Manual for the JHU generator and MELA package

For simulation, reweighting, and analysis of single-produced resonances at hadron colliders
(version v7.5.0, release date July 10, 2021)

The generator from [1–6] is a model-independent generator for studying spin-parity properties of new resonances, including the Higgs boson(s). Please cite [1–6] if using the "JHU generator" (or JHUGen). The code can be downloaded from [7]. The generator outputs LHE files which can be passed to parton shower programs for hadronization. Only relative values of cross sections are supposed to produce meaningful results, while absolute values are often subject to an arbitrary normalization.

Additionally, the package includes code for computing the matrix elements standalone which can be used in a numerical matrix element analysis. Please reference the above papers and refer to "MELA" when using the matrix element likelihood approach. The latter was also introduced in Ref. [8]. The matrix element package (MELA) and off-shell EW generation also depend on MCFM libraries for background parameterization which should be referenced [9] when used. In order to simplify translation between different coupling conventions, within the JHU generator framework, we provide the JHUGenLexicon program, which includes an interface to the generator and matrix element library and can also be used for standalone or other applications.

Contents

I. NOTATION

The notation has been set in Refs. [1–5], the HVV scattering amplitude of a spin-zero boson H and two vector bosons with polarization vectors $\varepsilon_1^\mu(q_1)$ and $\varepsilon_2^\mu(q_2)$ is parameterized by

$$A(HVV) = \frac{1}{v} \left\{ M_V^2 \left(g_1^{VV} + \frac{\kappa_1^{VV} q_1^2 + \kappa_2^{VV} q_2^2}{\left(\Lambda_1^{VV}\right)^2} + \frac{\kappa_3^{VV}(q_1 + q_2)^2}{\left(\Lambda_Q^{VV}\right)^2} + \frac{2 q_1 \cdot q_2}{M_V^2} g_2^{VV} \right)(\varepsilon_1 \cdot \varepsilon_2) \right.$$
$$\left. -2 g_2^{VV} (\varepsilon_1 \cdot q_2)(\varepsilon_2 \cdot q_1) - 2 g_4^{VV} \varepsilon_{\varepsilon_1 \varepsilon_2 q_1 q_2} \right\}. \tag{1}$$

This parametrization represents the most general Lorentz-invariant form. The couplings used in Eq. (1) are the direct input to the JHU generator, as discussed in the rest of this manual.

The parameterization of the amplitude in Eq. (1) can be related to a fundamental Lagrange density function, using effective field theory coefficients of the mass-eigenstate basis [10]:

$$\mathcal{L}_{\text{hvv}} = \frac{h}{v} \left[M_Z^2 \left(1 + \delta c_z\right) Z_\mu Z^\mu + \frac{M_Z^2}{v^2} c_{zz} Z_{\mu\nu} Z^{\mu\nu} + \frac{e^2}{s_w^2} c_{z\square} Z_\mu \partial_\nu Z^{\mu\nu} + \frac{M_Z^2}{v^2} \tilde{c}_{zz} Z^{\mu\nu} \tilde{Z}_{\mu\nu} \right.$$
$$+ 2 M_W^2 \left(1 + \delta c_w\right) W_\mu^+ W^{-\mu} + 2 \frac{M_W^2}{v^2} c_{ww} W_{\mu\nu}^+ W^{-\mu\nu} + \frac{e^2}{s_w^2} c_{w\square} \left(W_\mu^- \partial_\nu W^{+\mu\nu} + \text{h.c.}\right) + \frac{e^2}{2 s_w^2} \tilde{c}_{ww} W^{+\mu\nu} \tilde{W}_{\mu\nu}^-$$
$$+ \frac{e^2}{2 s_w c_w} c_{z\gamma} Z_{\mu\nu} A^{\mu\nu} + \frac{e^2}{2 s_w c_w} \tilde{c}_{z\gamma} Z_{\mu\nu} \tilde{A}^{\mu\nu} + \frac{e^2}{s_w c_w} c_{\gamma\square} Z_\mu \partial_\nu A^{\mu\nu}$$
$$\left. + c_{\gamma\gamma} \frac{e^2}{4} A_{\mu\nu} A^{\mu\nu} + \tilde{c}_{\gamma\gamma} \frac{e^2}{4} A^{\mu\nu} \tilde{A}_{\mu\nu} + c_{gg} \frac{g_s^2}{4} G_{\mu\nu}^a G^{a\mu\nu} + \tilde{c}_{gg} \frac{g_s^2}{4} G^{a\mu\nu} \tilde{G}_{\mu\nu}^a \right], \tag{2}$$

in accordance with Eq. (II.2.20) in Ref. [10], where $e^2 = 4\pi\alpha$ and $g_s^2 = 4\pi\alpha_s$ are the squared electromagnetic and strong coupling constants, respectively, and $s_w = \sqrt{1 - c_w^2}$ is the sine of the weak mixing angle. The covariant derivative used to derive this expression is $D_\mu = \partial_\mu - \mathrm{i}\frac{e}{2s_w}\sigma^i W_\mu^i - \mathrm{i}\frac{e}{2c_w}B_\mu$ [10, 11]. We note that the convention $\varepsilon_{0123} = +1$ defines the relative sign of the CP-odd $\tilde{c}_i$ and CP-even c_i couplings [5], while the relative sign in front of the W_μ^i and B_μ terms in the covariant derivative defines the sign of the $Z\gamma$ couplings relative to the ZZ and $\gamma\gamma$. The latter could be viewed as the sign of s_w, should a different convention be adopted. In the actual parameterization of the JHUGEN framework documented here and in Refs. [1–6] the $D_\mu = \partial_\mu - \mathrm{i}\frac{e}{2s_w}\sigma^i W_\mu^i + \mathrm{i}\frac{e}{2c_w}B_\mu$ convention was adopted for historical reasons. A transformation $g_i^{Z\gamma} \to -g_i^{Z\gamma}$ or $\kappa_i^{Z\gamma} \to -\kappa_i^{Z\gamma}$ of the input parameters in this framework would lead to the convention $D_\mu = \partial_\mu - \mathrm{i}\frac{e}{2s_w}\sigma^i W_\mu^i - \mathrm{i}\frac{e}{2c_w}B_\mu$, which is needed for consistent application of the formalism discussed in this section on notation.

There is unique representation of each EFT coefficient in Eq. (2) by couplings in Eq. (1), as shown below:

$$
\delta c_z = \frac{1}{2}g_1^{ZZ} - 1, \qquad c_{zz} = -\frac{2s_w^2 c_w^2}{e^2}g_2^{ZZ}, \qquad c_{z\square} = \frac{M_Z^2 s_w^2}{e^2}\frac{\kappa_1^{ZZ}}{(\Lambda_1^{ZZ})^2}, \qquad \tilde{c}_{zz} = -\frac{2s_w^2 c_w^2}{e^2}g_4^{ZZ},
$$

$$
\delta c_w = \frac{1}{2}g_1^{WW} - 1, \qquad c_{ww} = -\frac{2s_w^2}{e^2}g_2^{WW}, \qquad c_{w\square} = \frac{M_W^2 s_w^2}{e^2}\frac{\kappa_1^{WW}}{(\Lambda_1^{WW})^2}, \qquad \tilde{c}_{ww} = -\frac{2s_w^2}{e^2}g_4^{WW},
$$

$$
c_{z\gamma} = -\frac{2s_w c_w}{e^2}g_2^{Z\gamma}, \qquad \tilde{c}_{z\gamma} = -\frac{2s_w c_w}{e^2}g_4^{Z\gamma}, \qquad c_{\gamma\square} = \frac{s_w c_w}{e^2}\frac{M_Z^2}{(\Lambda_1^{Z\gamma})^2}\kappa_2^{Z\gamma},
$$

$$
c_{\gamma\gamma} = -\frac{2}{e^2}g_2^{\gamma\gamma}, \qquad \tilde{c}_{\gamma\gamma} = -\frac{2}{e^2}g_4^{\gamma\gamma}, \qquad c_{gg} = -\frac{2}{g_s^2}g_2^{gg}, \qquad \tilde{c}_{gg} = -\frac{2}{g_s^2}g_4^{gg}. \tag{3}
$$

Enforcing linear relations for the dependent coefficients [10] allows a unique relationship based on a minimal set of degrees of freedom:

$$
g_1^{WW} = g_1^{ZZ}, \tag{4}
$$

$$
g_2^{WW} = c_w^2 g_2^{ZZ} + s_w^2 g_2^{\gamma\gamma} + 2s_w c_w g_2^{Z\gamma}, \tag{5}
$$

$$
g_4^{WW} = c_w^2 g_4^{ZZ} + s_w^2 g_4^{\gamma\gamma} + 2s_w c_w g_4^{Z\gamma}, \tag{6}
$$

$$
\frac{\kappa_1^{WW}}{(\Lambda_1^{WW})^2}(c_w^2 - s_w^2) = \frac{\kappa_1^{ZZ}}{(\Lambda_1^{ZZ})^2} + 2s_w^2\frac{g_2^{\gamma\gamma} - g_2^{ZZ}}{M_Z^2} + 2\frac{s_w}{c_w}(c_w^2 - s_w^2)\frac{g_2^{Z\gamma}}{M_Z^2}, \tag{7}
$$

$$
\frac{\kappa_2^{Z\gamma}}{(\Lambda_1^{Z\gamma})^2}(c_w^2 - s_w^2) = 2s_w c_w\left(\frac{\kappa_1^{ZZ}}{(\Lambda_1^{ZZ})^2} + \frac{g_2^{\gamma\gamma} - g_2^{ZZ}}{M_Z^2}\right) + 2(c_w^2 - s_w^2)\frac{g_2^{Z\gamma}}{M_Z^2}. \tag{8}
$$

Automatic relationship of couplings in the EFT approach discussed here is not implemented directly in the generator, but could be obtained with a script following the formulas above. There is a one-to-one correspondence of the couplings used in the generator settings in Eq. (1) and the couplings used in the other bases (e.g. Higgs, Warsaw, SILH) [10].

We also allow modifications of gauge boson self couplings. Their parameterization reads

$$
A(VW^+W^-) = (-e)\, d^{VWW}\left\{(\varepsilon_V \cdot \varepsilon_+)(q_{12}^V \cdot \varepsilon_-) + (\varepsilon_+ \cdot \varepsilon_-)(q_{23}^V \cdot \varepsilon_V) + (\varepsilon_V \cdot \varepsilon_-)(q_{31}^V \cdot \varepsilon_+) + d_4^V \varepsilon^{\varepsilon_V\, \varepsilon_+\, \varepsilon_-\, p_1}\right\}, \tag{9}
$$

$$
A(V_1 V_2 W^+ W^-) = (+e^2)\, d^{VVWW}\left\{(\varepsilon_1 \cdot \varepsilon_+)(\varepsilon_2 \cdot \varepsilon_-) + (\varepsilon_1 \cdot \varepsilon_-)(\varepsilon_2 \cdot \varepsilon_+) - 2(\varepsilon_1 \cdot \varepsilon_2)(\varepsilon_+ \cdot \varepsilon_-)\right\}, \tag{10}
$$

where $q_{ij}^V = d_i^V p_i - d_j^V p_j$ is the relative momentum transfer. We fix $d^{\gamma WW} = 1$ and $d^{ZWW} = c_w/s_w$ per convention and allow all other couplings to vary. In the SM, their values are

$$
d_1^V = d_2^V = d_3^V = 1, \quad d_4^V = 0, \tag{11}
$$

$$
d^{\gamma\gamma WW} = 1, \quad d^{Z\gamma WW} = \frac{c_w}{s_w}, \quad d^{ZZWW} = \frac{c_w^2}{s_w^2}, \quad d^{WWWW} = \frac{1}{2s_w^2}.
$$

The relevant contributions of the dimension-six Lagrangian for the triple and quartic gauge boson self-interactions

are (see Eqs. (3.12, 3.14, 3.15) in Ref. [11])

$$
\begin{aligned}
\mathcal{L}_{\text{tgc}} = \quad & \text{ie}\left(W_{\mu\nu}^{+}W_{\mu}^{-} - W_{\mu\nu}^{-}W_{\mu}^{+}\right)A_{\nu} + \text{ie}\left[(1+\delta\kappa_{\gamma})A_{\mu\nu}\,W_{\mu}^{+}W_{\nu}^{-} + \tilde{\kappa}_{\gamma}\tilde{A}_{\mu\nu}\,W_{\mu}^{+}W_{\nu}^{-}\right] \\
+ \quad & \text{ie}\frac{c_{w}}{s_{w}}\left[(1+\delta g_{1,z})\left(W_{\mu\nu}^{+}W_{\mu}^{-} - W_{\mu\nu}^{-}W_{\mu}^{+}\right)Z_{\nu} + (1+\delta\kappa_{z})\,Z_{\mu\nu}\,W_{\mu}^{+}W_{\nu}^{-} + \tilde{\kappa}_{z}\,\tilde{Z}_{\mu\nu}\,W_{\mu}^{+}W_{\nu}^{-}\right],
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
\mathcal{L}_{\text{qgc}} = \quad & e^{2}(W_{\mu}^{+}A_{\mu}W_{\nu}^{-}A_{\nu} - W_{\mu}^{+}W_{\mu}^{-}A_{\nu}A_{\nu}) + \frac{e^{2}}{2s_{w}^{2}}(1+2c_{w}^{2}\delta g_{1,z})(W_{\mu}^{+}W_{\mu}^{+}W_{\nu}^{-}W_{\nu}^{-} - W_{\mu}^{+}W_{\mu}^{-}W_{\nu}^{+}W_{\nu}^{-}) \\
+ \quad & e^{2}\frac{c_{w}^{2}}{s_{w}^{2}}(1+2\delta g_{1,z})(W_{\mu}^{+}Z_{\mu}W_{\nu}^{-}Z_{\nu} - W_{\mu}^{+}W_{\mu}^{-}Z_{\nu}Z_{\nu}) \\
+ \quad & e^{2}\frac{c_{w}}{s_{w}}(1+\delta g_{1,z})(W_{\mu}^{+}Z_{\mu}W_{\nu}^{-}A_{\nu} + W_{\mu}^{+}A_{\mu}W_{\nu}^{-}Z_{\nu} - 2W_{\mu}^{+}W_{\mu}^{-}Z_{\nu}A_{\nu}).
\end{aligned}
\tag{13}
$$

The anomalous coefficients in Eqs. (12,13) are related to couplings in Eqs. (9,10) by

$$
\delta\kappa_{\gamma} = \frac{1}{2}\left(d_{1}^{\gamma}-1\right), \qquad \tilde{\kappa}_{\gamma} = \frac{1}{2}d_{4}^{\gamma}, \qquad \delta\kappa_{z} = \frac{1}{2}\left(d_{1}^{Z}-1\right), \qquad \tilde{\kappa}_{z} = \frac{1}{2}d_{4}^{Z},
$$

$$
\delta g_{1,z} = d_{2}^{Z}-1 = d_{3}^{Z}-1 = \frac{1}{2}\left(\frac{s_{w}^{2}}{c_{w}^{2}}d^{ZZWW}-1\right) = \frac{s_{w}}{c_{w}}d^{Z\gamma WW}-1.
\tag{14}
$$

Similar to the case of $\mathcal{L}_{\text{hvv}}$, not all coefficients in Eqs. (12,13) are independent in the effective field theory framework, and we discuss their dependence in the next subsection. Moreover, additional anomalous triple and quartic contributions, the $\lambda_{\gamma,Z}, \tilde{\lambda}_{\gamma,Z}$ terms in Ref. [10, 11], can arise. These additional terms are unrelated to any of the H boson contributions, and therefore, we do not consider them here.

