

H1jet, a fast program to compute transverse momentum distributions

Article (Published Version)

Lind, Alexander and Banfi, Andrea (2021) H1jet, a fast program to compute transverse momentum distributions. *European Physical Journal C: Particles and Fields*, 81 (a72). pp. 1-16. ISSN 1434-6044

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/97306/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

Copyright and reuse:

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.



H1JET, a fast program to compute transverse momentum distributions

Alexander Lind^{1,a} , Andrea Banfi^{1,b}

¹ Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, UK

Received: 26 November 2020 / Accepted: 28 December 2020
© The Author(s) 2021

Abstract We present H1JET, a fast code that computes the total cross section and differential distribution in the transverse momentum of a colour singlet. In its current version, the program implements only leading-order $2 \rightarrow 1$ and $2 \rightarrow 2$ processes, but could be extended to higher orders. We discuss the processes implemented in H1JET, give detailed instructions on how to implement new processes, and perform comparisons to existing codes. This tool, mainly designed for theorists, can be fruitfully used to assess deviations of selected new physics models from the Standard Model behaviour, as well as to quickly obtain distributions of relevance for Standard Model phenomenology.

Contents

1	Introduction
2	The method
3	User's manual
3.1	Installation
3.2	Usage
3.2.1	General options
3.2.2	Relevant options for process: H
3.2.3	Relevant options for process: $b\bar{b}H$
3.2.4	Relevant options for process: Z
3.2.5	Relevant options for process: <i>user</i>
3.2.6	Loop approximations
3.2.7	Output
4	Benchmarking
5	Adding new processes to H1JET
5.1	Example: axion-like-particle (ALP) effective theory
5.1.1	The total cross section
6	Conclusions

H1JET can be obtained from Ref. [1].

^a e-mail: a.lind@sussex.ac.uk (corresponding author)

^b e-mail: a.banfi@sussex.ac.uk

A	Implementation of scalar integrals
	References

1 Introduction

After the discovery of the Higgs boson in 2012 [2, 3], one of the most urgent tasks of the Large Hadron Collider (LHC) is the characterisation of the Higgs sector, in order to shed light on the exact mechanism for electroweak symmetry breaking. In particular, Higgs production data, either in the form of signal strengths [4, 5] or cross section measurements [6, 7], offer powerful constraints on Higgs anomalous couplings.

However, it has been pointed out that inclusive Higgs production through gluon fusion, the one with the largest rate, is not able to discriminate effectively between the Standard Model (SM) and another theory giving the same effective coupling between the gluons and the Higgs. In fact, top quarks running in loops give a dimension-6 effective interaction between the incoming gluons and the Higgs, with exact top-mass effects giving tiny corrections. Therefore, in many theories, the strength of dimension-6 contact gluon-gluon-Higgs interactions can conspire with an anomalous top Higgs Yukawa coupling to give exactly the same cross section for Higgs production as the SM [8–10].

There are essentially two ways of solving this problem. One is to put a direct constraint on the top Yukawa coupling by the observation of top quarks in association with the Higgs [11–13]. The other is to break the top loop by looking e.g. at Higgs production at large transverse momentum, where the Higgs recoils against a hard jet [8–10]. Both are indirect probes of new physics effects. The latter is more difficult experimentally, in that it relies on appreciating small deviations from the SM in the shape of the Higgs transverse momentum distribution, in a region where the phase space closes. It nevertheless can give a direct access to new physics coupling the gluons to the Higgs through loops.

Higgs sector aside, production of colour singlets at high transverse momentum is commonly used as a probe of new physics. A relevant example is the production of monojets, which can recoil either against dark matter, or against a SM particle decaying into invisible particles (see e.g. Refs. [14, 15]).

Theoretical predictions for the transverse momentum distribution of a colour singlet, both in the SM and beyond, can be currently obtained with Monte Carlo programs, such as MADGRAPH5_AMC@NLO [16] or SUSHI [17, 18]. These codes, although general, have the drawback of being quite slow. Also, interference terms between new physics and the SM, which carry information on the strength of new interactions, are difficult to extract from Monte Carlo event generators because they are not positive definite. The aim of this paper is to describe a method to obtain the transverse momentum spectrum of a colour singlet in a second or less, and its concrete implementation in the program H1JET. This program makes it possible to predict the effects of several models in a short amount of time. This in turn opens the way to devising more refined cut-based search strategies only for the models showing the largest deviations with respect to the SM.

More precisely, H1JET predicts the transverse momentum distribution of a colour singlet fully integrated over rapidity, and completely inclusive with respect to all coloured particles, i.e. the recoiling jets. Such an approximation is not too unrealistic, because the higher the transverse momentum, the more the colour singlet is central, and the more its decay products will be likely to pass the detector acceptance cuts.

The program is based on elementary analytic manipulations on the expression for the transverse momentum distribution. These make it possible to write the spectrum as a one dimensional integral, whose integrand is the product of an amplitude squared, which can be provided by ourselves or by the user, and a parton luminosity, which we extract from an external program. The relevant amplitudes can be either hard coded, or computed automatically and embedded in the program via a simple user interface.

H1JET already comes with a number of hard-coded processes and models. The main process is

$$pp \rightarrow H + \text{jet}, \quad (1)$$

where the initial state consists of gluons and light quarks, and Higgs production proceeds via quark loops. This process can be calculated in H1JET for different physics models including the SM, a CP-odd Higgs, a simplified SUSY model, and composite Higgs models with a single or multiple top-partners. In addition, the $b\bar{b} \rightarrow H + \text{jet}$ and $pp \rightarrow Z + \text{jet}$ processes for the SM are implemented. Moreover, H1JET is very flexible and can be easily interfaced to use a custom user-specified process.

The paper is organised as follows. In Sect. 2 we briefly describe the method underlying H1JET. In Sect. 3 we describe in detail how H1JET works in practice. In particular, we show how it can be installed and run, and present the features currently implemented. In Sect. 4 we present a detailed comparison with the existing program SUSHI for Higgs production both in the SM and beyond. In Sect. 5 we explain how a user can implement a model of new physics inside H1JET. We choose axion-like-particle (ALP) production, giving rise to a monojet. We then describe how to obtain ALP transverse momentum distributions from the generation of the Feynman rules with FEYNRULES, to the calculation of the amplitude with FEYNCALC and its subsequent interface with H1JET to obtain the ALP transverse momentum spectrum. Last, Sect. 6 presents our conclusions.

2 The method

Before explaining how the H1JET method works, it is instructive to consider first how to compute the Born cross section for producing a colour singlet X , e.g. a Higgs, of mass m_X . This will also allow us to set the notation for the rest of the paper. We consider the $2 \rightarrow 1$ process $p_1 p_2 \rightarrow X$, where p_1 and p_2 are the two incoming partons and p_X is the momentum of the considered colour singlet. From momentum conservation we have

$$\hat{s} = (p_1 + p_2)^2 = m_X^2. \quad (2)$$

There can be various partonic subprocesses that contribute to the production of the particle X . Let us denote with M_{ij} the amplitude for the subprocess $ij \rightarrow X$ (e.g. $gg \rightarrow H$), with $i, j = g, q_f, \bar{q}_{\bar{f}}$, where f, \bar{f} denotes quark or antiquark flavours. The Born partonic cross section for each subprocess is

$$\hat{\sigma}_{ij} = \frac{\pi}{m_X^2} |M_{ij}(m_X^2)|^2 \delta(\hat{s} - m_X^2). \quad (3)$$

The corresponding hadronic cross section is given by

$$\begin{aligned} \sigma &= \frac{\pi}{m_X^2} \sum_{i,j} \int_0^1 dx_1 f_{i/p}(x_1, \mu_F) \int_0^1 dx_2 f_{j/p}(x_2, \mu_F) \\ &\quad \times \sum_{i,j} |M_{ij}(m_X^2)|^2 \delta(x_1 x_2 s - m_X^2) \\ &= \frac{\pi}{m_X^4} \sum_{i,j} |M_{ij}(m_X^2)|^2 \mathcal{L}_{ij} \left(\frac{m_X^2}{s}, \mu_F \right), \end{aligned} \quad (4)$$

where $\mathcal{L}_{ij}(\tau, \mu_F)$ is the partonic luminosity

$$\mathcal{L}_{ij}(\tau, \mu_F) = \tau \int_{\tau}^1 \frac{dx}{x} f_{i/p}(x, \mu_F) f_{j/p} \left(\frac{\tau}{x}, \mu_F \right). \quad (5)$$

If we are able to obtain the luminosity $\mathcal{L}_{ij}(m_X^2/s, \mu_F)$, we are then able to obtain a numerical prediction for the cross section through a simple multiplication. There are indeed numerical tools that are able to compute, tabulate and interpolate luminosities with incredible efficiency, for instance the program HOPPET [19]. Through an interface with HOPPET, we are able to compute the Born cross section given the amplitudes M_{ij} . This procedure is the same adopted in the program JETVHETO [20], that computes cross sections for colour singlets with a veto on additional jets.

