Manual for the JHU generator

For simulation of a single-produced resonance at hadron colliders
(version v6.2.8, release date September 17, 2015)

The generator from [1–3] is a model-independent generator for studying spin and parity properties of new resonances. Please cite [1–3] if using the "JHU generator". The code can be downloaded from [4]. 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 analysis technique. The latter was also introduced in Ref. [5]. The matrix element package (MELA) also depends on MCFM libraries for background parameterization which should be referenced [6] when used.

Contents

I. INSTALLATION

Register and download the package from `www.pha.jhu.edu/spin` 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. CONFIGURATION

There are two ways to configure the program, from the command line and in the file `mod_Parameters.F90`. For documentation from the command line, one can use `JHUGen help`. In addition, the command line configurables are defined in the file `main.F90`. When one change the fortran code directly, one should also recompile the code for changes to take effect. In general, command-line configuration handles general event properties while the configuration file handles all of the couplings and physics handles.

A. Command line configuration

The list of command line configurables and the default values are (also defined in the `README`):

```
Collider:     1=LHC, 2=Tevatron, 0=e+e-
Process:      0=spin-0, 1=spin-1, 2=spin-2 resonance, 50=pp/ee->VH,
              60=weakVBF, 61=pp->Hjj, 62=pp->Hj, 80=ttH, 90=bbH
MReso:        resonance mass (default=126.00), format: yyy.xx
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
PChannel:     0=g+g, 1=q+qb, 2=both
OffXVV:       off-shell option for resonance(X),or vector bosons(VV)
PDFSet:       1=CTEQ6L1(2001),  2=MSTW(2008),
                                2xx=MSTW with eigenvector set xx=01..40,
                                3=NNPDF3.0LO
VegasNc0:     number of evaluations for integrand scan
VegasNc1:     number of evaluations for accept-reject sampling
VegasNc2:     number of events for accept-reject sampling
Unweighted:   0=weighted events, 1=unweighted events
Interf:       0=neglect interference for 4f final states,
              1=include interference
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)
TopDK:        For Process=80, 0=leave top quarks as stable, 1=decay top quarks
FilterNLept:  For decay mode, reject events that have less than FilterNLept leptons
FilterOSSF:   For decay mode, reject events that don't have 2 pairs of
              opposite-sign-same-flavor leptons.  Taus can stand in place of electrons
              or muons of the same charge.
LHAPDF:       PDF set to use if interfaced with LHAPDF.  See below.
```

A few more details on some particular parameters:

- `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.
- `OffXVV`: The program does not work for ZZ or WW if you set them to be on-shell (`OffXVV="000"`) and the mass of the X resonance to be below the m_{VV} threshold. In general, the more off-shell the process, or the more "1" you have, the less efficient the `VegasNc1` evaluations are. Specifically, if you are interested then, in producing a resonance with mass below threshold m_{VV} with a very narrow resonance, it is most efficient to generate with `OffXVV="011"`
- `PChannel`: This parameter is only meaningful in the spin-2 case and for Hjj, Hj, ttH, and bbH production mechanisms. For spin-0, production is possible only via the gg process; for spin-1 and for the VBF and VH processes, production is only possible via the $q\bar{q}$ process.
- `DecayMode1,2=7`: Valid for spin-0 and spin-2. Only OffXVV=000 or 100 are possible.
- 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.
- In ttH production, `DecayMode1` and `DecayMode2` are used to decay the W bosons from the top decay. Only W decay modes are allowed.

Then, as an example of running the generator, you could do:
- gg production:

```
./JHUGen Collider=1 Process=0 VegasNc2=100000 PChannel=0 OffXVV=011 DecayMode1=0 DecayMode2=0 \\
Unweighted=1 DataFile=test1
```

- ggH → Zgamma

```
./JHUGen DecayMode1=0 DecayMode2=7 OffXVV=010
```

- VH (both pp and e^+e^- Collider options possible):

```
./JHUGen Collider=1 Process=50 Unweighted=1 VegasNc2=100000 OffXVV=011 DataFile=test2
```

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

```
./JHUGen Collider=1 Process=50 DecayMode1=1 Unweighted=1 VegasNc2=100000 \\
OffXVV=011 DataFile=test2
```

- WH with leptonic W decay (change `DecayMode1` for other W decays; both pp and e^+e^- Collider options possible):

```
./JHUGen Collider=1 Process=50 DecayMode1=4 Unweighted=1 VegasNc2=100000 \\
OffXVV=011 DataFile=test2
```

- VBF:

```
./JHUGen Collider=1 Process=60 Unweighted=1 VegasNc2=100000 OffXVV=011 DataFile=test2
```