Enforcing linear relations for the dependent coefficients [8] allows a unique relationship based on a minimal set of degrees of freedom:

$$
d_{1}^{\gamma} = 1 + (g_{2}^{\gamma\gamma} - g_{2}^{ZZ})c_{w}^{2} + g_{2}^{Z\gamma}\left(\frac{c_{w}}{s_{w}} - 2s_{w}c_{w}\right),
\tag{15}
$$

$$
d_{4}^{\gamma} = (g_{4}^{\gamma\gamma} - g_{4}^{ZZ})c_{w}^{2} + g_{4}^{Z\gamma}\left(\frac{c_{w}}{s_{w}} - 2s_{w}c_{w}\right),
\tag{16}
$$

$$
d_{1}^{Z} = 1 - 2\frac{s_{w}^{2}c_{w}^{2}}{c_{w}^{2}-s_{w}^{2}}\left(g_{2}^{\gamma\gamma} - g_{2}^{ZZ}\right) - 2s_{w}c_{w}g_{2}^{Z\gamma} - \frac{M_{Z}^{2}}{2(c_{w}^{2}-s_{w}^{2})}\frac{\kappa_{1}^{ZZ}}{(\Lambda_{1}^{ZZ})^{2}},
\tag{17}
$$

$$
d_{2}^{Z} = d_{3}^{Z} = 1 - \frac{s_{w}^{2}}{c_{w}^{2}-s_{w}^{2}}\left(g_{2}^{\gamma\gamma} - g_{2}^{ZZ}\right) - \frac{s_{w}}{c_{w}}g_{2}^{Z\gamma} - \frac{M_{Z}^{2}}{2(c_{w}^{2}-s_{w}^{2})}\frac{\kappa_{1}^{ZZ}}{(\Lambda_{1}^{ZZ})^{2}},
\tag{18}
$$

$$
d_{4}^{Z} = -\frac{s_{w}^{2}}{c_{w}^{2}}d_{4}^{\gamma},
\tag{19}
$$

$$
d^{ZZWW} = \frac{c_{w}^{2}}{s_{w}^{2}}\left(2d_{2}^{Z}-1\right), \qquad d^{Z\gamma WW} = \frac{c_{w}}{s_{w}}d_{2}^{Z}.
\tag{20}
$$

The Standard Modell Zff couplings are defined as

$$
\begin{aligned}
g_{\text{L}}^{Zff} &= \frac{e}{s_{w}c_{w}}\left(T_{f_{\text{L}}}^{3} - s_{w}^{2}Q_{f}\right) &=: \frac{e}{2\,s_{w}\,c_{w}}a_{\text{L}}^{f}, \\
g_{\text{R}}^{Zff} &= \frac{e}{s_{w}c_{w}}\left(-s_{w}^{2}Q_{f}\right) &=: \frac{e}{2\,s_{w}\,c_{w}}a_{\text{R}}^{f}.
\end{aligned}
\tag{21}
$$

II. INSTALLATION

Download the package from `spin.pha.jhu.edu` and untar the file. Go to the directory `JHUGenerator` where the code exists for generating events with the JHU Generator. In the `makefile`, you have two options for compiler, `Comp = ifort` or `Comp = gfort`. Then simply compile with:

```
$ make
```

By default, the makefile options `linkMELA` and `linkCollierLib` are set to `No`. If you need to use the NLO features, you need to set `linkCollierLib = Yes`, and download and compile the COLLIER package using `./downloadcollier.sh` with optional arguments `compiler` and `-j` to specify the Fortran compiler and the number of cores to use for compilation, respectively. Likewise, if you need to use off-shell processes, which use modified MCFM matrix elements, you need to set `linkMELA = Yes` and compile JHUGenMELA with at least the option `deps` before compiling JHUGenerator (see compilation instructions for JHUGenMELA for more details). In the case `linkMELA = Yes`, the compiler has to be set to `Comp = gfort`, and the gcc version should be at least 4.6. Notice that JHUGenMELA has its own compilation routine and location for the COLLIER library.

Also note that after JHUGenerator v7.3.0 was released, the naming scheme used for COLLIER libraries was changed. For any version of JHUGenerator preceding and including v7.3.0, this change in package location must be accounted for before installing the COLLIER library. If you are compiling JHUGenMELA, the script `COLLIER/setup.sh` must be edited to use `pkgname = "collier-1.2.0"`. In the case you do not link JHUGenMELA but still need COLLIER, `./downloadcollier.sh` should reference `collier-1.2.0.tar.gz` instead of `collier-1.2.tar.gz`.

III.   QUICK START

Examples of running the generator. The number of events is determined using `VegasNc2`.

- $gg \to H$ production, `Process=0`:

  - SM resonance, decay to $ZZ \to 4l$

    `./JHUGen Process=0 VegasNc2=100000 DecayMode1=0 DecayMode2=0 DataFile=ggHZZ4l`

  - pseudoscalar resonance, decay $WW \to 2l2\nu$:

    `./JHUGen Process=0 VegasNc2=100000 DecayMode1=4 DecayMode2=4 DataFile=ggHWW2l2nu_0- \\`
    `ghz1=0,0 ghz4=1,0`

  - SM resonance, decay to $Z\gamma \to 2l\gamma$

    `./JHUGen Process=0 VegasNc2=100000 DecayMode1=0 DecayMode2=7 DataFile=ggHgammaZ2l \\`
    `ghzgs2=1`

- VH, `Process=50`:

  - ZH with hadronic Z decay (change `DecayMode1` for other Z decays; both pp (default) and e^+e^- Collider options possible):

    `./JHUGen Process=50 DecayMode1=1 VegasNc2=100000 DataFile=ZH`

  - WH with leptonic W decay (change `DecayMode1` for other W decays):

    `./JHUGen Process=50 DecayMode1=4 VegasNc2=100000 DataFile=WH`

  - $\gamma^* \to \gamma H$ (both pp (default) and e^+e^- Collider options possible):

    `./JHUGen Process=50 DecayMode1=7 VegasNc2=100000 DataFile=gammatogammaH ghgsgs2=1,0`

  - $Z^* \to \gamma H$ (both pp (default) and e^+e^- Collider options possible):

    `./JHUGen Process=50 DecayMode1=7 VegasNc2=100000 DataFile=ZtogammaH ghzgs2=1,0`

- $gg \to ZH$ and $qq \to VH$ at NLO, `Process=51`:

  - ZH production with g_2 anomalous coupling, decay to $Z \to 2l$, $H \to b\bar{b}$:

    `./JHUGen Collider=1 Process=51 VH_PC=gg HbbDK=1 DecayMode1=0 VegasNc2=100000 \\`
    `DataFile=ggZH2lbb ghz1=0,0 ghz2=1,0`

  - ZH production with SM couplings ($g_1 = 2$), with $Z \to 2l$ decay and "stable" Higgs (for further processing):

    `./JHUGen Collider=1 Process=51 VH_PC=gg HbbDK=0 DecayMode1=0 VegasNc2=100000 \\`
    `DataFile=ggtoZHtollH ghz1=2,0`

- – *ZH* production at NLO with SM couplings ($g_1 = 2$), with $Z \to 2l$ decay and "stable" Higgs (for further processing). Note: for *ZH* production at NLO, `Process=51` can produce weighted events only. One must specify `Unweighted=0`:

  ```
  ./JHUGen Collider=1 Process=51 VH_PC=nl HbbDK=0 DecayMode1=0 VegasNc2=100000 \\
  DataFile=qqtoZH_nlo_to_llbb ghz1=2,0 Unweighted=0
  ```

  – `Process=51` also includes all the features of `Process=50`, but is currently in beta; we recommend using `Process=51` only for $gg \to ZH$.

- VBF, `Process=60`:

  – $Z^*Z^*/W^*W^* \to H$:

    ```
    ./JHUGen Process=60 VegasNc2=100000 DataFile=VBF pTjetcut=0 deltaRcut=0
    ```

    (Jet cuts are generally not needed for VBF production with only ZZ and WW, since there is no divergence.)

  – $\gamma^*\gamma^* \to H$:

    ```
    ./JHUGen Process=60 VegasNc2=100000 DataFile=gammagammaVBF ghgsgs2=1,0 \\
    pTjetcut=15 deltaRcut=0
    ```

    (Requires cut on jet p_T to avoid divergences when $q_\gamma^2 \to 0$. 15 GeV is the default cut, but is shown explicitly as an example.)

  – $Z^*\gamma^* \to H$:

    ```
    ./JHUGen Process=60 VegasNc2=100000 DataFile=ZgammaVBF ghzgs2=1,0 \\
    pTjetcut=15 deltaRcut=0
    ```

- off-shell EW $q\bar{q} \to q\bar{q} + ZZ(\to 4f)$, `Process=66, 67, 68`

  – including VBF and hadronic VH with up to two scalar resonances and interference with the SM background
  – see below, page 11, for the configuration

- $H + jj$, `Process=61`:

  ```
  ./JHUGen Process=61 Unweighted=1 VegasNc2=100000 DataFile=Hjj pTjetcut=15 deltaRcut=0.3
  ```

  (Requires cuts on jet p_T and ΔR to avoid divergences when $q_g^2 \to 0$. 15 GeV and 0.3 are the default cuts, but are shown explicitly as an example.)

- $H + j$:

  ```
  ./JHUGen Process=62 Unweighted=1 VegasNc2=100000 DataFile=Hj pTjetcut=15
  ```

- $pp \to t\bar{t} + H$, `Process=80`:

  – inclusive top decay (change `DecayMode1,2` for specific tt decays):

    ```
    ./JHUGen Process=80 DecayMode1=11 DecayMode2=11 VegasNc2=100000 DataFile=ttH
    ```

- $pp \to b\bar{b} + H$, `Process=90`:

  ```
  ./JHUGen Process=90 VegasNc2=100000 DataFile=bbH pTjetcut=15 deltaRcut=0.3
  ```

- $pp \to tH$, `Process=114`:

  – $tH + \bar{t}H$, with inclusive top decay (change `DecayMode1` for specific t decays; use processes 110–113 restrict to either t or $\bar{t}$ and either s or t channels):

    ```
    ./JHUGen Process=114 DecayMode1=11 VegasNc2=100000 DataFile=tH_s
    ```

- $pp \to tHW$, `Process=117`:

– $tHW^- + \bar{t}HW^+$, with inclusive top and W decay (change `DecayMode1` for specific t decays and `DecayMode2` for specific W decays; use processes 115–116 restrict to either t or $\bar{t}$:

```
./JHUGen Process=117 DecayMode1=11 DecayMode2=11 VegasNc2=100000 DataFile=tHW
```

- Decay-only mode, for generating the decay of a stable Higgs boson produced by the JHU generator or another generator (e.g. POWHEG).

  – $H \rightarrow VV$ decay:

```
./JHUGen ReadLHE=ZH.lhe DecayMode1=0 DecayMode2=0 DataFile=ZH_ZZ4l
./JHUGen ReadLHE=ZgammaVBF.lhe DecayMode1=0 DecayMode2=7 DataFile=ZgammaVBF_Z2lgamma
./JHUGen ReadLHE=myPOWHEGfile.lhe DecayMode1=4 DecayMode2=4 ghz1=0,0 ghz4=1,0 \\
      DataFile=myPOWHEGfile_0-_WW2l2nu
```

  – Decay-only mode for generating $H \rightarrow \tau\tau$ decay. Note that τ decays to hadronic resonances are not supported.

```
./JHUGen ReadLHE=myPOWHEGfile.lhe TauDK=1 DecayMode1=4 DecayMode2=4 kappa=0,0 \\
kappa_tilde=1,0 DataFile=myPOWHEGfile_0-_tautau2l4nu
```

IV. CONFIGURATION

Most parameters should be set on the command line.

