December 2009

A proposal for a standard interface between Monte-Carlo tools and one-loop programs

T. Binoth [a,*] F. Boudjema [b] G. Dissertori [c] A. Denner [o]
S. Dittmaier [e] R. Frederix [d] W. Giele [m] T. Gleisberg [s]
N. Greiner [p] G. Heinrich [i] S. Hoeche [p] J. Huston [f] N. Kauer [k]
A. Lazopoulos [c] D. Maitre [i] F. Maltoni [d] C. Oleari [q]
G. Passarino [g] S. Pozzorini [h] R. Pittau [j] T. Reiter [ℓ]
S. Schumann [t] P. Skands [m] G. Zanderighi [n]
!!! Author list not final !!!

[a]*School of Physics and Astronomy, The University of Edinburgh, Edinburgh EH9 3JZ, Scotland, United Kingdom*

[b]*need address information for LAPTH Annecy*

[c]*Institute for Particle Physics, ETH Zurich, CH–8093 Zurich, Switzerland*

[d]*Centre for Particle Physics and Phenomenology (CP3), Université Catholique de Louvain, Chemin du Cyclotron 2, B–1348 Louvain-la-Neuve, Belgium*

[e]*Albert-Ludwigs-Universität Freiburg, Physikalisches Institut, D-79104 Freiburg, Germany*

[f]*Michigan State University, East Lansing, MI 48840, USA*

[g]*INFN, Sezione di Torino and Dip. Fisica Teorica, Universitá di Torino, via Giuria 1, I–10125 Torino, Italy*

[h]*CERN/PH, CH–1211 Geneva 23, Switzerland*

[i]*Institute for Particle Physics Phenomenology, University of Durham, South Rd, Durham DH1 3LE, United Kingdom*

[j]*need address information for University of Granada*

[k]*Department of Physics, Royal Holloway, University of London, Egham TW20 0EX, UK*

[ℓ]*Nikhef, Science Park 105, 1098 XG Amsterdam, The Netherlands*

[m]*FNAL, P.O. Box 500, Batavia, IL 60510, USA*

[n]**please provide contact details**

[o]*Paul Scherrer Institut, Würenlingen und Villigen, CH-5232 Villigen PSI, Switzerland*

[p]**please provide contact details**

First Draft 13 December 2009

[q]please provide contact details
[r]please provide contact details
[s]please provide contact details
[t]please provide contact details

Abstract

Many highly developed Monte-Carlo tools for the evaluation of cross sections based on tree matrix elements exist and are used by experimental collaborations in high energy physics. As the evaluation of one-loop matrix elements has recently been undergoing enormous progress the combination of one-loop matrix elements with existing Monte-Carlo tools is on the horizon. This would lead to phenomenological predictions at the next-to-leading order level. This note summarises the discussion of the "Next-to-leading order multi-leg" working group on this issue which has been taking place during the workshop on "Physics at TeV colliders" at Les Houches, France, in June 2009. The result is a proposal for a standard interface between Monte Carlo tools and one-loop matrix element programmes.

Key words: Monte Carlo tools, higher order computations, Les Houches Accord

1 Introduction

Monte Carlo event generators are of major importance in high energy physics as tools to simulate scattering processes defined within the Standard Model or one of its extensions. Probability distributions of events are determined from matrix elements which are defined in perturbation theory. At present there exists a plethora of highly developed Monte Carlo (MC) event generators which are based on tree matrix elements and which use sophisticated methods for matrix element generation and phase space integration [1,2]. In combination with parton showers and hadronisation models realistic event distributions can be obtained [3]. In predicting absolute rates these leading order predictions are plagued by uncertainties due to the missing higher order terms. Typical QCD uncertainties of N-jet observables scale like $\alpha_s(\mu_R)^N$ and are thus very sensitive to the renormalisation scale (μ_R) choice which can not be inferred from first principles. In the case of electroweak corrections higher order terms can become sizable at high energies, i.e. of the order of ten percent, due to large electroweak Sudakov logarithms. For precise predictions and simulations

[*] Corresponding author. *Email address*: `Thomas.Binoth@ed.ac.uk`.

of collider data it would be highly desirable to take such corrections into account.

The evaluation of one-loop matrix elements has seen major progress in the last few years. Most processes of the les Houches experimenters wishlist are evaluated meanwhile [4]. Until recently the bottleneck has been the numerically stable and efficient evaluation of these one-loop matrix elements. Unitarity based methods but also improvements of Feynman diagram based methods have now matured to the point where the evaluation of the one-loop matrix elements is no longer the stumbling block. As stated above the evaluation of tree-matrix elements is well under control and the corresponding MC tools can be used for the evaluation of the leading order (LO) and next-to-leading order (NLO) real emission contributions in full NLO computations. Together with methods to treat the inherent infra-red singularities everything is in place to combine tools of different communities and positive synergy effects can be expected from such an initiative.

Various groups which use different methods are meanwhile able to make NLO predictions for partonic processes with up to four final state particles [5]. Several groups aim for an automated evaluation of one-loop matrix elements or provide programme packages for certain classes of processes [6]. Thus a good coverage of next-to-leading order processes relevant for the LHC is in reach.

The incorporation of one-loop matrix element information into existing tree-level Monte Carlo tools is feasible due to the inherent modularity of higher order calculations as will be discussed below. Note that such a synthesis would speed up the production of NLO results considerably because the loop calculator would be dispensed from dealing with the tree-like parts of the computation and phase space integration issues. Of course numerical stability of the one-loop matrix element at least in a certain region of phase space (to be specified by the provider) has to be guaranteed. For the experimentalist such a coherent approach would allow to use familiar tools instead of dealing with many different stand-alone computer codes.

The members of the working group on next-to-leading order multi-leg processes have extensively and constructively discussed this issue during the Les Houches workshop in June 2009. This document summarises the discussion of the working group and provides a proposal for a standardised interface between Monte-Carlo tools and one-loop programmes.

A short outline of the structure of one-loop computations is provided in section two to set the stage for the proposal of the interface in section three. Section four contains details on how to implement such an interface and in section five an extension to electroweak corrections is discussed. The article closes with a summary and gives an outlook on further extensions of the proposed Les Houches accord.

