proVBFH-inclusive v.2.0.2 manual

October 10, 2022

This document provides a short documentation for the proVFBH-inclusive code.

1 Credits

The proVFBH program was developed by Matteo Cacciari, Frédéric Dreyer, Alexander Karlberg, Gavin Salam and Giulia Zanderighi and is based on

- M. Cacciari, F. A. Dreyer, A. Karlberg, G. P. Salam and G. Zanderighi, Phys. Rev. Lett. **115** (2015) no.8, 082002 [arXiv:1506.02660 [hep-ph]],
- F. A. Dreyer and A. Karlberg, Phys. Rev. Lett. **117** (2016) no.7, 072001 [arXiv:1606.00840 [hep-ph]].

When using results at next-to-next-to-next-to-leading order (N^3LO), the original references on third order coefficient functions should be cited as well

- A. Vogt, S. Moch and J. A. M. Vermaseren, Nucl. Phys. B **691** (2004) 129 [hep-ph/0404111].
- S. Moch, J. A. M. Vermaseren and A. Vogt, Phys. Lett. B **606** (2005) 123 [hep-ph/041.2.1].
- J. A. M. Vermaseren, A. Vogt and S. Moch, Nucl. Phys. B **724** (2005) 3 [hep-ph/0504242].
- S. Moch, M. Rogal and A. Vogt, Nucl. Phys. B **790** (2008) 317 [arXiv:0708.3731 [hep-ph]].

Finally, when evaluating cross sections for di-Higgs production the following reference should be cited

- F. A. Dreyer and A. Karlberg, Phys. Rev. D **98** (2018) no.11, 114016 [arXiv:1811.07906 [hep-ph]].

2 Installation/dependencies

To run proVFBH-inclusive, you will need an installation of the following packages:

- hoppet/struct-func-devel: http://hoppet.hepforge.org/. Note that it is specifically the struct-func-devel branch of hoppet that is required. It can be downloaded using:

```
svn checkout https://svn.hepforge.org/hoppetsvn/branches/struct-func-devel/
```

- LHAPDF: `http://lhapdf.hepforge.org/`.

With these installed, `proVFBH-inclusive` can be compiled by going to the base directory and running

```
./configure [options]
make
```

Available options in the `configure` script can be accessed through the `--help` or `-h` argument, as well as being described in the `INSTALL` text file.

We suggest that the references for `hoppet` and `LHAPDF` are also cited when using our program.

3 Setting up a run

To run `proVFBH-inclusive`, you can simply call the executable `provbfh_incl` created by the installation steps above. All parameters are passed as command line options, with the full list of settings detailed in the section below, or accessible in `src/parameters.f90`. Since executing the code will produce several files, the recommended usage is to start individual runs in dedicated subfolders.

An example setup:

```
mkdir nlo-14tev; cd $_
../provbfh_incl -nlo -sqrts 14000 -iseed 2
```

This will produce three files, two files `grids_0002.dat` and `grids_0002.top` containing the grid and an output file `xsct_nlo_seed0002.dat` containing the results of the run. Different seed numbers and orders in α_s can be executed in the same location, as they will result in distinct output files.

4 Dihiggs production

To run `proVFBH-inclusive` for dihiggs production, one needs to compile the `provbfhh_incl` program with

```
make provbfhh_incl
```

which can then be executed exactly like the single Higgs program to calculate the total cross section in VBF Higgs pair production.

5 Input parameters

All accessible parameters can be specified as command line arguments.

The available options, and their default value (in **bold**), are:

- `-lo, -nlo, -nnlo, -n3lo`: Order in α_s.
- `-sqrts` **13600**: Center-of-mass energy in GeV.
- `-scale-choice` **1**: Scale choice to use, with the options
  - 0: Fixed scale at $\mu_0^2 = m_h^2$.

  - 1: Vector boson momentum $\mu_0^2(Q_i^2) = Q_i^2 = -q_i^2$.
  - 2: Common scale given by $\mu_0^2(Q_1, Q_2) = Q_1 Q_2$.
  - 3: $\mu_0^2(p_{t,H}) = \frac{m_H}{2}\sqrt{\left(\frac{m_H}{2}\right)^2 + p_{t,H}^2}$ from 1506.02660.
- `-xmuf` **1.0**: Factor x_{μ_F} multiplying the factorisation scale $\mu_F = x_{\mu_F}\mu_0$.
- `-xmur` **1.0**: Factor x_{μ_R} multiplying the renormalisation scale $\mu_R = x_{\mu_R}\mu_0$.
- `-pdf` **PDF4LHC21_40**: Choice of PDF set (by name).
- `-nmempdf` **0**: Member of the PDF.
- `-mh` **125.**: Higgs mass.
- `-hwidth` **0.00402964**: Higgs width.
- `-mw` **80.398**: W mass.
- `-wwidth` **2.141**: W width.
- `-mz` **91.187**: Z mass.
- `-zwidth` **2.4952**: Z width.
- `-nf` **5**: number of quark flavours.
- `-mt` **172.4**: top mass.
- `-mb` **4.75**: bottom mass.
- `-readingrid`: Use available grid if possible.
- `-higgsbreitwigner`: Use a Breit-Wigner propagator for the higgs.
- `-higgsmasswindow` **30.**: Number of width to integrate around the BW peak.
- `-ncall1` **100000**: Number of calls for the initialisation of the integration grid.
- `-itmx1` **3**: Number of iterations for the grid initialisation.
- `-ncall2` **100000**: Number of calls for evaluating the integral.
- `-itmx2` **3**: Number of iterations for the integration.
- `-iseed` **10**: Random seed.
- `-pdfuncert`: Compute PDF uncertainties on the fly. Significantly increases computation time (scales linearly with the number of PDF members). Also computes α_S uncertainty if this is included in the PDF.
- `-alphasuncert`: Separates the PDF and α_S uncertainty on the printout.
- `-3scaleuncert`: Compute 3 point (ie symmetric) scale variation.
- `-7scaleuncert`: Compute 7 point scale variation.