is not practical to imagine that any of the above general purpose generators will contain *every*, or even only *observable*, processes which will occur at LHC collisions. A more practical solution can be found in a library of dedicated matrix-element-based generators, with the standardised interfaces like that proposed in [3], to the more universal one which is used to complete the event generation.
Method of solution: The **AcerMC** Event Generator provides itself library of the matrix-element-based generators for several processes. The initial- and final- state showers, beam remnants and underlying events, fragmentation and remaining decays are supposed to be performed by the other universal generator to which this one is interfaced. We will call it *supervising generator*. The interfaces to PYTHIA 6.2, ARIADNE 4.1 and HERWIG 6.5, as such generators, are provided. Provided is also interface to TAUOLA [4] and PHOTOS [5] packages for τ-lepton decays (including spin correlations treatement) and QED radiations in decays of particles. At present, the following matrix-element-based processes have been implemented: $gg, q\bar{q} \rightarrow t\bar{t}b\bar{b}$, $q\bar{q} \rightarrow W(\rightarrow \ell\nu)b\bar{b}$; $q\bar{q} \rightarrow W(\rightarrow \ell\nu)t\bar{t}$; $gg, q\bar{q} \rightarrow Z/\gamma^*(\rightarrow \ell\ell)b\bar{b}$; $gg, q\bar{q} \rightarrow Z/\gamma^*(\rightarrow \ell\ell, \nu\nu, b\bar{b})t\bar{t}$; complete EW $gg, q\bar{q} \rightarrow (Z/W/\gamma^* \rightarrow)t\bar{t}b\bar{b}$; $gg, q\bar{q} \rightarrow t\bar{t}t\bar{t}$; $gg, q\bar{q} \rightarrow (t\bar{t} \rightarrow)f\bar{f}bf\bar{f}\bar{b}$; $gg, q\bar{q} \rightarrow (WWbb \rightarrow)f\bar{f}f\bar{f}b\bar{b}$. Both interfaces allow the use of the LHAPDF/LHAGLUE library of parton density functions. Provided is also set of *control processes*: $q\bar{q} \rightarrow W \rightarrow \ell\nu$; $q\bar{q} \rightarrow Z/\gamma^* \rightarrow \ell\ell$; $gg, q\bar{q} \rightarrow t\bar{t}$ and $gg \rightarrow (t\bar{t} \rightarrow)WbW\bar{b}$;
Restriction on the complexity of the problem: The package is dedicated for the 14 TeV pp collision simulated in the LHC environment. In particular, using it for the 2 TeV $p\bar{p}$ collision although technically possible might not be efficient and would require dedicated optimisation of the phase space generation. The consistency between results of the complete generation using PYTHIA 6.2 or HERWIG 6.5 interfaces is technically limited by the different approaches taken in both these generators for evaluating α_{QCD} and α_{QED} couplings and by the different models for fragmentation/hadronisation. For the consistency check, in the **AcerMC** library contains native coded definitions of the α_{QCD} and α_{QED}. Using these native definitions leads to the same total cross-sections both with PYTHIA 6.2 or HERWIG 6.5 interfaces.
Typical running time: On an PIII 800 MHz PC it amounts to $\sim 0.05 \rightarrow 1.1$ events/sec, depending on the choice of process.

[1]. T. Sjostrand et al., *High energy physics generation with PYTHIA 6.2*, eprint hep-ph/0108264, LU-TP 01-21, August 2001.

[2]. G. Marchesini et al., Comp. Phys. Commun. **67** (1992) 465, G. Corcella et al., JHEP **0101** (2001) 010.

[3]. E. Boos at al., *Generic user process interface for event generators*, hep-ph/0109068.

[4]. S. Jadach, J. H. Kuhn, Z. Was, Comput. Phys. Commun. **64** (1990) 275; M. Jezabek, Z. Was, S. Jadach, J. H. Kuhn, Comput. Phys. Commun. **70** (1992) 69; R. Decker, S. Jadach, J. H. Kuhn, Z. Was, Comput. Phys. Commun. **76** (1993) 361.

[5]. E. Barberio and Z. Was, Comp. Phys. Commun. **79** (1994) 291.

2 Changes since AcerMC 1.0

- **AcerMC version 1.1**: The changes include transition to HERWIG 6.4, updated scale choices for processes **5-8** (c.f. Section 7.8) and the inclusion of control processes 91-94 for consistent process evaluation (c.f. Section 4.9). Also, a possibility of an event dump according to the Les Houches standard was added (see Section 7.10).
- **AcerMC version 1.2**: A bug fix in HERWIG 6.4, affecting the shower evolution, was made. It carries no immediate impact on the AcerMC processes but was discovered by the AcerMC authors and added for the convenience of the users. This bug fix will be included in future versions of HERWIG. Also, the PYTHIA version was upgraded to PYTHIA 6.208 and the the implementation of storing/reading back of hard process events according to the Les Hauches standard was simplified with respect to **AcerMC 1.1** (see Section 7.10). Also, for the convenience of the users the Pythia code was modified so that the top decay products from **AcerMC** processes are now stored in the history part of the event record (status code 21) and have the correct pointers to the top quark they originate from. This feature will be added to the future versions of PYTHIA.
- **AcerMC version 1.3**: A transition to the HERWIG 6.5 was made, which now supports the Les Houches standard for handling the external processes. Consequently, the **AcerMC** interface to HERWIG was completely rewritten (c.f. Section 7.7). As a direct consequence, the same event record can freely be swapped (i.e. read back) to either PYTHIA or HERWIG for fragmentation and hadronisation treatment. In addition, the PYTHIA version was upgraded to PYTHIA 6.214. Also, the build procedure of the libraries and executables was greatly simplified (see Section 7.9).
- **AcerMC version 1.4**: The interfaces to external TAUOLA and PHOTOS libraries were added. The necessary modifications in the interface routines and the native PHOTOS code were made to enable the user to process the events with TAUOLA and/or PHOTOS using PYTHIA or HERWIG as the supervising generators.
- **AcerMC version 2.0**: **New algorithm for phase space generation implemented and optimised.** New processes were added: $q\bar{q} \to (Z/W/\gamma^* \to)t\bar{t}b\bar{b}$; $gg, q\bar{q} \to t\bar{t}t\bar{t}$; $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}bf\bar{f}\bar{b}$; $gg, q\bar{q} \to t\bar{t}$ and $gg \to (t\bar{t} \to)WbW\bar{b}$. New control channel added: $gg \to (t\bar{t} \to)WbW\bar{b}$.
- **AcerMC version 2.1**: Interface to ARIADNE implemented (current version 4.12).
- **AcerMC version 2.2**: Certain minor bug fixes implemented.
- **AcerMC version 2.3**: Another switch for fully leptonic boson pair decay (ACSET(13)=17) added.
- **AcerMC version 2.4**: Interface to LHAPDF/LHAGLUE implemented (current version 3).

3 Introduction

Despite a large repertoire of processes implemented for generation in the universal generators like PYTHIA [1] or HERWIG [2] a number of Standard Model background processes for studying expected physics potential of the LHC experiments is still missing. For some of these processes the matrix element expressions are rather lengthy and/or to achieve a reasonable generation efficiency it is necessary to tailor the phase-space selection procedure to the dynamics of the process. Due to this fact it cannot be expected that any of the universal purpose generators will contain *every* proces that is expected to participate at LHC pp collisions with an observable rate. A more practical solution could come in form of dedicated matrix-element-based generators with standardised interfaces, like the one proposed in [4], to the more general ones which are used to complete event generation.

The **AcerMC** Monte Carlo Event Generator follows up on this idea. It is dedicated for the simulation of the specific Standard Model background processes at LHC collisions: the $gg, q\bar{q} \to t\bar{t}b\bar{b}$, $q\bar{q} \to W(\to \ell\nu)b\bar{b}$; $q\bar{q} \to W(\to \ell\nu)t\bar{t}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell)b\bar{b}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$; complete EW $gg, q\bar{q} \to (Z/W/\gamma^* \to)t\bar{t}b\bar{b}$; 4 top-quark production $gg, q\bar{q} \to t\bar{t}t\bar{t}$; extended treatement of the 2 top-quark production $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}bf\bar{f}\bar{b}$ and $gg, q\bar{q} \to (WWbb \to)f\bar{f}f\bar{f}b\bar{b}$. They are characterised by the presence of the heavy flavour jets and multiple isolated leptons in the final state. For the Higgs boson searches, the $t\bar{t}H$, ZH, WH with $H \to b\bar{b}$, the $gg \to H$ with $H \to ZZ^* \to 4\ell$, the $b\bar{b}h/H/A$ with $h/H/A \to \tau\tau, \mu\mu$ are the most obvious examples of signals where the implemented processes would contribute to the dominant irreducible backgrounds. The same background processes should also be considered for e.g. estimating the observability of SUSY events with a signature of multi-b-jet and multi-lepton production.

The program itself provides library of the massive matrix elements and phase space modules for the generation of the implemented processes. The hard process event, generated with these modules, can be completed by the initial and final state radiation, hadronisation and decays, simulated with either PYTHIA 6.2, ARIADNE 4.1 [3] or HERWIG 6.5 Monte Carlo Event Generators. These will subsequently be called the *Supervising Generators*. Interfaces of **AcerMC** to PYTHIA 6.2, ARIADNE 4.1 and HERWIG 6.5 generators, are provided in the distribution version. Provided is also the interface to TAUOLA [6] and PHOTOS [5] packages, for the more correct treatement of the τ-lepton decays and photon radiation, than what available in the *Supervising Generators*. The **AcerMC** also uses several other external libraries: CERNLIB, HELAS [7], VEGAS [8]. The matrix element codes have been derived with the help of MADGRAPH [9] package. The achieved typical efficiency for the generation of unweighted events is of **20% - 30%**, rather high given a complicated topology of the implemented processes.

This paper superseeds the first version of the manual, published in [10]. The outline of this paper is as follows. In Section 3, we describe physics motivation for implementing each of the above processes and we collect some numerical results (plots, tables) which can be used as benchmarks. In Section 4 we describe the overall Monte Carlo algorithm. Section 5 gives details on the structure of the program. Section 6 collects information on how to use this program and existing interfaces to PYTHIA 6.2, ARIADNE 4.1 and HERWIG 6.5, TAUOLA and PHOTOS. Summary, Section 7, closes the paper. Appendix A documents sets of Feynman diagrams used for calculation of the matrix element for each subprocess, Appendices B and C give examples of the input/output of the program.

4 Physics content

The physics programme of the general purpose LHC experiments, ATLAS [15] and CMS [16], focuses on the searches for the *New Physics* with the distinctive signatures indicating production of the Higgs boson, SUSY particles, exotic particles, etc. The expected environment will in most cases be very difficult, with the signal to background ratio being quite low, on the level of a few percent after final selection in the signal window.

Efficient and reliable Monte Carlo generators, which allow one to understand and predict background contributions, are becoming the key point to the discovery. As the cross-section for signal events is rather low, even rare Standard Model processes might become the overwhelming background in such searches. In several cases, generation of such a process is not implemented in the general purpose Monte Carlo generators, when the complicated phase space behaviour requires dedicated (and often rather complex) pre-sampling, whilst the general purpose Monte Carlo generators due to a large number of implemented processes tend to use simpler (albeit more generic) phase space sampling algorithms. In addition, the matrix element for these processes is often lengthy and thus requiring complicated calculations. Only recently, with the appearance of modern techniques for automatic computations, their availability *on demand* became feasible for the tree-type processes. With the computation power becoming more and more easily available, even very complicated formulas can now be calculated within a reasonable time frame.

The physics processes implemented in **AcerMC** library represent such a set of cases. They are all being key background processes for the discovery in the channels characterised by the presence of the heavy flavour jets and/or multiple isolated leptons. For the Higgs boson searches, the $t\bar{t}H, ZH, WH$ with $H \to b\bar{b}$, the $gg \to H$ with $H \to ZZ^* \to 4\ell$, the $b\bar{b}h/H/A$ with $h/H/A \to \tau\tau, \mu\mu$ are the most obvious examples of such channels.

It is not always the case that the matrix element calculations in the lowest order for a given topology represent the total expected background of a given type. This particularly concerns the heavy flavour content of the event. The heavy flavour in a given event might occur in the hard process of a much simpler topology, as the effect of including higher order QCD corrections (eg. in the shower mechanism). This is the case for the b-quarks present in the inclusive Z-boson or W-boson production, which has a total cross-section orders of magnitude higher than the discussed matrix-element-based $Wb\bar{b}$ or $Zb\bar{b}$ production. Nevertheless, the matrix-element-based calculation is a very good reference point to compare with parton shower approaches in different fragmentation/hadronisation models. It also helps to study matching procedures between calculations in a fixed α_{QCD} order and parton shower approaches. For very exclusive hard topologies matrix-element-based calculations represent a much more conservative approximation than the parton shower ones [17].

Let us shortly discuss the motivation for these few Standard Model background processes which are implemented in the **AcerMC 2.0** library.

The $t\bar{t}b\bar{b}$ production at LHC is a dominant irreducible background for the Standard Model (SM) and Minimal Supersymmetric Standard Model (MSSM) Higgs boson search in the associated production, $t\bar{t}H$, followed by the decay $H \to b\bar{b}$. The potential for the observability of this channel has been carefully studied and documented in [15] and [18]. Proposed analysis requires identifying four b-jets, reconstruction of both top-quarks in the hadronic and leptonic mode and visibility of the peak in the invariant mass distribution of the remaining b-jets. The irreducible $t\bar{t}b\bar{b}$ background contributes about 60-70% of the total background from the $t\bar{t}$ events ($t\bar{t}b\bar{b}$, $t\bar{t}bj$, $t\bar{t}jj$).

The $Wb\bar{b}$ production at LHC is recognised as a substantial irreducible background for the Standard Model (SM) and Minimal Supersymmetric Standard Model (MSSM) Higgs boson search in the associated production, WH, followed by the decay $H \to b\bar{b}$. The **AcerMC** library discussed here includes even more efficient implementation of the algorithm presented in [11].

The $Wt\bar{t}$ production at LHC has to our knowledge not been implemented in the publicly available code so far[1]. It is of interest because it contributes an overwhelming background [14] for the measurement of the Standard Model Higgs self-couplings at LHC in the most promising channel $pp \to HH \to WWWW$.

The $Z/\gamma^*(\to \ell\ell)b\bar{b}$ production at LHC has since several years been recognised as one of the most substantial irreducible (or reducible) backgrounds for the several Standard Model (SM) and Minimal Supersymmetric Standard Model (MSSM) Higgs boson decay modes as well as for observability of the SUSY particles. There is a rather wide spectrum of *regions of interest* for this background. In all cases the leptonic Z/γ^* decay is asked for, but events with di-lepton invariant mass around the mass of the Z-boson mass or with the masses above or below the resonance peak could be of interest. The presented process enters an analysis either by the accompanying b-quarks being tagged as b-jets, or by the presence of leptons from the b-quark semi-leptonic decays in these events, in both cases thus contributing to the respective backgrounds.

Good understanding of this background, and having a credible Monte Carlo generator available, which allows studying of expected acceptances for different final states topologies, is crucial for several analyses at LHC. The **AcerMC** library

[1] We thank M. L. Mangano for bringing this process to our attention and for providing benchmark numbers for verifying the total cross-section.

discussed here includes more efficient implementation of the algorithm presented in [12].

The $Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$ production at LHC is an irreducible background to the Higgs search in the invisible decay mode (case of $Z \to \nu\nu$) in the production with association to the top-quark pair [29]. With the $Z/\gamma^*(\to b\bar{b})$ it is also an irreducible resonant background to the Higgs search in the $t\bar{t}H$ production channel but with the Higgs boson decaying to the b-quark pair [18].

The complete **EW production** of the $gg, q\bar{q} \to (Z/W/\gamma^* \to)b\bar{b}t\bar{t}$ final state is also provided. It can be considered as a benchmark for the previous process, where only the diagrams with resonant $gg, q\bar{q} \to (Z/\gamma^* \to)b\bar{b}t\bar{t}$ are included. It thus allows the verification of the question, whether the EW resonant contribution is sufficient in case of studying the $t\bar{t}b\bar{b}$ background away from the Z-boson peak, like for the $t\bar{t}H$ with Higgs-boson mass of 120 GeV.

The $gg, q\bar{q} \to t\bar{t}t\bar{t}$ production , interesting process per se, is a background to the possible Higgs self-coupling measurement in the $gg \to HH \to WWWW$ decay, [14].

The $gg, q\bar{q} \to (WWbb \to)f\bar{f}f\bar{f}b\bar{b}$ and $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}bf\bar{f}\bar{b}$ processes give posiblity to study spin correlations in the top-quark pair production and decays as well as the effect from the off-shell production. Those are important for the selection optimisation eg. in the $gg \to H \to WW$ channel, see the discussion in [36]

A set of **control channels**, i.e. the $q\bar{q} \to Z/\gamma^* \to \ell\ell$, $gg, q\bar{q} \to t\bar{t}$, $q\bar{q} \to W \to \ell\nu$ and $gg \to (t\bar{t} \to)WbW\bar{b}$ processes, have been added to **AcerMC** in order to provide a means of consistency and cross-check studies.

This completes the list of the native **AcerMC** processes implemented so far. Having all these different production processes implemented in the consistent framework, which can also be directly used for generating standard subprocesses implemented in either PYTHIA or HERWIG Monte Carlo, represents a very convenient environment for several phenomenological studies dedicated to the LHC physics.

For the cases, where radiative photon emission from final state leptons is important the package PHOTOS [5] can be used in the chain of event generation. In similar way also package TAUOLA [6] can be interfaced directly to the generation chain and used for events generation in cases where more detailed treatment of the tau-lepton decay and including spin correlations effects is relevant.

In the following subsections we discuss in more detail implementation of each subprocess. We also give benchmark Tables with the total cross-sections obtained with AcerMC processes but different implementations and setting of $\alpha_{\rm QCD}(Q_{QCD})$: the native AcerMC, PYTHIA and HERWIG ones. For a more detailed discussion on this topic the reader is referred to Section 5.6. If the native **AcerMC** definition is used, the same cross-section is obtained either with PYTHIA or HERWIG generation chains.

4.1 The $gg, q\bar{q} \to t\bar{t}b\bar{b}$ processes

In the implementation discussed here, the matrix element was derived using the MADGRAPH package [9]. These matrix elements are not covering the decay of the top-quarks, the latter are considered as massive final states of the process. The top-quark decays is than performed by the supervising generator. Rather important spin effects (spin correlations) in the top decays are therefore not yet included. The similar solution, like for tau decay in the Z-boson production process discussed in [30], is planned to be implemented here in the near future.

As a benchmark, the processes $gg, q\bar{q} \to t\bar{t}b\bar{b}$ have been simulated for pp collisions with 14 TeV centre-of-mass energy and CTEQ5L [31] parton density functions, using event generation with massive $2 \to 4$ matrix element implemented as an external process to PYTHIA 6.2 (see Section 4 and 5). The decays of the top-quarks have been left under control of PYTHIA 6.2 generator. The $q\bar{q} \to t\bar{t}b\bar{b}$ subprocess contributes less than 10% of the total cross-section.

The total cross-section is very sensitive to the choice of the QCD energy scale used for calculation of that process, thus indicating potentially large contributions from higher order corrections. The same definition for the factorisation and renormalisation scale is used. The example values of the total cross-section for implemented choices of the QCD energy scale are given in Table **1**[2].

As a cross-check, the processes $gg, q\bar{q} \to t\bar{t}b\bar{b}$ have been coded independently using the COMPHEP package [32]. The same set of diagrams was selected and only the integrating part of the package was used to calculate total cross-section. The choices for the QCD energy scale were kept consistent. A very good agreement between the cross-sections obtained with two independent calculation streams prepared for this study has been achieved [13].

One can observe a very strong scale dependence of the cross-section for the $gg, q\bar{q} \to t\bar{t}b\bar{b}$ process (c.f. Table **1**). Factor four (!!) can be expected on the predicted cross-section when changing from the scale $Q^2_{QCD} = \hat{s}$ to the scale $Q^2_{QCD} = <$

[2]Numbers obtained with HERWIG generator (third collumn) are slightly different than what published in [10]. This is related to the internal changes in HERWIG between version 6.3 and 6.5.

Table 1: **AcerMC** *cross-sections for the* $gg, q\bar{q} \rightarrow t\bar{t}b\bar{b}$ *production at different choices of the QCD energy scale and* α_{QCD} *implementations. The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used for the simulation with interfaces to* PYTHIA 6.2 *and* HERWIG 6.5. *The* $m_H = 120$ *GeV and* $m_t = 175$ *GeV were used for calculating the* Q^2_{QCD} *in the last row of this table. The default settings of* α_{QCD} *as implemented in* **AcerMC**, PYTHIA 6.2 *and* HERWIG 6.5 *were used.*

Factorisation scale	α_{QCD} (1L) native AcerMC	α_{QCD} (1L) as in PYTHIA 6.2	α_{QCD} (2L) as in HERWIG 6.5
	$gg \rightarrow t\bar{t}b\bar{b}$		
$Q^2_{QCD} = \hat{s}$	4.2 [pb]	3.9 [pb]	3.0 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2 + m_i^2)/4$	10.3 [pb]	10.2 [pb]	7.2 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2)/4$	17.0 [pb]	16.9 [pb]	11.5 [pb]
$Q^2_{QCD} = (m_t + m_H/2)^2$	8.2 [pb]	8.1 [pb]	5.8 [pb]
	$q\bar{q} \rightarrow t\bar{t}b\bar{b}$		
$Q^2_{QCD} = \hat{s}$	0.30 [pb]	0.29 [pb]	0.22 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2 + m_i^2)/4$	0.61 [pb]	0.60 [pb]	0.43 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2)/4$	0.91 [pb]	0.90 [pb]	0.62 [pb]
$Q^2_{QCD} = (m_t + m_H/2)^2$	0.52 [pb]	0.51 [pb]	0.37 [pb]

$p_T^2 >$. This very strong dependence on the energy scale is also observed in the case of the $t\bar{t}H$ production, for recent discussion see [33]. There, the recommended *central* factorisation and renormalisation energy scale is $\mu_0 = (m_t + m_H/2)$. Having in mind that the primary interest of evaluating this background is the Higgs search in the $t\bar{t}H$ production, i.e. with the b-quark system being produced with the invariant mass of the expected Higgs boson, we have also introduced this *central* energy scale, with $m_H = 120$ GeV as one of the possible choices.

Fig. **1** shows the distributions of the $Q_{QCD} = \sqrt{Q^2_{QCD}}$ (distributions have been normalised to one) for the $t\bar{t}b\bar{b}$ events with the invariant mass of the b-jets system, calculated using the default PYTHIA (LO) $\alpha_{\rm QCD}$ implementation[3]. $m_{bb-jets} = 120 \pm 30$ GeV. The distribution is well collimated around the average value when Q^2_{QCD} is defined as $< m_T^2 >$ or $< p_T^2 >$ while it is much broader when Q^2_{QCD} is defined as $\hat{s}$. The kinematic distributions are very similar in shape for separate $gg \rightarrow t\bar{t}b\bar{b}$ and $q\bar{q} \rightarrow t\bar{t}b\bar{b}$ contributions. The total cross-section for accepted events as a function of the averaged Q^2_{QCD} (for these events) is shown in the bottom plot. It can be noted that the cross-section decreases rather fast with the increasing value of the average $< Q^2_{QCD} >$. Also shown is the $\alpha_s^4(Q_{QCD})$ dependence scaled to match the cross-section at $Q_{QCD} = (m_H/2 + m_t)$ with $m_H = 120$ GeV, it being the only calculated cross-section point with a fixed scale. The other cross-sections are shown to follow the expected $\alpha_s^4(Q_{QCD})$ dependence rather well, while the deviations are induced by the parton density function dependence on the Q^2_{QCD} scale, most notably at $Q^2_{QCD} = \hat{s}$ value. The deviations induced by the parton density functions dependence on the Q^2_{QCD} scale are different for the gg and $q\bar{q}$ contributions, as can be concluded from results given in Table **1**.

The series of plots illustrating the most relevant differential distributions for the top-quarks and b-quarks can be found in [13].

4.2 The $q\bar{q} \rightarrow W(\rightarrow \ell\nu)g^*(\rightarrow b\bar{b})$ process

The matrix element for the implemented process was coded by using the MADGRAPH package [9]. This process is represented by only two Feynman diagrams, with quark exchange in the t-channel, leading to the production of the W-boson and virtual gluon splitting into $b\bar{b}$ pair. Only the u, d, s, c quarks were considered in this implementation, the possibility of the b-quark in the initial state was omitted as expected to be negligible numerically (e.g.. $|V_{bc}/V_{ud}|^2 \sim 0.002$) but leading to several additional diagrams which would have to be included. The massive matrix element takes into account spin correlations in the W-boson decay and angular correlations between leptons and quarks. Due to the massive treatment of the final state

[3] This would makes distributions directly relevant for the $t\bar{t}H$ analysis. For details on the jet reconstruction see [34].

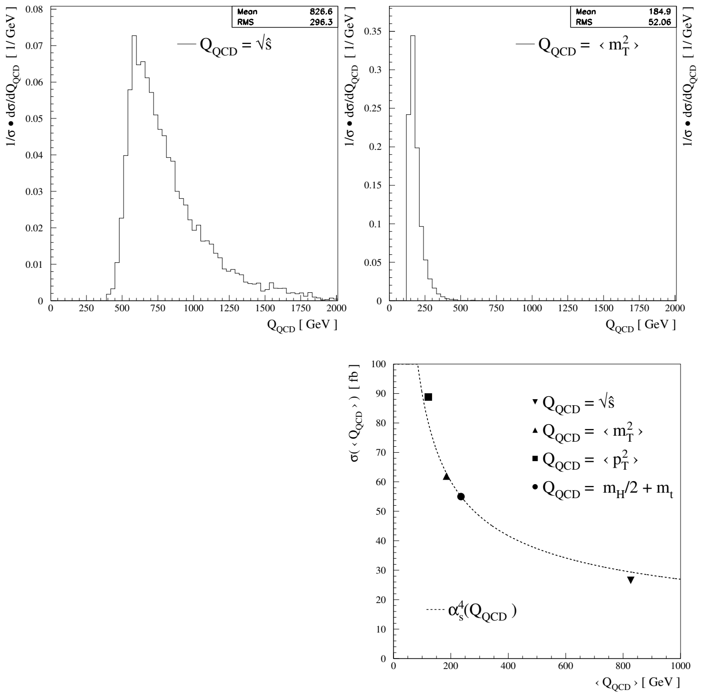


Fig. 1: *Top: the Q^2_{QCD} distributions for $t\bar{t}b\bar{b}$ events with the invariant mass of the b-jets system $m_{bb-jets} = 120 \pm 30$ GeV. Bottom: the total cross-section of accepted events as a function of the averaged Q^2_{QCD} (for these events).*

fermions the amplitude has no singularities; the total cross-section is well defined. The effect from the W-boson natural width and the W-boson propagator are also properly included.

Table 2: **AcerMC** *production cross-sections for the $q\bar{q} \to Wb\bar{b}$ with $W \to e\nu$ decay (single flavour). The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used with different definitions of α_{QED}, α_{QCD} (as in default* PYTHIA 6.2 *and* HERWIG 6.5*) and several choices of the factorisation scale, α_{QED} and α_{QCD} implementations.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}$ (1L) native AcerMC	$\alpha_{QED}, \alpha_{QCD}$ (1L) as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}$ (2L) as in HERWIG 6.5
$Q^2 = M_W^2$	36.5 [pb]	36.4 [pb]	30.6 [pb]
$Q^2 = s^*_{b\bar{b}}$	44.1 [pb]	44.0 [pb]	36.0 [pb]
$Q^2 = M_W^2 + pT_W^2$	36.0 [pb]	36.0 [pb]	29.8 [pb]
$Q^2 = (s^*_W + s^*_{b\bar{b}})/2 + pT_W^2$	37.2 [pb]	37.1 [pb]	30.4 [pb]

As a benchmark, the process $q\bar{q} \to W(\to \ell\nu)g^*(\to b\bar{b})$ has been simulated for pp collision with 14 TeV centre-of-mass energy. The total cross-section, including branching ratio for $W \to \ell\nu$ (single flavour) is 36.5 pb (CTEQ5L parton density functions, $Q^2 = M_W^2$, PYTHIA 6.2 interface)[4].

The dependence on the choice of the factorisation scale is rather modest (c.f. Table **2**) and does not exceed 20% for the choices implemented in **AcerMC** library. The variation of the cross-section due to different α_{QED} and α_{QCD} implementations and default settings is again evident; as one can expect the two-loop α_{QCD} implementation given in HERWIG gives a $\sim$20 % lower cross-section when compared to the cases when native **AcerMC** and PYTHIA one-loop α_{QCD} were used[5].

The differential distributions of the $q\bar{q} \to Wb\bar{b}$ events turn out to be interesting when compared to the corresponding ones of the $q\bar{q} \to Zb\bar{b}$ and $gg \to Zb\bar{b}$ events (generated with pure Z-boson exchange). Such comparison is well documented in [12].

4.3 The $q\bar{q} \to W(\to \ell\nu)g^*(\to t\bar{t})$ process

The $2 \to 4$ matrix elements, coded by the MADGRAPH package [9], are not covering the decay of the top-quarks; the latter are considered as massive final states of the process. The top decay is than performed by the supervising generator. As in the case of $gg, q\bar{q} \to t\bar{t}b\bar{b}$ process spin effects in the top decays are therefore not yet included. This process, although rare, contributes an overwhelming irreducible background to possible measurement of the Higgs-boson self-coupling in the $HH \to WWWW$ decay mode [14].

Table 3: **AcerMC** *production cross-sections for the $q\bar{q} \to Wt\bar{t}$ with primary $W \to e\nu$ decay (single flavour). The 14 TeV centre-of-mass energy, CTEQ5L parton density functions with different factorisation scales and different definitions of the α_{QED} and α_{QCD} were used in the matrix element calculations.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}$ (1L) native AcerMC	$\alpha_{QED}, \alpha_{QCD}$ (1L) as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}$ (2L) as in HERWIG 6.5
$Q^2_{QCD} = M^2_W$	69.3 [fb]	69.1 [fb]	57.4 [fb]
$Q^2_{QCD} = s^*_{t\bar{t}}$	40.9 [fb]	39.9 [fb]	34.7 [fb]
$Q^2_{QCD} = M^2_W + pT^2_W$	59.7 [fb]	59.5 [fb]	49.6 [fb]
$Q^2_{QCD} = (s^*_W + s^*_{t\bar{t}})/2 + pT^2_W$	43.7[fb]	42.8 [fb]	36.9 [fb]

Table **3** shows the expected **AcerMC** cross-sections for different choices of the energy scale and coupling ($\alpha_{QED}, \alpha_{QCD}$) definitions. One should notice the effect of almost a factor two from different choices of the energy scale.

4.4 The $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell)b\bar{b}$ processes

The matrix elements, derived using the MADGRAPH package [9], properly take into account spin correlations in the Z-boson decay and angular correlations between leptons and quarks. Thank to keeping non-zero b-quark masses the amplitude has no singularities; the total cross-section is well defined.

The full Z/γ^* exchange proves to be important: For events well below the Z-boson resonance the contribution from γ^* becomes dominant; the γ^* contribution is also sizeable in the high mass tail and increases proportionally with the effective mass of the di-lepton system.

As a benchmark result, the process has been simulated for pp collisions at 14 TeV centre-of-mass energy. The total cross-sections, including the branching ratio for $Z/\gamma^* \to \ell\ell$ (single flavour) are given in Table **4** for different definitions of α_{QED}, α_{QCD} couplings.

Several differential benchmark distributions for leptons and b-quarks originating from the hard process has been collected and discussed in [11].

[4] This can be compared with the matrix element implementation to HERWIG 5.6, used in [35],[37], where originally this cross-section was estimated to 19.8 pb (CTEQ2L parton density functions) but, when implementing CTEQ5L parton density functions and setting kinematic parameters to be in approximate accordance with PYTHIA defaults, rises to 36.0 pb, which is consistent with the AcerMC implementation by taking into account the remaining differences in the two calculations (e.g. the former implementation uses an on-shell W boson in the ME calculation).

[5] While performing further comparisons of native **AcerMC** and PYTHIA processes we discovered a misinterpretation of our CKM matrix implementation. This correction efectively changes the cross-section for $q\bar{q} \to Wb\bar{b}$ and $q\bar{q} \to Wt\bar{t}$ processes by $\sim$10% compared to the draft versions of this paper, which is nevertheless still well within the physics precision of the program. The affected tables in this paper are already updated.

Table 4: **AcerMC** *production cross-sections for the $gg, q\bar{q} \to Z/\gamma^* b\bar{b}$ with $Z/\gamma^* \to ee$ decay (single flavour). The 14 TeV centre-of-mass energy, CTEQ5L parton density functions and different definitions for the α_{QED}, α_{QCD} (as in default* PYTHIA 6.2 *and* HERWIG 6.5*) were used in the matrix element calculations. The threshold $m_{\ell\ell} \geq 10$ GeV was used in the event generation.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
		$gg \to Z/\gamma^* bb$	
$Q^2 = m_Z^2$	49.5 [pb]	45.8 [pb]	38.0 [pb]
$Q^2 = s^*_{b\bar{b}}$	53.8 [pb]	53.9 [pb]	44.0 [pb]
$Q^2 = s^*_Z$	54.7 [pb]	54.6 [pb]	44.4 [pb]
$Q^2 = (pT_Z^2 + s^*_{b\bar{b}})/2$	49.7 [pb]	49.7 [pb]	40.5 [pb]
		$q\bar{q} \to Z/\gamma^* bb$	
$Q^2 = m_Z^2$	6.7 [pb]	6.7 [pb]	5.6 [pb]
$Q^2 = s^*_{b\bar{b}}$	8.0 [pb]	8.0 [pb]	6.4 [pb]
$Q^2 = s^*_Z$	7.0 [pb]	7.0 [pb]	5.7 [pb]
$Q^2 = (pT_Z^2 + s^*_{b\bar{b}})/2$	6.9 [pb]	6.9 [pb]	5.7 [pb]

4.5 The $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$ processes

This process, in spite of having a very small cross-section at LHC energies, contributes as irreducible background to the $t\bar{t}H$ production at low masses. In case the Higgs boson is searched within the $H \to b\bar{b}$ mode, this contribution becomes less and less important with the Higgs boson mass moving away from the Z-boson mass. In case of the Higgs-boson search in the invisible decaying mode, the $Z \to \nu\nu$ might be more relevant also for the higher masses, as the mass peak cannot be reconstructed for signal events. The $Z/\gamma^* \to \ell\ell$ decay is of less interest, as the expected observability at LHC is very low (Table **5**).

