Published by Institute of Physics Publishing for SISSA
Received: *April 20, 2005*
Accepted: *June 3, 2005*
Published: *July 8, 2005*

PHASE, a Monte Carlo event generator for six-fermion physics at the LHC*

Elena Accomando, Alessandro Ballestrero and Ezio Maina
Dipartimento di Fisica Teorica, Università di Torino, and
INFN Sezione di Torino
Via P. Giuria 1, 10125-Torino, Italy
E-mail: accomand@to.infn.it, ballestr@to.infn.it, maina@to.infn.it

Abstract: PHASE is a new event generator dedicated to the study of Standard Model processes with six fermions in the final state at the LHC. The code is intended for analyses of vector boson scattering, Higgs search, three gauge boson production, and top physics. This first version of the program describes final states characterized by the presence of one neutrino, $\mathrm{pp} \to 4\mathrm{q} + \mathrm{l}\nu_\mathrm{l}$, at $\mathrm{O}(\alpha^6)$. PHASE is based on a new iterative-adaptive multichannel technique, and employs exact leading order matrix elements. The code can generate unweighted events for any subset of all available final states. The produced parton-level events carry full information on their colour and flavour structure, enabling the evolution of the partons into fully hadronised final states. An interface to hadronization packages is provided via the Les Houches Protocol.

Keywords: Spontaneous Symmetry Breaking, Higgs Physics, Standard Model, Hadronic Colliders.

*Work supported by the Ministero dell'Istruzione, dell'Università e della Ricerca under contract 2004021808_009. The work of E.A. is supported by the MIUR under Contract "Rientro dei cervelli" Decreto MIUR 26-01-2001 N.13.

JHEP07(2005)016

Contents

1. Introduction

The large energies available in the forthcoming Large Hadron Collider (LHC) will make it possible to access many-particle final states with much more statistics than before. Among these final states, six-fermion signals are of particular interest for several topics. They have a great potential in Higgs boson discovery and for analyzing vector boson scattering. The origin of the electroweak symmetry breaking is still an open problem. The most direct way to address this question is searching for the Higgs boson. At the LHC, the SM Higgs production is driven by gluon-gluon fusion. The fusion of W and Z gauge bosons (qq $\rightarrow$ qqH) represents the second most important contribution to the Higgs production cross section. Among all possible final states which might be generated by this process, the Higgs decay channel into WW, giving rise to two forward-backward jets plus four leptons or two leptons and two jets from the W's, is particularly clean. This channel has been found to be quite promising for the Higgs search in the low-intermediate mass range ($115 \lesssim M_H \lesssim 200$ GeV) favoured by present electroweak precision measurements (see for instance ref. [1]). If the Higgs exists, kinematical configurations with six fermions in the final state are then an important tool for its detection and for measuring its properties. If the Higgs is not present, the complementary approach to the question of electroweak symmetry breaking is studying vector boson scattering. In the absence of the Higgs, general arguments based on unitarity imply that massive gauge bosons become strongly interacting at the TeV scale. Processes mediated by massive vector boson scattering, VV → VV (V=W,Z), are then the most sensitive to the symmetry breaking mechanism. LHC will be able to produce for the first time processes containing boson-boson interactions at TeV scale (qq → qqVV → 6f), offering a unique possibility to understand the nature of electroweak symmetry breaking. Six-fermion processes are also strictly related to the production of three vector bosons, which would allow to extract new informations on quartic self-couplings. Moreover, they open the window on the broad field of top quark physics. These reactions give in fact access to $t\bar{t}$ and single-top production in six-fermions, enabling measurements of top mass, Wtb coupling, decay branching ratios, rare decays and all other interesting features related to the top quark. Finally, we should mention that multi-particle final states of this kind constitute a direct background to most searches for new physics.

This paper presents a new event generator, `PHASE` [2], which is designed to evaluate all Standard Model processes pp → 6f in lowest order. The code is therefore particularly appropriate to compute and analyse Higgs physics, vector boson scattering and triple gauge boson production. This first version takes into account only quark/antiquark initiated processes. Enabling the code to compute $t\bar{t}$ production, which receives its dominant contribution from gluon-gluon initiated processes, is one of the most important evolutions planned for the near future. We have built an event generator *dedicated* to all classes and topologies of final states specific for these studies. A recent example of *dedicated* program for LHC physics is `Alpgen` [3]. The complementary approach is given by *multi-purpose* programs for the automatic generation of any user-specified parton level process. The following codes for multi-parton production are available: `Amegic-Sherpa` [4], `CompHEP` [5], `Grace-Gr@ppa` [6], `Madevent` [7], `Phegas & Helac` [8], `O'Mega` & `Whizard` [9].

In the following sections we give a full description and documentation of `PHASE`. Three are the key features of our code. The first one consists in the use of a modular helicity formalism for computing matrix elements. Scattering amplitudes get contributions from thousands of diagrams. In this context, computation efficiency has a primary role. The helicity method [10] we use is suited to compute in a fast and compact way parts of diagrams of increasing size, and recombine them later to obtain the final set. In this manner, parts common to various diagrams are evaluated just once for all possible helicity configurations, optimizing the computation procedure. The second main feature concerns the integration. We have devised a new integration method to address the crucial point of reaching good stability and efficiency in event generation. Our integration strategy combines the commonly used multichannel approach [11, 12] with the adaptivity of VEGAS [13]. As the number of particles increases, the multichannel technique becomes rather cumbersome, given the thousands of resonant structures which can appear in the amplitude at the same time. For this reason, its efficiency in event generation is still debated. Conversely, VEGAS adaptivity is not powerful enough to deal with all possible peaks of the amplitude. We have merged the two strategies in a single procedure. The outcome is that PHASE adapts to different kinematical cuts and peaks with good efficiency, using only few channels per process. As third main feature, PHASE employs the *one-shot* method developed for WPHACT [14], and used for four-fermion data analyses at LEP2. In this running mode, all processes (of order 1000) are simultaneously generated in the correct relative proportion for any set of experimental cuts, and directly interfaced to hadronization and detector simulation programs, giving a fully comprehensive physical description. This possibility is relevant at the LHC, where one has to deal with a huge multiplicity of final states as well as initial states.

Section 2 reviews the general structure of the code. Section 2.1 provides a classification of the processes $\mathrm{pp} \to 4\mathrm{q} + \mathrm{l}\nu_\mathrm{l}$. Section 2.2 describes how matrix elements are computed. Section 2.3 explains the integration method, and section 2.4 addresses the event generation strategy, covering also aspects of shower evolution and hadronization. The two available running modes of PHASE are discussed in section 3. Some applications of the code for Higgs boson production are presented in section 4. Two technical appendixes describe in detail input parameters and *input-files* the user must provide. A summary is given in section 5.

2. General features of the program

PHASE is composed of several building blocks. A main body encloses the overall structure of the program, defining the sequence of operations via a set of subroutine calls. There are two possible running modes: *single-process* and *one-shot*. The sequence of operations depends on the selected mode. In the former case, the set of subroutine calls includes initialization of the selected process, and evaluation of cross section, integrand maxima and phase-space grids. The outcome obtained in *single-process* mode constitutes the essential ingredient of the *one-shot* run. In this latter mode, one generates unweighted events for all desired processes in the same run. In the following sections, we describe the general framework and criteria the sequence of main operations is based on.

2.1 Process classification

PHASE is designed to compute SM processes with six fermions in the final state at the LHC. In this first version the code includes all $\mathrm{O}(\alpha^6)$ electroweak processes with four quarks, one lepton and one neutrino in the final state, $\mathrm{pp} \to 4\mathrm{q} + \mathrm{l}\nu_\mathrm{l}$. More than one thousand processes belong to this class of final configurations, each one being described by hundreds of diagrams. At first sight, the evaluation of such reactions appears rather daunting. By making use of symmetries, the problem can be highly simplified. Taking into account one lepton type, charge conjugation and the symmetry between first and second family, the number of processes reduces to 161. A given reaction, its charge-conjugate, and the ones related by family exchange can be indeed described by the same matrix element;

particles	type	diagrams	process number
$c\bar{s}d\bar{u}c\bar{s}l\bar{\nu}$	4W	202=**101**×2	8
$u\bar{u}u\bar{u}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	8
$u\bar{u}c\bar{c}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	11
$u\bar{u}s\bar{s}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	11
$u\bar{u}b\bar{b}c\bar{s}l\bar{\nu}$	2Z2W	233=**211**+22	15
$d\bar{d}d\bar{d}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	8
$d\bar{d}c\bar{c}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	11
$d\bar{d}s\bar{s}c\bar{s}l\bar{\nu}$	2Z2W	422=**211**×2	11
$d\bar{d}b\bar{b}c\bar{s}l\bar{\nu}$	2Z2W	233=**211**+22	15
$c\bar{c}c\bar{c}c\bar{s}l\bar{\nu}$	2Z2W	1266=**211**×6	5
$c\bar{c}b\bar{b}c\bar{s}l\bar{\nu}$	2Z2W	466=(**211**+**22**)×2	11
$s\bar{s}s\bar{s}c\bar{s}l\bar{\nu}$	2Z2W	1266=**211**×6	5
$s\bar{s}b\bar{b}c\bar{s}l\bar{\nu}$	2Z2W	466=(**211**+**22**)×2	11
$b\bar{b}b\bar{b}c\bar{s}l\bar{\nu}$	2Z2W	610=(**211**+**22**+**72**)×2	8
$u\bar{u}d\bar{d}c\bar{s}l\bar{\nu}$	2Z2W+4W	312=**101**+**211**	15
$c\bar{c}s\bar{s}c\bar{s}l\bar{\nu}$	2Z2W+4W	1046=**101**×2+**211**×4	8

