

Manual for the JHU generator

For simulation of a single-produced resonance at hadron colliders
(version 4.2.1, release date November 2, 2013)

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 generator.

The code can be downloaded from [4]. The generator outputs LHE files which can be passed to parton shower programs for hadronization. The generator purposely does not output sensible cross-sections but rather deals with numbers of events which can be compared for different signal hypotheses.

Additionally, the package now includes code for computing the matrix elements standalone which can be used in a numerical matrix element analysis.

Contents

I. Installation	2
II. Configuration	2
A. Command line configuration	2
B. Configuration in parameter file	3
1. General parameters	3
2. Spin-0 parameters	4
3. Spin-1 parameters	5
4. Spin-2 parameters	5
III. Examples	5
A. $J^P = 0_m^+$ resonance, $X \rightarrow ZZ$ or WW	6
B. $J^P = 0_m^-$ resonance, $X \rightarrow ZZ$ or WW	6
C. $J^P = 0_m^+$ resonance, $X \rightarrow \gamma\gamma$	6
D. $J^P = 0_m^-$ resonance, $X \rightarrow \gamma\gamma$	6
E. $J^P = 2_m^+$ resonance, $X \rightarrow ZZ$ or WW or $\gamma\gamma$	6
F. Cross-section calculation and fraction notation	7
IV. JHU Generator Matrix Elements (JHUGenMELA)	7
A. Native matrix elements	7
B. Interface with MCFM	8
V. Release notes	8
A. Specific configurations	10
1. "SM-like spin-zero", 0^+	10
2. "Higher order spin-zero", 0_h^+	10
3. "Pseudoscalar spin-zero", 0^-	10
4. "Vector spin-one", 1^-	11
5. "Pseudovector spin-one", 1^+	11
6. "Minimal Graviton, spin-two", 2^+	11
7. "Higher order Graviton, spin-two", 2_h^+	11
8. "Higher order Graviton, spin-two", 2_h^-	12
References	13

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, either from the command line or in the file `mod_Parameters.F90`. 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=x (Collider: 1=LHC, 2=Tevatron), default value:1
- VegasNc1=x (number of evaluations), default value:5000000
- PChannel=x (partonic channel: 0=glu+glu, 1=quark+antiquark, 2=both), default value:2
- PDFSet=x (parton distr. functions, 1=CTEQ6L1 (2001), 2=MSTW(2008),
2xx=MSTW with eigenvector set xx=01..40), default=1
- DecayMode1=x (decay mode for vector boson 1), default x=0
- DecayMode2=x (decay mode for vector boson 2), default x=0
 - x=0: X-> Z1 Z2, Z1->2l,
 - x=1: X-> Z1 Z2, Z1->2q,
 - x=2: X-> Z1 Z2, Z1->2tau,
 - x=3: X-> Z1 Z2, Z1->2nu,
 - x=4: X-> W1 W2, W1->l nu,
 - x=5: X-> W1 W2, W1->2q,
 - x=6: X-> W1 W2, W1->taunu
 - x=7: X-> gamma gamma
 - x=8: Z --> l+ l- (l=e,mu,tau)
 - x=9: Z --> anything
 - x=10: W --> l nu_l (l=e,mu,tau)
 - x=11: W --> anything
- DataFile=x (name of the output file), default value: "output"
- Process=x (gives the resonance spin 0,1 or 2, gg production; \\
60 = VBF production of spin-0; 61 = Hjj QCD)
- OffXVV=xyz (only applies to Higgs production, x,y,z can be 1 or 0 indicating real off-shellness
for Higgs, Z/W boson 1/2, resp.), default 000 if a parameter is not specified
- Unweighted=x (generates weighted or unweighted events), default value: ".true."
- ReadLHE="x" (allows NLO production of spin-0 particles by reading in
LHE file from NLO generator, e.g., POWHEG)

A few more details on some particular parameters:

- **VegasNc1**: This is the number of evaluations for the program to make, NOT the number of events generated. The efficiency depends on the set of parameters one uses.
- **OffXVV**: The program does not work for ZZ or WW if you set them to be on-shell (`OffXVV="*00"`) 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. For spin-0, production is possible only via the gg process and for spin-1, production is only possible via the $q\bar{q}$ process.
- **DecayMode2=7** note: Valid for spin-0 and spin-2, only OffXVV=000 or 100 are possible.
- **VegasNc** note: **VegasNc1** allows the user to set the number of tries for the reject/accept method, **VegasNc2** allows the user to let the program run until the number of events is generated

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=.true. DataFile=test1
```

- 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
```

B. Configuration in parameter file

The `mod_Parameters.F90`, one does all the configuration of the couplings of the resonance.