2 Modular structure of one-loop computations

Higher order corrections to scattering processes require the evaluation of virtual and real emission corrections for a given process. These corrections have to be provided for all contributing partonic cross sections. In hadronic collisions the hadronic cross section is given by

$$\sigma_{\text{had}}(p_1,p_2) = \sum_{a,b} \int_0^1 dx_1 f_{a/H_1}(x_1,\mu_F^2) \int_0^1 dx_2 f_{b/H_2}(x_2,\mu_F^2) \times \Big[d\sigma_{ab}^{LO}(x_1p_1,x_2p_2;\mu_R^2) + d\sigma_{ab}^{NLO}(x_1p_1,x_2p_2;\mu_R^2,\mu_F^2)\Big] \quad , \tag{1}$$

where a,b label the partonic subprocesses. Note that in the case of electroweak corrections photons have to be allowed in the initial state at next-to-leading order in α. The partonic contributions consist of the Born term (B), real (R) and virtual (V) corrections and collinear counterterms (C). For observables with m particles in the final state the real emission term contains one extra particle in the final state.

$$\sigma_{ab}^{LO} = \int_m d\sigma_{ab}^{B} \tag{2}$$

$$\sigma_{ab}^{NLO} = \int_{m+1} d\sigma_{ab}^{R} + \int_m d\sigma_{ab}^{V} + \int_m d\sigma_{ab}^{C}(\mu_F^2, F.S.) \tag{3}$$

The collinear counterterm, $d\sigma_{ab}^{C}(\mu_F^2, F.S.)$, is induced through a perturbative reparametrisation of the parton distribution functions and cancels the collinear initial state initial state singularities of the real emission contribution. It is actually a convolution of the Born term with a splitting function which depends on a factorisation scheme (F.S.) and introduces a factorisation scale (μ_F) dependence. In the case of fragmentation processes a similar treatment for final state hadrons and/or photons apply and the fragmentation scheme has to be specified.

The Born, real emission and collinear terms are defined by $2 \to m$ and $2 \to m+1$ tree amplitudes, which can be efficiently evaluated with existing matrix element generators. The virtual contribution is defined by the interference term, $\mathcal{I}(\{k_j\})$, between the Born and one-loop amplitude,

$$d\sigma^V_{ab} = d\text{LIPS}(\{k_j\})\, \mathcal{I}(\{k_j\}) \quad . \tag{4}$$

Here $\{k_j\}$ with $j = 1, \ldots, m+2$ indicate the partonic momenta. The interference term is defined by the tree- and one-loop contribution of the given process. These parts share a common colour basis, $|c\rangle$, which defines the corresponding colour correlation matrix, $\langle c|c'\rangle$.

$$\mathcal{I}(\{k_j\}) = \sum_{h,c,c'} \Big(\mathcal{A}^{LO\dagger}_{h,c} \langle c|c'\rangle \mathcal{A}^{NLO,V}_{h,c'} + \mathcal{A}^{NLO,V\dagger}_{h,c} \langle c|c'\rangle \mathcal{A}^{LO}_{h,c'} \Big) \quad .$$

The indices h and c enumerate helicity and colour degrees of freedom. Different representations of the underlying amplitudes are possible. For example, specifying a helicity and colour basis may allow for Monte-Carlo sampling over different components of the cross section contributions. On the other hand the summation over colour and helicity leads to a simple, basis-independent object. In some cases it might be sufficient to provide approximations for the loop amplitude, e.g. the leading colour term(s). The one-loop amplitude contains ultra-violet (UV) and infra-red (IR) divergences. Dimensional regularisation converts these divergences into poles in $1/\epsilon$ where typically the convention $n = 4 - 2\epsilon$ is adopted for the space-time dimensionality. Applying a UV renormalisation scheme removes the UV poles from the loop amplitude and one is left with IR singularities. Those are very often treated by using dimensional regularisation as well which converts them into IR $1/\epsilon$ poles but of course other methods are applicable, too. For example, mass regulators are often used in QED and electroweak computations as will be discussed below. Any IR regularisation scheme (R.S.) defines uniquely the finite remainders. Commonly used schemes differ by the dimensional treatment of internal and external particles. Standard examples are conventional dimensional regularisation (`DREG`), dimensional reduction (`DRED`) and the 't Hooft-Veltman scheme (`tHV`) [8]. Schemes where the external (observed) particles are 4-dimensional are preferable from the point of view of helicity amplitudes. `CDR` and `tHV` have the same dimensional treatment concerning the soft/collinear particles. For a discussion of different schemes, see e.g. [7–10]. In the context of electroweak one-loop corrections different computational schemes are used to take into account generic loop contributions and the treatment of unstable particles. This will be extensively discussed below.

Any standardised treatment of these issues should allow for maximal flexibility to choose an adequate method and scheme for a computation. The only general assumption we adopt in our proposal is that the UV renormalisation is done internally by the provider of the one-loop contribution and that the IR singularities are treated uniformly such that interfacing with simple information passing is possible.

Due to its wide acceptance we will describe here the use of dimensional regularisation for IR divergences in more detail. In the section on EW corrections we will discuss how regulator masses can be treated with the same general interface. If dimensional regularisation is applied one is left with IR poles only after the UV renormalisation. The general structure of the virtual correction is thus given as

$$\mathcal{I}(\{k_j\}, R.S., \mu_R^2, \alpha_s(\mu_R^2), \alpha, \ldots) = C(\epsilon)\left(\frac{A_2}{\epsilon^2} + \frac{A_1}{\epsilon} + A_0\right) \quad . \tag{5}$$

The Laurent coefficients A_j are real-valued. Apart from the regularisation scheme and the renormalisation scale μ_R they depend of course on all relevant coupling and mass parameters of the underlying model which is indicated by the dots in the argument list of Eq. (5). A choice for the overall constant which is used by several authors is

$$C(\epsilon) = \frac{(4\pi)^\epsilon}{\Gamma(1-\epsilon)} = (4\pi)^\epsilon \frac{\Gamma(1+\epsilon)\Gamma(1-\epsilon)^2}{\Gamma(1-2\epsilon)} + \mathcal{O}(\epsilon^3)\ . \tag{6}$$

As such it seems natural to propose this factor as a general convention. Note that we identify here the dimensional regularisation scale μ with the renormalisation scale μ_R. Other choices will not lead to any problem as long as the information is transparent. The remaining infrared poles cancel when the virtual part is combined with the real emission corrections after integration over soft/collinear phase space regions. In actual calculations subtraction methods are applied which define local subtraction terms for the real emission amplitude and add back the integrated version such that IR divergences cancel with the m and $m+1$ particle contributions separately. A prominent example is the widely used Catani-Seymour dipole subtraction method [17–19], which has been implemented in computer programmes by several groups [26–32].