Table 1: Classification of pp $\rightarrow$ qq$'$ $\rightarrow$ 4q+lν_l processes. The first column shows the group list, the second the process type, the third the diagram number, and the last one the number of processes which belong to the corresponding group. The numbers in boldface represent the independent sets of diagrams, as explained in the text.

they differ by the Pdf's convolution. Moreover, all processes which share the same total particle content, with all eight partons taken to be outgoing, can be described by a single master amplitude. As a consequence, all thousand processes can be classified into 16 groups which are enumerated in table 1. By selecting two initial quarks in each particle group, one obtains all possible processes whose number is given in the last entry of the same table. For example, from the particle set $c\bar{s}d\bar{u}c\bar{s}l\bar{\nu}$ given in the first row of table 1, where all fermions are by convention outgoing, one can derive the following processes:

$$\begin{array}{llll} \bar{c}s \rightarrow d\bar{u}c\bar{s}l\bar{\nu} & \bar{c}\bar{d} \rightarrow \bar{s}\bar{u}c\bar{s}l\bar{\nu} & \bar{c}u \rightarrow \bar{s}dc\bar{s}l\bar{\nu} & \bar{c}\bar{c} \rightarrow \bar{s}d\bar{u}\bar{s}l\bar{\nu} \\ s\bar{d} \rightarrow c\bar{u}c\bar{s}l\bar{\nu} & su \rightarrow cdc\bar{s}l\bar{\nu} & ss \rightarrow cd\bar{u}cl\bar{\nu} & \bar{d}u \rightarrow c\bar{s}c\bar{s}l\bar{\nu} \end{array}$$

The calculation can be further simplified examining more closely the full set of Feynman diagrams. The amplitudes of the aforementioned 16 groups are in fact not completely independent. One can show that they are combinations of just four basic sets of Feynman diagrams, composed of 101, 211, 22, 72 diagrams respectively (they are reported in table 1 in boldface). This means that all thousand processes can be described using just a few building blocks. The immediate advantage is that any modification, like including new couplings or vertices, has to be performed only in a very restricted area of the program, and then it will be automatically communicated to all processes. The first two sets of 101 and 211 diagrams are related to the basic topologies the various processes can be classified in. In some processes, fermions can be paired only into charged currents (4W), giving rise to the first set of 101 Feynman diagrams. In some other process they can form two charged and two neutral currents (2Z2W), generating the second set of 211 diagrams. Mixed processes are described by a combination of the two sets (2Z2W+4W). If a b-pair is present, 2Z2W processes acquire an additional set of 22 diagrams, describing Higgs contributions. In case of two b-pairs, 72 more diagrams are called in. This exhausts the diagram classification for all processes with one neutrino in the final state, as we discuss in more detail in the next section.

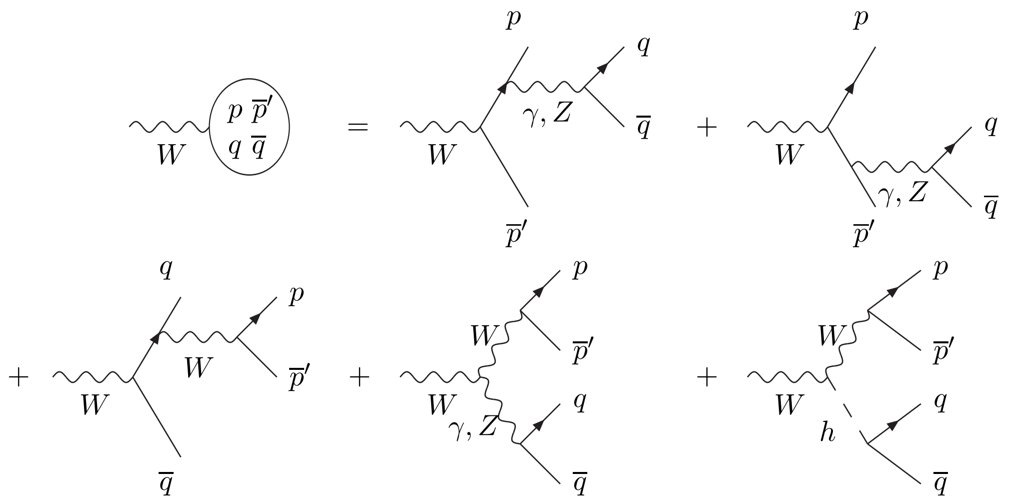
For every selected process `PHASE` employs exact matrix elements, thus providing a complete description of signals and irreducible backgrounds. As an example, the reaction $c\bar{c} \to b\bar{b}c\bar{s}l\nu$ contains contributions coming from WZ-boson scattering, Higgs production with subsequent decay to b$\bar{\text{b}}$ or WW, top pair and WWZ production (where each of them can be considered either as a signal or as a background to the others). Since our approach is based on Feynman diagrams, it is possible to compute subsets of diagrams for a given amplitude. In `PHASE`, this possibility has been exploited for the Higgs diagrams, which can be switched off by the user. For practical details on how to select the different options we refer to section B.1.

2.2 Matrix elements

All amplitudes have been written with the help of the program `PHACT` [15] (**P**rogram for **H**elicity **A**mplitudes **C**alculations with **T**au matrices), which is based on the helicity formalism described in ref. [10]. This method is very powerful in coping with the complexity of this kind of calculations. It is in fact based on a modular and diagrammatic approach. From the computational point of view, each Feynman diagram is not considered as a whole, but as a collection of several different pieces. One can thus independently compute parts of diagrams of increasing size and complexity, store them and assemble the various pieces only at the end. In this way, common subdiagrams are evaluated once, with a substantial efficiency gain. In the following, we explain the method in a pictorial way, considering both 4W and 2Z2W processes. In computing any amplitude, one starts with the most elementary building blocks given by the subdiagrams corresponding to γ, Z, W$^{\pm}$ and Higgs boson decay into a pair of external fermions:

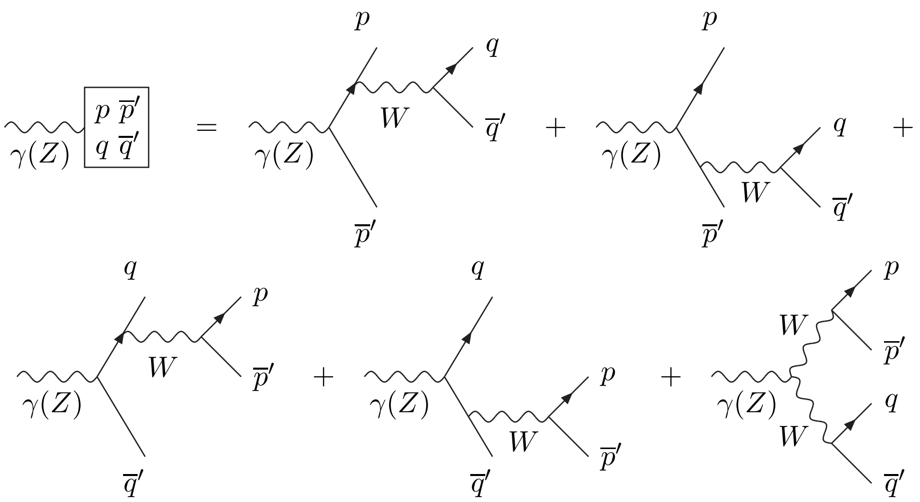
(2.1)

Here and in the following, p and p' indicate the isospin doublet components. By making use of these basic decays and of their insertions in a fermion line, one can then build the subdiagrams corresponding to a virtual γ, Z, W$^{\pm}$ or Higgs decaying into four outgoing fermions. For W-bosons we have:



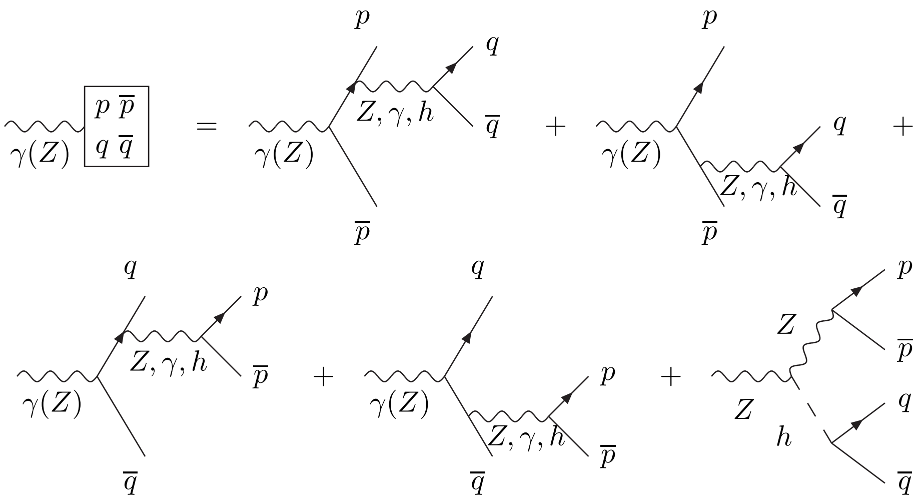
(2.2)

In the lower left-hand corner, the rightmost W-boson can be attached either on q or q̄, depending on its charge. According to the type of four-particle state (2W or 2Z), the subdiagrams corresponding to a virtual Z or γ decaying into four outgoing fermions are instead given by:



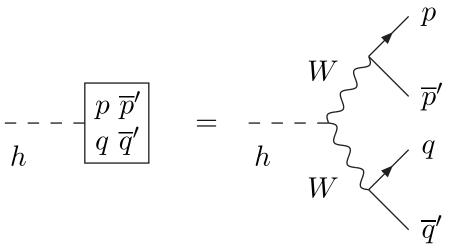
(2.3)

and



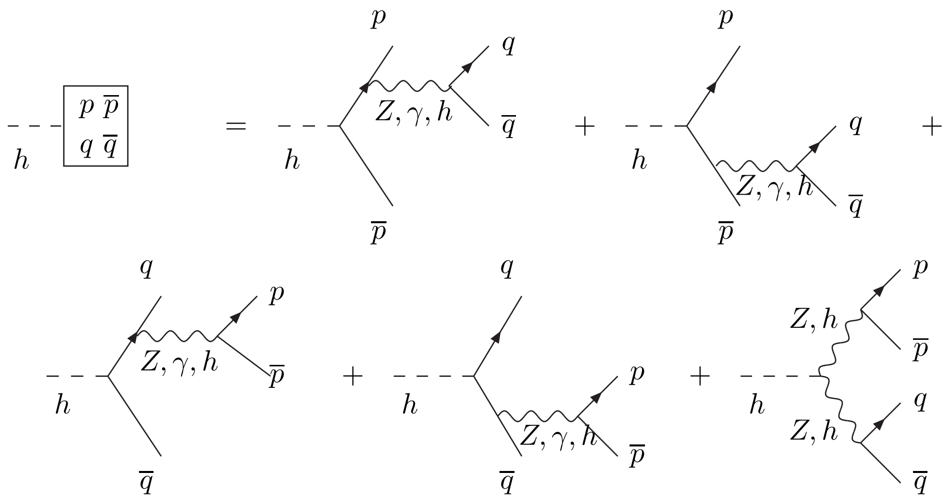
(2.4)

Diagrams with a Higgs attached to a fermion line are computed only when b-quarks are present. Finally, for the Higgs decay into four particles we have two possible sets:



(2.5)

and



(2.6)

depending on the specific four-particle configuration. Using these sets of $1 \to 2$ and $1 \to 4$ particle subdiagrams as building blocks, the 4W-type amplitude assumes the extremely concise structure given in figure 1. The full matrix element can be expressed just in terms of four main topologies. The second one drawn in the figure is described by two diagrams, as W^+W^- pairs can be formed in two different ways. The third topology represents eight diagrams, as all four fermion pairs can emit three W's and for each given fermion line one can exchange the two like-sign vector bosons. Finally, the last graph includes ninety diagrams. In case of Z-boson exchange we have in fact five subdiagrams for each side, already summed up as shown in eq. (2.3), and two different ways to form a W^+W^- pair. In presence of one neutrino in the final state, which is the case we are addressing in this first version, the number of diagrams with γ exchange gets reduced to forty. For 4W-type processes, we therefore end up with 101 basic diagrams, as reported in the first row of table 1. Analogously, 2Z2W-type processes have the simple structure outlined in figure 2. Considering eqs. (2.2)–(2.6), one can easily see that these processes have 211 diagrams if there is no b-quark. In presence of a $b\bar{b}$ pair, there are 22 additional diagrams which constitute a further independent set. Finally, one more separate set given by 72 diagrams contributes to channels with four b-quarks. Mixed processes need both 4W and 2Z2W contributions. These two sets of diagrams describe (unrealistic) processes where all fermions are different. They constitute the essential kernel, from which all other related diagrams

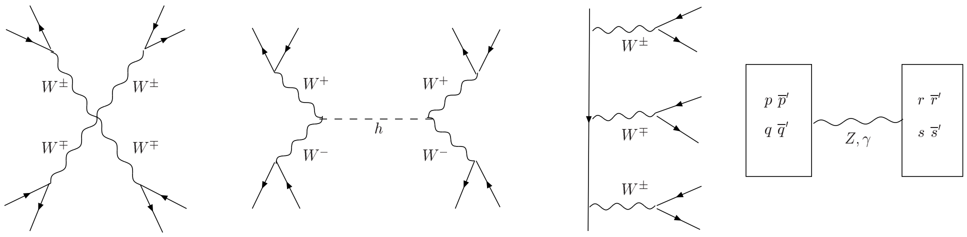


Figure 1: Diagrams for 4W-type processes of the kind $c\bar{s}d\bar{u}c\bar{s}l\bar{\nu}$.

can be derived. Additional diagrams accounting for identical particles are in fact simply obtained by fermion exchange. This explains the numbers reported in the third column of table 1. These numbers are quoted only for reference, as we do not compute every single diagram but only the few topologies of figures 1,2. The helicity amplitude formalism is appropriate both for massless and massive fermions. At the present stage, fermion masses are taken into account for *bottom* and *top* fermion lines. This strategy provides an excellent approximation to the full result in all cases which do not exhibit collinear or mass singularities. In this first version, we aim to cover all possible processes pp $\rightarrow$ 4q + l$\nu_{\rm l}$ with hard and well separated fermions in the final state. Full massive amplitudes would be however just a straightforward extension of the code, with the only drawback of slowering the program. The number of helicity states increases and new terms appear in the diagram evaluation. However the logic of constructing progressively the building blocks stays unaltered. `PHASE` is structured in such a way that makes it easy to accomodate possible future developments.

`PHASE` matrix elements, squared and summed over polarizations, have been extensively compared with `Madgraph`[7] amplitudes. A very good numerical agreement has been found. To conclude this section, let us briefly comment on the inclusion of weak boson finite-width effects. As well known, this requires a careful treatment. It is in fact closely related to the gauge invariance of the theory, and even tiny violations of Ward identities can lead to totally wrong results in many cases. There are several schemes in the literature for the introduction of the decay width in the propagators. The most appealing approach is the Fermion-Loop scheme [16], which preserves gauge invariance. It however requires the computation of a considerable number of additional terms in the amplitude. An alternative simpler option is the fixed-width scheme (FW). In the unitary gauge we work in, it consists in replacing M^2 with $M^2 - iM\Gamma$ both in the denominator and in the $p^\mu p^\nu$ term of the vector boson propagator. This scheme preserves U(1) gauge invariance at the price of introducing unphysical widths for space-like vector bosons. In `PHASE` we have chosen to implement this latter scheme.

2.3 Iterative-adaptive multichannel

In this section a new integration method is described. It employs an iterative and adaptive multichannel technique. The ability to adapt is the overriding consideration for multidimensional integrals of discontinuous and sharply peaked functions.

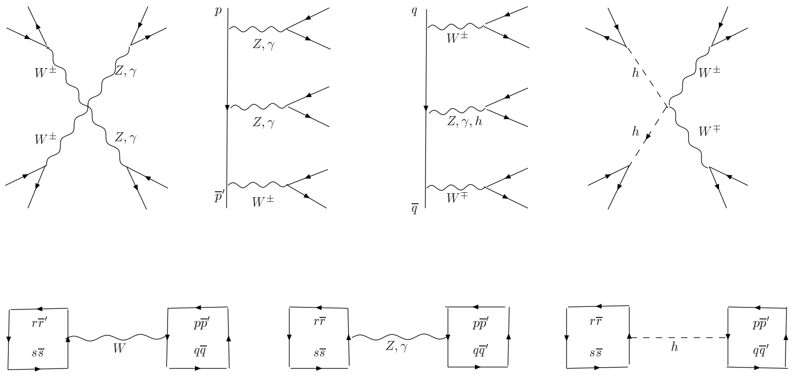


Figure 2: Diagrams for 2Z2W-type processes of the kind $b\bar{b}b\bar{b}c\bar{s}l\bar{\nu}$.

Computing a six-fermion process at hadron colliders requires an integration over a 15-dimensional space. The generic process can be written as

$$h_1 + h_2 \to f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + X \tag{2.7}$$

where h_1 and h_2 denote the incoming protons, f_i the outgoing fermions, and X the remnants of the protons. In the parton model the corresponding cross sections are obtained from the following convolution

$$\sigma^{h_1 h_2}(P_1, P_2, p_f) = \sum_{i,j} \int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 F_{i,h_1}(x_1, Q^2) F_{j,h_2}(x_2, Q^2) \int_{\Phi_{6f}} \mathrm{d}\hat{\sigma}^{ij}(x_1 P_1, x_2 P_2, p_f)$$

$$\int_{\Phi_{6f}} \mathrm{d}\hat{\sigma}^{ij} = \frac{1}{2\hat{s}} \int_{\Phi_{6f}} \prod_{i=3}^{8} \frac{\mathrm{d}^3\mathbf{p}_i}{2p_i^0} \delta^{(4)}\left(x_1 P_1 + x_2 P_2 - \sum_{j=3}^{8} p_j \right) \frac{|M(x_1 P_1, x_2 P_2, p_f)|^2}{(2\pi)^{14}} \tag{2.8}$$

where p_f summarizes the final-state momenta, F_{i,h_1} and F_{j,h_2} are the distribution functions of partons i and j in the incoming protons h_1 and h_2 with momenta P_1 and P_2, respectively, Q is the factorization scale, and $\hat{\sigma}^{ij}$ represent the cross sections for the partonic processes averaged over colours and spins of the partons. The sum $\sum_{i,j}$ runs over all possible quarks $\mathrm{u, d, c, s, b}$. Finally, the symbol Φ_{6f} denotes the six-particle phase space and $\hat{s} = (x_1 P_1 + x_2 P_2)^2$ the center of mass (CM) energy squared in the partonic system.

Integrating numerically eq. (2.8) is rather complicated. An individual process can contain hundreds of diagrams. The resonant peaking structure of the amplitude is therefore generally very rich. As a consequence the 15-dimensional space has non-trivial kinematical regions corresponding to the enhancements of the matrix elements. In a fully extrapolated setup, these peaks are simultaneously present. Of course, the requirement of suitable cuts can enhance some resonances while suppressing others. Our aim is to have maximal coverage of phase space so as to fully exploit the LHC potential in measurements and searches. Given the complexity of the final state, it is necessary to develop a reliable and efficient phase-space sampling algorithm.

