

The POWHEG BOX user manual: Z/γ^* production in the SM-EFT

Simone Alioli,

*Universita' degli Studi di Milano-Bicocca,
20126, Milano, Italy
E-mail: simone.alioli@unimib.it*

Wouter Dekens,

*Department of Physics, University of California at San Diego,
La Jolla, CA 92093-0319, USA
E-mail: wdeken@ucsd.edu*

Michael Girard,

*Ernest Orlando Lawrence Berkeley National Laboratory, University of California,
Berkeley, CA 94720, USA
E-mail: michael.girard@berkeley.edu*

Emanuele Mereghetti,

*Theoretical Division, Los Alamos National Laboratory,
Los Alamos, NM 87545, USA
E-mail: emereghetti@lanl.gov*

ABSTRACT: This note documents the use of the package POWHEG BOX for Z/γ^* production processes, in the presence of SM-EFT operators.

Contents

1. Introduction	1
2. Generation of events	1
3. Process specific input parameters	2

1. Introduction

The POWHEG BOX program is a framework for implementing NLO calculations in Shower Monte Carlo programs according to the POWHEG method. An explanation of the method and a discussion of how the code is organized can be found in refs. [1, 2, 3]. The code is distributed according to the “MCNET GUIDELINES for Event Generator Authors and Users” and can be found at the web page

<http://powhegbox.mib.infn.it>.

This program is an implementation of the Drell-Yan NLO cross sections $pp \rightarrow Z/\gamma^* \rightarrow \ell^+ \ell^-$ induced by SM-EFT operators. A detailed description of the implementation can be found in Ref. [4]. Please cite the paper when you use the program. The implementation included in the POWHEG-BOX/Z_smeft subdirectory package is based on the original work of Ref. [5], to which we refer for more details.

In order to run the POWHEG BOX program, we recommend the reader to start from the POWHEG BOX user manual, which contains all the information and settings that are common among all subprocesses. In this note we focus on the settings and parameters specific to Z/γ^* implementation.

2. Generation of events

Due to the large number of coefficients of dimension-six operators, before building the executables, the user should increase the value of the parameter `maxnum` in the file POWHEG-BOX-V2/include/pwhg.pwin.h. `maxnum = 500` is sufficient to run the program. Build the executables

```
$ cd POWHEG-BOX-V2/Z_smeft
$ make pwhg_main
$ make main-PYTHIA8-lhef
```

	Operator			Operator	
Γ_W^u	ReGUw_ik	ImGUw_ik	Γ_W^d	ReGDw_jl	ImGDw_jl
Γ_γ^u	ReGUe_ik	ImGUe_ik	Γ_γ^d	ReGDe_jl	ImGDe_jl
Γ_W^e	ReGEw	ImGEw	Γ_γ^e	ReGEe	ImGEe
$c_{Q\varphi,U}$	QphiU_ik	$i \leq k$	$c_{Q\varphi,D}$	QphiD_jl	$j \leq l$
$c_{U\varphi}$	Uphi_ik	$i \leq k$	$c_{D\varphi}$	Dphi_jl	$j \leq l$
$C_{LQ,U}$	QLu_ik	$i \leq k$	$C_{LQ,D}$	QLd_jl	$j \leq l$
C_{eu}	Ceu_ik	$i \leq k$	C_{ed}	Ced_jl	$j \leq l$
C_{Lu}	CLu_ik	$i \leq k$	C_{Ld}	CLd_jl	$j \leq l$
$C_{LeQu}^{(3)}$	ReLeQu3_ik	ImLeQu3_ik	C_{Qe}	Qe_jl	$j \leq l$
$C_{LeQu}^{(1)}$	ReLeQu_ik	ImLeQu_ik	C_{LedQ}	ReLedQ_jl	ImLedQ_jl

Table 1: Notation for the coefficients of the dimension-six operators that can be set in POWHEG. The indices i, k run over u -type quark flavors $i, k \in \{u, c, t\}$, while j, l run over d -type quark flavors $j, l \in \{d, s, b\}$. With the assumptions of Ref. [4], $c_{Q\varphi,U}$, $c_{Q\varphi,D}$, $c_{U\varphi}$, $c_{D\varphi}$, $C_{LQ,U}$, $C_{LQ,D}$, C_{eu} , C_{ed} , C_{Lu} , C_{Ld} and C_{Qe} are real symmetric matrices. The notation $i \leq k$ and $j \leq l$ indicates that the elements $\{uu, uc, ut, cc, ct, tt\}$ and $\{dd, ds, db, ss, sb, bb\}$ can be set by the user, while the remaining elements are not independent.

Then do (for example)

```
$ cd testrun-lhc-smeft-13TeV
$ ../pwhg_main
```

At the end of the run, the file `pwgevents.lhe` will contain 100000 events for the process $Z/\gamma^* \rightarrow e^+e^-$ in the Les Houches format. In order to shower them with PYTHIA do

```
$ cd testrun-lhc-smeft-13TeV
$ ../main-PYTHIA8-lhef
```

3. Process specific input parameters

The process inherits all the parameters of the Standard Model Z/γ^* production POWHEG process, described in Ref. [5] and in `manual-BOX-Z.pdf`

By default, all dimension-six corrections are switched off in the POWHEG input card. To investigate the effect of one or more dimension-six operator, the user needs to set the flag `dim6` to 1 and specify the values of one or more coefficients of dimension-six operators in the input file. The definitions of the SM-EFT operators are given in Ref. [4], and their coefficients are defined to be dimensionless, and scaling as $(v/\Lambda)^2$, where Λ is the scale of new physics.