A similar strategy can be devised to obtain a fast calculation of distributions in the transverse momentum of particle X . A non-zero transverse momentum for X is obtained via a generic $2 \rightarrow 2$ partonic process $p_1 p_2 \rightarrow p_3 X$, where p_1, p_2 , and p_3 are massless partons, and p_X is the momentum of the colour singlet X . We wish to compute $d\sigma/dp_T$, where p_T is the transverse momentum of p_X with respect to the beam axis. At Born level only, p_T is also the transverse momentum of the recoiling jet originated by p_3 . The partonic subprocesses contributing to $d\sigma/dp_T$ are $gg \rightarrow gX, q_f \bar{q}_{\bar{f}} \rightarrow gX, q_f g \rightarrow q_f X, gq_f \rightarrow q_f X$. The corresponding amplitudes M_{ij} (with $i, j = g, q_f, \bar{q}_{\bar{f}}$) are functions of the three Mandelstam invariants

$$\begin{aligned} \hat{s} &= (p_1 + p_2)^2 = (p_3 + p_X)^2, \\ \hat{t} &= (p_1 - p_3)^2 = (p_2 - p_X)^2, \\ \hat{u} &= (p_2 - p_3)^2 = (p_1 - p_X)^2. \end{aligned} \tag{6}$$

Without loss of generality, in the centre-of-mass frame of the partonic collision, we can parameterise momenta as follows

$$\begin{aligned} p_1 &= \frac{\sqrt{\hat{s}}}{2}(1, 0, 0, 1), \\ p_2 &= \frac{\sqrt{\hat{s}}}{2}(1, 0, 0, -1), \\ p_3 &= p_T(\cosh \eta, 1, 0, \sinh \eta), \\ p_X &= \left(\sqrt{m_X^2 + p_T^2 \cosh^2 \eta}, -p_T, 0, -p_T \sinh \eta \right), \end{aligned} \tag{7}$$

where η is the rapidity of parton p_3 in the centre-of-mass frame. The partonic p_T spectrum for the process initiated by partons ij is given by

$$\begin{aligned} \frac{d\hat{\sigma}_{ij}}{dp_T} &= \frac{p_T}{16\pi} \int d\eta \frac{|M_{ij}(\hat{s}, \hat{t}, \hat{u})|^2}{E_X \hat{s}} \\ &\times \delta \left(\sqrt{\hat{s}} - p_T \cosh \eta - \sqrt{m_X^2 + p_T^2 \cosh^2 \eta} \right), \end{aligned} \tag{8}$$

where $E_X = \sqrt{m_X^2 + p_T^2 \cosh^2 \eta}$ is the energy of the colour-singlet particle p_X . The above equation selects two values of η , as follows

$$\eta = \ln \left(\hat{x}_M \pm \sqrt{\hat{x}_M^2 - 1} \right), \quad \hat{x}_M \equiv \frac{\hat{s} - m_X^2}{2p_T \sqrt{\hat{s}}}. \tag{9}$$

The corresponding hadronic cross section reads

$$\begin{aligned} \frac{d\sigma}{dp_T} &= \sum_{i,j} \int_0^1 dx_1 f_{i/p}(x_1, \mu_F) \int_0^1 dx_2 f_{j/p}(x_2, \mu_F) \\ &\times \left[\frac{d\hat{\sigma}_{ij}}{dp_T} \Theta \left(\hat{s} - p_T - \sqrt{m_X^2 + p_T^2} \right) \right]_{\hat{s}=x_1 x_2 s}. \end{aligned} \tag{10}$$

Since Eq. (9) gives two monotonic functions of \hat{s} for $s > p_T + \sqrt{m_X^2 + p_T^2}$, varying \hat{s} in the allowed range spans all possible values of η in the range $-\eta_M < \eta < \eta_M$ with

$$\eta_M \equiv \ln \left(x_M + \sqrt{x_M^2 - 1} \right), \quad x_M \equiv \frac{s - m_X^2}{2p_T \sqrt{s}}. \tag{11}$$

This allows us to perform the η integration last, and obtain, after some manipulations,

$$\begin{aligned} \frac{d\sigma}{dp_T} &= \frac{p_T}{8\pi} \int_{-\eta_M}^{\eta_M} d\eta \sum_{i,j} \left[\frac{M_{ij}^2(\hat{s}, \hat{t} = -p_T e^{-\eta} \sqrt{\hat{s}}, \hat{u} = -p_T e^{\eta} \sqrt{\hat{s}})}{E_X \hat{s}^{3/2}} \right. \\ &\times \mathcal{L}_{ij} \left(\frac{\hat{s}}{s}, \mu_F \right) \left. \right]_{\hat{s}=(p_T \cosh \eta + \sqrt{m_X^2 + p_T^2 \cosh^2 \eta})^2}, \end{aligned} \tag{12}$$

where again \mathcal{L}_{ij} is the partonic luminosity for the ij incoming channel as defined in Eq. (5). If we are able to obtain the partonic luminosity \mathcal{L}_{ij} , say, from HOPPET, we can obtain the transverse momentum spectrum with a one-dimensional integration, which can be performed extremely quickly with a Gaussian numerical integrator.

Summarising, by interfacing HOPPET with a code that provides amplitudes for $2 \rightarrow 1$ and $2 \rightarrow 2$ partonic subprocesses producing a colour singlet X , we are able to perform fast computations of total cross sections and transverse momentum spectra for X . In the following sections we describe our implementation of the method for Born processes. Note that, if one were able to perform the analytic integration over the phase space of final-state partons, the method can also be applied to higher-order cross sections and differential spectra.

3 User’s manual

This section describes the most important technical details of H1JET, including its installation and usage.

3.1 Installation

The source code of H1JET can be obtained from Ref. [1]. The source code consists of a main directory H1jet with the following subdirectories:

`bin`: contains the executable program `h1jet` after compilation, as well as the Python 3 helper scripts `PlotH1jet.py` and `DressUserAmpCode.py`.
`src`: source files.

The `README.md` file contains information on installation and usage.

In the main directory, the user needs to run the configure script:

```
./configure [options]
```

It will attempt to find a Fortran compiler (`gfortran` or `ifort`), as well as the dependencies on the user's machine. A specific compiler and/or compiler flags can be selected with the options `./configure FC=<compiler>` and `./configure FFLAGS=<flags>`. H1JET has a number of external dependencies which it must be linked to:

- LHAPDF [21]: Provides the PDF sets for H1JET.
- HOPPET [19]: For QCD DGLAP evolution of PDFs and numerical integrations.
- CHAPLIN [22]: For complex harmonic polylogarithms used to represent scalar integrals in loop-induced processes.

For the CHAPLIN library, it may be necessary to explicitly state the path to the library files with:

```
./configure LDFLAGS=-L/path/to/chaplin/lib
```

To compile with a custom user interface:

```
./configure  
USERFILE=/path/to/custom/user_interface.f90
```

See Sect. 5 below for the implementation of custom user-specified amplitudes.

To install in a specific location:

```
./configure --prefix=/path/to/installation
```

The default installation path is the main `H1jet`-directory.

The configure script will generate the `Makefile`.

To compile H1JET with the generated `Makefile`, run:

```
make [options]
```

This command takes the following options: `make clean` will delete all module and object files; `make distclean` will delete all module and object files as well as the executable `h1jet`.

After compilation, the `bin`-directory can then be added to the user's `PATH` environment variable. Alternatively, if the user has specified an installation directory with the `--prefix` option, the executable can be installed with:

```
make install
```

The executable `h1jet` can then be found in the `bin`-directory at the path specified by `--prefix`.

3.2 Usage

After compilation, H1JET can be run from the `bin`-directory with:

```
./bin/h1jet [options]
```

H1JET will print out a brief summary of the settings and parameters used, as well as the Born cross section σ_0 , followed by a five-column table. The first three entries of each row specifies the lower end, the midpoint, and the upper end of each p_T bin. The fourth entry is $d\sigma/dp_T$ evaluated at the midpoint of the corresponding p_T bin. The fifth entry is the integrated cross section $\sigma(p_T)$ with a lower bound in p_T corresponding to the lower end of the given p_T bin. We remark that the fundamental object we compute is $d\sigma/dp_T$. The integrated cross section $\sigma(p_T)$ is obtained by summing $d\sigma/dp_T$ over the appropriate range and multiplying by the bin width. Therefore, this procedure gives a reliable estimate of $\sigma(p_T)$ only if the binning is fine enough.

The following standard UNIX options are available:

- `-h, --help` Display the help message along with all possible options.
- `-v, --version` Display the version of the installed H1JET.

H1JET will display the requested information and then terminate.

The output can be directed to a file with the option:

- `-o, --out <file>` Direct the output to `<file>`.
Default: standard output.

The physics process can be selected with:

- `--proc <arg>` Specify the process. Arguments:
 - `H` $pp/p\bar{p} \rightarrow H + \text{jet}$ (default).
 - `bbH` $b\bar{b} \rightarrow H + \text{jet}$.
 - `Z` $pp/p\bar{p} \rightarrow Z + \text{jet}$.
 - `user` User specified process. See Sect. 5 below for details on the implementation.

Depending on the process selected, there exists different relevant options.

3.2.1 General options

The options listed here apply to all processes.

- `--collider <arg>` Specify the collider type.
Arguments: `pp` (default), `ppbar`.

--roots <value> Centre-of-mass energy, \sqrt{s} [GeV].
Default: 13000 GeV.

--pdf_name <arg> Specify the PDF set name from LHAPDF. The specified PDF set must be available in the local installation of LHAPDF.
Default: MSTW2008nlo68cl1.

--pdf_mem <value> Integer value specifying the PDF member.
Default: 0.

--scale_strategy <arg> Set the scale strategy, i.e. the dynamic $\mu = \mu_R = \mu_F$ value.
Arguments:

$$\begin{aligned} \text{M} & \quad \mu = M. \\ \text{HT} & \quad \mu = p_T + \sqrt{p_T^2 + M^2} \text{ (default).} \\ \text{MT} & \quad \mu = \sqrt{p_T^2 + M^2}. \end{aligned}$$

The mass M is given by option --mH for processes H and bbH, option --mZ for process Z, and option --mass for process user.

--xmur <value> Additional factor x_R for the renormalisation scale, i.e. $\mu_R = x_R \mu$, where μ is determined by the choice from --scale_strategy.
Default: 0.5.

--xmuf <value> Additional factor x_F for the factorisation scale, i.e. $\mu_F = x_F \mu$, where μ is determined by the choice from --scale_strategy.
Default: 0.5.