Table 5: **AcerMC** *production cross-sections for the $gg, q\bar{q} \to Zt\bar{t}$ with $Z \to \nu_e\nu_e$ decay (3 flavours). The 14 TeV centre-of-mass energy, CTEQ5L parton density functions and different definitions for the α_{QED}, α_{QCD} (as in native* **AcerMC**, *default* PYTHIA 6.2 *and* HERWIG 6.5*) were used.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
		$gg \to Z(\to \nu_e\nu_e)tt$	
$Q^2 = m_Z^2$	126.0 [fb]	125.8 [fb]	104. [fb]
$Q^2 = s^*_{t\bar{t}}$	61.9 [fb]	60.3 [fb]	52.4 [fb]
$Q^2 = s^*_Z$	126.2 [fb]	126.1 [fb]	105. [fb]
$Q^2 = pT_Z^2 + s^*_{t\bar{t}})/2$	67.6 [fb]	66.3 [fb]	57.0 [fb]
		$q\bar{q} \to Z(\to \nu_e\nu_e)tt$	
$Q^2 = m_Z^2$	64.7 [fb]	64.6 [fb]	53.7 [fb]
$Q^2 = s^*_{t\bar{t}}$	39.2 [fb]	38.2 [fb]	33.3 [fb]
$Q^2 = s^*_Z$	64.8 [fb]	64.7 [fb]	53.5 [fb]
$Q^2 = (pT_Z^2 + s^*_{t\bar{t}})/2$	41.7 [fb]	41.0 [fb]	35.2 [fb]

4.6 The electroweak $gg, q\bar{q} \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$ process

One should be well aware, that the $gg, q\bar{q} \to Z/\gamma^* t\bar{t}$ with $Z/\gamma^* \to b\bar{b}$ does not represent a complete electroweak production of the $t\bar{t}b\bar{b}$ final state. Consequently, a separate implementation for generation of the complete set of such diagrams (including as well W-boson exchange) was addressed. In fact this final state leads to complicated pattern of the 72 Feynman diagrams (in case of the gg initial state).

The contribution from all non-resonant channels is a dominant one for the inclusive cross-section, see Table **6**. An almost factor 10 higher cross-section is calculated with the full electroweak $gg \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$ with respect to calculated with the $gg \to (Z/\gamma^* \to b\bar{b})t\bar{t}$ process only. One should also note that the electroweak $gg \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$ inclusive cross-section is on the level of 10% of the QCD $gg \to b\bar{b}t\bar{t}$ cross-section, see Table **1**, for the same choice of the energy scale. But in the mass range around 120 GeV it is on the level of 50% of the QCD contribution, [10].

Table 6: **AcerMC** *production cross-sections for the electroweak* $gg, q\bar{q} \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$*. The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used along with different definitions for the* α_{QED}, α_{QCD} *(as in native* **AcerMC**, *default* PYTHIA 6.2 *and* HERWIG 6.5*). The* $m_H = 120$ *GeV was used for calculation of the energy scale.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
		$gg \to (Z/W/\gamma^* \to) bbtt$	
$Q^2_{QCD} = \hat{s}$	0.58 [pb]	0.56 [pb]	0.50 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2 + m_i^2)/4$	1.10 [pb]	1.05 [pb]	0.84 [pb]
$Q^2_{QCD} = \sum ({p_T^i}^2)/4$	1.50 [pb]	1.50 [pb]	1.16 [pb]
$Q^2_{QCD} = (m_t + m_H/2)^2$	0.90 [pb]	0.89 [pb]	0.71 [pb]
		$q\bar{q} \to (Z/W/\gamma^* \to) bbtt$	
$Q^2_{QCD} = \hat{s}$	0.029 [pb]	0.029 [pb]	0.025 [pb]
$Q^2 = \sum ({p_T^i}^2 + m_i^2)/4$	0.043 [pb]	0.042 [pb]	0.036 [pb]
$Q^2 = \sum ({p_T^i}^2)/4$	0.049 [pb]	0.048 [pb]	0.040 [pb]
$Q^2 = (m_t + m_H/2)^2$	0.041 [pb]	0.041 [pb]	0.035 [pb]

4.7 The $gg, q\bar{q} \to (WWb\bar{b} \to) f\bar{f}f\bar{f}b\bar{b}$; $gg, q\bar{q} \to (t\bar{t} \to) f\bar{f}f\bar{f}b\bar{b}$ processes

The implemented $2 \to 6$ matrix elements for the resonant $gg, q\bar{q} \to (t\bar{t} \to) f\bar{f}f\bar{f}b\bar{b}$ and complete $gg, q\bar{q} \to (WWb\bar{b} \to) f\bar{f}f\bar{f}b\bar{b}$ processes give possibility to study background from top-quark production in more details, than with resonant on-shell $t\bar{t}$ production only (as implemented in PYTHIA and HERWIG). In particular, for the Higgs boson search in the $H \to WW \to \ell\nu\ell\nu$ decay channel, the analysis foresees strong suppression against $t\bar{t}$ background using topological feaures of events (jet veto, lepton angluar correlations), but does not foresees implicit top-quarks reconstruction. To reliably predict such backgrounds, availability of the complete $2 \to 6$ matrix element in Monte Carlo is mandatory, see eg. discussion in [36]. The total cross-sections are given in Table **7** for different definitions of $\alpha_{QED}, \alpha_{QCD}$ couplings.

As an example, Fig. **2** illustrates spin correlation effects in the top-pair production and decays, namely asymmetry in the correlations between lepton and antylepton direction in the rest frame of top-quark, for events generated with $2 \to 6$ matrix element. Such correlation is absent if only $2 \to 2$ matrix element is used for events generation, followed by the independent decays of each top-quark.

Table 7: **AcerMC** *production cross-sections for the* $gg, q\bar{q} \to (WWbb \ \to) f\bar{f} f\bar{f} b\bar{b}$ *process. The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used along with different definitions for the* α_{QED}*,* α_{QCD} *(as in native* **AcerMC***, default* PYTHIA 6.2 *and* HERWIG 6.5*).*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
	$gg \to (WWbb \ \to) ffffbb$		
$Q^2_{QCD} \ = \ (2 \cdot m_t^2)$	400. [pb]	390. [pb]	330. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2 + m_i^2)/4$	450. [pb]	450. [pb]	380. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2)/2$	550. [pb]	550. [pb]	460. [pb]
$Q^2_{QCD} \ = \ \hat{s}$	355. [pb]	350. [pb]	300. [pb]
	$q\bar{q} \to (WWbb \ \to) ffffbb$		
$Q^2_{QCD} \ = \ (2 \cdot m_t^2)$	63. [pb]	62. [pb]	53. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2 + m_i^2)/2$	69. [pb]	69. [pb]	58. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2)/2$	78. [pb]	78. [pb]	65. [pb]
$Q^2_{QCD} \ = \ \hat{s}$	59. [pb]	58. [pb]	50. [pb]

Table 8: **AcerMC** *production cross-sections for the* $gg, q\bar{q} \to (t\bar{t} \ \to) f\bar{f} b f\bar{f} \bar{b})$*. The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used along with different definitions for the* α_{QED}*,* α_{QCD} *(as in native* **AcerMC***, default* PYTHIA 6.2 *and* HERWIG 6.5*).*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
	$gg \to (t\bar{t} \to) ffbffb$		
$Q^2_{QCD} \ = \ (2 \cdot m_t^2)$	370. [pb]	365. [pb]	310. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2 + m_i^2)/2$	425. [pb]	420. [pb]	355. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2)/2$	512. [pb]	510. [pb]	425. [pb]
$Q^2_{QCD} \ = \ \hat{s}$	330. [pb]	320. [pb]	280. [pb]
	$q\bar{q} \to t\bar{t} \to) ffbffb$		
$Q^2_{QCD} \ = \ (2 \cdot m_t^2)$	63. [pb]	62. [pb]	53. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2 + m_i^2)/2$	69. [pb]	68. [pb]	58. [pb]
$Q^2_{QCD} \ = \ \sum ({p_T^i}^2)/2$	78. [pb]	78. [pb]	65. [pb]
$Q^2_{QCD} \ = \ \hat{s}$	59. [pb]	57. [pb]	50. [pb]

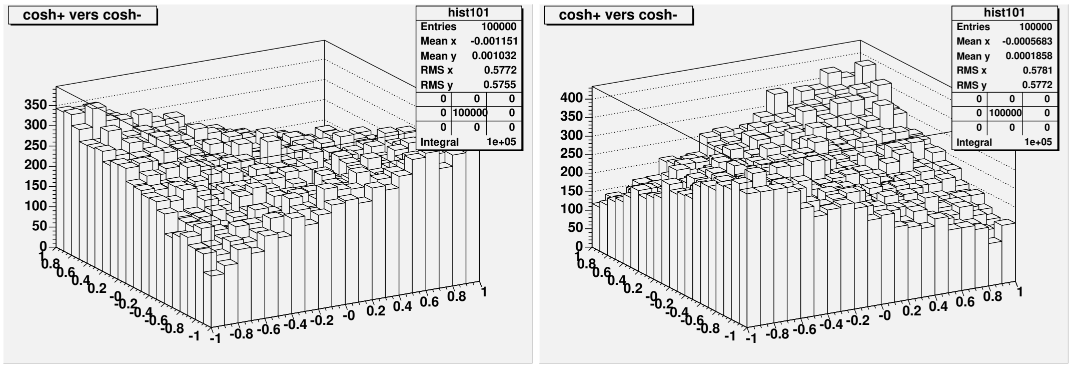


Fig. 2: *The correlations between $\cos\Theta$ (azimutal angle) of lepton and antylepton from $t\bar{t} \to \ell\bar{\nu}b\bar{\ell}\nu\bar{b}$ decays measured in the rest frame of the top-quark with respect to the anty-top quark direction. Left plot is for $gg \to (WWb\bar{b} \to)f\bar{f}f\bar{f}b\bar{b}$ process, right plot for $q\bar{q} \to (WWb\bar{b} \to)f\bar{f}f\bar{f}b\bar{b}$ process.*

4.8 The $gg, q\bar{q} \to t\bar{t}t\bar{t}$ process

This process, in spite of having a very small cross-section at LHC energies, contributes as reducible background to the $HH \to WWWW$ production at low masses, [14]. Availability of the complete Monte Carlo generator is mandatory to give reliable predictions of theis background and to optimise selection criteria.

Table 9: **AcerMC** *production cross-sections for the $gg, q\bar{q} \to t\bar{t}t\bar{t}$ process. The 14 TeV centre-of-mass energy and CTEQ5L parton density functions were used along with different definitions for the α_{QED}, α_{QCD} (as in native* **AcerMC**, *default* PYTHIA 6.2 *and* HERWIG 6.5*). The $m_H = 120$ GeV and $m_t = 175$ GeV were used for calculating the Q^2_{QCD}.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
	$gg \to tttt$		
$Q^2_{QCD} = \hat{s}$	26.5 [fb]	24.4 [fb]	19.3 [fb]
$Q^2_{QCD} = \sum(p_T^{i\ 2} + m_i^2)/4$	75.7 [fb]	73.8 [fb]	53.4 [fb]
$Q^2_{QCD} = \sum(p_T^{i\ 2})/4$	94.7 [fb]	93.2 [fb]	66.2 [fb]
$Q^2_{QCD} = (m_t + m_H/2)^2$	89.5 [fb]	87.8 [fb]	62.9 [fb]
	$q\bar{q} \to tttt$		
$Q^2_{QCD} = \hat{s}$	0.5 [fb]	0.5 [fb]	0.4 [fb]
$Q^2_{QCD} = \sum(p_T^{i\ 2} + m_i^2)/4$	1.2 [fb]	1.2 [fb]	0.9 [fb]
$Q^2_{QCD} = \sum(p_T^{i\ 2})/4$	1.5 [fb]	1.5 [fb]	1.0 [fb]
$Q^2_{QCD} = (m_t + m_H/2)^2$	1.5 [fb]	1.4 [fb]	1.0 [fb]

4.9 The *control channel* processes

The set of simple $2 \rightarrow 2$ control channel processes was added to **AcerMC** in order to provide a means of consistency and cross-check studies. Although these processes are already implemented in PYTHIA and/or HERWIG (except the $gg \rightarrow WbWb$ one), the availability of the native implementations is supposed to offer a more consistent control of generation parameters when performing e.g. the comparison of parton shower PYTHIA/HERWIG produced additional pair of heavy quarks with the exact leading-order matrix elements implemented in the core group of the **AcerMC** $2 \rightarrow 4$ processes.

To the list of control channel processes we have added also $2 \rightarrow 4$ process, the $gg \rightarrow WbWb$ process, as a control channel for the $2 \rightarrow 6$ process $gg \rightarrow f\bar{f}bf\bar{f}\bar{b}$. The $2 \rightarrow 4$ process we consider as very usefull for studying in more detail the resonant and complete $WbWb$ production at LHC.

The benchmark results, given in Table **10**, are obtained for simulated pp collisions at 14 TeV centre-of-mass energy. The total cross-sections are listed for different definitions of α_{QED}, α_{QCD} couplings and different definitions of the energy scale Q^2.

Table 10: **AcerMC** *production cross-sections for the* $q\bar{q} \rightarrow Z/\gamma^* \rightarrow \ell\ell$, $gg, q\bar{q} \rightarrow t\bar{t}$ *and* $q\bar{q} \rightarrow W \rightarrow \ell\nu$ *with single flavour* $Z/\gamma^* \rightarrow \ell\ell$ *and/or* $W \rightarrow \ell\nu$ *decays. The 14 TeV centre-of-mass energy, CTEQ5L parton density functions and different definitions for the* α_{QED}, α_{QCD} *(as in native* **AcerMC**, *default* PYTHIA 6.2 *and* HERWIG 6.5*) were used. The threshold* $m_{\ell\ell} \geq 60$ *GeV was used in the event generation. In* PYTHIA 6.2 *the setting* MSTU(115)=2 *was used to set the lower* Q^2 *limit in* α_{QCD} *evolution to* $4\ \mathrm{GeV}^2$ *as done in the native* **AcerMC** *implementation.*

Factorisation scale	$\alpha_{QED}, \alpha_{QCD}(1L)$ native AcerMC	$\alpha_{QED}, \alpha_{QCD}(1L)$ as in PYTHIA 6.2	$\alpha_{QED}, \alpha_{QCD}(2L)$ as in HERWIG 6.5
	$q\bar{q} \rightarrow Z/\gamma^* \rightarrow \ell\ell$		
$Q^2 = \hat{s}$	1620 [pb]	1630 [pb]	1630 [pb]
$Q^2 = (\sum (p_T^{i\,2}) + M_Z^2)/2$	1550 [pb]	1560 [pb]	1560 [pb]
$Q^2 = \sum (p_T^{i\,2})/2$	1260 [pb]	1260 [pb]	1260 [pb]
$Q^2 = M_Z^2$	1630 [pb]	1630 [pb]	1640 [pb]
	$gg \rightarrow tt$		
$Q^2 = \hat{s}$	365 [pb]	360 [pb]	310 [pb]
$Q^2 = \sum (p_T^{i\,2} + m_i^2)/2$	430 [pb]	420 [pb]	355 [pb]
$Q^2 = \sum (p_T^{i\,2})/2$	595 [pb]	590 [pb]	490 [pb]
$Q^2 = (2m_t)^2$	320 [pb]	315 [pb]	270 [pb]
	$q\bar{q} \rightarrow tt$		
$Q^2 = \hat{s}$	62. [pb]	61. [pb]	52. [pb]
$Q^2 = \sum (p_T^{i\,2} + m_i^2)/2$	69. [pb]	68. [pb]	58. [pb]
$Q^2 = \sum (p_T^{i\,2})/2$	86. [pb]	85. [pb]	71. [pb]
$Q^2 = (2m_t)^2$	57. [pb]	56. [pb]	48. [pb]
	$q\bar{q} \rightarrow W \rightarrow \ell\nu$		
$Q^2 = \hat{s}$	17200 [pb]	17230 [pb]	17310 [pb]
$Q^2 = (\sum (p_T^{i\,2}) + M_W^2)/2$	16480 [pb]	16490 [pb]	16460 [pb]
$Q^2 = \sum (p_T^{i\,2})/2$	12920 [pb]	12920 [pb]	13020 [pb]
$Q^2 = M_W^2$	17360 [pb]	17380 [pb]	17300 [pb]
	$gg \rightarrow WbWb$		
$Q^2 = \hat{s}$	370 [pb]	365 [pb]	310 [pb]
$Q^2 = \sum (p_T^{i\,2} + m_i^2)/2$	430 [pb]	425 [pb]	355 [pb]
$Q^2 = \sum (p_T^{i\,2})/2$	525 [pb]	520 [pb]	435 [pb]
$Q^2 = (2m_t)^2$	330 [pb]	320 [pb]	275 [pb]

5 Monte Carlo algorithm

The conceptual motivation leading to the present implementation of **AcerMC** was to exploit the possibility of dedicated matrix-element-based generation interfaced to a more general event generator, called *supervising* event generator, which is subsequently used to complete the event generation procedure.

The goal of the dedicated matrix-element-based part is to efficiently generate complicated event topologies using native (multi-channel based) phase space generation procedures. The strategy is based on the understanding that a case-by-case optimisation is in complex cases of phase space topologies preferable to an universal algorithm. Given that phase-space is optimised on a case-by-case basis, an user-defined pre-selection for the generated regions of the phase-space is not implemented. Due to the fact that the $2 \rightarrow 4$ and $2 \rightarrow 6$ matrix elements, provided by the MADGRAPH/HELAS [9] package, contain full massive treatment of the final state particles, there are no explicit divergences present for implemented processes and **AcerMC** can indeed cover the *full* (kinematically allowed) phase space of the processes at hand.

The matrix-element-based part uses $\alpha_{QCD}(Q^2)$ and $\alpha_{QED}(Q^2)$ couplings and mass spectra, as calculated by the supervising event generator, to insure the full internal consistency in treatment of the event itself. Optionally, the native $\alpha_{QCD}(Q^2)$ and $\alpha_{QED}(Q^2)$ definitions can also be invoked.

The generation chain is built from the following steps:

- The PYTHIA 6.2 or HERWIG 6.5 interfaces to the library of the structure functions LHAPDF [42] are used to calculate convolution of the partonic density.
- **AcerMC** modules produce unweighted hard-process events with colour flow information and pass them to the supervising generator PYTHIA 6.2 or HERWIG 6.5 as an external event.
- The generated events are then further treated within PYTHIA 6.2 or HERWIG 6.5 event generators, where the fragmentation and hadronisation procedures, as well as the initial and final state radiation are added and final unweighted events are produced.

The **AcerMC** efficiency[6] for generating unweighted events, using the implementation of the phase-space generation discussed below, is summarised in Table **11**. A certain (very small) fraction of events is further rejected in the showering/fragmentation procedures of the supervising generators.

In the following we will briefly describe the key points of the implemented **AcerMC** modules and developed algorithms: matrix element calculations, n-fermion phase-space generation based on the modified Kajantie-Byckling methods[55], the issue of the s-dependent width and mass threshold effects for resonances and finally, the modification of the VEGAS algorithm.

5.1 The Matrix Element Calculation

The squared matrix elements of the processes were obtained by using the MADGRAPH/HELAS [9] package. They take properly into account the masses and helicity contributions of final states particles, incoming quarks are considered as massless. The particle masses, charges and coupling values that are passed to the code derived with the MADGRAPH package are calculated from functions consistent with the ones used in supervising generators (PYTHIA/HERWIG). This allows to preserve the internal consistency of the event generation procedure. In particular, the (constant) coupling values of α_s and α_{QED} were replaced with the appropriate running functions that were either taken from the interfaced generators or provided by the **AcerMC** code according to user settings. Slightly modified MADGRAPH/HELAS allowed for obtaining colour flow information of the implemented processes.

The sets of the MADGRAPH/HELAS coded diagrams, for each of the implemented processes, are collected in Appendix A.

[6] Note that efficiency is energy scale dependent and phase-space optimisation is done individually for each choice. So it might vary for the same process but different choices of the energy scale definition.

Table 11: *Efficiency for the generation of unweighted events with the default definition of the energy scale,* ACES2=1 *(see Section 7.8 for details). For generation of the $q\bar{q}, gg \to Z/\gamma^*(\to \ell\ell)b\bar{b}$ and $q\bar{q}, gg \to Z/\gamma^*(\to \ell\ell)t\bar{t}$ events threshold $m_{\ell\ell} \geq 60$ GeV has been used. The $f = e, \mu, \tau, q, b$.*

Process	Description	Internal AcerMC efficiency
	$2 \to 4$	
[1]	$gg \to t\bar{t}b\bar{b}$	36.3 %
[2]	$q\bar{q} \to t\bar{t}bb$	29.7 %
[3]	$q\bar{q} \to W(\to \nu\ell)bb$	35.2 %
[4]	$q\bar{q} \to W(\to \nu\ell)tt$	30.0 %
[5]	$gg \to Z/\gamma^*(\to \ell\ell)bb$	42.8 %
[6]	$q\bar{q} \to Z/\gamma^*(\to \ell\ell)bb$	35.1 %
[7]	$gg \to Z/\gamma^*(\to ff, \nu\nu)t\bar{t}$	47.0 %
[8]	$q\bar{q} \to Z/\gamma^*(\to ff, \nu\nu)t\bar{t}$	42.6 %
[9]	$gg \to (Z/W/\gamma^* \to)t\bar{t}bb$	9.3 %
[10]	$q\bar{q} \to (Z/W/\gamma^* \to)t\bar{t}bb$	32.4 %
[15]	$gg \to tttt$	48.0 %
[16]	$q\bar{q} \to tttt$	50.2 %
	$2 \to 6$	
[11]	$gg \to (t\bar{t} \to)ffbffb$	14.2 %
[12]	$q\bar{q} \to (t\bar{t} \to)ffbffb$	12.0 %
[13]	$gg \to (WWbb \to)ffffbb$	18.2 %
[14]	$q\bar{q} \to (WWbb \to)ffbffbb$	4.2 %
	Control processes	
[91]	$q\bar{q} \to Z/\gamma^* \to \ell\ell$	68.4 %
[92]	$gg \to tt$	65.6 %
[93]	$q\bar{q} \to tt$	62.1 %
[94]	$q\bar{q} \to W \to \nu\ell$	69.4 %
[95]	$gg \to (t\bar{t} \to)WbWb$	40.2 %

5.2 The Phase Space Generation Procedure

The general objective in simulation of physics processes for the LHC environment is to improve the integration of the differential cross-section using Monte-Carlo sampling methods[7]. The sampling method used should aim to minimise the variance of the integral as well as maximise the sampling efficiency given a certain number of iterations and the construction of the sampling method itself should aim to be sufficiently general and/or modular to be applicable to a wide range of processes. Writing down a (process) cross-section integral for LHC type (hadron-hadron) collisions:

$$\sigma = \int \sum_{a,b} f_a(x_1, Q^2) f_b(x_2, Q^2) \frac{|\mathcal{M}_n|^2}{(2\pi)^{3n-4}(2\hat{s})} \, dx_1 \, dx_2 \, d\Phi_n, \tag{1}$$

where $\mathrm{f_{a,b}(x, Q^2)}$ represent the gluon or (anti)quark parton density functions, $|\mathcal{M}_\mathrm{n}|^2$ the squared n-particle matrix element divided by the flux factor $[(2\pi)^{3\mathrm{n}-4} 2\hat{\mathrm{s}}]$ and $d\Phi_n$ denotes the n-particle phase space differential. The quantity $\hat{\mathrm{s}} = \mathrm{x_1\, x_2\, s}$ is the effective centre-of-mass energy, and the sum $\sum_{\mathrm{a,b}}$ runs in case of quark-antiquark incident partons over all possible quark-antiquark combinations $(\mathrm{a, b} = \mathrm{u, d, s, c, \bar{u}, \bar{d}, \bar{s}, \bar{c}})$. In case of gg initial state the sum has only one term with $\mathrm{a = b = g}$.

[7]For a nice discussion on the topic see e.g. [50, 51]...

It is often convenient to re-write the differential cross-section in the form:

$$\sigma = \int \sum_{a,b} x_1 f_a(x_1, Q^2)\; x_2 f_b(x_2, Q^2) \frac{|\mathcal{M}_n|^2}{(2\pi)^{3n-4}(2\hat{s}^2)}\, dy\, d\hat{s}\, d\Phi_n, \tag{2}$$

with the new (rapidity) variable given by $\mathrm{y} = 0.5\log(\mathrm{x}_1/\mathrm{x}_2)$. The n-body phase-space differential $\mathrm{d}\Phi_\mathrm{n}$ and its integral Φ_n depend only on $\hat{s}$ and particle masses m_i due to Lorentz invariance:

$$\Phi_n(\hat{s}, m_1, m_2, \ldots, m_n) = \int d\Phi_n(\hat{s}, m_1, m_2, \ldots, m_n) = \int \delta^4\left((p_a + p_b) - \sum_{i=1}^{n} p_i\right) \prod_{i=1}^{n} d^4 p_i \delta(p_i^2 - m_i^2)\Theta(p_i^0), \tag{3}$$

with a and b denoting the incident particles and i running over all outgoing particles $i = 1, \ldots, n$. What one would like to do is to split the n-body phase parameterised by 3n-4 essential (i.e. non-trivial) independent variables into manageable subsets (modules) to be handled by techniques which reduce the variance of the result and/or the sampling efficiency (e.g. importance sampling[41] or adaptive integration like VEGAS[8] or FOAM[45]). Stating this in formal terms, the above Equation 2 should be transformed into an expression like:

$$\sigma = \left(\prod_{i=1}^{n} \int_{s_i^-}^{s_i^+} ds_i\right) \left(\prod_{j=1}^{m} \int_{t_j^-}^{t_j^+} dt_j\right) \left(\prod_{k=1}^{l} \int_{\Omega_k^-}^{\Omega_k^+} d\Omega_k\right) |\mathcal{J}_n| \int \sum_{a,b} x_1 f_a(x_1, Q^2)\; x_2 f_b(x_2, Q^2) \frac{|\mathcal{M}_n|^2}{(2\pi)^{3n-4}(2\hat{s}^2)}\, dy\, d\hat{s} \tag{4}$$

where one integrates over Mandelstam type (Lorentz invariant) momenta transfers $\mathrm{s_i}, \mathrm{t_j}$ and space angles $\Omega_k \equiv (\cos\vartheta_k, \phi_k)$ within the kinematically allowed limits (3n-4 variables in total) with the term $|\mathcal{J}_\mathrm{n}|$ denoting the Jacobian of the transformation. If one would then decide to introduce importance sampling functions in order to reduce the peaking behavior of the integrand [41], the integrals would take the form:

$$\int_{s_i^-}^{s_i^+} ds_i = \int_{s_i^-}^{s_i^+} \frac{g_i(s_i)}{g_i(s_i)} ds_i, \tag{5}$$

where the importance sampling function $\mathrm{g_i}$ is probability density function normalised in the integration region $[\mathrm{s_i^-}, \mathrm{s_i^+}]$:

$$\int_{s_i^-}^{s_i^+} g_i(s_i) ds_i = 1, \tag{6}$$

which exhibits a similar peaking behavior as the integrand. Formally, one then inserts the identity:

$$1 = \int_0^1 \delta\left(r_i - \int_{s_i^-}^{s_i} g_i(s_i) ds_i\right) dr_i \tag{7}$$

into the integral and then derives the *unitary* sampling prescription:

$$\int_0^1 dr_i \int_{s_i^-}^{s_i^+} \delta\left(r_i - \int_{s_i^-}^{s_i} g_i(s_i) ds_i\right) \frac{g_i(s_i)}{g_i(s_i)} ds_i = \int_0^1 dr_i \int_{s_i^-}^{s_i^+} \delta\left(s_i - G^{-1}(r_i)\right) \frac{1}{g_i(s_i)} ds_i = \int_0^1 \frac{dr_i}{g_i(G^{-1}(r_i))}, \tag{8}$$

which formally means that the $\mathrm{s_i}$ values are sampled from the interval according to the $\mathrm{g_i(s_i)}$ distribution by using (pseudo-)random variable $\mathrm{r_i}$ together with the $\mathrm{g_i(s_i)}$ cumulant $G(s_i) = \int_{s_i^-}^{s_i} g_i(s_i) ds_i$ with its inverse G^{-1}. The unitarity of the algorithm states that each trial ($\mathrm{r_i}$ value) produces a result (i.e. a corresponding $\mathrm{s_i}$ value distributed according to $\mathrm{g_i(s_i)}$).

Performing such substitutions on all integration parameters would give as the cross-section expression;

$$\sigma = \prod_{i=1}^{3n-4} \int_0^1 dr_i \frac{f(r_1, r_2 \ldots)}{g(r_1, r_2 \ldots)} \tag{9}$$

where the integrand would (hopefully) have as low variation as possible at least for a subset of contributing Feynman diagrams[8]. To improve the sampling method further, the $\mathrm{r_i}$ (pseudo-)random variables can be sampled from adaptive algorithms of the VEGAStype [8].

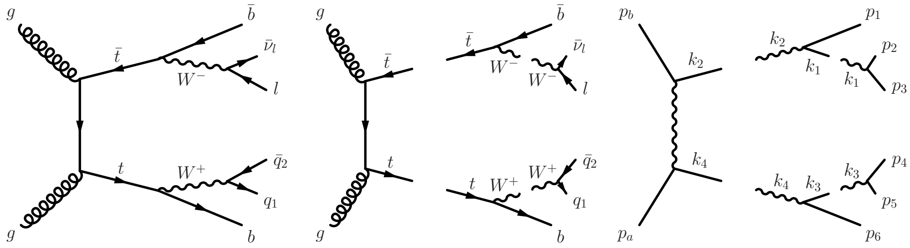


Fig. 3: A representative Feynman diagram describing a $2 \rightarrow 6$ process $gg \rightarrow t\bar{t} \rightarrow b\bar{b}W^+W^- \rightarrow b\bar{b}\ell\bar{\nu}_\ell q_1\bar{q}_2$ and its decomposition into a set of $2 \rightarrow 2$ t-channel and s-channel sub-processes.

A representative Feynman diagram describing a $2 \rightarrow 6$ process is shown in Figure **3**. As one can see, the process can be split in several consecutive branchings, this approximation is often used in matrix element (probability amplitude) calculations. It seems rather obvious that any Feynman diagram can be split in a series of horizontal and vertical branchings that one can denote as s-type and t-type(u-type) using the analogy with the Mandelstam variables. What one would like to do is thus to modularise the phase space in the form of sequential s- and t-type splits.

The s-splitting of phase space is relatively easy to do and has as such been used in many instances of Monte–Carlo generation (e.g. FermiSV [38], Excalibur [39], Tauola [6] etc..); the t-type branchings (often tagged as multi-(peri)pheral topologies) have in contrast generally been calculated only for specific cases (e.g. for 3 or 4 particles in the final state [38, 39]). As it turns out, the problem of several massive particles in the final state has already appeared more than 30 years ago when several hadrons (e.g. pions) have been produced in (comparatively low energy) nuclear interactions. At that time Kajantie and Byckling [55] have derived the formulae for simulating any sequence of s- and t- type branchings which, with some modifications, can also be applied to the EW and QCD processes involving heavy quarks and/or massive bosons at LHC.

[8] The 'modularisation' can be performed for several topologies at the same time and multi-channel techniques can be applied.

5.3 Modified Kajantie-Byckling Formalism

5.3.1 The s-type Branching Algorithms

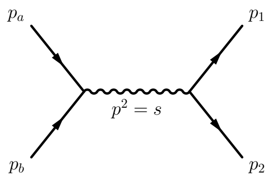


Fig. 4: A diagram of a generic $2 \rightarrow 2$ s-channel process.

The s-splits are the simplest method in the KB formalism. For the sake of completeness one should start with the definition of the two-body phase space integral (c.f. Fig **4**):

$$\Phi_2(s, m_1, m_2) = \int d^4p_1 d^4p_2 \delta(p_1^2 - m_1^2)\delta(p_2^2 - m_2^2)\delta^4(p - p_1 - p_2)\Theta(p_1^0)\Theta(p_2^0), \tag{10}$$

with the incoming momentum sum $\mathrm{p} = (\mathrm{p_a} + \mathrm{p_b}), \mathrm{p}^2 = \mathrm{s}$ and outgoing momenta $\mathrm{p}_{1,2}, \mathrm{p}_{1,2}^2 = \mathrm{m}_{1,2}^2$. The phase space integral is Lorentz invariant (as one can observe in the above Equation where it is written in a manifestly Lorentz invariant form). Subsequently, due to Lorentz invariance, the integral is necessarily a function of the Lorentz scalars s, m_1 and m_2 only. The step function product $\Theta(\mathrm{p}_1^0)\Theta(\mathrm{p}_2^0)$ is the explicit requirement of the positiveness of the energy terms in $\mathrm{p}_{1,2}$ while the delta functions represent the on-shell conditions on $\mathrm{p}_{1,2}$ and the total momentum conservation.