The general modular structure of a one-loop computation can be summarised in a simple diagram, see Fig. 1.

There are cases where a partonic process is not allowed at tree level but is initiated at the one-loop level only. In that case Fig. 1 contains only a one-loop module, the squared one-loop amplitude $|\mathcal{A}^{LO}_{\mathrm{One-Loop}}|^2$, instead of the interference term.

After this reminder of the general structure of one-loop computations we are in the position to discuss our proposal for an interface between Monte-Carlo event generators based on tree matrix elements and implementations of one-loop amplitudes.

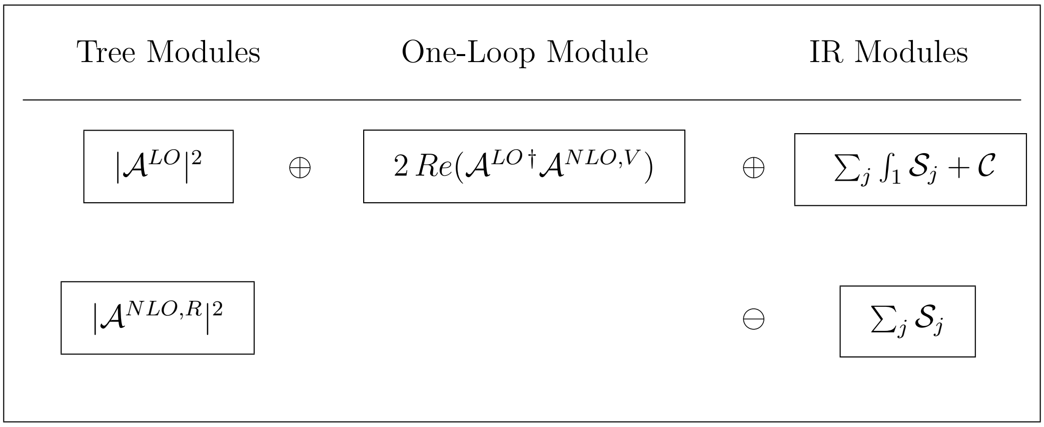
Tree Modules
One-Loop Module
IR Modules
$|\mathcal{A}^{LO}|^2$
$\oplus$
$2\,Re(\mathcal{A}^{LO\,\dagger}\mathcal{A}^{NLO,V})$
$\oplus$
$\sum_j \int_1 \mathcal{S}_j + \mathcal{C}$
$|\mathcal{A}^{NLO,R}|^2$
$\ominus$
$\sum_j \mathcal{S}_j$

Fig. 1. Modular structure of next-to-leading order computations for partonic processes. All structures related to tree amplitudes can be evaluated using LO MC tools. The one-loop module contains the UV renormalised interference term. The treatment of IR subtraction should be kept separate to allow for flexibility. The IR modules contain subtraction terms, $\mathcal{S}_j$, for the real emission part and their integrated variants which compensate IR divergences in the One-Loop Module. In case of collinear divergences of partonic initial states also collinear subtraction terms, $\mathcal{C}$, have to be provided. Subsequently, the contributions in each horizontal line are independently finite after summation.

3 A computational model for an interface

The goal of the interface is to facilitate the transfer of information between one-loop programs, abbreviated as OLP in the following, and programs which provide tree amplitude information and incorporate methods to do the integration over phase space. We call the latter simply Monte-Carlo tool (MC) in the following. It has to be stressed that any agreement on a common interface should avoid to constrain a provider of an OLP. The idea is to provide a platform to exchange information between an OLP and a MC in a compatible way. The information which has to be passed is already defined in section 2. Note that the MC and the OLP may work in very different ways. Some programs might either evaluate one-loop amplitudes or produce amplitude code on the fly as requested by a user, while others may consist of a library of hard coded matrix elements for a certain number of processes.

This is why we would like to propose a computational model to fix the interaction between an OLP and a MC, which assumes two phases which we will call the `initialisation` and the `run-time` phase.

`Initialisation:` During this phase the MC communicates basic information to the OLP. This means it will ask for the availability of partonic one-loop sub-processes, set up the input parameters and fix options that are offered by the OLP.

`Run-time`: During this phase the MC queries the OLP for the value of phase-space dependent one-loop contributions and additional information. The latter should also contain a return value for the squared LO amplitude which is useful for checking purposes and gives flexibility to apply different renormalisation/factorisation schemes. The most relevant information is the finite part of the interference term. It may be split into several contributions to allow for more efficient Monte-Carlo integration sampling.

We propose to pass the in/output information in the `initialisation` phase through a file and during the `run-time` phase by function calls. This will be explained in more detail below. We propose to pass model parameters by use of the Les Houches accord file with its specific format [34] as it is widely used in the community. Schematically the input/output information is collected in Tab. 1. A few comments are in order.

OLP	Input	Output
`Initialisation`	Model parameters: $\alpha(0), \alpha_s(M_Z), \ldots$, $m_t, m_b, \ldots$, CKM values	`confirm values`
	Schemes: UV-renormalisation / IR-factorisation	`confirm schemes`
	Operational information: colour/helicity treatment, approximations, etc.	`confirm options`
`Run-time`	Events: $(E, p_x, p_y, p_z, M)_{j=1,\ldots,m+2}, \mu, \alpha_s(\mu)$	$(A_2, A_1, A_0, \lvert\mathrm{Born}\rvert^2)$ `optional information`

Table 1
The basic input/output information of a one-loop program. In the initialisation phase communication is preferably achieved through files whereas in the run-time phase the information is passed by function calls.

- Input parameters should generally be provided by the MC during the initialisation phase. We recommend that for electroweak calculations the fine-structure constant is passed by the MC as $\alpha(0)$. EW computations might entail modifications here as discussed in section 5.
- The IR regularisation schemes can be labelled by keywords like `DREG`, `DRED`, `tHV`, etc.
- Any approximation which is used in the OLP has to be transparently defined and implemented such that it can be flagged during initialisation. Examples are the use of massless approximations for certain quarks or putting the CKM matrix to unity.
- The minimal one-loop information to be returned by the OLP is the colour

and helicity summed interference term (at least in some approximation). Additional information can be returned, too. It is the responsibility of the OLP author to define the different additional return values/options of his code in the documentation.