--nbins <value> Number of histogram bins in the output of the transverse momentum distribution.
Default: 400.

--log Enables logarithmic x -axis of the histogram, i.e. logarithmic bins in p_T . The option --ptmin must be set to a non-zero value in order to use this option, otherwise the program will quit with an error.

--ptmin <value> Minimum p_T value [GeV].
Default: 0 GeV.

--ptmax <value> Maximum p_T value [GeV].
Default: 4000 GeV.

--accuracy <value> The desired integration accuracy.
Default: 0.001.

3.2.2 Relevant options for process: H

If process H is selected, i.e. $pp/p\bar{p} \rightarrow H + \text{jet}$, then the following options are relevant:

--mH <value> Higgs mass, m_H [GeV].
Default: 125 GeV.

--mW <value> W boson mass, m_W [GeV].
Default: 80.385 GeV.

--mZ <value> Z boson mass, m_Z [GeV].
Default: 91.1876 GeV.

--mt <value> Top quark mass, m_t [GeV].
Default: 173.5 GeV.

--mb <value> On-shell bottom quark mass, m_b^{OS} [GeV].
Default: 4.65 GeV.

--yt <value> Top Yukawa factor, κ_t [GeV].
Default: 1.

--yb <value> Bottom Yukawa factor, κ_b [GeV].
Default: 1 (0 for CP-odd Higgs).

--GF <value> Fermi coupling constant, G_F [GeV⁻²].
Default: 0.116638×10^{-4} GeV⁻².

Note that the Yukawa couplings are given by $y_q = \kappa_q(\sqrt{2}m_q/v)$, where κ_q are the dimensionless factors specified by the options --yt and --yb above, and $v/\sqrt{2}$ is the vacuum expectation value of the Higgs field.

Note also that H1JET uses the G_μ scheme for the all electroweak parameters [23]. Hence, the Higgs vacuum expectation value is given by $v = 1/\sqrt{\sqrt{2}G_F}$, and the Weinberg angle is $\sin^2 \theta_W = 1 - m_W^2/m_Z^2$.

To consider a CP-odd Higgs instead, it is necessary to select the following option:

--cpodd Toggle for calculation of CP-odd Higgs.

The interaction between the CP-odd Higgs H and a SM quark q is:

$$\mathcal{L} \supset i\tilde{\kappa}_q \frac{m_q}{v} \bar{q} \gamma_5 q H, \quad (13)$$

where the implementation in H1JET uses by default $\tilde{\kappa}_t = 1$ and $\tilde{\kappa}_b = 0$. Both parameters can be changed with the options --yt and --yb.

Here, both CP-even and CP-odd Higgs production are loop-induced processes. The amplitudes for $2 \rightarrow 1$ processes are taken from Ref. [24]. For CP-even Higgs production in

$2 \rightarrow 2$, the amplitudes are taken from Ref. [25], and their interface is adapted from HERWIG 6 [26]. We have taken the CP-odd $2 \rightarrow 2$ amplitudes from Ref. [9].

Top-partner. H1JET allows the calculation of Higgs production via loops of top partners in addition to top loops. To include a top-partner T in the quark loops, it is necessary to set the top-partner mass m_T to a non-zero value by using the `--mtp` option.

The SM top Yukawa factor can be modified by the mixing angle,

$$\kappa_t \rightarrow \kappa_t \cos^2 \theta_T, \tag{14}$$

where κ_t is the Yukawa factor set by option `--ytp`.

The top-partner Yukawa factor will likewise be modified

$$\kappa_T \rightarrow \kappa_T \sin^2 \theta_T. \tag{15}$$

with κ_T set by `--ytp`.

The top-partner specific options are:

- `--mtp <value>` Top-partner mass, m_T [GeV].
Default: 0 GeV.
- `--ytp <value>` Top-partner Yukawa factor, κ_T .
Default: 1.
- `--sth2 <value>` Top-partner mixing angle, $\sin^2 \theta_T$.
Default: 0.

The above is for a simplified composite Higgs model, where the compositeness scale f is set to infinity. The top-partner can also be considered in the explicit composite Higgs models of Ref. [27], all with finite f . Four different models are implemented, **M1₅**, **M1₁₄**, **M4₅**, and **M4₁₄**, which modify the Yukawa coupling factors in the following way:

M1₅:

$$\begin{aligned} \kappa_b &\rightarrow \kappa_b c_\epsilon \\ \kappa_t &\rightarrow \kappa_t c_\epsilon \cos^2 \theta_L \\ \kappa_T &\rightarrow \kappa_T c_\epsilon \sin^2 \theta_L \\ \tilde{\kappa}_b &= \tilde{\kappa}_t = \tilde{\kappa}_T = 0 \end{aligned}$$

M1₁₄:

$$\begin{aligned} \kappa_b &\rightarrow \kappa_b \frac{2c_\epsilon^2 - 1}{c_\epsilon} \\ \kappa_t &\rightarrow \kappa_t \frac{2c_\epsilon^2 - 1}{c_\epsilon} \cos^2 \theta_L \\ \kappa_T &\rightarrow \kappa_T \frac{2c_\epsilon^2 - 1}{c_\epsilon} \sin^2 \theta_L \\ \tilde{\kappa}_b &= \tilde{\kappa}_t = \tilde{\kappa}_T = 0 \end{aligned}$$

M4₅:

$$\begin{aligned} \kappa_b &\rightarrow \kappa_b c_\epsilon \\ \kappa_t &\rightarrow \kappa_t c_\epsilon \left(\cos^2 \theta_R - \frac{s_\epsilon^2}{1+c_\epsilon^2} (\cos^2 \theta_L - \cos^2 \theta_R) \right) \\ \kappa_T &\rightarrow \kappa_T c_\epsilon \left(\sin^2 \theta_R - \frac{s_\epsilon^2}{1+c_\epsilon^2} (\sin^2 \theta_L - \sin^2 \theta_R) \right) \\ \tilde{\kappa}_b &= 0 \\ \tilde{\kappa}_t &= \frac{4c_\epsilon s_\epsilon}{\sqrt{2(1+c_\epsilon^2)}} \text{Im}(c_1) \sin \theta_R \cos \theta_R \\ \tilde{\kappa}_T &= -\tilde{\kappa}_t \end{aligned}$$

M4₁₄:

$$\begin{aligned} \kappa_b &\rightarrow \kappa_b \frac{2c_\epsilon^2 - 1}{c_\epsilon} \\ \kappa_t &\rightarrow \kappa_t \left(\frac{(2c_\epsilon^2 - 1)}{c_\epsilon} \cos^2 \theta_R - \frac{s_\epsilon^2 c_\epsilon (8c_\epsilon^2 - 3)}{1 - 3c_\epsilon^2 + 4c_\epsilon^4} (\cos^2 \theta_L - \cos^2 \theta_R) \right) \\ \kappa_T &\rightarrow \kappa_T \left(\frac{(2c_\epsilon^2 - 1)}{c_\epsilon} \sin^2 \theta_R - \frac{s_\epsilon^2 c_\epsilon (8c_\epsilon^2 - 3)}{1 - 3c_\epsilon^2 + 4c_\epsilon^4} (\sin^2 \theta_L - \sin^2 \theta_R) \right) \\ \tilde{\kappa}_b &= 0 \\ \tilde{\kappa}_t &= \frac{4s_\epsilon(1-2s_\epsilon^2)}{\sqrt{2(1-3c_\epsilon^2+4c_\epsilon^4)}} \text{Im}(c_1) \sin \theta_R \cos \theta_R \\ \tilde{\kappa}_T &= -\tilde{\kappa}_t \end{aligned}$$

where the $\tilde{\kappa}$'s are the CP-odd couplings, and,

$$s_\epsilon = \frac{v}{f}, \quad \text{and} \quad c_\epsilon = \sqrt{1 - s_\epsilon^2}. \tag{16}$$

For **M1₅** and **M1₁₄**, the option `--sth2` sets the mixing angle θ_L , while for **M4₅** and **M4₁₄**, the same option sets the angle θ_R . The reason for this is that we want to reproduce the $f \rightarrow \infty$ limit, where $\theta_T = \theta_L, \theta_R$ depending on the chosen model. When needed, the angles θ_L and θ_R are derived one from the other by using the relation

$$m_t^2 \tan^2 \theta_L = m_T^2 \tan^2 \theta_R. \tag{17}$$

The composite Higgs top-partner model specific options are:

- `--model <arg>` Specify the top-partner model.
Arguments:
 - `M1_5` **M1₅**, with a light top-partner transforming as a **1_{2/3}** of SO(4), and the SM top-bottom doublet embedded in a **5** of SO(5) (default).
 - `M1_14` **M1₁₄**, with a light top-partner transforming as a **1_{2/3}** of SO(4), and the SM top-bottom doublet embedded in a **14** of SO(5).
 - `M4_5` **M4₅**, with a light top-partner transforming as a **4_{2/3}** of SO(4), and the SM top-bottom doublet embedded in a **5** of SO(5).
 - `M4_14` **M4₅**, with a light top-partner transforming as a **4_{2/3}** of SO(4),

and the SM top-bottom doublet embedded in a **14** of SO(5).

--imc1 <value> Imaginary part of the c_1 coefficient, $\text{Im}(c_1)$.
Default: 0.

-f, --fscale <value> Compositeness scale, f [GeV].
If the option is not set, all couplings will be automatically computed in the limit $f \rightarrow \infty$. If the option is set but no value is provided, the program will return a floating point exception.

Multiple top-partners models. H1JET makes it possible to include multiple top-partners in the particle loops. To do that, it will be necessary to specify an input file with the masses and Yukawa coupling factors for each particle running in the loop, including SM quarks. This can be done with the following option:

-i, --in <file> Include input file with top-partner masses and Yukawas. See the file SM.dat for the SM case, without any top partners.