A. Command line configuration

1. Quick reference

This is the list of command line configurables (also defined in the `README`). The following sections give further details about some of these parameters.

```
help:                Print all command line options
version:             Print the version tag
header:              Print the header/logo
DryRun:              Check that the command line is valid, then exit
Process configuration:
  Collider:          1=LHC (default), 2=Tevatron, 0=e+e-
  ColliderEnergy:    in TeV.  default is 13 TeV for LHC, 1.96 TeV for Tevatron,
                     250 GeV for e+e-
  Process:           0=spin-0, 1=spin-1, 2=spin-2 resonance,
                     50=qq/ee->VH, 51=gg->ZH,
                     60=VBF, 61=pp->Hjj, 62=pp->Hj,
                     66=EW VVJJ sig., 67=EW VVJJ bkg., 68=EW VVJJ tot.,
                     69=QCD VVJJ bkg.,
                     70=EW VVLL sig., 71=EW VVLL bkg., 72=EW VVLL tot.,
                     73=gg 4f sig., 74=gg 4f bkg., 75=gg 4f tot.,
                     80=ttH, 90=bbH,
                     110=t+H t channel, 111=tbar+H t channel,
                     112=t+H s channel, 113=tbar+H s channel
                     114=t/tbar+H t/s channels
                     115=t+W+H, 116=tbar+W+H
                     117=t/tbar+W+H
  DecayMode1:        decay mode for vector boson 1 (Z/W/gamma)
  DecayMode2:        decay mode for vector boson 2 (Z/W/gamma)
                       0=Z->2l,  1=Z->2q, 2=Z->2tau, 3=Z->2nu,
                       4=W->lnu, 5=W->2q, 6=W->taunu,
                       7=gamma, 8=Z->2l+2tau,
                       9=Z->anything, 10=W->lnu+taunu, 11=W->anything
```

```
                        Exclusive difermion modes other than those with the tau flavor:
                        The numbers p=(1; 2, 3; 5, 7, 11, 13, 17)
                        represent f=(nu; e, mu; d, u, s, c, b). In order to have a
                        V->f1 f2 decay, please specify the decay mode as -p1*p2
                        for the p1 and p2 that match f1 and f2.
                        (Note: negative DecayMode is in beta development mode in this version)
    Interf:             0=neglect interference for 4f final states,
                        1=include interference
    ReweightInterf:     if true, include interference as LHE event weights for
                        off-shell EW events
    RandomizeVVdecays:  Randomizes the order of DecayMode1 and DecayMode2,
                        per event (default true)
                        For a WW decay, turning this off will mean
                        DecayMode1 is W+ and DecayMode2 is W-
    PChannel:           0=g+g, 1=q+qb, 2=both
    ChannelRatio:       ratio of qqb / (qqb + gg), for Process=2 PChannel=2
                         default is to allow this ratio to come from the couplings
    PDFSet:             1=CTEQ6L1(2001),  2=MSTW(2008),
                        2xx=MSTW with eigenvector set xx=01..40,
                        3=NNPDF3.0LO
                        (only valid if not interfaced with LHAPDF)
    LHAPDF:             info file to use if interfaced with LHAPDF
                        (example: NNPDF30_lo_as_0130/NNPDF30_lo_as_0130.info)
    LHAPDFMem:          member number in LHAPDF set
    epPolarization:     Polarization of e+ for e+e- collider
    emPolarization:     Polarization of e- for e+e- collider
                          0:      no polarization
                          +/-100: helicity=+/-1
    TopDK:              For ttH or t+H, 0=leave top quarks as stable, 1=decay top quarks
    TauDK:              In ReadLHE mode, specify this option as either 0 or 1
                        to decay H->tautau.  If it is 0, the taus are written as
                        stable; if it is 1, they decay to Wnu, with the W's decaying
                        according to DecayModes1,2.
    HbbDK:              For VH production, decay H->bb
    VH_PC:              VH partonic channel and mode selection
                        ee ( = e+ e- @LO)
                        gg ( = triangles + boxes of gg)
                        qq ( = q q~ @LO)
                        lo ( = q q~ @LO)
                        tr ( = triangles of gg)
                        bo ( = boxes of gg)
                        in ( = interference = 2*dble(box*dconjg(triangle)) of gg)
                        qg ( = real - dipoles, for g q/q~ > VH + q/q~, for development only)
                        gq ( = K + P, for g q/q~ > VH + q/q~, for development only)
                        sb ( = real - dipoles, for q q~ @NLO, for development only)
                        sp ( = virtual + I + K + P, for q q~ @NLO, for development only)
                        nl ( = NLO = q q~ @LO + NLO + gq)
                        VH_PC overrides Pchannel.
    alpha_dip           extra non-physical degree of freedom for Process=51 & VH_PC=nl, defaulted at 1.
                        Vary to check indepedence (of alpha_dip).
    VBFoffsh_run:       For EW off-shell or QCD continuum productions, set this to an index
                        for each of the jobs.  See manual for more details.
Resonance parameters:
  MReso:                resonance mass in GeV (default=125.00)
  GaReso:               resonance width in GeV (default=0.00407)
  ctauReso:             resonance decay length in mm (default=0)
  OffshellX:            Whether to allow resonance (X) to go off-shell
                        in processes 0, 1 or 2
```

```
  MZ:               Z mass in GeV (default=91.1876)
  GaZ:              Z width in GeV (default=2.4952)
  MW:               W mass in GeV (default=80.399)
  GaW:              W width in GeV (default=2.085)
  MReso2:           2nd resonance mass in GeV in off-shell EW
  GaReso2:          2nd resonance width in GeV in off-shell EW
EW coupling parameters:
  Vud:              CKM element for W-ud couplings
  Vus:              CKM element for W-us couplings
  Vub:              CKM element for W-ub couplings
  Vcd:              CKM element for W-cd couplings
  Vcs:              CKM element for W-cs couplings
  Vcb:              CKM element for W-cb couplings
  Vtd:              CKM element for W-td couplings
  Vts:              CKM element for W-ts couplings
  Vtb:              CKM element for W-tb couplings
Cuts:
  pTjetcut:         Minimum pT for jets in GeV (default: 15)
  deltaRcut:        Minimum deltaR for jets (default: 0.3)
  mJJcut:           Minimum dijet mass in GeV (default: 0)
  MPhotonCutoff:    Minimum mass for off-shell photons in GeV, when included (default: 4)
  etajetcut:        Maximum |eta| for jets in off-shell EW (default: 4)
  detajetcut:       Minimum deltaeta between jets in off-shell EW (default: 2)
  JetsOppositeEta:  Require sgn(eta) to be opposite for the two jets in off-shell EW
                    (default: true)
  pTlepcut:         Minimum pT for leptons in off-shell EW, in GeV (default: 3)
  etalepcut:        Maximum |eta| for leptons in off-shell EW (default: 2.7)
  m4l_min, m4l_max: Minimum and maximum four-lepton mass in off-shell EW
  m2l_min:   Minimum invariant mass of V (on-shell) in new VH (\texttt{Process=51}) (default: 0)
  m2l_max:   Maximum invariant mass of V (on-shell) in new VH (\texttt{Process=51}) (default: infinity)
  mVH_min:   Minimum invariant mass of VH in new VH (\texttt{Process=51}) (default: 0)
  mVH_max:   Maximum invariant mass of VH in new VH (\texttt{Process=51}) (default: infinity)
Renormalization and factorization scales:
  FacScheme:        PDF factorization scale scheme
  MuFacMultiplier:  Multiplier for the factorization scale chosen by FacScheme
  RenScheme:        QCD renormalization scale scheme
  MuRenMultiplier:  Multiplier for the renormalization scale chosen by RenScheme
Lepton and jet filter:
  NLepMin:          Reject events that have less than this many charged leptons
  NLepMax:          Reject events that have more than this many charged leptons
  NJetMin:          Reject events that have less than this many jets
  NJetMax:          Reject events that have more than this many jets
  NOSMin:           Reject events that have less than this many
                    opposite-sign pairs of leptons of any flavor
  NOSMax:           Reject events that have more than this many
                    opposite-sign pairs of leptons of any flavor
  NOSSFMin:         Reject events that have less than this many
                    opposite-sign same-flavor pairs of leptons
  NOSSFMax:         Reject events that have more than this many
                    opposite-sign same-flavor pairs of leptons
  CountTauAsAny:    For OSSF pairs, taus can stand in place of electrons or muons
                    of the same charge.
  WriteFailedEvents: Write events that fail in the LHE file, but with a weight of 0
                    (off by default)
Higgs propagator and decay width:
  WidthScheme:      Higgs width scheme: 1 for running width, 2 for fixed width (default),
                    and 3 for the CPS
  WidthSchemeIn:    For decay mode, reweight from one propagator to another by setting
```

```
                         WidthScheme and WidthSchemeIn to different values
      ReweightDecay:     For decay mode, reweight input decay by the decay probability
      PmHstarEvals:      For ReweightDecay, number of evaluations per mass point (default: 200000)
      ReadPmHstar:       For ReweightDecay, read the decay probability distribution from a file
      PmHstarFile:       File to write and read the decay probability distribution
   Statistics options:
      VegasNc0:          number of evaluations for integrand scan
      VegasNc1:          number of evaluations for accept-reject sampling
      VegasNc2:          number of events for accept-reject sampling
      ReadCSmax:         Read the results of the grid generation step from a file
      CSmaxFile:         File to use for reading (if ReadCSmax is set) or writing (otherwise)
                         the results of the grid generation step.  Depending on the process,
                         suffixes are appended to this base name. (default: DataFile without .lhe)
      Seed:              Random seed for event generation
   I/O options:
      Unweighted:        0=weighted events, 1=unweighted events
      WriteWeightedLHE:  For Unweighted=0, write weighted events to an LHE file
                         (note that the output could be huge)
      DataFile:          LHE output file
      ReadLHE:           LHE input file from external file (only spin-0)
      ConvertLHE:        Convert decay of the V from VH production.
                         Use DecayMode1 to specify the decay.
                         (should be a Z or W mode, depending on the input file)
      UnformattedRead:   Turn this on if the normal, faster reading fails
      VegasVerbosity:    Turn on or off verbose vegas printout (1: on, 0: off; default: 1)
   Couplings:
      See manual for the full list
```

2. Process configuration

- `DecayMode1,2`:

  - `DecayMode1,2=7`: Valid for spin-0 and spin-2 or for VH.

  - In VH production and `ConvertLHE` mode, `DecayMode1` is used for the decay of the V. If it is a Z decay mode, ZH will be produced; if it is a W decay mode, WH will be produced; if it is 7, γH will be produced, with γ stable.

  - In ttH production, if `TopDK` is true, `DecayMode1` and `DecayMode2` are used to decay the W bosons from the top decay. Only W decay modes are allowed.

  - In $t + H$ production, if `TopDK` is true, `DecayMode1` is used to decay the W boson from the top decay.

  - In $t + H + W$ production, if `TopDK` is true, `DecayMode1` is used to decay the W boson from the top decay while `DecayMode2` is used to decay the associated W boson.

  - In $H \to \tau\tau$ decay, if `TauDK` is true, `DecayMode1,2` are used to decay the W bosons from the τ decays.

- `Interf`: For final states with 4 same flavor fermions, this parameter controls whether or not to include interference effects.

- `ReweightInterf`: For off-shell EW production, this includes the 4 lepton interference in the event weights. We recommend setting this option instead of `Interf` because the phase space integration is much more efficient without interference. This option is not an approximation: it is mathematically correct as far as the `.lhe` files go. However, the `.dat` files produced will not be filled correctly with this option.

- `RandomizeVVdecays`: For VV decay (and similarly in ttH production), by default the V's are randomized so that, for example, `DecayMode1=5 DecayMode2=11` will produce all combinations that include at least one hadronic W decay (rather of specifically hadronic decay of the W^+). This can be switched off by setting `RandomizeVVdecays=0`. In this case, `DecayMode1` will refer specifically to the W^+ and `DecayMode2` to the W^-. For ZZ decay, this parameter only determines whether the decays of first and second Z written to the LHE file are randomized or not; in this case the only effect is the appearance of the output.

- `PChannel`: This parameter is only meaningful in the spin-2 case and for the ttH and bbH production mechanisms. For spin-0, production is possible only via the gg process; for spin-1 and for the VBF, VH, and tH processes, production is only possible via the qq' process. For $H + jj$ and $H + j$, all parton channels gg, qq', and qg are included.

- `ChannelRatio`: In the case when `PChannel=2` for a spin-2 resonance, this parameter determines an approximate fraction of the events that should come from $q\bar{q}$. This is equivalent to appropriately scaling the $Xq\bar{q}$ couplings with respect to the Xgg ones, but can be done without advance knowledge of the relative cross sections.

- `PDFSet`, `LHAPDF`, `LHAPDFMem`: To use LHAPDF, this functionality must be enabled in the makefile and the relevant environment variables must be set. See the instructions in Sec. V H. Otherwise `PDFSet` gives the choice between a more limited number of PDF sets.

3. Process-specific configuration

a. $gg \to ZH$ Process 51 is a new implementation of VH with matrix elements expressed with the spinor-helicity formalism, including LO $q\bar{q} \to V \to VH$ (currently in beta, Process 50 should be used instead), one-loop induced LO $gg \to ZH$, and $q\bar{q} \to V \to VH$ at NLO. The initial state, order of calculation, and diagrams (triangle, box, both, or the interference term between the two, for gg) is set by the parameter `VH_PC`, with its options listed above. The γH process, i.e. `DecayMode1=7`, is not yet implemented for `Process=51`.

b. *Off-shell Higgs production in VBF/VH.* Processes 66 (signal), 67 (background), and 68 (signal+background+interference) provide an interface to a modified version of MCFM [9] to off-shell EW production $q\bar{q} \to q\bar{q} + VV(\to 4\ell)$, including all anomalous couplings. $V(\to q\bar{q}) + H$ and the equivalent VVV background topologies are part of the background. We note that only $VV = ZZ(\to 4\ell, \ell = e, \mu, \tau)$ is available and tested. Final states with $VV = WW(\to 2\ell 2\nu)$ and other $VV = ZZ \to$non-4ℓ final states can be generated but are in a beta version that is not fully tested. For these processes, a second spin-zero resonance can be activated by setting `MReso2` and `GaReso2` to nonzero values. The second resonance's couplings can be set to arbitrary values, similar to the first resonance's. These couplings are denoted as `gh2*`.