- Further keywords maybe used for the treatment of massive or unstable particles or for EW schemes, see section 5.

4 Proposal for the implementation of the MC/OLP interface

This section outlines a proposal for the implementation of the interface model presented above. Of course any proposal should allow for easy implementation in all common programming languages.

In the easiest case of merging two programs written in a compatible language, the connection between the caller (the MC) and the receiver (the OLP) can be achieved by linking the program together. In more difficult cases with mixed languages or special environments a bridge program or transmission protocol will have to be put in place. This is however not the topic or concern of this proposal. We only want to ensure that the in/output values are defined in a mutually compatible way.

In the initialisation phase communication between the MC and the OLP will happen such that availability or generation of amplitude information of specified processes including necessary settings and options is communicated. After successful initialisation the codes are prepared to pass actual amplitude information for cross section evaluations in the `run-time` phase. We propose to pass loop amplitude information by function calls located in the MC and executed by the OLP. How the function is actually implemented is left to the OLP but we propose to use the generic name `OLP_EvalSubProcess` and to fix the arguments and return values following Tab. 1 as specified below.

In detail we propose the following set-up for the implementation of the two interaction phases of the MC with the OLP.

Initialisation phase

In this phase the MC creates a file that contains the information about the subprocesses it will need to perform the computation. By a subprocess we understand either a partonic subprocess or a component thereof. To introduce such components is useful, if for example detailed colour or helicity information is provided by the OLP such that sampling methods can be applied. The subprocesses are represented by integer numbers which are allocated in the initialisation phase. We will call the generated file the `order`. The choice to transfer the set-up parameters through files allows to invoke the initialisation of the MC and the OLP without having to link the two programs together. The order file is then read by the OLP and it creates a new file with the agreement about how to link the MC and OLP at run-time. We will call this file the `contract`. These two files are discussed in more detail below.

The `order` *file*

The order file contains the information needed for the OLP to declare whether or not it can provide the subprocess the MC wants. It should be allowed, e.g. to ask for colour and helicity summed amplitudes. The first part of the file contains some general settings as defined in the `initialisation` part of Tab. 1.

The settings that an OLP has to provide concern the colour and helicity information and the treatment of divergences.

`MatrixElementSquareType:` the type of the returned amplitude information. This flag can be used to distinguish colour (C) and helicity (H) treatment. Possible values could be `CHsummed`, `Csummed`, `Hsummed`, `NOTsummed`. Note that helicity/colour basis information has to be provided if summations are not performed. Similar flags for colour/helicity averaged expressions like `CHaveraged`, etc. might also be used.

`CorrectionType:` the type of higher order correction should be specified. Standard keywords are `QCD`, `EW` or `QED`.

`IRregularisation:` the IR regularisation scheme used. Common choices for QCD are `DREG`, `DRED`, `tHV`, as explained in the text are. The OLP provider might define his scheme of choice, as long as it is clearly documented.

Further settings which are related to EW processes are discussed in section 5. In the context of EW computations mass regulators (`MassReg`) or mixed approaches are common to handle IR divergences. Note that it is assumed that the OLP produces UV renormalised output such that the MC user should not have to deal with that. One has to keep in mind though that wave function renormalisation constants do contain IR contributions when external particles are put on-shell and that IR subtraction schemes in general make assumptions on the treatment of these wave function contributions. This fact is disguised by the use of dimensional regularisation for UV and IR divergences. Thus it might be useful to allow for flags which indicate the used prescription.

`MassiveParticleScheme` : A standard choice is `OnShell`.

As most OLP authors use the cancellation of IR divergences as a check an IR subtracted result might also be returned by the OLP in addition to the unsubtracted output. In this case a flag and the program documentation should indicate which IR subtraction scheme has been used.

`IRsubtractionMethod:` specifies if an IR subtracted result like `DipoleSubtraction`, `FKSSubtraction`, `AntennaSubtraction`, etc. applied to the one-loop result is available, too. The default should be `None`.

Although the IR structure of one-loop amplitudes is universal we do not recommend to provide only subtracted information, because details of IR subtraction methods have to be passed then between the MC, the OLP and the IR subtraction provider which spoils the modularity of the whole set-up.

In addition, as pointed out above, other specifications have to be added: Most importantly the model parameters like couplings, mass and width of particles. Depending on the process one has to define the treatment of unstable particles, number of active flavours, and etc.

Settings which concern parameter, technical and operational information are

`ModelFile:` the model file from which parameters have to be read. We propose to use files in the Les Houches accord format, `*.slh` for passing model parameters [34].

`OperationMode:` the operating mode of the OLP. This optional flag can be used to specify OLP defined approximations to the one-loop contribution, e.g. `LeadingColour`, `ReggeLimit`, etc.

`SubdivideSubprocess:` This flag tells the user if a given process is represented in a split form to allow for multi-channel Monte-Carlo sampling.

The OLP can provide defaults for its settings such that the `ModelFile` and `OperationMode` options are not mandatory. In this case the settings do not need to be present in the order file. The flag `SubdivideSubprocess` is proposed to allow for flexibility in the evaluation of the one-loop contribution. It defines whether the OLP returns the amplitude information cut into parts which have different Monte-Carlo weights when integrated over phase space. Sampling over various contributions like colour, helicity, channels, etc. improves the efficiency of integrators considerably. It is at the discretion of the OLP to provide such a feature.

The second part of the order file contains the list of the partonic subprocesses the MC needs. Here each line is of the form

```
[PDGcode1]             -> [PDGcode2] ... [PDGcodeM]
[PDGcode1] [PDGcode2] -> [PDGcode3] ... [PDGcodeM]
```

The first line describes a 1 → $M - 1$ decay and the second a 2 → $M - 2$ scattering process. [PDGcodeJ] is the particle data group code of particle J as defined in [33]. The initial and final state particles do in general not fix the powers of the coupling constants. If this is ambiguous it is useful to specify this flags like

`AlphasPower`: integer which specifies the α_s power of the Born cross section.

`AlphaPower`: integer which specifies the α power of the Born cross section.

Some OLP providers leave the evaluation of couplings entirely to the MC and provide only kinematical information where the coupling parameters are stripped off. This is useful, as no compatibility issues arise in this case. We propose to indicate this by an `OperationMode` flag: `CouplingsStrippedOff` which indicates that overall coupling terms $\sim \alpha^I \alpha_s^J$ are treated as equal to one by the OLP. It is then the responsibility of the MC to multiply the OLP information by the correct coupling terms which are defined by `AlphasPower`, `AlphaPower` and `CorrectionType`.