The first line of the input file should specify the number of particles running in the loops, e.g.:

nmax: 2

This should be followed by nmax number of lines – one for each particle loop – in the format of four numbers specifying the mass, κ , $\tilde{\kappa}$, and loop approximation (see later), in that order. For example, for a SM top quark:

173.5 1.0 0.0 1

with mass 173.5 GeV, $\kappa = 1$, $\tilde{\kappa} = 0$, and the loop approximation set to 1.

The dimensionless Yukawa coupling factors κ_q and $\tilde{\kappa}_q$ are respectively the CP-even and CP-odd couplings for a quark q , with the following Lagrangian:

$$\mathcal{L} \supset -m_q \bar{q}q - \kappa_q \frac{m_q}{v} \bar{q}qH + i\tilde{\kappa}_q \frac{m_q}{v} \bar{q}\gamma_5 qH, \quad (18)$$

where m_q is the mass of the quark.

The integer value specifying the loop approximations can take the following values:

- 0 Small mass limit for fermions.

- 1 Full mass effects for fermions.
- 2 Large mass limit for fermions.
- 3 Full mass effects for scalars.
- 4 Large mass limit for scalars.

See Sect. 3.2.6 for more information on the loop approximations. Note that, for implemented processes, using an input file is the only way to change the approximation in which loops are computed.

SUSY_H1JET includes the simplified SUSY model with two stops \tilde{t}_1 and \tilde{t}_2 considered in Refs. [28,29]. To include the SUSY stops \tilde{t}_1 and \tilde{t}_2 in the quark loops, it will be necessary to set the first stop mass $m_{\tilde{t}_1}$ to a non-zero value by using the --mst option. The second stop mass is then given by

$$m_{\tilde{t}_2} = \sqrt{m_{\tilde{t}_1}^2 + (\Delta m)^2}, \quad (19)$$

where Δm is set with the --delta option.

The stop Yukawa coupling factors will be given by:

$$\kappa_{\tilde{t}_1} = \frac{m_{\tilde{t}_1}^2}{m_{\tilde{t}_1}^2} \left[\alpha_1 \cos^2 \theta_{\tilde{t}} + \alpha_2 \sin^2 \theta_{\tilde{t}} + 2 - \frac{(\Delta m)^2}{2m_{\tilde{t}_1}^2} \sin^2 (2\theta_{\tilde{t}}) \right], \quad (20)$$

$$\kappa_{\tilde{t}_2} = \frac{m_{\tilde{t}_2}^2}{m_{\tilde{t}_2}^2} \left[\alpha_1 \sin^2 \theta_{\tilde{t}} + \alpha_2 \cos^2 \theta_{\tilde{t}} + 2 + \frac{(\Delta m)^2}{2m_{\tilde{t}_2}^2} \sin^2 (2\theta_{\tilde{t}}) \right], \quad (21)$$

where

$$\alpha_1 = \frac{m_Z^2}{m_{\tilde{t}}^2} \cos (2\beta) \left[1 - \frac{4}{3} \sin^2 \theta_W \right], \quad (22)$$

$$\alpha_2 = \frac{4}{3} \frac{m_Z^2}{m_{\tilde{t}}^2} \cos (2\beta) \sin^2 \theta_W. \quad (23)$$

Note that $m_{\tilde{t}}$, m_Z , and $\sin^2 \theta_W$ can be set with the --mt, --mZ, and --sinwsq options respectively, while $\sin^2 \theta_{\tilde{t}}$ and $\tan \beta$ are SUSY specific options and can be set with the --sth2 and --tbeta options.

All of the SUSY specific options are:

- mst <value> SUSY stop mass, $m_{\tilde{t}_1}$ [GeV].
Default: 0 GeV.
- delta <value> SUSY stop mass separation, Δm [GeV].
Default: 0 GeV.
- sth2 <value> Stop mixing angle, $\sin^2 \theta_{\tilde{t}}$.
Default: 0.
- tbeta <value> Ratio of VEVs of the two SUSY Higgs fields, $\tan \beta$.
Default: 0.

Note that the top partner mass m_T and SUSY stop mass $m_{\tilde{t}_1}$ can not both be set non-zero at the same time via command-line options. However, if one uses an input file, one can

explicitly specify masses, couplings and loop approximations for an arbitrary number of fermions and scalars. This would also allow a user to implement a specific SUSY model with more supersymmetric partners, each with the appropriate coupling.

3.2.3 Relevant options for process: bbH

If process bbH is selected, i.e. $b\bar{b} \rightarrow H + \text{jet}$, then the following options are relevant:

```
--mH <value>   Higgs mass,  $m_H$  [GeV].
                Default: 125 GeV.
--GF <value>   Fermi coupling constant,  $G_F$ 
                [GeV-2].
                Default:  $0.116638 \times 10^{-4}$  GeV-2.
--mbmb <value>  $\overline{MS}$  bottom quark mass,  $m_b^{\overline{MS}}$ 
                [GeV].
                Default: 4.18 GeV.
```

3.2.4 Relevant options for process: Z

If process Z is selected, i.e. $pp/p\bar{p} \rightarrow Z + \text{jet}$, then the following options are relevant:

```
-mZ <value>   Z boson mass,  $m_Z$  [GeV].
                Default: 91.1876 GeV.
-mW <value>   W boson mass,  $m_W$  [GeV].
                Default: 80.385 GeV.
-GF <value>   Fermi coupling constant,  $G_F$  [GeV-2].
                Default:  $0.116638 \times 10^{-4}$  GeV-2.
```

3.2.5 Relevant options for process: *user*

If process *user* is selected, i.e. a custom user-specified process, any of the above physics options may be relevant if they are used in the custom amplitude code. The code will have to be inspected to determine this. The only built-in process-relevant option is:

```
-M, --mass <value> Relevant mass in the user
                    specified process,  $M$  [GeV].
                    Used in the scale choice and
                    in the setup of kinematics.
                    Default: 0 GeV.
```

Additional options may be added depending on the custom process/amplitude. See Sect. 5 below for more details on the implementation of a custom process.

3.2.6 Loop approximations

Small and large mass limits can be used as approximations for the quarks in the loop calculations. This requires some knowledge of the meaning of the approximations, hence this needs to be set by the user at compile time.

In the file `input.f90` located in the `src`-directory, the subroutine `reset_iloop_array` can be found. This subroutine can be used by the user to set the `iloop_array`, which is an array specifying the approximation used for each loop particle. The approximations that can be used are:

```
iloop_sm_fermion Small mass limit for fermions.
iloop_fm_fermion Full mass effects for fermions.
iloop_lm_fermion Large mass limit for fermions.
iloop_fm_scalar Full mass effects for scalars.
iloop_lm_scalar Large mass limit for scalars.
```

The size of the array must match the number of particles appearing in the loops, which should be checked by the user. Below is an example Fortran code snippet for the `reset_iloop_array` subroutine, which sets the loop approximations for an effective theory with SM top and bottom quarks, and one infinitely heavy top-partner:

```
if (size(iloop_array) == 3) then
    iloop_array = (/ iloop_fm_fermion, &
                   & iloop_fm_fermion, &
                   & iloop_lm_fermion /)
else
    call wae_error('reset_iloop_array', &
                  & 'Expected size of iloop_array is
                    3, &
                    & whereas actual one is', &
                  & intval = size(iloop_array))
end if
```

3.2.7 Output

The helper script `PlotH1jet.py` facilitates easy and quick plotting of the output from H1JET.

The script requires Python 3 installed in order to run.

The user needs to simply pipe the output of H1JET to the script:

```
./bin/h1jet [options] | python PlotH1jet.py
```

Alternatively, the plotting script can run on an output file from H1JET:

```
./bin/h1jet [options] -o result.out
python PlotH1jet.py result.out
```

A resulting example plot with default settings in H1JET is shown in Fig. 1.

A comparison between the various built-in models is shown in Fig. 2. Default SM parameters has been used with $m_{\tilde{t}_1} = 600$ GeV, $\Delta m = 200$ GeV, $\tan \beta = 5$, $m_T = 1.7$ TeV,

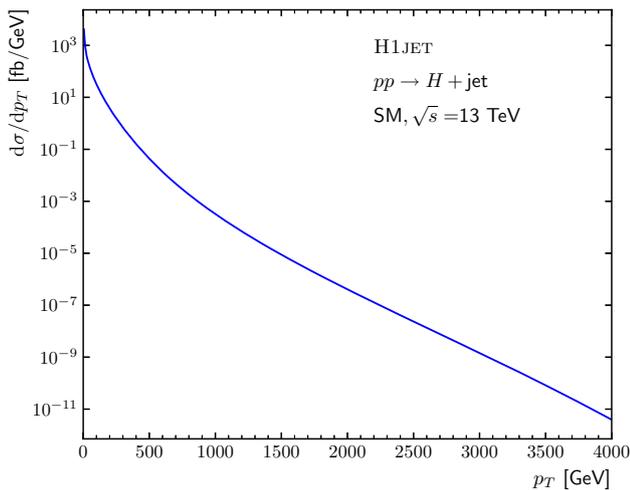


Fig. 1 The p_T distribution for the SM process $pp \rightarrow H + \text{jet}$ from H1JET with default settings

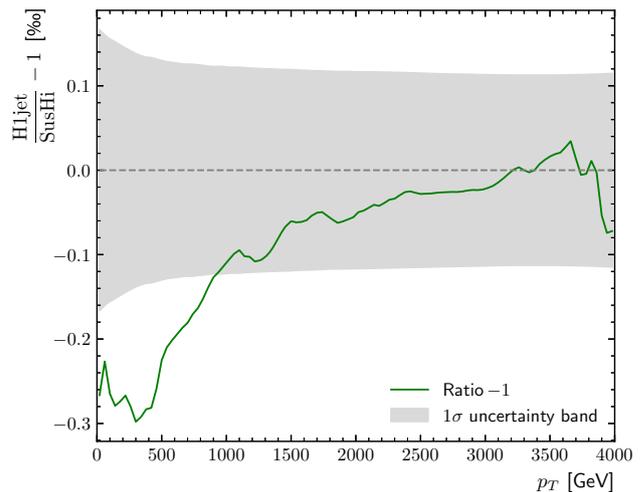


Fig. 3 The relative difference in permille between the H1JET result and that of SUSHI for the p_T distribution for a CP-odd Higgs. The grey band indicates the Monte Carlo error of SUSHI

