Manual for the JHU generator and MELA package

For simulation, reweighting, and analysis of a single-produced resonance at hadron colliders
(version v7.2.7, release date January 31, 2018)

The generator from [1–4] is a model-independent generator for studying spin and parity properties of new resonances. Please cite [1–4] if using the "JHU generator". The code can be downloaded from [5]. 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. [6]. The matrix element package (MELA) and off-shell VBF generation also depend on MCFM libraries for background parameterization which should be referenced [7] when used.

Contents

I. Installation 2

II. Quick start 2

III. Configuration 4
- A. Command line configuration 4
  - 1. Quick reference 4
  - 2. Process configuration 7
  - 3. Process-specific configuration 7
  - 4. Resonance parameters 8
  - 5. Cuts 8
  - 6. Renormalization and factorization scales 9
  - 7. Lepton and jet filter 9
  - 8. Higgs propagator and decay width 10
  - 9. Statistics options 10
  - 10. Couplings 11
- B. Configuration in parameter file 15

IV. Examples 15
- A. $J^P = 0^+_m$ resonance, $X \to ZZ$ or WW 15
- B. $J^P = 0^-_m$ resonance, $X \to ZZ$ or WW 15
- C. $J^P = 0^+_m$ resonance, $X \to \gamma\gamma$ 15
- D. $J^P = 0^-_m$ resonance, $X \to \gamma\gamma$ 16
- E. $J^P = 0^-_m$ resonance, $X \to \gamma^*\gamma^* \to 4\ell$ 16
- F. $J^P = 2^+_m$ resonance, $X \to ZZ$ or WW or $\gamma\gamma$ 16
- G. Cross-section calculation and fraction notation 16
- H. LHAPDF 16

V. JHU Generator Matrix Elements (JHUGenMELA) 16
- A. Compiling 16
- B. Starting out 17
- C. Event input 17
- D. Process selection 17
- E. Calculate 18

F. Clean up 18
G. Python interface 18

VI. Release notes 18

A. Specific configurations 23
1. Spin 0 23
a. "SM-like spin-zero", 0^+ 23
b. "Higher order spin-zero", 0_h^+ 23
c. "Pseudoscalar spin-zero", 0^- 23
2. Spin 1 23
a. "Vector spin-one", 1^- 23
b. "Pseudovector spin-one", 1^+ 23
3. Spin 2 23
a. "Minimal Graviton, spin-two", 2^+ 23
b. "Higher order Graviton, spin-two", 2_h^+ 24
c. "Higher order Graviton, spin-two", 2_h^- 24

References 24

I. 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
```

II. 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
  ```

- $gg \to ZH$, `Process=51`:
  - ZH production with a_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
    ```

  - `Process=51` also includes all the features of `Process=50`, but is currently in beta; we recommend using it 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
    ```