Note that the convention of the `ghz*/ghw*` couplings differs such that the SM value of `ghz1` and `ghw1` are defined as 1, rather than 2 as defined in other processes (they becomes essentially scaling factors with respect to SM). This also means that all the couplings (not only `ghz1` and `ghw1`) which affect HVV interactions have to be scaled down by a factor of 2 coherently, with respect to what is expected in Eqs. (1) or (2). This affects only the direct HVV couplings in Eqs. (1) or (2) and the origin of this convention is in the MCFM convention for `ghz1` and `ghw1` (g_1^{ZZ} and g_1^{WW}) couplings in the off-shell matrix elements. The relative size of all HVV couplings in our framework is preserved this way. It is important to note that the continuum background matrix elements are not affected by the above convention and all the SM and anomalous couplings discussed on page 18 section IV A 10 c in application to the background parameterization do not need to be rescaled, even if they are expressed though the HVV couplings.

When producing unweighted EW off-shell events, jobs must be submitted in two steps, with 164 jobs per step, each job corresponding to a particular mother-associated parton combination. Each set of jobs can be run in parallel, but none of the jobs in the the second set can start until all jobs in the first set are completed.

1. Grid generation (`VBFoffsh=1..164`):

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=1`

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=2`

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=3`

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=4`

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=5`

2. Event generation, which can only begin after all previous jobs have ended:

   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=1 VegasNc2=10000 ReadCSmax`

- `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=2 VegasNc2=10000 ReadCSmax`

- `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=3 VegasNc2=10000 ReadCSmax`

- `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=4 VegasNc2=10000 ReadCSmax`

- `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 ReweightInterf=0 ghz1=1d0,0d0 VBFoffsh_run=5 VegasNc2=10000 ReadCSmax`

Similarly, processes 70 (signal), 71 (background), and 72 (signal+background+interference) generate off-shell EW production $VV \to VV$ in association with a W or a Z boson decaying leptonically. However, these processes are still in a state of an untested beta version. The same couplings for processes 66 or 68 can be used for 70 or 72. These processes require 84 partonic channels in the first grid generation step instead of 164. In addition, process 69 implements an untested beta version of QCD production of $VV(\to 4f)$ in association with two partons, but this process only includes the continuum production without Higgs contribution. There are 175 partonic channels associated with this process.

The final result will be as many `.lhe` files as the number of partonic channels, each containing events from that particular partonic channel. The number of events in each file will be proportional to the cross section of those channels. The total number of events in the five files may be larger or smaller than the number of events requested. If you want the exact number of events, use the same random seed five all five jobs of the second step.

It is also possible to reuse the output of the first step to run more jobs in the second step. For example, to run 1 million events using 400 jobs, you can run the first step once, then run 100 sets of 4 jobs each, using 100 different random seeds.

c. Off-shell Higgs production in gluon fusion. The MCFM interface also allows the production of $ZZ/WW \to 4f$ (without any other associated particles) through gluon fusion with infinite- or finite-mass quarks in the ggH interaction. The corresponding processes are 73 (signal), 74 (background), and 75 (signal+background+interference). These three processes are still in a state of an untested beta version. (In the meantime, these processes are supported with the matrix elements extended for anomalous couplings within the JHUGen framework and integrated into the MCFM framework for simulation of events.) Only a single partonic channel, gg, is relevant for this process. The second resonance parameters can be used in this process as well. In addition, the ggH vertex can be modified via `gh*g*` or `gh*g*_4gen` for two separate couplings per resonance for point-like ggH interactions, or `kappa*_*_top` and `kappa*_*_bot`, or `kappa*_tilde*_4gen_top` and `kappa*_tilde*_4gen_bot` for a third-generation top and a bottom quark, or a fourth-generation top or bottom quark in the loop, respectively, again separately implemented for each resonance. For the fourth-generation interactions, the corresponding masses `m_top_4gen` and `m_bot_4gen` need to be set and are common to both resonances. The masses modifying the third generation quark loops are `m_top` and `m_bot`.

4. Resonance parameters

- `OffshellX`: The flag specifies whether the events for processes 0, 1 or 2 feature a delta-function resonance mass (when set to 0) at or a finite-width distribution around the resonance mass (when set to 1). This option replaces the `OffXVV` numerical flag in previous versions.

5. Cuts

- `pTjetcut`, `etajetcut`, `deltaRcut`, `mJJcut` All of these cuts are used in processes 60 (VBF), 61 ($H + jj$), 90 (bbH), 66-75 (off-shell), and also 80 (ttH) if `m_Top` is set to be light (< 10GeV). In addition, `pTjetcut` is used in process 62 ($H + j$) and also processes 110–113 ($t/\bar{t} + H$) if `m_Top` is light.

- `pTlepcut`, `etalepcut` similarly adjust the p_T or $|\eta|$ of leptons.

- `MPhotonCutoff` If the decay is to ZZ and any of the couplings to photons (`ghzgsX` or `ghgsgsX`, or continuum amplitudes with virtual photons) is set, the γ^* contribution will be included along with the Z's. Similarly, if the decay is to $Z\gamma$ and any of the `ghgsgsX` couplings are set, the $\gamma\gamma^*$ contribution will be included as well. In this case, a lower cut on the photon invariant mass has to be placed in order to avoid the collinear singularity. This can be set using MPhotonCutoff.

- `pTjetcut`, `etajetcut`, `detajetcut`, `deltaRcut`, `mJJcut`, `m4l_min`, `m4l_max`, `m2l_min`, `m2l_max`, `mVH_min`, `mVH_max`, `pTlepcut`, `etalepcut`, and `pTHcut` are available for Process 51. The default values are set such that there is no cuts.

6. Renormalization and factorization scales

- `FacScheme`, `MuFacMultiplier`, and `RenScheme`, `MuRenMultiplier`: There are currently 10 different schemes, which set the basis of the scale up to the scale multiplier. A postive integer uses running scales per event whereas a negative one uses a fixed scale, and `MuFacMultiplier` and `MuRenMultiplier` determine the scale multipliers for the factorization and renormalization scales, respectively:

  - ± 0: $\mu_{F,R}$ are set to the default values of each process. The command line values of `MuFacMultiplier` and `MuRenMultiplier` are disregarded.

  - ± 1: $\mu_{F,R} \propto \sqrt{q_H^2}$ if the scheme number is positive, or $\mu_{F,R} \propto m_H$ if the scheme number is negative. $+1$ is the default value for `Process` 0, 1 and 2 with scale multiplier 0.5, and -1 is the default value for `Process` 50, 60, 61 and 62 with scale multiplier 1.

  - ± 2: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{JJH}^2}$, where J refers to a particle (partons or leptons) immediately associated in the Higgs production. If the scheme number is negative and the Higgs and the associated partons originate from a common particle (e.g. V^* in VH), $\mu_{F,R} \propto m_{JJH}$.

  - ± 3: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{JJ}^2} + \sqrt{q_H^2}$. If the scheme number is negative and the associated particles originate from a common particle without the Higgs (e.g. V in VH), $\mu_{F,R} \propto m_{JJ} + m_H$.

  - ± 4: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{J1}^2} + \sqrt{q_{J2}^2} + \sqrt{q_H^2}$, where $J_{1,2}$ are the associated particles. If the scheme number is negative and the matrix element treats the associated particles as massive (e.g. t and $\bar{t}$ in $t\bar{t}H$), $\mu_{F,R} \propto m_{J1} + m_{J2} + m_H$. -4 is the default value for `Process` 80 and 90 with scale multiplier 0.5.

  - ± 5: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{JJ}^2}$. If the scheme number is negative and the associated particles originate from a common particle without the Higgs (e.g. V in VH), $\mu_{F,R} \propto m_{JJ}$.

  - ± 6: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{J1}^2} + \sqrt{q_{J2}^2}$, where $J_{1,2}$ are the associated particles. If the scheme number is negative and the matrix element treats the associated particles as massive (e.g. t and $\bar{t}$ in $t\bar{t}H$), $\mu_{F,R} \propto m_{J1} + m_{J2}$.

  - ± 7: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_{JH}^2}$, where J is the more massive associated particle by its pole mass. If the scheme number is negative and the matrix element treats an associated particle as massive (e.g. t in $t + H$), $\mu_{F,R} \propto m_J + m_H$ for the most massive associated particle.

  - ± 8: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_J^2} + \sqrt{q_H^2}$, where J is the more massive associated particle. If the scheme number is negative and the matrix element treats an associated particle as massive (e.g. t in $t + H$), $\mu_{F,R} \propto m_J + m_H$ for the most massive associated particle (same as option -7). -8 is the default value for `Process` 110, 111, 112 and 113. with scale multiplier 0.25.

  - ± 9: If the scheme number is positive, $\mu_{F,R} \propto \sqrt{q_J^2}$, where J is the more massive associated particle. If the scheme number is negative and the matrix element treats an associated particle as massive (e.g. t in $t + H$), $\mu_{F,R} \propto m_J$.

  - $+10$: $\mu_{F,R} \propto p_T^J$, where J is the hardest associated jet.

  - $+11$: $\mu_{F,R} \propto p_T^J$, where J is the softest associated jet.

7. Lepton and jet filter

- Any combination of `NLepMin/Max`, `NOSMin/Max`, `NOSSFMin/Max` can be used in ReadLHE mode to filter the events. `CountTauAsAny`, off by default, counts each tau as an electron or muon of the same sign towards an OSSF pair.

- `NJetMin/Max` can be used alone or in conjunction with the lepton filter (e.g. request two leptons and a jet). It counts each final state quark or gluon as a jet.

- **WriteFailedEvents**: By default, events that fail the lepton filter are not written to the LHE file. If this option is set to 1, they are written, but their weight is set to 0. If it is set to 2, they are written with a weight of zero and with no particles in the event (this saves disk space with and is accepted by Pythia, but some parsers may not like it). This option is useful if the same number of events are required to be in the input and the output.

8. Higgs propagator and decay width

- **WidthScheme**: This option controls the width scheme in JHUGen. The options are 1 for running width, 2 for fixed width, and 3 for the complex pole scheme [12, 13]. 2 is the default option. Note that the parameter values have the same meaning as in POWHEG.

- **WidthSchemeIn**: In **ReadLHE** mode, if this is specified and is different than **WidthScheme**, the resonance propagator will be reweighted between the three options specified above by multiplying the LHE weight of each event by the ratio of the propagators.

- **ReweightDecay**: By default, in **ReadLHE** mode, output events are written with the same weight as in the input file. For a wide resonance, this is not fully correct. The full probability for a $gg \to X \to VV \to 4f$ event can be written as

$$P(gg \to X \to VV \to 4f) = P_{prod}(m_{4f}, \vec{\Omega}_{prod}) \times P_{dec}(m_{4f}) \times P_{dec}(\vec{\Omega}_{decay} \mid m_{4f})$$

While the first factor is generated by the input generator and the third by JHUGen, by default the second factor, the decay contribution to the mass shape, is lost. If **ReweightDecay=1**, this term will be multiplied into the weight of the event written in the LHE file.

  - If POWHEG was generated with a fixed width line shape, the full $P_{dec}(m_{4f})$ is multiplied into the weight.
  - If POWHEG was generated with the CPS option, then $m_{4f}\Gamma_H$, which is proportional to $P_{X \to \text{anything}}(m_{4l})$, is already included in the line shape. This is not exactly correct, as what is needed is the more specific $P_{X \to (\text{decay mode})}$. The weight is therefore multiplied by the branching fraction.

To select one of these options, POWHEG input decay mode is specified by **WidthSchemeIn**. Alternatively, JHUGen will try to read the parameter **bwshape** from the header of the POWHEG file. Using other input generators requires an advance knowledge of how the line shape is generated; if neither of these options applies it may be simpler to apply a similar weight after the fact rather than in the LHE file.

  - **PmHstarEvals**: Controls the number of evaluations per mass point. The default is 200000 at high masses, corresponding to a precision of about 0.2%, but increases at lower **m_Reso**, where the offshell Z's make the integration less efficient.
  - **PmHstarFile**: After each run using **ReweightDecay**, the distribution is written to a file specified by this option (default: **PMZZdistribution.out**).
  - **ReadPmHstar**: Read in distribution from **PmHstarFile** instead of calculating it again. This only makes sense if the **DecayModes**, couplings, etc. are the same as last time; otherwise you will read a distribution that no longer makes sense.

9. Statistics options

  - **VegasNc0,1,2**: For unweighted event generation **VegasNc0** specifies the number of evaluations for the initial integrand scan. The actual event generation is controlled by either **VegasNc1** or **VegasNc2**. **VegasNc1** specifies the number of tries in the accept/reject phase and **VegasNc2** is the number of generated events. When generating unweighted events in **ReadLHE** mode, both **VegasNc1** or **VegasNc2** can be used to specify the number of generated events. For the generation of weighted events **VegasNc1** specifies the number of evaluations for each of 5 iterations during the initial integrand scan. **VegasNc2** gives the (approximate) number of generated weighted events.
  - **ReadCSmax**: If **ReadCSmax** is specified on the command line, the grid generation step is skipped and the results are read from a file. This can speed up the process by running a short job first, and then many jobs in parallel using this option. This only makes sense if the couplings, etc. are the same as last time; otherwise your results won't make sense.

– `Seed`: To reproduce previous results, the random seed can be set on the command line. Because Fortran uses multiple random seeds (the exact number is compiler dependent), the seed provided is used along with other fixed seeds specified in `mod_Parameters.F90` to generate however many seeds are needed. If no seed is provided on the command line, the seed is determined from the system time and process id. Either way, it is printed in the header of the output file so that the results can be reproduced.