The integral can be transformed into a more compact form by integrating out the spurious variables; one thus first integrates over $\mathrm{d}^4\mathrm{p}_2$ and chooses the centre-of-mass system (CMS) as the integration system of reference with $\mathrm{p} = (\sqrt{\mathrm{s}}, 0, 0, 0)$ and then evaluates the integrals over p_1^0 and E_1^*:

$$\begin{aligned}
\Phi_2(s, m_1, m_2) &= \int d^4p_1 \delta(p_1^2 - m_1^2)\delta((p - p_1)^2 - m_2^2)\Theta(p_1^0) \\
&= \int \frac{d^3p_1^*}{2E_1^*}\delta(s + m_1^2 - 2\sqrt{s}E_1^* - m_2^2) \\
&= \frac{1}{4\sqrt{s}} \int p_*^1 dE_1^* d\Omega_1^* \delta\left(E_1^* - \frac{s + m_1^2 - m_2^2}{2\sqrt{s}}\right) \\
&= \frac{p_1^*(s, m_1, m_2)}{4\sqrt{s}} \int d\Omega_1^*,
\end{aligned} \tag{11}$$

with the stars explicitly denoting the values in the centre-of mass system. The first integration simply sets $\mathrm{p}_1^0 = \sqrt{(\mathrm{p}_1^*)^2 + \mathrm{m}_1^2} = \mathrm{E}_1^*$ and the second integral leads to the well known relations for the energy:

$$E_1^* = \frac{s + m_1^2 - m_2^2}{2\sqrt{s}}, \quad E_2^* = \sqrt{s} - E_1^* = \frac{s + m_2^2 - m_1^2}{2\sqrt{s}}, \tag{12}$$

and momenta sizes:

$$p_1^* = |\vec{p_1^*}| = \frac{\sqrt{\lambda(s, m_1^2, m_2^2)}}{2\sqrt{s}}, \quad p_2^* = p_1^* \tag{13}$$

of two particle production. The $\lambda(\mathrm{s}, \mathrm{m}_1^2, \mathrm{m}_2^2)$ denotes the Lorentz invariant function:

$$\lambda(s, m_1^2, m_2^2) = (s - (m_1 + m_2)^2)(s - (m_1 - m_2)^2) \tag{14}$$

and thus explicitly contains the phase space cutoff, i.e. the requirement that the available CMS energy $\sqrt{\mathrm{s}}$ should be bigger than the mass sum $\sqrt{\mathrm{s}} \geq (\mathrm{m}_1 + \mathrm{m}_2)$. Note that the integration was so far done only over the spurious parameters, leaving the polar and azimuthal angle of the p_1 particle as the two independent parameters $\mathrm{d}\Omega^* = \mathrm{d}\cos\theta^* \mathrm{d}\varphi^*$. The integral becomes trivial to sample in case the outgoing particles can be approximated as massless (the 'boost' factor lambda transforms to unity). As already claimed, the latter approximation is however often unjustified when studying processes representative for the LHC environment.

Kajantie and Byckling [55] introduced the *recursion* and *splitting* relations for the n-particle phase space $\Phi_\mathrm{n}(\mathrm{s})$ given by Eq. 3. The recursion relation can be derived by defining the momentum sum:

$$k_i = \sum_{j=1}^{i} p_j = (k_i^0, \vec{k_i}); \quad M_i^2 = k_i^2. \tag{15}$$

Subsequently one can interpret $\mathrm{p} = \mathrm{k_n}$ and $s = M_n^2$ from Eq. 3. One continues by introducing the identities:

$$1 = \int dM_{n-1}^2 \delta(k_{n-1}^2 - M_{n-1}^2)\Theta(k_{n-1}^0) \tag{16}$$

and

$$1 = \int d^4k_{n-1}\delta^4(p - k_{n-1} - p_n) \tag{17}$$

into the integral of Equation 3; separating out the arguments containing $\mathrm{k_{n-1}}$ and p_n terms one obtains:

$$\begin{aligned} \Phi_n(M_n^2, m_1, m_2, \ldots, m_n) &= \int dM_{n-1}^2 \times \\ &\times \left\{ \int d^4k_{n-1} d^4p_n \delta(k_{n-1}^2 - M_{n-1}^2)\delta(p_n^2 - m_n^2)\delta^4(p - k_{n-1} - p_n)\Theta(k_{n-1}^0)\Theta(p_n^0) \right\} \times \\ &\times \Phi_{n-1}(M_{n-1}^2, m_1, m_2, \ldots, m_{n-1}), \end{aligned} \tag{18}$$

where the remaining p_i terms form the (n-1)-particle phase space integral $\Phi_{\mathrm{n-1}}(\mathrm{M_{n-1}^2}, \mathrm{m_1}, \mathrm{m_2}, \ldots, \mathrm{m_{n-1}})$ and the terms in curly brackets give a two particle phase space term (c.f. Eq. 12):

$$\begin{aligned} \Phi_n(M_n^2, m_1, m_2, \ldots, m_n) &= \int dM_{n-1}^2 \Phi_2(M_n^2, M_{n-1}, m_n)\Phi_{n-1}(M_{n-1}^2, m_1, m_2, \ldots, m_{n-1}) \\ &= \int dM_{n-1}^2 \frac{p_n^*}{4M_n} \Phi_{n-1}(M_{n-1}^2, m_1, m_2, \ldots, m_{n-1}) \\ &= \int_{(\sum_{i=1}^{n-1} m_i)^2}^{(M_n - m_n)^2} dM_{n-1}^2 \frac{\sqrt{\lambda(M_n^2, M_{n-1}^2, m_n^2)}}{8M_n^2} \int d\Omega_n^* \Phi_{n-1}(M_{n-1}^2, m_1, m_2, \ldots, m_{n-1}), \end{aligned} \tag{19}$$

with the integration limits on $\mathrm{M_{n-1}^2}$ following from its definition in Eq. 15. It has to be emphasized that the angles in $d\Omega_i^*$ are each time calculated in the centre-of-mass system of $\mathrm{k_i}$ with the invariant mass $\mathrm{M_i}$. The resulting recursion relation is clearly of advantage when describing cascade decays of particles $k_n \to k_{n-1}p_n \to k_{n-2}, p_n, p_{n-1} \to \ldots$; it also proves that the n-particle phase space of Eq. 3 can be reduced into a sequence of two-particle phase space terms, as shown in Figure **5**.

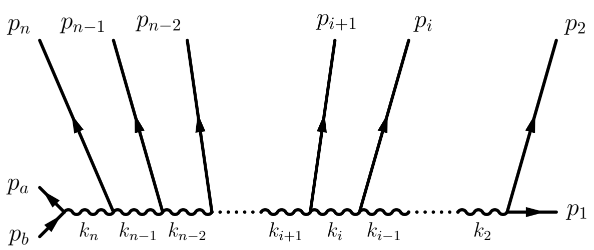


Fig. 5: The diagrammatic representation of consecutive s-splits.

It can further prove of advantage to loosen up the splitting terms of Eqns. 16,17 so that instead of summing to n-1 one groups an arbitrary set of ℓ particles:

$$1 = \int dM_l^2 \delta(k_l^2 - M_l^2)\Theta(k_l^0), \tag{20}$$

$$1 = \int d^4 k_l \delta^4(p - k_l - \sum_{j=l+1}^{n} p_j), \tag{21}$$

which, when repeating the procedure in recursion relation of Eq. 19, results in an expression:

$$\Phi_n(M_n^2, m_1, m_2, \ldots, m_n) = \int\limits_{(\sum_{i=1}^{l} m_i)^2}^{(M_{l+1}-m_{l+1})^2} dM_l^2 \Phi_{n-l+1}(M_n^2, M_l, m_{l+1}, \ldots, m_n)\Phi_l(M_l^2, m_1, m_2, \ldots, m_l), \tag{22}$$

and thus effectively splits the phase space into two subsets, equivalent to introducing an intermediate(virtual) particle with momentum k_l.

The number of splitting relations and the number of particles in each group as given in Eq. 21 can be chosen in any possible sequence, thus meaning that the grouping sequence is arbitrary and can be adjusted to fit the topology in question.[9]

At this point some modifications were introduced to the algorithm in order to adapt it to the specifics of the processes expected at the LHC. Kajantie and Byckling namely assumed that the generation sequence would be 'down' the cascade (i.e. by sampling first a $\mathrm{M_n}$ value, then M_{n-1} value etc... as is indeed most often done in Monte-Carlo Generators). This might however not be optimal in the LHC environment since the available centre-of-mass energy for the hard process ($\hat{s}$) can vary in a wide range of values (c.f. Equation 2) and has to be sampled from a distribution itself. The shape of the distribution function for $\hat{s}$ is expected to behave as a convolution of the peaking behavior of all participating invariant masses times the parton density functions (c.f. Eq. 2); it subsequently seems to be more natural (and efficient) first to sample the individual propagator peaks and then their subsequent convolutions. Furthermore, by generating the invariant masses 'up' the cascade (i.e. first $\mathrm{M_2}$, $\mathrm{M_3} \ldots \mathrm{M_n}$ and finally $\hat{\mathrm{s}}$) the kinematic limits on the branchings occur in a more efficient way (bound on the $\sqrt{\lambda}$ values, see Equations 14 and 24), which is very convenient since in the LHC environment no stringent generation cuts should be made on the inherently non-measurable $\hat{\mathrm{s}}$ as it cannot be accounted for by an analogous cut in a physics analysis.

A necessary modification of the algorithm would thus be to reverse the generation steps by starting with the last pair(s) of particles. In terms of integration (i.e. sampling) limits this translates into changing the limits of Eq. 19:

$$\begin{aligned}
\Phi_n(M_n^2, m_1, m_2, \ldots, m_n) &= \qquad\qquad (23)\\
&= \int\limits_{(\sum_{i=1}^{n-1} m_i)^2}^{(M_n-m_n)^2} dM_{n-1}^2 \frac{\sqrt{\lambda(M_n^2, M_{n-1}^2, m_n^2)}}{8M_n^2} \int d\Omega_n^* \\
&\times \int\limits_{(\sum_{i=1}^{n-2} m_i)^2}^{(M_{n-1}-m_{n-1})^2} dM_{2-1}^2 \frac{\sqrt{\lambda(M_{n-1}^2, M_{n-2}^2, m_{n-1}^2)}}{8M_{n-1}^2} \int d\Omega_{n-1}^* \\
&\times \ \ldots \int\limits_{(\sum_{j=1}^{i-1} m_j)^2}^{(M_i-m_i)^2} dM_i^2 \frac{\sqrt{\lambda(M_i^2, M_{i-1}^2, m_i^2)}}{8M_i^2} \int d\Omega_i^* \ldots \\
&\times \int\limits_{(m_1+m_2)^2}^{(M_3-m_3)^2} dM_2^2 \frac{\sqrt{\lambda(M_2^2, m_1^2, m_2^2)}}{8M_2^2} \int d\Omega_2^*,
\end{aligned}$$

which accommodates the mass generation sequence: $k_n \to k_{n-1} + p_n \to \ldots$ (i.e. first sample M_{n-1}^2, then M_{n-2} etc...),

[9]Suggestions of [55] on how to pick random number sequences will not be used since one might like to couple this method with an adaptive algorithm to improve the sampling efficiencies.

into

$$
\begin{aligned}
\Phi_n(M_n^2, m_1, m_2, \ldots, m_n) &= \int\limits_{(M_{n-2}+m_{n-1})^2}^{(M_n-m_n)^2} dM_{n-1}^2 \frac{\sqrt{\lambda(M_n^2, M_{n-1}^2, m_n^2)}}{8M_n^2} \int d\Omega_n^* \\
&\times \int\limits_{(M_{n-3}+m_{n-2})^2}^{(M_n-m_n-m_{n-1})^2} dM_{n-2}^2 \frac{\sqrt{\lambda(M_{n-1}^2, M_{n-2}^2, m_{n-1}^2)}}{8M_{n-1}^2} \int d\Omega_{n-1}^* \\
&\times \ \ldots \int\limits_{(M_{i-1}+m_i)^2}^{(M_n-\sum_{j=i+1}^n m_j)^2} dM_{i-1}^2 \frac{\sqrt{\lambda(M_i^2, M_{i-1}^2, m_i^2)}}{8M_i^2} \int d\Omega_i^* \ldots \\
&\times \int\limits_{(m_1+m_2)^2}^{(M_n-\sum_{j=3}^n m_j)^2} dM_2^2 \frac{\sqrt{\lambda(M_2^2, m_1^2, m_2^2)}}{8M_2^2} \int d\Omega_2^*,
\end{aligned}
\tag{24}
$$

where one first samples the mass $\mathrm{M}_2, \mathrm{M}_3 \ldots M_{n-1}$ in the appropriate limits.

In some topologies symmetric cases of mass generation can appear (as shown in Figure **3**) where the integration sequence is ambivalent (e.g. in Figure **3** the ambivalence is which top quark invariant mass to generate first…) and after a choice is made (since one of the two cases in the symmetric pair has to take precedence) the procedure itself remains not entirely symmetric. Detailed studies have shown that it proves useful to include all permutations of such ambiguous sequences into the MC algorithm in order to 'symmetrise' the solution and thus make it easier to process by further additions (e.g. adaptive algorithms).

5.3.2 The t-type Branching Algorithms

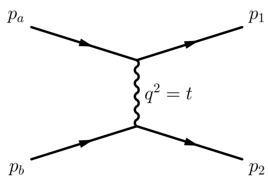


Fig. 6: A diagram of a generic $2 \to 2$ t-channel process.

The t-splits are a specialty of the KB formalism due to the advanced calculation of the limits on the (massive) t-variable. The formalism can be introduced by observing that in case of a $\mathrm{p_a} + \mathrm{p_b} \to \mathrm{p_1} + \mathrm{p_2}$ scattering the momentum transfer is characterised by the (Mandelstam) variable $\mathrm{t} = (\mathrm{p_1} - \mathrm{p_a})^2$ (c.f. Fig **6**). It is thus sensible to replace the $d\Omega_1^* = d\cos\theta_1^* d\varphi_1^*$ integration in the two body phase space integral of Eq. 12 with integration over the t variable. Writing the definition of t in the centre-of-mass system one gets:

$$
\begin{aligned}
t &= q^2 = (p_a - p_1)^2 \\
&= m_a^2 + m_1^2 - 2E_a^* E_1^* + 2p_a^* p_1^* \cos\theta_1^*
\end{aligned}
\tag{25}
$$

and hence:

$$
dt = 2p_a^* p_1^* d\cos\theta^* \tag{26}
$$

Using the latter substitution together with Eq. 12,13 and the analogue for $\mathrm{p_a}$:

$$
p_a^* = \frac{\sqrt{\lambda(s, m_a^2, m_b^2)}}{2\sqrt{s}} \tag{27}
$$

one obtains in place of Eq. 12

$$\begin{aligned}\Phi_2(s,m_1,m_2) &= \frac{p_1^*(s,m_1,m_2)}{4\sqrt{s}}\int d\Omega_1^* \\ &= \frac{1}{8\sqrt{s}p_a^*}\int dt\, d\varphi^* \\ &= \frac{1}{4\sqrt{\lambda(s,m_a^2,m_b^2)}}\int_{t^-}^{t^+} dt \int_0^{2\pi} d\varphi^*\end{aligned} \tag{28}$$

With the integration variable change the integration domain changes from $[-1,1]$ for $\mathrm{d}\cos\theta^*$ to $[\mathrm{t}^-,\mathrm{t}^+]$ for the dt integration. The $\mathrm{t}^\pm$ limits are obtained by inserting the $\cos\theta^*$ limits into Equation 26:

$$t^\pm = m_a^2 + m_1^2 - 2E_a^*E_1^* \pm 2p_a^*p_1^* \tag{29}$$

or in the Lorentz invariant form (c.f. Eq. 12,13):

$$\begin{aligned} t^\pm &= m_a^2 + m_1^2 - \frac{(s+m_a^2-m_b^2)(s+m_1^2-m_2^2)}{2s} \\ &\pm \frac{\sqrt{\lambda(s,m_a^2,m_b^2)\lambda(s,m_1^2,m_2^2)}}{2s}\end{aligned} \tag{30}$$

As a step towards generalisation one has to note that the kinematic limits $t^\pm$ can also be derived from *the basic four-particle kinematic function G(x,y,z,u,v,w)*[56, 55], where the function G can be expressed as a Cayley determinant:

$$G(x,y,z,u,v,w) = -\frac{1}{2}\begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & v & x & z \\ 1 & v & 0 & u & y \\ 1 & x & u & 0 & w \\ 1 & z & y & w & 0 \end{vmatrix} \tag{31}$$

The kinematic limits on t are in this case given by the condition

$$G(s,t,m_2^2,m_a^2,m_b^2,m_1^2) \leq 0, \tag{32}$$

it should be noted that the above condition gives either $\mathrm{t}^\pm$ limits given a fixed value of s or equivalently $\mathrm{s}^\pm$ limits given a fixed t value.

In search of a recursion relation involving t-variables one can note that in Eq. 19 the angle in $\cos\theta_n^*$ is equivalent to the scattering angle in the centre-of-mass system of the reaction $\mathrm{p_a} + \mathrm{p_b} \to \mathrm{k_{n-1}} + \mathrm{p_n}$ and thus given by:

$$\begin{aligned} t_{n-1} &= (p_a - k_{n-1})^2 \\ &= m_a^2 + M_{n-1}^2 - 2E_a^*k_{n-1}^{0*} + 2p_a^*k_{n-1}^*\cos\theta_{n-1}^*\end{aligned} \tag{33}$$

with the $\mathrm{t}_{\mathrm{n}-1}^\pm$ limits expressed by:

$$G(M_n^2, t_{n-1}, m_n^2, m_a^2, m_b^2, M_{n-1}^2) \leq 0, \tag{34}$$

and the p_a^* given by Eq. 27. In order to produce a more general picture it can further be deduced that the next angle in the recursion θ_{n-1}^*, is the scattering angle of the subsequent process $\mathrm{p_a} + (\mathrm{p_b} - \mathrm{p_n}) \to \mathrm{k_{n-2}} + \mathrm{p_{n-1}}$ in the centre-of-mass system of k_{n-1}; the $(\mathrm{p_b} - \mathrm{p_n}) = \mathrm{q_{n-1}}$ is in this case considered as a virtual incoming particle with momentum $\mathrm{q_{n-1}}$ (c.f. Figure **7**).
It immediately follows that for a general process $p_a + q_{i+1} \to k_i + p_{i+1}$ with:

$$q_i = p_b - \sum_{j=i+1}^{n} p_j = p_a - k_i; \quad q_i^2 = t_i; \quad q_n^2 = t_n = m_b^2 \tag{35}$$

a general expression for $\mathrm{t_i}$ becomes in the centre-of-mass frame of $\mathrm{k_{i+1}}$:

$$\begin{aligned} t_i &= (p_a - k_i)^2 \\ &= m_a^2 + M_i^2 - 2E_a^{*(i+1)}k_i^{0*(i+1)} + 2p_a^{*(i+1)}k_i^{*(i+1)}\cos\theta_i^*\end{aligned} \tag{36}$$

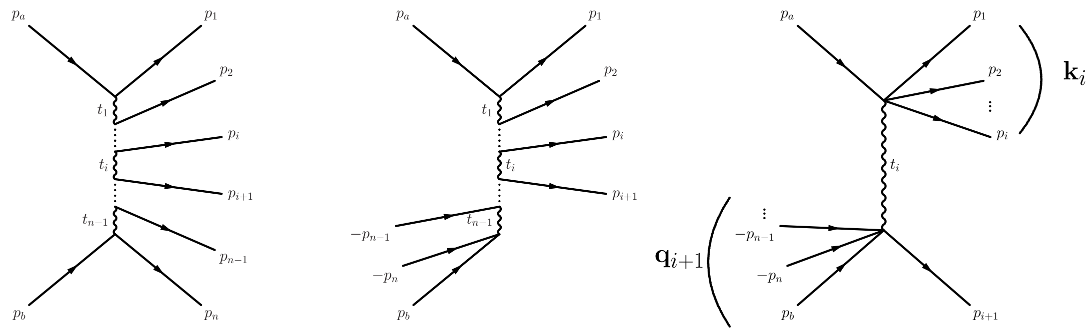


Fig. 7: The diagrammatic representation of the method applied in translating the multi-(peri)pheral splits into a $2 \to 2$ t-channel configuration.

where momenta in centre-of-mass frame of $\mathrm{k_{i+1}}$, denoted with the superscript $*(\mathrm{i}+1)$, are given by:

$$k_i^{*(i+1)} = \frac{\sqrt{\lambda(M_{i+1}^2, M_i^2, m_{i+1}^2)}}{2M_{i+1}} \tag{37}$$

$$p_a^{*(i+1)} = \frac{\sqrt{\lambda(M_{i+1}^2, m_a^2, t_{i+1})}}{2M_{i+1}} \tag{38}$$

and the corresponding energies $k_i^{0*(i+1)}$ and $\mathrm{E_a^{*(i+1)}}$ can simply be obtained by using the analogues of Equations 12,13 or the usual Einstein mass-energy relations directly. The corresponding $\mathrm{t_i^{\pm}}$ limits given by:

$$G(M_{i+1}^2, t_i, m_{i+1}^2, m_a^2, t_{i+1}, M_i^2) \leq 0, \tag{39}$$

and the recursion relation of Eq. 19 becomes:

$$\Phi_n(M_n^2, m_1, m_2, \ldots, m_n) = \tag{40}$$

$$= \int\limits_{(\sum_{i=1}^{n-1} m_i)^2}^{(M_n - m_n)^2} \frac{dM_{n-1}^2}{4\sqrt{\lambda(M_n^2, m_a^2, t_n)}} \int\limits_0^{2\pi} d\varphi_n^* \int\limits_{t_{n-1}^-}^{t_{n-1}^+} dt_{n-1}\, \Phi_{n-1}(M_{n-1}^2, m_1, m_2, \ldots, m_{n-1}),$$

As already argued the resulting set of $(\mathrm{s_i} = \mathrm{M_i^2}, \mathrm{t_i})$ can again be sampled in any direction with respect to the cascade by applying the appropriate change in the integration limits (c.f. Eq. 19 and 24). The recommended approach (i.e. the introduced modification of the algorithm) is again to first sample the invariant masses in the reverse cascade direction (i.e. in the sequence $\mathrm{M_2, M_3, \ldots, M_n}$) and then the t_i values within the limits calculated from Eq. 34 down the cascade (i.e. in the order of $\mathrm{t_{n-1}, t_{n-2}, \ldots, t_1}$).

To sum up, it has been shown that using the Kajantie–Byckling formalism the phase space for any topology can be split in a set of s-type and t-type $2 \to 2$ branching steps (modules) given by recursive formulae of Equations 24 and 40.

5.4 Propagator Sampling

A well known theoretical issue is that one can expect the most prominent peaks in the differential cross-section of a specific process in the phase space regions of high propagator values in the corresponding probability density. Consequently, in the scope of complementing the modular structure of the derived Kajantie-Byckling based phase space sampling, new approaches were also developed concerning the numerical sampling methods of the relevant kinematic quantities.

In order to get small variance in the Monte Carlo procedure one would thus like to include the appropriate peaking dependence of the relevant momentum transfers q^2 in the importance sampling function. It however turns out that since the momenta transfers q participate also in the propagator numerators (typically in $\mathrm{p}_\mu \mathrm{q}^\mu/\mathrm{q}^2$) and since in process of interest one mostly finds several Feynman diagrams contributing to the final probability density, thus causing interferences, it is very difficult or even impossible to estimate the exact power of momenta transfers in the sampling functions for different propagator peaks. In other words, the probability density dependence on the momentum transfer q^2 can in general be approximated with the dependence $1/(\mathrm{q}^2)^\nu$ where the best value of ν must be determined separately (on a process by process basis).

In view of the latter, general formulae have been developed for sampling the $x^{-\nu}$ shape [38, 57]: Given a pseudo random number $r \in [0,1]$ and limits $x \in [x_-, x_+]$ the value x distributed as $x^{-\nu}$ is obtained from the formulae in Eq. 8 as:

$$x = \left[x_-^{-\nu+1} \cdot (1-r) + x_-^{-\nu+1} \cdot r\right]^{-\frac{1}{\nu+1}}; \quad \nu \neq 1; \tag{41}$$

$$x = \frac{x_+^r}{x_-^{r-1}}; \quad \nu = 1. \tag{42}$$

Using the analogous (unitary) approach a recipe for resonant (Breit-Wigner) propagator contributions of the type:

$$BW(s) = \frac{1}{(s-M^2)^2 + M^2\Gamma^2} \tag{43}$$

with $\mathrm{s} \in [\mathrm{s}_-, \mathrm{s}_+]$ and a pseudo random number $r \in [0,1]$ is available by the prescription:

$$s = M^2 + M\Gamma \cdot \tan\left[(u_+ - u_-) \cdot r + u_-\right] \tag{44}$$

$$u^\pm = \mathrm{atan}\left(\frac{s^\pm - M^2}{M\Gamma}\right) \tag{45}$$

Following similar arguments as for the non-resonant propagators one can surmise that the best sampling function for resonant propagators could in general be a Breit-Wigner shape modified by a factor $\mathrm{s}^\nu, \nu \in [0,1]$. As shown in the following section it was found that a shape:

$$BW(s) = \frac{s}{(s-M^2)^2 + M^2\Gamma^2} \tag{46}$$

works quite well for a set of processes and a corresponding sampling recipe was developed. In addition, studies in [58] show that a resonant $\sqrt{\mathrm{s}}\times$ Breit-Wigner shape:

$$BW(s) = \frac{\sqrt{s}}{(s-M^2)^2 + M^2\Gamma^2} \tag{47}$$

should be expected in a range of decay processes. Detailed studies have shown that it is in general better to introduce a $\mathrm{s}^\nu, \nu \in [0,1]$ dependence even if it over-compensates the high mass tails of the corresponding differential cross-section distribution since this provides an overall reduction of the maximal weight fluctuations in the Monte–Carlo event generation procedure.

5.4.1 Breit-Wigner Function with s-dependent Width

In some topologies of the processes involving $W^\pm$ or Z^0 bosons, a bias of the matrix element towards large values in the high $s^*_{W/Z}$ region is evident, which in turn means that a more accurate description of the tails of $s^*_{W/Z}$ distribution is needed. Consequently, the Breit-Wigner sampling function was replaced by[10]:

$$BW_s(s^*_W) = \frac{s^*_W}{(s^*_W - M^2_W)^2 + M^2_W \Gamma^2_W}, \tag{48}$$

[10] To our knowledge this implementation is original and done for the first time in **AcerMC**.

which is proportional to the (more accurate) Breit-Wigner function with an s_W^* dependent width (W in the above formula denotes either a $W^\pm$ or a Z^0 boson).

In order to implement a unitary algorithm (an algorithm that produces a result for every trial, i.e. there is no rejection) of value sampling on the above function one first has to calculate the normalisation integral (cumulant) and then its inverse function. Introducing a new variable $\eta = (s_W^* - M_W^2)/(M_W\ \Gamma_W)$ the integral of the above function can be expressed as:

$$\int BW_s(s_W^*)\ ds_W^* = \int \left\{ \frac{M_W^2}{M_W \Gamma_W} \cdot \frac{1}{1+\eta^2} + \frac{\eta}{1+\eta^2} \right\} d\eta, \tag{49}$$

where the upper integral limit is left as a free parameter. The integral thus gives a function:

$$F(\eta) = \left\{ \frac{M_W^2}{M_W \Gamma_W} \cdot \mathrm{atan}(\eta) \right\} + \left\{ \frac{1}{2} \cdot \log(\eta^2 + 1) \right\}, = F_1(\eta) + F_2(\eta) \tag{50}$$

with $F(\eta_{\max}) - F(\eta_{\min})$ defining the normalisation. One of the undesirable features is that the function $F(\eta)$ does not have a (simple) analytical inverse, which is a prerequisite for unitary sampling. Taking a closer look at the two above expressions one can quickly spot another undesirable feature, namely that the second term in the Equation 49 is an odd function of η, which after the integration gives an even term $F_2(\eta)$ in η in Equation 50. In other words the second term alone is neither a non-negative function nor does it have an unique inverse - one has to deal with a *negative probability*. A reasonably elegant solution to this problem has been developed and implemented here:

- One samples values of η by using only the first term of the above expressions (the *usual* Breit-Wigner function).
- One then re-samples the obtained value of η using the full expression of Equation 49: If η is less than zero the value is mapped to $-\eta$ with the probability given by Equation 49.

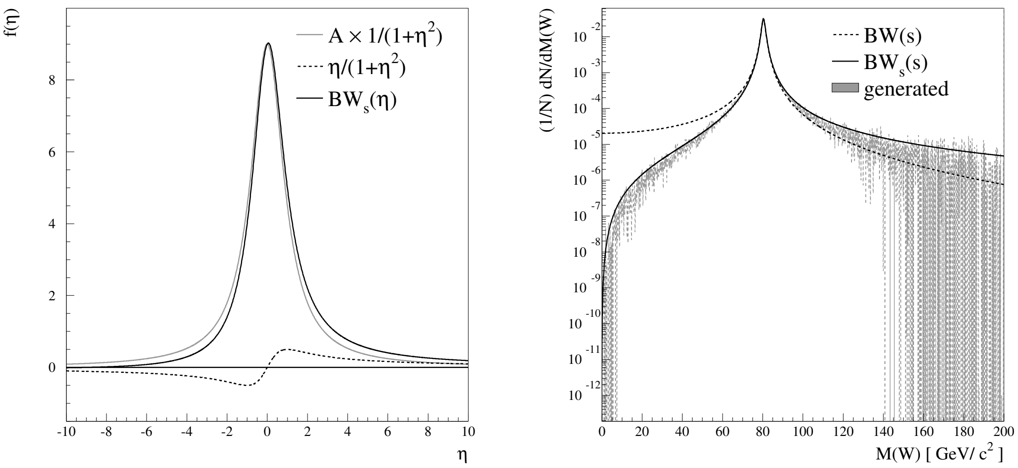


Fig. 8: **Left** *Comparisons of the two functional terms of Eq. 49 to* $\mathrm{BW}_s(\eta)$ *given by Equation 48. Note that the scaling factor* A *is chosen in view of making the contributions more transparent; it is much too small compared to the real case of* $W^\pm/Z^0$ *bosons.*
Right *Comparison of the (normalised) distributions of differential cross-section for the process* $q\bar{q} \to Wb\bar{b}$ *(dashed) and sampling functions (solid line) with respect to the variables obtained by importance sampling, as described in the text.*

Why this works can quickly be deduced by looking at the Figure **8**: At negative values of η the second term of Equation 49 gives a *negative probability* in the region $\eta < 0$, i.e. using a simple Breit-Wigner (Cauchy) probability function too many events are generated in this region. Correspondingly, since the second term of Eq. 49 is an odd function, exactly the same fraction (distribution) of events is *missing* in the region $\eta > 0$. By mapping events with $\eta < 0$ over the $\eta = 0$ axis one thus solves both problems at the same time. Using the above re-sampling procedure the whole approach remains unitary, i.e. no events are rejected when there are no limits set on the value of η or they are symmetric $|\eta_{\min}| = \eta_{\max}$. In the contrary case, a small fraction of sampling values is rejected.

After some calculation the whole unitary procedure can thus be listed as follows:

- Calculate the kinematic limits $\eta_{\min}$ and $\eta_{\max}$.
- Calculate the *normalisation* factors $\Delta_1 = F_1(\eta_{\max}) - F_1(\eta_{\min})$, $\Delta_2 = F_2(\eta_{\max}) - F_2(\eta_{\min})$ and $\Delta_s = \Delta_1 + \Delta_2$; the term Δ_2 can actually be negative and thus does not represent proper normalisation.
- Obtain a (pseudo-)random number ρ_1.
- If $\rho_1 \leq \Delta_2/\Delta_s$ then:
  - Obtain a (pseudo-)random number ρ_2;
  - Construct η as:

$$
\begin{aligned}
X &= \Delta_2 \cdot \rho_2 + F_2(\eta_{\min}), \\
\eta &= \sqrt{(}e^{2X} - 1),
\end{aligned}
$$

    which is the inverse of the (normalised) cumulant $(F_2(\eta) - F_2(\eta_{\min}))/\Delta_2$.
  - Note that the condition $\rho_1 \leq \Delta_2/\Delta_s$ can be fulfilled only if $\Delta_2 \geq 0$, which means that $\eta_{\max}$ is positive and greater than $\eta_{\min}$.
- Conversely, if $\rho_1 > \Delta_2/\Delta_s$ then:
  - Obtain a (pseudo-)random number ρ_2;
  - Construct η as:

$$
\begin{aligned}
X &= \Delta_1 \cdot \rho_2 + F_1(\eta_{\min}), \\
\eta &= \tan(\frac{M_W \Gamma_W}{M_W^2} \cdot X)
\end{aligned}
$$

    which is the inverse of the (normalised) cumulant $(F_1(\eta) - F_1(\eta_{\min}))/\Delta_1$.
  - If the obtained η is less than zero then calculate the normalised probability densities:

$$
\begin{aligned}
P_1 &= \frac{1}{\Delta_1} \cdot \{\frac{M_W^2}{M_W \Gamma_W} \cdot \frac{1}{1+\eta^2}\} \\
P_s &= \frac{1}{\Delta_s} \cdot \{\frac{M_W^2}{M_W \Gamma_W} \cdot \frac{1}{1+\eta^2} + \frac{\eta}{1+\eta^2}\}
\end{aligned}
$$

  - Obtain a (pseudo-)random number ρ_3;
  - If $\rho_3 > P_s/P_1$ map $\eta \to -\eta$.
  - If the new η falls outside the kinematic limits $[\eta_{\min}, \eta_{\max}]$ the event is rejected.
  - Note also that the last mapping can only occur if the original η was negative, since $P_s < P_1$ only in the region $\eta < 0$.
- Calculate the value of s_W^* using the inverse of η definition:

$$
s_W^* = (M_W\ \Gamma_W) \cdot \eta + M_W^2 \tag{51}
$$

  The weight corresponding to the sampled value η is exactly:

$$
\Delta_s \cdot \frac{(s_W^* - M_W^2)^2 + M_W^2 \Gamma_W^2}{s_W^*}, \tag{52}
$$

  which is the (normalised) inverse of Equation 48 as requested.

As it turns out in subsequent generator level studies, this generation procedure gives much better agreement with the differential distributions than the *usual* (width independent) Breit-Wigner; an example obtained for the $q\bar{q} \to Wb\bar{b}$ process is shown in Figure **8**. The evident consequence is that the unweighting efficiency is substantially improved due to the reduction of the event weights in the high s_W^* region.

5.4.2 The Inclusion of Mass Effects in Propagator Sampling

Studies have shown that the $x^{-\nu}$ approximation works quite well for t-channel type propagators since the phase space suppression factor $\sqrt{\lambda}$ participates in the denominator, as shown in Eq. 40, and thus contributes only to the $x^{-\nu}$ slope. Contrariwise, while the $x^{-\nu}$ approximation still works reasonably well when describing the s-channel type propagators involving particles with high virtuality and/or decay products with low masses, it can be shown that this is not necessarily the case in the LHC environment, where the presence of massive decay products can significantly affect the invariant mass distributions. As it can be seen in Figure **9** the shape of the propagator dependence can be strongly suppressed by the phase space $\sqrt{\lambda}$ (boost) factor at low values; thus the sampling function approximation for non-resonant propagators could be approximated with something like:

$$f_{\rm NR}(s) = \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}}{s}\cdot\frac{1}{s^\nu} = \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}}{s^{\nu+1}} \tag{53}$$

and similarly

$$f_{\rm R}(s) = \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}}{s}\cdot\frac{\sqrt{s}}{(s-M^2)^2+M^2\Gamma^2} = \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}}{\sqrt{s}\cdot((s-M^2)^2+M^2\Gamma^2)} \tag{54}$$

for resonant propagators.