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 the 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)
```

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
```

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

```
! we are using units of 100GeV, i.e. Lambda=10 is 1TeV  
M_Z    = 91.1876d0 *GeV      ! Z boson mass (PDG-2011)  
Ga_Z    = 2.4952d0 *GeV      ! Z boson width(PDG-2011)  
M_W    = 80.399d0 *GeV      ! W boson mass (PDG-2011)  
Ga_W    = 2.085d0 *GeV      ! W boson width(PDG-2011)  
M_Reso  = 125d0 *GeV         ! X resonance mass (spin 0, spin 1, spin 2)  
Ga_Reso = 5d0 *GeV          ! X resonance width  
Lambda  = 1000d0 *GeV       ! Lambda coupling enters in two places  
! overall scale for x-section and in power suppressed  
! operators/formfactors (former r).  
alpha_QED = 1d0/128.0d0      ! el.magn. coupling  
sitW = dsqrt(0.23119d0)      ! sin(Theta_Weinberg) (PDG-2008)  
Mu_Fact = M_Reso             ! pdf factorization scale  
LHC_Energy=7000d0 *GeV       ! LHC hadronic center of mass energy
```

```

TEV_Energy=1960d0  *GeV          ! Tevatron hadronic center of mass energy

Br_Z_up = 0.1657d0  ! branching fraction Ga(up)/Ga(hadronic)
Br_Z_ch = 0.1657d0  ! branching fraction Ga(charm)/Ga(hadronic)
Br_Z_dn = 0.2229d0  ! branching fraction Ga(down)/Ga(hadronic)
Br_Z_st = 0.2229d0  ! branching fraction Ga(strange)/Ga(hadronic)
Br_Z_bo = 1d0-Br_Z_up-Br_Z_ch-Br_Z_dn-Br_Z_st ! branching fraction Ga(bottom)/Ga(hadronic)

```

2. Spin-0 parameters

The ***hg*** parameters control the coupling of a spin-0 resonance to gluons in the production mechanism. The ***hz*** parameters control the decay. In practice, the production parameters are not having a large effect since angular corrections from the production mechanism are lost for spinless particles. 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**. For the parameters in Eq.(9) from Ref. [2], we now allow for q^2 dependent form factors as described in Eq.(3) of [3] as:

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

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
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 :: ghg2_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghg3_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghg4_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghz1_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghz2_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghz3_prime = (0.0d0,0d0)
complex(8), public, parameter :: ghz4_prime = (0.0d0,0d0)
real(8), public, parameter :: Lambda_g2 = 10000d0*GeV
real(8), public, parameter :: Lambda_g3 = 10000d0*GeV
real(8), public, parameter :: Lambda_g4 = 10000d0*GeV
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

```

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, $X \rightarrow ZZ$ 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, $X \rightarrow ZZ$ 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, $X \rightarrow \gamma\gamma$

In practice, the example $X \rightarrow \gamma\gamma$ from this section, Sec. IIIC and the next Sec. IIID 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 :: 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
```

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

```
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 :: 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
```

E. $J^P = 2_m^+$ resonance, $X \rightarrow ZZ$ 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)

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

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

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` as:

```
double Hggcoupl[3][2];
double Hvvcoupl[4][2];
double Zqqcoupl[2][2];
double Zvvcoupl[2][2];
double Gqqcoupl[2][2];
double Gggcoupl[5][2];
double Gvvcoupl[10][2];
```

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 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 $ee \rightarrow ZH$ and $pp \rightarrow ZH$ 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, the file headers, and minor cleanup.
- Updates to deal with non-zero lepton masses, the 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". arXiv:1309.4819 [hep-ph].
 - [4] See webpage: www.pha.jhu.edu/spin