Please note that random number generation is compiler dependent, so to ensure reproducibility the same compiler (and preferably the same version of the compiler) should be used. To facilitate this, the compiler and version are written in the header of the LHE file. It is also not guaranteed that the results can be reproduced using a different version of the generator.

10.   Couplings

The general syntax is:

– `ghgX`, `ghzX`, `ghzX_primeY`, `cz_qXsq`, `Lambda_zXY`, `ghwX`, `ghwX_primeY`, `cw_qXsq`, `Lambda_wXY`, `ghzgsX`, `ghzgs1_prime2`, and `ghgsgsX` for spin 0 HVV couplings (see the list for the ranges of X and Y)

– `zprime_qq_left`, `zprime_qq_right`, `zprime_zz_1`, and `zprime_zz_2` for spin 1

– `aX`, `bX`, `graviton_qq_left`, and `graviton_qq_right` for spin 2 (see the list for the range of X)

– The Hff couplings `kappa` and `kappa_tilde` and variants

– The Zff couplings $a_{\mathrm{L,R}}^{f}$ in Eq. (21) can be modified with corrections to their SM values as `daz_Lep_left`, `az_Lep_right`, `daz_Neu_left`, `az_Neu_right`, `daz_QUp_left`, `daz_QUp_right` and `daz_QDn_left`, `daz_QDn_right`. For example, a_{L}^{ℓ} is set as `aL_lep + daz_Lep_left`

Except for the form factors (`cz_qXsq`, `Lambda_zXY`, and their WW equivalents), the couplings are complex, and must be set with a comma between the real and imaginary parts. For example:

`ghz1=0,0 ghz4=1,0`

If an anomalous coupling is turned on, the default coupling in the same category must also be specified to leave it on or turn it off. For example, if `ghz4` is specified, `ghz1` must either be set to `0,0` to generate a pure pseudoscalar or to some other value to generate a CP-violating mixture. In the weak vector boson fusion process (VBF) we also allow for different ZZH and WWH couplings. Per default, they are assumed to be equal, set by the Z couplings; however, if any of the W couplings are set, they will be used instead for WWH fusion. WWH can be turned off entirely by explicitly setting one of them to `0,0`. For the Hff couplings, one can also specify `kappa_top` or `kappa_tilde_top` for the processes that involve Htt coupling, and `kappa_bot` or `kappa_tilde_bot` for the processes that involve Hbb coupling. Setting `kappa` and `kappa_tilde` instead of these sets of couplings sets all relevant Hff couplings to the same value.

a.   Spin-0 parameters   The `ghg*` parameters control the coupling of a spin-0 resonance to gluons in the production mechanism. In practice, the production parameters do not have a large effect on Higgs decay since angular corrections from the production mechanism are lost for spinless particles. The `ghz*` parameters control the decay into Z and W bosons. These spin-0 couplings are described in Eq.(9) from Ref. [2], and can be set from the command line. We allow for q^2 dependent form factors similar to those described in Ref. [3]:

$$
\begin{aligned}
g_i^{\Lambda}(q_1, q_2) \;=\;& g_i' \frac{\Lambda_i^4}{(\Lambda_i^2 + |q_1^2|)(\Lambda_i^2 + |q_2^2|)} \\
&+\; g_i'^2 \frac{(q_1^2 + q_2^2)}{\Lambda_i^2} + g_i'^3 \frac{(q_1^2 - q_2^2)}{\Lambda_i^2} + g_i'^4 \frac{(q_1 + q_2)^2}{\Lambda_Q^2} \\
&+\; g_i'^5 \frac{\left((q_1^2)^2 + (q_2^2)^2\right)}{\Lambda_i^4} + g_i'^6 \frac{\left((q_1^2)^2 - (q_2^2)^2\right)}{\Lambda_i^4} + g_i'^7 \frac{q_1^2 \, q_2^2}{\Lambda_i^4} \\
g_i(q_1, q_2) \;=\;& g_i \delta_{i1} + \left(g_i^{\Lambda}(q_1, q_2) + g_i(1 - \delta_{i1})\right) \frac{\Lambda_{i1}^2 \, \Lambda_{i2}^2 \, \Lambda_{i3}^2}{(\Lambda_{i1}^2 + c_{i1}|q_1^2|)(\Lambda_{i2}^2 + c_{i2}|q_2^2|)(\Lambda_{i3}^2 + c_{i3}|(q_1 + q_2)^2|)}
\end{aligned}
$$

The user has the option to choose between these functional forms, where the term multiplying g_i' corresponds to the full functional form and the $g_i'^2...g_i'^7$ correspond to an expansion in Λ^2. The names and default values of these parameters are found in `mod_Parameters.F90`. To set them on the command line, use, for example, `ghz1=1,0`, where 1 represents the real part and 0 represents the imaginary part.

```
!-- parameters that define offshell spin 0 coupling to SM fields, see note
complex(8), public :: ghg2 = (1.0d0,0d0)
complex(8), public :: ghg3 = (0.0d0,0d0)
complex(8), public :: ghg4 = (0.0d0,0d0)    ! pseudoscalar

complex(8), public :: ghz1 = (2.0d0,0d0)    ! =1 for off-shell processes in SM
complex(8), public :: ghz2 = (0.0d0,0d0)
complex(8), public :: ghz3 = (0.0d0,0d0)
complex(8), public :: ghz4 = (0.0d0,0d0)    ! pseudoscalar


!-- parameters that define q^2 dependent form factors
complex(8), public :: ghz1_prime = (0.0d0,0d0)
complex(8), public :: ghz1_prime2= (0.0d0,0d0)
complex(8), public :: ghz1_prime3= (0.0d0,0d0)
complex(8), public :: ghz1_prime4= (0.0d0,0d0)
complex(8), public :: ghz1_prime5= (0.0d0,0d0)
complex(8), public :: ghz1_prime6= (0.0d0,0d0)
complex(8), public :: ghz1_prime7= (0.0d0,0d0)

complex(8), public :: ghz2_prime = (0.0d0,0d0)
complex(8), public :: ghz2_prime2= (0.0d0,0d0)
complex(8), public :: ghz2_prime3= (0.0d0,0d0)
complex(8), public :: ghz2_prime4= (0.0d0,0d0)
complex(8), public :: ghz2_prime5= (0.0d0,0d0)
complex(8), public :: ghz2_prime6= (0.0d0,0d0)
complex(8), public :: ghz2_prime7= (0.0d0,0d0)

complex(8), public :: ghz3_prime = (0.0d0,0d0)
complex(8), public :: ghz3_prime2= (0.0d0,0d0)
complex(8), public :: ghz3_prime3= (0.0d0,0d0)
complex(8), public :: ghz3_prime4= (0.0d0,0d0)
complex(8), public :: ghz3_prime5= (0.0d0,0d0)
complex(8), public :: ghz3_prime6= (0.0d0,0d0)
complex(8), public :: ghz3_prime7= (0.0d0,0d0)

complex(8), public :: ghz4_prime = (0.0d0,0d0)
complex(8), public :: ghz4_prime2= (0.0d0,0d0)
complex(8), public :: ghz4_prime3= (0.0d0,0d0)
complex(8), public :: ghz4_prime4= (0.0d0,0d0)
complex(8), public :: ghz4_prime5= (0.0d0,0d0)
complex(8), public :: ghz4_prime6= (0.0d0,0d0)
complex(8), public :: ghz4_prime7= (0.0d0,0d0)

real(8),    public, parameter :: Lambda_z1 = 10000d0*GeV
real(8),    public, parameter :: Lambda_z2 = 10000d0*GeV
real(8),    public, parameter :: Lambda_z3 = 10000d0*GeV
real(8),    public, parameter :: Lambda_z4 = 10000d0*GeV
real(8),    public, parameter :: Lambda_Q  = 10000d0*GeV

integer,    public :: cz_q1sq = 0d0 ! Sign of q1,2,12**2 for the following Lambda's
integer,    public :: cz_q2sq = 0d0
integer,    public :: cz_q12sq = 0d0

real(8),    public :: Lambda_z11 = 100d0*GeV ! For Z1
```

```
real(8),    public :: Lambda_z21 = 100d0*GeV
real(8),    public :: Lambda_z31 = 100d0*GeV
real(8),    public :: Lambda_z41 = 100d0*GeV
real(8),    public :: Lambda_z12 = 100d0*GeV ! For Z2
real(8),    public :: Lambda_z22 = 100d0*GeV
real(8),    public :: Lambda_z32 = 100d0*GeV
real(8),    public :: Lambda_z42 = 100d0*GeV
real(8),    public :: Lambda_z10 = 100d0*GeV ! For the Higgs
real(8),    public :: Lambda_z20 = 100d0*GeV
real(8),    public :: Lambda_z30 = 100d0*GeV
real(8),    public :: Lambda_z40 = 100d0*GeV
```

Intermediate offshell photons can be included along with the Z boson for Z boson final states, ZH production, and VBF production. Their couplings to the spin-0 resonance are controlled by separate parameters,

```
complex(8), public :: ghzgs2  = (0.00d0,0d0)
complex(8), public :: ghzgs3  = (0.00d0,0d0)
complex(8), public :: ghzgs4  = (0.00d0,0d0)
complex(8), public :: ghgsgs2 = (0.00d0,0d0)
complex(8), public :: ghgsgs3 = (0.00d0,0d0)
complex(8), public :: ghgsgs4 = (0.00d0,0d0)
```

where the first three correspond to $Z\gamma^*$ couplings and the latter three corresponds to $\gamma^*\gamma^*$ interactions. These two sets of parameters also control the coupling strength in final states with on-shell photons, i.e. $H \to Z\gamma$ or $H \to \gamma\gamma$ decays and $Z^*/\gamma^* \to \gamma H$ production. The anomalous coupling involving the offshell photon momentum (in γ^*Z interactions)

$$g_1' \frac{q_\gamma^2}{\Lambda_1^{Z\gamma}} m_Z^2 \epsilon_1^* \epsilon_2^*$$

is set by

```
complex(8), public :: ghzgs1_prime2= (0.0d0,0d0)
real(8),    public, parameter :: Lambda_zgs1 = 10000d0*GeV.
```

Each `ghzX`, `ghzX_primeY`, `Lambda_zXY`, and `cz_qXsq` parameter has a `ghwX`, `ghzwX_primeY`, `Lambda_wXY`, and `cw_qXsq` counterpart, which are used for the WWH coupling in VBF if at least one of them is set on the command line. Similarly, `gh2zX`, `gh2zX_primeY`, `Lambda2_zXY`, `c2z_qXsq`, `gh2wX`, `gh2wX_primeY`, `Lambda2_wXY`, and `c2w_qXsq` can be used to set the couplings of the second resonance in off-shell EW production.

   b.   *Contact terms and V' couplings*   JHUGen also supports a spin zero resonance coupling to a new spin-1 boson, Z' or W', in VBF and VH production and in decay. These couplings are set by parameters equivalent to the HZZ couplings. For example,

```
complex(8), public :: ghzzp1 = (0d0,0d0)
complex(8), public :: ghzpzp1 = (0d0,0d0)
complex(8), public :: ghzpgs2 = (0d0,0d0)
```

give couplings for HZZ', $HZ'Z'$, and $HZ'\gamma$ that are equivalent to `ghz1` or `ghzgs2`. All supported couplings for HZZ have equivalent couplings for HZZ' and $HZ'Z'$, and all supported couplings for $HZ\gamma$ have equivalent couplings for $HZ'\gamma$.
   As in the WW case, by default the HWW' and $HW'W'$ couplings are assumed to be the same as HZZ' and $HZ'Z'$. In the VBF production, the equivalent couplings `ghwwp*` and `ghwpwp*` can be set. If any of the HWW, HWW', or $HW'W'$ couplings are set explicitly, $HW(')W(')$ is completely separated from $HZ(')Z(')$, and the specified couplings are used instead.
   In both Z' and W' cases, the M_V parameter in Eq. 1 is set to M_Z and M_W, respectively, in order to avoid assumptions for a particular mass matrix. The user needs to set the couplings accordingly in order to their particular model.
   The couplings between V' and fermions are set by

```
complex(8), public :: ezp_El_left   = (0d0,0d0)
complex(8), public :: ezp_El_right  = (0d0,0d0)
complex(8), public :: ezp_Mu_left   = (0d0,0d0)
complex(8), public :: ezp_Mu_right  = (0d0,0d0)
complex(8), public :: ezp_Ta_left   = (0d0,0d0)
complex(8), public :: ezp_Ta_right  = (0d0,0d0)
complex(8), public :: ezp_NuE_left  = (0d0,0d0)   !same for NuMu and NuTau
complex(8), public :: ezp_NuE_right = (0d0,0d0)   !same for NuMu and NuTau
complex(8), public :: ezp_Up_left   = (0d0,0d0)
complex(8), public :: ezp_Up_right  = (0d0,0d0)
complex(8), public :: ezp_Chm_left  = (0d0,0d0)
complex(8), public :: ezp_Chm_right = (0d0,0d0)
complex(8), public :: ezp_Dn_left   = (0d0,0d0)
complex(8), public :: ezp_Dn_right  = (0d0,0d0)
complex(8), public :: ezp_Str_left  = (0d0,0d0)
complex(8), public :: ezp_Str_right = (0d0,0d0)
complex(8), public :: ezp_Bot_left  = (0d0,0d0)
complex(8), public :: ezp_Bot_right = (0d0,0d0)

complex(8), public :: ewp_El_left   = (0d0,0d0)
complex(8), public :: ewp_El_right  = (0d0,0d0)
complex(8), public :: ewp_Mu_left   = (0d0,0d0)
complex(8), public :: ewp_Mu_right  = (0d0,0d0)
complex(8), public :: ewp_Ta_left   = (0d0,0d0)
complex(8), public :: ewp_Ta_right  = (0d0,0d0)
complex(8), public :: ewp_Up_left   = (0d0,0d0)
complex(8), public :: ewp_Up_right  = (0d0,0d0)
complex(8), public :: ewp_Chm_left  = (0d0,0d0)
complex(8), public :: ewp_Chm_right = (0d0,0d0)
```

The values of `ezp*left` (`ezp*right`) couplings that give SM couplings can be set by $2\left(T_f^3 - Q_f \sin^2\theta_W\right)$ $(-2Q_f \sin^2\theta_W)$, where T_f^3 and Q_f are the weak isospin and charge of the fermion in consideration. With the similar conventions, the values of `ewp*left` (`ewp*right`) couplings that give SM couplings can be set by $\sqrt{2\left(1 - \sin^2\theta_W\right)}$ (0). The CKM mechanism is assumed to affect the $W'ff$ couplings the same way as it affects the Wff couplings.