As it turns out the two functions cannot be sampled by the well known unitary algorithms (i.e. the biggest collection of recipes [57] yielded no results); already the integral values of the functions yield complicated expressions which cannot be easily calculated, let alone inverted analytically. The solution was to code numerical algorithms to calculate the integrals (i.e. cumulants) explicitly.

After the integrals are calculated, their inverse and the subsequent sampling value can again be obtained numerically. Namely, resorting to the original definition of the unitarity sampling recipe in Eq. 8, by replacing the normalised $\mathrm{g_i(x)}$ with:

$$g_i(x) \rightarrow \frac{f(x)}{\int\limits_{x_-}^{x_+} f(x)\,dx}, \tag{55}$$

which in turn gives:

$$\int\limits_{x_-}^{x} f(x)\;dx = r\cdot\int\limits_{x_-}^{x_+} f(x)\;dx, \tag{56}$$

where f(x) is the non-negative function one wants to sample from, $[\mathrm{x_-,x_+}]$ is the range of values of the parameter x we want to sample and r a pseudo random number $r\in[0,1]$. As already stated (c.f. Eq.8), in the case when the integral of the function f(x) is an analytic function, $F(x)=\int_{x_-}^{x} f(x)\,dx$, and has a known inverse $\mathrm{F^{-1}(x)}$ one can construct explicit unitary prescriptions by:

$$x = F^{-1}\left(r\cdot[F(x_+)-F(x_-)]+F(x_-)\right) \tag{57}$$

as given for two particular cases in Eq. 42,45.

In the cases the integral can not be inverted, the prescription of the Eq. 56 can directly be transformed into a zero-finding request; thus, since both the integral and the first derivative (i.e. the sampling function and its cumulant) are known, the Newton-Rhapson method is chosen as the optimal one for root finding:

$$g(x) = \left\{\int\limits_{x_-}^{x} f(x)dx - r\cdot\int\limits_{x_-}^{x_+} f(x)dx\right\} = 0 \tag{58}$$

$$g'(x) = \frac{d}{dx}\left\{\int\limits_{x_-}^{x} f(x)dx - r\cdot\int\limits_{x_-}^{x_+} f(x)dx\right\} = f(x) \tag{59}$$

With a sensible choice of starting points the procedure generally takes on the order of ten cycles until finding the root with adequate numerical precision. The overall generation speed is still deemed quite reasonable.

The integration of the phase-space suppressed resonant propagator of Eq. 54 yields a rather non-trivial expression:

$$\begin{aligned}
\int\limits_{(m_a+m_b)^2}^{s} f_{\rm R}(s)\,ds &= \int\limits_{(m_a+m_b)^2}^{s} \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}\,ds}{\sqrt{s}\cdot((s-M^2)^2+M^2\Gamma^2)} \\
&= \int\limits_{a}^{s} \frac{\sqrt{(s-a)(s-b)}\,ds}{\sqrt{s}\cdot((s-M^2)^2+M^2\Gamma^2)} \\
&= \frac{1}{\sqrt{-b}\,\Gamma\,M^2} \times \frac{-2\,i\,a\,b\,\Gamma}{(\Gamma^2+M^2)} \\
&\times \left\{\mathbf{F}\left[i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{a}}),\frac{a}{b}\right] - \mathbf{F}\left[i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{s}}),\frac{a}{b}\right]\right. \\
&+ (i\,\Gamma+M)\,(a+i\,(\Gamma+i\,M)\,M)\,(b+i\,(\Gamma+i\,M)\,M)\,\mathbf{\Pi}\left[\frac{M\,(-i\,\Gamma+M)}{b}, i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{a}}),\frac{a}{b}\right] \\
&+ (\Gamma+i\,M)\,(b+(-i\,\Gamma-M)\,M)\,(i\,a+(\Gamma-i\,M)\,M)\,\mathbf{\Pi}\left[\frac{M\,(i\,\Gamma+M)}{b}, i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{a}}),\frac{a}{b}\right] \\
&- (i\,\Gamma+M)\,(a+i\,(\Gamma+i\,M)\,M)\,(b+i\,(\Gamma+i\,M)\,M)\,\mathbf{\Pi}\left[\frac{M\,(-i\,\Gamma+M)}{b}, i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{s}}),\frac{a}{b}\right] \\
&- \left.(\Gamma+i\,M)\,(b+(-i\,\Gamma-M)\,M)\,(i\,a+(\Gamma-i\,M)\,M)\,\mathbf{\Pi}\left[\frac{M\,(i\,\Gamma+M)}{b}, i\,\mathrm{arcsinh}(\frac{\sqrt{-b}}{\sqrt{s}}),\frac{a}{b}\right]\right\}
\end{aligned} \tag{60}$$

where the variables a, b stand for $\mathrm{a} = (\mathrm{m_a}+\mathrm{m_b})^2$ and $\mathrm{b} = (\mathrm{m_a}-\mathrm{m_b})^2$ and the functions $\mathbf{F}[\varphi,\mathrm{k}]$ and $\mathbf{\Pi}[\varphi,\mathrm{k},\mathrm{n}]$ are the Legendre's incomplete elliptic integrals of the second and third kind with complex arguments. In order to perform the calculations the latter functions had to be coded from scratch since they were not found in any (publicly available) computer libraries or code repositories. The prescriptions for calculating them were found in [59]; the results were checked against the values given by Mathematica$^{\rm TM}$.
In the special case $\mathrm{m_a} = \mathrm{m_b}$ the above expression simplifies into:

$$\begin{aligned}
\int\limits_{(2m_a)^2}^{s} f_{\rm R}(s)\,ds &= \int\limits_{(2m_a)^2}^{s} \frac{\sqrt{\lambda(s,m_a^2,m_a^2)}\,ds}{\sqrt{s}\cdot((s-M^2)^2+M^2\Gamma^2)} \\
&= \int\limits_{a}^{s} \frac{\sqrt{(s-a)}\,ds}{\cdot((s-M^2)^2+M^2\Gamma^2)} \\
&= \frac{1}{\Gamma\,M\,\sqrt{a+(-i\,\Gamma-M)\,\,M}} \\
&\times \left\{(i\,a+(\Gamma-i\,M)\,\,M)\,\,\arctan(\frac{\sqrt{-a+z}}{\sqrt{a+(-i\,\Gamma-M)\,\,M}})\right. \\
&- \left. i\,\sqrt{a+(-i\,\Gamma-M)\,\,M}\,\sqrt{a+i\,\,(\Gamma+i\,M)\,\,M}\,\,\arctan(\frac{\sqrt{-a+z}}{\sqrt{a+i\,\,(\Gamma+i\,M)\,\,M}})\right\}
\end{aligned} \tag{61}$$

The result of integrating the phase-space suppressed non-resonant propagator (Eq. 53) yields a similarly non-trivial result:

$$\begin{aligned}
\int\limits_{(m_a+m_b)^2}^{s} f_{\rm NR}(s)\,ds &= \int\limits_{(m_a+m_b)^2}^{s} \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}\,ds}{s^{\nu+1}} \\
&= \frac{1}{2\,\sqrt{1-\frac{s}{a}}\,\nu}\left\{\frac{-2\,\sqrt{(a-s)\,\,(b-s)}\,\mathbf{F_1}\left[-\nu,-\left(\frac{1}{2}\right),-\left(\frac{1}{2}\right),1-\nu,\frac{s}{a},\frac{s}{b}\right]}{s^{\nu}\,\sqrt{1-\frac{s}{b}}}\right. \\
&+ \left.\frac{\sqrt{\pi}\,\sqrt{(-a+b)\,\,(a-s)}\,\mathbf{\Gamma}\left[1-\nu\right]\,\mathbf{F}\left[-\nu,-\left(\frac{1}{2}\right),\frac{3}{2}-\nu,\frac{a}{b}\right]}{a^{\nu}\,\sqrt{1-\frac{a}{b}}\,\mathbf{\Gamma}\left[\frac{3}{2}-\nu\right]}\right\}
\end{aligned} \tag{62}$$

where the function $\mathrm{F}[\alpha,\beta,\gamma,\mathrm{x}]$ is the Gauss Hypergeometric function and the $\mathrm{F}_1[\alpha,\beta,\beta',\gamma,\mathrm{x},\mathrm{y}]$ is the two-parameter (Appell) Hypergeometric function [60]. Both functions can be calculated by using the prescriptions in [60]; it however turns out that the calculation of the $\mathrm{F}_1[\alpha,\beta,\beta',\gamma,\mathrm{x},\mathrm{y}]$ to a certain (high) precision is almost two times slower than the explicit numerical calculation of the integral to the same precision. Subsequently the numerical evaluation of the Gauss Hypergeometric function $\mathrm{F}[\alpha,\beta,\gamma,\mathrm{x}]$ was retained since it participates in the $\mathrm{m_a}=\mathrm{m_b}$ simplification and the calculation of the integral was done by using a 50-point Gauss-Legendre quadrature with $\sqrt{s}$ weight function; the weights were calculated by [61]. The implementation of the (Appell) Hypergeometric function calculation was used as a cross-check to confirm the correct implementation and precision of the numerical method.
As already mentioned, the above integral again simplifies for $\mathrm{m_a}=\mathrm{m_b}$:

$$\begin{aligned}\int\limits_{(2m_a)^2}^{s} f_{\mathrm{NR}}(s)\,ds &= \int\limits_{(2m_a)^2}^{s} \frac{\sqrt{\lambda(s,m_a^2,m_b^2)}\,ds}{s^{\nu+1}} \\ &= \int\limits_{a}^{s} \frac{\sqrt{(s-a)}\,ds}{s^{\nu+\frac{1}{2}}} \\ &= \frac{2}{3}\,a^{1-\nu}\,s^{\frac{3}{2}}\,\mathbf{F}\left[\frac{3}{2},\nu+\frac{1}{2},\frac{5}{2},-s\right],\end{aligned} \tag{63}$$

and the Gauss Hypergeometric function $\mathrm{F}[\alpha,\beta,\gamma,\mathrm{x}]$ is in this case calculated by the methods described in [60] with some improvements analogous to the ones described e.g. in [62].

5.5 Application of the Phase Space Generation Algorithms

The AcerMC 2.0 Monte-Carlo generator uses the multi-channel phase space generation where each channel corresponds to an expected phase space topology as derived from the participating Feynman diagrams. In AcerMC 2.0 this information was obtained from the modified MadGraph[9] program which also supplied the probability amplitudes for the implemented processes. Each channel topology was in turn constructed from the t-type and s-type modules and sampling functions described in this paper together with some additional importance sampling techniques for space angles and rapidity distributions described in detail elsewhere [1, 38, 39]. The unknown slope parameters (denoted ν in the text, c.f. Eq. 53) of the invariant mass sampling functions for non-resonant propagators were obtained by short training runs of the program on a process by process basis.

As a further step the multi-channel self-optimisation procedure was implemented in order to minimise the variance of the event weights further [41]. Eventually, additional smoothing of the phase space was obtained by using a modified VEGAS routine to improve the generation efficiency (see next Section).

The procedure of multi-channel importance sampling used in the event generation can briefly be outlined as follows. The analytically integrable function $g(\vec{\Phi})$ (c.f. Eq. 9, which aims to approximate the peaking behaviour of the differential cross-section dependence on various kinematic quantities is introduced into the differential cross-section equation as a weighted sum of several channels $g_i(\vec{\Phi})$, each adapted to a certain event topology:

$$g(\vec{\Phi}) = \sum_i \alpha_i \cdot g_i(\vec{\Phi}). \tag{64}$$

The values of relative weights α_i are determined from multi-channel self-optimisation procedure in order to minimise the variance of the weights $w(\vec{\Phi})$ [41]. The phase space points are than sampled from the function $g(\vec{\Phi})$, first by randomly choosing a channel i according to the relative frequencies α_i and then deriving the required four momenta from the chosen $g_i(\vec{\Phi})$ using unitary[11] algorithms [38].

A few representative invariant mass distribution comparisons between the implemented sampling functions and the actual differential distributions are shown in Figure **9**.

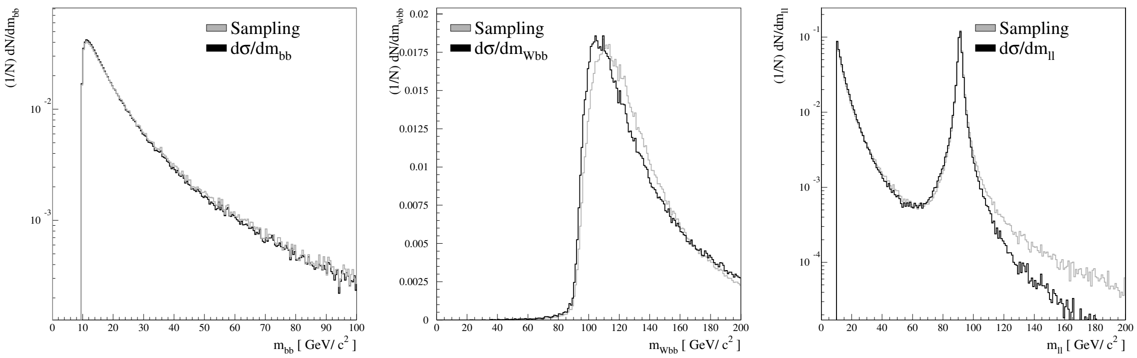


Fig. 9: A few representative invariant mass distribution comparisons between the (normalised) sampling functions and the normalised differential cross-section as obtained with AcerMC 2.0 Monte–Carlo generator. Left: The invariant mass of the $\mathrm{b\bar{b}}$ pair in the process $\mathrm{u\bar{d}} \to \mathrm{W^+g^*} \to \mathrm{l^+}\nu_\mathrm{l}\mathrm{b\bar{b}}$. Center: The invariant mass of the $\mathrm{Wb\bar{b}}$ system (equivalently the hard centre-of-mass energy $\sqrt{\hat{\mathrm{s}}}$) for the same process. Right: The invariant mass of the $\ell\bar{\ell}$ pair in the process $gg \to Z^0/\gamma^* b\bar{b} \to \ell\bar{\ell}b\bar{b}$. All the distributions were obtained using the prescriptions of this paper without the adaptive algorithms also used in the AcerMC 2.0 Monte–Carlo generator. As one can see the approximations used seem to work quite well.

[11] Unitary in this context meaning that there is no event rejection in the algorithm.

5.5.1 Modified VEGAS Algorithm

Using the described multi-channel approach, the total generation (unweighting) efficiency amounts to about $3 - 10\%$ depending on the complexity of the chosen process. In order to further improve the efficiency, a set of modified VEGAS [8] routines was used as a (pseudo-)random number generator for sampling the peaking quantities in each kinematic channel. The conversion into a *(pseudo-)random number generator* consisted of re-writing the calling routines so that instead of passing the analysed function to VEGAS for sampling and integration, VEGAS calls produce only (weighted) random numbers in the region $[0, 1]$ and the corresponding sampling weight, while the VEGAS grid training is done using a separate set of calls.

After training all the sampling grids (of dimensions 4-7, depending on the kinematic channel), the generation efficiency increased to the order of $6 - 14\%$. The motivation for this approach was that in unitary algorithms only a very finite set of simple sampling functions is available, since the functions have to have simple analytic integrals for which an inverse function also exists. Consequently, the non-trivial kinematic distributions can not be adequately described by simple functions at hand in the whole sampling domain (e.g. the τ distribution, c.f. Figure **10**) and some additional smoothing might be welcome. In addition, the random numbers distributions should, due to the applied importance sampling, have a reasonably flat behaviour to be approached by an adaptive algorithm such as VEGAS[12].

The further modification of VEGAS, beside adapting it to function as a (pseudo-)random number generator instead of the usual *integrator*, was based on the discussions [44, 45] that in case of event generation, i.e. unweighting of events to the weight one, reducing the maximal value of event weights is in principle of higher importance than achieving the minimal weight variance. Since the VEGAS algorithm was developed with the latter scope, some modification of the algorithm was necessary. As it turned out, the modification was fairly easy to implement: Instead of the usual cumulants:

$$< I >_{\mathrm{cell}} \;= \sum_{\mathrm{cell}} \mathrm{wt}_i, \tag{65}$$

according to the size of which VEGAS decides to split its cells, the values:

$$< F >_{\mathrm{cell}} \;= \Delta_{\mathrm{cell}} \cdot \mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}} \; - \sum_{\mathrm{cell}} \mathrm{wt}_i, \tag{66}$$

were collected and used as the splitting criterion. The above value (called *loss integral* in [45]) is basically a measure of the deviation between the maximal weight sampled in the given cell $\mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}}$ and the average weight in the cell $< \mathrm{wt}_{\mathrm{cell}} >= (\sum_{\mathrm{cell}} \mathrm{wt}_{\mathrm{i}})/\Delta_{\mathrm{cell}}$ (the quantity Δ_{cell} denoting the cell width, i.e. the integration range). Re-writing the above expression as:

$$< F >_{\mathrm{cell}} \;= \left(\Delta_{\mathrm{cell}} \cdot \mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}}\right) \cdot \left\{ 1 - \frac{< \mathrm{wt}_{\mathrm{cell}} >}{\mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}}} \right\} \tag{67}$$

clearly indicates that the value $< F >_{\mathrm{cell}}$ is actually a measure of the generation *inefficiency* in the cell, since the term in the curly brackets is equivalent to one minus the generation efficiency $< \mathrm{wt}_{\mathrm{cell}} > /\mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}}$. In addition, the inefficiency is weighted with the *crude*/maximal estimation of the function integral over the cell $\Delta_{\mathrm{cell}} \cdot \mathrm{wt}_{\mathrm{cell}}^{\mathrm{max}}$ and cells with the highest $< F >_{\mathrm{cell}}$ are split.

This method is of relevance because the VEGAS cells are actually projections of the whole phase space on the (chosen) side axes, i.e. VEGAS cannot isolate a maximal weight in a certain point in phase-space and build a cell around it, which in principle would be an ideal solution. An implementation with this scope in view has been made in FOAM [45], nevertheless we have not found it competitive with respect to the modified VEGAS for the given application.

The thus modified ac-VEGAS algorithm further increased the unweighting efficiency for almost a factor of two.

One of the sampling distributions is shown in Fig. **10** as a gray histogram (marked *channel*) and the actual (*generated*) differential cross-section dependence is drawn in black. In the first figure, the random variable used for sampling values from $1/\tau^{\mu}$ distribution was drawn from a flat probability in the interval $[0, 1]$; in the second plot the ac-VEGAS algorithm was used to give an optimal grid for sampling the random variables needed for parameter generation (the grid is trained for each kinematic channel separately, the sum of all channels is shown in the plot). The improvement is evident; one has to stress that the use of ac-VEGAS algorithm to generate the values of τ directly would be much less efficient since VEGAS gives a grid of 50 bins/dimension, which would give a very crude description of the τ distribution compared to the one at hand.

Observing the distributions of the event weights before and after the inclusion of the modified ac-VEGAS algorithm (Fig. **11**) it is evident that ac-VEGAS quite efficiently clusters the weights at lower values. Note that the principal effect

[12] At this point also a disadvantage of using the adaptive algorithms of the VEGAS type should be stressed, namely that these are burdened with the need of training them on usually very large samples of events before committing them to event generation.

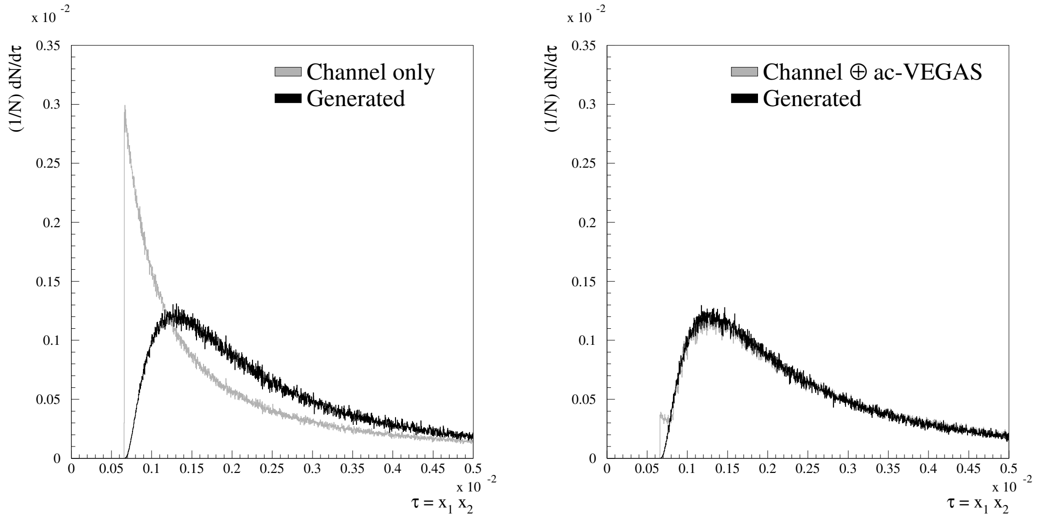


Fig. 10: *Comparison between the sampling distribution for the $\tau = \hat{s}/s \in [\tau_{\min}, 1]$ variable in $gg- > t\bar{t}b\bar{b}$ process before and after the application of modified* ac-VEGAS *[8] smoothing procedure (light gray histogram). The generated (normalised) differential cross-section is also drawn (black histogram, labelled* Generated*).*

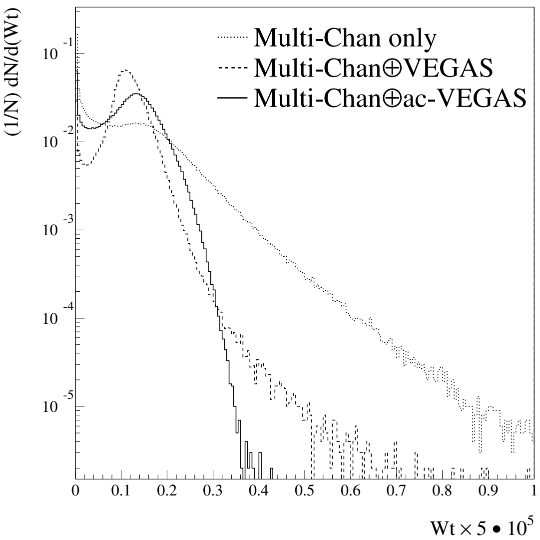


Fig. 11: *The distribution of event weights using only the Multi-Channel approach (dotted histogram) and after application of* VEGAS *(dashed histogram) and* ac-VEGAS *(full histogram) algorithms in the $gg \to (Z^0 \to) l\bar{l}b\bar{b}$ process.*

of original VEGAS is indeed to cluster event weights in a narrow region, nevertheless a tail towards the high-weight region remains. On the other hand, the ac-VEGAS efficiently reduces the tail in the high weight region; only a few of the event weights still retain their large values, thus reducing the generation efficiency. Given the difference in distributions, the observed increase of the generation efficiency seems relatively modest. To better understand this result one should consider that the formula for the MC generation efficiency is given by:

$$\epsilon = \frac{< \mathrm{wt} >}{\mathrm{wt}_{\max}}, \tag{68}$$

where $< \mathrm{wt} >$ is the average weight of the sample and equals the total event cross-section, while $\mathrm{wt}_{\max}$ represents the maximum event weight in the applied generation procedure and is determined through a pre-sampling run with a high statistic. Since the average weight $< \mathrm{wt} >$ equals the total cross-section of the process, it remains (necessarily) unchanged after the application of the VEGAS refining; consequently the change of efficiency results in the reduction of the maximum weight $\mathrm{wt}_{\max}$ by approximately a factor two, which is from technical point of view quite an achievement.

A further step to profit from the clustering of weights induced by ac-VEGAS is to adopt a re-definition of the MC generation efficiency as proposed by [43, 45]. In this approach, the alternative definition of $\mathrm{wt}_{\max}$ is: For a given precision level $\alpha << 1$, the $\mathrm{wt}_{\max}$ is determined from the total weight distribution in such a way that the contribution of the events exceeding this value to the total weight sum (i.e. cross-section integral) equals α. Such a quantity is referred to as $\mathrm{wt}^{\alpha}_{\max}$ and the efficiency expression becomes:

$$\epsilon = \frac{< \mathrm{wt} >}{\mathrm{wt}^{\alpha}_{\max}}. \tag{69}$$

The argument presented in [43, 45] seems to be quite reasonable since the *true* event weight is in any case only estimated from a finite sample of events and the new definition simply takes into account a certain level of accuracy in the maximum weight determination. In addition, certain very weak singularities that might exist in the simulated process and might occasionally result in a very high event weight are automatically taken into account. The use of new $\mathrm{wt}^{\alpha}_{\max}$ consequently results in a generation efficiency of about $\epsilon \geq 20\%$ for all the implemented processes, which is a significant improvement in terms of time needed for MC generation.

5.5.2 Colour Flow Information

Before the generated events are passed to PYTHIA/HERWIG to complete the event generation, additional information on the colour flow/connection of the event has to be defined. Below we discuss the implemented method of the colour flow determination on the example of two processes, $gg \to t\bar{t}b\bar{b}$ and $q\bar{q} \to t\bar{t}b\bar{b}$.

For the process $gg \to t\bar{t}b\bar{b}$ six colour flow configurations are possible, as shown in Figure **12**. With 36 Feynman diagrams contributing to the process and at least half of them participating in two or more colour flow configurations, calculations by hand would prove to be very tedious. Consequently, a slightly modified colour matrix summation procedure from MADGRAPH [9] was used to determine the colour flow combinations of the diagrams and the corresponding colour factors. The thus derived squared matrix elements for separate colour flow combinations $|\mathcal{M}_{\mathrm{flow}}|^2$ were used as sampling weights on an event-by-event basis to decide on a colour flow configuration of the event before passing it on to PYTHIA/HERWIG for showering and fragmentation. The procedure was verified to give identical results regarding the colour flow combinations and corresponding colour factors when applied to the processes published in [46]. As one can see this approach neglects the interference terms between the distinct colour-ordered amplitudes and is indeed exact only in the $N_C \to \infty$ limit[13] [47, 48]. Since there is no *a priori* rule of how to split the interference terms between the colour ordered amplitudes this approach is generally deemed to be the best one can do; recent developments in this field [49] however suggest additional improvements to the method that indeed might be incorporated into later versions of **AcerMC**.

The colour flow configuration in the $q\bar{q} \to t\bar{t}b\bar{b}$ channel is much simpler since only two colour flow topologies exist (Fig. **13**); the choice between the two has been solved in a manner identical to the one for the $gg \to t\bar{t}b\bar{b}$ process, as described above.

Some specifications of the implemented matrix-element-based processes: number of Feynman diagrams, channels used in the phase-space generation and colour flow configurations are collected in Table **12**. The *control* processes ID=91-94 are omitted from the table due to their simplicity.

5.6 The α_{QED} and α_s calculations

Native functions of running $\alpha_{\mathrm{QED}}(Q^2)$ and $\alpha_s(Q^2)$ have been implemented inside **AcerMC** with the main objective of providing a means to keep the (total) cross-sections of the processes unchanged when interfacing with the two supervising

[13] The matrix elements used in the cross-section calculation and event generation are of course complete and do not employ any approximation.

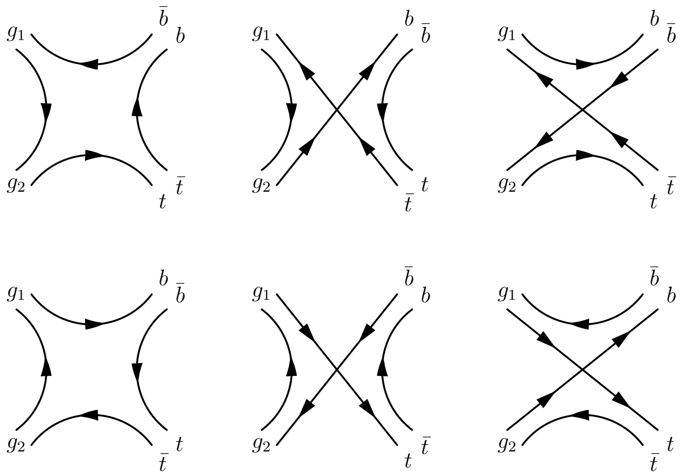


Fig. 12: *A diagrammatic representation of the six colour flow configurations in the process $gg \to t\bar{t}b\bar{b}$. Certain colour combinations, leading for example to colourless (intermediate) gluons, are not allowed.*

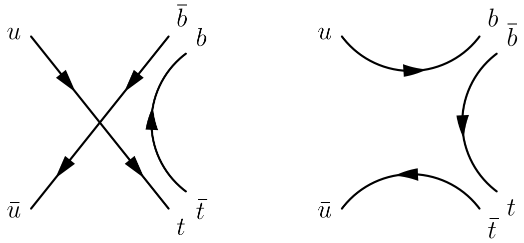


Fig. 13: *A diagrammatic representation of the two colour flow configurations in the process $q\bar{q} \to t\bar{t}b\bar{b}$.*

generators, since the implementations of the two functions in PYTHIA and HERWIG differ to some extent. Especially the $\alpha_s(Q^2)$ is subject to experimental and theoretical uncertainties, however obtaining a different cross-sections for the same **AcerMC** process due to different interface, could be regarded (at least to some extent) as an inconsistency[14].

- $\boldsymbol{\alpha_{\mathrm{QED}}}$ is implemented in **AcerMC** using the formulae given in [63] and is in complete accordance with the implementations in PYTHIA and HERWIG apart from the updated hadronic component published recently by Burkhardt *et. al.* [64]. As one can see in Figure **14**, the latter minimally lowers the α_{QED} values.
- $\boldsymbol{\alpha_s}$ has one and three loop implementations in **AcerMC** following the calculations of W. J. Marciano [65] and using $\Lambda_{\bar{MS}}^{(nf)}$ transformations for flavour threshold matches. The three loop version gives good agreement with the HERWIG implementation (both functions have been set to the same $\Lambda_{\bar{MS}}^{(nf=5)}$ value) as one can see in Figure **14**. The PYTHIA two loop implementation deviates somewhat from the latter two; the kinks observed in the plot are due to approximate $\Lambda_{\bar{MS}}^{(nf)}$ transformations at flavour thresholds, which are exact to one loop only.

Although the **AcerMC** and PYTHIA one loop implementations are identical in form the resulting values differ by a small amount because the default PYTHIA implementation reads the $\Lambda_{\bar{MS}}^{(nf=4)}$ value from LHAPDF instead of the $\Lambda_{\bar{MS}}^{(nf=5)}$ one used by **AcerMC** and HERWIG; the difference thus occurs due to $\Lambda_{\bar{MS}}^{(nf)}$ propagation at flavour thresholds.

[14]The values will still differ by a small amount in processes containing W bosons (processes 3,4) due to different values of the CKM matrix in the two supervising generators.

Table 12: *Some details on matrix-element-based process implementation in* **AcerMC** *library. In case of $q\bar{q}$ initial state the number of Feynman diagrams corresponds to one flavour combination. The $f = e, \mu, \tau, b$.*

Process id	Process specification	Feyn. diagrams	Channels	Colour flows
1	$gg \to t\bar{t}bb$	36	12	6
2	$q\bar{q} \to t\bar{t}bb$	7	5	2
3	$q\bar{q} \to W(\to \ell\nu)bb$	2	2	1
4	$q\bar{q} \to W(\to \ell\nu)tt$	2	2	1
5	$gg \to Z/\gamma^*(\to \ell\ell)bb$	16	6	2
6	$q\bar{q} \to Z/\gamma^*(\to \ell\ell)bb$	8	6	1
7	$gg \to Z/\gamma^*(\to ff, \nu\nu)tt$	16	6	2
8	$q\bar{q} \to Z/\gamma^*(\to ff, \nu\nu)tt$	8	6	1
9	$gg \to (Z/W/\gamma^* \to)t\bar{t}bb$	72	20	12
10	$q\bar{q} \to (Z/W/\gamma^* \to)t\bar{t}bb$	28	12	5
11	$gg \to (t\bar{t} \to)ffbffb$	3	2	2
12	$q\bar{q} \to (t\bar{t} \to)ffbffb$	1	1	1
13	$gg \to (WWbb)ffffbb$	31	13	2
14	$q\bar{q} \to (WWbb \to)ffffbb$	14	7	1
15	$gg \to tttt$	72	10	6
16	$q\bar{q} \to tttt$	14	4	2

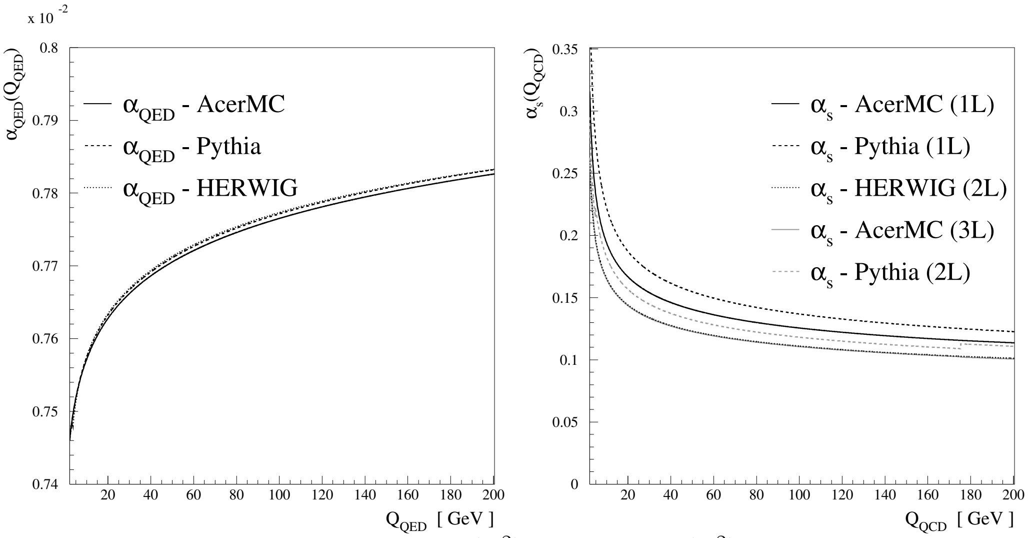


Fig. 14: *Comparison between the $\alpha_{\rm QED}(Q^2)$ (**Left**) and $\alpha_s(Q^2)$ (**Right**) implementations in* **AcerMC**, PYTHIA *and* HERWIG. *For $\alpha_s(Q^2)$ calculations with different loop orders (*L*) are given where applicable.*

6 Structure of the package

The **AcerMC** package consists of a library of the matrix-element-based generators for selected processes, interfaces to the PYTHIA 6.2 , ARIADNE 4.1 and HERWIG 6.5 generators, sets of data files and three main programs: demo_hw.f, demo_py.f and demo_ar.f. Provided makefiles allow to build the executables with either of these generators as the *supervising generator*: demo_hw.e, demo_py.e and demo_ar.e.