- H+jj:

```
./JHUGen Collider=1 Process=61 Unweighted=1 VegasNc2=100000 OffXVV=011 DataFile=test3
```

- H+j:

```
./JHUGen Collider=1 Process=62 Unweighted=1 VegasNc2=100000 OffXVV=011 DataFile=test4
```

- $pp \to t\bar{t} + H$ with inclusive top decay (change `DecayMode1,2` for specific tt decays):

```
./JHUGen Collider=1 Process=80 DecayMode1=11 DecayMode2=11 Unweighted=1 \\
VegasNc2=100000 OffXVV=011 DataFile=test5
```

- $pp \to b\bar{b} + H$:

```
./JHUGen Collider=1 Process=90 Unweighted=1 VegasNc2=100000 OffXVV=011 DataFile=test6
```

For generating Higgs decay in VBF, H+J(J), VH, or ttbH production modes by the JHU generator or NLO gluon fusion with another generator (e.g. POWHEG), use JHU generator in `ReadLHE` mode and specify the decay mode of interest (ZZ, WW, gam gam, Z gam), while the SM fermionic decays may be generated by Pythia without loss of generality.

B. Configuration in parameter file

In the file `mod_Parameters.F90`, one does all the configuration of the couplings of the resonance. After modifying this file, one needs to recompile.

1. General parameters

Each generation run is different when this is `.true.`

```
seed_random = .true.
```

In the case when `PChannel=2` for a spin-2 resonance, the user can define an approximate ratio of the production of gg and $q\bar{q}$ production.

```
fix_channels_ratio = .true.
channels_ratio_fix = 0.25d0    ! desired ratio of
                               ! N_qq/(N_qq+N_gg)
```

For final states with a Z-boson decaying into $f\bar{f}$, intermediate off-shell photons can be included by switching

```
logical, public, parameter :: includeGammaStar = .false.
```

to the value `.true.`. In such case, a lower cut on the photon invariant mass has to be placed in order to avoid the collinear singularity. This cutoff parameter is controlled by

```
real(8),parameter :: MPhotonCutoff = 4d0*GeV.
```

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 changing

```
logical, public, parameter :: RandomizeVV = .true.
```

to `.false.`. 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.
When reading in LHE files, JHUGen determines the format by looking at the first lines of the first event. If this determination fails for any reason (for example, if different events are written in inconsistent formats), try changing

```
+logical, public, parameter :: UseUnformattedRead = .false.
```

to `.true.`. However, this will noticeably slow down event generation in `ReadLHE` mode.
Only for final states with 4 same flavor fermions , one can include interference effects between the leptons. The interference is controlled by the command line parameter:

```
Interf=0 or 1
```

For the generation of weighted events (command line `Unweighted=0`) an LHE output file is created if

```
logical, public, parameter :: writeWeightedLHE = .false.
```

is set to `.true.`.