By default, Z' and W' are assumed to have infinite mass. Under these conditions, setting `ghzzp1` and some of the $V'ff$ couplings to nonzero values is equivalent to the contact terms described in [14]. Z' (W') can also be given finite mass by setting its mass `MZprime` (`MWprime`) and width `GaZprime` (`GaWprime`) on the command line. The lifetime of the V' is determined by the `ctauVprime` option in units of milimeters. For cases where only the $V'V'$ couplings are set in the processes 0, 1, 2, 50 and 51, a nonzero decay length is recorded for all such V' difermion pairs. In all other cases, the factor `VprimeDecayLengthMassCutoffFactor` controls the extent of the invariant mass of the difermion pair in units of $\Gamma_{V'}$ in order to record a nonzero decay length; a value less than or equal to 0 for this factor assigns a decay length to all V' difermion pairs with a default value of 10.

c. Gauge boson self couplings Off-shell VBF processes 66 (signal), 67 (background), and 68 (signal+background+interference) involve triple and quartic gauge couplings in Eqs. (9-10). To set the couplings in

Eq. (11) on the command line use e.g.

$$d_1^\gamma \quad \leftrightarrow \quad \texttt{dVA}, \qquad d_2^\gamma \quad \leftrightarrow \quad \texttt{dPA}, \qquad d_3^\gamma \quad \leftrightarrow \quad \texttt{dMA},$$

$$d_1^Z \quad \leftrightarrow \quad \texttt{dVZ}, \qquad d_2^Z \quad \leftrightarrow \quad \texttt{dPZ}, \qquad d_3^Z \quad \leftrightarrow \quad \texttt{dMZ},$$

$$d_4^\gamma \times d^{\gamma WW} \quad \leftrightarrow \quad \texttt{dFourA},$$

$$d_4^Z \times d^{ZWW} \quad \leftrightarrow \quad \texttt{dFourZ},$$

$$d^{\gamma\gamma WW} \quad \leftrightarrow \quad \texttt{dAAWpWm},$$

$$\frac{s_w}{c_w} \times d^{Z\gamma WW} \quad \leftrightarrow \quad \texttt{dZAWpWm},$$

$$\frac{s_w^2}{c_w^2} \times d^{ZZWW} \quad \leftrightarrow \quad \texttt{dZZWpWm}.$$

For example, in order to simulate the hypothesis with

$$g_1^{ZZ} = 2.0, \quad g_1^{WW} = 2.0, \quad \kappa_1^{ZZ} = \kappa_2^{ZZ} = -10.0, \quad \kappa_1^{WW} = \kappa_2^{WW} = -18.6 \quad \kappa_2^{Z\gamma} = -15.68,$$
$$d_1^Z = d_2^Z = d_3^Z = 1.08, \quad d^{Z\gamma WW} = 1.96, \quad d^{ZZWW} = 3.84.$$

the command line should be

```
./JHUGen Process=68 ghz1=1.0,0.0 ghw1=1.0,0.0 ghz2=0.0,0.0 ghw2=0.0,0.0 ghz1_prime2=-5.0,0.0
        ghw1_prime2=-9.3,0.0 ghzgs1_prime2=-7.84,0.0 dV_A=1.0,0.0 dV_Z=1.08,0.0 dP_A=1.0,0.0
        dP_Z=1.08,0.0 dM_A=1.0,0.0 dM_Z=1.08,0.0 dZZWpWm=1.15,0.0 dZAWpWm=1.08,0.0 ...
```

Note that as discussed on page 11, the direct HVV couplings need to be divided by 2 in processes 66-68. However, when using these couplings to express the triple and quartic gauge couplings following Eqs. (15 - 20), no such division is necessary. In other words, the scaling convention applies only to direct HVV couplings configured in the generator, while it does not apply to TGC and QGC in any way. This convention may appear confusing at first, but should become transparent with careful consideration, and please check the example above as well. It is the unfortunate consequence of the SM convention $g_1^{ZZ} = g_1^{WW} = 1$ in the MCFM matrix element (as opposed to $g_1^{ZZ} = g_1^{WW} = 2$ elsewhere), which we would like to preserve for compatibility purpose.

d. Spin-1 parameters The parameters below represent the couplings given in Eq. (16) from Ref. [2]. The `zprime_qq_*` parameters control the production of the spin-1 resonance while the `zprime_zz_*` parameters control the decay.

```
!---parameters that define spin 1 coupling to SM fields, see note
complex(8), public :: zprime_qq_left  = (1.0d0,0d0)   ! see note Eq. (4)
complex(8), public :: zprime_qq_right = (1.0d0,0d0)
complex(8), public :: zprime_zz_1 =  (0.0d0,0d0)!  =1 for JP=1-
complex(8), public :: zprime_zz_2 =  (0.0d0,0d0)!  =1 for JP=1+
```

e. Spin-2 parameters The `a*` parameters control the coupling of a spin-2 resonance to gluons in the production mechanism. The `b*` parameters control the decay to either ZZ or WW (`b`), $Z\gamma$ (`zgs`), or $\gamma\gamma$ (`gsgs`). The spin-2 couplings are described by Eq.(18) from Ref. [2], and can be set on the command line.

```
complex(8), public :: a1 = (1.0d0,0d0)     ! g1  -- c.f. note
complex(8), public :: a2 = (0.0d0,0d0)     ! g2
complex(8), public :: a3 = (0.0d0,0d0)     ! g3
complex(8), public :: a4 = (0.0d0,0d0)     ! g4
complex(8), public :: a5 = (0.0d0,0d0)     ! pseudoscalar, g8

complex(8), public :: graviton_qq_left  = (1.0d0,0d0)! graviton coupling to quarks
complex(8), public :: graviton_qq_right = (1.0d0,0d0)

complex(8), public :: b1 = (0.0d0,0d0)
complex(8), public :: b2 = (0.0d0,0d0)
complex(8), public :: b3 = (0.0d0,0d0)
complex(8), public :: b4 = (0.0d0,0d0)
```

```
complex(8), public :: b5 = (0.0d0,0d0)
complex(8), public :: b6 = (0.0d0,0d0)
complex(8), public :: b7 = (0.0d0,0d0)
complex(8), public :: b8 = (0.0d0,0d0)
complex(8), public :: b9 = (0.0d0,0d0)
complex(8), public :: b10 =(0.0d0,0d0)

complex(8), public :: bzgs1 = (0.0d0,0d0)
complex(8), public :: bzgs2 = (0.0d0,0d0)
complex(8), public :: bzgs3 = (0.0d0,0d0)
complex(8), public :: bzgs4 = (0.0d0,0d0)
complex(8), public :: bzgs8 = (0.0d0,0d0)

complex(8), public :: bgsgs1 = (0.0d0,0d0)
complex(8), public :: bgsgs2 = (0.0d0,0d0)
complex(8), public :: bgsgs3 = (0.0d0,0d0)
complex(8), public :: bgsgs4 = (0.0d0,0d0)
complex(8), public :: bgsgs8 = (0.0d0,0d0)
```

As in the spin 0 case, decay to Z' is fully supported. Each of the parameters listed above for VV has an equivalent parameter that describes the spin 2 resonance's coupling to VV' and $V'V'$; examples include `bzzp1` and `bzpzp2`. Similarly, the parameters describing the decay to $Z\gamma$ have equivalent parameters, such as `bzpgs3`, that describe the decay to $Z'\gamma$. The parameters controlling the mass, width, and fermion couplings of Z' and W' are described above in Section IV A 10 b.

B.   Configuration in parameter file

Some more advanced parameters can be found in `mod_Parameters.F90`. These include SM parameters such as the masses of particles and electroweak couplings, which can be changed for precise syncing with other generators used, for example. Look for the `!constants` section.

`mod_Parameters.F90` also contains alternate schemes for setting the spin 0 and spin 2 couplings, specified by Eq. (11) and Eq. (20) of [2] instead of Eq. (9) and Eq. (18). To enable these couplings, set `generate_as` to `.true.` and set the `ahg*` and `ahz*` couplings for spin 0, or set `generate_bis` to `.false.` and set the `c*` couplings for spin 2.

Note that these alternate couplings only work for processes 0 and 2, not for other production modes that use spin 0 couplings.

Any modification to `mod_Parameters.F90` requires recompiling JHUGen. We recommend contacting the authors before modifying `mod_Parameters.F90`. If the functionality you want is useful and really cannot be achieved any other way, we can consider making it available as a command line option.

V.   EXAMPLES

The below examples are not meant to be a complete set, but rather some interesting and relevant cases. In many cases, the example is not the only way to produce such a scenario.

A. $J^P = 0_m^+$ resonance, $X \to ZZ$ or WW

```
./JHUGen ghz1=1,0 (...other options)
```

B. $J^P = 0_m^-$ resonance, $X \to ZZ$ or WW

```
./JHUGen ghz1=0,0 ghz4=1,0 (...other options)
```

C. $J^P = 0_m^+$ resonance, $X \to \gamma\gamma$

In practice, the example $X \to \gamma\gamma$ from this section, Sec. V C and the next Sec. V D are kinematically the same but are presented only to illustrate how one takes care of this final state. However, in the case of Sec. V E, the couplings chosen do affect the kinematics.

```
./JHUGen DecayMode1=7 DecayMode2=7 ghgsgs2=1,0 (...other options)
```

D. $J^P = 0_m^-$ resonance, $X \to \gamma\gamma$

```
./JHUGen DecayMode1=7 DecayMode2=7 ghgsgs4=1,0 (...other options)
```

E. $J^P = 0_m^-$ resonance, $X \to \gamma^*\gamma^* \to 4\ell$

```
./JHUGen DecayMode1=0 DecayMode2=0 ghz1=0,0 ghgsgs4=1,0 (...other options)
```

F. $J^P = 2_m^+$ resonance, $X \to ZZ$ or WW or $\gamma\gamma$

```
./JHUGen Process=2 a1=1,0 a2=0,0 b1=1,0 b2=0,0 b5=1,0 (...other options)
```

G. Cross-section calculation and fraction notation

For a vector boson coupling, we can represent the four independent parameters by two fractions (f_{g2} and f_{g4}) and two phases (ϕ_{g2} and ϕ_{g4}), defined for the HZZ and HWW couplings as follows (ignoring g_3)

$$f_{gi} = \frac{|g_i|^2 \sigma_i}{|g_1|^2 \sigma_1 + |g_2|^2 \sigma_2 + |g_4|^2 \sigma_4} \, ; \qquad \phi_{gi} = \arg\left(\frac{g_i}{g_1}\right) .$$

In order to obtain the cross-sections σ_i corresponding to the $g_i = 1$ coupling, generate large enough (e.g. VegasNc1=1000000, VegasNc2=50000000) number of weighted (Unweighted=0) with the corresponding couplings setup ($g_i = 1$, $g_{j\neq i} = 0$).

H. LHAPDF

It is possible to interface to an LHAPDF setup instead of compiling with local PDF's. To accomplish this:

- In the makefile:
    - Set UseLHAPDF=Yes
    - Set MyLHADir to a directory with your LHAPDF setup. This can be in terms of environment variables; for example MyLHADir=${LHAPDF_DATA_PATH}/../../lib/ should work as long as ${LHAPDF_DATA_PATH} only contains one directory.
- Ensure that $LHAPDF_DATA_PATH and $LD_LIBRARY_PATH are set (both when compiling and when running).
- Compile
- Run with the extra command line parameter LHAPDF specifying your PDF set's .info file. For example: LHAPDF=NNPDF30_lo_as_0130/NNPDF30_lo_as_0130.info

I. Including weight for decay of a stable wide resonance

In decay of a BSM Higgs boson with a large width, e.g. generated by POWHEG, it might be important to account for the final state phase space, in addition to generating the actual final state particle, such as $H \to ZZ \to 4f$ or $H \to WW \to 4f$. This could be configured with a weight discussed below. In addition, correction to the propagator could be applied in this step, such as for example the CPS scheme adopted in the POWHEG generator. To use these options, add the following command line options in the JHUGen card:

```
ReadLHE=(input file).lhe ReweightDecay WidthSchemeIn=3 ReadPmHstar PmHstarFile=(filename.out)
```

The weight distributions depend strongly on the HVV couplings. Weights generated for Standard Model couplings would not be applicable to BSM couplings. One can get the weight as a function of m_{VV} by running (WW example below):

```
./JHUGen DecayMode1=10 DecayMode2=10 MReso=(mass) PrintPMZZ=min,max PMZZIntervals=intervals
         ReweightDecay WidthSchemeIn=3 PMZZEvals=200000
```

Options:

- DecayMode1,2=10: specifies WW

- MReso: the weights are normalized to 1 at the pole mass

- PrintPMZZ=min,max: prints the weights from min to max

- PMZZIntervals=intervals: number of intervals

- ReweightDecay: this is what causes the probability to be corrected for the decay

- WidthSchemeIn=3: specifies that you want to reweight from POWHEG CPS, which means that the total decay width is already included and needs to be divided out.