6.1 Main event loop and interface to PYTHIA/HERWIG, TAUOLA and PHOTOS

The main event loop is coded in the demo_hw.f, demo_py.f and demo_ar.f files, where the opening/closing of the input/output files, reading of the data-cards and event-loop execution is performed. Main event loop consists only of calls to the acermc_py, acermc_ar or acermc_hw subroutines, with parameter MODE = -1, 0, 1 respectively set for initialisation, generation and finalisation of the event loop. The call to acermc_xx activates respective procedures of the supervising generator, which in turn activates the acevtgen procedure steering the native **AcerMC** generation of the matrix element event. Fig. **15** illustrates this calling sequence in some details.

As one can deduce from the diagram in Fig. **15**, certain functions called by **AcerMC**, as e.g. pseudo-random number generator acr are re-routed through the interfaces to the linked supervising generator, depending on the choice at compilation time (e.g. acr function giving (pseudo-)random numbers is linked to either pyr or hwrgen as shown in the plot), providing the internal consistency of the package. The generated event is rewritten to the format required by the supervising generator by means of the acdump_xx routines. While ARIADNE provides an alternative (colour-dipole) based implementation of initial and final state radiation it relies on PYTHIA for hadronisation and particle decays.

The pythia_ac.f and herwig_ac.f files contain sets of re-routing/interface functions, specialised for the respective supervising generator. The main library of **AcerMC** is well screened from dependencies on the supervising generator, all dependencies are hidden in herwig_ac.f and pythia_ac.f respectively.

The PYTHIA, ARIADNE and HERWIG libraries remain essentially untouched[15], without introducing any dependencies on the **AcerMC** code. The input cards are common for all interfaces.

The calling sequence is further enhanced by the optional calls to the external TAUOLA [6] library, which handles the τ-lepton decays using the spin information of the hard process, and the PHOTOS [5] library which adds the final state QED radiation to final state leptons and hadrons. The two interfaces are controlled by two additional input cards. The details of the interface are given in Section 7.3.

[15]For specification of exceptions see Section 7.4, 7.5. and 7.6

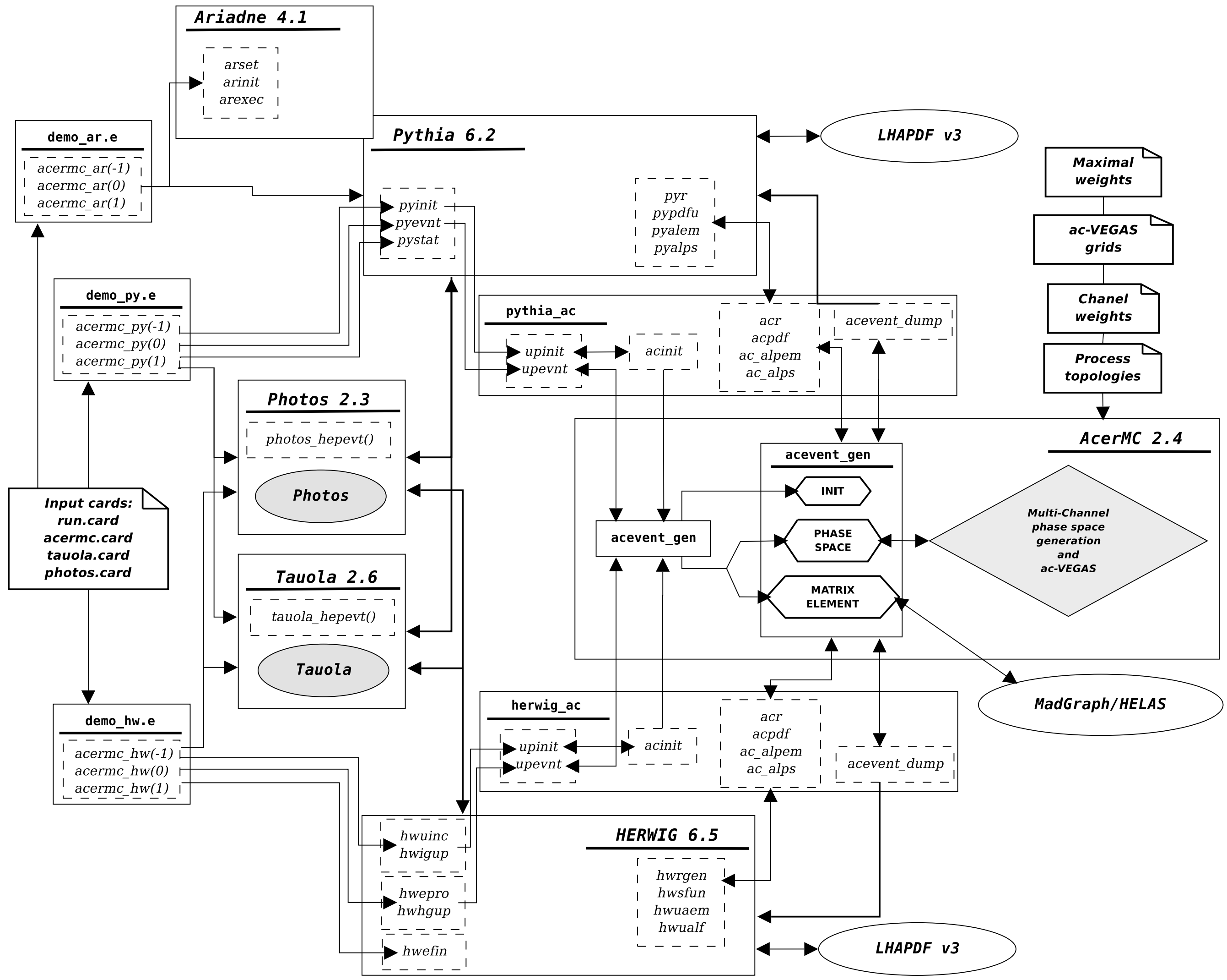


Fig. 15: *The calling sequence of the main event generation routine* acevent_xx. *The routine is called either through* demo_py → acermc_py *sequence when interfacing the* PYTHIA 6.2 *generator or* demo_hw → acermc_hw *sequence when the* HERWIG 6.5 *is linked. When the* ARIADNE 4.1 *setup is called via acermc_ar calls it still relies on* PYTHIA 6.2 *for hadronisation and particle(resonance) decays. The structure of the interface subroutines and relations with the corresponding ones from supervising generators and/or external libraries is also evident.*

6.2 Structure of the AcerMC matrix-element and phase-space code

The **AcerMC** core code performs the generation of a matrix-element-based event. Fig. **16** illustrates the calling sequence for generating an unweighted event. The steering subroutine is called acevent_gen which subsequently constructs the weight by calling the Madgraph/HELAS subroutines for the matrix element evaluati To stress again, this subroutine calls only a sequence of the native **AcerMC** subroutines, any call to the supervising generator goes via the respective interface function/subroutine. A more detailed representation of calling sequence is shown in the Figure **16**.

Code for the phase space generation is since the version 2.0 greatly simplified compared to the earlier versions. Code for

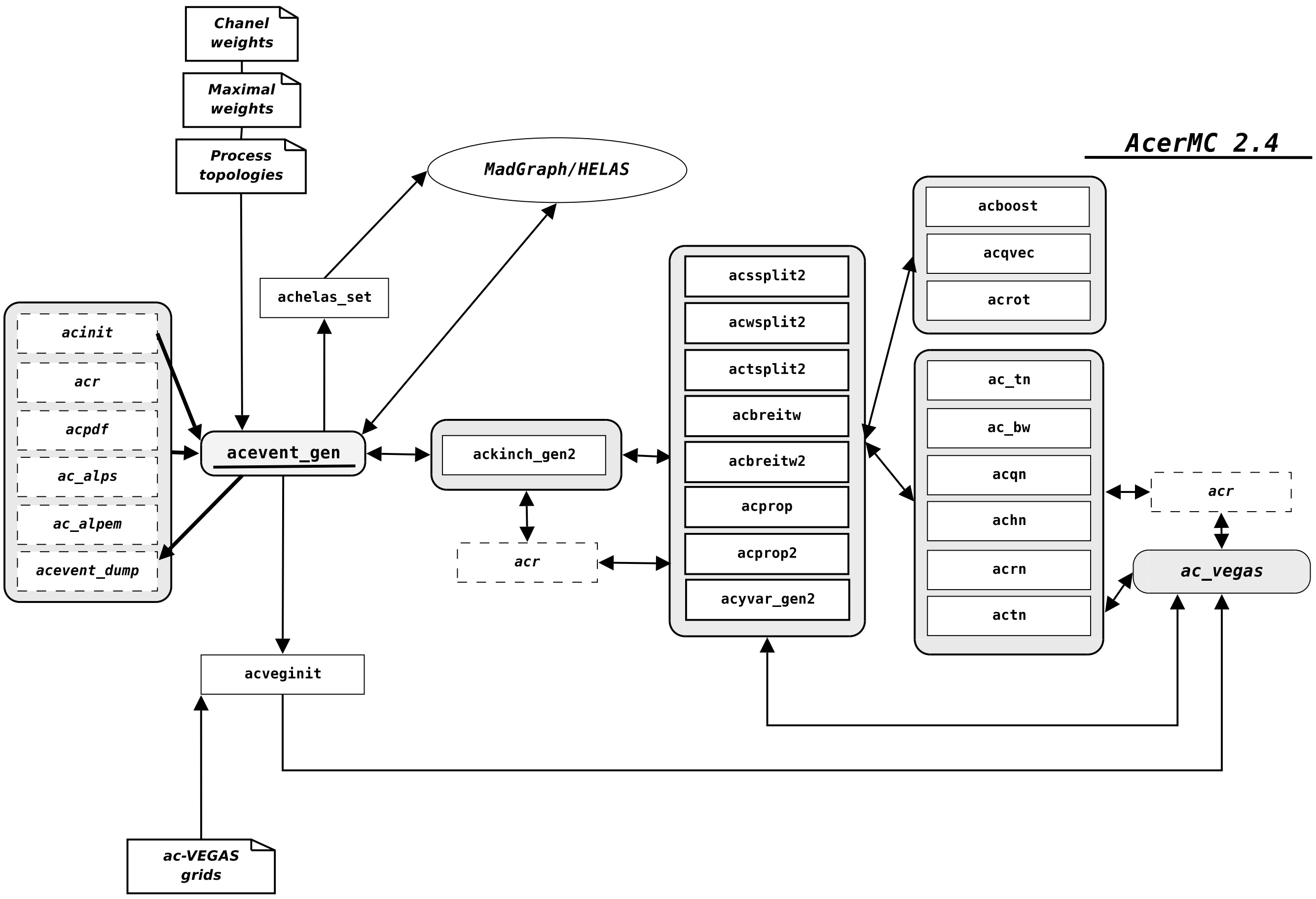


Fig. 16: *The event generation sequence controlled by* acevent_gen *subroutine. Phase space generation is sequenced by calling the* ackinch_gen2 *routine to obtain the incoming gluon and the outgoing four-momenta of the participating particles. The latter routine handles the possible momenta permutations and calls the explicit four-momenta generation (and PS weight calculation) in the sequence prescribed by the event topology. These (channel-specific) routines are constructed from common building blocks listed in the next two columns. The* acevent_gen *routine also initialises* MADGRAPH/HELAS *package and retrieves the matrix element values. All the generated four-momenta, as well as the event weight are finally passed back to the supervising generator via the* acevent_dump *call.*

matrix element calculations is grouped together for all processes in subdirectory matel. Code with different utility subroutines, e.g. kinematic transformations used by all subprocesses, is in the subdirectory common. Subdirectory interface contains code with interfaces to supervising generators, finally subdirectory include contains all include files. The overall view on the structure of the **AcerMC** directories is shown in Fig. **17**.

The core code builds one library libacermc.a.

6.3 Data files for the phase-space optimisation

The **AcerMC** matrix-element-based generators are very highly optimised, using multi-channel optimisation and additional improvement with the ac-VEGAS grid. The generation modules require three kinds of the input data to perform the generation of unweighted events:

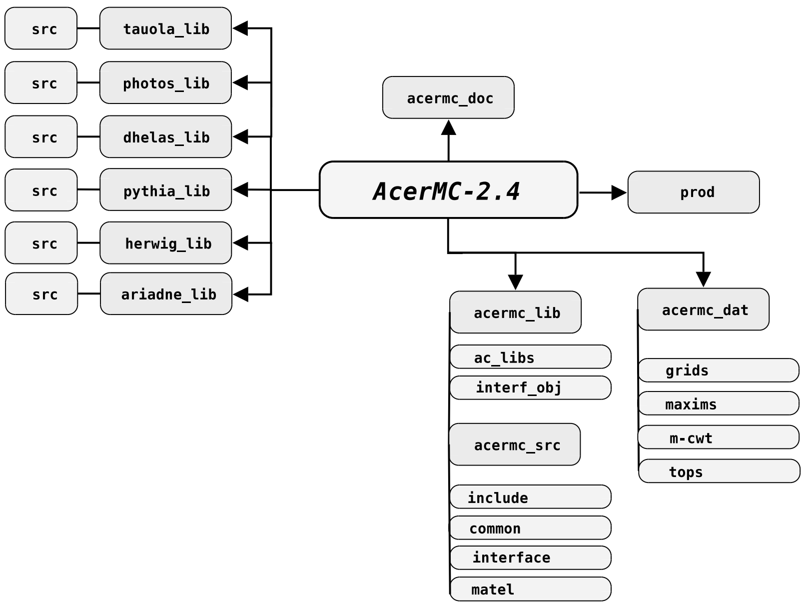


Fig. 17: *The structure of the* **AcerMC** *directories.*

- A file descrbing the construction of the topologies relevant for the chosen process, the implemented sets are stored in the directory acermc_dat/tops.
- A file containing the list of the values of relative channel weights obtained by the multi-channel optimisation, defaults being stored in acermc_dat/m-cwt.
- A file containing the pre-trained ac-VEGAS grid, the pre-trained (default) ones located in acermc_dat/grids.
- A file containing the maximum weight $\mathrm{wt}_{\mathrm{max}}$, α-cutoff maximum weight $\mathrm{wt}^{\alpha}_{\mathrm{max}}$ and the 100 events with the highest weights, the default ones being provided in acermc_dat/maxims.

In case of changing the default running conditions, like parton density functions or centre-of-mass energy, the user should repeat the process of preparation of the listed data files containing the inputs for the phase-space generator modules in order to preserve the initial event generation efficiency.

The reading sequence of data files inside **AcerMC** is shown in Figure **18**.

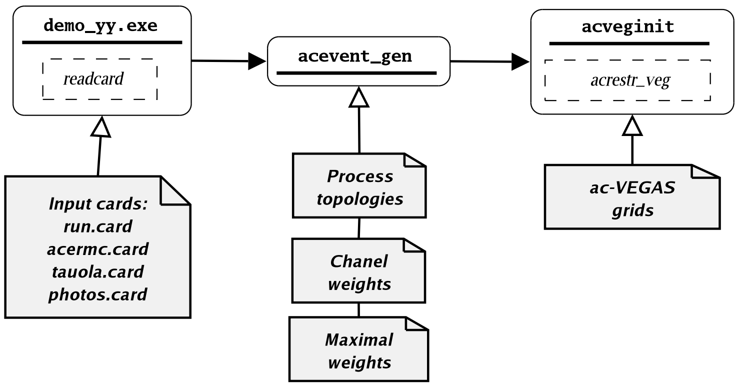


Fig. 18: *The reading sequence of the input files and the performing subroutines in the* **AcerMC** *code.*

Pre-trained data sets are obtained using $\sqrt{s} = 14$ TeV, PYTHIA default $\alpha_s(Q^2)$ and $\alpha_{\rm QED}(Q^2)$ and CTEQ5L (parametrised) parton density function set and are provided for each implemented process[16]. For these, the relative channel weights are stored in the INCLUDE files in acermc_src/include/chanwt_xx.inc where xx denotes the process id (c.f. Table **11**); the default/pre-trained ac-VEGAS grids are listed in the directory acermc_dat/grids/vscalA_xxYYY.veg, where A denotes the scale choice of the process xx and YYY denotes the cutoff value of the m_{Z^0/γ^*} for the **AcerMC** processes xx = 05 → 08. The files containing the maximal weights $\rm wt_{max}$ and $\rm wt^{\alpha}_{max}$ as well as the 100 events with the highest weights are stored in the directory acermc_dat/grids/vtmaxA_xxYYY.dat, following the same labelling convention. Both the trained ac-VEGAS grids and the weight files were obtained from test runs with at least $2 \cdot 10^6$ weighted events being generated.

The number of required input files might at first look seem large, considering that many event generators do not require any input files for operation; the difference is not in so much in the complexity of the phase space generation as in the fact that many event generators require a *warming run* instead, i.e. before the generation of unweighted events is performed a certain number of weighted events (typically of the order of 10^4) is generated in order to obtain the relative multi-channel weights (in case multi-channel phase space generation is used) and/or the optimised VEGAS grid and/or an estimate of the maximal weight. Such an approach can have an advantage when event generation is very fast and the phase space regions with the highest weights are well known (as done for the $2 \to 2$ processes in PYTHIA); on the other hand, when the phase space topology of the process is more complex and the event generation is comparatively slow, generating a relatively small number of e.g. 10^4 weighted events *every time* a generator is started can become CPU wasteful and/or inaccurate in terms of maximum weight estimation.

Reasonably accurate estimation of the latter is namely crucial for correct event unweighting; event generators using *warming-up* method for maximal weight search often find still higher weights during the production run and reset the maximal weight accordingly. In this case however, statistically correct approach would be to reject all events generated beforehand and start the event generation anew, which is almost never implemented due to the CPU consumption and the possibility of hitting a weak singularity (the same argument leads to the definition of the $\rm wt^{\alpha}_{max}$, c.f. Section 5.5.1). With a small pre-sampled set the generator can however badly under-estimate the maximum weight and a large number of events can be accepted with a too-high probability. The only hope of obtaining correct results is in such cases that the weight *plateau* will be hit sufficiently early in the event generation process. Consequently, such approach can be very dangerous when generating small numbers of events[17].

In contrast to the *warming-up* approach for the **AcerMC** we decided that using separate *training* runs with large numbers of weighted events to obtain the optimised grids and maximum weight estimates are preferable. In case a user wants to produce data sets for non-defaults setting, this can easily be done by configuring the switches in the acermc.card (see Section 6.2).

[16]These can also be used for a series of other settings, see Section 7.4 for details

[17]*Small* being a somewhat relative quantifier, since the size of an representative sample should depend on the phase space dimension, i.e. the number of particles in the final state; with e.g. 4 particles in the final state, 10^5 events can still be considered a relatively small statistics.

7 How to use the package

There are two main steering input files: **run.card** and **acermc.card** which share a common format for both executables. The **run.card** (see App. B.1) provides switches for modifying: generated process, number of events, parton density functions, predefined option for hadronisation/fragmentation in the *supervising generator*, random number, etc.. The **acermc.card** (see App. B.2) provides switches for modifying more specialised settings for the **AcerMC** library itself. Once the user decides on a setup for the generated process, only **run.card** is very likely to be modified for the job submission. In case the user decides to use the external TAUOLA or PHOTOS libraries by selecting the appropriate switches in the **run.card** there are two additional files **tauola.card**(see App. B.3) and **photos.card** (see App. B.4) which steer the performance of these two libraries. All input files are read by AcerMC executables through the CERNLIB FFREAD routines, some commands given in the input files (e.g. LIST entry, see Appendix B) are internal FFREAD commands which should be disregarded by the user.

The same executables can also be used for running standard PYTHIA 6.2 and HERWIG 6.5 processes. The example how to require such process is provided as well, in demo_hw.f and demo_py.f respectively. If the user requires that the **AcerMC** library is not used, the ttH production will be generated with demo_py.exe and HERWIG 6.5 implementation of the Zbb production will be generated with demo_hw.exe. In this case only the run.card file will be read, so in case the user requires different processes and/or settings of the supervising generators the user has to implement her/his steering there or create another xxx.card file, together with the corresponding code added to the demo_xx.f.

7.1 Steering switches of the overall run

The overall run is controlled by the switches read from the run.card file (see also App. B.1). Some of these general switches are also passed to the **AcerMC** library.

- CMS : Sets the centre-of-mass energy in GeV.
- ACER : Specifies if the internal **AcerMC** process will be used
  ACER=0 - use process from PYTHIA/HERWIG
  ACER=1 - use internal **AcerMC** process
  ACER=2 - use internal **AcerMC** process & dump events into a record
  ACER=3 - use internal **AcerMC** process & read events from a record
- PROCESS : Sets process id
- HAD : Sets predefined option for QCD ISR/FSR and hadronisation
  HAD=0 - only hard process
  HAD=1 - only ISR (works for PYTHIA interface only)
  HAD=2 - only ISR and FSR
  HAD=3 - full treatment
  HAD=4 - only FSR
  HAD=5 - only FSR and hadronisation
- LHAPDF NSET
  Sets the value of the LHAPDF/LHAGLUE NSET parton density function choice.
- RSEED : Choose the random seed for (pseudo-)random generator initialisation
- TAUOLA : Specifies if the TAUOLA library will be used for tau decays
  TAUOLA=0 - use internal PYTHIA/HERWIG mechanisms for τ-lepton decays
  TAUOLA=1 - use the TAUOLA library for τ-lepton decays.
- PHOTOS : Specifies if the PHOTOS routines will be used for final state photon radiation
  PHOTOS=0 - use internal PYTHIA/HERWIG mechanisms for FSR photon radiation
  PHOTOS=1 - use the PHOTOS mechanisms for FSR photon radiation.
- NEVENT : Required number of generated events

7.2 Steering switches of the AcerMC processes

The **AcerMC** processes are controlled by values set in a simple arrays specified in acermc_src/include/AcerMC.inc:

```
C CROSS-TALK PARAMETERS
DOUBLE PRECISION ACSET
```

```
INTEGER IACPROC
COMMON/ACPAR1/ACSET(200),IACPROC(200)
C PARTICLE PROPERTIES
DOUBLE PRECISION ACCHG,ACMAS,ACCKM
COMMON/ACPAR2/ACCHG(50,4),ACMAS(50,4),ACCKM(4,4)
C ROUTINE I/O
INTEGER LACSTD,LACIO
COMMON/ACPAR3/LACSTD,LACIO
```

The IACPROC array activates the process IPROC=PROCESS (read from run.card file) by setting IACPROC(IPROC)=1.

The list of currently implemented processes in **AcerMC** can be found in Table **11**. When running in the generation mode with ACER=0 full list of processes implemented in either PYTHIA 6.2 or HERWIG 6.5 can be activated, however the mechanism for passing information about process id to either of these generators has to be coded by user individually in demo_xx.f.

The main control switches reside in the array ACSET. The COMMON block ACPAR2 contains the particle charges, masses and decay widths as well as the CKM matrix using the PYTHIA convention. The values are filled by the interface routines to be equal to the PYTHIA/HERWIG internal values in order to preserve consistency within the generation stream. In case the user wants to change some of the particle properties this should be done through the native PYTHIA/HERWIG switches; **AcerMC** will copy them and use the new values.

The COMMON block ACPAR3 contains the two logical I/O unit numbers used by **AcerMC**. The LACSTD value determines the output unit of the **AcerMC** messages and the LACIO unit is used for reading/writing the **AcerMC** data files.

The main control switches which reside in the array ACSET (see also App. B.2):

- ACSET(1) : Sets the centre-of-mass energy in GeV.

- ACSET(2) : Scale of the hard process
  Choose the Q^2 scale for the active **AcerMC** process. The implemented values differ for various processes, the currently implemented settings are specified in Section 7.8.

- ACSET(3) : Fermion code
  The flavour of the final state fermions produced in $W^{\pm}$, $Z^0/\gamma^* \to f\bar{f}$ decays of **AcerMC** processes $3 \to 8$. The PYTHIA/PDG naming convention is used:
  ACSET(3)=1 - $W \to q\bar{q}$; $Z^0/\gamma^* \to q\bar{q}$
  ACSET(3)=4 - $W \to e\nu_e, \mu\nu_\mu$; $Z^0/\gamma^* \to e^+e^-, \mu^+\mu^-$
  ACSET(3)=11 - $W \to e\nu_e$; $Z^0/\gamma^* \to e^+e^-$
  ACSET(3)=12 - $Z^0 \to \nu_e\nu_e, \nu_\mu\nu_\mu, \nu_\tau\nu_\tau$,
  ACSET(3)=13 - $W \to \mu\nu_\mu$; $Z^0/\gamma^* \to \mu^+\mu^-$
  ACSET(3)=15 - $W \to \tau\nu_\tau$; $Z^0/\gamma^* \to \tau^+\tau^-$
  ACSET(3)=5 - $Z^0/\gamma^* \to b\bar{b}$
  At present the ACSET(3)=5 is implemented only for processes $7 \to 8$.

- ACSET(4) : Z^0/γ^* propagator
  Use full Z^0/γ^* propagator instead of the pure Z^0 propagator in matrix element calculation for the **AcerMC** processes $5 \to 8$ and 91. The switch is provided since in some of the analyses the γ^* contribution is of relevance in the selected mass windows; for the analyses selecting the mass window around the Z^0 peak this contribution can safely be neglected.
  ACSET(4)=0 - only Z^0 propagator.
  ACSET(4)=1 - full Z^0/γ^* propagator.

- ACSET(5) : m_{Z^0/γ^*} mass cut
  Cutoff value on the invariant mass m_{Z^0/γ^*} in GeV when ACSET(4)=1. Note that the provided data files exist only for values of ACSET(5)=2,5,10,15,30,60, 120,270,300 and 500 GeV which should satisfy most user requirements for the analyses foreseen at LHC. In case a different value is set the user has also to provide the user data files for the run.

- ACSET(7) : Sets the value of the LHAPDF/LHAGLUE NSET parton density function choice.

- ACSET(8): The implementation of $\alpha_s(Q^2)$
  Selects the implementation of $\alpha_s(Q^2)$ to be used in the matrix element calculation:
  ACSET(8)=0 - Use the $\alpha_s(Q^2)$ as provided by the supervising generator
  ACSET(8)=1 - Use the $\alpha_s(Q^2)$ (one loop) provided by the **AcerMC**; this option gives $\alpha_s(Q^2)$ values equal to the default PYTHIA implementation.
  ACSET(8)=2 - Use the $\alpha_s(Q^2)$ (three loop) provided by the **AcerMC**.

- ACSET(9): $\alpha_s(M_Z^2)$ value
  Sets the $\alpha_s(M_Z^2)$ value to be used in the $\alpha_s(Q^2)$ calculations in case the **AcerMC** native implementation (ACSET(8)=1) is used.
  ACSET(9)=-1 - The $\Lambda_{\bar{MS}}^{(nf=5)}$ value is taken from the LHAPDF for the selected parton density function set.
  ACSET(9)$>$0 - The provided value is taken.

- ACSET(10): The implementation of $\alpha_{\rm QED}(Q^2)$
  Selects the implementation of $\alpha_{\rm QED}(Q^2)$ to be used in the matrix element calculation:
  ACSET(10)=0 - Use the $\alpha_{\rm QED}(Q^2)$ as provided by the supervising generator.
  ACSET(10)=1 - Use the $\alpha_{QED}(Q^2)$ implemented in the **AcerMC**.

- ACSET(11): $\alpha_{\rm QED}(0)$ value
  Specifies the value of $\alpha_{\rm QED}(0)$ for **AcerMC** $\alpha_{\rm QED}(Q^2)$ calculation.
  ACSET(11)=-1 - The $\alpha_{\rm QED}(0)$ value is set to $\alpha_{\rm QED}(0) = 0.0072993$.
  ACSET(11)$>$0 - The provided value is taken.

- ACSET(12): Decay mode of the produced $t\bar{t}$ pair
  Sets the decay mode of the W boson pair from the $t\bar{t}$ final state in the **AcerMC** processes 1,2,4,7,8,9,92 and 93. For ACSET(12)$>$0 the combinatoric value of the $\sigma \times BR$ is recalculated and printed in the output. This switch was implemented since the supervising generators (PYTHIA/HERWIG) do not allow for forcing specific decays of the top quark pairs generated by external processes. This switch imposes a modification of the decay tables of the supervising generators on an event by event basis.
  ACSET(12)=0 - both W bosons decay according to PYTHIA/HERWIG switches.
  ACSET(12)=1 - $W_1 \to e\nu_e$ and $W_2 \to q\bar{q}$.
  ACSET(12)=2 - $W_1 \to \mu\nu_\mu$ and $W_2 \to q\bar{q}$.
  ACSET(12)=3 - $W_1 \to \tau\nu_\tau$ and $W_2 \to q\bar{q}$.
  ACSET(12)=4 - $W_1 \to e\nu_e, \mu\nu_\mu$ and $W_2 \to q\bar{q}$.
  ACSET(12)=5 - $W_1 \to e\nu_e, \mu\nu_\mu$ and $W_2 \to q\bar{q}$.
  The setting ACSET(12)=5 works for PROCESS=4 only and implies leptonic decay of the W-boson with the same charge as the one of the primary W boson produced in the hard process. Folowing configurations are posible:
  $q\bar{q} \to W^+ t\bar{t} \to (W^+ \to) L^+ \nu_L \quad (W_1^+ \to) l^+ \nu_l \ b \quad (W_2^- \to) q' \bar{q}' \ \bar{b},$
  or:
  $q\bar{q} \to W^- t\bar{t} \to (W^- \to) L^- \bar{\nu}_L \quad (W_1^+ \to) q' \bar{q}' \ b \quad (W_2^- \to) l^- \bar{\nu}_l \ \bar{b},$
  where $L^\pm$ is the lepton from the primary W decay (controlled by ACSET(3) switch) and $l^\pm$ is either an $e^\pm$ or $\mu^\pm$ as for ACSET(12)=4. Since the charge of the semi-leptonic decaying W is correlated with the charge of the primary W boson, the $\sigma \times BR$ is consequently a factor two smaller than the one for ACSET(12)=4.

- ACSET(13): Decay mode of the produced WW pair, works for PROC=11-14 only
  ACSET(13)=0 - both W bosons decay according to PYTHIA/HERWIG switches.
  ACSET(13)=1 - $W_1 \to e\nu_e$ and $W_2 \to q\bar{q}$.
  ACSET(13)=2 - $W_1 \to \mu\nu_\mu$ and $W_2 \to q\bar{q}$.
  ACSET(13)=3 - $W_1 \to \tau\nu_\tau$ and $W_2 \to q\bar{q}$.
  ACSET(13)=4 - $W_1 \to e\nu_e, \mu\nu_\mu$ and $W_2 \to q\bar{q}$.
  ACSET(13)=11 -both $W \to e\nu_e$.
  ACSET(13)=13 -both $W \to \mu\nu_\mu$.
  ACSET(13)=15 -both $W \to \tau\nu_\tau$.
  ACSET(13)=17 -both $W \to e\nu_e \mathrm{or} \mu\nu_\mu$.
  ACSET(13)=20 -both $W \to q\bar{q}$.

- ACSET(50): **AcerMC** training mode
  The switch controls the mode in which **AcerMC** is run:
  ACSET(50)=0 - production run, generate unweighted events.
  ACSET(50)=1 - perform multi-channel optimisation and output the user file with channel weights.
  ACSET(50)=2 - perform ac-VEGAS grid training and output the user file with trained ac-VEGAS grid.
  ACSET(50)=3 - perform ac-VEGAS grid training as in ACSET(50)=2 but do this by updating a provided grid.

- ACSET(51): Required number of generated events NEVENT
  In case the switch ACSET(50) is set to the non-zero value (i.e. in one of the training modes) the ACSET(51) entry is used and defines the number of (weighted) events that will be generated; this information is necessary for the learning algorithms to decide on steps in the learning sequence.

- ACSET(52): User data files
  Use the data files provided by user:
  ACSET(52)=0 - no, use native (default) **AcerMC** data files.
  ACSET(52)=1 - use the user's multi-channel optimisation and VEGAS grid files.
  ACSET(52)=2 - use the default multi-channel optimisation and user's VEGAS grid files.

ACSET(52)=3 - use the default multi-channel optimisation and VEGAS grid files; read the user maximal weight file.

- ACSET(53): Maximum weight search
  Mode for the maximum weight search needed for unweighting procedure:
  ACSET(53)=0 - no, use the provided files containing maximal weights.
  ACSET(53)=1 - use the provided files for max. weights and re-calculate the max. weights using the stored 100 events with the highest weight.
  ACSET(53)=2 - perform the search and give the new wtmax_xx_new.dat file; the switch is equivalent to generation of weighted events.

- ACSET(54): Maximum weight choice
  Use the α-cutoff maximal weight $\mathrm{wt}^{\alpha}_{\mathrm{max}}$ or the overall maximal weight $\mathrm{wt}_{\mathrm{max}}$ found in training (see Section 5.5.1 for the explanation on these two options).
  ACSET(54)=0 - use the $\mathrm{wt}^{\alpha}_{\mathrm{max}}$ weight.
  ACSET(54)=1 - use the $\mathrm{wt}_{\mathrm{max}}$ weight.

- ACSET(56): Naive QCD correction for width calculations
  Use the naive QCD multiplicative corrections in the resonant width calculations of the weak boson and top quark decay widths. It is recommended to use them except for some specific cases or studies.
  ACSET(56)=1 - use the naive QCD corrections
  ACSET(56)=0 - don't use naive QCD corrections

7.3 Steering TAUOLA and PHOTOS

It is highly recommended that in case TAUOLA is used for τ decays that PHOTOS is also activated and the setting PMODE is set at least to 2 in order to keep the τ-leptons radiating (see below), since it gives the necessary contribution to the expected τ branching ratios.

It should also be stressed that the actions of the interfaced TAUOLA and PHOTOS are meaningful only when the full hadronisation procedure of the event is selected (by setting HAD 3 in run.card).