The remaining parameters are more-or-less self-explanatory:

```
! we are using units of 100GeV, i.e. Lambda=10 is 1TeV
real(8), public, parameter :: GeV=1d0/100d0
real(8), public, parameter :: percent=1d0/100d0
real(8), public :: M_V,Ga_V
real(8), public, parameter :: M_Top   = 173d0     *GeV      !
real(8), public, parameter :: Ga_Top  = 1.33d0    *GeV     !
real(8), public, parameter :: M_Z     = 91.1876d0 *GeV      ! Z boson mass (PDG-2011)
real(8), public, parameter :: Ga_Z    = 2.4952d0  *GeV      ! Z boson width(PDG-2011)
real(8), public, parameter :: M_W     = 80.399d0  *GeV      ! W boson mass (PDG-2011)
real(8), public, parameter :: Ga_W    = 2.085d0   *GeV      ! W boson width(PDG-2011)
real(8), public            :: M_Reso  = 125.6d0   *GeV      ! X resonance mass (spin 0, spin 1,
                                                            !  spin 2) (careful: no longer a
                                                            !  parameter, can be overwritten by
                                                            !  command line argument)
real(8), public, parameter :: Ga_Reso = 0.00415d0 *GeV      ! X resonance width
real(8), public, parameter :: Lambda  = 1000d0    *GeV      ! Lambda coupling enters in two
                                                            !  places: overall scale for
                                                            !  x-section and in power suppressed
                                                            !  operators/formfactors (former r).

real(8), public, parameter :: m_el = 0.00051100d0  *GeV     ! electron mass
real(8), public, parameter :: m_mu = 0.10566d0  *GeV        ! muon mass
real(8), public, parameter :: m_tau = 1.7768d0  *GeV        ! tau mass
real(8), public, parameter :: m_bot = 4.2000d0     *GeV     ! bottom quark mass


real(8), public, parameter :: Gf = 1.16639d-5/GeV**2        ! fermi constant
real(8), public, parameter :: vev = 1.0d0/sqrt(Gf*sqrt(2.0d0))
real(8), public, parameter :: gwsq = 4.0d0 * M_W**2/vev**2  ! weak constant squared
real(8), public, parameter :: alpha_QED = 1d0/132.2319d0    ! el.magn. coupling
real(8), public, parameter :: alphas = 0.13229060d0         ! strong coupling
real(8), public, parameter :: sitW = dsqrt(0.23119d0)       ! sin(Theta_Weinberg) (PDG-2008)
real(8), public            :: Mu_Fact                       ! pdf factorization scale
                                                            ! (set to M_Reso in main.F90)
real(8), public, parameter :: LHC_Energy=14000d0  *GeV      ! LHC hadronic center of mass energy
real(8), public, parameter :: TEV_Energy=1960d0  *GeV       ! Tevatron hadronic center of mass
                                                            !  energy
real(8), public, parameter :: ILC_Energy=250d0  *GeV        ! Linear collider center of mass
                                                            !  energy
real(8), public, parameter :: POL_A = 0d0                   !e+ polarization. 0: no polarization,
                                                            !                 100: helicity = 1,
                                                            !                -100: helicity = -1
real(8), public, parameter :: POL_B = 0d0                   !e- polarization. 0: no polarization,
                                                                              100: helicity = 1,
                                                                             -100: helicity = -1
logical, public, parameter :: H_DK =.true.                  !default to false so H in
                                                             V > VH (Process = 50) does not decay
real(8), public, parameter :: ptjetcut = 15d0*GeV           ! jet min pt
real(8), public, parameter :: Rjet = 0.5d0                  ! jet deltaR, antikt algorithm
```

The branching fractions of Z and W bosons depend on the above input parameter and can slightly vary from the given PDG measurements. Those branchings can be rescaled with the parameters below.

```
real(8), public, parameter :: scale_alpha_Z_uu = 1.037560d0 ! scaling factor of alpha
                                                           ! (~partial width) for Z > u u~, c c~
real(8), public, parameter :: scale_alpha_Z_dd = 1.037560d0 ! scaling factor of alpha
```

```
                                                           ! (~partial width) for
                                                           ! Z > d d~, s s~, b b~
real(8), public, parameter :: scale_alpha_Z_ll = 1d0       ! scaling factor of alpha
                                                           ! (~partial width) for Z > l+ l-
                                                           ! (l=e, mu)
real(8), public, parameter :: scale_alpha_Z_tt = 1d0       ! scaling factor of alpha
                                                           ! (~partial width) for Z > tau+ tau-
real(8), public, parameter :: scale_alpha_Z_nn = 1d0       ! scaling factor of alpha
                                                           ! (~partial width) for Z > nu nu~
real(8), public, parameter :: scale_alpha_W_ud = 1.038200d0  ! scaling factor of alpha
                                                             ! (~partial width) for W > u d
real(8), public, parameter :: scale_alpha_W_cs = 1.038200d0  ! scaling factor of alpha
                                                             ! (~partial width) for W > c s
real(8), public, parameter :: scale_alpha_W_ln = 1d0       ! scaling factor of alpha
                                                           ! (~partial width) for W > l nu
real(8), public, parameter :: scale_alpha_W_tn = 1d0       ! scaling factor of alpha
                                                           ! (~partial width) for W > tau nu
```

These default values rescale the branchings to include the NLO QCD corrections $(1 + \alpha_s/\pi)$.

2. Spin-0 parameters

N.B. The parameters "ptjetcut" and "Rjet" only apply to Process=60,61.

The `*hg*` parameters control the coupling of a spin-0 resonance to gluons in the production mechanism. In practice, the production parameters are not having a large effect since angular corrections from the production mechanism are lost for spinless particles. The `*hz*` parameters control the decay into Z and W bosons. One has the options to set the spin-0 couplings either from Eq.(9) or Eq.(11) from Ref. [2]. To switch between the two, use the parameter `generate_as`. We now allow for q^2 dependent form factors as described in Ref. [3] and given in more detail in the equation below:

$$g_i(q_1^2,q_2^2) = g_i^{\rm SM} + g_i' \frac{\Lambda_i^4}{(\Lambda_i^2+|q_1^2|)(\Lambda_i^2+|q_2^2|)}$$

$$+g_i^{\prime 2}\frac{(|q_1^2|+|q_2^2|)}{\Lambda_i^2} + g_i^{\prime 3}\frac{(|q_1^2|-|q_2^2|)}{\Lambda_i^2} + g_i^{\prime 4}\frac{|(q_1+q_2)^2|}{\Lambda_Q^2} + g_i^{\prime 5}\frac{(|q_1^2|^2+|q_2^2|^2)}{\Lambda_i^4} + g_i^{\prime 6}\frac{(|q_1^2|^2-|q_2^2|^2)}{\Lambda_i^4} + g_i^{\prime 7}\frac{|q_1^2|\,|q_2^2|}{\Lambda_i^4}$$