The user can choose between two different assumptions for the flavor structures of the coefficients, as discussed in detail in Ref. [4].

The first possibility is a generic flavor structure. The notation for the coefficients of SM-EFT operators in this scenario is given in Table 1, and the only restriction we impose is that the matrices $c_{Q\varphi,U}$, $c_{Q\varphi,D}$, $c_{U\varphi}$, $c_{D\varphi}$, $C_{LQ,U}$, $C_{LQ,D}$, C_{eu} , C_{ed} , C_{Lu} , C_{Ld} and C_{Qe} are real symmetric matrices, instead of hermitian matrices (i.e. the phases of the off-diagonal entries are set to zero). In particular, the user can set flavor off-diagonal

Operator	a_i	b_i	Operator	a_i	b_i
$c_{Q\varphi,U}$	A_QphiU	-	$c_{Q\varphi,D}$	A_QphiD	B_QphiD
$c_{U\varphi}$	A_Uphi	-	$c_{D\varphi}$	A_Dphi	-
$C_{LQ,U}$	A_QLu	-	$C_{LQ,D}$	A_QLd	B_QLd
C_{eu}	A_eu	-	C_{ed}	A_ed	-
C_{Lu}	A_Lu	-	C_{Ld}	A_Ld	-
			C_{Qe}	A_Qe	B_Qe

Table 2: Notation for the coefficients of the different flavor structures in the MFV framework, discussed in Appendix B of Ref. [4], which can be set in `POWHEG`.

coefficients, which induce tree-level processes that are absent in the SM, such as $d\bar{s} \rightarrow e^+e^-$ or $c\bar{u} \rightarrow e^+e^-$. To generate these processes, the flag `fcnc` should be set to 1 in the `POWHEG` input card. An example for the `powheg.input` card in this scenario is given in the folder `testrun-lhc-smeft-13TeV`.

As an alternative to considering generic flavor structures, the code also provides the option to assume the Minimal Flavor Violation (MFV) framework. In this scenario, the allowed forms of the couplings are constrained by flavor symmetries, which significantly decrease the number of free parameters. The implications for the flavor structures of the couplings are discussed in Appendix B of Ref. [4]. To choose this scenario, the user should set the flag `mfv` to 1, and select some of the couplings in Table 2. An example for the `powheg.input` card in this scenario is given in the folder `testrun-lhc-mfv-13TeV`.

In addition to the couplings of dimension-six operators, the user can input the elements of the CKM matrix by setting `CKM.Vij`, with $i \in \{u, c, t\}$ and $j \in \{d, s, b\}$, in the input file. By default the CKM matrix is assumed to be real, and terms of $\mathcal{O}(\lambda^2)$, where $\lambda \sim |V_{us}|$, and higher are neglected.

The dipole, scalar and tensor operators in Table 1 run in QCD, and therefore the user should specify the scale at which the coefficients are evaluated. The new physics scale Λ , at which the coefficients are defined, can be specified by setting the flag `LambdaNP` to the desired value. The coefficients are then run from Λ to μ_R , the renormalization scale of the process. By default, `LambdaNP` = 1 TeV. For the coefficients that do not evolve under QCD, the flag `LambdaNP` is irrelevant.

The contribution of dimension-six operators is most notable at large transverse mass. In the generation of events, it might then be useful to include a Born suppression factor. The suppression factor can be included by setting `bornsuppfact` in the input card. The value of `bornsuppfact` determines the p_T used in

$$\text{fact} = \left(\frac{m_{l^+l^-}^2}{m_{l^+l^-}^2 + p_T^2} \right)^4. \quad (3.1)$$

The functional form of the suppression factor can be changed by modifying the `born_suppression` routine in `Born_phsp.f`.

References

- [1] P. Nason, “A new method for combining NLO QCD with shower Monte Carlo algorithms,” *JHEP* **0411** (2004) 040 [arXiv:hep-ph/0409146].
- [2] S. Frixione, P. Nason and C. Oleari, “Matching NLO QCD computations with Parton Shower simulations: the POWHEG method,” *JHEP* **0711** (2007) 070 [arXiv:0709.2092 [hep-ph]].
- [3] S. Alioli, P. Nason, C. Oleari and E. Re, “A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX,” [arXiv:1002.2581 [hep-ph]].
- [4] S. Alioli, W. Dekens, M. Girard and E. Mereghetti, *JHEP* **1808**, 205 (2018) doi:10.1007/JHEP08(2018)205 [arXiv:1804.07407 [hep-ph]].
- [5] S. Alioli, P. Nason, C. Oleari and E. Re, “NLO vector-boson production matched with shower in POWHEG,” *JHEP* **0807**, 060 (2008) [arXiv:0805.4802 [hep-ph]].