Two are the most advanced and commonly adopted integration techniques: the *multichannel* method [11] and the *adaptive* approach à la `VEGAS` [13]. The two strategies are completely different. In the *multichannel* approach, mappings into phase-space variables are chosen in such a way that the corresponding Jacobians cancel the peaks of the differential cross section. These mappings are not in general unique. One normally needs several different phase-space parametrizations, called channels, one for each possible peaking structure of the amplitude. In principle every single diagram can have a different resonant pattern described by a different set of variables. In addition, some variables corresponding to an individual diagram can resonate or not. This gives rise to a huge combinatorics which requires a correspondingly large number of channels. Number and type of these mappings must be fixed a priori, before starting the integration. The *multichannel* method thus requires a guess on the behavior of the integrand function. It indeed relies on the expectation that the selected set of channels, properly weighted [12], is able to describe reasonably well the amplitude. As no adaptivity is provided (apart from the freedom to vary the relative weight of the different channels), neglecting even one channel might worsen considerably the convergence of the integral. It is thus clear that, as the number of particles increases, the use of this technique becomes rather cumbersome and time consuming.

The criteria of *adaptive* integration as performed by `VEGAS` are rather different. This approach bases its strenght on the ability to deal automatically with totally unknown integrands. By employing an iterative method, it acquires knowledge about the integrand during integration, and adapts consequently its phase-space grid in order to concentrate the function evaluations in those regions where it peaks more. In this case, the capability of adapting well to the function while integrating depends on two factors: the choice of phase-space variables and the binning refinement. `VEGAS` divides the N-dimensional space in hypercubes, and scans the integrand along the axes. For a good convergence of the integral, it thus requires amplitude peaks to be aligned with the axes themselves. The problem can be easily solved if one set of phase-space variables is sufficient to describe the full amplitude peaking structure. In this case, the alignement can always be obtained by an appropriate variable trasformation. The method becomes inefficient when it is impossible to align all enhancements with a single trasformation. This is the main weakness of the adaptive algorithm. In addition, if the binning is too coarse, some narrow peaks can excape detection, even if along the axes, with consequent instability or underestimation of the integral. The two approaches have clearly complementary advantages and disadvantages.

We have devised a new integration method, called *iterative-adaptive multichannel*, which merges the multichannel strategy with the iterative-adaptive approach. An algorithm based on a similar philosophy has been proposed in [17]. Our integration method makes use of the `VEGAS` routine. It is characterized by two main features, named *multi-mapping* and `VEGAS`-*multichannel*, which we are going to describe in the next two sections. The first one aims at reducing the number of separate channels one has to consider in the *multichannel*. The latter provides the necessary adaptivity.

2.3.1 Multi-mapping

In this section, we describe how the integrand peaking structure gets smoothed through the employment of proper random number mappings into phase-space variables. We suppose to have a unique phase-space parametrization defined by a certain set of variables. A typical example of amplitude enhancement is given by a bosonic resonance decaying into two particles. Let us take for instance the case of a fermion pair $f_i\bar{f}_j$. A natural variable is then the invariant mass $m_{ij} = \sqrt{(p_i + p_j)^2}$. Whenever m_{ij} is close to the mass of a W, Z or Higgs boson, the corresponding amplitude squared shows a Breit-Wigner resonance. The total integral can be represented as

$$I = \int d\Phi \frac{f'(\Phi)}{\left(m_{ij}^2 - M_B^2\right)^2 + M_B^2\Gamma_B^2} \qquad B = W, Z, H \tag{2.9}$$

where Φ is the full set of phase-space variables, including m_{ij}, and $f'(\Phi)$ a smooth function of m_{ij}. It is therefore convenient to perform a variable trasformation and use, instead of the invariant mass m_{ij}, an integration variable proportional to

$$x \propto \arctan\left(\frac{m_{ij}^2 - M_B^2}{M_B\Gamma_B}\right) \tag{2.10}$$

In this way, the integral in eq.(2.9) can be written as

$$I = \int d\Phi \; g(\Phi) \; f'(\Phi) = \int dx \; f''(\Phi(x)) \tag{2.11}$$

$g(\Phi)$ being the non-uniform probability density according to which phase-space variables are distributed. The function f'' is given by $f'' = f'(\Phi)/(2m_{ij}M_B\Gamma_B)$. The Jacobian of the $\Phi \to x$ trasformation cancels the Breit-Wigner peak. We refer to (2.10) as resonant mapping. The example we just discussed is very simple, and often inadequate to deal with the actual complexity of matrix elements. One can have in fact more peaks appearing on the same variable, and long not-resonant tails which extend far away from the peaks. This latter case is more and more severe as the collider energy increases. To solve this problem, we have introduced *multi-mapping*. Let us consider the case of a neutral fermion pair $f_i\bar{f}_j$ which could originate from the decay of a Z or a Higgs boson. For this particle configuration, a double mapping of the type (2.10) is performed simultaneously on m_{ij}. In order to cover all not-resonant regions, a uniform mapping (flat) is employed in the remaining integration range. In this particular case and assuming $M_H > M_Z$, we end up with five integration domains and three corresponding mappings: flat, Z-resonant, flat, Higgs-resonant, flat. This is what we call *multi-mapping* on a given variable. The advantage of using a *multi-mapping* is that the same channel can enclose several phase-space parametrizations, with a substantial gain in efficiency and CPU time. The number of separate channels decreases considerably. Just to give an idea of the mapping combinatorics, let us consider the decay of a neutral boson into four fermions $B \to f\bar{f}f'\bar{f}'$. Among all possible integration variables, we can choose the three invariant masses m_B, $m_{f\bar{f}}$ and $m_{f'\bar{f}'}$. With three mappings per variable, as we said before, this generates 27 mappings. In a standard multichannel, 27 distinct channels would be required. In our approach, a single channel can cover all different kinds of triply, doubly, singly and not-resonant topologies, relying on integration adaptivity.

The previous example of resonant multi-mapping does not exhaust all possible amplitude peaking structure. For instance, one can have narrow peaks also in t-channel propagators, when one of the outgoing particle is emitted in the forward/backward direction with respect to the beam. The mapping for these small angle regions is inspired to the method of ref. [18]. *Multi-mapping* applies here as well. The range of the phase-space variable, which is the denominator of the t-channel propagator, can be divided into two regions. One corresponds to small scattering angles, and is mapped to a power-like behaviour. The other one is related to a possibly large not-resonant range. Both types of multi-mapping, acting on different phase-space variables, can be combined within the same channel to describe the most relevant part of the amplitude peaking structure. Integration adaptivity takes care of the residual discrepancy between our parametrization and the actual behaviour of the amplitude.

2.3.2 `VEGAS` multichannel

While *multi-mapping* is extremely useful to improve the convergence of `VEGAS` integration within a single phase-space parametrization, in general several such parametrizations are needed. In this case, one has to introduce N different channels (in standard multichannel language) with their proper *multi-mapping*. Each channel defines a non-uniform probability density $g_i(\Phi)$, which describes a specific class of amplitude peaks. If we had just a single channel, as in the previous section, denoting with $f(\Phi)$ the function to be integrated, we would write

$$I = \int d\Phi f(\Phi) = \int d\Phi \; g(\Phi)\frac{f(\Phi)}{g(\Phi)} = \int d\Phi \; g(\Phi) f'(\Phi) = \int dx \; f'(\Phi) \qquad (2.12)$$

where $f'(\Phi)$ represents the smooth part of the integrand, once the peaking structure has been canceled. In presence of N channels, the sum of the probability densities, properly weighted, should give the best description of the matrix element squared. Generalizing eq.(2.12) to a number N of channels, one can then write

$$I = \int d\Phi f(\Phi) = \sum_{i=1}^{N} \alpha_i \int \frac{d\Phi \; g_i(\Phi) f(\Phi)}{\sum_{i=1}^{N} \alpha_i \; g_i(\Phi)} = \sum_{i=1}^{N} \alpha_i \int dx_i \; f'(G_i^{-1}(x_i)) = \sum_{i=1}^{N} \alpha_i I_i \qquad (2.13)$$

α_i being the so called *weight* of the i-th channel ($x_i = G_i(\Phi)$), and $f'(\Phi)$ the smoothed integrand. The α_i's quantify the relevance of the different peaking structures of the amplitude. They must be chosen reasonably well in order to fit the integrand, i.e. to obtain a well behaved function $f'(\Phi)$. Owing to the very poor knowledge of the integrand, it is rather difficult to guess these values a priori. Usually, they are computed and optimized during the integration run. The algorithm described in eq.(2.13) is nothing else than the standard multichannel. In this method, the integral is computed in a single run, picking up the various channels with probability given by the corresponding α_i weight. In the *iterative-adaptive multichannel*, the integral in eq. (2.13) splits in N distinct contributions. The presence of identical final-state particles increases the possible list of resonant structures. In order to keep the number of separate integration runs manageable, we include all jacobians generated by particle exchange in the denominator of eq.(2.13), while exploiting the freedom to relabel the momenta to regroup all integration runs related by particle exchange to a single one. Owing to the *multi-mapping* technique previously described, and to the adaptivity of the integration algorithm, a maximum of seven channels is required to calculate all processes in table 1. The criteria to automatically define number and type of channels needed for a given process are the following. We identify phase-space variables in which enhancements can appear due to boson and top propagators. We then consider the different sets of variables in which the maximum number of such propagators can be simultaneously present. These will determine our channels. Multi-mapping and adaptivity will take care of all related partially-resonant or not-resonant configurations, as explained in section 2.3.1. Each channel in eq. (2.13) is integrated separately with VEGAS. In the *iterative-adaptive multichannel* method, a thermalization stage with a relatively small number of points is employed to determine the relative weights α_i of the various channels as follows. In every thermalizing iteration, all channels are independently integrated for some set of α_i. At the end of each iteration, a new set of phase-space grids (one for each channel), and an improved set of α_i are computed. The criteria for weight optimization we adopt is

$$\alpha_i = \frac{I_i}{\sum_{i=1}^{N} I_i} \tag{2.14}$$

where I_i is the i-channel integral. The new sets of α_i and grids are then used in the next iteration. The procedure is repeated until a good stability of the α_i is reached. In the standard multichannel method, the final result depends sensitively on the accuracy obtained for the α_i values. In our approach, owing again to integration adaptivity, only a rough estimate of the relative weights of the individual channels is sufficient for an accurate integration. Having established the relative weights and having obtained the initial grids, one can start the actual integration run, where the N channels are evaluated in sequence. The iterative-adaptive algorithm is applied at this stage as well, and new grids are generated after each step. The last iteration produces N final grids, which contain full information on the integrand function. The grids are stored in files, called in the following *grid-files*, and used whenever needed in the so called *one-shot* event generation, which we are going to describe in the next section.