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 $g_i''...g_i'''''''$ correspond to an expansion in Λ^2. All parameters can be modified in `mod_Parameters.F90` by:

```
!-- parameters that define on-shell spin 0 coupling to SM fields, see note
   logical, public, parameter :: generate_as = .false.
   complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
   complex(8), public, parameter :: ahg3 = (0.0d0,0d0)  ! pseudoscalar
   complex(8), public, parameter :: ahz1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahz2 = (0.0d0,0d0)  ! this coupling does not contribute for
                                                        ! gamma+gamma final states
   complex(8), public, parameter :: ahz3 = (0.0d0,0d0)  ! pseudoscalar

!-- parameters that define off-shell spin 0 coupling to SM fields, see note
   complex(8), public, parameter :: ghg2 = (1.0d0,0d0)
   complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghg4 = (0.0d0,0d0)   ! pseudoscalar
   complex(8), public, parameter :: ghz1 = (2.0d0,0d0)
   complex(8), public, parameter :: ghz2 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz4 = (0.0d0,0d0)   ! pseudoscalar

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

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

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

   complex(8), public, parameter :: ghz4_prime = (0.0d0,0d0)
   complex(8), public, parameter :: ghz4_prime2= (0.0d0,0d0)
   complex(8), public, parameter :: ghz4_prime3= (0.0d0,0d0)
   complex(8), public, parameter :: ghz4_prime4= (0.0d0,0d0)
   complex(8), public, parameter :: ghz3_prime5= (0.0d0,0d0)
   complex(8), public, parameter :: ghz3_prime6= (0.0d0,0d0)
   complex(8), public, parameter :: ghz3_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
```

If the switch `includeGammaStar` is set to `.true.` then intermediate off-shell photons are included for Z boson final states. Their couplings to the spin-0 resonance are controlled by separate parameters,

```
   complex(8), public, parameter :: ghzgs2  = (0.00d0,0d0)
   complex(8), public, parameter :: ghzgs3  = (0.00d0,0d0)
   complex(8), public, parameter :: ghzgs4  = (0.00d0,0d0)
   complex(8), public, parameter :: ghgsgs2 = (0.00d0,0d0)
   complex(8), public, parameter :: ghgsgs3 = (0.00d0,0d0)
   complex(8), public, parameter :: 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 controll the coupling strength in final states with on-shell photons, i.e. $Z\gamma$ and $\gamma\gamma$. The anomalous coupling involving the off-shell photon momentum (in $\gamma^* Z$ interactions)

$$g_1' \frac{q_\gamma^2}{\Lambda_2^{Z\gamma}} m_Z^2 \epsilon_1^* \epsilon_2^*$$

is set by

```
   complex(8), public, parameter :: ghzgs1_prime2= (0.0d0,0d0)
   real(8),    public, parameter :: Lambda_z5 = 10000d0*GeV.
```

In the weak vector boson fusion process we also allow for different ZZH and WWH couplings. Per default, they are assumed to be equal, set by the `ghzi` and `ghzi_primej` couplings. If any of the `ghwi` or `ghwi_primej` couplings is different from zero, their value will be used instead for the WWH interactions.

3. Spin-1 parameters

The parameters below represent the couplings given in Eq. (16) from Ref. [2]. The `*left*` and `*right*` parameters control the production of the spin-1 resonance while the `*_v` and `*_a` parameters control the decay.

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

4. Spin-2 parameters

The `a*` parameters control the coupling of a spin-2 resonance to gluons in the production mechanism. The `b*` and `c*` parameters control the decay. One has the options to set the spin-2 couplings either from Eq.(18) or Eq.(19) from Ref. [2]. To switch between the two, use the parameter `generate_bis`.

```
  logical, public, parameter :: generate_bis = .true.
  logical, public, parameter :: use_dynamic_MG = .true. ! .true. (=default),
    ! the spin-2 resonance mass with MG^2=(p1+p2)^2, otherwise fixed at M_Reso^2.

