COFFE v2.0
User guide

Goran Jelic-Cizmek and Vittorio Tansella

November 7, 2020

Contents

1 How to run COFFE **3**
- 1.1 `Docker` (multiplatform) . . . 3
- 1.2 Building from source . . . 3

2 Settings **4**
- 2.1 Input . . . 4
- 2.2 Output . . . 6
- 2.3 Precision Settings . . . 8
- 2.4 Generation of the matter power spectrum using an external library . . . 8
- 2.5 Detailed description of outputs . . . 10
  - 2.5.1 Angular correlation function $\xi(\theta)$ . . . 10
  - 2.5.2 Correlation function $\xi(r, \mu)$ . . . 10
  - 2.5.3 Multipoles $\xi_\ell(r)$ . . . 11
  - 2.5.4 z-averaged Multipoles $\Xi_\ell(r)$ . . . 11
  - 2.5.5 Covariance matrices $\mathrm{cov}_{\ell\ell'}(r_i, r_j)$ . . . 12
  - 2.5.6 2D correlation function $\xi(r_\parallel, r_\perp)$ . . . 12
- 2.6 Background functions . . . 12

3 FAQ **13**

4 The `functions.c` list **13**

Contacts

COFFE is the product of the work of Camille Bonvin, Ruth Durrer, Goran Jelic-Cizmek and Vittorio Tansella, developed at the University of Geneva Cosmology group (https://cosmology.unige.ch/content/coffe).
Please report bugs, ask questions and send comments/suggestions through the GitHub website. Go to https://github.com/JCGoran/coffe/issues and submit a new issue. This will automatically send an e-mail to the COFFE developers, and they will answer you as soon as possible. You can also directly write to Goran Jelic-Cizmek for technical support.

Papers

- V. Tansella, C. Bonvin, R. Durrer, B. Ghosh and E. Sellentin, "*The full-sky relativistic correlation function and power spectrum of galaxy number counts. Part I: Theoretical aspects*", **JCAP 1803** (2018) 019, [arXiv:1708.00492].
- V. Tansella, G. Jelic-Cizmek, C. Bonvin and R. Durrer "COFFE*: a code for the full-sky relativistic galaxy correlation function*", **JCAP 1810** (2018) 032, [arXiv:1806.11090].
- G. Jelic-Cizmek, "*The flat-sky approximation to galaxy number counts - redshift space correlation function*", [arXiv:2011.01878].

Copyright

Copyright © 2020 Goran Jelic-Cizmek and Vittorio Tansella. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

1 How to run COFFE

1.1 Docker (multiplatform)

Docker is a containerised virtual machine that allows us to run the code in a platform independent way, assuring that it has all the necessary library requirements built in across all platforms. To run COFFE:

- Install Docker, available at
  - Mac OS: https://docs.docker.com/docker-for-mac/install/.
  - Windows: https://docs.docker.com/docker-for-windows/install/.
  - GNU/Linux: https://docs.docker.com/install/linux/docker-ce/debian/.
- When Docker is successfully installed, open a terminal and run:

```
docker pull jcgoran/coffe
```

- Download a sample parameters file, available at Github, and save it somewhere.
- Change the settings file using your favorite editor.
- You can now launch a copy of the image (called a container) by running:

```
docker run -ti --rm \
    --user "$(id -u):$(id -g)" \
    -v "$(pwd):/data" \
    jcgoran/coffe coffe -s <SETTINGSFILE> -n <NUMTHREADS>
```

  where <SETTINGSFILE> is the path to the settings file, and <NUMTHREADS> is the number of processes you want to use for the calculation.

1.2 Building from source

To build COFFE from source, you need to have a C99 compatible compiler, and install the following libraries:

- FFTW
- libconfig
- GSL
- CUBA (optional, but recommended)

- `CLASS` (optional, but recommended)

Then you need to download a `tar.gz` archive of COFFE, which are available on Github, and usually have the format `coffe-vX.Y.Z.tar.gz`, where X, Y, Z, denote the major, minor, and patch version numbers, respectively. Save the `tar.gz` archive somewhere on your machine, and from the terminal, `cd` to where you saved it. Then run

```
tar xf coffe-vX.Y.Z.tar.gz
```

This will create a directory `coffe-vX.Y.Z`.
Then, while inside the `coffe-vX.Y.Z` directory, run

```
./configure && make -j coffe
```

If you encounter errors with library dependencies while running `configure`, make sure that the script can find them. To enable CLASS or CUBA, use the options `--enable-class` and `--enable-cuba` when running the `./configure` script.
When you have successfully compiled COFFE, to run the program

```
./coffe -s settings.cfg
```

or

```
./coffe -s settings.cfg -n <number of cores>
```

where `<number of cores>` defaults to 1.