The switches and settings specific to the TAUOLA library are set in the tauola.card:

- POLAR : Polarisation switch for tau decays
  POLAR=0 - switch polarisation off
  POLAR=1 - switch polarisation on

- RADCOR : Order(alpha) radiative corrections for tau decays
  RADCOR=0 - switch corrections off
  RADCOR=1 - switch corrections on

- PHOX : Radiative cutoff used in tau decays
  PHOX=0.01 - default value by TAUOLA authors

- DMODE : Tau and tau pair decay mode
  DMODE=0 - all decay modes allowed
  DMODE=1 - (LEPTON-LEPTON): only leptonic decay modes
  DMODE=2 - (HADRON-HADRON): only hadronic decay mode
  DMODE=3 - (LEPTON-HADRON): one tau decays into leptons and the other one into hadrons
  DMODE=4 - ($\tau \to \pi\nu$) : taus are restricted to decay to a pion and neutrino

- JAK1/JAK2 : Decay modes of taus according to charge, the list is taken from the TAUOLA output. The listing gives only τ^- modes, the τ^+ are charge conjugate, neutrinos are omitted. JAK1 : decay mode of τ^+, JAK2 : decay mode of τ^-:
  JAK1/2 = 1 - $\tau^- \to e-$
  JAK1/2 = 2 - $\tau^- \to \mu-$
  JAK1/2 = 3 - $\tau^- \to \pi^-$
  JAK1/2 = 4 - $\tau^- \to \pi^-, \pi^0$
  JAK1/2 = 5 - $\tau^- \to A_1^-$ (two subch)
  JAK1/2 = 6 - $\tau^- \to K^-$
  JAK1/2 = 7 - $\tau^- \to K^{*-}$ (two subch)
  JAK1/2 = 8 - $\tau^- \to 2\pi^-, \pi^0, \pi^+$
  JAK1/2 = 9 - $\tau^- \to 3\pi^0, \pi^-$
  JAK1/2 = 10 - $\tau^- \to 2\pi^-, \pi^+, 2\pi^0$
  JAK1/2 = 11 - $\tau^- \to 3\pi^-, 2\pi^+$
  JAK1/2 = 12 - $\tau^- \to 3\pi^-, 2\pi^+, \pi^0$
  JAK1/2 = 13 - $\tau^- \to 2\pi^-, \pi^+, 3\pi^0$
  JAK1/2 = 14 - $\tau^- \to K^-, \pi^-, K^+$
  JAK1/2 = 15 - $\tau^- \to K^0, \pi^-, \bar{K}^0$
  JAK1/2 = 16 - $\tau^- \to K^-, K^0, \pi^0$

JAK1/2 = 17 - $\tau^- \to \pi^0\pi^0K^-$
JAK1/2 = 18 - $\tau^- \to K^-\pi^-\pi^+$
JAK1/2 = 19 - $\tau^- \to \pi^-\bar{K^0}\pi^0$
JAK1/2 = 20 - $\tau^- \to \eta\pi^-\pi^0$
JAK1/2 = 21 - $\tau^- \to \pi^-\pi^0\gamma$
JAK1/2 = 22 - $\tau^- \to K^-K^0$

The switches and settings specific to the PHOTOS routines are set in the photos.card:

- PMODE : Radiation mode of photos:
  PMODE=1 - enable radiation of photons for leptons and hadrons
  PMODE=2 - enable radiation of photons for τ-leptons only
  PMODE=3 - enable radiation of photons for leptons only

- XPHCUT : Infrared cutoff for photon radiation:
  XPHCUT=0.01 - default value by PHOTOS authors

- ALPHA : Alpha(QED) value used in PHOTOS:
  ALPHA ¡ 0 - leave default (0.00729735039)

- INTERF : Photon interference weight switch
  INTERF = 1 - interference is switched on
  INTERF = 0 - interference is switched off

- ISEC : Double bremsstrahlung switch:
  ISEC=1 - double bremsstrahlung is switched on
  ISEC=0 - double bremsstrahlung is switched off

- IFTOP : Switch for $gg(qq) \to t\bar{t}$ process radiation:
  IFTOP=1 - the procedure is is switched on
  IFTOP=0 - the procedure is is switched off

Detailed information about TAUOLA and PHOTOS implemetations in the **AcerMC** setup can be found in Sections 7.11 and 7.12.

7.4 How to prepare data-files for the non-default setup

The following actions are possible, to recover better efficiency of the generator modules with the non-default settings:

- *The user wants to generate events using different parton density function sets and/or different coupling values (e.g.* **AcerMC** *third order $\alpha_s(Q^2)$ instead of the first order one)*:

  It should suffice to set the the switch ACSET(53)=1, which signals **AcerMC** to re-calculate the $\mathrm{wt}_{\max}$ and $\mathrm{wt}^{\alpha}_{\max}$ using the 100 events stored in the file acermc_dat/grids/vtmaxA_xxYYY.dat. The coupling and parton density functions values should not change significantly the process topology but affect foremost the overall scale of the event weights; thus, the stored hundred events should still remain the ones with the highest weights and the re-calculated approximate estimates of the highest weight should be accurate enough.

  In case the user is not confident in the obtained result, the new maximal weight estimation can be initiated by setting the switch ACSET(53)=2, which will result in generation of weighted events. The number of generated events is determined by the usual NEVENT in run.card. At the end of the run **AcerMC** will produce a file called wtmax_xx_new.dat, with xx specifying the process number. The user should then start the generation of unweighted events with the setting ACSET(52)=3 and linking(renaming) the new file to wtmax_xx_usr.dat, with xx denoting the process number (e.g. wtmax_01_usr.dat).

- *The user wants to generate events using different values of particle/boson masses or other significant changes of the parameters apart from the centre-of-mass energy and/or m_{Z^0/γ^*} cutoff value for processes 5-8*:

  In this case the user should re-train the VEGAS grid since the process topology is assumed to undergo minor changes. This is done by setting the switch ACSET(50)=2 or ACSET(50)=3; in the first case **AcerMC** starts with an untrained grid and in the second one it starts modifying the existing grid provided for the process at the selected hard process scale. In general the second option should be preferable since the topology should still be close to the pre-trained one. **AcerMC** again produces weighted events and at the end of the run outputs a file grid_xx_new.veg. The number of generated events is determined by the usual NEVENT in run.card. As in the previous case, the user should re-name the file to grid_xx_usr.veg, re-set the switch to ACSET(50)=0 and repeat the maximal weight search procedure described above, by setting the switch ACSET(53)=2 etc.. When the maximum weight search is completed the user switch ACSET(52)=2, which will cause **AcerMC** to read the wtmax_xx_usr.dat as well as grid_xx_usr.veg files and produce unweighted events with the new setup.

- *The user wants to generate events at a different m_{Z^0/γ^*} cutoff value and/or different centre-of-mass energy $\sqrt{s}$*:

  When the user changes at least one of these two parameters the event topology is significantly changed as well as the contributions from different kinematic channels. The user should thus start with a new multi-channel optimisation by setting the mode switch ACSET(50)=1 and start an **AcerMC** run. The number of generated events is determined by the usual NEVENT in run.card. At the end of the run **AcerMC** will produce a file chanwt_xx_new.dat which should be renamed/linked to chanwt_xx_usr.dat. The user should then set the switch ACSET(52)=1 and first put ACSET(50)=2 and and repeat the VEGAS grid training as described above and consequently ACSET(50)=0 and ACSET(53)=2 to perform the maximum weight search. After obtaining all three user files the ACSET(53) should again be put back to ACSET(53)=0 and a normal run should be started; the switch ACSET(52)=1 will in this case force **AcerMC** to read all three user files and produce unweighted events.

At the first look procedure for listed action scenaria might seem a bit complex but should after a few trials and errors become a straightforward routine; it is expected that the vast majority of users would have to deal with at most the first scenario.

7.5 Details on the interface to PYTHIA 6.2

The **AcerMC** interface to Pythia 6.2 is implemented close to the new standard specified at the Les Houches workshop 2001 [4]. The full description of the standard can be found in the PYTHIA 6.2 manual ([1]). In addition to the UPINIT and UPEVNT routines the file acermc_src/interface/pythia_ac.f provides links between a list of **AcerMC** routines and the corresponding PYTHIA ones, as e.g. the (pseudo-)random number generator, α_s and $\alpha_{\rm QED}$ calculations as well as a series of routines that re-write the **AcerMC** event output to the required PYTHIA format. Using this strategy, the native **AcerMC** code is completely de-coupled from the linked hadronisation library (at the moment PYTHIA/HERWIG) and new interfaces can thus easily be added. The special **AcerMC** requirement is the call to the ACFINAL subroutine at the end of the run which signals the **AcerMC** to close the various I/O files and produce the final output. An example of the implementation of the PYTHIA/**AcerMC** interface can be found in the provided demo_py.f. The PYTHIA code is unmodified apart from making a small modification in PYINIT routine:

```
CALL UPINIT(1)
..parameter initalisation..
CALL UPINIT(2)
..process initialisation..
```

since the user-supplied processes in this new interface are not allowed to (re-)estimate maximal weights (as e.g. the native PYTHIA processes do). In the original code the call to UPINIT is set before the PYTHIA parameters and functions (e.g. PYALPS for $\alpha_s(Q^2)$ calculation) are initialised with the user settings[18].

An additional modification was added in the PYEVNT routine so that undecayed resonances from AcerMC (e.g. top quarks) are decayed by PYTHIA before the ARIADNE routines are called (when requested by the user).

The user can thus add the most recent PYTHIA library without other neccessary modifications but for the two lines of code in PYINIT routine as described above (the dummy routines UPINIT, UPEVNT, STRUCTM, STRUCTP and PDFSET however have to be removed from the code for an external process to work and to activate the LHAPDF/LHAGLUE interface).

By setting ACER=1 user decides to generate hard process from **AcerMC** library. Modeling of ISR/FSR shower, hadronisation and decays are generated by PYTHIA generator. All steering parameters, relevant for these steps of full event generation remain the same as in standard PYTHIA execution.

By setting ACER=0 user decides to generate standard PYTHIA process. The simple example how to generate $t\bar{t}$ and $t\bar{t}H$ production process within **AcerMC** framework is provided in demo_py.f.

The additional settings ACER=2 and ACER=3, which allow the user to produce event records according to the Les Houches standard are given in the Section 7.8.

7.6 Details on the interface to ARIADNE 4.1

The **AcerMC** interface to ARIADNE 4.1 [3] is done via the ARIADNE $\leftrightarrow$ PYTHIA interface provided in the ARIADNE distribution and the further PYTHIA $\leftrightarrow$ **AcerMC** interface as described in the previous section. This is necessary since ARIADNE, while providing the advanced colour-dipole model of initial/final state radiation, is still relying on PYTHIA for particle/resonance decays and hadronisation. Some modifications were made to the ARIADNE $\leftrightarrow$ PYTHIA interface in order to accomodate resonance (e.g. top-quark) decays in Pythia before the event is passed to ARIADNE for shower addition and to properly search for Drell-Yan type processes for hard processes not implemented in PYTHIA (but e.g. in **AcerMC**).

[18] This was however possible in the old PYTHIA 6.1 interface

7.7 Details on the interface to HERWIG 6.5

Interfacing the **AcerMC** to HERWIG 6.5 is almost identical with the PYTHIA implementation since the interfaced version now also complies with the Les Houches standard [4]. The interface routines are written in accordance with the Les Houches description; in the HERWIG 6.5 interface the UPINIT routine also has to be called in two steps in order to enable the user to change the HERWIG 6.5 default settings and get the correct re-evaluation of the maximal weight (as in the PYTHIA interface):

```
CALL UPINIT(1)
..user values..
CALL UPINIT(2)
```

The first UPINIT call is made from the original location in the HWIGUP routine and the second call to UPINIT is placed at the end of the HWUINC routine, after all the internal HERWIG settings have been modified. As in the PYTHIA implementation all the interface subroutines needed for communication between HERWIG and **AcerMC** are stored in acermc_src/interface/herwig_ac.f.

One minor additional modification of the original HERWIG 6.5 code was albeit necessary, namely the IMPLICIT NONE was commented out in the herwig6500.inc file; this was needed since the **AcerMC** code is written with the implicit IMPLICIT DOUBLE PRECISION(A-H,O-Z).

In principle the implemented changes should be very easy and transparent for the transfer into new HERWIG releases; an example of the use of **AcerMC**/HERWIG interface is provided in the file demo_hw.f.

By setting ACER=1 the user decides to generate a hard process from **AcerMC** library. Modeling of ISR/FSR shower, hadronisation and decays are generated by HERWIG generator. All steering parameters, relevant for these steps of full event generation remain valid as for the standard HERWIG execution.

By setting ACER=0 the user decides to generate standard HERWIG processes. The simple example how to generate $Zb\bar{b}$ production process within the **AcerMC** framework is provided in demo_hw.f.

The additional settings ACER=2 and ACER=3, which allow the user to produce event records according to the Les Houches standard are given in the Section 7.8.

The output logs of the run are produced in the directory prod, the acermc.out file containing the **AcerMC** specific information and the outputs pythia.out and/or herwig.out listing the outputs of the respective supervising generators. The information about the input values of the steering files is stored in run.out in order to facilitate the event generation 'book-keeping'. The sample outputs are given in Appendix C.

Specific information produced by TAUOLA and PHOTOS is stored in respective files tauola.out and photos.out. A point to stress is that in case the tau decay was restricted the hard process cross-section given in pythia.out,herwig.out or acermc.out should be multiplied by a branching ratio as detailed at the end of tauola.out, for example:

```
------< TAUOLA BRANCHING RATIO FOR TAU DECAYS >------

 THE TAU DECAYS ARE RESTRICTED TO A:

           LEPTON-HADRON  DECAY MODE

 THE PROCESS CROSS-SECTION MUST BE MULTIPLIED BY:

  -> A BRANCHING RATIO =   0.459303E+00 FOR TWO TAUS

 IN THE HARD PROCESS DECAY PRODUCTS!!!

------> TAUOLA BRANCHING RATIO FOR TAU DECAYS <------
```

7.8 Definition of the energy scale

A few different values of scale Q^2 used in the evolution of parton density functions as well as the running couplings $\alpha_s(Q^2)$ and $\alpha_{\rm QED}(Q^2)$ can be set by the switch ACSET(2) (remember that the factorisation and renormalisation scales are assumed to be equal in **AcerMC**). Note that the *correct* value of the scale to be used for certain processes is in principle not known; what was implemented in **AcerMC** are the most probable/usual choices on the market; in measurements the *best* value will have to be determined by data analysis.

- Processes $1, 2, 9, 10$:
  ACSET(2): (D=1)
  1 - $Q^2 = \hat{s}$
  2 - $Q^2 = \sum (p_T^{i\,2} + m_i^2)/4 \; = < m_T^2 >$
  3 - $Q^2 = \sum (p_T^{i\,2})/4 \; = < p_T^2 >$
  4 - $Q^2 = (m_t + m_H/2)^2, \quad m_H = 120$ GeV/c^2
- Processes $3 \rightarrow 4$:
  ACSET(2): (D=1)
  1 - $Q^2 = M_W^2$
  2 - $Q^2 = s^*_{q\bar{q}}$, where $q = b, t$
  3 - $Q^2 = M_W^2 + pT_W^2$
  4 - $Q^2 = 0.5 \cdot (s^*_W + s^*_{q\bar{q}}) + (p_T^W)^2$, where $q = b, t$
- Processes $5 \rightarrow 8$:
  ACSET(2): (D=1)
  1 - $Q^2 = M_Z^2$
  2 - $Q^2 = s^*_{q\bar{q}}$, where $q = b, t$
  3 - $Q^2 = s^*_Z$
  4 - $Q^2 = pT_Z^2 + s^*_{q\bar{q}})/2$, where $q = b, t$
- Processes $11 \rightarrow 14$:
  ACSET(2): (D=1)
  1 - $Q^2 = (2 \cdot m_t^2)$
  2 - $Q^2 = \sum (p_T^{i\,2} + m_i^2)/4$
  3 - $Q^2 = \sum (p_T^{i\,2})/2$
  4 - $Q^2 = \hat{s}$
- Processes $15 \rightarrow 16$:
  ACSET(2): (D=1)
  1 - $Q^2 = \hat{s}$
  2 - $Q^2 = \sum (p_T^{i\,2} + m_i^2)/4 \; = < m_T^2 >$
  3 - $Q^2 = \sum (p_T^{i\,2})/4 \; = < p_T^2 >$
  4 - $Q^2 = (m_t + m_H/2)^2, \quad m_H = 120$ GeV/c^2

- Processes 91:
  ACSET(2): (D=1)
  1 - $Q^2 = \hat{s}$
  2 - $Q^2 = (\sum(p_T^{i\,2}) + M_Z^2)/2$
  3 - $Q^2 = \sum(p_T^{i\,2})/2$
  4 - $Q^2 = M_Z^2$
- Processes 92 $\rightarrow$ 93:
  ACSET(2): (D=1)
  1 - $Q^2 = (2m_t)^2$ 2 - $Q^2 = \sum(p_T^{i\,2} + m_i^2)/2$
  3 - $Q^2 = \sum(p_T^{i\,2})/2$
  4 - $Q^2 = \hat{s}$
- Processes 94:
  ACSET(2): (D=1)
  1 - $Q^2 = \hat{s}$
  2 - $Q^2 = (\sum(p_T^{i\,2}) + M_W^2)/2$
  3 - $Q^2 = \sum(p_T^{i\,2})/2$
  4 - $Q^2 = M_W^2$
- Processes 95:
  ACSET(2): (D=1)
  1 - $Q^2 = (2 \cdot m_t^2)$
  2 - $Q^2 = \sum(p_T^{i\,2} + m_i^2)/2$
  3 - $Q^2 = \sum(p_T^{i\,2})/2$
  4 - $Q^2 = \hat{s}$

there are the settings implemented so far.

7.9 Installation procedure

The installation requires availability of the CERNLIB fortran library.

- Ungzip and untar distribution file.
- In the main directory type make demo_py, make demo_hw or make demo_ar. It will compile demo_py.f, demo_hw.f or demo_ar.fand produce the executables demo_py.e, demo_hw.e or demo_ar.e depending on the selected option. The first-time call will also build and install all the required libraries; this will not be repeated when the demo_xx.f are subsequently changed.
- To execute the programs type make run_py,make run_hw or make run_ar.
  The scripts will change directory to prod and create respective links to data directories there. The execution will also be performed there. All input files should be accessible/routed from directory prod, the output files will also be produced in that directory.

7.10 Storing and reading events using the Les Houches accord

When setting the ACER=2 switch the **AcerMC 1.4** is instructed to dump the events generated by the **AcerMC** library in combination with PYTHIA or HERWIG into a pair of output files using the Les Houches format [4]. The files are AcerMC_pXXX_rYYYYYYY.inparm and AcerMC_pXXX_rYYYYYYY.events, with XXX denoting the AcerMC process ID while YY.. is currently the random seed used in **AcerMC**, to provide a unique identifier in the absence of other choices (run number & similar).

The *.inparm file is basically an information header, containing the relevant run parameters in accordance with the Les Houches standard [4], e.g. the cross-section of the process and the number of dumped/stored events; it is written in a 'human-readable' format in order to provide the user with the necessary information. The *.events file on the other hand contains the actual record of the hard process events produced by the **AcerMC**. The event dump is obtained by compiling and running the provided programs demo_xx.e as detailed in Section 7.9.

Using the ACER=3 switch the thus produced event record can subsequently be read back either into the **AcerMC 1.4** generator to be processed further by PYTHIA or HERWIG, i.e. the ISR/FSR and hadronisation can be added.

The same event records can also be read into the stand-alone PYTHIA or HERWIG generators, by re-directing the original(dummy) UPINIT routine to the INITACERMC routine and the original UPEVNT routine to USEACERMC routine (In the **AcerMC 1.4** setup this is done automatically).

The files containing the respective routines are named initacermc.f and useacermc.f and are provided in the leshouches directory of the **AcerMC 1.4** distribution. The reading of a specified set of generated events is achieved by copying the *.inparm and *.events files into the running directory and linking/copying the desired *.inparm file to a file named inparmAcerMC.dat; there is no need to modify the name of the *.events file since its name is already stored in the corresponding *.inparm file.

7.11 Interface of TAUOLA to PYTHIA and HERWIG

- Interface to PYTHIA :

The choice of TAUOLA library in PYTHIA sets the τ-s to be stable by setting MDCJ(15,1)=0, thus leaving them to be treated by TAUOLA procedures. The TAUOLA library is called via the TAUOLA_HEPEVT(IMODE) routine, where IMODE=-1,0,1 represent the initialisation, operation and finalisation respectively. In the operation mode the call to TAUOLA_HEPEVT(0) is made after the PYEVNT and PYHEPC(1) calls which generate the full event and translate and fill it into the HEPEVT common block. After TAUOLA has finished a subsequent call PYHEPC(2) is made to translate the new event back into the PYTHIA internal structure and a subsequent call to PYEXEC is made so that any undecayed particles (e.g. the π^0) are decayed. It was decided against using the provided PYTHIA interface via the PYTAUD routine since this routine is called for each occurrence of a τ-lepton separately so no (complex) polarisation options can be included without substantial changes to PYTHIA routines.

- Interface to HERWIG :

The present version 6.5 of HERWIG contrary to PYTHIA does internal tracking of the polarisations of the τ-leptons for the more complex built-in processes (e.g. Higgs decays); contrariwise nothing can be done for the external hard processes passed to HERWIG for hadronisation. In order to preserve this feature the HWDTAU routine has been modified to perform the TAUOLA calls only for internal HERWIG processes which provide the τ polarisation information; other τ-lepton decays are again executed via a call to TAUOLA_HEPEVT. Since HERWIG is using the HEPEVT common block already as the internal event record no translation of events is needed. In order to also keep other HERWIG parameters current (e.g. IDHW of the particle) a call to a new routine HWHEPC is provided; this new routine also corrects the vertex positions of the τ decay products and sets the status codes of new particles to the HERWIG recognisable values. Subsequently a call to HWDHAD is made in order to decay any undecayed particles.

7.12 Interface of PHOTOS to PYTHIA and HERWIG

- Interface to PYTHIA :

Choosing PHOTOS to provide the QED final state radiation sets the parameter PARJ(90)= $2 \cdot 10^4$ order to prevent PYTHIA to radiate photons off leptons, thus inducing double counting. The parameter is representing the threshold in GeV below which leptons do not radiate. Pythia does not contain the routines to handle radiation of hadrons however some caution is necessary due to some exceptions/specific decays (e.g. $\pi^0 \to e^+e^-\gamma$) which are generally recognised by PHOTOS itself. The initialisation, execution and finalisation are done by calls to the new PHOTOS_HEPEVT(IMODE), IMODE=-1,0,1 subroutine, constructed for this purpose by the AcerMC authors. Since PHOTOS operates on the HEPEVT record the calls to PYHEPC routine are again necessary.

- Interface to HERWIG :

The present version 6.5 of HERWIG does itself not provide the final state QED radiation of any kind, thus the inclusion of PHOTOS is simple and possibly also rather necessary. The call sequence is the same as in PYTHIA , with no need for event record conversion.

7.13 Details of the TAUOLA implementation

The TAUOLA library is built from the latest distribution source with the *Cleo* setup option. The native random generator was replaced with a link to the random generator of the linked event generator (PYTHIA or HERWIG) in the same manner as done for AcerMC , which decreases the number of random seeds which need to be initialised.

The TAUOLA_HEPEVT routine is a modification of the original TAUOLA routine provided in the file tauface_jetset.f in the TAUOLA distribution. The modifications were restricted to allow for the 'overloaded' use of the HEPEVT record by

PYTHIA and HERWIG , where respective mother and daughter pointers (which should match) sometimes point to different particles (e.g. hard process copies and similar). The original TAUOLA routine already worked when interfaced to PYTHIA , albeit requiring the non-default setting of MSTP(128)=1, whereby all the (sometimes useful) links to the hard record were lost. The modified version also works with the default PYTHIA MSTP(128)=0 setting.

Additional modifications were made in order to use the parameters set in tauola.card and the call to INIMAS routine was replaced by a call to TAU_INIMAS, which is a copy of the former but sets the particle masses to the values of PYTHIA or HERWIG defaults.

A routine TAUBRS that defines the special decay modes (e.g. LEPTON-HADRON, where one tau decays hadronically and the other one leptonically) was written; it operates by modifying the internal GAMPRT array on an event by event basis. A related routine TAUBR_PRINT prints the value of the branching ratio into the tauola.out file.No changes to the native tauola code was necessary (and therefore not made).

7.14 Details of the PHOTOS implementation

The PHOTOS is also built from the latest distribution source. As in TAUOLA The native random generator was replaced with a link to the random generator of the linked event generator (PYTHIA or HERWIG) in the same manner as done for AcerMC .

The PHOTOS_HEPEVT routine and the subsequently called PHOTOS_HEPEVT_MAKE are newly written routines that inspects the HEPEVT record for charged particles and finds their highest 'mother' particle, which is consequently passed to photos routines as the starting point. Photos itself then walks down the branches and performs radiation where possible. A bookkeeping of the starting points is made in PHOTOS_HEPEVT in order to prevent multiple invocations of radiation on the same particle.

Original photos code had to be modified due to the 'overloaded' HEPEVT record, since its requirements for matching mother-daughter pointers were too strict for either PYTHIA (with external processes and/or MSTP(128)=0 setting) or HERWIG . The modification was limited to PHOTOS_MAKE and PHOBOS routines. In addition the tracking of IDHW array was added to PHOTOS_MAKE to accommodate the HERWIG event record. A further modification was however necessary in the PHOIN routine since in HERWIG the entry JMOHEP(2,I) is not empty but filled with colour flow information, which in turn inhibited PHOTOS radiation off participating particles[19]. No modifications of the core (physics) PHOTOS code was made.

[19] PHOTOS expects the non-zero second 'mother' JMOHEP(2,I) entry only for $gg(qq) \to t\bar{t}$ process, which is treated by a set of dedicated routines; this in turn clashes with HERWIG 'overloaded' JMOHEP(2,I) entry.

8 Outlook and conclusions

In this paper we presented the **AcerMC** Monte Carlo Event Generator, based on the library of the matrix-element-based generators and interfaces to the universal event generators PYTHIA 6.2 and HERWIG 6.5. The interfaces are based on the standard proposed in [4].

The presented library fulfills the following goals:

- It gives a possibility to generate the few Standard Model background processes which were recognised as very dangerous for the searches for the *New Physics* at LHC, and generation of which was either unavailable or not straightforward so far.
- Although the hard process event is generated with matrix-element-based generator, the provided interface allows to complete event generation with initial and final state radiation, multiple interaction, hadronisation, fragmentation and decays, using implementation of either PYTHIA 6.2 or HERWIG 6.5.
- These interfaces can be also used for studying systematic differences between PYTHIA 6.2 or HERWIG 6.5 predictions for the underlying QCD processes.

The complete list of the native **AcerMC** processes implemented so far is: $gg, q\bar{q} \to t\bar{t}b\bar{b}$; $q\bar{q} \to W(\to \ell\nu)b\bar{b}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell)b\bar{b}$; $q\bar{q} \to W(\to \ell\nu)t\bar{t}$; $gg, q\bar{q} \to Z/\gamma^*(\to \ell\ell, \nu\nu, b\bar{b})t\bar{t}$; $gg, q\bar{q} \to (Z/W/\gamma^* \to)t\bar{t}b\bar{b}$; $gg, q\bar{q} \to t\bar{t}t\bar{t}$; $gg, q\bar{q} \to (t\bar{t} \to)f\bar{f}bf\bar{f}\bar{b}$; $gg, q\bar{q} \to (WWbb \to)f\bar{f}f\bar{f}b\bar{b}$. We plan to extend this not too exhaustive, but very much demanded list of processes, in the near future.

Several improvements of the existing Monte Carlo algorithms/programs have been developed in the process of this work. Let us make short list of the most interesting ones: (1) The use of the adapted Kajantie-Byckling enables one to automatise and modularise the phase space generation of n-particle final states. (2) The additions and extensions to the available (multi-channel) phase space algorithms (e.g. Breit-Wigner function with s-dependent width, mass-threshold effects) lead to substantial improvement of the unweighting efficiency; Figs. **8**, **10** and **11** illustrate the improvements achieved in the generation efficiency. (3) The power of the multi-channel optimisation was enhanced by using the modified ac-VEGAS package. We believe that the modification in the VEGAS code represents a very powerful extension of this package; (4)the colour flow information has been obtained after some modification of MADGRAPH package.

Having all these different production processes implemented in the consistent framework, which can be also directly used for generating standard processes available in either PYTHIA 6.2 or HERWIG 6.5 Monte Carlo, represents very convenient environment for several phenomenological studies dedicated to the LHC physics. Such frame was not available to our knowledge so far. We hope that it can serve as an interesting example or even a framework. This way some tools for discussing the ambiguities due to QCD effects are collected, however the necessary discussion for the appropriate uncertainties is still not exhausted. Nevertheless some discussions using this tool can be already found in [11], [12], [13], [17].

Acknowledgments

We would like to thank Daniel Froidevaux, Torbjorn Sjostrand, Bryan Webber and Alessandro Ballestrero, Zbigniew Was and Svjetlana Fajfer for several very valuable discussions. We would like also to thank all our colleagues from ATLAS Collaboration who were the first and very enthusiastic users of the preliminary versions of this package. We both very warmly acknowledge the support from the CERN PH division.

References

[1] T. Sjöstrand *et al.*, Computer Phys. Commun. **135** (2001) 238. eprint hep-ph/0108264, LU-TP 01-21, August 2001.

[2] G. Marchesini et al., Comp. Phys. Commun. **67** (1992) 465; G. Corcella et al., JHEP **0101** (2001) 010,

[3] L. Lönnblad, Computer Phys. Commun. **71** (1992) 15.
Manual for ARIADNE version 4 available with the distributed ARIADNE code

[4] E. Boos at al., *Generic user process interface for event generators*, hep-ph/0109068.

[5] E. Barberio and Z. Was, Comp. Phys. Commun. **79** (1994) 291.

[6] S. Jadach, J. H. Kuhn, Z. Was, Comput. Phys. Commun. **64** (1990) 275; M. Jezabek, Z. Was, S. Jadach, J. H. Kuhn, Comput. Phys. Commun. **70** (1992) 69; R. Decker, S. Jadach, J. H. Kuhn, Z. Was, Comput. Phys. Commun. **76** (1993) 361.

[7] H. Murayama, I. Watanabe and K. Hagiwara, *HELAS: helicity amplitude subroutines for Feynman diagrams evaluation*, preprint KEK-91-11 (1992).

[8] G.P. Lepage, J. Comput. Phys. **27** (1978) 192.

[9] T. Stelzer and W. F. Long, Comput. Phys. Commun. **81** (1994) 357.

[10] B. Kersevan and E. Richter-Was, Comput. Phys. Commun. **149** (2003) 142.

[11] B. Kersevan and E. Richter-Was, ATLAS Physics Note, ATL-PHYS-2001-020 (2001).

[12] B. Kersevan and E. Richter-Was, ATLAS Physics Note, ATL-PHYS-2001-021 (2001).

[13] B. Kersevan and E. Richter-Was, ATLAS Physics Note, ATL-PHYS-2001-022 (2001).

[14] A. Blondel, A. Clark and F. Mazzucato, ATLAS Physics Note, ATL-PHYS-2002-029 (2002).

[15] ATLAS Collaboration, *ATLAS Detector and Physics Performance TDR*, CERN-LHCC/99-15 (1999).

[16] CMS Collaboration, Technical Proposal, report CERN/LHCC/94-38 (1994).

[17] B. Kersevan and E. Richter-Was, *What is the $Wb\bar{b}$ background at LHC?...*, ATLAS Physics Note, ATL-PHYS-2003-018 (2001).

[18] E. Richter-Was and M. Sapinski, ATLAS Physics Note, ATL-PHYS-98-132 (1998), Acta Phys. Polon. **B30** (1999) 1001.

[19] M. L. Mangano, Nucl. Phys. **B405** (1993) 536.

[20] G. Marchesini et al., Comp. Phys. Commun. **67** (1992) 465.

[21] M. L. Mangano, M. Moretti and R. Pittau,*WBBGEN: program for the generation of $Wb\bar{b}$ + multijet final states in hadronic collisions*, *URL:mlm.home.cern.ch/mlm/wbb/wbb.html*.

[22] J.Campbell and R. K. Ellis, Phys. Rev. **D62** (2000) 114012.

[23] Proceedings of the Large Hadron Collider Workshop, Aachen, 1990, edited by G. Jarlskog and D. Rein, CERN 90-10/ECFA 90-133.

[24] B. van Eijk and R. Kleiss, in [23], page 184.

[25] B. van Eijk et al., EUROJET Monte Carlo, Nucl. Phys. **B292** (1987) 1.

[26] M. Della Negra et al., in [23], page 509.

[27] T. Sjostrand, Comp. Phys Comm. **82** (1994) 74.

[28] T. Sjostrand et al.,*High energy physics generation with PYTHIA 6.1*, Comp. Phys. Commun. **135** (2001) 238. eprint hep-ph/0010017, LU-TP 00-30, October 2000.

[29] J. Gunion, Phys. Rev. Lett. **72** (1994) 199.

[30] T. Pierzchala et al., Acta Phys. Polon. **B32** (2001) 1277.

[31] H. L. Lai et al, Eur. Phys. J. **C12** (2000) 375-392.

[32] A. Pukhov et al., *CompHEP - a package for evaluation of Feynman diagrams and integration over multi-particle phase space*, hep-ph/9908288 (1999).

[33] W. Beenakker et al., *Higgs radiation off top quarks at the Tevatron and at the LHC* Phys. Rev. Lett. **87** (2001) 201805. preprint DESY-01-077, hep-ph/0107081.

[34] E. Richter-Was, D. Froidevaux and L. Poggioli, ATLAS Physics Note ATL-PHYS-98-131 (1998).

[35] D. Froidevaux and E. Richter-Was, ATLAS Physics Note, ATL-PHYS-94-043 (1994), Z. Phys. **C67** (1995) 213.

[36] N. Krauer and D. Zeppenfeld, Phys.Rev. **D65** (2002) 014021.

[37] E. Richter-Was, ATLAS Physics Note, ATL-PHYS-2000-024 (2000), Acta Phys. Polon. **31** (2000) 1931.

[38] J. Hilgart, R. Kleiss, F. Le Dibider, Comp. Phys. Comm. **75** (1993) 191.

[39] F.A. Berends, R. Kleiss, Comput. Phys. Commun. **85** (1996) 11.

[40] F. A. Berends, C. G. Papadopoulos and R. Pittau, Comp. Phys. Commun. **136** (2001) 148.

[41] R. Kleiss and R. Pittau, Comput. Phys. Commun. **83** (1994) 141.