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

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

  complex(8), public, parameter :: b1 = (1.0d0,0d0)
  complex(8), public, parameter :: b2 = (0.0d0,0d0)
  complex(8), public, parameter :: b3 = (0.0d0,0d0)
  complex(8), public, parameter :: b4 = (0.0d0,0d0)
  complex(8), public, parameter :: b5 = (0.0d0,0d0)
  complex(8), public, parameter :: b6 = (0.0d0,0d0)
  complex(8), public, parameter :: b7 = (0.0d0,0d0)
  complex(8), public, parameter :: b8 = (0.0d0,0d0)
  complex(8), public, parameter :: b9 = (0.0d0,0d0)
  complex(8), public, parameter :: b10 =(0.0d0,0d0)


  complex(8), public, parameter  :: c1 = (1.0d0,0d0)
  complex(8), public, parameter  :: c2 = (0.0d0,0d0)
  complex(8), public, parameter  :: c3 = (0.0d0,0d0)
  complex(8), public, parameter  :: c41= (0.0d0,0d0)
  complex(8), public, parameter  :: c42= (0.0d0,0d0)
  complex(8), public, parameter  :: c5 = (0.0d0,0d0)
  complex(8), public, parameter  :: c6 = (0.0d0,0d0)
  complex(8), public, parameter  :: c7 = (0.0d0,0d0)
```

III. 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

```
logical, public, parameter :: generate_as = .true.

complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
complex(8), public, parameter :: ahg3 = (0.0d0,0d0)   ! pseudoscalar
complex(8), public, parameter :: ahz1 = (1.0d0,0d0)
complex(8), public, parameter :: ahz2 = (0.0d0,0d0)
complex(8), public, parameter :: ahz3 = (0.0d0,0d0)   ! pseudoscalar
```

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

```
logical, public, parameter :: generate_as = .true.

complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
complex(8), public, parameter :: ahg3 = (0.0d0,0d0)   ! pseudoscalar
complex(8), public, parameter :: ahz1 = (0.0d0,0d0)
complex(8), public, parameter :: ahz2 = (0.0d0,0d0)
complex(8), public, parameter :: ahz3 = (1.0d0,0d0)   ! pseudoscalar
```

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

In practice, the example $X \to \gamma\gamma$ from this section, Sec. III C and the next Sec. III D are kinematically the same but are presented only to illustrate how one takes care of this final state.

```
logical, public, parameter :: generate_as = .false.

complex(8), public, parameter :: ghg2 = (1.0d0,0d0)
complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
complex(8), public, parameter :: ghg4 = (0.0d0,0d0)    ! pseudoscalar
complex(8), public, parameter :: ghgsgs2 = (1.0d0,0d0)
complex(8), public, parameter :: ghgsgs3 = (0.0d0,0d0)
complex(8), public, parameter :: ghgsgs4 = (0.0d0,0d0)    ! pseudoscalar
```

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

```
logical, public, parameter :: generate_as = .false.

complex(8), public, parameter :: ghg2 = (1.0d0,0d0)
complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
complex(8), public, parameter :: ghg4 = (0.0d0,0d0)    ! pseudoscalar
complex(8), public, parameter :: ghgsgs2 = (1.0d0,0d0)
complex(8), public, parameter :: ghgsgs3 = (0.0d0,0d0)
complex(8), public, parameter :: ghgsgs4 = (0.0d0,0d0)    ! pseudoscalar
```

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

```
complex(8), public, parameter :: a1 = (1.0d0,0d0)    ! g1  -- c.f. draft
complex(8), public, parameter :: a2 = (0.0d0,0d0)    ! g2
complex(8), public, parameter :: a3 = (0.0d0,0d0)    ! g3
complex(8), public, parameter :: a4 = (0.0d0,0d0)    ! g4
complex(8), public, parameter :: a5 = (0.0d0,0d0)    ! pseudoscalar, g8
complex(8), public, parameter :: graviton_qq_left  = (1.0d0,0d0)! graviton coupling to quarks
complex(8), public, parameter :: graviton_qq_right = (1.0d0,0d0)

logical, public, parameter :: generate_bis = .true.
logical, public, parameter :: use_dynamic_MG = .true.