It is recommended that the OLP adds these default settings to the `contract` file. The OLP can define new settings. If a setting is mandatory but has not been provided, the OLP should issue an error message and write only this error message in the contract file.

Here is an example of an order file for the partonic 2 → 3 processes, $gg \to t\bar{t}g$, $q\bar{q} \to t\bar{t}g$ and $qg \to t\bar{t}q$ needed for the evaluation of $pp \to t\bar{t} + jet$:

```
# example order file

MatrixElementSquareType  CHsummed
IRregularisation         DREG
OperationMode            LeadingColour
ModelFile                ModelInLHFormat.slh
SubdivideSubprocess      yes
AlphasPower              3
CorrectionType           QCD

# g g  -> t tbar g
 21 21 -> 6 -6 21
# u ubar -> t tbar g
  1 -1 -> 6 -6 21
# u g  -> t tbar u
  1 21 -> 6 -6  1
```

In this example three partonic subprocesses are listed. The `AlphasPower=3` indicates that the Born process is of order $\mathcal{O}(\alpha_s^3)$ which implies that only QCD interactions are considered, else Z-boson/photon exchange would not be restricted in $q\bar{q} \to t\bar{t}g$. The `CorrectionType` indicates that the MC asks for the virtual $\mathcal{O}(\alpha_s^4)$ contribution to the partonic processes.

The flag `SubdivideSubprocess` indicates that the MC asks if a split representation is provided. The fact that not all flags occur in this file indicates that the MC user has read the OLP instructions and knows the default. As this is of course the exception, preferably all flags should be present together with a corresponding flag. Lines starting with a # character are treated as comments. Comments are encouraged to increase the human-readability of the file.

The `contract` *file*

In the initialisation phase the OLP reads the order file and creates its "answer" in the form of a contract file. The contract file is basically a copy of the order file, but with information from the OLP appended at the end of each (non content-free) line. We propose the following format:

The OLP answer is separated from the MC request by a "|" character. These answers are explained in the following paragraphs. For settings, the answer should be appended as follows: `OK` if the setting is supported and the supplied arguments are correct. Errors might be indicated as follows: `Error: unknown option` if the option/setting is not recognised, `Error: unknown flag` if the option is supported but the keyword/flag is not recognised/valid. If a `ModelFile` is not found the error message is: `Error: file not found`.

For subprocesses, the answer should be the number of separate contributions that the OLP can compute which add up to the required subprocess, followed by the integer labels which identify each of these contributions. This allows the multi-channel integrator (if present) to take advantage of importance difference between the provided contributions. The integer label will be used at run-time to identify the subprocesses. An OLP which provides this kind of separation might also provide a single subprocess that is the sum of all contributions. Whether or not to return the contributions separately could be set using the setting `SubdivideSubprocess`.

Here is an example for a contract file where all options and settings are available and confirmed by the OLP (&& indicates line continuation):

```
# example contract file
# contract produced by OLP, OLP authors, citation policy

MatrixElementSquareType CHsummed                | OK
IRregularisation        DREG                    | OK
OperationMode           LeadingColour           | OK
ModelFile               ModelFileInLHFormat.slh | OK
SubdivideSubprocess     yes                     | OK
CorrectionType          QCD                     | OK

# g g -> t tbar g
21 21 -> 6 -6 21       | 2 13 35 # channels: cut-constructable,&
                                        & rational part
# u ubar -> t tbar g
 1 -1 -> 6 -6 21       | 1 29
# u g -> t tbar u
 1 21 -> 6 -6  1       | 3 8 23 57 # 3 channels: leading,&
                         & subleading, subsubleading colour
```

If a partonic process is split into several contributions a description of these different subprocesses is recommended as indicated in this example.

It would be very useful if the OLP could provide some help in case of failure, e.g. listing the parameters which are accepted. All default parameters, which were not specified in the order file, but are relevant for the evaluation should be written down by the OLP in the contract file.

Here is an example for a contract file where options and settings are not available:

```
# example contract file
# contract produced by OLP

MatrixElementSquareType CHsummed | Error: unsupported flag
# CHaveraged is supported
IRregularisation        DRED     | Error: unsupported flag
# DREG, tHV are supported
OperationMode      LeadingColour | Error: unsupported flag
# see OLP README
ModelFile     FavouriteModel.slh | Error: file not found
# Modelfile is called: SM.slh
SubdivideSubprocess yes          | Error: unsupported flag
# no is supported
CorrectionType           EW      | Error: unsupported flag
# QCD is supported
MyWayOfDoingThings true          | Error: unknown option

# g g -> t tbar g
21 21 -> 6 -6 21    |   Error: massive quarks not supported
# u ubar -> t tbar g
 1 -1 -> 6 -6 21    |   Error: process not available
# u g -> t tbar u
 1 21 ->> 6 -6  1   |   Error: check syntax
```

It would be useful that all assumptions and settings which are made and are available in an OLP are added as comments and/or as warning/error messages. The latter might be appended in the response line. The detailed standardisation of error/warning handling is beyond the scope of this document. A minimal strategy for the OLP is to use only `OK`, `ERROR` as identifiers and to encourage the MC user to read the OLP documentation. Preferably MC users should be able to correct their order file settings from the error/warning information and the comments provided in the contract file.

Run-time phase

To start the run-time phase, the OLP needs to be reminded of the contract it committed to. This is especially the case for programs which dynamically generate their evaluation structure as opposed to libraries of generated code. To initialise the OLP for the run, we propose to use a function `OLP_Start` which has to be called by the MC. After this the MC can start querying the OLP for the subprocesses. This is done using the function `OLP_EvalSubProcess`. In more detail these two functions are defined by:

OLP_Start

The function OLP_Start should receive a character string that contains the name of the contract file as an argument. The function returns 1 if the initialisation is successful, 0 if not.

OLP_EvalSubProcess

The parameters to be passed to the OLP_EvalSubProcess function are (in this order):

- the integer label of the subprocess
- an array containing the components of the momenta. The momenta are placed in a one dimensional array. We propose to use physical scattering kinematics which differentiates between in and outgoing momenta, so that $p_1 + p_2 = p_3 + \ldots p_m$. For each particle the kinematics is specified by a 5-tuple: $E_j, p^x_j, p^y_j, p^z_j, M_j$. A full event is specified by an array of $m * 5$ double precision numbers filled with the m 5-tuples ordered by the particle enumeration.
- the renormalisation scale, μ_R, as a double precision number, or an array of scales, if different scales need to be passed.
- the strong coupling[1] $\alpha_s(\mu_R)$, or an array of phase space dependent couplings and/or variables.
- the array where the result is to be put.