- PMZZEvals=200000: number of evaluations (below $2m_V$ threshold may need more for better sampling)

VI. JHU GENERATOR MATRIX ELEMENTS (JHUGENMELA)

These instructions are meant as an introduction to MELA and do not illustrate the full flexibility provided by the package. For further instructions, please contact the authors.

A. Compiling

After extracting the code, you can go to the directory `JHUGenMELA/MELA` to find code for computing matrix elements directly. To compile the code, simply run `./setup.sh`. You can specify the option `-j N` for multi-core compilation, or option `clean` for cleanup. There is also the option `deps` to only set up dependencies without compiling MELA itself, which is useful if you only intend to use the dependencies for linking to JHUGenerator. The script will print a set of instructions to run at theend of its execution in order to set up to three environment variables that are needed to run MELA: `LD_LIBRARY_PATH` needs to include the library path for MELA, `PYTHONPATH` needs to contain the path for thte Python MELA interface, and `SCRAM_ARCH` needs to be set to refer to one of the folders inside MELA/data for the correct version of the MCFM library.

Please take note: The setup relies on MCFM libraries, which are downloaded from the main webpage during the `setup.sh` script. They are provided for `gfort + gcc version 4.6` or later, and only for SLC. They work out-of-the-box with other compiler versions that are not too far away, and we regularly monitor new versions. If you are using SLC but a very new gfort/gcc version that requires a new precompiled library, let us know and we will provide it promptly. If you are not using SLC, or the precompiled libraries do not work, you can always compile the MCFM library yourself by going into the MCFM-JHUGen directory and running `./compile.sh (-j N)` (`./compile.sh clean` for cleanup).

MELA requires the ROOT package [15], and the simplest way to use MELA is to compile in ROOT. Run the |loadMELA.C— script in the test directory before compiling or loading any other scripts that rely on MELA. It is also possible to link against the MELA code directly.

B.   Starting out

To begin, create a Mela object. For example, `std::unique_ptr<Mela> mela(new Mela(13, 125))` indicates a 13 TeV LHC energy and a 125 GeV Higgs boson. It is not recommended to create multiple Mela objects simultaneously, as some parameters in Fortran will be shared between the objects.

C.   Event input

The next step is to provide `mela` with the particles in the event. An event in MELA contains three sets of particles: the Higgs boson decay products ("daughters"), particles produced in association with the Higgs boson, such as VBF jets or particles from the V decay in VH ("associated"), and the incoming partons ("mothers"). Each particle has a PDG id and a 4-momentum. The mothers can and should be left out when dealing with reconstructed events, since they are not known. In that case either an empty vector or a null pointer can be used. Similarly, the associated collection can be empty when there are no associated particles.

If you need to calculate matrix elements for the Higgs boson decay, of course it is necessary to supply the decay products' PDG ids and momenta. If you only need to calculate production matrix elements (VBF or VH, for instance), the "daughters" can just contain the Higgs boson itself, with id 25 and the Higgs boson's 4-momentum.

To construct an event, use this syntax:

```
SimpleParticleCollection_t daughters, associated, mothers;
daughters.emplace_back(11, TLorentzVector(px, py, pz, E));    //for an electron
daughters.emplace_back(-12, TLorentzVector(px, py, pz, E));   //for an electron antineutrino
associated.emplace_back(1, TLorentzVector(px, py, pz, E));    //for a down quark
associated.emplace_back(0, TLorentzVector(px, py, pz, E));    //for an unknown particle,
                                                             //which may be a quark or a gluon
```

As the final step, pass the particle collections to `mela`:

```
mela->setInputEvent(&daughters, &associated, &mothers, false);
```

The last argument tells MELA whether the event is a generator-level event (true) or a reconstructed event (false). This affects how the output probabilities are normalized.

D.   Process selection

Once MELA has the event, you can calculate as many matrix elements as you want. Each matrix element is calculated for a particular process and Higgs boson hypothesis and uses the matrix elements from a particular generator (either `JHUGen` or `MCFM`). These are set using the `setProcess` function, with the help of the enums in `TVar.hh`. For example

```
//For a Standard Model Higgs boson in gluon fusion production
mela->setProcess(TVar::HSMHiggs, TVar::JHUGEN, TVar::ZZGG);
//For qq->ZZ background
mela->setProcess(TVar::bkgZZ, TVar::MCFM, TVar::ZZQQB);
//For VBF production of a pseudoscalar Higgs boson
mela->setProcess(TVar::HOminus, TVar::JHUGEN, TVar::JJVBF);
```

For full flexibility in the couplings, the textttSelfDefine processes can be used instead of the pre-cooked ones. In that case the couplings are set manually:

```
mela->setProcess(TVar::SelfDefine_spin0, TVar::JHUGEN, TVar::Had_ZH)
mela->selfDHzzcoupl[0][gHIGGS_VV_1][0] = 1;
mela->selfDHzzcoupl[0][gHIGGS_VV_4][0] = 1;
mela->selfDHzzcoupl[0][gHIGGS_VV_4][1] = 1;
```

In the notation of the generator, this is equivalent to `ghz1=1,0 ghz4=1,1`, which sets the Standard Model coupling to 1 and the pseudoscalar coupling to $1 + i$.

E. Calculate

Once the event, process, and (if necessary) couplings are set, the probability can be computed using one of these functions:

```
float result;
mela->computeP(result, useConstant);        //for decay
mela->computeProdP(result, useConstant);    //for production
mela->computeProdDecP(result, useConstant); //for the combined matrix element, using MCFM
```

The second argument affects the normalization, and should generally be set to true for reconstructed events, false for generator level events.

Any of these functions reset the process and couplings to their default values. If you want to compute another probability for the same event, there is no need to call `setInputEvent` again, but you do have to call `setProcess`, as well as set the couplings if the process is `SelfDefine`.

F. Clean up

The last step is to call

```
mela->resetInputEvent()
```

to avoid memory leaks. This is particularly important when processing many events in one job.

G. Python interface

We also provide a python interface for MELA, which works essentially the same way as the C++ interface. For more information, use the Python help function:

```
>>> import mela
>>> help(mela)
```

Since text processing is simple in Python, the Python MELA interface is especially useful for running quick jobs on LHE files.

VII. JHU GENERATOR COUPLING CONVERTER (JHUGENLEXICON)

In order to simplify translation between different coupling conventions, within the JHU generator framework, we provide the JHUGenLexicon program, which includes an interface to the generator and matrix element library and can also be used for standalone or other applications. In the present version, the program can take as input either the native coefficients used in JHUGen, or the Higgs basis coefficients for the HVV and Hgg couplings. The input can also be used either with or without the SU(2)×U(1) and custodial symmetry. There is also translation between the Warsaw basis coefficients and either of the above two sets of couplings, with the custodial symmetry introduced. The output, which can be used in the command line of the JHU generator or as input to MELA, provides all coefficients appearing without expectation of the above symmetry, as expected by the generator and MELA. The output can also include the triple and quartic gauge boson couplings used in off-shell EW processes. Various options can be checked within the JHUGenLexicon program. Below we list explicit examples how the program can be used for translating 15 HVV and Hgg couplings in the Higgs basis into the coefficients used in the generator command line. The other example includes translation of 10 HVV and Hgg couplings in the Higgs basis, which are the ten independent couplings in the Higgs basis under the SU(2)×U(1) and custodial symmetry $\delta c_z, c_{zz}, c_{z\square}, \tilde{c}_{zz}, c_{z\gamma}, \tilde{c}_{z\gamma}, c_{\gamma\gamma}, \tilde{c}_{\gamma\gamma}, c_{gg}$, and $\tilde{c}_{gg}$. For further details, see [16].

```
./JHUGenLexicon input_basis=hbasis output_basis=amp_jhu dCz=1,0 Czz=.5,0 Czbx=0,1E-7 tCzz=1.05,0
dCw=2,0 Cww=1,0 Cwbx=0,2E-7 tCww=1,1 Cza=1,.5 tCza=.5,1 Cabx=0,0 Caa=0,.5 tCaa=.5,1 Cgg=2.1,0
tCgg=.5,.5
```

```
./JHUGenLexicon input_basis=eft_hbasis output_basis=amp_jhu dCz=1,0 Czz=.5,0 Czbx=0,1E-7
tCzz=1.05,0 Cza=1,.5 tCza=.5,1 Cabx=0,0 Caa=0,.5 tCaa=.5,1 Cgg=2.1,0
```

It is possible to set the value of constants used in the translations in Eq. (8) of Ref. [5]. Parameters that manually set are given below. To modify these parameters, simply add param=num to the command line when calling JHUGenLexicon. Available parameters to set include the following:

```
Lambda_z1 (default = 10000 GeV)
Lambda_w1 (default = 10000 GeV)
Lambda_zgs1 (default = 10000 GeV)
MZ (default = 91.1876 GeV)
MW (default = 80.399 GeV)
sin2ThetaW (default = 0.23119)
alpha (default = 0.302862)
vev_lam (default =.0024622)
```

where Lambda_z1,Lambda_w1,Lambda_zgs1 correspond to Λ_1^{ZZ},Λ_1^{WW},$\Lambda_1^{Z\gamma}$ seen in Eq. (1) respectively, MZ is the mass of the Z Boson, MW is the of the W Boson, sin2ThetaW is $sin^2(\theta_W)$, alpha corresponds to the fine structure constant α and vev_lam is $\frac{v^2}{\Lambda^2}$ as in Eq.(A1) of [5]. Notice that there is no option to set the strong coupling, α_s. Since this is a running parameter, and JHUGen internally handles α_s, we make sure to remove all factors of g_s from translations of gluon couplings. Relations between gluon couplings are given by $c_{gg} = -\frac{2}{g_s^2} g_2^{gg}$ between the JHUGen amplitude basis and higgs basis and $c_{gg} = \frac{v^2}{\Lambda^2} \frac{4}{g_s^2} \omega_{\phi G}$ between the higgs and warsaw basis. Users should be careful to scale gluon coupling inputs and outputs accordingly.

Off-shell JHUGen processes, such as Process 68 can utilize triple and quartic gauge couplings. In order to return these couplings using JHUGenLexicon, use the command line argument include_triple_quartic_gauge=true. This option in its current implementation, is able to calculate tripe and quartic gauge couplings in the both the Higgs basis and JHUGen convention, using the relations in Eq. (14–20). When using JHUGenLexicon alongside JHUGen-MCFM processes, be sure to set either useMCFMAtInput=true or useMCFMAtOutput=true. useMCFMAtInput=True assumes that the $g^V V$ couplings are divided by 2. useMCFMAtOutput=True divides all of the g^{VV} couplings by 2 as per MCFM conventions.

The following example demonstrates how to modify constants used in the coupling translation, as well as the couplings that are output by JHUGenLexicon. In this case we include translation from 10 independent HVV and Hgg couplings in the Higgs basis to the native coefficients used in JHUGen. We return all couplings, including triple and quartic gauge couplings, with Lambda_w1, Lambda_z1 and Lambda_zgs1 set to 100GeV.

```
./JHUGenLexicon input_basis=eft_hbasis output_basis=amp_jhu dCz=1,0 Czz=.5,0 Czbx=0,1E-7
tCzz=1.05,0 Cza=1,.5 tCza=.5,1 Cabx=0,0 Caa=0,.5 tCaa=.5,1 Cgg=2.1,0 neutral_current=false
Lambda_z1=100 Lambda_w1=100 Lambda_zgs1=100
```

VIII.   RELEASE NOTES

In going from `v7.4.0` to `v7.5.0` (July 10, 2021), the updates are as follows:

- Add Processes 115,116,117: $pp \to tHW$

- Flexible parameterization of Zff couplings as corrections of the SM values

- Optimize VH processes

- Cross section calibration in $gg \to H$ Process 0

- Update DecayMode symmetry factors and relative cross sections (negative DecayMode in beta)

- Expand JHUGenLexicon with more options

In going from `v7.3.0` to `v7.4.0` (July 4, 2020), the updates are as follows:

- Added command line setting of triple and quartic gauge boson couplings in off-shell EW processes 66-68.

- Improvements on the phase space optimization for off-shell processes.

- Extension of DecayMode options, final state filters, and implementation of the final state filters in the off-shell processes. Filter options are also renamed as a result of the extension of functionality.

- Enlarge allowed invariant mass ranges in the $q\bar{q} \to \gamma H$ and $q\bar{q} \to Z/\gamma^* H$ processes.

- Completion of the implementation of couplings with V' in all signal-only JHUGen processes at leading-order in QCD, where relevant, and decay lifetime for $V'V'$ final states in ReadLHE mode.

- Introduced JHUGenLexicon program for converting coupling convention, with the translation available between the JHU generator / MELA convention, Higgs basis, and Warsaw basis.

In going from `v7.2.7` to `v7.3.0` (April 28, 2019), the update is as follows:

- Faster option for generating interference in 4ℓ final states in off-shell EW production, 164 jobs per step.

In going from `v7.2.4` to `v7.2.7` (January 31, 2019), the updates are as follows:

- $gg \to ZH$ with anomalous couplings using Process 51, using the same syntax as for other processes.

- Faster option for generating interference in 4ℓ final states in off-shell EW production.

- Additional options for the renormalization and factorization scales in JHUGen and MELA.

In going from `v7.1.4` to `v7.2.4` (November 15, 2018), the updates are as follows:

- Processes 66 (signal), 67 (background), and 68 (signal+background+interference) are added to generate off-shell EW events, including hadronic VH, the continuum background, and up to two scalar resonances. The matrix elements are provided through an interface to MCFM, modified to include HVV and XVV anomalous couplings.

- Process 51 is added with a new procedure to generate $gg \to ZH$ events. ($qq \to VH$ and $ee \to ZH$ are also supported but currently in beta. We recommend using Process 50 instead for them.)