[42] Documentation and code available from:
http://durpdg.dur.ac.uk/lhapdf/index.html

[43] S. Jadach, E. Richter-Was, B.F.L. Ward and Z Was, Comput. Phys. Commun. **70** (1992) 305.

[44] T. Ohl, *Electroweak Gauge Bosons at Future Electron-Positron Colliders*, preprint hep-ph/9911437

[45] S. Jadach, Comput. Phys. Commun. **130** (2000) 244.

[46] H.-U. Bengtsson, Comput. Phys. Commun. **31** (1984) 323.

[47] M.L. Mangano, Nucl. Phys. **B309** (1988) 461.

[48] F. Caravaglios, M.L. Mangano, M. Moretti, R. Pittau, Nucl. Phys. **B539** (1999) no.1-2, 215.

[49] K. Odagiri, JHEP **9810:006**, (1998).

[50] S. Jadach, e-print:**physics/9906056**, (1999).

[51] M. Skrzypek, Z. Was, Comput. Phys. Commun. **125** (2000) 8.

[52] W.J. Stirling, R. Kleiss, and S.D. Ellis, Comput. Phys. Commun. **40** (1986) 359.

[53] P.D. Draggiotis, A. van Hameren and R. Kleiss, Phys.Lett. **B483** (2000) 124.

[54] A. van Hameren and C. G. Papadopoulos, Eur.Phys.J.**C25** (2002), 563.

[55] E. Byckling and K. Kajantie, Nucl. Phys. **B9** (1969) 568.
E. Byckling and K. Kajantie, "*Particle Kinematics*", Wiley & Sons. , London (1973) 328p.

[56] P. Nyborg, H.S. Song *et al.*, Phys. Rev. **140**, B914, (1965).

[57] Everett, C J; Cashwell, Edmond D, **LA-9721-MS** (1983) 150p.

[58] P. Lichard, Acta Phys.Slov. **49** (1999) 215.

[59] B. C. Carlson, **math.CA/9409227**.

[60] P. Appell and J. Kampé de Fériet, "*Fonctions hypergéométriques et hypersphériques: polynomes d'Hermite*", Gauthier-Villars, Paris, 1926.

[61] Daniel Zwillinger, editor, "*CRC Standard Mathematical Tables and Formulae*", CRC Press, 30th Edition, (2000).
code derived from http://www.csit.fsu.edu/∼burkardt/f_src/quadrule/quadrule.f90

[62] W.H. Press *et al.*, "*Numerical Recipes in Fortran 77*", Cambridge Univ. Press, second edition, (1992).

[63] J.H. Field , preprint UGVA-DPNC (1997) 2-170.

[64] H. Burkhardt and B. Pietrzyk, Phys. Lett. **B513** (2001) 46.

[65] W. J. Marciano, Phys. Rev. **D29** (1984), 580.

A Feynman Diagrams

The 38+7 Feynman diagrams contributing to the $gg, q\bar{q} \to t\bar{t}b\bar{b}$ production. Only four flavours are included for incoming quarks. Contribution of the incoming b-quarks could be excluded from the calculations thanks to very high suppression induced by either the parton density functions and/or CKM matrix elements.

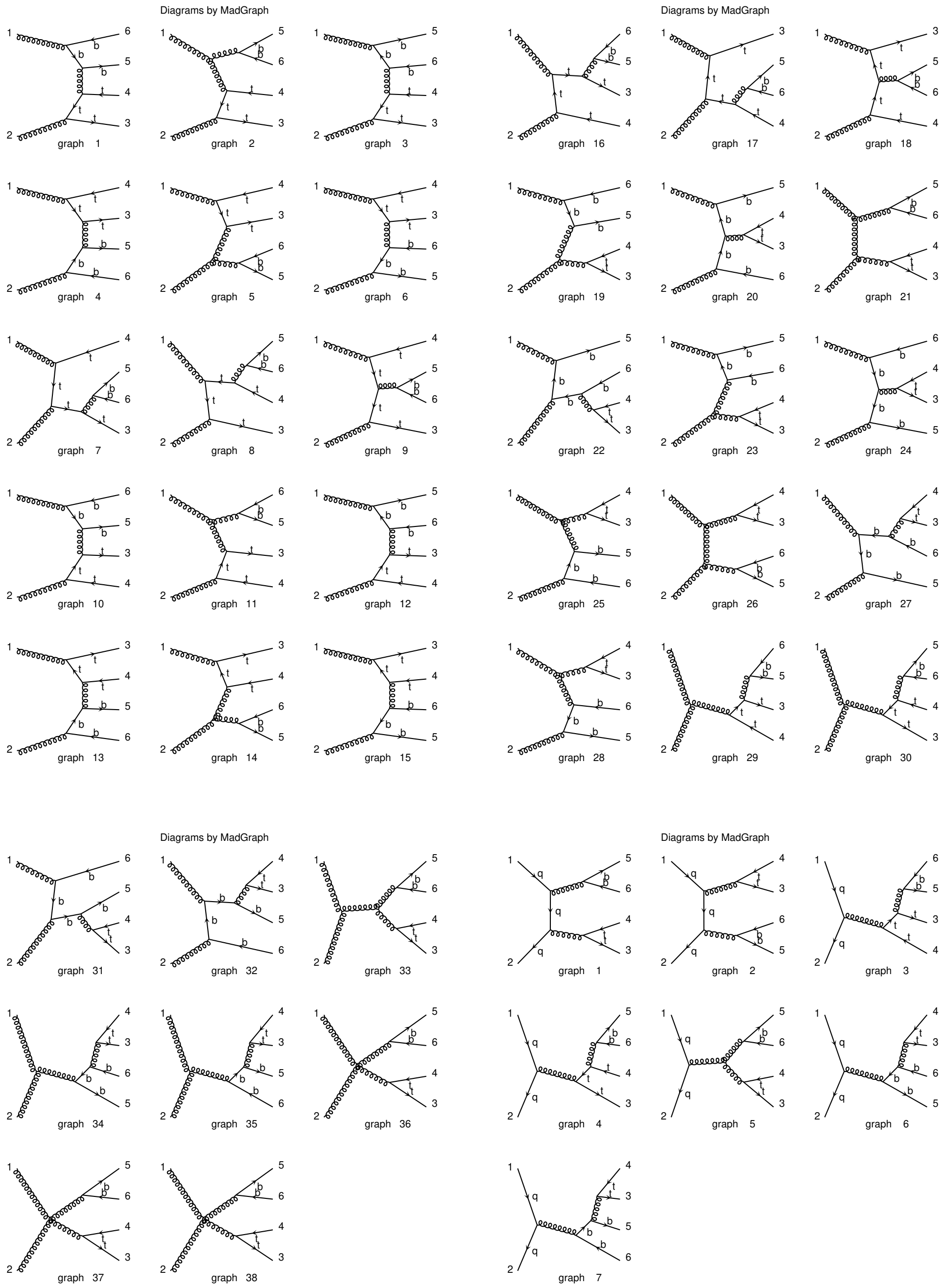


Fig. 19: *The Feynman diagrams for the processes $gg, q\bar{q} \to t\bar{t}b\bar{b}$.*

The Feynman diagrams contributing to the $q\bar{q} \to W(\to \ell\nu)b\bar{b}$ and $q\bar{q} \to W(\to \ell\nu)t\bar{t}$ matrix element are just two t-channel diagrams with fermion exchange and double conversion into an off-shell W boson and a virtual gluon; the W boson subsequently decays leptonically into $\ell\nu$ and the gluon splits into a $b\bar{b}$ pair or $t\bar{t}$ pair respectively. (c.f. Figure **20**).

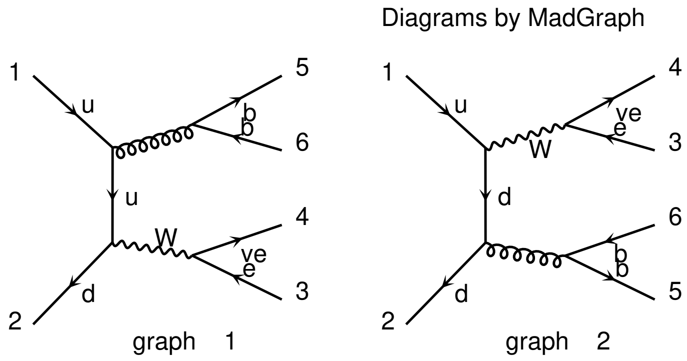


Fig. 20: *The Feynman diagrams for the process $q\bar{q} \to Wb\bar{b} \to e^{+}\nu_e\ b\bar{b}$. The same set is used for $q\bar{q} \to Wb\bar{b} \to e^{+}\nu_e\ t\bar{t}$ process, with b-quarks replaced by top-quarks.*

The Feynman diagrams contributing to the $gg, q\bar{q} \to Z/\gamma^*(f\bar{f}, \nu\nu)b\bar{b}$ production are shown in Figure **21**. The dominant contribution comes from the (2) and (6) configurations for the processes with gg initial state and the double conversion configuration (2),(4) for the ones with $q\bar{q}$ initial state. The same set of Feynman diagrams is used for the $gg, q\bar{q} \to Z/\gamma^*(f\bar{f}, \nu\nu)t\bar{t}$ process, with b-quarks being replaced by the top-quarks. If the $Z/\gamma^*(\to b\bar{b})$ decay mode is simulated, it represents only subset of the EW production of $t\bar{t}b\bar{b}$ final state.

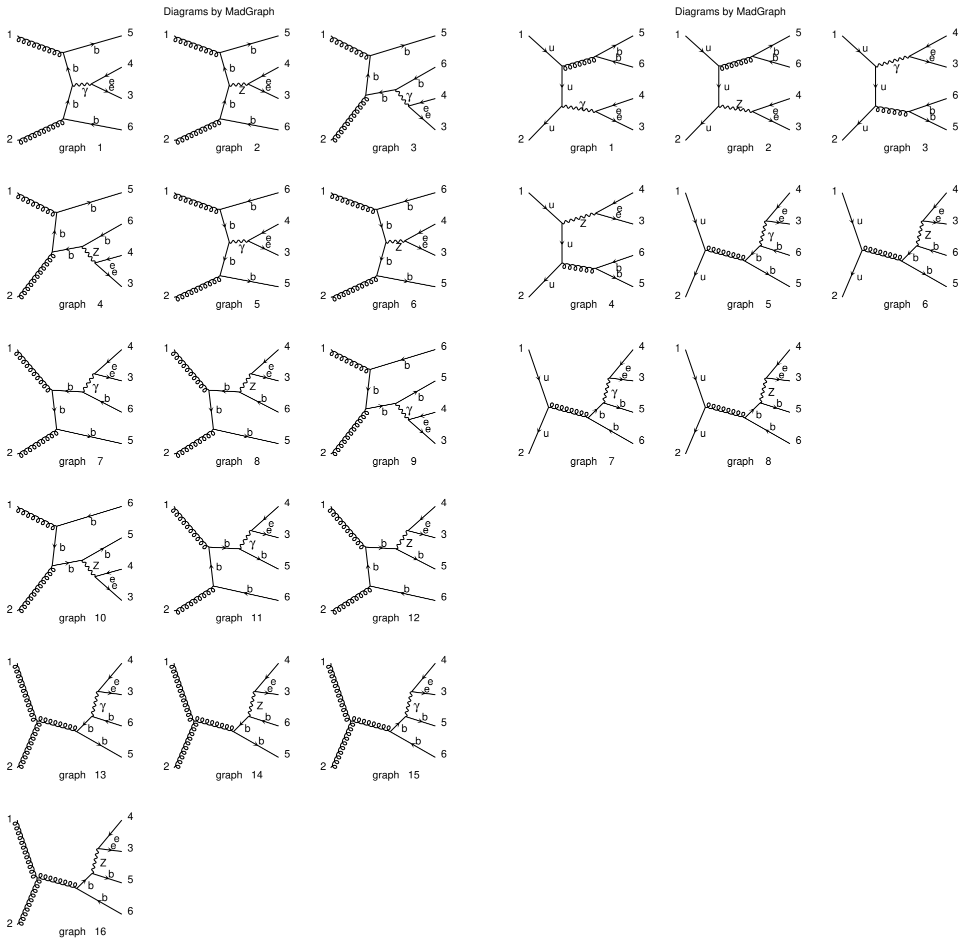
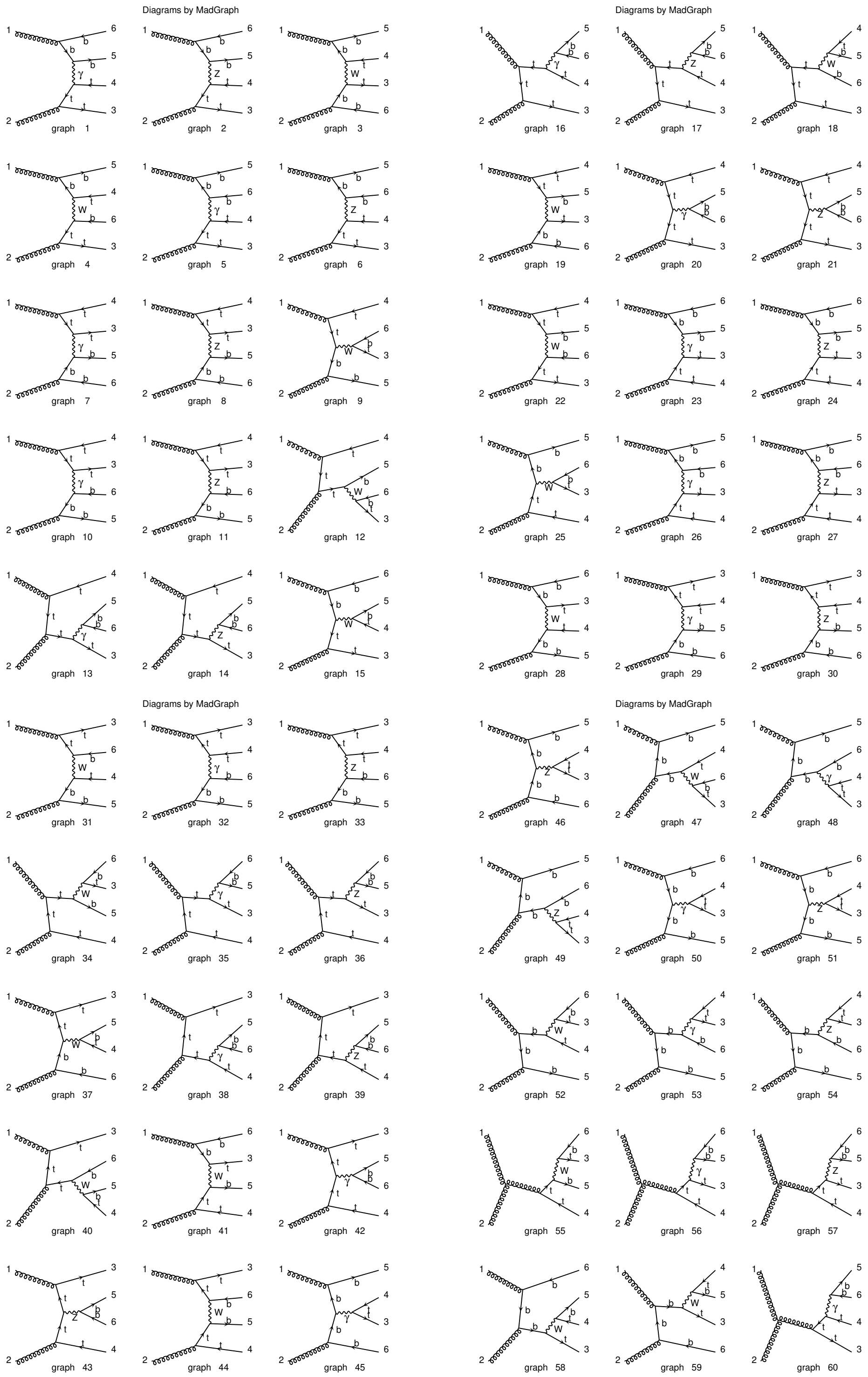


Fig. 21: *The Feynman diagrams for the processes $gg, q\bar{q} \to Z/\gamma^* b\bar{b} \to eeb\bar{b}$.*

The complete set of th the Feynman diagrams contributing to the full electro-weak $gg, q\bar{q} \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$ production mediated by exchange of the $Z/W/\gamma^*$ bosons is shown in Fig. **22**



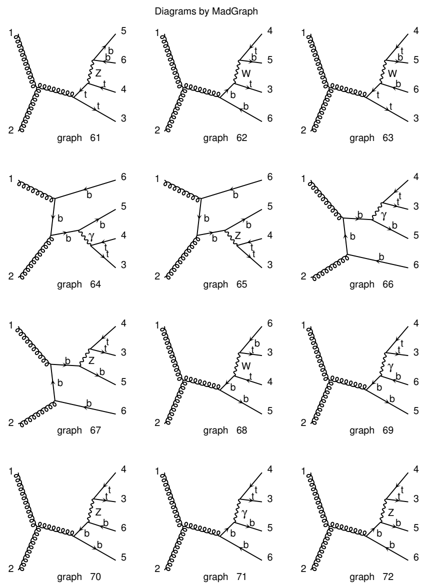


Fig. 22: *The Feynman diagrams for the processes* $gg \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$.

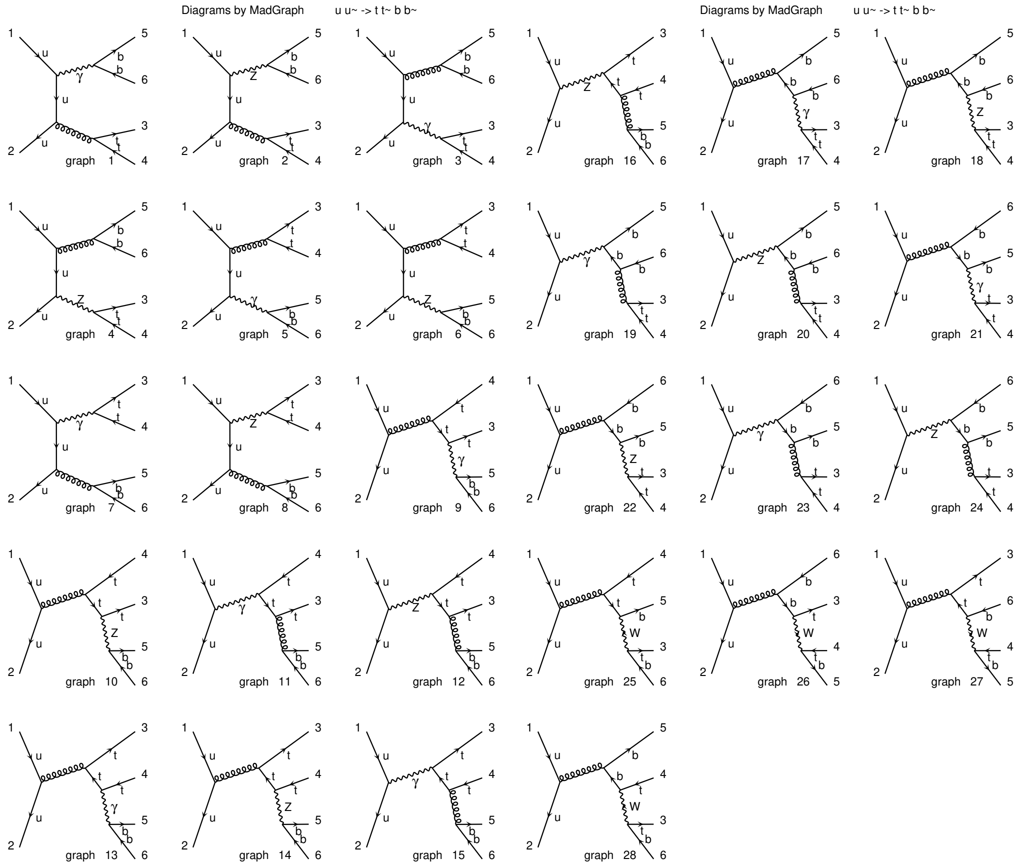


Fig. 23: *The Feynman diagrams for the processes* $q\bar{q} \to (Z/W/\gamma^* \to) b\bar{b}t\bar{t}$.

The set of the Feynman diagrams contributing to the $gg, q\bar{q} \to t\bar{t}$ production implemented with different approaches: resonant only $2 \to 6$ process $gg, q\bar{q} \to (t\bar{t} \to) f\bar{f}bf\bar{f}\bar{b}$, complete $2 \to 6$ process $gg, q\bar{q} \to (WbW\bar{b} \to) f\bar{f}bf\bar{f}\bar{b}$ and $2 \to 4$ process $gg \to WbW\bar{b} \to$ are shown in Figs. **24**- **27** .

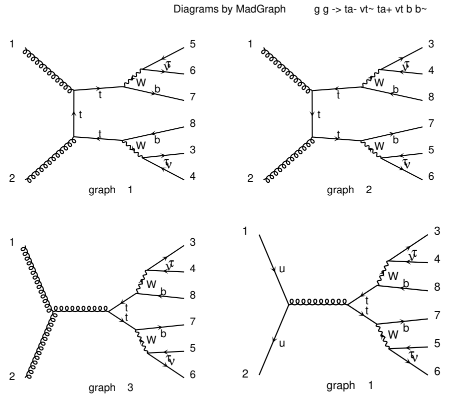


Fig. 24: *The Feynman diagrams for the processes $gg, q\bar{q} \to (t\bar{t} \to) f\bar{f}bf\bar{f}\bar{b}$.*

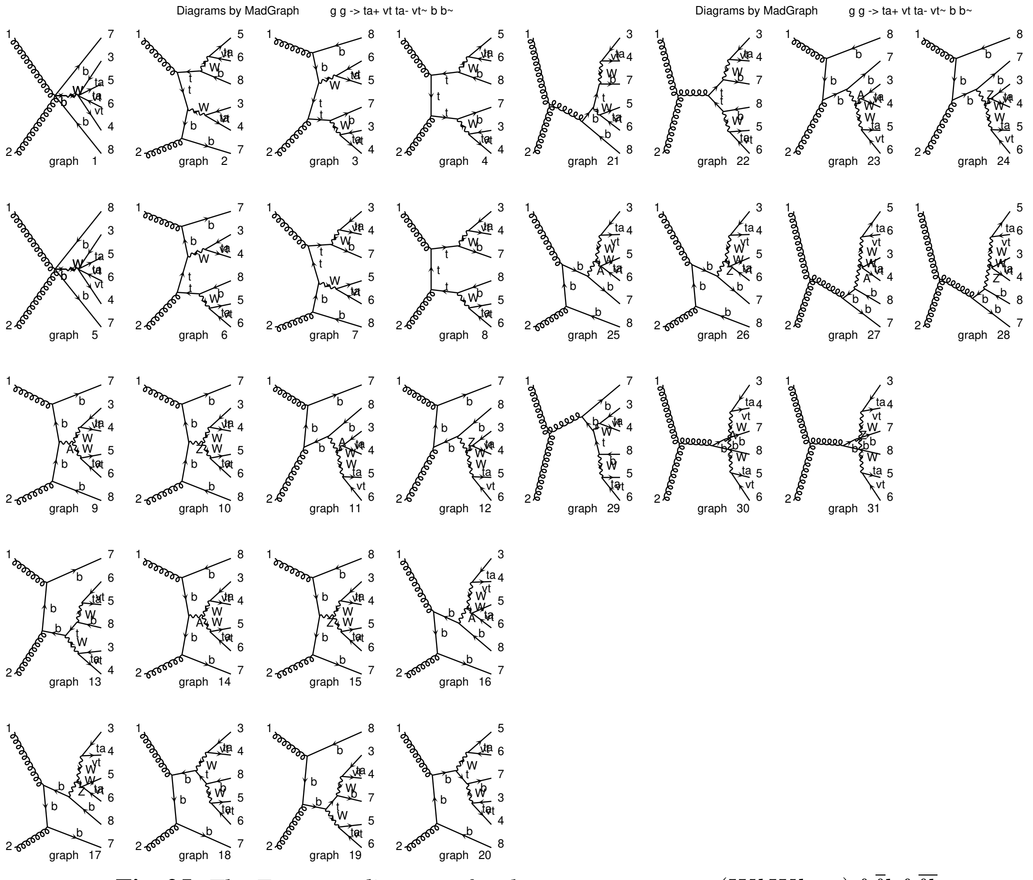


Fig. 25: *The Feynman diagrams for the processes $gg \to (WbWb \to) f\bar{f}bf\bar{f}\bar{b}$.*

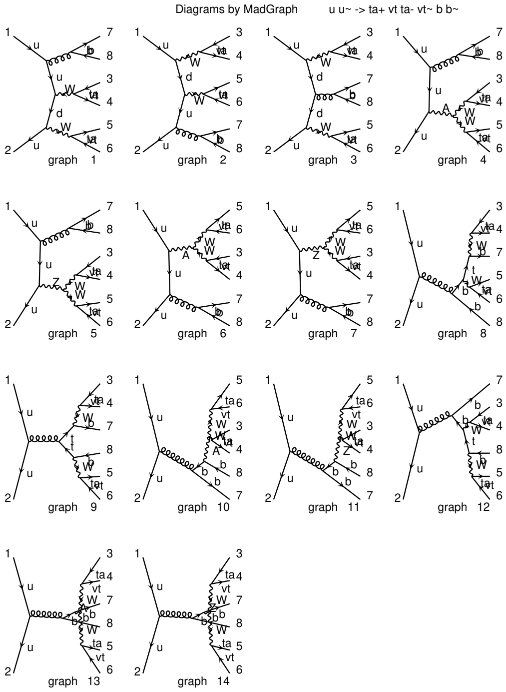


Fig. 26: *The Feynman diagrams for the processes* $q\bar{q} \rightarrow (WbWb \rightarrow) f\bar{f}bf\bar{f}\bar{b}$.

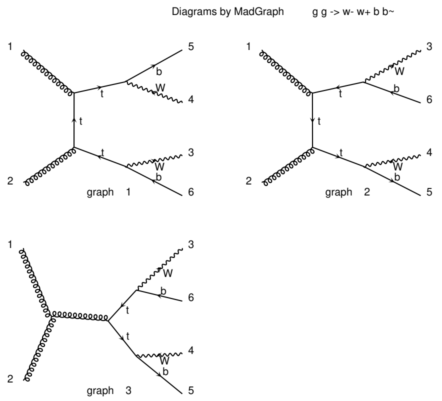


Fig. 27: *The Feynman diagrams for the processes* $gg \rightarrow WbWb$.

The 76+14 Feynman diagrams contributing to the $gg, q\bar{q} \rightarrow t\bar{t}t\bar{t}$ production are shown in Figures **28** and **29**. Only four flavours are included for incoming quarks. Contribution of the incoming b-quarks could be excluded from the calculations thanks to very high suppression induced by either the parton density functions and/or CKM matrix elements.

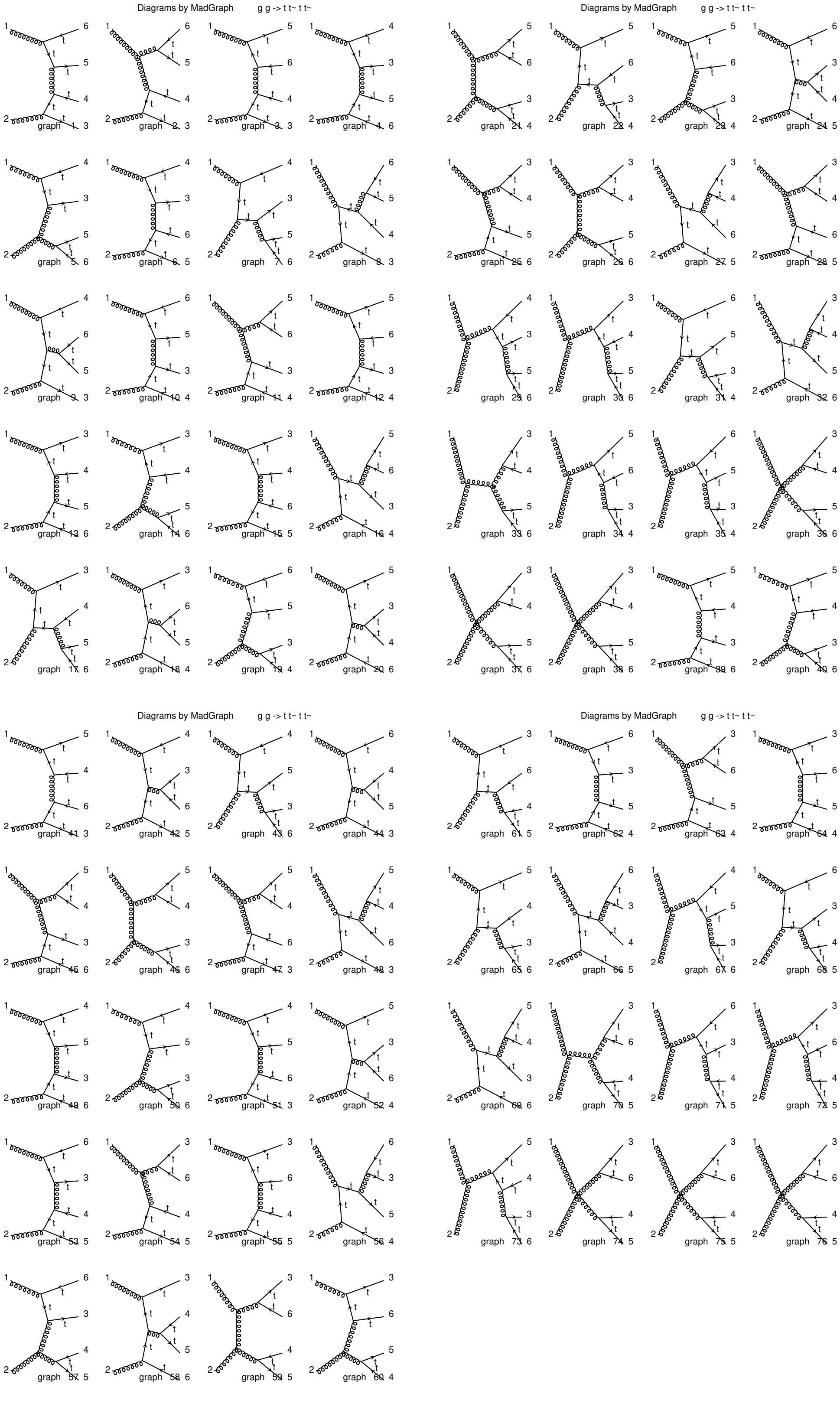


Fig. 28: *The Feynman diagrams for the processes $gg \to t\bar{t}t\bar{t}$.*

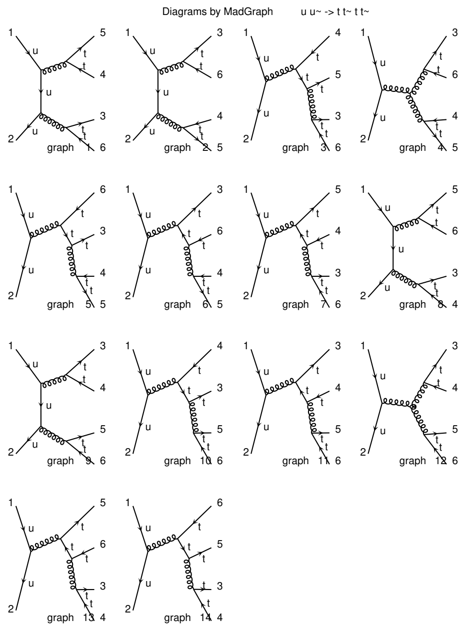


Fig. 29: *The Feynman diagrams for the processes $\bar{q} \to t\bar{t}t\bar{t}$.*