We propose that all dimensional objects use as standard unit GeV with the implicit assumption that only the energy unit has to be defined, i.e. the standard choice $\hbar = c = 1$ is understood. The returned array is expected to contain at least four real-valued double precision numbers

```
PoleCoeff2, PoleCeoff1, PoleCoeff0, BornSquare
```

which correspond to the colour and helicity summed/averaged terms A_2, A_1, A_0, $|\mathrm{Born}|^2$ as defined in Tab. 1 and the settings in the contract file. If the OLP provider wants to hand over more colour and/or helicity information, these variables can be promoted to arrays of double precision numbers which contain the corresponding information. Care has to be taken, that colour and helicity conventions match.

It is recommended to use the OLP value(s) for $|\mathrm{Born}|^2$ to perform a consistency check between the OLP and MC evaluation of this quantity, as the

[1] Note that $\alpha_s(\mu_R) = 1$ can be used to indicate that the MC multiplies the returned vales with the adequate coupling constants.

compatibility of settings, parameters, and the colour and helicity basis is efficiently tested in this way. The OLP provider might want to hand over additional information, e.g. arrays containing other possible decompositions of the result. All these features can also be used for importance sampling over different subexpressions of the subprocesses. The length and the content of the returned array is to be documented by the OLP.

As most OLP authors do check the cancellation of IR poles as a check of their calculation, a value for the IR subtracted interference term, A_0^{sub}, might be passed in addition. A flag and/or the program documentation should indicate which IR subtraction scheme has been used.

5 Extending the interface to include EW OLP

This section outlines a proposal for an extension of the interface model to include NLO EW corrections. More communication between the MC and the OLP is necessary because renormalisation and the treatment of unstable particles is more complicated than in the QCD case. The role of input parameters in electroweak computations in relation with a computational scheme has to be carefully understood when combining output from various sources. This implies that the exchange of information between an OLP and a MC is, by far, more complex in case EW NLO corrections are included and necessarily constrain both contractors; for instance, knowledge of the Born amplitude alone is not enough to construct the divergent parts of the real corrections (provided by the MC) fully consistent with the virtual ones (provided by the OLP). In order to guarantee cancellation of soft and/or collinear IR divergences and proper re-weighting of all real NLO corrections more information has to be provided.

In the following we describe the minimal amount of information that should be exchanged during negotiation, taking into account that the interface should not constrain the OLP in a substantial way.

Treatment of resonances

The calculation of EW corrections requires to fix a scheme for the treatment of resonances. The presently preferred scheme is the complex-mass scheme which was introduced at tree level in Ref. [11] and extended to one loop in Ref. [12]. It is gauge invariant and straightforward to implement in the sense that it follows the usual perturbative calculus of local quantum field theories, i.e. Feynman rules, counterterms, and so on. It relies however on the use of complex masses and couplings also in the LO matrix element. To efficiently interrelate MC tools with EW OLPs, we strongly suggest that LO MCs support this possibility. Nevertheless we allow for other schemes like the naive fixed-width scheme and the pole scheme (see, e.g., Refs. [13–16]). Note, however, that in this case more information has to be exchanged between the OLP and the LO MC, like the used gauge for the fixed-width scheme[2] or information on the included diagrams in case of the pole scheme. This in general requires, that the user studies the description (e.g. documented in a README file) of the OLP.

The basic communication happens via the flag `ResonanceTreatment`.

`ResonanceTreatment:` This flag defines the treatment of resonances. Standard values are `ComplexMassScheme`, `FixedWidthScheme`, `PoleScheme` or an OLP-defined variant thereof.

If the OLP does not support the requested scheme, it should return a list of supported schemes in the `contract` file.

Electroweak regularisation scheme

In EW computations the use of mass regulators for collinear and/or soft divergences are common. The flag `IRregularisation` which is defined above should be used to communicate with the OLP. A prominent approach is to use fermion masses to regulate collinear divergences but use dimensional regularisation for the soft divergences. Allowing for several keywords for the IR regularisation flag, all cases can be covered. The MC supporting `MassReg` has to provide the information which masses are to be considered small, and specific details how regulator masses have to be applied in the various parts of the full computation.

The entry in the order file may look like

```
IRregularisation  MassRegColDiv DimRegSoftDiv
IRregulatorMasses MD,MU,MS,ME
```

or

```
IRregularisation  MassReg
IRregulatorMasses MD,MU,MS,MC,MB,ME,MMU,MTAU,MPHOTON
```

Mass regularisation is, for instance, relevant for processes with isolated muons

[2] Strictly speaking the fixed-width scheme does not respect gauge invariance but for practical purposes the gauge dependence often turns out to be numerically insignificant. Note that additional information on the gauge choice has to be provided.

in the final state[3].

If regulator masses are used the cut-off dependence of the loop amplitudes is much less transparent as in the case of dimensional regularisation. The flag `IRsubtraction`, introduced above, may be used to indicate that a IR subtracted result is also provided. If the functionality is not sufficient, e.g. if QCD and EW corrections are treated by the same OLP, special additional flags like `EWIRregularisation` and `EWIRsubtraction` might be useful with similar keywords as the QCD counterparts but supplemented by the "EW" prefix.

The OLP directly subtracts the collinear and soft divergences, for example via the "endpoint contributions", quantified for example by the $\mathbf{I}(\epsilon)$ operator of Catani and Seymour in dipole subtraction for QCD [17–19] or QED [20,21] corrections. While this introduces a dependence on the subtraction scheme, it renders the sum of the virtual corrections and the endpoint of the real corrections independent of any regularisation. As a consequence, the LO MC needs less information and can in particular calculate the real corrections without any dependence on the IR regularisation scheme.

Electroweak renormalisation scheme

The next point is the agreement on the renormalisation scheme which in essence defines relations between electroweak couplings and parameters beyond leading order. Electroweak radiative corrections are absorbed differently in renormalised couplings in various schemes. The default, supported by any OLP should be the $\alpha(0)$ scheme for real [22–24] or complex [12] on-shell masses and electroweak couplings, where the weak mixing angle is defined via $\cos\theta_W = M_W/M_Z$. In this respect it is essential that OLP subprocesses have identifiable overall coupling factors like $\alpha^N\,\alpha_s^M$ which is guaranteed by the correct use of the flag `PowerAlphas` together with a similar flag for α, `PowerAlpha`.