2.4 Unweighted event generation

Once phase-space grids are ready, the generation of unweighted events can start. This procedure, called *one-shot*, represents one of the main features of PHASE. Inspired to the method used in ref. [14], it allows the user to generate unweighted events not on a process by process basis, but for all possible processes (or any selected subset of them) in just a single run. The result is a complete event sample, where all included final states appear in the right relative proportion. The algorithm is based on the *hit-or-miss* method. Thus, it needs to know the maximum value of the integrand functions of all channels and processes. When running in *one-shot* mode, all necessary informations about processes are read from the *grid-files*, where they have been recorded during the grid preparation. In addition to the phase-space grid, these files contain also process and channel labels, the corresponding α_i weights, and the maximum value of the integrand function. Relying on these inputs, the code computes the probability according to which every single channel is picked up during the unweighted generation. Events will be generated using a modified version of VEGAS, which chooses at random a cell of the phase-space grid read from the *grid-files*. A good determination of the grids traslates into high efficiency. The procedure is repeated until the required number of unweighted events is produced.

Every generated event may be either directly passed to `PYTHIA` [19], for showering and hadronization, or can be stored into files for further processing. In this way, one has a complete and accurate tool for realistic experimental simulations. This step is performed according to the Les Houches Protocol [20], a set of common blocks for passing event configurations from parton level generators to parton shower and hadronization packages.

3. Running modes

In this section we discuss how to run the code. We just give a guideline, useful to understand running mechanism and possible options. For a more detailed description we refer to appendix B. The program has two modes of operation: *single-process* and *one-shot*, which are selected by the input values `ionesh=0,1` respectively. In the former mode (`ionesh=0`), the code evaluates all N processes one wants to generate in N separate runs, and prepares the *grid-files* to be used in the *one-shot* generation. In this latter mode (`ionesh=1`), the code generates instead unweighted events for all processes (or any subset of them specified by the user) in the same run. The two modes correspond to two distinct branches of the program. They thus need two different input sets. Both sets are included in the same *input-file*. The first part of this file is common to both modes. The rest depends on the selected mode. A practical feature of the input routine is that variables, which do not need to be specified in the chosen running mode, can be left in the *input-file* without harm. They are simply ignored. The *input-file* must always be called `r.in`. A sample `r.in` is supplied with the program package. A detailed description, explaining meaning and possible values of input variables, is given in appendix B. In this section, we just discuss the computational strategy, and the main options which are available.

3.1 Single-process

The use of this mode is twofold. The easiest option is computing the cross section of a specific process. This might be useful for some test or dedicated analysis. The alternative choice is to employ the *single-process* mode as necessary pre-run for the *one-shot* generation. In this case, the main purpose is the production of phase-space grids. This implies an extensive use of `ionesh=0`, devoted to compute in separate runs all processes one intends to consider in the *one-shot* generation. A Perl script (`setupdirSGE.pl`) for creating a tree-structure with subdirectories and *input-files*, one for each process, is provided in the program package. For every single evaluation, the user must specify the desired reaction. The variable to fill is called `iproc`, and uses the standard Monte Carlo particle numbering scheme, as described in appendix B. Once the process has been selected, a first routine initializes parameters and variables. It defines number and kind of channels appearing in the integration, according to the algorithm discussed in section 2.3.2. These informations, along with the corresponding phase-space parameters, are then passed to the phase-space generation routines, and lately to the integration algorithm. The integration routine is based on VEGAS; it thus needs VEGAS parameters to be defined. The user must specify integration accuracy, number of iterations and Monte Carlo points per iteration, both for thermalization and actual integration. As explained in section 2.3.2, the program has an initial warm-up stage, followed by the actual process evaluation. For every single process, in thermalization the code determines the relative weight of each channel appearing in the multichannel integration, and produces a first instance of phase-space grids (one per channel). These grids are then used as a starting point for the second step, which consists of M separate integrations, where M is the number of channels. Each integration typically proceeds through several iterations. At the end of each iteration, the phase-space grid gets refined in an effort to decrease the overall variance. After the last iteration, the optimal grid is recorded in the *grid-file* named PHAVEGAS0i.DAT, where i represents the corresponding channel index. In the same file, are also stored the maximum of the integrand function $w0$ produced in the next-to-last iteration, and the maximum $w1$ produced in the last iteration.

Before concluding this section, let us discuss the PHASE options for imposing kinematical cuts. The *input-file* (r.in) provides a predetermined set of kinematical cuts. Basically, two different types of cuts have been predisposed. A first set allows to approximately simulate detector acceptance and separation requirements. The second one allows to require two forward-backward jets, and two jets and one charged lepton in the central region. This signature helps to suppress QCD background, enhancing Higgs and vector boson scattering signals. The complete list of predetermined cuts, their meaning and logic are given in appendix B. In addition, it is also possible to include extra user-specified cuts via a routine called IUSERFUNC, an example of which is provided in the program package. This part of the input is common to both running modes, and must be always kept unchanged when passing from ionesh=0 to ionesh=1. It constitutes in fact the setup under which phase-space grids are produced. In order to give the possibility of imposing other cuts at generation level, the *input-file* has also a cut section specific of the *one-shot* mode, which we describe in the next section.

3.2 One-shot

Once phase-space grids are ready, the *one-shot* mode allows the user to generate unweighted events for all processes simultaneously. The outcome is a complete event sample, able to simulate the full six-fermion production. When running in this mode, the user must specify which processes (and corresponding channels) should be considered in the event generation. One can choose to produce events for all possible processes, or just for a specified subset of them. The simplest option is generating events for an individual process. In any case, for a meaningful generation, the *grid-files* of all channels corresponding to each selected process must be included in r.in. From the *grid-files*, ionesh=1 mode reads all necessary informations for the *hit-or-miss* selection.

Phase-space grids and integrand maxima are prepared according to the cuts specified during the ionesh=0 pre-run. When running in *one-shot*, one can impose new kinematical cuts. This option is implemented as follows. The structure of the common inputs, given in section 3.1, is exactly repeated in the cut section specific of the *one-shot* mode. The corresponding variables are the same as those in the common input section; they are just renamed with a suffix - `os` - appended. These additional cuts, operating at generation level, are obviously effective only if more restrictive than the common ones. The reason for doubling the cuts is the following. There are different attitudes concerning signal selection and cuts. One possible choice is to generate unweighted events with the loosest conceivable setup, and apply cuts directly on the produced event sample. This gives more freedom in varying the setup, according to the analysis at hand, without redoing the event generation; the drawback is an overproduction of events in regions which might not be of any interest. In `PHASE`, this strategy traslates in producing both phase-space grids and generated events with the same setup. A different general attitude is to implement cuts during event generation, in order to produce a sample already focused on the particular analysis to be performed. In this case, `PHASE` provides two options. One can choose to run in *single-process* mode under the preferred cuts, in order to prepare grids specific for a certain study, and generate events with those same grids and cuts. This case does not differ from the previous one, as to the *input-file*. Otherwise, one could also produce phase-space grids with a looser setup (to retain all possible information) and impose more restrictive cuts later, when running in *one-shot* mode. Common cuts in `r.in` must be kept identical in both runs. What changes is the set of cuts specific of the *one-shot* mode. Let us notice that, in this latter case, phase-space grids are prepared once for all, and one can perform different analyses by simply varying the *one-shot* specific cuts.

After choosing the processes to be represented in the event sample, one should specify the desired number of unweighted events. Once produced, these events are recorded in the file `phamom.dat`, using the information stored in the two `COMMON BLOCK`, `HEPRUP` and `HEPEUP`, according to the Les Houches Protocol. If required, each generated event is then passed to `PYTHIA` for showering and hadronization via a call to `PYEVNT`.

4. Sample results

In this section we present some applications of `PHASE`. In particular, we focus on the Higgs signal and its electroweak irreducible background. In the class of processes we are addressing in this first version of the code, Higgs production is dominated by vector boson fusion followed by the Higgs decay into WW pairs. This gives rise to a well known distinctive signature with two forward/backward tagging jets, and two quarks and one charged lepton from the W's in the central region. In the following, we show examples of cross sections and distributions for two values of the Higgs mass: $M_{\rm H}$=140 and $M_{\rm H}$=500 GeV. In the first case, the Higgs width is extremely narrow and WW pairs are produced below threshold. In the latter case, the Higgs resonance is rather broad, and the W's are generated around and above their on-shell values.