complex(8), public, parameter :: b1 = (1.0d0,0d0)
complex(8), public, parameter :: b2 = (0.0d0,0d0)
complex(8), public, parameter :: b3 = (0.0d0,0d0)
complex(8), public, parameter :: b4 = (0.0d0,0d0)
complex(8), public, parameter :: b5 = (1.0d0,0d0)
complex(8), public, parameter :: b6 = (0.0d0,0d0)
complex(8), public, parameter :: b7 = (0.0d0,0d0)
complex(8), public, parameter :: b8 = (0.0d0,0d0)
complex(8), public, parameter :: b9 = (0.0d0,0d0)
complex(8), public, parameter :: b10 =(0.0d0,0d0)
```

F. 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 ($g_i = 1$, $g_{j\neq i} = 0$).

G. LHAPDF

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

- In the makefile:
  - Make sure the compiler is ifort. LHAPDF integration is currently not supported with gfortran.
  - Set UseLHAPDF=Yes
  - Set MyLHADir to a directory with your LHAPDF setup. This can be in terms of environment variables; for example `MyLHADir=${LHAPDF_DATA_PATH}/../../lib/`.
- Ensure that `$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`

IV. JHU GENERATOR MATRIX ELEMENTS (JHUGENMELA)

A. Native matrix elements

After extracting the code, you can go to the directory `JHUGenMELA` to find code for computing matrix elements directly. To compile the code, simple do:

```
$ make
```

Please take note: The setup is configured for `gfort + gcc version 4.1.2 20080704 (Red Hat 4.1.2-50)` **and it is highly dependent on the compiler version. Please configure for your own setup accordingly.** (Using `'nm'` command will help decipher the module names you will need)

The usage of the package is straight-forward and an example is given in `testprogram.c`. There are 6 main modules allowing both specific production process and production-independent calculation:

- "modhiggs__evalamp_gg_h_vv": spin-0 matrix elements for gg initiated processes
- "modzprime__evalamp_qqb_zprime_vv": spin-1 matrix elements for $q\bar{q}$ initiated processes
- "modgraviton__evalamp_gg_g_vv:" spin-2 matrix elements for gg initiated processes
- "modgraviton__evalamp_qqb_g_vv": spin-2 matrix elements for $q\bar{q}$ initiated processes
- "modzprime__evalamp_zprime_vv": spin-1 matrix elements production-independent
- "modgraviton__evalamp_g_vv": spin-2 matrix elements production-independent

The inputs are the 4-vectors of the incoming patrons and outgoing particles in the CM frame of the object X. In addition the mass and width of the resonance are required as well as the ID of the outgoing particles. Finally the last set of inputs are the couplings themselves. They are arrays for parameters for a given spin hypothesis which mirror the parameters configurable in `mod_Parameters.F90`. As an example, the arrays are initialized in `testprogram.c`.

B. Interface with MCFM

Instructions for setting up the JHUGenMELA with MCFM are in the file `JHUGenMELA/ggZZ_MCFM/README`.

V. RELEASE NOTES

In going from `v5.6.3` to `v6.2.8`, 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`, 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`, 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`, 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 off-shell 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`, 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`, 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`, 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`, 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`, 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`, 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
- $q\bar{q} \rightarrow$ 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`, 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`, 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 off-shell 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. "SM-like spin-zero", 0^+

```
!-- parameters that define on-shell spin 0 coupling to SM fields, see note
   logical, public, parameter :: generate_as = .false.
   complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
   complex(8), public, parameter :: ahg3 = (0.0d0,0d0)  ! pseudoscalar
   complex(8), public, parameter :: ahz1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahz2 = (0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states
   complex(8), public, parameter :: ahz3 = (0.0d0,0d0)  ! pseudoscalar

!-- parameters that define off-shell spin 0 coupling to SM fields, see note
   complex(8), public, parameter :: ghg2 = (1.0d0,0d0)
   complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghg4 = (0.0d0,0d0)   ! pseudoscalar
   complex(8), public, parameter :: ghz1 = (1.0d0,0d0)
   complex(8), public, parameter :: ghz2 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz4 = (0.0d0,0d0)   ! pseudoscalar
```

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

```
!-- parameters that define on-shell spin 0 coupling to SM fields, see note
   logical, public, parameter :: generate_as = .false.
   complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
   complex(8), public, parameter :: ahg3 = (0.0d0,0d0)  ! pseudoscalar
   complex(8), public, parameter :: ahz1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahz2 = (0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states
   complex(8), public, parameter :: ahz3 = (0.0d0,0d0)  ! pseudoscalar

!-- parameters that define off-shell spin 0 coupling to SM fields, see note
   complex(8), public, parameter :: ghg2 = (1.0d0,0d0)
   complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghg4 = (0.0d0,0d0)   ! pseudoscalar
   complex(8), public, parameter :: ghz1 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz2 = (1.0d0,0d0)
   complex(8), public, parameter :: ghz3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz4 = (0.0d0,0d0)   ! pseudoscalar
```

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