2 Settings

To change the input and output of COFFE, you will need to modify the settings file: `settings.cfg`

2.1 Input

(1.a) The list of separations (in Mpc/h) for which to compute the correlation function is read from the file parsed with `input_separations`. This is relevant for `output_type=1,2,3`. In case the file is not found, or cannot be read, COFFE will use the following default separations: $[10., 20., 40., 100., 120., 150., 200., 250., 300., 350.]$ Mpc/h.
(1.b) The linear power spectrum $P(k, z = 0)$ is read from the file parsed with `input_power_spectrum`. Note that k must be in h/Mpc and $P(k)$ in units (Mpc/h)^3, in the form $[k \,|\, P(k)]$, and the power spectrum must be in the synchronous gauge (which is equivalent to the comoving gauge during matter domination). In case the file is not found or cannot be read, COFFE will use a default power spectrum.

If `have_window` is set to 1, the power spectrum is multiplied with a real-space tophat, of size `window_size` (input in Mpc/h), denoted by R:

$$P_{\rm windowed}(k) = \left[3\frac{j_1(kR)}{kR}\right]^2 P(k)$$

where j_1 is the spherical Bessel function of order 1.
(1.c) Matter and radiation density parameters today

$$\Omega_{\rm cdm} \to \texttt{omega\_cdm},$$

$$\Omega_{\rm baryon} \to \texttt{omega\_baryon},$$

$$\Omega_{\gamma} \to \texttt{omega\_gamma}.$$

The dark energy density is computed as $\Omega_\Lambda = 1-\Omega_m-\Omega_\gamma$. Note that the code does not distinguish between dark matter and baryons. The input power spectrum is what set most of the cosmological parameters and must of course satisfy $\Omega_m = \Omega_{\rm baryon} + \Omega_{\rm cdm}$.
(1.d) Dark energy equation of state parameters `w0` and `wa`. We use the parametrisation

$$w(z) = w_0 + w_a\frac{z}{1+z}\,.$$

(1.e) Galaxy bias, magnification bias and evolution bias (with the notation of 1708.00492)

$$b \to \texttt{galaxy\_bias1}, \texttt{galaxy\_bias2}\,,$$

$$s \to \texttt{magnification\_bias1}, \texttt{magnification\_bias2}\,,$$

$$f_{\rm evo} \to \texttt{evolution\_bias1}, \texttt{evolution\_bias2}\,.$$

The labels `1` and `2` allow for two different population of galaxies with different biases. Note that they must be the same if you are interested in only one population of galaxies. If the biases are function of redshift you can read a file which contains the biases evolution in the form $[z\,|\,b(z)]$.
(1.f) Parameters for the covariance matrix (relevant if `output_type=4,5`). The mean number density $\bar{n}$ (in $(h/{\rm Mpc})^3$) and the sky coverage $f_{\rm sky}$. To optimise the runtime you can compute the covariance for a list of redshift bins with mean redshift $\bar{z}_i$ and thickness δz_i:

$$\bar{z} \to \texttt{covariance\_z\_mean} = \{\bar{z}_1, \bar{z}_2, ...\}\,,$$

$$\delta z \to \texttt{covariance\_deltaz} = \{\delta z_1, \delta z_2, ...\}\,,$$

$$\bar{n} \to \texttt{covariance\_density} = \{\bar{n}(z_1), \bar{n}(z_2), ...\}\,,$$

$$f_{\rm sky} \to \texttt{covariance\_fsky} = \{f_{\rm sky,1}, f_{\rm sky,2}, ...\}\,.$$

The pixel size L_p (in Mpc/h) is fixed for all bins

$$L_p \to \texttt{covariance\_pixelsize}\,.$$

The covariance for the redshift averaged multipoles (`output_type=5`) does not read the $\bar{z}_i$ and δz_i but a list of $z_{\mathrm{min},i}$ and $z_{\mathrm{max},i}$ that delimit the bins.

$$z_{\mathrm{min}} \rightarrow \texttt{covariance\_zmin} = \{z_{\mathrm{min},1}, z_{\mathrm{min},2}, ...\}\,,$$

$$z_{\mathrm{max}} \rightarrow \texttt{covariance\_zmax} = \{z_{\mathrm{max},1}, z_{\mathrm{max},2}, ...\}\,,$$