After producing phase-space grids, we have generated two samples of one million unweighted events each, for all possible processes with one muon in the final state: pp $\rightarrow$ 4q$+\mu\nu_\mu$. In our notation $\mu\nu_\mu$ indicates both $\mu^-\bar{\nu}_\mu$ and $\mu^+\nu_\mu$. The produced event samples, one for each Higgs mass, are thus representative of all reactions shown in table 1, including all possible quark flavours. We consider a total of 644 processes. Not all of them contain the Higgs signal. Some channels only contribute to the irreducible background, but they must be included for any meaningful analysis.

For the Standard Model parameters we use the input values:

$$\begin{aligned} \mathrm{M_W} &= 80.40\,\mathrm{GeV}\,, & \mathrm{M_Z} &= 91.187\,\mathrm{GeV}\,, \\ \mathrm{M_t} &= 175.0\,\mathrm{GeV}\,, & \mathrm{M_b} &= 4.8\,\mathrm{GeV}\,, \\ \Gamma_\mathrm{W} &= 2.042774\,\mathrm{GeV}\,, & \Gamma_\mathrm{Z} &= 2.5007\,\mathrm{GeV}\,. \end{aligned} \tag{4.1}$$

We adopt the so called G_μ-scheme (see appendix A), and use the CTEQ5L parton distributions [21] at the factorization scale:

$$Q^2 = \frac{1}{6}\sum_{i=1}^{6} \mathrm{P}_T^2(i) + M_W^2 \tag{4.2}$$

where $\mathrm{P}_T(i)$ is the transverse momentum of the i-th final state particle. We have implemented a general set of cuts, appropriate for LHC analyses. For the charged lepton we require

$$\mathrm{E}(l) > 20\,\mathrm{GeV} \qquad \mathrm{P}_T(l) > 10\,\mathrm{GeV} \qquad |\eta_l| < 3 \tag{4.3}$$

Analogously, for the quarks we have

$$\mathrm{E}(j) > 20\,\mathrm{GeV} \qquad \mathrm{P}_T(j) > 10\,\mathrm{GeV} \qquad |\eta_j| < 6.5 \qquad m_{jj'} > 20\,\mathrm{GeV} \tag{4.4}$$

where $m_{jj'}$ denotes the invariant mass of any jet pair. These cuts approximately simulate detector acceptance, and are common to all results presented in the following. In order to analyse the Higgs signal, we have plotted in figure 3 the total invariant mass of the two most central quarks, the muon and the neutrino, which are supposed to originate from the Higgs decay into WW. The two upper plots include the basic acceptance cuts of eqs. (4.3)-(4.4). In the lower ones, we have also specifically required two forward and backward jets, two central quarks, and one central charged lepton as follows:

$$1 < |\eta_{jf}| < 6.5 \qquad -6.5 < |\eta_{jb}| < -1 \qquad |\eta_{jc}| < 3 \qquad |\eta_l| < 3 \tag{4.5}$$

where jf, jb and jc indicate forward, backward and central jets respectively. From the figure, one can clearly see the Higgs peak and the huge continuum background. A realistic study with detector simulation and reconstruction effects is needed to determine the actual shape of the peak, and the signal to background ratio. In this simple illustration, to clarify the origin of the background, we have considered two sets of events for each Higgs mass and each setup. One is the full sample, containing events from all available processes. The other represents a subset, where only final states with no b-quarks are included. The Higgs signal is essentially the same in the two cases, which instead differ substantially outside the resonance. The latter sample, which is much cleaner and does not suffer from possible electroweak top background, includes about one half(third) of the total number of events for M_H=140(500) GeV. If one only imposes basic acceptance cuts, final states with b-quarks are dominant. In order to see this in more details, we show in table 2 cross sections for all

JHEP07(2005)016

M_H (GeV)	σ_{0b} (pb)	σ_{1b} (pb)	σ_{2b} (pb)	σ_{3b} (pb)	σ_{4b} (pb)
140	0.19844(4)	0.11174(2)	0.14502(4)	$0.6290(4)\times10^{-2}$	$0.1395(3)\times10^{-3}$
500	0.12982(2)	0.10767(2)	0.14266(3)	$0.5507(4)\times10^{-2}$	$0.956(3)\times10^{-4}$

Table 2: Cross sections for processes with n b-quarks/antiquarks in the final state, with $n = 0, 4$, for two values of the Higgs mass. Basic acceptance cuts are included.

processes with n outgoing b-quarks, where $0 \leq n \leq 4$. Most of the contribution comes from processes with one and two b's in the final state, which in our sample are dominated by electroweak single-top and $t\bar{t}$ production, respectively. If we require two forward-backward jets as in eq.(4.5), the signal to background ratio improves even in presence of b-quarks in the final state, as shown in the two lower plots of figure 3. This suggests that the possibly dangerous top background to the Higgs search can be reduced either employing b-tagging techniques or imposing appropriate cuts. This analysis goes beyond the scope of this paper, in which we simply present the potentiality of `PHASE` for phenomenological studies.

5. Conclusions

The analysis of six-fermion final states is an important task at the LHC, owing to the several interesting subprocesses involved. These include Higgs and top production, vector boson scattering, and triple gauge boson production. In this paper, we have presented the Monte Carlo event generator `PHASE`, which in this first version computes all processes $pp \to 4q + l\nu_l$ at $O(\alpha^6)$. `PHASE` works with exact matrix elements. It employs a new *iterative-adaptive multichannel* method for the phase-space integration. The algorithm considerably improves integral convergence and generation efficiency. The code makes use of the *one-shot* technique, which allows the user to generate in a single run an event sample fully representative of all available final states (of the order of 1000). Upon request, the unweighted parton-level events are passed to hadronization packages via the Les Houches Protocol, and eventually to detector simulation codes. In this way, `PHASE` can provide realistic event samples, merging complete and precise theoretical computations for six-fermion processes with a detailed simulation of the experimental apparatus.

We have discussed in detail the general features of `PHASE`. Some examples of the performance of the code have been shown. In particular, we have presented cross sections and distributions relevant to Higgs production, including all final states with one muon, $pp \to 4q + \mu\nu_\mu$. The flexibility of the underlying concepts and the general structure of `PHASE` makes it easy to accomodate future developments. Enabling the code to calculate all processes $pp \to 6f$ at $O(\alpha_s^2\alpha^4)$ is the most important evolution planned for the near future.

The first version of the program, `PHASE 1.0`, can be downloaded from the following URL: `http://www.to.infn.it/~ballestr/phase/`. All new versions of the code will be posted in this website.

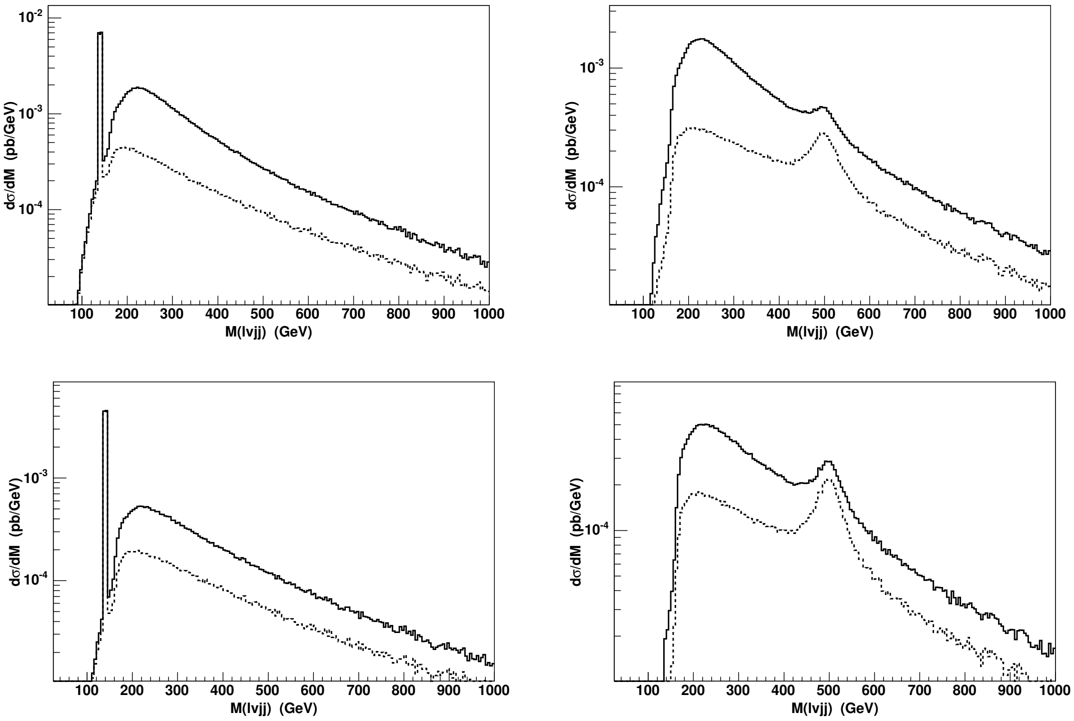


Figure 3: Invariant mass distribution of the two leptons and the two central jets for $M_{\rm H}$=140 and $M_{\rm H}$=500 GeV, left- and right-hand side respectively. The solid curve in-cludes all processes, the dashed one only final states with no b-quark. The two upper plots have basic acceptance cuts, the lower ones include additional forward-backward jet requirements, as explained in the text.

Acknowledgments

We thank Chiara Mariotti for the constant interest in our work and for stimulating discussions and suggestions. Giuseppe Bevilacqua and Sara Bolognesi are gratefully acknowledged for their advice and tests. Fabio Maltoni is acknowledged for comparisons. All diagrams have been drawn with JaxoDraw [22].

A. Parameters

Standard model parameters are defined in the routine `coupling.f`. In our notation, `rmw`, `rmz`, `rmt`, and `rmb` are the W, Z, top and bottom masses respectively. The total W and Z widths are given by `gamw` and `gamz`. Higgs and top widths are computed in the same routine by standard formulas.