```
!-- parameters that define on-shell spin 0 coupling to SM fields, see note
   logical, public, parameter :: generate_as = .false.
   complex(8), public, parameter :: ahg1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahg2 = (0.0d0,0d0)
   complex(8), public, parameter :: ahg3 = (0.0d0,0d0)  ! pseudoscalar
   complex(8), public, parameter :: ahz1 = (1.0d0,0d0)
   complex(8), public, parameter :: ahz2 = (0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states
   complex(8), public, parameter :: ahz3 = (0.0d0,0d0)  ! pseudoscalar

!-- parameters that define off-shell spin 0 coupling to SM fields, see note
   complex(8), public, parameter :: ghg2 = (0.0d0,0d0)
   complex(8), public, parameter :: ghg3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghg4 = (1.0d0,0d0)   ! pseudoscalar
   complex(8), public, parameter :: ghz1 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz2 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz3 = (0.0d0,0d0)
   complex(8), public, parameter :: ghz4 = (1.0d0,0d0)   ! pseudoscalar
```

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

```
!---parameters that define spin 1 coupling to SM fields, see note
   complex(8), public, parameter :: zprime_qq_left  = (1.0d0,0d0)
   complex(8), public, parameter :: zprime_qq_right = (0.0d0,0d0)
   complex(8), public, parameter :: zprime_zz_v =  (1.0d0,0d0)!  =1 for JP=1-
   complex(8), public, parameter :: zprime_zz_a =  (0.0d0,0d0)!  =1 for JP=1+
```

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

```
!---parameters that define spin 1 coupling to SM fields, see note
   complex(8), public, parameter :: zprime_qq_left  = (1.0d0,0d0)
   complex(8), public, parameter :: zprime_qq_right = (0.0d0,0d0)
   complex(8), public, parameter :: zprime_zz_v =  (0.0d0,0d0)!  =1 for JP=1-
   complex(8), public, parameter :: zprime_zz_a =  (1.0d0,0d0)!  =1 for JP=1+
```

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

N.B. 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.

```
!-- parameters that define spin 2 coupling to SM fields, see note
! minimal coupling corresponds to a1 = b1 = b5 = 1 everything else 0
  complex(8), public, parameter :: a1 = (1.0d0,0d0)    ! g1  -- c.f. draft
  complex(8), public, parameter :: a2 = (0.0d0,0d0)    ! g2
  complex(8), public, parameter :: a3 = (0.0d0,0d0)    ! g3
  complex(8), public, parameter :: a4 = (0.0d0,0d0)    ! g4
  complex(8), public, parameter :: a5 = (0.0d0,0d0)    ! pseudoscalar, g8
  complex(8), public, parameter :: graviton_qq_left  = (1.0d0,0d0)! graviton coupling to quarks
  complex(8), public, parameter :: graviton_qq_right = (1.0d0,0d0)

!-- see mod_Graviton
  logical, public, parameter :: generate_bis = .true.
  logical, public, parameter :: use_dynamic_MG = .true.

  complex(8), public, parameter :: b1 = (1.0d0,0d0)    !  all b' below are g's in the draft
  complex(8), public, parameter :: b2 = (0.0d0,0d0)
  complex(8), public, parameter :: b3 = (0.0d0,0d0)
  complex(8), public, parameter :: b4 = (0.0d0,0d0)
  complex(8), public, parameter :: b5 = (1.0d0,0d0)
  complex(8), public, parameter :: b6 = (0.0d0,0d0)
  complex(8), public, parameter :: b7 = (0.0d0,0d0)
  complex(8), public, parameter :: b8 = (0.0d0,0d0)
  complex(8), public, parameter :: b9 = (0.0d0,0d0)
  complex(8), public, parameter :: b10 =(0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states


  complex(8), public, parameter  :: c1 = (1.0d0,0d0)
  complex(8), public, parameter  :: c2 = (0.0d0,0d0)
  complex(8), public, parameter  :: c3 = (0.0d0,0d0)
  complex(8), public, parameter  :: c41= (0.0d0,0d0)
  complex(8), public, parameter  :: c42= (0.0d0,0d0)
  complex(8), public, parameter  :: c5 = (0.0d0,0d0)
  complex(8), public, parameter  :: c6 = (0.0d0,0d0)
  complex(8), public, parameter  :: c7 = (0.0d0,0d0)
```

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