`covariance_minimum_separation` sets the minimum separation (in units Mpc/h) for which the covariance will be computed.
`covariance_integration_method` sets the integration method for the computation of the covariance integrals, which are of the form:

$$\mathcal{D}_{\ell,\ell'}(r,r') = \int_0^\infty \mathrm{d}k\, k^2\, P(k)\, j_\ell(kr)\, j_{\ell'}(kr'), \quad \mathcal{G}_{\ell,\ell'}(r,r') = \int_0^\infty \mathrm{d}k\, k^2\, P^2(k)\, j_\ell(kr)\, j_{\ell'}(kr')$$

The two possible options are:

- `1` → use the standard GSL quadrature method
- `2` → use a 2D FFTlog method, described in arXiv:2004.04833

In case `covariance_integration_method = 2`, `covariance_integration_bins` sets the number sampling points N for the FFTlog. Note that the time taken, as well as the memory (RAM) usage, scale like $\mathcal{O}(N^2)$, so make sure that for a given number of sampling points you have sufficient free memory available (otherwise your machine may freeze up, or COFFE may crash). As a rough guideline, do not set $N \gtrsim 10000$ if you have less than 5 GB of free memory available.
Additionally, if `covariance_integration_method = 2`, you can set the 2D interpolation method by setting `covariance_interpolation_method`. The two possible options are:

- `1` → bilinear interpolation (less memory usage, faster, less accurate)
- `2` → bicubic interpolation (larger memory usage, slower, more accurate)

2.2 Output

(2.a) Output path and name: `output_path`, `output_prefix`. If `output_path` doesn't exist, COFFE will attempt to create it. If `output_path = ‘‘$TIME"`, the prefix is automatically set to `YYYY-MM-DD-HH-mm-SS_`, denoting the year, month, day, hour, minute and second, respectively.
(2.b) Select the relativistic effects to include in the computation. For example the full list is given by:

```
correlation_contributions = ["den","rsd","len","d1","d2","g1","g2","g3","g4","g5"];
```

By default, the total signal is computed as:

$$A\,A + B\,B + B\,A + A\,B$$

with A and B one of the above terms in `correlation_contributions`.
`only_cross_correlations` sets whether we want to consider just the cross-correlations in the signal, i.e.

$$A\,B + B\,A$$

The possible values are:

- `0` $\rightarrow$ consider the full signal
- `1` $\rightarrow$ consider just the cross-correlations

(2.c) The desired type of output: `output_type` can be equal to

$$\begin{aligned}
&\texttt{0} = \text{ angular correlation function } \xi(\theta, \bar{z})\,,\\
&\texttt{1} = \text{ correlation function } \xi(r, \mu, \bar{z})\,,\\
&\texttt{2} = \text{ multipoles of the correlation function } \xi_\ell(r, \bar{z})\,,\\
&\texttt{3} = \text{ averaged multipoles of the correlation function } \Xi_\ell(r, \bar{z})\,,\\
&\texttt{4} = \text{ covariance for the multipoles } \mathrm{cov}^{\xi}_{\ell\ell'}(r_i, r_j, \bar{z})\,,\\
&\texttt{5} = \text{ covariance for the averaged multipoles } \mathrm{cov}^{\Xi}_{\ell\ell'}(r_i, r_j, \bar{z})\,,\\
&\texttt{6} = \text{ 2D correlation function } \xi(r_\parallel, r_\perp)\,.
\end{aligned} \tag{1}$$

(2.d) The mean redshift for $\xi(\theta, \bar{z})$, $\xi(r, \mu, \bar{z})$, $\xi_\ell(r, \bar{z})$, $\xi(r_\parallel, r_\perp)$: `z_mean`.
(2.e) Redshift bin in which the output is computed. For $\xi(\theta, \bar{z})$, $\xi(r, \mu, \bar{z})$, $\xi_\ell(r, \bar{z})$ use `deltaz` (which has to be bigger than $\bar{z}$). For Ξ_ℓ use `zmin`, `zmax`.
(2.f) Only relevant if `output_type=1`. The value of μ: `mu`.
(2.g) For `output_type=2,3,4,5`, the multipoles you want to compute:

$$\texttt{multipoles} = [\ell_1, \ell_2, \ell_3, ...]$$

When a covariance matrix is requested this gives $\mathrm{cov}_{\ell\ell'}$ for all the combination $\{\ell_i, \ell_j\}$ with $\ell_i \leq \ell_j$ (the remaining one can be found by transposing the covariance matrix $\mathrm{cov}_{\ell'\ell} = (\mathrm{cov}_{\ell\ell'})^{\mathrm{T}}$).