As a default, `PHASE` employs the G_μ-scheme defined by the input set: $M_{\rm W}$, $M_{\rm Z}$ and G_F. According to this scheme, the calculated parameters are

$$sin^2\theta_{\rm W} = 1 - \left(\frac{M_{\rm W}}{M_{\rm Z}}\right)^2 \qquad \alpha_{em}(M_{\rm W}) = \frac{\sqrt{2}}{\pi} G_F M_{\rm W}^2 sin^2\theta_{\rm W}$$

where $\theta_{\rm W}$ is the weak mixing angle, and α_{em} the electromagnetic fine structure constant.

The code uses the `CTEQ5*` Pdf parametrization, where the * indicates the possible schemes. As a default, we have implemented the `LO`-scheme. This can be modified by the user through the variable `Iset` defined in the main body of the program, `phase.f`.

B. Input-file

In the following sections, we describe how to use input parameters and flags to exploit the various possibilities of `PHASE`. The syntax of the input is almost identical to the one required by the CERN library routine `FFREAD`. Routines internal to `PHASE` are however used (`iread, rread`), so that real variables can (and must) be given in double precision. All lines in the *input-file* must not exceed 80 characters. Writing `*` or `C` characters at the beginning of a line identifies it as a comment line. Comment lines can be freely interspersed within the *input-file*, with the only obvious exception that they must not interrupt a list of input values for a single array variable. The name of the variable to be read must be specified as the first word of a line. Its value (values) must follow it. The list of values can span several lines. A practical feature of the input routine is that variables, which are not needed to be specified in a given run, can be left in the *input-file* without any harm. They are simply ignored. The input values which are actually read are then written in the *output-file*. Two sample *input-files* for `PHASE` are provided in the program package, `inp.st0` for `ionesh=0` and `inp.st1` for `ionesh=1`. When running the program, they have to be renamed. The actual *input-file* must always be called `r.in`. All energy values must be given in GeV, while angles are in degrees. Kinematical variables are all defined in the collider frame. For yes/no flags, we adopt the convention that `0` corresponds to `no` and `1` to `yes`. All entries in the following sections, which describe input variables in more detail, are in the form:

<u>`variable-name`</u>:

or

<u>`variable-name`, (`full-list-of-possible-values:  val1/val2/...`)</u>:

B.1 Common inputs

<u>`ionesh`, (`0/1`)</u>: this flags selects the basic operation mode of `PHASE` as explained in section 3.

<u>`idum`</u>: random number generation seed. Must be a large negative integer.

<u>`ecoll`</u>: total center of mass energy of pp collisions.

<u>`rmh`</u>: Higgs mass. Setting `rmh` to a negative number allows the user to switch off Higgs diagrams.

i_ccfam, (0/1): if i_ccfam=0 only processes explicitely required by the user are computed. If i_ccfam=1 the required processes are computed along with the reactions obtained interchanging first and second family of quarks and antiquarks, and with the reactions obtained by charge conjugation. For instance, if the user-specified process is

$$u\bar{u} \to b\bar{b}c\bar{s}\mu^-\bar{\nu}_\mu \tag{B.1}$$

with i_ccfam=1, all the following processes are computed or generated in the same run:

$$\begin{array}{ll} u\bar{u} \to b\bar{b}c\bar{s}\mu^-\nu_\mu & c\bar{c} \to b\bar{b}u\bar{d}\mu^-\nu_\mu \\ \bar{u}u \to \bar{b}b\bar{c}s\mu^+\bar{\nu_\mu} & \bar{c}c \to \bar{b}b\bar{u}d\mu^+\bar{\nu_\mu} \end{array}$$

This computation involves a sum over Parton Distribution Functions, and it gives different cross section and different grids, if compared to the i_ccfam=0 case. When generating events in (ionesh=1) mode, it is thus important to give i_ccfam the same value used for preparing the *grid-files*.

B.1.1 Cuts

All cuts in PHASE are meant at parton level, before showering and hadronization. As described in section 3.1, two types of predetermined cuts are provided in PHASE. The basic one simulates detector acceptance and separation criteria. The corresponding variables are characterized by the suffixes lep and j, which refer to *charged* lepton and quark/antiquark, respectively. The second kind of cuts is instead focused on Higgs search and vector boson scattering analyses. In particular, we define the most forward and most backward jets. The remaining two jets are called central. In this case, the suffixes jf, jb and jc denote forward, backward and central jets, respectively. A yes/no flag specifies whether the corresponding cut is activated or not. The name of this flag in most cases is the name of the corresponding variable with i_ prepended. Exceptions to this rules will be pointed out; in all other cases we will give only the variable name and the corresponding flag will be understood. Variables are defined as follows:

e_min_lep: minimum energy of charged leptons.

pt_min_lep: minimum transverse momentum of charged leptons.

eta_max_lep: maximum absolute value of charged lepton pseudo-rapidity.

ptmiss_min: minimum missing transverse momentum (at present it coincides with the neutrino transverse momentum).

e_min_j: minimum energy of quarks/antiquarks.

pt_min_j: minimum transverse momentum of quarks/antiquarks.

eta_max_j: maximum absolute value of quark/antiquark pseudo-rapidity.

i_eta_jf_jb_jc, (0/1): specifies whether the following triplet of cuts are activated:

eta_def_jf_min: minimum value of the pseudo-rapidity of the most forward quark/antiquark.

eta_def_jb_max: maximum value of the pseudo-rapidity of the most backward quark/antiquark.

eta_def_jc_max: maximum absolute value of the pseudo-rapidity of the remaining two (*central*) quarks/antiquarks.

pt_min_jcjc: minimum total transverse momentum of the two central quarks/antiquarks.

rm_min_jj: minimum invariant mass of quark/antiquark pairs.

rm_min_jlep: minimum invariant mass of any pair of charged-lepton and quark/antiquark.

rm_min_jcjc: minimum invariant mass of the two central quarks/antiquarks.

rm_max_jcjc: maximum invariant mass of the two central quarks/antiquarks.

rm_min_jfjb: minimum invariant mass of the most forward and most backward quark/antiquark.

eta_min_jfjb: minimum absolute value of the difference in pseudo-rapidity between most forward and most backward quark/antiquark.

d_ar_jj: minimum separation in $\Delta R = \sqrt{\Delta\phi + \Delta\eta}$ between any two quarks/antiquarks.

d_ar_jlep: minimum separation in ΔR between any quark/antiquark and charged lepton.

thetamin_jj: minimum angular separation between two quarks/antiquarks.

thetamin_jlep: minimum angular separation between quarks/antiquarks and charged leptons.

i_usercuts, (0/1): determines whether additional user-specified cuts are required. These requirements must be implemented in a routine called IUSERFUNC, an example of which is provided in the program package.

B.2 ionesh=0 input

iproc: specifies the desired process using the standard Monte Carlo particle numbering scheme:

d	u	s	c	b	e^-	ν_e	μ	ν_μ	τ	ν_τ
1	2	3	4	5	11	12	13	14	15	16

Antiparticles are coded with the opposite sign. The variable iproc is an eight-component vector, where the first two entries represent the initial state partons. As an example, iproc=(3,-4,2,-2,3,-3,13,-14) corresponds to the reaction $s\bar{c} \to u\bar{u}s\bar{s}\mu\nu_\mu$. In ionesh=0 mode, the process is computed exactly as written by the user, assuming the first incoming particle to be moving in the $+z$ direction and the second one in the $-z$ direction (the realistic case at a pp collider, which accounts for the exchange of the two initial particles, is implemented only in ionesh=1 mode with the flag i_exchincoming).

acc_therm: integration accuracy in thermalization. When this accuracy is reached for a given channel, thermalization of that channel stops.

ncall_therm: maximum number of points for each iteration during thermalization.

itmx_therm: maximun number of iterations used to evaluate each integral in thermalization.

acc: accuracy of the actual integration. When this accuracy is reached for a given channel, integration of that channel stops.

ncall: maximum number of points for each iteration of the actual integration. In general, VEGAS uses a number of ncall_therm and ncall lower than the input ones. The actual value is written in the *output-file*, where also the number of points which survive all the cuts (effective ncall) is reported.

itmx: maximum number of iterations used to evaluate the integral and refine the grid. A number of iterations between 3 and 5 is normally the best choice. If higher precision is requested, it is usually more convenient to increase ncall rather than itmx. The user must be aware of the fact that if no point survives the cuts during an iteration, either during thermalization or at the integration stage, VEGAS will stop with an error.

iflat, (0/1): this yes/no flag must be set to 1 in order to produce the phase-space grids for later unweighted event generation. If iflat=1, the program also returns the maximum of the integrand function $w0$ produced in the next-to-last iteration, the maximum $w1$ produced in the last iteration, and the number of points with weight greater than $w0 * scalemax0$ visited during the last iteration. By default we take $scalemax0 = 1.1$. The maximum $w1$, stored in PHAVEGAS0i.DAT, is then used in *one-shot* mode, which employs the *hit-or-miss* method for the unweighted event generation.

B.3 ionesh=1 input

nunwevts: number of unweighted events the user desires to produce. The program stops only when this number has been reached.

scalemax: factor used to replace the integrand maximum in the *hit-or-miss* procedure. This coefficient multiplies the maximum value of the differential cross section, found during the phase-space grid preparation. It can be used to compensate for the fact that the maximum determined by the program may be smaller than the true maximum. Setting this parameter too high would decrease the efficiency in generation. On the other side, it is not advisable to lower the maximum value for a more efficient unweighting. The generated sample could be biased.

`iwrite_event`, `(0/1)`: yes/no flag which decides whether the generated events are recorded in the file named `phamom.dat`. For writing this file, the code uses the information stored in the two `COMMON BLOCK HEPRUP` and `HEPEUP` according to the Les Houches Protocol.