B Example input files

B.1 File run.card

```
C-------------------------------------------------------------------------------
C                  STEERING FILE FOR ACERMC (2.0) - ACERMC SETTINGS
C-------------------------------------------------------------------------------

C==== TURN ON FFKEY STEERING FILE (DEBUG)
LIST

C THE AcerMC EVENT SETTINGS ---------------------------------------------------

C==== SCALE/ACSET(2)
C Choose the Q^2 scale for the active AcerMC process.
C The implemented values differ for various processes, please look into the manual
C for details

ACSET2 1
C====

C==== FERMION/ACSET(3)
C The flavour of the final state leptons produced in W or Z decays of AcerMC
C processes.The Pythia/PDG naming convention is used:
C FERMION=1  - only hadronic decays
C FERMION=4  - electron and muon
C FERMION=5  - b-quark decay
C FERMION=11 - electron only
C FERMION=13 - muon only
C FERMION=15 - tau only
C FERMION=12 - neutrinos, all three flavours are generated and
C the cross-section is calculated accordingly.
C The setting FERMION=5 works only for processes 7 and 8
ACSET3 13
C====

C==== Z/GAMMA/ACSET(4)
C Use the full Z/gamma* propagator in AcerMC processes 5-8.
C ZGAMMA=0 - only Z propagator
C ZGAMMA=1 - full Z/gamma* propagator

ACSET4 1
C====

C==== Z/G CUT/ACSET(5)
C Cutoff value on the invariant mass m_Z/gamma* in GeV when ZGAMMA=1.
C Note that the provided data files exist only for values of
C ZGCUT=10,30,60,120,300 GeV which should satisfy most
C users. In case a different value is set the user has also to provide
C the user data files for the run.

ACSET5 60.0
C====

C THE AcerMC ADVANCED SWITCHES ------------------------------------------------

C==== ALPHA_S/ACSET(8)
C Use the alpha_s provided by the linked generator (Pythia/Herwig) or the
C one provided by AcerMC
herwig.pdf
C ALPHAS=0 - use the linked generator's alpha_s
C ALPHAS=1 - use the AcerMC's alpha_s (one loop calculation)
C ALPHAS=2 - use the AcerMC's alpha_s (three loop calculation)
ACSET8 0
C====

C==== ALPHA_S(M_Z^2)/ACSET(9)
C Specify the value of alpha_QCD(M_Z^2) for AcerMC alpha_s calculation
C ALPHASMZ=-1 - the value is taken from PDFLIB
C ALPHASMZ>0 - the provided value is taken
ACSET9 -1.
C====
```

```
C==== ALPHA_EM/ACSET(10)
C Use the alpha_QED provided by the linked generator (Pythia/Herwig) or the
C one provided by AcerMC
C ALPHAEM=0 - use the linked generator's alpha_QED
C ALPHAEM=1 - use the AcerMC's alpha_QED
ACSET10 0
C====

C==== ALPHA_EM(0)/ACSET(11)
C Specify the value of alpha_QED(0) for AcerMC alpha_QED calculation
C ALPHAEM0=-1 - the default AcerMC value is used
C ALPHAEM0>0 - the provided value is taken

ACSET11 -1.
C====

C==== TOP S-L/ACSET(12)
C Specify the decay mode of WW pair produced by external top decays in AcerMC
C processes 1,2,4,7,8,9,92 and 93:
C TOPDEC=0 - both W bosons decay according to Pythia/Herwig switches
C TOPDEC=1 - one W decays into electron + nu and the other one hadronically
C TOPDEC=2 - one W decays into muon + nu and the other one hadronically
C TOPDEC=3 - one W decays into tau + nu and the other one hadronically
C TOPDEC=4 - one W decays into el or mu + nu and the other one hadronically
C TOPDEC=5 - one W decays into el or mu + nu and the other one hadronically, the
C            W decaying leptonically has the same charge as the primary W; the decay
C            mode makes sense only for AcerMC processes 4!
C When TOPDEC>0 the output cross-section is ALREADY MULTIPLIED by the corresponding
C branching ratio(s)! (Courtesy of AcerMC authors)

ACSET12 0
C====

C==== BOSON PAIR DECAYS/ACSET(13)
C Specify the decay mode of boson pairs inside AcerMC processes:
C
C BOSDEC=0 - both bosons decay in all possible modes
C BOSDEC=1 - one boson decays into electron + nu and the other one hadronically
C BOSDEC=2 - one boson decays into muon + nu and the other one hadronically
C BOSDEC=3 - one boson decays into tau + nu and the other one hadronically
C BOSDEC=4 - one boson decays into el or mu + nu and the other one hadronically
C BOSDEC=11 - both bosons decay into el + nu
C BOSDEC=13 - both bosons decay into muon + nu
C BOSDEC=15 - both bosons decay into tau + nu
C BOSDEC=20 - both bosons decay hadronically
C The output cross-section is ALREADY MULTIPLIED by the corresponding
C branching ratio(s)!

ACSET13 13
C====

C THE AcerMC TRAINING SETUP AND UNWEIGHTING TREATMENT -------------------------

C==== MODE/ACSET(50)
C Specify the AcerMC training mode:
C MODE=0 - normal run, generate unweighted events
C MODE=1 - perform multi-channel optimisation.
C MODE=2 - perform VEGAS grid training.
C MODE=3 - perform VEGAS grid training as MODE=2 but does this by upd
C MODE=-1 - all flat
C MODE=-2 - only VEGAS flat
ating a provided grid

ACSET50 0
C====

C==== USER/ACSET(52)
C Use the data files provided by user
C USER=0 - no, use internal files
C USER=1 - use the user's multi-channel optimisation and VEGAS grid files
C USER=2 - use the default multi-channel optimisation and user's VEGAS grid files
C USER=3 - use the default multi-channel optimisation and VEGAS grid files, read the user
C           maximal weight file.
```

```
ACSET52 0
C====

C==== MAXFIND/ACSET(53)
C Search for the maximum weight needed for event unweighting
C MAXFIND=0 - no, use the provided file for max. weights
C MAXFIND=1 - use the provided file for max. weights, re-calculate the max. weights using
C             the stored 100 highest events
C MAXFIND=2 - perform the search and give the wtmax file, equivalent to generation of
C             weighted events

ACSET53 0
C====

C==== EPSILON/ACSET(54)
C Use the epsilon maximal weight or the overall maximal weight found in training (see the
C manual for the difference)
C EPSILON=0 - use the epsilon max. weight
C EPSILON=1 - use the overall maximal weight

ACSET54 0
C====

C==== NQCD/ACSET(56)
C Use the naive qcd correction for width calculations
C (see the manual for details)
C NQCD=1 - use the naive QCD corrections
C NQCD=0 - don't use naive QCD corrections

ACSET56 1
C====

C ------------------------------------------------------------------------------
END
```

B.2 File acermc.card

```
C------------------------------------------------------------------------------
C                 STEERING FILE FOR ACERMC (2.0) - ACERMC SETTINGS
C------------------------------------------------------------------------------

C==== TURN ON FFKEY STEERING FILE (DEBUG)
LIST

C THE AcerMC EVENT SETTINGS ---------------------------------------------------

C==== SCALE/ACSET(2)
C Choose the Q^2 scale for the active AcerMC process.
C The implemented values differ for various processes, please look into the manual
C for details

ACSET2 1
C====

C==== FERMION/ACSET(3)
C The flavour of the final state leptons produced in W or Z decays of AcerMC
C processes.The Pythia/PDG naming convention is used:
C FERMION=1  - only hadronic decays
C FERMION=4  - electron and muon
C FERMION=5  - b-quark decay
C FERMION=11 - electron only
C FERMION=13 - muon only
C FERMION=15 - tau only
C FERMION=12 - neutrinos, all three flavours are generated and
C the cross-section is calculated accordingly.
C The setting FERMION=5 works only for processes 7 and 8
ACSET3 13
C====

C==== Z/GAMMA/ACSET(4)
C Use the full Z/gamma* propagator in AcerMC processes 5-8.
C ZGAMMA=0 - only Z propagator
C ZGAMMA=1 - full Z/gamma* propagator
```

```
ACSET4 1
C====

C==== Z/G CUT/ACSET(5)
C Cutoff value on the invariant mass m_Z/gamma* in GeV when ZGAMMA=1.
C Note that the provided data files exist only for values of
C ZGCUT=10,30,60,120,300 GeV which should satisfy most
C users. In case a different value is set the user has also to provide
C the user data files for the run.

ACSET5 60.0
C====

C THE AcerMC ADVANCED SWITCHES ------------------------------------------------

C==== ALPHA_S/ACSET(8)
C Use the alpha_s provided by the linked generator (Pythia/Herwig) or the
C one provided by AcerMC
herwig.pdf
C ALPHAS=0 - use the linked generator's alpha_s
C ALPHAS=1 - use the AcerMC's alpha_s (one loop calculation)
C ALPHAS=2 - use the AcerMC's alpha_s (three loop calculation)
ACSET8 0
C====

C==== ALPHA_S(M_Z^2)/ACSET(9)
C Specify the value of alpha_QCD(M_Z^2) for AcerMC alpha_s calculation
C ALPHASMZ=-1 - the value is taken from PDFLIB
C ALPHASMZ>0 - the provided value is taken
ACSET9 -1.
C====

C==== ALPHA_EM/ACSET(10)
C Use the alpha_QED provided by the linked generator (Pythia/Herwig) or the
C one provided by AcerMC
C ALPHAEM=0 - use the linked generator's alpha_QED
C ALPHAEM=1 - use the AcerMC's alpha_QED
ACSET10 0
C====

C==== ALPHA_EM(0)/ACSET(11)
C Specify the value of alpha_QED(0) for AcerMC alpha_QED calculation
C ALPHAEM0=-1 - the default AcerMC value is used
C ALPHAEM0>0 - the provided value is taken

ACSET11 -1.
C====

C==== TOP S-L/ACSET(12)
C Specify the decay mode of WW pair produced by external top decays in AcerMC
C processes 1,2,4,7,8,9,92 and 93:
C TOPDEC=0 - both W bosons decay according to Pythia/Herwig switches
C TOPDEC=1 - one W decays into electron + nu and the other one hadronically
C TOPDEC=2 - one W decays into muon + nu and the other one hadronically
C TOPDEC=3 - one W decays into tau + nu and the other one hadronically
C TOPDEC=4 - one W decays into el or mu + nu and the other one hadronically
C TOPDEC=5 - one W decays into el or mu + nu and the other one hadronically, the
C            W decaying leptonically has the same charge as the primary W; the decay
C            mode makes sense only for AcerMC processes 4!
C When TOPDEC>0 the output cross-section is ALREADY MULTIPLIED by the corresponding
C branching ratio(s)! (Courtesy of AcerMC authors)

ACSET12 0
C====

C==== BOSON PAIR DECAYS/ACSET(13)
C Specify the decay mode of boson pairs inside AcerMC processes:
C
C BOSDEC=0 - both bosons decay in all possible modes
C BOSDEC=1 - one boson decays into electron + nu and the other one hadronically
C BOSDEC=2 - one boson decays into muon + nu and the other one hadronically
C BOSDEC=3 - one boson decays into tau + nu and the other one hadronically
C BOSDEC=4 - one boson decays into el or mu + nu and the other one hadronically
```

```
C BOSDEC=11 - both bosons decay into el + nu
C BOSDEC=13 - both bosons decay into muon + nu
C BOSDEC=15 - both bosons decay into tau + nu
C BOSDEC=20 - both bosons decay hadronically
C The output cross-section is ALREADY MULTIPLIED by the corresponding
C branching ratio(s)!

ACSET13 13
C====

C THE AcerMC TRAINING SETUP AND UNWEIGHTING TREATMENT -------------------------

C==== MODE/ACSET(50)
C Specify the AcerMC training mode:
C MODE=0 - normal run, generate unweighted events
C MODE=1 - perform multi-channel optimisation.
C MODE=2 - perform VEGAS grid training.
C MODE=3 - perform VEGAS grid training as MODE=2 but does this by upd
C MODE=-1 - all flat
C MODE=-2 - only VEGAS flat
ating a provided grid

ACSET50 0
C====

C==== USER/ACSET(52)
C Use the data files provided by user
C USER=0 - no, use internal files
C USER=1 - use the user’s multi-channel optimisation and VEGAS grid files
C USER=2 - use the default multi-channel optimisation and user’s VEGAS grid files
C USER=3 - use the default multi-channel optimisation and VEGAS grid files, read the user
C           maximal weight file.

ACSET52 0
C====

C==== MAXFIND/ACSET(53)
C Search for the maximum weight needed for event unweighting
C MAXFIND=0 - no, use the provided file for max. weights
C MAXFIND=1 - use the provided file for max. weights, re-calculate the max. weights using
C             the stored 100 highest events
C MAXFIND=2 - perform the search and give the wtmax file, equivalent to generation of
C             weighted events

ACSET53 0
C====

C==== EPSILON/ACSET(54)
C Use the epsilon maximal weight or the overall maximal weight found in training (see the
C manual for the difference)
C EPSILON=0 - use the epsilon max. weight
C EPSILON=1 - use the overall maximal weight

ACSET54 0
C====

C==== NQCD/ACSET(56)
C Use the naive qcd correction for width calculations
C (see the manual for details)
C NQCD=1 - use the naive QCD corrections
C NQCD=0 - don’t use naive QCD corrections

ACSET56 1
C====

C ----------------------------------------------------------------------------
END
```

B.3 File tauola.card

```
C----------------------------------------------------------------------------
C                  STEERING FILE FOR TAUOLA & ACERMC (1.4)
C----------------------------------------------------------------------------
```

```
C==== TURN ON FFKEY STEERING FILE (DEBUG)
LIST

C==== POLAR
C Polarisation switch for tau decays
C POLAR=0 - switch polarisation off
C POLAR=1 - switch polarisation on
POLAR 1
C====

C==== RADCOR
C Order(alpha) radiative corrections for tau decays
C RADCOR=0 - switch corrections off
C RADCOR=1 - switch corrections on

RADCOR 1
C====

C==== PHOX
C Radiative cutoff used in tau decays
C PHOX=0.01 - default value by TAUOLA authors

PHOX 0.01
C====

C==== DMODE
C Tau and tau pair decay mode
C DMODE=0 - all decay modes allowed
C DMODE=1 - (LEPTON-LEPTON): only leptonic  decay modes
C DMODE=2 - (HADRON-HADRON): only hadronic  decay modes
C DMODE=3 - (LEPTON-HADRON): one tau decays into leptons and the other one into hadrons
C DMODE=4 - (TAU->PI NU)   : taus are restricted to decay to a pion and neutrino

DMODE 2
C====

C==== JAK1/JAK2
C Decay modes of taus according to charge, the list is taken from TAUOLA output
C The listing gives only Tau- modes, the Tau+ are charge conjugate, neutrinos
C are omitted.
C JAK1/2 = 1   - TAU-  -->  E-
C JAK1/2 = 2   - TAU-  -->  MU-
C JAK1/2 = 3   - TAU-  -->  PI-
C JAK1/2 = 4   - TAU-  -->  PI-, PI0
C JAK1/2 = 5   - TAU-  -->  A1- (two subch)
C JAK1/2 = 6   - TAU-  -->  K-
C JAK1/2 = 7   - TAU-  -->  K*- (two subch)
C JAK1/2 = 8   - TAU-  -->  2PI-,  PI0,  PI+
C JAK1/2 = 9   - TAU-  -->  3PI0,        PI-
C JAK1/2 = 10  - TAU-  -->  2PI-,  PI+, 2PI0
C JAK1/2 = 11  - TAU-  -->  3PI-, 2PI+
C JAK1/2 = 12  - TAU-  -->  3PI-, 2PI+,  PI0
C JAK1/2 = 13  - TAU-  -->  2PI-,  PI+, 3PI0
C JAK1/2 = 14  - TAU-  -->  K-, PI-,  K+
C JAK1/2 = 15  - TAU-  -->  K0, PI-, K0B
C JAK1/2 = 16  - TAU-  -->  K-,  K0, PI0
C JAK1/2 = 17  - TAU-  -->  PI0  PI0   K-
C JAK1/2 = 18  - TAU-  -->  K-  PI-  PI+
C JAK1/2 = 19  - TAU-  -->  PI-  K0B  PI0
C JAK1/2 = 20  - TAU-  -->  ETA  PI-  PI0
C JAK1/2 = 21  - TAU-  -->  PI-  PI0  GAM
C JAK1/2 = 22  - TAU-  -->  K-  K0

C==== DECAY MODE OF TAU+
JAK1 13
C====

C==== DECAY MODE OF TAU-
JAK2 13
C====

C ----------------------------------------------------------------------------
END
```

B.4 File photos.card

```
C-----------------------------------------------------------------------------
C                  STEERING FILE FOR PHOTOS & ACERMC (1.4)
C-----------------------------------------------------------------------------

C==== TURN ON FFKEY STEERING FILE (DEBUG)
LIST

C==== PMODE
C Radiation mode of photos
C PMODE=1 - enable radiation of photons for leptons and hadrons
C PMODE=2 - enable radiation of photons for taus only
C PMODE=3 - enable radiation of photons for leptons only
PMODE 1

C====

C==== XPHCUT
c Infrared cutoff for photon radiation
C XPHCUT=0.01 - default value by PHOTOS authors
XPHCUT 0.01

C====

C==== ALPHA
C Alpha(QED) value
C ALPHA < 0 - leave default (0.00729735039)
ALPHA -1.
C====

C==== INTERF
C Photon interference weight switch
C INTERF = 1 - interference is switched on
C INTERF = 0 - interference is switched off
INTERF 1
C====

C==== ISEC
C Double bremsstrahlung switch
C ISEC=1 - double bremsstrahlung is switched on
C ISEC=0 - double bremsstrahlung is switched off
ISEC 1
C====

C==== IFTOP
C Switch for gg(qq)->tt~ process radiation
C IFTOP=1 - the procedure is is switched on
C IFTOP=0 - the procedure is is switched off

IFTOP 0
C====
```

C Example output files

C.1 File acermc.out

```
 ------------------------------------------------------------------------------
                                  ._
                                .j%3]:,
                              ~!%%%%%%% ,._.
                           _|xx%xxxx%%%%%+`
                            :~]%xxxx]xx%x_,+_x_%`
                     -__||x||xx]+]]]+]]]]x|xxx]`
                      -+%%xxxx]]]]+]]++]+]x]|>- .;..;.:_/`
                         -+x]]]]|+]+]+]++]++]]+|+|]+|]]+-
             ,.   .  ., |x]+]+||=++]+]=+++++++=|++]=+]|~-
             -|%x]]];x]]||=++++++++|];|++++++++++=+]=];    ..   ..  :..,;
               -/]||]+]|]++++++++||||==|:;:=|==++==+;|,   :;;. :;=;;===|`
                _|]|+>+++]+]]|+|||=|=;|;;:|;=:===|==;::.;;:;,;:;;;;;;;:-
                    -++]+++:+|x+::||=||:=:::::;;::::::::::;:;:;:;:;;;;=::
                  .|x+|+|]++:,-..:::|=;=:-.:.:-::.:.:.::-:::::::::;:-
        .,  .:||_   --||;:|:::.-.:.-:|;||::.:...-.-.-....::::-:::;::.
    __._;++;;;|=|;. -::::::::---:.--;;|==|:;:.:-:-:.-.-.:::.--:::;:;--
    -+++=+=======;==:::::-:.::...:.|+=;;===:-..:.:.:::::-...--:  -
    :|:-:|===;:;:;::::::--::.:-::::.|||===:::.:.:::::.-......--:: .
       -;|===;;:;;;;;::;::.:-:: -:=;;|+||=;==|=:::::::::...-.:..-.::..
        ---|;:,::::.::::=;=:=::  -++===|======:--:   ...-...-:-:.-
            ---;::::;;:--:===;;;:|-|+=+|+|==;:`    ...-....-..
            .::;;:::;:;;:--- -- --.|=||=+|=:=,       ..... .
           .;;=;;:-:::;:::.        -+;,|]| ;;=`
             - -:- --:;;:- -         - :|; -
                      -                 :`
                                         :
                                         :
                                         .

  40000L,                                            |0000i   j000&  .a00000L#0
  --?##0L        .aaaa aa      .aaaa;      aaaa, _aaa, -000A  _0001- _d0!`  -400
    d0 40,     _W0#V9N0#&     d0#V*N#0,   0##0LW0@4#@' 00j#; J0|01  d0'      40
   J01 -#W     #0'    ?#W   ##~    -#0;     j##9       00 4#|01|01  00
  _00yyyW0L   :0f      ^-   :000###00001     j#1        00 ?#0@`|01  #0
  ##!!!!!#0;  -0A       _  -0A              j#1        00  HH< |01  j0L       _
ad0La,  aj0Aa  4#Aaa_aj#0`   ?0Laa_aaa0L  aaJ0Laaa,   _a00aa  _aj0La  *0Aaa_aad
HHHRHl  HHHHH   `9##009!     `9NW00@!!`   HHHHRHHHl   :HHHHH  ?HHHRH   ?!##00P!`

 ------------------------------------------------------------------------------

       AcerMC 2.0 (April 2004),  B. P. Kersevan, E. Richter-Was

 ------------------------------------------------------------------------------

 ---------------------------< ACTIVATED PROCESSES >----------------------------

                      1)  g + g  ->  t t~ b b~ (MG)                ON
                      2)  q + q~ ->  t t~ b b~ (MG)                OFF
                      3)  q + q~ ->  (W->) l nu_l b b~ (MG)        OFF
                      4)  q + q~ ->  (W->) l nu_l t t~ (MG)        OFF
                      5)  g + g  ->  (Z0->) l l~ b b~ (MG)         OFF
                      6)  q + q~ ->  (Z0->) l l~ b b~ (MG)         OFF
                      7)  g + g  ->  (Z0->) f f~ t t~ (MG)         OFF
                      8)  q + q~ ->  (Z0->) f f~ t t~ (MG)         OFF
                      9)  g + g  -> (Z0/W/gamma->) t t~ b b~  (MG) OFF
                      10)  q + q~ -> (Z0/W/gamma->) t t~ b b~  (MG) OFF
                     11)  g + g  ->  t t~ off-shell (MG)           OFF
                     12)  q + q~ ->  t t~ off-shell (MG)           OFF
                     13)  g + g  ->  (W W b b~ ->)  (MG)           OFF
                     14)  q + q~ ->  (W W b b~ ->)  (MG)           OFF
                     15)  g + g  ->  t t~ t t~ (MG)                OFF
                     16)  q + q~ ->  t t~ t t~ (MG)                OFF
                     91)  q + q~ -> (Z0/gamma->) l l~ (MG)         OFF
                     92)  g + g  ->  t t~ (MG)                     OFF
                     93)  q + q~ ->  t t~ (MG)                     OFF
                     94)  q + q~ ->  (W->) l nu_l (MG)             OFF
                     95)  g + g  ->  W+ W- b b~ (MG)               OFF
```

```
----------------------------< ACERMC SETTINGS >----------------------------

                    C.M.S ENERGY  =       14000.00       [ACSET(1)]
                    SCALE CHOICE  =              1       [ACSET(2)]
                    ACERMC ALPHA_QCD =           0       [ACSET(8)]
                    ALPHA_QCD(M_Z)   =   -1.000000       [ACSET(9)]
                    ACERMC ALPHA_QED =           0       [ACSET(10)]
                    ALPHA_QED(0)     =   -1.000000       [ACSET(11)]
                    TOP->W S-L DECAY =           0       [ACSET(12)]
                    BOSON PAIR DECAY =          13       [ACSET(13)]

                    OPTIMIZATION   =             0       [ACSET(50)]
                    OPTIM. STEPS   =           100       [ACSET(51)]
                    USER FILES     =             0       [ACSET(52)]
                    MAX. SEARCH    =             0       [ACSET(53)]
                    EPSILON CUTOFF =             0       [ACSET(54)]
                    NAIVE QCD      =             1       [ACSET(56)]

---------------------------------------------------------------------------


           READ MAXIMUM WEIGHT(MB)  =   0.236905E-07
           READ EPSILON WEIGHT(MB)  =   0.108907E-07



         SET MAXIMUM WEIGHT(MB)  =   0.108907E-07
         SET WEIGHT CORRECTION   =   0.100000E+01



  ---------< FINALIZATION FOR PROCESS: 1 >---------

    --------------< WEIGHT SURVEY >--------------

    ------------< TOTAL STATISTICS >-------------

     CROSS-SECTION ESTIMATE =   0.411408E+01 PB
                       +/-   0.153340E+00 PB
          VARIANCE ESTIMATE =   0.235132E-01 PB^2
                       +/-   0.159711E-02 PB^2

     MAXIMUM WEIGHT  =   0.976277E-08
     NO.WEIGHTS NE 0 =            252
     NO.WEIGHTS EQ 0 =              0
     NO.WEIGHTS LT 0 =              0
     EFFICIENCY FOR ALL WEIGHTS     =  42.141 %
     EFFICIENCY FOR NONZERO WEIGHTS =  42.141 %

     NO.WEIGHTS ABOVE EPSILON-CUT  =           0

    --------------> WEIGHT SURVEY <--------------
```

C.2 File pythia.out

```
******************************************************************************
**                                                                          **
**                                                                          **
**              *......*                  Welcome to the Lund Monte Carlo!  **
**         *:::!!:::::::::::*                                               **
**      *::::::!!::::::::::::::*          PPP  Y   Y TTTTT H   H III   A    **
**    *::::::::!!::::::::::::::::*        P  P  Y Y    T   H   H  I   A A   **
**   *:::::::::!!:::::::::::::::::*       PPP    Y     T   HHHHH  I  AAAAA  **
**   *:::::::::!!:::::::::::::::::*       P      Y     T   H   H  I  A   A  **
**    *::::::::!!::::::::::::::::*!       P      Y     T   H   H III A   A  **
**      *::::::!!::::::::::::::* !!                                         **
**      !! *:::!!:::::::::::*    !!       This is PYTHIA version 6.214      **
**      !!     !* -><- *         !!       Last date of change: 22 Jan 2003  **
**      !!     !!                !!                                         **
**      !!     !!                !!       Now is  6 Feb 2003 at 18:39:46    **
**      !!                       !!                                         **
**      !!        lh             !!       Disclaimer: this program comes    **
**      !!                       !!       without any guarantees. Beware    **
**      !!                 hh    !!       of errors and use common sense    **
**      !!    ll                 !!       when interpreting results.        **
**      !!                       !!                                         **
**      !!                                Copyright T. Sjostrand (2003)     **
**                                                                          **
** An archive of program versions and documentation is found on the web:    **
** http://www.thep.lu.se/~torbjorn/Pythia.html                              **
**                                                                          **
** When you cite this program, currently the official reference is          **
** T. Sjostrand, P. Eden, C. Friberg, L. Lonnblad, G. Miu, S. Mrenna and    **
** E. Norrbin, Computer Physics Commun. 135 (2001) 238.                     **
** The large manual is                                                      **
** T. Sjostrand, L. Lonnblad and S. Mrenna, LU TP 01-21 [hep-ph/0108264].   **
** Also remember that the program, to a large extent, represents original   **
** physics research. Other publications of special relevance to your        **
** studies may therefore deserve separate mention.                          **
**                                                                          **
** Main author: Torbjorn Sjostrand; Department of Theoretical Physics 2,     **
**   Lund University, Solvegatan 14A, S-223 62 Lund, Sweden;                **
**   phone: + 46 - 46 - 222 48 16; e-mail: torbjorn@thep.lu.se              **
** Author: Leif Lonnblad; Department of Theoretical Physics 2,              **
**   Lund University, Solvegatan 14A, S-223 62 Lund, Sweden;                **
**   phone: + 46 - 46 - 222 77 80; e-mail: leif@thep.lu.se                  **
** Author: Stephen Mrenna; Computing Division, Simulations Group,           **
**   Fermi National Accelerator Laboratory, MS 234, Batavia, IL 60510, USA; **
**   phone: + 1 - 630 - 840 - 2556; e-mail: mrenna@fnal.gov                 **
** Author: Peter Skands; Department of Theoretical Physics 2,               **
**   Lund University, Solvegatan 14A, S-223 62 Lund, Sweden;                **
**   phone: + 46 - 46 - 222 31 92; e-mail: zeiler@thep.lu.se                **
**                                                                          **
**                                                                          **
******************************************************************************
******************************************************************************
1****************** PYINIT: initialization of PYTHIA routines *****************

 ==============================================================================
 I                                                                            I
 I         PYTHIA will be initialized for p+ on p+ user configuration         I
 I            with   7000.000 GeV on   7000.000 GeV beam energies             I
 I                                                                            I
 I           corresponding to  14000.000 GeV center-of-mass energy            I
 I                                                                            I
 ==============================================================================

 ******** PYMAXI: summary of differential cross-section maximum search ********

           ==========================================================
           I                                      I                 I
           I  ISUB  Subprocess name               I  Maximum value  I
           I                                      I                 I
           ==========================================================
           I                                      I                 I
           I    4   User process 601              I    1.9366E-08   I
           I                                      I                 I
           ==========================================================
```

```
********************* PYINIT: initialization completed *********************

         Event listing of user process at input (simplified)

  I IST     ID Mothers   Colours    p_x      p_y      p_z       E        m
  1 -1      21   0   0  501  503    0.000    0.000  332.784  332.784    0.000
  2 -1      21   0   0  502  504    0.000    0.000 -614.582  614.582    0.000
  3  1       6   1   2  501    0   81.702   20.957 -203.328  281.214  175.000
  4  1      -6   1   2    0  504   28.925   29.174 -310.224  358.541  175.000
  5  1       5   1   2  502    0  -52.937  -77.151   33.793   99.598    4.800
  6  1      -5   1   2    0  503  -57.690   27.020  197.960  208.013    4.800


                          Event listing (summary)

   I particle/jet KS     KF  orig    p_x      p_y      p_z       E        m

   1 !p+!         21    2212   0    0.000    0.000 7000.000 7000.000    0.938
   2 !p+!         21    2212   0    0.000    0.000-7000.000 7000.000    0.938
==============================================================================
   3 !g!          21      21   1   -0.290    0.974  332.784  332.785    0.000
   4 !g!          21      21   2   -0.177    0.335 -614.582  614.582    0.000
   5 !g!          21      21   3   -0.290    0.974  332.784  332.785    0.000
   6 !g!          21      21   4   -0.177    0.335 -614.582  614.582    0.000
   7 !t!          21       6   0   81.598   21.204 -203.329  281.203  175.000
   8 !tbar!       21      -6   0   28.808   29.427 -310.250  358.575  175.000
   9 !b!          21       5   0  -53.004  -76.938   33.870   99.495    4.800
  10 !bbar!       21      -5   0  -57.868   27.617  197.911  208.094    4.800
==============================================================================
  11 t         A   2       6   7   81.598   21.204 -203.329  281.203  175.000
  12 ud_0      V   1    2101   1   -0.058   -0.344 4727.872 4727.872    0.579
  13 tbar      A   2      -6   8   28.808   29.427 -310.250  358.575  175.000
  14 u         V   1       2   2    0.233   -0.275  -19.002   19.009    0.330
  15 b         A   2       5   9  -53.004  -76.938   33.870   99.495    4.800
  16 ud_1      V   1    2103   2   -0.056   -0.061-6366.410 6366.410    0.771
  17 bbar      A   2      -5  10  -57.868   27.617  197.911  208.094    4.800
  18 u         V   1       2   1    0.348   -0.630 1939.344 1939.344    0.330
==============================================================================
                  sum:  2.00          0.00     0.00     0.01 14000.00 14000.00

********* PYSTAT:  Statistics on Number of Events and Cross-sections *********

==============================================================================
I                                  I                            I            I
I            Subprocess            I      Number of points      I    Sigma   I
I                                  I                            I            I
I----------------------------------I----------------------------I    (mb)    I
I                                  I                            I            I
I N:o Type                         I    Generated         Tried I            I
I                                  I                            I            I
==============================================================================
I                                  I                            I            I
I   0 All included subprocesses    I         100            491 I  3.913E-09 I
I   4 User process 601             I         100            491 I  3.913E-09 I
I                                  I                            I            I
==============================================================================
```

C.3 File herwig.out

```
        HERWIG 6.500    16 Oct 2002

        Please reference:  G. Marchesini, B.R. Webber,
        G.Abbiendi, I.G.Knowles, M.H.Seymour & L.Stanco
        Computer Physics Communications 67 (1992) 465
                           and
        G.Corcella, I.G.Knowles, G.Marchesini, S.Moretti,
        K.Odagiri, P.Richardson, M.H.Seymour & B.R.Webber,
        JHEP 0101 (2001) 010

        INPUT CONDITIONS FOR THIS RUN

        BEAM 1 (P       ) MOM. =   7000.00
        BEAM 2 (P       ) MOM. =   7000.00
        PROCESS CODE (IPROC)   =    -601
        NUMBER OF FLAVOURS     =    6
        STRUCTURE FUNCTION SET =    8
        AZIM SPIN CORRELATIONS =    T
        AZIM SOFT CORRELATIONS =    T
        QCD LAMBDA (GEV)       =    0.1800
        DOWN     QUARK  MASS   =    0.3200
        UP       QUARK  MASS   =    0.3200
        STRANGE  QUARK  MASS   =    0.5000
        CHARMED  QUARK  MASS   =    1.5500
        BOTTOM   QUARK  MASS   =    4.8000
        TOP      QUARK  MASS   =  175.0000
        GLUON EFFECTIVE MASS   =    0.7500
        EXTRA SHOWER CUTOFF (Q)=    0.4800
        EXTRA SHOWER CUTOFF (G)=    0.1000
        PHOTON SHOWER CUTOFF   =    0.4000
        CLUSTER MASS PARAMETER =    3.3500
        SPACELIKE EVOLN CUTOFF =    2.5000
        INTRINSIC P-TRAN (RMS) =    0.0000
        DECAY SPIN CORRELATIONS=    T
        SUSY THREE BODY ME     =    T
        SUSY FOUR  BODY ME     =    F
        MIN MTM FRAC FOR ISR   =1.0000E-04
        1-MAX MTM FRAC FOR ISR =1.0000E-06

        NO EVENTS WILL BE WRITTEN TO DISK

        B_d: Delt-M/Gam =0.7000 Delt-Gam/2*Gam =0.0000
        B_s: Delt-M/Gam = 10.00 Delt-Gam/2*Gam =0.2000

        LHAPDF USED FOR BEAM 1: SET 19070 OF 0
        LHAPDF USED FOR BEAM 2: SET 19070 OF 0


        Checking consistency of particle properties


        Checking consistency of decay tables


        CHECKING SUSY DECAY MATRIX ELEMENTS

        INPUT EVT WEIGHT   =  1.0000E+00
        INPUT MAX WEIGHT   =  0.0000E+00

EVENT       1: 7000.00 GEV/C P        ON  7000.00 GEV/C P        PROCESS:  -601
SEEDS:      725123 &     1450246   STATUS:   40 ERROR:   0  WEIGHT:  1.0000E+00


                                                     ---INITIAL STATE---

IHEP    ID      IDPDG IST MO1 MO2 DA1 DA2   P-X     P-Y     P-Z  ENERGY    MASS     V-X   ..
   1 P           2212 101   0   0   0   0    0.00    0.00 7000.0 7000.0    0.94 0.000E+00 ..
   2 P           2212 102   0   0   0   0    0.00    0.00-7000.0 7000.0    0.94 0.000E+00 ..
   3 CMF            0 103   1   2   0   0    0.00    0.00    0.014000.014000.00 0.000E+00 ..

                                                    ---HARD SUBPROCESS---
```

```
IHEP     ID       IDPDG IST MO1 MO2 DA1 DA2   P-X     P-Y     P-Z  ENERGY    MASS     V-X   ..
   4 GLUON          21 121   6  10  11   7    0.00    0.00  204.5  204.5    0.00 0.000E+00 ..
   5 GLUON          21 122   6   8  17   9    0.00    0.00 -266.7  266.7    0.00 0.000E+00 ..
   6 HARD            0 120   4   5   7  10  -33.00  -38.11  -62.2  474.0  467.16 0.000E+00 ..
   7 TQRK            6 123   6   4  26   8   -5.57   54.25   82.7  201.1  175.00 0.000E+00 ..
   8 TBAR           -6 124   6   7  28   5  -10.74  -81.04 -127.9  231.7  175.00 0.000E+00 ..
   9 BQRK            5 124   6   5  30  10   12.62   23.88  -11.1   29.6    4.80 0.000E+00 ..
  10 BBAR           -5 124   6   9  32   4    3.70    2.91   -5.8    8.9    4.80 0.000E+00 ..

         OUTPUT ON LES HOUCHES EVENTS


     PROC CODE  XSECT(pb)        XERR(pb)      Max wgt(nb) No. of events

       -601    0.29506E+01     0.10898E+00     0.15177E-01      100

         OUTPUT ON ELEMENTARY PROCESS

         N.B. NEGATIVE WEIGHTS NOT ALLOWED

         NUMBER OF EVENTS   =         100
         NUMBER OF WEIGHTS  =         490
         MEAN VALUE OF WGT  =  2.9506E-03
         RMS SPREAD IN WGT  =  0.0000E+00
         ACTUAL MAX WEIGHT  =  1.5177E-02
         ASSUMED MAX WEIGHT =  1.5177E-02

         PROCESS CODE IPROC =        -601
         CROSS SECTION (PB) =   2.951
         ERROR IN C-S  (PB) =  0.1090
         EFFICIENCY PERCENT =   19.44
```