- Offshell VBF, `Process=66`, `67`, `68`
  - including VBF and hadronic VH with up to two scalar resonances and interference with the SM background
  - see below, page 8, 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
```

- Decay-only mode, for generating the decay of a stable Higgs boson produced by the JHU generator or another generator (e.g. POWHEG).
  - $H \to VV$ 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 $H \to \tau\tau$ 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
    ```

III. 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
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.,
                     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
  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
  Interf:            0=neglect interference for 4f final states,
                     1=include interference
  ReweightInterf:    if true, include interference as LHE event weights for
                     offshell VBF 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 VBF offshell production, set this to an index
                    for each of the jobs.  See below 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 offshell
                    in processes 0, 1 or 2
  MReso2:           2nd resonance mass in GeV in offshell VBF
  GaReso2:          2nd resonance width in GeV in offshell VBF
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 offshell photons in GeV, when included (default: 4)
  etajetcut:         Maximum |eta| for jets in offshell VBF (default: 4)
  detajetcut:        Minimum deltaeta between jets in offshell VBF (default: 2)
  JetsOppositeEta:   Require sgn(eta) to be opposite for the two jets in offshell VBF
                     (default: true)
  pTlepcut:          Minimum pT for leptons in offshell VBF, in GeV (default: 3)
  etalepcut:         Maximum |eta| for leptons in offshell VBF (default: 2.7)
  m4l_min, m4l_max:  Minimum and maximum four-lepton mass in offshell VBF
  m2l_min:   Minimum invariant mass of V (onshell) in new VH (\texttt{Process=51}) (default: 0)
  m2l_max:   Maximum invariant mass of V (onshell) 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:
  FilterNLept:       For decay mode, reject events that have less than FilterNLept leptons
  FilterOSPairs:     For decay mode, reject events that have less than FilterOSPairs pairs of
                     sign leptons of any flavor.
  FilterOSSFPairs:   For decay mode, reject events that have less than FilterOSSFPairs pairs of
                     opposite-sign-same-flavor leptons.
  CountTauAsAny:     For FilterOSSFPairs, taus can stand in place of electrons or muons
                     of the same charge.
  FilterNJets:       For decay mode, reject events that have less than FilterNJets quarks
                     and/or gluons
  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
Couplings:
  See below 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 $H \to \tau\tau$ 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 offshell VBF 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. IV 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), one-loop induced LO $gg \to ZH$, and $q\bar{q} \to V \to VH$ at NLO (currently in development). 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. γH, i.e. `DecayMode1=7`, is not yet implemented for `Process=51`.

b. Offshell VBF Processes 66 (signal), 67 (background), and 68 (signal+background+interference) provide an interface to a modified version of MCFM [7] to produce offshell VBF production $VV \to VV$, including all anomalous couplings. Hadronic VH and the equivalent VVV background topologies are also included. 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.

When producing unweighted VBF offshell events, jobs must be submitted in two steps, with five jobs per step. Each set of five jobs can be run in parallel, but none of the jobs in the the second set can start until all five jobs of the first set finish.

1. Grid generation:
   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 Interf=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 Interf=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 Interf=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 Interf=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 Interf=0 ghz1=1d0,0d0 VBFoffsh_run=5`
2. Event generation, which can only begin after all five previous jobs have ended:
   - `./JHUGen Process=68 deltaRcut=0.3 pTjetcut=10 mJJcut=70 m4l_min=70 m4l_max=13000 VegasNc0=10000000 Interf=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 Interf=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 Interf=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 Interf=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 Interf=0 ghz1=1d0,0d0 VBFoffsh_run=5 VegasNc2=10000 ReadCSmax`

The final result will be five `.lhe` files, each containing events from some of the partonic channels. 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.

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`, `deltaRcut`, `mJJcut` All three cuts are used in processes 60 (VBF), 61 ($H + jj$), 90 (bbH), 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.

- `MPhotonCutoff` If the decay is to ZZ and any of the couplings to photons (`ghzgsX` or `ghgsgsX`) 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 `FilterNLept`, `FilterOSPairs`, `FilterOSSFPairs` 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.

- `FilterNJets` 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 [8, 9]. 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`

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`.

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 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]:

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

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)
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 $\gamma^* Z$ 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 offshell VBF production.

Contact terms 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 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. 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 W' equivalent of the CKM matrix is assumed to be diagonal.

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 [10]. Z' (W') can also be given finite mass by setting its mass `MZprime` (`MWprime`) and width `GaZprime` (`GaWprime`) on the command line.

b. 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+
```

c. 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 III A 10 a.

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.

IV. 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. IV C and the next Sec. IV 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. IV 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`

V. 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`. At the end, it will print two environment variables that have to be exported in order to use `MELA`. `LD_LIBRARY_PATH` is needed for any use of MELA, while `PYTHONPATH` is needed to use the Python MELA interface.

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 5.3.0` or later, on SLC, and probably work with other compiler versions that are not too far away. Better portability coming soon.

MELA requires the ROOT package [11], 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::H0minus, TVar::JHUGen, TVar::JJVBF);
```

For full flexibility in the couplings, the texttttSelfDefine 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.

VI. RELEASE NOTES

In going from `v7.2.4` to `v7.2.7` (January 31, 2018), 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 offshell VBF 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 offshell VBF 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 [10]
- 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: $pp \to H$+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: $H \to Z\gamma$ 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 $ee \to ZH$ and $pp \to ZH$ 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] See webpage: `http://spin.pha.jhu.edu`

[6] 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]].

[7] 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

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

[9] 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]].

[10] 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].

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