`ihadronize`, `(0/1)`: yes/no hadronization required. If `ihadronize=1`, each generated event is passed to `PYTHIA` for showering and hadronization via a call to `PYEVNT`, using the Les Houches Protocol.

`i_exchincoming`, `(0/1)`: yes/no flag which symmetrizes the initial state. If the value is set to zero, the process is generated exactly as required, assuming the first incoming particle to be moving in the $+z$ direction and the second one in the $-z$ direction. Otherwise, the two initial particles are assigned at random to the two protons, doubling the corresponding cross section if they are not identical. As a consequence, cross sections computed in `ionesh=1` mode can be different from those computed in `ionesh=0`.

`i_emutau`, `(0/1/2)`: determines which charged leptons are present in the generated sample. If `i_emutau=0`, only events containing the charged lepton specified by the user in the vector `iproc` will be generated. If `i_emutau=1`, events containing μ and events containing e will be generated with the same frequency. Finally, if `i_emutau=2` events containing μ, events containing e and events containing τ will be generated with the same frequency.

B.3.1 Cuts

`iextracuts`: determines whether additional cuts are required at the generation stage. The *input-file* for `ionesh=1` must contain the same set of cuts used for generating phase-space grids, and defined in section B.1. As a consequence, additional cuts will be effective only if they are more stringent than those imposed in the `ionesh=0` pre-run. These extra cuts must be defined setting `iextracuts=1`, followed by the new cut list. The names of the corresponding variables are equal to those in the common input section with the suffix - `os` - appended.

`i_usercutsos`: determines whether additional user-specified cuts are required at the generation stage. These requirements must be implemented in a routine called `IUSERFUNCOS`, an example of which is provided in the code package. Obviously, the comments concerning the relationship between cuts in the grid-production and event-generation stage also apply to the user-specified cuts.

B.3.2 Processes

<u>`nfiles`</u>: number of *grid-files* (`PHAVEGAS0i.DAT`) to be considered in generation. This input should be immediately followed, with no intervening blank line, by `nfiles` filenames, each on a separate line, as in the following example:

```
nfiles   3
/home/user/dir1/phavegas01.dat
/home/user/dir1/phavegas02.dat
/home/user/dir2/phavegas01.dat
```

All phase-space grids for each selected process must be included for a meaningful generation. In the example at hand, the first two files from the top represent the two *grid-files* of the two channels corresponding to the same process stored in directory `dir1`. The last file contains the single channel grid of the process in directory `dir2`.

References

[1] N. Kauer, T. Plehn, D. Rainwater, D. Zeppenfeld, *$H \to WW$ as the discovery mode for a light Higgs boson*, *Phys. Lett.* **B 503** (2001) 113 [`hep-ph/0012351`].

[2] E. Accomando, A. Ballestrero and E. Maina, *PHASE - An event generator for six fermion physics at the LHC*, *Nucl. Instrum. Meth.* **534** (2004) 265 [`hep-ph/0404236`].

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

[4] F. Krauss, R. Kuhn and G. Soff, *AMEGIC++ 1.0: a matrix element generator in C++*, *JHEP* **02** (2002) 044 [`hep-ph/0109036`];
A. Schalicke, F. Krauss, R. Kuhn and G. Soff, *Implementing initial state radiation for lepton induced processes in AMEGIC++*, *JHEP* **12** (2002) 013 [`hep-ph/0203259`].

[5] E.E. Boos, M.N. Dubinin, V.A. Ilyin, A.E. Pukhov and V.I. Savrin, *Comphep: specialized package for automatic calculations of elementary particle decays and collisions*, `hep-ph/9503280`;
A. Pukhov et al., *Comphep: a package for evaluation of Feynman diagrams and integration over multi-particle phase space. user's manual for version 33*, `hep-ph/9908288`.

[6] Minami-Tateya collaboration, *Grace manual: automatic generation of tree amplitudes in standard models: version 1.0*, T. Ishikawa et.al. , KEK-92-19. Minami-Tateya collaboration, H. Tanaka et.al., *The Grace system for the minimal supersymmetric standard model*, *Nucl. Instrum. Meth.* **389** (1997) 295;
F. Yuasa et.al. , *Automatic computation of cross sections in HEP*, *Prog. Theor. Phys. Suppl.* **138** (2000) 18 [`hep-ph/0007053`];
S. Tsuno et al., *GR@PPA_4b: a four bottom quark production event generator for $pp/p\bar{p}$ collisions*, *Comput. Phys. Commun.* **151** (2003) 216 [`hep-ph/0204222`].

[7] F. Maltoni and T. Stelzer, *Madevent: automatic event generation with Madgraph*, *JHEP* **02** (2003) 027 [`hep-ph/0208156`];
T. Stelzer and W.F. Long, *Automatic generation of tree level helicity amplitudes*, *Comput. Phys. Commun.* **81** (1994) 357 [`hep-ph/9401258`];

JHEP07(2005)016

H. Murayama, I. Watanabe and K. Hagiwara, *HELAS: Helicity amplitude subroutines for Feynman diagram evaluations*, KEK-91-11.

[8] C.G. Papadopoulos, *Phegas: a phase space generator for automatic cross-section computation*, *Comput. Phys. Commun.* **137** (2001) 247 [hep-ph/0007335];
A. Kanaki and C.G. Papadopoulos, *Helac: a package to compute electroweak helicity amplitudes*, *Comput. Phys. Commun.* **132** (2000) 306 [hep-ph/0002082].

[9] M. Moretti, T. Ohl and J. Reuter, *O'Mega: an optimizing matrix element generator*, hep-ph/0102195;
W. Kilian, *Whizard manual*, in *2nd ECFA/DESY Study 1998-2001* 1924-1980, LC-TOOL-2001-039, Jan 2001.

[10] A. Ballestrero and E. Maina, *A new method for helicity calculations*, *Phys. Lett.* **B 350** (1995) 225 [hep-ph/9403244].

[11] F.A. Berends, R. Pittau and R. Kleiss, *EXCALIBUR: a Monte Carlo program to evaluate all four fermion processes at LEP-200 and beyond*, *Comput. Phys. Commun.* **85** (1995) 437 [hep-ph/9409326];
F.A. Berends, P.H. Daverveldt and R. Kleiss, *Complete lowest order calculations for four lepton final states in electron-positron collisions*, *Nucl. Phys.* **B 253** (1985) 441;
J. Hilgart, R. Kleiss and F. Le Diberder, *An electroweak Monte Carlo for four fermion production*, *Comput. Phys. Commun.* **75** (1993) 191.

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

[13] G.P. Lepage, *A new algorithm for adaptive multidimensional integration*, *Jour. Comp. Phys.* **27** (1978) 192.

[14] E. Accomando, A. Ballestrero and E. Maina, *WPHACT 2.0: a fully massive Monte Carlo generator for four fermion physics at e^+e^- colliders*, *Comput. Phys. Commun.* **150** (2003) 166 [hep-ph/0204052];
E. Accomando and A. Ballestrero, *WPHACT 1.0: a program for WW, Higgs and four-fermion physics at e^+e^- colliders*, *Comput. Phys. Commun.* **99** (1997) 270 [hep-ph/9607317].

[15] A. Ballestrero, *Phact: helicity amplitudes for present and future colliders*, hep-ph/9911318.

[16] E.N. Argyres et.al., *Stable calculations for unstable particles: restoring gauge invariance*, *Phys. Lett.* **B 358** (1995) 339 [hep-ph/9507216];
W. Beenakker et al., *The fermion-loop scheme for finite-width effects in e^+e^- annihilation into four fermions*, *Nucl. Phys.* **B 500** (1997) 255 [hep-ph/9612260];
U. Baur and D. Zeppenfeld, *Finite width effects and gauge invariance in radiative W productions and decay*, *Phys. Rev. Lett.* **75** (1995) 1002 [hep-ph/9503344];
G. Passarino, *Unstable particles and non-conserved currents: a generalization of the fermion-loop scheme*, *Nucl. Phys.* **B 574** (2000) 451 [hep-ph/9911482];*Single-W production and fermion-loop scheme: numerical results*, *Nucl. Phys.* **B 578** (2000) 3 [hep-ph/0001212];
E. Accomando, A. Ballestrero and E. Maina, *Non-conserved currents and gauge-restoring schemes in single W production*, *Phys. Lett.* **B 479** (2000) 209 [hep-ph/9911489].

[17] T. Ohl, *Vegas revisited: adaptive Monte Carlo integration beyond factorization*, *Comput. Phys. Commun.* **120** (1999) 13 [hep-ph/9806432].

[18] R. Bhattacharya, J. Smith, G. Grammer, Jr., *Two photon production processes at high-energy. I.*, *Phys. Rev.* **D 15** (1977) 3267.

[19] T. Sjöstrand, L. Lönnblad, S. Mrenna and P. Skands, *Pythia 6.3: physics and manual*, hep-ph/0308153;
T. Sjöstrand et al., *High-energy-physics event generation with Pythia 6.1*, *Comput. Phys. Commun.* **135** (2001) 238 [hep-ph/0010017];
T. Sjöstrand, *High-energy physics event generation with Pythia 5.7 and Jetset 7.4*, *Comput. Phys. Commun.* **82** (1994) 74.

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

[21] J. Pumplin et al., *New generation of parton distributions with uncertainties from global QCD analysis*, *JHEP* **07** (2002) 012 [hep-ph/0201195].

[22] D. Binosi and L. Theussl, *Jaxodraw: a graphical user interface for drawing Feynman diagrams*, *Comput. Phys. Commun.* **161** (2004) 76 [hep-ph/0309015];
J.A.M. Vermaseren, *Axodraw*, *Comput. Phys. Commun.* **83** (1994) 45.