In the simplest case, the MC calculates $\mid \mathcal{A}_{LO} \mid^2$ with only $\alpha(0)$ as EW coupling. The OLP might use another value for α such that the two Born evaluations will not match. The ratio can be used to infer an effective coupling

[3] Note that dimensional regularisation for QED-like soft divergences is fully equivalent to the regularisation by an infinitesimal photon mass `MPHOTON` in the sense that there is, up to irrelevant terms of order $\mathcal{O}(\epsilon)$, a one-to-one correspondence between the logarithmic term $\log \texttt{MPHOTON}^2$ and the term $\Gamma(1+\epsilon)(4\pi\mu^2)^\epsilon/\epsilon$ without additional finite terms.

α_{OLP} which is defined by

$$\alpha_{\mathrm{OLP}} = \alpha(0) \left(\frac{\mid \mathcal{A}_{LO}^{\mathrm{OLP}} \mid^2}{\mid \mathcal{A}_{LO}^{\mathrm{MC}} \mid^2} \right)^{1/P} \tag{7}$$

where P is the value of the `PowerAlpha` flag. This defines the value for α which is consistent with the OLP. A safer way would be to pass α_{OLP} as an additional parameter during run-time. In this way the MC could still perform a consistency check between the Born cross sections after adjusting the coupling parameters.

In general, e.g. if LO MC and OLP do not use the same masses or $\sin^2\theta_W$, more information has to be exchanged. The minimal information needed by the MC includes the LO matrix element as computed by the OLP and the quantities appearing in the IR modules. Then the MC has all elements to cancel the IR singularities properly. The IR modules basically depend on the electromagnetic coupling and the regularisation scheme. The LO matrix element can in principle be taken from the OLP, LO(OLP). In practice, however, the MC should be able to calculate the LO matrix element independently. To this end, the MC needs all parameters entering the LO(OLP) matrix element.

Therefore, the OLP specifies all parameters needed to calculate the LO matrix element, $\mathcal{A}_{LO}^{\mathrm{OLP}}$: couplings, masses, $\sin^2\theta_W$, etc. Masses for unstable particles are always understood as defined from complex poles, i.e. including real and imaginary parts, $M_j - i\,\Gamma_j$, but no running widths.

While the $\alpha(0)$ scheme should be the default for computing $\mathcal{A}_{LO}$, we should allow for other schemes to keep the interface more general and to improve the accuracy of the predictions. Various electroweak renormalisation schemes are discussed in the literature. The $\alpha(M_Z^2)$ scheme and the G_F scheme are defined, for instance, in Ref. [25]. Similarly $\alpha(\mu^2)$ with some scale μ or other definitions such as $\overline{\mathrm{MS}}$ could be used. Moreover, the OLP could also support other schemes where different couplings are used in the same amplitude, like $\alpha(0)$ and the Fermi constant, G_F.

`EWRenormalisationScheme:` used schemes (discussed in the text) can be flagged by the keywords `alpha0` (default), `alphaMZ`, `alphaGF`, `alphaRUN`, `alphaMSbar`, `OLPdefined`.

If the electromagnetic coupling is defined by the OLP (`OLPdefined`), the OLP returns an effective α_{OLP} (`alphaOLP`) to be used in the LO matrix element. To give an example consider the process $qq \to W\gamma$. Here the effective α would result from $\alpha_{\mathrm{OLP}}^2 = \alpha(0)\alpha(G_F)$, where $\alpha(G_F)$ is α derived from the Fermi coupling constant. In some approaches, where the QED corrections are separated from the pure EW part of the calculation, an α_{IR} (`alphaIR`) is introduced which have to be used in the IR modules multiplying LO matrix elements. In addition to the effective α_{OLP} that is used already in LO α_{IR} is the EW coupling of the photonic correction only. The coupling parameters α_{OLP} and α_{IR} should be returned by the OLP during run-time together with the amplitude information. After ensuring cancellation of IR singularities, the MC can use or not use α_{OLP} for real corrections, QCD contributions etc.

After this exchange of information the MC knows how to cancel IR singularities and can proceed to calculate the remaining part of the real corrections. Clearly, the main issue is to transfer all the knowledge needed by the MC for the calculation of this part of the NLO correction. This knowledge, in particular, includes the issues of complex masses, potentially complex couplings etc. Therefore, it is essential that the OLP is returning not only the complex masses, but also explicitly the needed values of the couplings, which can be either complex or real, according to the OLP choice.

Run-time phase

During `run-time` the exchange of information proceeds as follows:

- The MC calculates $\mid \mathcal{A}_{LO}^{\mathrm{MC}} \mid^2$ for a given process according to its internal conventions (e.g. $\alpha(0)$ as EW coupling).
- The OLP computes $\mid \mathcal{A}_{LO}^{\mathrm{OLP}} \mid^2$ with an internal choice of EW couplings and parameters and the interference term $2\,\mathrm{Re}[\mathcal{A}_{LO}^{\mathrm{OLP}}]^\dagger\, \mathcal{A}_{NLO}^{\mathrm{OLP}}$. It returns these values according to the definitions of Tab. 1, i.e. the EW correction split up into coefficients of poles in ϵ.
- The OLP returns an effective α_{OLP} (`alphaOLP`) and if required an α_{IR} (`alphaIR`). These values may change event by event if variable scales are used for their definition.
- The MC uses α_{OLP} for the LO matrix element and α_{IR} for the IR modules. Alternatively, it can determine α_{OLP} from the evaluation of the Born amplitudes by the OLP and the MC.

Note that in the case of mass regularisation the coefficient of the double pole in ϵ is zero.

As the correct use of coupling parameters in EW computations is a complicated task, the MC should have the functionality to evaluate all necessary amplitudes in a "coupling stripped" mode, such that the couplings can be imported by the OLP during run-time. Sometimes it might be useful for the MC to use the Born term which is returned by the OLP. We remind that the command `OLP_EvalSubProcess` can be adapted to pass additional information between the MC and the OLP.

It is clear from the discussion that a standardised interface between a MC and an OLP is a non-trivial task for electroweak processes, because much more cross talk between the computer programmes is necessary. We would like to motivate research groups which are active in this field to provide public examples of their individual solutions such that the presented framework can be further improved in the future.