```
!-- parameters that define spin 2 coupling to SM fields, see note
! minimal coupling corresponds to a1 = b1 = b5 = 1 everything else 0
  complex(8), public, parameter :: a1 = (0.0d0,0d0)    ! g1  -- c.f. draft
  complex(8), public, parameter :: a2 = (0.0d0,0d0)    ! g2
  complex(8), public, parameter :: a3 = (0.0d0,0d0)    ! g3
  complex(8), public, parameter :: a4 = (1.0d0,0d0)    ! g4
  complex(8), public, parameter :: a5 = (0.0d0,0d0)    ! pseudoscalar, g8
  complex(8), public, parameter :: graviton_qq_left  = (1.0d0,0d0)! graviton coupling to quarks
  complex(8), public, parameter :: graviton_qq_right = (1.0d0,0d0)

!-- see mod_Graviton
  logical, public, parameter :: generate_bis = .true.
  logical, public, parameter :: use_dynamic_MG = .true.

  complex(8), public, parameter :: b1 = (0.0d0,0d0)    !  all b' below are g's in the draft
  complex(8), public, parameter :: b2 = (0.0d0,0d0)
  complex(8), public, parameter :: b3 = (0.0d0,0d0)
  complex(8), public, parameter :: b4 = (1.0d0,0d0)
  complex(8), public, parameter :: b5 = (0.0d0,0d0)
  complex(8), public, parameter :: b6 = (0.0d0,0d0)
  complex(8), public, parameter :: b7 = (0.0d0,0d0)
  complex(8), public, parameter :: b8 = (0.0d0,0d0)
  complex(8), public, parameter :: b9 = (0.0d0,0d0)
  complex(8), public, parameter :: b10 =(0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states


  complex(8), public, parameter  :: c1 = (1.0d0,0d0)
  complex(8), public, parameter  :: c2 = (0.0d0,0d0)
  complex(8), public, parameter  :: c3 = (0.0d0,0d0)
  complex(8), public, parameter  :: c41= (0.0d0,0d0)
  complex(8), public, parameter  :: c42= (0.0d0,0d0)
  complex(8), public, parameter  :: c5 = (0.0d0,0d0)
  complex(8), public, parameter  :: c6 = (0.0d0,0d0)
  complex(8), public, parameter  :: c7 = (0.0d0,0d0)
```

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

```
!-- parameters that define spin 2 coupling to SM fields, see note
! minimal coupling corresponds to a1 = b1 = b5 = 1 everything else 0
  complex(8), public, parameter :: a1 = (0.0d0,0d0)    ! g1  -- c.f. draft
  complex(8), public, parameter :: a2 = (0.0d0,0d0)    ! g2
  complex(8), public, parameter :: a3 = (0.0d0,0d0)    ! g3
  complex(8), public, parameter :: a4 = (0.0d0,0d0)    ! g4
  complex(8), public, parameter :: a5 = (1.0d0,0d0)    ! pseudoscalar, g8
  complex(8), public, parameter :: graviton_qq_left  = (1.0d0,0d0)! graviton coupling to quarks
  complex(8), public, parameter :: graviton_qq_right = (1.0d0,0d0)

!-- see mod_Graviton
  logical, public, parameter :: generate_bis = .true.
  logical, public, parameter :: use_dynamic_MG = .true.

  complex(8), public, parameter :: b1 = (0.0d0,0d0)    !  all b' below are g's in the draft
  complex(8), public, parameter :: b2 = (0.0d0,0d0)
  complex(8), public, parameter :: b3 = (0.0d0,0d0)
  complex(8), public, parameter :: b4 = (0.0d0,0d0)
  complex(8), public, parameter :: b5 = (0.0d0,0d0)
  complex(8), public, parameter :: b6 = (0.0d0,0d0)
  complex(8), public, parameter :: b7 = (0.0d0,0d0)
  complex(8), public, parameter :: b8 = (1.0d0,0d0)
  complex(8), public, parameter :: b9 = (0.0d0,0d0)
  complex(8), public, parameter :: b10 =(0.0d0,0d0)  ! this coupling does not contribute for gamma+gamma final states


  complex(8), public, parameter  :: c1 = (1.0d0,0d0)
  complex(8), public, parameter  :: c2 = (0.0d0,0d0)
  complex(8), public, parameter  :: c3 = (0.0d0,0d0)
  complex(8), public, parameter  :: c41= (0.0d0,0d0)
  complex(8), public, parameter  :: c42= (0.0d0,0d0)
  complex(8), public, parameter  :: c5 = (0.0d0,0d0)
  complex(8), public, parameter  :: c6 = (0.0d0,0d0)
  complex(8), public, parameter  :: c7 = (0.0d0,0d0)
```

---

[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] See webpage: www.pha.jhu.edu/spin

[5] 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]].

[6] 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