In going from `v7.1.2` to `v7.1.4` (March 23, 2018), the updates are as follows:

- Reusing the previous grid via the `ReadCSmax` command line option is improved.

In going from `v7.0.2` to `v7.1.2` (February 22, 2018), the updates are as follows:

- MELA package interface is completely rewritten and streamlined, including new processes implemented

- Add the option for an intermediate Z' or W' in spin 0 and spin 2 resonance decay and in VBF and VH production. V' can have finite or infinite mass; in the second case, the `ghzzp1` coupling is equivalent to the contact terms formulation [14]

- Fix in the `<init>` block when reading an LHE file and reweighting the mass shape

In going from `v7.0.0` to `v7.0.2` (October 4, 2016), the updates are as follows:

- Improvements in WH, VBF, $H + JJ$, ttH, bbH, and $t + H$ production.

- New process 114, which combines all $t + H$ channels together.

In going from `v6.9.8` to `v7.0.0` (July 29, 2016), the updates are as follows:

- Further improvements in phasespace sampling in the `ReadLHE` mode.

- Improvements in VBF and VH production

In going from `v6.9.5` to `v6.9.8` (April 11, 2016), the updates are as follows:

- Phasespace sampling in the `ReadLHE` mode is improved.

- Standalone XVV production with V decay to any final state is improved.

In going from `v6.8.4` to `v6.9.5` (March 23, 2016), the updates are as follows:

- $X \to Z\gamma$ for spin-2 process `2` is implemented.

- Second resonance, including interference with $H(125)$ and $gg \to ZZ$ background, added in the $gg \to ZZ$ MCFM process under JHUGenMELA.

- Improvements are added to the cross section and phasespace in standalone productions for processes `0-2`.

- In `ReadLHE` mode, the mass shape can be reweighted:

  - to change from one propagator scheme to another, and/or
  - to account for $P_{dec}(m_{4f})$

- Additional command line parameters have been added:

  - Resonance width
  - Anomalous XVV and Xff couplings
  - Jet cuts
  - Random seed

- The command line option `OffXVV` (000-111) is replaced with the option `OffshellX` (1/0, 1 as default).

In going from `v6.7.8` to `v6.8.4` (January 7, 2016), the updates are as follows:

- Added running renormalization scale and generalized the running scales

- Added more options for the Higgs mass shape

- Improved LHE reading and writing

In going from `v6.2.8` to `v6.7.8` (December 21, 2015), the updates are as follows:

- Added running factorization scale ($\mu_F = m_{H*}/2$) in spin-0,1,2 decays

- Enabled identical 4ν and $4q$ final state interference in ZZ decay

- Faster VBF and Hjj event generation with deterministic jet flavor assignment

- VH code clean-up

- Update of the CKM matrix and of hadronic W decay

- New processes `110-113` for $t/\bar{t} + H$ production

- More flexible lepton filter

In going from `v5.6.3` to `v6.2.8` (September 17, 2015), the updates are as follows:

- Flexible LHE reading process for compatibility with more generators

- Small bugfixes

In going from `v5.2.5` to `v5.6.3` (June 8, 2015), the updates are as follows:

- Add `Process=90` for bbH production

- Add lepton filtering option

- Allow W from ttH to decay to any decay mode

- Allow W to decay to off-diagonal elements of the CKM matrix

- Add support for LHAPDF linking

- Fixes for LHE printout in VBF, Hjj, and VH

In going from `v4.8.1` to `v5.2.5` (March 17, 2015), the updates are as follows:

- Add `Process=80` for ttH production, with optional top decays

- Add support for NNPDF

- Make `DecayMode1` $\neq$ `DecayMode2` equivalent to generating everything and then filtering

- Add option for randomizing the V's in HVV decays

- Fixes for smoother reading of LHE files: mother assignment and invariant mass for all intermediate particles

- Add `ConvertLHE` option for converting VH decay to any DecayMode

- In `ReadLHE` and `ConvertLHE`, preserve comments and optional tags from the input LHE

In going from `v4.5.2` to `v4.8.1` (October 31, 2014), the updates are as follows:

- More flexibility for q^2-dependent form factors

- Separate couplings for ZZH and WWH in weak boson fusion

- Add new process: $pp \rightarrow H$+jet (Process=62)

- Extended LHE output format to allow for more digits

- MCFM plug-in for anomalous couplings in offshell Higgs boson production in gg-¿ZZ

- Synchronize JHUGenMELA with MCFM library v6.8

- JHUGenMELA: extended MCFM ggHZZ matrix elements by anomalous couplings

- JHUGenMELA: add matrix elements for H+jet and V+H

In going from `v4.3.2` to `v4.5.2` (February 16, 2014), the updates are as follows:

- Add an option of intermediate photons for the modes with Z-bosons

- More flexibility for q^2-dependent form factors

- Option of hadronic branching rescaling (NLO QCD corrections) for inclusive decays

- Synchronize JHUGenMELA with the generator and with MCFM library v6.7

In going from `v4.2.1` to `v4.3.2` (December 12, 2013), the updates are as follows:

- Update LHE file format and index of partons

- Improve log printout

- Update ReadLHE mode: $H \rightarrow Z\gamma$ output and more flexible input

- VH production (replaces beta version)

- More flexibility for q^2-dependent form factors

- Tune q^2-dependence of couplings for some of the spin-2_h models

- Synchronize JHUGenMELA with the generator

In going from `v4.0.x` to `v4.2.x` (November 2, 2013), the updates are as follows: To JHUGenerator:

- Fix BR in "all" decay mode

- Updates to LHE output

- Option to print out CS_max, output for g' and Lambdas

- Introduction of AnalyticMELA for $ee \rightarrow ZH$ and $pp \rightarrow ZH$ and analytic parton distribution functions

In going from `v3.1.x` to `v4.0.x` (October 7, 2013), the updates are as follows: To JHUGenerator:

- Addition of VBF and Hjj process channels

- Possibility to read in VBF LHE event files

To JHUGenMELA:

- Interface with the MCFM program for ggZZ process

- Matrix elements for VBF and Hjj processes

In going from `v2.2.6` to `v3.1.8` (June 18, 2013), the updates are as follows: To `JHUGenerator`:

- Capability reading LHE files with Higgs boson production, allows NLO production of spin-0;

- Extended the list of final state combinations;

- Log messages, lhe file headers, and minor cleanup.

- Updates to deal with non-zero lepton masses, lhe file format, and adjust default settings (e.g. lepton interference applied by default and can be configured in command line)

To `JHUGenMELA`:

- Production-independent JHUGenMELA for spin-0, 1, 2;

- Complex couplings in JHUGenMELA input.

In going from `v2.2.3` to `v2.2.6` (December 22, 2012), the updates are as follows:

- A small fix which corrects the *relative fraction* between the $2e2\mu$ and $4e/4\mu$ channels when using interference

- beta version is still under development

- $q\bar{q} \to$ spin-2 production is more safely performed with settings `PChannel = 2` and $q\bar{q}$ fraction $= 1$.

In going from `v2.1.3` to `v2.2.3` (October 30, 2012), the updates are as follows:

- Fix interference and randomization in the *beta* version

- Add the `JHUGenMELA` modules

- Small change for compilation on Mac OSX platforms

- Fix for tau masses in W decays

In going from `v2.0.2` to `v2.1.x`, (August 20, 2012) the updates are as follows:

- Histograms are written in file (default: ./data/output.dat) and no longer on the screen. How to understand the histogram data and how to plot is briefly described in the output.dat file.

- Added tau masses

- Added lepton interference in the ZZ4l final state

- Added switch `generate_as` to choose couplings in spin-0 case (works for on- and offshell resonance). The default is ".false.".

- Added the possibility to change graviton-quark couplings. The new parameters are `graviton_qq_left`, `graviton_qq_right` and correspond to $0.5*(1-\gamma^5)$ and $0.5*(1+\gamma^5)$ helicity projectors, respectively. Up to now the coupling was always vector-like. This is also the new default, `graviton_qq_left = graviton_qq_right =1`.

- The random seed is now fixed with gfortran.

- The call "./JHUGen help" prints out all available command line options

- Added new command line option "Unweighted=0 or 1" (default is 1)

Appendix A: Specific configurations

We define configurations for certain models which are defined in Table 1 of [2].

1. Spin 0

All couplings not specified are 0 by default except for the SM couplings `ghz1` and `ghg2`, which must be set to 0 explicitly if that is desired. For protection, if any anomalous couplings in the same category are set, those couplings must be explicitly set either to 0,0 or to some other value.

a. "SM-like spin-zero", 0^+

`ghz1=1,0`

b. "Higher order spin-zero", 0_h^+

`ghz1=0,0 ghz2=1,0`

c. "Pseudoscalar spin-zero", 0^-

`ghz1=0,0 ghz4=1,0 ghg2=0,0 ghg4=1,0`

2. Spin 1

The $Z'VV$ couplings `zprime_zz_1` and `zprime_zz_2` are both 0 by default; at least one of them must be explicitly turned on. The $Z'q\bar{q}$ couplings `zprime_qq_left` and `zprime_qq_right` are both set to 1 by default; if one is changed, then both must be explicitly set.

a. "Vector spin-one", 1^-

`zprime_qq_left=1,0 zprime_qq_right=0,0 zprime_zz_1=1,0`

b. "Pseudovector spin-one", 1^+

`zprime_qq_left=1,0 zprime_qq_right=0,0 zprime_zz_2=1,0`

3. Spin 2

The Ggg and GVV couplings `a1-5` and `b1-10` are set to 0 by default; at least one in each category must be explicitly turned on. The $Gq\bar{q}$ couplings `graviton_qq_left` and `graviton_qq_right` are both set to 1 by default; if one is changed, then both must be explicitly set. If an exclusive production mode is desired (e.g. $q\bar{q}$ or gg), this is handled at command-line configuration level via the `PChannel` variable.

a. "Minimal Graviton, spin-two", 2^+

`a1=1,0 b1=1,0 b5=1,0`

b.   *"Higher order Graviton, spin-two", 2_h^+*

a4=1,0 b4=1,0

c.   *"Higher order Graviton, spin-two", 2_h^-*

a5=1,0 b8=1,0

---

[1] Y.Y. Gao, A. V. Gritsan, Z.J. Guo, K. Melnikov, M. Schulze and N. V. Tran, "Spin-Determination of Single-Produced Resonances at Hadron Colliders." Phys. Rev. D **81**, 075022 (2010). arXiv:1001.3396 [hep-ph].

[2] S. Bolognesi, Y.Y. Gao, A. V. Gritsan, K. Melnikov, M. Schulze, N. V. Tran and A. Whitbeck, "On the Spin and Parity of Single-Produced Resonance at the LHC." Phys. Rev. D **86**, 095031 (2012). arXiv:1208.4018 [hep-ph].

[3] I. Anderson, S. Bolognesi, F. Caola, Y.Y. Gao, A. V. Gritsan, C. B. Martin, K. Melnikov, M. Schulze, N. V. Tran, A. Whitbeck, Y. Zhou, "Constraining anomalous HVV interactions at proton and lepton colliders." Phys. Rev. D **89**, 035007 (2014). arXiv:1309.4819 [hep-ph].

[4] A. V. Gritsan, R. Röntsch, M. Schulze, M. Xiao, "Constraining anomalous Higgs boson couplings to the heavy flavor fermions using matrix element techniques." Phys. Rev. D **94** (2016) 055023. arXiv:1606.03107 [hep-ph].

[5] A. V. Gritsan, J. Roskes, U. Sarica, M. Schulze, M. Xiao and Y. Zhou, "New features in the JHU generator framework: constraining Higgs boson properties from on-shell and off-shell production." Phys. Rev. D **102** (2020) 056022. arXiv:2002.09888 [hep-ph].

[6] T. Martini, R. Q. Pan, M. Schulze and M. Xiao, "Probing the CP structure of the top quark Yukawa coupling: Loop sensitivity vs. on-shell sensitivity." [arXiv:2104.04277 [hep-ph]].

[7] See webpage: `http://spin.pha.jhu.edu`

[8] S. Chatrchyan *et al.* [CMS Collaboration], "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," Phys. Lett. B **716**, 30 (2012) [arXiv:1207.7235 [hep-ex]].

[9] J. M. Campbell and R. K. Ellis, "MCFM for the Tevatron and the LHC," Nucl. Phys. Proc. Suppl. **205-206**, 10 (2010) [arXiv:1007.3492 [hep-ph]]. See also http://mcfm.fnal.gov

[10] "Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector," CERN Yellow Report 10.23731/CYRM-2017-002. [arXiv:1610.07922 [hep-ph]].

[11] A. Falkowski, "Higgs Basis: Proposal for an EFT basis choice for LHC HXSWG", Technical Report LHCHXSWG-INT-2015-001, 2015. `https://cds.cern.ch/record/2001958`.

[12] S. Goria, G. Passarino and D. Rosco, "The Higgs-boson lineshape," Nucl. Phys. B **864** (2012) [arXiv:1112.5517 [hep-ph]].

[13] G. Passarino, C. Sturm and S. Uccirati, "Higgs pseudo-observables, second riemann sheet and all that," Nucl. Phys. B **834** (2010) [arXiv:1001.3360 [hep-ph]].

[14] M. Gonzalez-Alonso, A. Greljo, G. Isidori, and D. Marzocca, "Pseudo-observables in Higgs decays," Eur. Phys. J. C **75** (2015) 128, doi:10.1140/epjc/s10052-015-3345-5, arXiv:1412.6038 [hep-ph].

[15] See webpage: `https://root.cern.ch`

[16] A. Falkowski et al., "Rosetta: an operator basis translator for Standard Model effective field theory", Eur. Phys. J. **C75** (2015) 583, doi10.1140/epjc/s10052-015-3806-x, arXiv:1508.05895 [hep-ph].