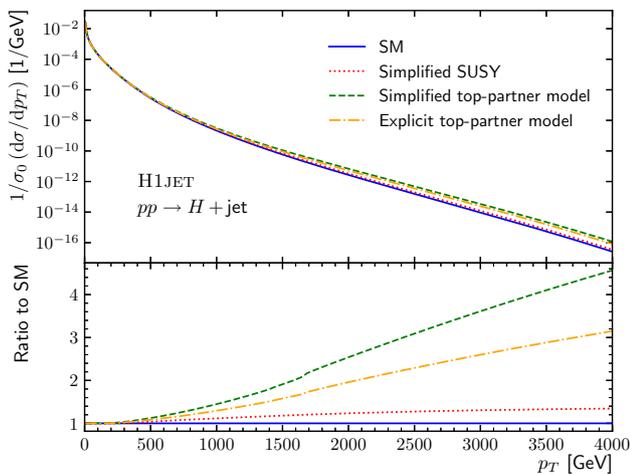


Fig. 2 Comparison between the CP-even Higgs p_T distributions for various built-in models

$\sin^2 \theta_{\tilde{t}} = \sin^2 \theta_T = 0.1$, and $f = 900$ GeV, and considering the **M4₅** model as the explicit top-partner model.

4 Benchmarking

The various processes implemented in H1JET have been compared to those of SUSHI [17, 18], and have all been found to be in agreement. The relative ratio between the H1JET result and the SUSHI result for the p_T distribution for the CP-odd Higgs are shown in Fig. 3, and is found to be in agreement within the Monte Carlo error of SUSHI for a large range of p_T values. Overall the agreement with SUSHI is within 3×10^{-4} . Note that the largest discrepancies were observed in the low p_T region. To validate the H1JET results we have compared

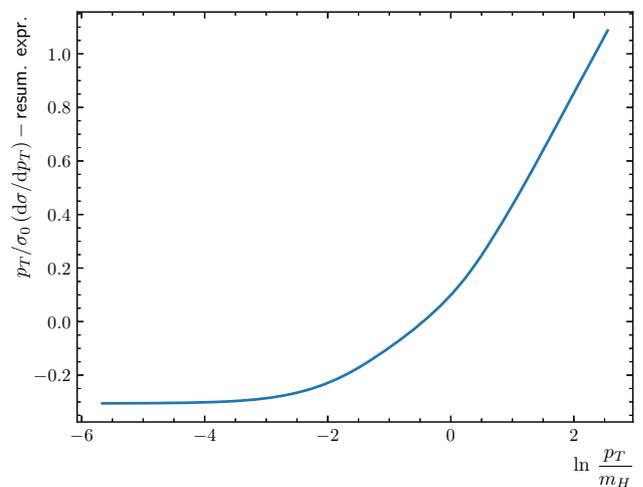


Fig. 4 The CP-odd distribution $\frac{p_T}{\sigma_0} \left(\frac{d\sigma}{dp_T} \right)$ with the first term of the p_T resummation subtracted, as a function of $\ln \frac{p_T}{m_H}$. For low p_T values it converges to a constant value

them to the approximate expression valid at low p_T

$$\frac{d\sigma}{dp_T} \xrightarrow{p_T \rightarrow 0} \sigma_0 \left[4 C_A \frac{\alpha_s}{\pi} \frac{1}{p_T} \left(\ln \frac{m_H}{p_T} + \mathcal{O}(1) \right) \right], \quad (24)$$

where σ_0 is the total Born cross section for $gg \rightarrow H$. In Fig. 4, we show $\frac{p_T}{\sigma_0} \left(\frac{d\sigma}{dp_T} \right)$ with the first term of Eq. (24) subtracted, as a function of $\ln \frac{p_T}{m_H}$. For $p_T \rightarrow 0$ this goes nicely towards a constant as expected.

The relative ratio between the H1JET and SUSHI results for the SUSY are shown in Fig. 5 and is within 2×10^{-4} . Again the low p_T behaviour can be checked by comparing to the resummed expression in Fig. 6.

Note that our numerical accuracy crucially depends not only on the accuracy of the numerical integration, but of that

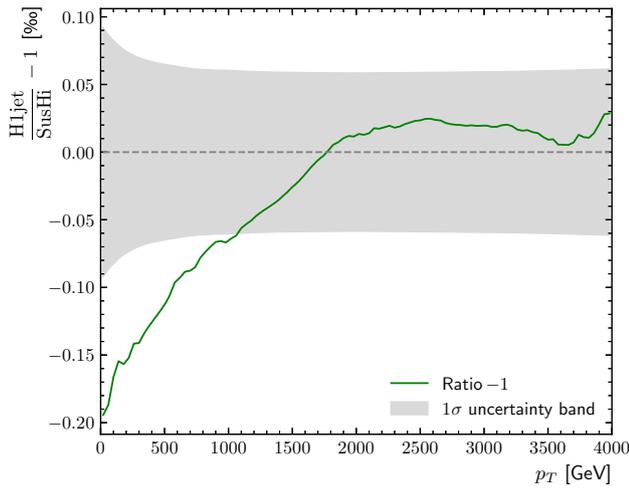


Fig. 5 The relative difference in permille between the H1JET result and that of SUSHi for the p_T distribution in SUSY. The grey band indicates the Monte Carlo error of SUSHi

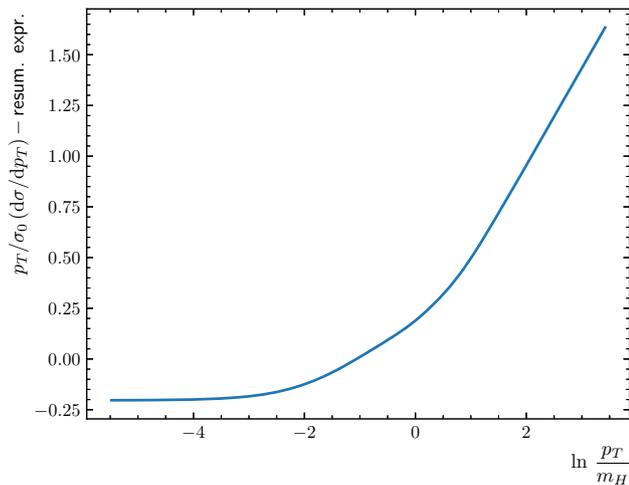


Fig. 6 The SUSY distribution $\frac{p_T}{\sigma_0} \left(\frac{d\sigma}{dp_T} \right)$ with the first term of the p_T resummation subtracted, as a function of $\ln \frac{p_T}{m_H}$. For low p_T values it converges to a constant value

of the auxiliary programs used to compute the PDF evolution (HOPPET) and the scalar integrals (CHAPLIN). We have modified various internal parameters of the two libraries, and we obtained differences that are less than permille level. So, a conservative estimate of the numerical uncertainty of H1JET is 1×10^{-3} .

5 Adding new processes to H1JET

H1JET can be interfaced to use the squared matrix element evaluated from a custom Fortran code. The implementation may be most easily explained with a specific example. This

section should be read very carefully before attempting to use the interface.

5.1 Example: axion-like-particle (ALP) effective theory

We will present here a specific example of adding to H1JET the production of a light axion-like-particle (ALP), a , along with a jet. For simplicity, we only consider the gluon-fusion channel,

$$gg \rightarrow ga. \quad (25)$$

This is a tree-level process due to an effective ALP-gluon coupling,

$$\delta\mathcal{L}_a \supset -c_{\tilde{G}} \frac{a}{f_a} G_{\mu\nu}^a \tilde{G}^{a\mu\nu}, \quad (26)$$

where $G_{\mu\nu}^a$ is the gluon field strength tensor and $\tilde{G}_{\mu\nu}^a = (1/2)\epsilon_{\mu\nu\rho\sigma} G^{a\rho\sigma}$ its dual. The model and the FEYNRULES [30] model files are described and provided in Ref. [31]. We use FEYNCALC [32–34] to evaluate the amplitude from the model, so we have to convert the FEYNRULES model to a FEYNARTS [35] model in Mathematica:

```
In[1]:= << FeynRules`

In[2]:= LoadModel["SM.fr", "alp_linear.fr",
  "alp_linear_operators.fr"];

In[3]:= WriteFeynArtsOutput[LSM + LALP,
  CouplingRename -> False];
```

The resulting FEYNARTS model files are written to a new directory ALP_linear_FA, which needs to be moved to the FEYNARTS/MODELS directory. Note that in the FEYNARTS model, the ALP field is called S[4] and the gluon fields are called V[4].

In a new Mathematica session, we load FEYNCALC with FEYNARTS:

```
In[4]:= $LoadAddOns = {"FeynArts"};

In[5]:= << FeynCalc`
```

The FeynArts/Models directory can be located with:

```
In[6]:= $FeynArtsDir
```

First, we patch the ALP_linear_FA model with:

```
In[7]:= FAPatch[PatchModelsOnly -> True]
```

This ensures that the model files works with FEYNCALC.