6 Summary and Outlook

This paper summarises a detailed discussion on how to interface tree-level Monte-Carlo event generators with computer codes which provide one-loop information. Due to the fast progress in the field over the last few years such a discussion was imminent, as it became more and more transparent that synergies between the MC and the loop calculator community can speed-up the production rate of higher order computations considerably. One-loop precision for a large number of QCD and electroweak observables is already available and some of these computations already use process specific interfaces. Automated approaches for one-loop amplitude computations are pursued by several groups and recent progress shows that efficient and numerically reliable implementations exist. A standardised interface will not only dispense the loop-calculator from doing the rather involved tree-level computations, it eventually will allow to efficiently import one-loop information into state of the art Monte-Carlo tools. Due to the modular set-up exchange of code from various providers will speed-up debugging and validation of computations considerably.

Following our extensive discussion on this issue during and after the Les Houches workshop 2009 we have presented in this paper the first proposal for a standardised interface between leading order Monte-Carlo tools and one-loop programs. Several program authors are committed to implement the proposed interface at the moment. The first experiences are very promising and will be documented in the Les Houches proceedings.

The current proposal must not be regarded as the endpoint of a standardisation discussion. As nothing is carved in stone yet modifications of various details are likely to improve the interface in the future, e.g. in/output formats might be specified differently. It is planned to make some programme fragments public on the Les Houches web page:

`http://www.lpthe.jussieu.fr/LesHouches09Wiki/index.php/Draft`

A natural extension of the proposed interface is to include information which is relevant for adding a parton shower on top of the proposed NLO fixed order accord. This rather concerns the treatment of the IR subtraction and is thus orthogonal to the passing of one-loop information. Discussion on this issue has already started during the 2009 Les Houches workshop.

Finally we would like to invite the community to give us feedback on this initiative. We are committed to incorporate good proposals and constructive comments in later versions. We believe that this initiative will in the end lead to a more efficient use of resources and hopefully will help to describe a wide range of Tevatron and LHC data with Monte-Carlo tools beyond the leading order level.

References

[1] M. A. Dobbs *et al.*, arXiv:hep-ph/0403045.

[2] to be done

[3] to be done

[4] to be done

[5] to be done

[6] to be done

[7] Z. Kunszt, A. Signer and Z. Trocsanyi, Nucl. Phys. B **411** (1994) 397 [arXiv:hep-ph/9305239].

[8] S. Catani, M. H. Seymour and Z. Trocsanyi, Phys. Rev. D **55** (1997) 6819 [arXiv:hep-ph/9610553].

[9] J. Smith and W. L. van Neerven, Eur. Phys. J. C **40** (2005) 199 [arXiv:hep-ph/0411357].

[10] A. Signer and D. Stockinger, Nucl. Phys. B **808** (2009) 88 [arXiv:0807.4424 [hep-ph]].

[11] A. Denner, S. Dittmaier, M. Roth and D. Wackeroth, Nucl. Phys. B **560** (1999) 33 [hep-ph/9904472].

[12] A. Denner, S. Dittmaier, M. Roth and L. H. Wieders, Nucl. Phys. B **724** (2005) 247 [arXiv:hep-ph/0505042].

[13] R. G. Stuart, Phys. Lett. B **262**, 113 (1991).

[14] A. Aeppli, F. Cuypers and G. J. van Oldenborgh, Phys. Lett. B **314** (1993) 413 [hep-ph/9303236].

[15] A. Aeppli, G. J. van Oldenborgh and D. Wyler, Nucl. Phys. B **428** (1994) 126 [hep-ph/9312212].

[16] H. G. J. Veltman, Z. Phys. C **62** (1994) 35.

[17] S. Catani and M. H. Seymour, Nucl. Phys. B **485** (1997) 291 [Erratum-ibid. B **510** (1998) 503] [hep-ph/9605323].

[18] L. Phaf and S. Weinzierl, JHEP **0104** (2001) 006 [arXiv:hep-ph/0102207].

[19] S. Catani, S. Dittmaier, M. H. Seymour and Z. Trócsányi, Nucl. Phys. B **627** (2002) 189 [arXiv:hep-ph/0201036].

[20] S. Dittmaier, Nucl. Phys. B **565** (2000) 69 [hep-ph/9904440].

[21] S. Dittmaier, A. Kabelschacht and T. Kasprzik, Nucl. Phys. B **800** (2008) 146 [arXiv:0802.1405 [hep-ph]].

[22] K. I. Aoki, Z. Hioki, M. Konuma, R. Kawabe and T. Muta, Prog. Theor. Phys. Suppl. **73** (1982) 1.

[23] M. Böhm, H. Spiesberger and W. Hollik, Fortsch. Phys. **34** (1986) 687.

[24] A. Denner, Fortsch. Phys. **41** (1993) 307.

[25] S. Dittmaier and M. Kramer, Phys. Rev. D **65** (2002) 073007 [arXiv:hep-ph/0109062].

[26] M. H. Seymour and C. Tevlin, arXiv:0803.2231 [hep-ph].

[27] T. Gleisberg and F. Krauss, Eur. Phys. J. C **53**, 501 (2008) [arXiv:0709.2881 [hep-ph]].

[28] K. Hasegawa, S. Moch and P. Uwer, Nucl. Phys. Proc. Suppl. **183**, 268 (2008) [arXiv:0807.3701 [hep-ph]].

[29] R. Frederix, T. Gehrmann and N. Greiner, JHEP **0809**, 122 (2008) [arXiv:0808.2128 [hep-ph]].

[30] M. Czakon, C. G. Papadopoulos and M. Worek, arXiv:0905.0883 [hep-ph].

[31] K. Hasegawa, S. Moch and P. Uwer, arXiv:0911.4371 [Unknown].

[32] R. Frederix, S. Frixione, F. Maltoni and T. Stelzer, JHEP **0910**, 003 (2009) [arXiv:0908.4272 [hep-ph]].

[33] `http://pdg.ift.unesp.br/2009/reviews/rpp2009-rev-monte-carlo-numbering.pdf`

[34] E. Boos et al., in proceedings of the Workshop on Physics at TeV Colliders, Les Houches, France, 21 May – 1 June 2001 [hep-ph/0109068]

[35] J. Alwall *et al.*, Comput. Phys. Commun. **176** (2007) 300 [arXiv:hep-ph/0609017].