`proVBFH-inclusive v.1.0.1` manual

June 28, 2017

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

1 Installation

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 http://hoppet.hepforge.org/svn/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.

2 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.

3 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` **13000**: 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 [**?**].
- `-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` **PDF4LHC15_nnlo_mc**: 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` **12**: Number of iterations for the grid initialisation.
- `-ncall2` **100000**: Number of calls for evaluating the integral.
- `itmx2` **12**: Number of iterations for the integration.
- `iseed` **10**: Random seed.