Then we create the tree-level 2 \rightarrow 2 topologies and insert the relevant fields for our process:

```
In[8]:= tops = CreateTopologies[0, 2 -> 2];

In[9]:= ins = InsertFields[tops, {V[4], V[4]} ->
  {V[4], S[4]}, InsertionLevel ->
  {Classes}, Model -> "ALP_linear_FA",
  GenericModel -> "ALP_linear_FA";
```

It is possible to draw the Feynman diagrams for the process as a check:

```
In[10]:= Paint[ins, ColumnsXRows -> {2, 1},
  Numbering -> Simple, SheetHeader ->
  None,
  ImageSize -> {512, 256}];
```

We then set up the amplitude:

```
In[11]:= feynamp = CreateFeynAmp[ins];

In[12]:= amp = FCFAConvert[feynamp,
  IncomingMomenta -> {k1, k2},
  OutgoingMomenta -> {k3, k4},
  UndoChiralSplittings -> True,
  ChangeDimension -> 4,
  TransversePolarizationVectors -> {k1,
  k2, k3},
  List -> False,
  SMP -> True, Contract -> True,
  DropSumOver -> True]
```

While not strictly necessary, it is recommended to enable the SMP option. Any additional substitutions in the amplitude can be specified with the FinalSubstitutions option.

We then set up the kinematics:

```
In[13]:= FCClearScalarProducts[];

In[14]:= SetMandelstam[s, t, u, k1, k2, -k3, -k4,
  0, 0, 0, mA];
```

We introduce here a parameter mA for the ALP mass m_a .

We then square the amplitude:

```
In[15]:= ampsquared = Simplify[
  (TrickMandelstam[#1, {s, t, u, mA^2}] &
  ) [
  (DoPolarizationSums[#1, k2, k1,
  ExtraFactor -> 1/2] & ) [
```

```
(DoPolarizationSums[#1, k1, k2,
  ExtraFactor -> 1/2] & ) [
  (DoPolarizationSums[#1, k3, 0] & ) [
  (SUNSimplify[#1, Explicit -> True,
  SUNNToCACF -> False] & ) [
  FeynAmpDenominatorExplicit[(1 /
  (SUNN^2 - 1)^2) * (amp *
  ComplexConjugate[amp])]]]]]]] /.
  SUNN -> 3
```

Setting the SUNNToCACF option in SUNSimplify[] to False is not necessary, nor is it necessary to fix SUNN to 3. This can be handled by the dressing script and H1JET.

Finally, we write the amplitude as Fortran code to a file:

```
In[16]:= Write2["ALP_amp.f90", gg = ampsquared,
  FormatType -> FortranForm,
  FortranFormatDoublePrecision -> False]
```

Note here that we specify the gluon-gluon channel with the gg = ampsquared input to the function. This is required for the subsequent dressing script to work properly. It is important to specify the 2-particle initial state by using combinations of g, u, d, c, s, b, ubar, dbar, cbar, sbar, and bbar. One can also use q and qbar for all the light quarks and antiquarks respectively, i.e. u, d, c, and s. For example, bbbbar will be the $b\bar{b}$ channel.

The generated Fortran code ALP_amp.f90 has the following content:

```
gg = (3.*CGtil**2*(mA**8 - 2.*mA**6*(t + u) +
  & 3.*mA**4*(t + u)**2 - 2.*mA**2*(t +
  u)**3 +
  & (t**2 + t*u +
  u**2)**2)*SMP("g_s")**2)/
  & (fa**2*s*t*u)
```

This code has to be dressed by the Python helper script DressUserAmpCode.py:

```
python DressUserAmpCode.py ALP_amp.f90
```

This produces a dressed Fortran code file called by default user_interface.f90.

The helper script provides a help message which can be called with -h or --help. The name of the output file can be specified with the -o option. Multiple input Fortran files can be given as arguments to the helper script. The full usage is:

```
python DressFeynCalcCode.py [-h] [-o
  [OUTFILE]] inputfile [inputfile ...]
```

The provided input Fortran code files does not necessarily have to be generated with FEYNCALC. They can be generated by any other program or even be written by hand by providing the appropriate expression for gg or the desired channel.

To use the new dressed custom Fortran code with H1JET, it is necessary to recompile H1JET with the custom Fortran code:

```
./configure
USERFILE=/path/to/custom/user_interface.f90
make clean
make
```

Running `./h1jet --help` we see that three new additional options have been added:

```
--c_CGtil <value> The Wilson coefficient  $c_{\tilde{G}}$  in
                    Eq. (26).
--c_mA <value>     The ALP mass,  $m_a$ .
--c_fa <value>     The ALP suppression scale  $f_a$ 
                    in
                    Eq. (26).
```

The leading `c_` in the name stands for “custom” and is automatically added in order to avoid naming issues in the code.

The result from the ALP implementation in H1JET is shown in Fig. 7 and can be compared to the H1JET result for the CP-odd Higgs by using a single top quark in the loop with an infinite mass limit, resulting in an effective coupling between the CP-odd Higgs and the gluons. In fact, the respective ALP and CP-odd couplings are then related as such,

$$\frac{c_{\tilde{G}}^2}{f_a^2} = \frac{\alpha_s^2}{64\pi^2 v^2}. \quad (27)$$

The comparison is shown in Fig. 8, where we see agreement within 4×10^{-6} .

The result can also be compared to the same FEYNRULES model used with MADGRAPH5_AMC@NLO [16], where our code takes one second to run, while MADGRAPH can take up to several hours depending on the number of events, due to MADGRAPH running a full Monte Carlo integration. We have found that H1JET agrees with MADGRAPH5_AMC@NLO within Monte Carlo errors. We have also seen that MADGRAPH5_AMC@NLO runs into numerical instabilities at low p_T , while H1JET has by construction the correct behaviour.

5.1.1 The total cross section

While not strictly necessary for the user interface to run, it is still recommended to add the code for the evaluation of the total cross section to the custom user interface. This is easy to do as well. We will here show it for the ALP model.

We start with considering the gluon-fusion ALP production, $gg \rightarrow a$. In Mathematica, create a tree-level $2 \rightarrow 1$ topology, and insert the fields:

```
In[17]:=
  tops = CreateTopologies[0, 2 -> 1];
```

```
In[18]:=
```

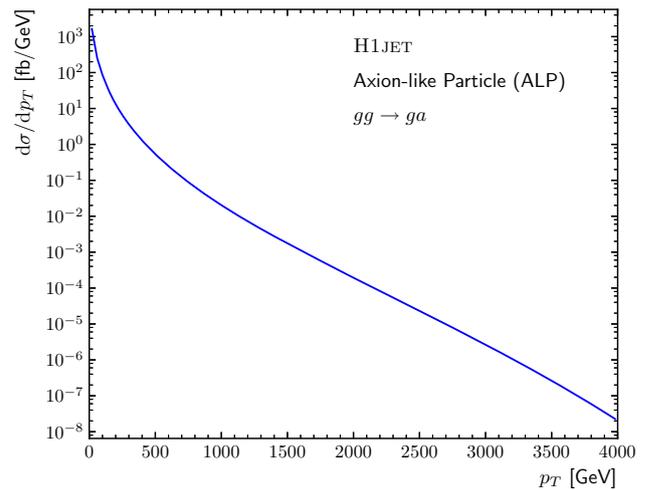


Fig. 7 The p_T distribution for the process $gg \rightarrow ga$ from H1JET with the amplitude from the user interface

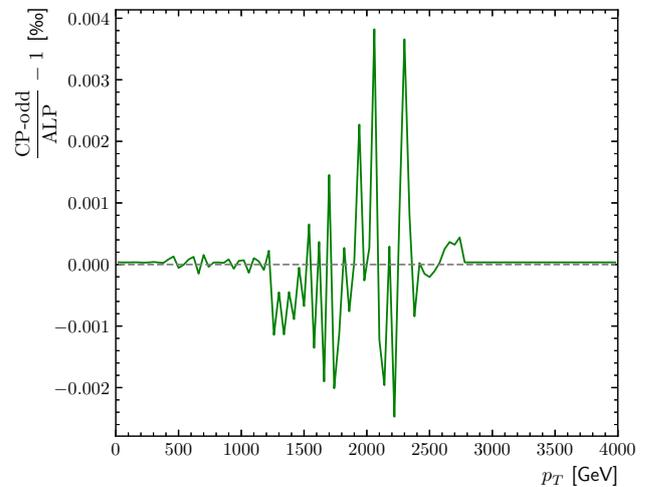


Fig. 8 The relative ratio in permille between the CP-odd result and the result with the amplitude from the user interface

```
ins = InsertFields[tops, {V[4],
  V[4]} -> {S[4]},
  InsertionLevel -> {Classes},
  Model -> "ALP_linear_FA",
  GenericModel -> "ALP_linear_FA";
```

Then we set up the amplitude:

```
In[19]:=
  feynamp = CreateFeynAmp[ins];
```

```
In[20]:=
  amp = FCFAConvert[feynamp,
  IncomingMomenta -> {k1, k2},
  OutgoingMomenta -> {k3},
  UndoChiralSplittings -> True,
  ChangeDimension -> 4,
```

```

TransversePolarizationVectors -> {k1,
k2},
List -> False, SMP -> True,
Contract -> True, DropSumOver -> True]

```

as well as the kinematics:

```

In[21]:=
FCClearScalarProducts[];

```

```

In[22]:=
SP[k1, k1] = 0;

```

```

In[23]:=
SP[k2, k2] = 0;

```

```

In[24]:=
SP[k3, k3] = mA^2;

```

```

In[25]:=
SP[k1, k2] = mA^2 / 2;

```

We then square the amplitude:

```

In[26]:=
ampsquared = Simplify[
(DoPolarizationSums[#1, k2, k1,
ExtraFactor -> 1/2] & ) [
(DoPolarizationSums[#1, k1, k2,
ExtraFactor -> 1/2] & ) [
(SUNSSimplify[#1, Explicit -> True,
SUNNToCACF -> False] & ) [
FeynAmpDenominatorExplicit[(1 /
(SUNN^2 - 1)^2) * (amp *
ComplexConjugate[amp])]]]]]]

```

For a $2 \rightarrow 1$ process, the hadronic cross section is given in Eq. (4), where the partonic luminosity \mathcal{L}_{gg} is handled by H1JET. Hence, we need to multiply our squared matrix element by

```

In[27]:=
xsec = Pi * ampsquared / mA^4

```

Finally, we can write the cross section as a Fortran code file:

```

In[28]:=
Write2["ALP_xsec.f90", xsgg = xsec,
FormatType -> FortranForm,
FortranFormatDoublePrecision -> False]

```

We again specify the gluon-gluon channel with the `xsgg = xsec`, but this time indicate with the leading `xs` that the code is for the Born cross section. Otherwise, the same rules apply. It is crucial to make sure not to save the Born cross section in the same file as the squared amplitude code for the transverse momentum distribution.

The new generated code `ALP_xsec.f90` is provided to `DressUserAmpCode.py` along with the squared amplitude code:

```

python DressUserAmpCode.py ALP_amp.f90
ALP_xsec.f90

```

And H1JET can be recompiled to include the new source code:

```

./configure
USERFILE=/path/to/custom/user_interface.f90
make clean
make

```

After which H1JET will calculate the Born cross section for the custom process.

6 Conclusions

We have presented a method that allows a fast computation of the transverse momentum distribution of a colour singlet. The method is implemented in the program H1JET, which returns a transverse momentum spectrum for the specified colour singlet in about a second. H1JET is similar in spirit to SUSHi, but is incomparably faster.

The program implements various processes, including Higgs production both in gluon fusion and bottom-antibottom annihilation, as well as Z production. Loop-induced Higgs production is implemented not only in the SM, but also in attractive BSM scenarios, such as SUSY or composite Higgs. For SUSY, we implement a simplified model with two stops, as done in Ref. [29]. For composite Higgs, we implement both the simplified model of Ref. [10], as well as some explicit models with one or more top partners [27]. Loop integrals can be computed either exactly or in the infinite-mass limit. The latter limit implements in practice a dimension-6 contact interaction between the Higgs and the gluon field. The program is very flexible, and the only process-dependent input is the corresponding amplitude in terms of Mandelstam invariants. This can be computed by the user either manually, or with the use of automated programs such as FeynCalc [34], and connected to the program via a simple interface. As an example, we have included in the package the calculation of the transverse momentum distribution of an ALP starting from the general Feynman rules of Ref. [31]. Note that the possibility of automatically implementing a new model inside the program is a feature that is not available in SUSHi. We also stress that it is also possible to take advantage of input files to obtain results for an arbitrary number of fermions and

scalars in loops, with appropriate couplings. This could be used, for instance, to implement the MSSM instead of the provided simplified SUSY model.

We stress that H1JET is not a replacement for a proper Monte Carlo analysis implementing realistic experimental cuts. However, we believe it will be invaluable for BSM experts to assess whether a given model gives sizeable deviations from the SM. In fact, due to its fast implementation, H1JET makes it possible to perform parameter scans in seconds, and to take into account mass effects in specific models. Also, due to the fact that H1JET is not based on a Monte Carlo integration, one can separate interference between different contributions very precisely, something which is very difficult to achieve with Monte Carlo event generators.

H1JET can be also useful to precision phenomenology. In fact, it makes it possible to easily perform theoretical studies of the transverse momentum distribution of a colour singlet, especially those involving the matching of resummed calculations with exact fixed order. Also, although the implemented cross sections are computed at the lowest order in QCD, nothing prevents the inclusion of higher orders, provided one integrates over all coloured particles.

Acknowledgements The idea of having a fast program to compute transverse momentum spectra started from AB’s collaboration with G. Zanderighi, P. F. Monni and F. Caola. AB acknowledges many useful discussions with them on the topic. AB also wishes to thank B. Dillon, W. Ketaiam and S. Kvedaraitė for contributions to a private preliminary version of H1JET. We thank J. M. Lindert for all his helpful remarks and suggestions on this paper. The studentship of AL is supported by the Science Technology and Facilities Council (STFC) under grant number ST/P000819/1. The work of AB is supported by the Science Technology and Facilities Council (STFC) under grants number ST/P000819/1 and ST/T00102X/1.

Data Availability Statement This manuscript has associated data in a data repository. [Authors’ comment: The program code is publicly available at <https://h1jet.hepforge.org/>.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.
Funded by SCOAP³.

A Implementation of scalar integrals

This appendix contains the details of how H1JET computes one-loop scalar integrals that are relevant for Higgs produc-

tion. These integrals depend on one internal mass, which we denote by m , and are functions of Mandelstam invariants.

Scalar integrals can be written in terms of logarithms and dilogarithms of complex arguments, which require appropriate analytic continuations. Instead of performing such manipulations ourselves, we have decided to use the implementation of the library CHAPLIN, and recast all relevant transcendental functions into harmonic polylogarithms $H(\mathbf{a}; z)$, with $\mathbf{a} = (a_1, \dots, a_n)$. For real values of the argument of a polylogarithm, CHAPLIN uses the $+i\varepsilon$ prescription, i.e. $H(\mathbf{a}; z)$ with z real is interpreted as $H(\mathbf{a}; z + i\varepsilon)$. Therefore, we need to make sure that the imaginary part of the argument of scalar integrals is consistent with the convention of CHAPLIN.

The relevant one-loop integrals we need to deal with are bubbles, triangles and boxes.

Bubbles. The bubble integral is defined as

$$B_0(s) = 2 - \sqrt{1 - \frac{4(m^2 - i\varepsilon)}{s}} \ln \left[-\frac{z}{1 - z} \right], \tag{A.1}$$

where

$$z = \frac{1}{2} \left(1 + \sqrt{1 - \frac{4(m^2 - i\varepsilon)}{s}} \right). \tag{A.2}$$

The argument of the logarithm in Eq. (A.1) has a different form according to the value of s :

$$-\frac{z}{1 - z} = \begin{cases} -\frac{1 + \sqrt{1 + \frac{4m^2}{|s|}}}{1 - \sqrt{1 + \frac{4m^2}{|s|}}}, & s < 0, \\ \frac{1 + i\sqrt{\frac{4m^2}{s} - 1}}{1 - i\sqrt{\frac{4m^2}{s} - 1}}, & 0 < s < 4m^2, \\ -\frac{1 - \sqrt{1 - \frac{4m^2}{s}}}{1 - \sqrt{1 - \frac{4m^2}{s}}} - i\varepsilon, & s > 4m^2 \end{cases}, \tag{A.3}$$

Note that the only case in which one needs a small imaginary part is the case $s > 4m^2$. This imaginary part has the opposite convention as in CHAPLIN. As a solution, we invert the argument of the logarithm and use the identity $\ln z = -\ln(1/z)$. In practice, after an appropriate analytic continuation of the square root, we define

$$z \equiv \begin{cases} \frac{1}{2} \left(1 + \sqrt{1 - \frac{4m^2}{s}} \right), & \frac{4m^2}{s} < 1 \\ \frac{1}{2} \left(1 + i\sqrt{\frac{4m^2}{s} - 1} \right), & \frac{4m^2}{s} > 1 \end{cases} \tag{A.4}$$

and implement the bubble as follows:

$$B_0(s) = 2 - (2z - 1) H \left(1; \frac{1}{z} \right). \tag{A.5}$$

Note that a logarithm of a negative number can also be correctly analytically continued by using the default Fortran implementation of the complex logarithm. As for CHAPLIN, Fortran assumes that a negative number has a small positive imaginary part. Therefore, in case we have a small negative imaginary part, we can still use the relation $\ln z = -\ln(1/z)$, which gives the correct analytic continuation.

Triangles. The triangle integral $C_0(s)$ is defined as

$$C_0(s) = \frac{1}{2s} \ln^2 \left[-\frac{z}{1-z} \right], \tag{A.6}$$

where z is given in Eq. (A.2). Again, for $s > 4m^2$, the argument of the logarithm has the opposite sign with respect to what is implicitly assumed by CHAPLIN. Therefore, we invert again the argument of the logarithm, and using the definition of z in Eq. (A.4), we implement the triangle as follows:

$$s C_0(s) = H \left(1, 1; \frac{1}{z} \right). \tag{A.7}$$

Boxes. The scalar four-point function with three massless (the gluons) and one massive (the Higgs boson) external lines is given by [25],

$$D(s, t) = \frac{1}{st} \int_0^1 \frac{dx}{x(1-x) + m^2 u/(ts)} \times \left[-\ln \left(1 - i\varepsilon - \frac{m_H^2}{m^2} x(1-x) \right) + \ln \left(1 - i\varepsilon - \frac{s}{m^2} x(1-x) \right) + \ln \left(1 - i\varepsilon - \frac{t}{m^2} x(1-x) \right) \right], \tag{A.8}$$

which can be expressed in terms of complex dilogarithms by using the exact result

$$\frac{1}{st} \int_0^1 \frac{dx}{x(1-x) + m^2 u/(ts)} \ln \left(1 - i\varepsilon - \frac{v}{m^2} x(1-x) \right) = \frac{2}{\sqrt{1 + 4m^2 u/(ts)}} \times \left[\text{Li}_2 \left(\frac{x_-}{x_- - y} \right) - \text{Li}_2 \left(\frac{x_+}{x_+ - y} \right) + \text{Li}_2 \left(\frac{x_-}{y - x_+} \right) - \text{Li}_2 \left(\frac{x_+}{y - x_-} \right) + \ln \left(\frac{-x_-}{x_+} \right) \ln \left(1 - i\varepsilon - \frac{v}{m^2} x_- x_+ \right) \right], \tag{A.9}$$

where

$$x_{\pm} \equiv \frac{1}{2} \left(1 \pm \sqrt{1 + \frac{4m^2 u}{ts}} \right) \tag{A.10}$$

are real numbers, with $x_+ > 1$ and $x_- < 0$, and

$$y \equiv \frac{1}{2} \left(1 + \sqrt{1 - \frac{4(m^2 - i\varepsilon)}{v}} \right), \tag{A.11}$$

acquires an imaginary part according to the value of v . In particular, keeping track of the imaginary part of y yields

$$y = \begin{cases} \frac{1}{2} \left(1 + \sqrt{1 + \frac{4m^2}{|v|}} \right) - i\varepsilon, & v < 0 \\ \frac{1}{2} \left(1 + i\sqrt{\frac{4m^2}{v} - 1} \right), & 0 < v < 4m^2 \\ \frac{1}{2} \left(1 + \sqrt{1 - \frac{4m^2}{v}} \right) + i\varepsilon, & v > 4m^2 \end{cases} \tag{A.12}$$

From the above, we see that for $0 < v < 4m^2$ we can use the dilogarithms as given by CHAPLIN. For $v < 0$, $x_-/(x_- - y)$ and $x_+/(y - x_-)$ acquire a small positive imaginary part, whereas $x_+/(x_+ - y)$ and $x_-/(y - x_+)$ a small negative imaginary part. The reverse happens for $v > 4m^2$. Therefore, we need to perform some formal manipulations to use the harmonic polylogarithms provided by CHAPLIN.

In practice, whenever the argument z of the dilogarithm is complex, we just use the definitory relation $\text{Li}_2(z) = H(0, 1; z)$. When $z = x + i\varepsilon$, with x real, we use $\text{Li}_2(x + i\varepsilon) = H(0, 1; x)$, with $H(0, 1; x)$ the complex number provided by CHAPLIN. If $z = x - i\varepsilon$, we use the identities

$$\begin{aligned} \text{Li}_2(x - i\varepsilon) &= -H \left(0, 1; \frac{1}{x} \right) - \frac{\pi^2}{6} - H(0, 0; -x) \\ &= -H(0, -1; -x) \\ &= -H(0, 1, 1 - x) + \frac{\pi^2}{6} \\ &\quad + H(1; 1 - x)H(-1; -x), \end{aligned} \tag{A.13}$$

and we select the one that gives the smallest imaginary part. This of course give numerically indistinguishable results when the imaginary part is large, but is of crucial importance when the imaginary part should be zero but it is not because of the specific numerical methods employed by CHAPLIN.

References

1. <https://h1jet.hepforge.org/>
2. G. Aad et al., Phys. Lett. B **716**, 1 (2012). <https://doi.org/10.1016/j.physletb.2012.08.020>
3. S. Chatrchyan et al., Phys. Lett. B **716**, 30 (2012). <https://doi.org/10.1016/j.physletb.2012.08.021>
4. G. Aad et al., Phys. Rev. D **101**(1), 012002 (2020). <https://doi.org/10.1103/PhysRevD.101.012002>
5. A.M. Sirunyan et al., Eur. Phys. J. C **79**(5), 421 (2019). <https://doi.org/10.1140/epjc/s10052-019-6909-y>
6. M. Aaboud et al., Phys. Lett. B **786**, 114 (2018). <https://doi.org/10.1016/j.physletb.2018.09.019>
7. A.M. Sirunyan et al., Phys. Lett. B **792**, 369 (2019). <https://doi.org/10.1016/j.physletb.2019.03.059>

8. A. Azatov, A. Paul, *JHEP* **01**, 014 (2014). [https://doi.org/10.1007/JHEP01\(2014\)014](https://doi.org/10.1007/JHEP01(2014)014)
9. C. Grojean, E. Salvioni, M. Schlaffer, A. Weiler, *JHEP* **05**, 022 (2014). [https://doi.org/10.1007/JHEP05\(2014\)022](https://doi.org/10.1007/JHEP05(2014)022)
10. A. Banfi, A. Martin, V. Sanz, *JHEP* **08**, 053 (2014). [https://doi.org/10.1007/JHEP08\(2014\)053](https://doi.org/10.1007/JHEP08(2014)053)
11. G. Aad et al., *Phys. Rev. Lett.* **125**(6), 061802 (2020). <https://doi.org/10.1103/PhysRevLett.125.061802>
12. A.M. Sirunyan et al., *Phys. Rev. Lett.* **125**(6), 061801 (2020). <https://doi.org/10.1103/PhysRevLett.125.061801>
13. F. Maltoni, E. Vryonidou, C. Zhang, *JHEP* **10**, 123 (2016). [https://doi.org/10.1007/JHEP10\(2016\)123](https://doi.org/10.1007/JHEP10(2016)123)
14. A.M. Sirunyan et al., *JHEP* **07**, 014 (2017). [https://doi.org/10.1007/JHEP07\(2017\)014](https://doi.org/10.1007/JHEP07(2017)014)
15. M. Aaboud et al., *JHEP* **05**, 142 (2019). [https://doi.org/10.1007/JHEP05\(2019\)142](https://doi.org/10.1007/JHEP05(2019)142)
16. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, *JHEP* **07**, 079 (2014). [https://doi.org/10.1007/JHEP07\(2014\)079](https://doi.org/10.1007/JHEP07(2014)079). <https://launchpad.net/mg5amcnlo>
17. R.V. Harlander, S. Liebler, H. Mantler, *Comput. Phys. Commun.* **184**, 1605 (2013). <https://doi.org/10.1016/j.cpc.2013.02.006>
18. R.V. Harlander, S. Liebler, H. Mantler, *Comput. Phys. Commun.* **212**, 239 (2017). <https://doi.org/10.1016/j.cpc.2016.10.015>. <https://sushi.hepforge.org>
19. G.P. Salam, J. Rojo, *Comput. Phys. Commun.* **180**, 120 (2009). <https://doi.org/10.1016/j.cpc.2008.08.010>. <https://hoppet.hepforge.org/>
20. A. Banfi, F. Caola, F.A. Dreyer, P.F. Monni, G.P. Salam, G. Zanderighi, F. Dulat, *JHEP* **04**, 049 (2016). [https://doi.org/10.1007/JHEP04\(2016\)049](https://doi.org/10.1007/JHEP04(2016)049). <https://jetvheto.hepforge.org/>
21. A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, G. Watt, *Eur. Phys. J. C* **75**, 132 (2015). <https://doi.org/10.1140/epjc/s10052-015-3318-8>. <https://lhpdf.hepforge.org/>
22. S. Buehler, C. Duhr, *Comput. Phys. Commun.* **185**, 2703 (2014). <https://doi.org/10.1016/j.cpc.2014.05.022>. <https://chaplin.hepforge.org/>
23. H. Georgi, *Nucl. Phys. B* **363**, 301 (1991). [https://doi.org/10.1016/0550-3213\(91\)80022-E](https://doi.org/10.1016/0550-3213(91)80022-E)
24. M. Spira, A. Djouadi, D. Graudenz, P. Zerwas, *Nucl. Phys. B* **453**, 17 (1995). [https://doi.org/10.1016/0550-3213\(95\)00379-7](https://doi.org/10.1016/0550-3213(95)00379-7)
25. U. Baur, E. Glover, *Nucl. Phys. B* **339**, 38 (1990). [https://doi.org/10.1016/0550-3213\(90\)90532-I](https://doi.org/10.1016/0550-3213(90)90532-I)
26. G. Corcella, I. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M. Seymour, B. Webber, *JHEP* **01**, 010 (2001). <https://doi.org/10.1088/1126-6708/2001/01/010>
27. A. Banfi, B.M. Dillon, W. Ketaiam, S. Kvedaraitė, *JHEP* **01**, 089 (2020). [https://doi.org/10.1007/JHEP01\(2020\)089](https://doi.org/10.1007/JHEP01(2020)089)
28. J.F. Guion, H.E. Haber, *Phys. Rev. D* **67**, 075019 (2003). <https://doi.org/10.1103/PhysRevD.67.075019>
29. A. Banfi, A. Bond, A. Martin, V. Sanz, *JHEP* **11**, 171 (2018). [https://doi.org/10.1007/JHEP11\(2018\)171](https://doi.org/10.1007/JHEP11(2018)171)
30. A. Alloul, N.D. Christensen, C. Degrande, C. Duhr, B. Fuks, *Comput. Phys. Commun.* **185**, 2250 (2014). <https://doi.org/10.1016/j.cpc.2014.04.012>. <https://feynrules.irmp.ucl.ac.be>
31. I. Brivio, M. Gavela, L. Merlo, K. Mimasu, J. No, R. del Rey, V. Sanz, *Eur. Phys. J. C* **77**(8), 572 (2017). <https://doi.org/10.1140/epjc/s10052-017-5111-3>. <https://feynrules.irmp.ucl.ac.be/wiki/ALPsEFT>
32. R. Mertig, M. Bohm, A. Denner, *Comput. Phys. Commun.* **64**, 345 (1991). [https://doi.org/10.1016/0010-4655\(91\)90130-D](https://doi.org/10.1016/0010-4655(91)90130-D)
33. V. Shtabovenko, R. Mertig, F. Orellana, *Comput. Phys. Commun.* **207**, 432 (2016). <https://doi.org/10.1016/j.cpc.2016.06.008>
34. V. Shtabovenko, R. Mertig, F. Orellana, *Comput. Phys. Commun.* **256**, 107478 (2020). <https://doi.org/10.1016/j.cpc.2020.107478>. <https://feyncalc.github.io>
35. T. Hahn, *Comput. Phys. Commun.* **140**, 418 (2001). [https://doi.org/10.1016/S0010-4655\(01\)00290-9](https://doi.org/10.1016/S0010-4655(01)00290-9). <http://www.feynarts.de>