(2.h) The background quantities to be printed to file. For example

```
output_background = ["z", "a", "H","D1", "f", "comoving_distance", ...];
```

(2.i) The various flat-sky settings for the signal, with possible values `0` (full-sky) and `1` (flat-sky).
`flatsky_local = 1` turns on the flat-sky approximation for the `den`, `rsd`, and `d1` terms. The flat-sky approximation has not been implemented for the local terms `d2`, `g1`, `g2`, `g3`, and those are always computed in full-sky.
`flatsky_local_nonlocal = 1` turns on the flat-sky approximation for the cross-correlations between `den`, `rsd`, `d1` terms, and the `len` term. Note that in flat-sky, the terms RSD-lensing and d1-lensing are zero, and the only non-zero contribution comes from density-lensing. The flat-sky approximation has not been implemented for:

- cross-correlation between local terms, `d2`, `g1`, `g2`, `g3`, and any of the non-local terms, `len`, `g4`, `g5`
- cross-correlation between local terms, `den`, `rsd`, `d1`, and the non-local terms `g4`, `g5`

`flatsky_nonlocal = 1` turns on the flat-sky approximation for the lensing-lensing term. The flat-sky approximation has not been implemented for the non-local terms `g4`, `g5`.

2.3 Precision Settings

(3.a) Sampling rate for the background and time-dependent functions: `background_sampling`.
(3.b) Sampling points for the $I^n_\ell(r)$: `bessel_sampling`.
(3.c) Sampling for the angular correlation function $\xi(\theta, \bar{z})$: `theta_sampling`.
(3.d) Multi-dimensional integrals are computed using monte carlo methods from GSL or CUBA. The integration sampling is given with `integration_sampling`. You can switch with `integration_method` being equal to:

`0` = standard random sampling ,
`1` = `MISER` algorithm, based on recursive stratified sampling ,
`2` = `VEGAS` algorithm, based on importance sampling .

If the `CUBA` library is linked, changing the `integration_method` setting has no effect. The `Docker` version of COFFE comes with the `CUBA` library built-in.
(3.e) Integration range for the I^n_ℓ:

$$k_{\rm min} \to \texttt{k\_min}\,,$$
$$k_{\rm max} \to \texttt{k\_max}\,.$$

(3.f) Interpolation type, parsed with `interpolation`, equal to

`1` = linear ,
`2` = polynomial ,
`3` = cubic spine with natural boundary condition ,
`4` = cubic spine with periodic boundary condition ,
`5` = non-rounded Akima spline with natural boundary conditions ,
`6` = non-rounded Akima spline with periodic boundary conditions ,
`7` = monotone cubic spline .

2.4 Generation of the matter power spectrum using an external library

(4.a) Various settings related to the use of an external library (if linked at compilation) to generate the matter power spectrum on-the-fly. Currently, only CLASS is supported.

`have_class = 1` turns on the generation of the matter power spectrum at $z = 0$ on-the-fly with CLASS.
The following parameters for CLASS are supported:

- `h` $\rightarrow$ the reduced Hubble constant, h (the so called "little h")
- `k_pivot`, `ln_10_pow_10_A_s`, `n_s` $\rightarrow$ the pivot scale k_* of the primordial power spectrum, its amplitude as $\ln(10^{10}A_s)$, and its tilt n_s. The primordial power spectrum is assumed to be of the form:

$$\mathcal{P}_\zeta(k) = A_s \left(\frac{k}{k_*}\right)^{n_s-1}$$

WARNING: SETTINGS BELOW ARE EXPERIMENTAL!
`pk_type` controls how the matter power spectrum will be generated:

- `0` $\rightarrow$ the **linear** matter power spectrum is generated *once*, at $z = 0$, $P_{\rm linear}(k, z = 0)$. The linear matter power (or cross) spectrum can be obtained at any redshift using the relation:

$$P_{\rm linear}(k, z_1, z_2) = D_1(z_1)D_1(z_2)P_{\rm linear}(k, z = 0)$$

- `1` $\rightarrow$ the **linear** matter power spectrum is generated *at every redshift*, i.e. it generates $P_{\rm linear}(k, z_1, z_2)$ above. This is primarily useful for comparing against the case with `pk_type = 0`, to make sure it has been implemented correctly in the code.
- `2` $\rightarrow$ the **nonlinear** matter power spectrum is generated *at every redshift*, using the halofit model, described in arXiv:1208.2701. The **nonlinear** matter cross-spectrum is computed as:

$$P(k, z_1, z_2) = \sqrt{P_{\rm nonlinear}(k, z_1)P_{\rm nonlinear}(k, z_2)}$$

- `3` $\rightarrow$ the **nonlinear** matter power spectrum is generated *at every redshift*, using the augmented halo model, described in arXiv:1505.07833. The **nonlinear** matter cross-spectrum is computed as:

$$P(k, z_1, z_2) = \sqrt{P_{\rm nonlinear}(k, z_1)P_{\rm nonlinear}(k, z_2)}$$

In case `pk_type` is set to 1, 2, or 3, `zeldovich_approximation = 1` additionally turns on the Zel'dovich approximation, as described in arXiv:1905.02078, which computes the cross-spectrum as:

$$P(k, z_1, z_2) = \sqrt{P(k, z_1)P(k, z_2)}\exp\left(-(D_1(z_1) - D_1(z_2))^2\,(k/k_{\rm NL})^2\right)$$

where $P(k, z)$ is either the linear or the nonlinear matter power spectrum from above, and $k_{\rm NL}$ is the nonlinearity scale, defined via:

$$k_{\rm NL}^{-2} \equiv \frac{1}{12\pi^2}\int_0^\infty {\rm d}k\, P_{\rm linear}(k, z = 0)$$

2.5 Detailed description of outputs

In the following section we give some details on the different possible `output_types`. Note that the precise definitions of the quantities that COFFE outputs can be found in the papers listed above. Here we just give some technical details which are not specified there.

2.5.1 Angular correlation function $\xi(\theta)$

This corresponds to `output_type=0`. The code computes the transverse correlation function $\xi(r, \mu = 0)$ (see the following section) and than converts r in angular separations with

$$\theta = \mathrm{ArcCos}\left[1 - \frac{r^2}{2\chi(\bar{z})^2}\right] .$$

2.5.2 Correlation function $\xi(r, \mu)$

This corresponds to `output_type=1`. For the non-integrated terms we define

$$\xi^{AB}(\theta, \chi_1, \chi_2) = D_1(\chi_1) D_1(\chi_2) \sum_{\ell,n} \left(X_\ell^n\big|_A + X_\ell^n\big|_{AB} + X_\ell^n\big|_{BA} + X_\ell^n\big|_B \right) I_\ell^n(r) \,,$$

while for the integrated terms

$$\xi^{AB}(\theta, \chi_1, \chi_2) = \left(Z\big|_A + Z\big|_{AB} + Z\big|_{BA} + Z\big|_B \right)$$

Here A, B are tags that can take the values given in `correlation_contributions`. We have also defined the integrals

$$I_\ell^n(r) = \int \frac{dk\, k^2}{2\pi^2} P(k) \frac{j_\ell(kr)}{(kr)^n} \,,$$

which are the core computation of the correlation function. We have implemented the very accurate 2-FAST algorithm in C and included it in our code[1] to compute these integrals. We then set $\chi_1 = \bar{\chi} - \frac{1}{2} r\mu$, $\chi_2 = \bar{\chi} + \frac{1}{2} r\mu$, and

$$\cos\theta = \frac{2\bar{\chi}^2 - r^2 + \mu^2 r^2/2}{2\bar{\chi}^2 - \mu^2 r^2/2} \,.$$

COFFE has two `for` loops, one going over the separations r and one over the angles μ, parallelized using the `openMP` standard.
For the integrated terms, we use the following rescaling:

$$\lambda = \chi_1 x_1, \quad \lambda' = \chi_2 x_2 \,.$$

[1]The original, publicly available, 2-FAST code (https://github.com/hsgg/twoFAST) is implemented in the high-level language **julia**.

This brings the limits of integration to [0, 1].
The $D_1(z_1)D_1(z_2)X_{AB}$ terms are defined in `functions_nonintegrated`, while the Z_{AB} terms are defined in `functions_single_integrated` and `functions_double_integrated`
for single and double integrals over the comoving distance respectively.

2.5.3 Multipoles $\xi_\ell(r)$

This corresponds to `output_type=2`. In this case, we use the above `functions` to set up the following integral:

$$\xi_\ell(r) = \frac{2\ell+1}{2}\int_{-1}^{1} d\mu\, P_\ell(\mu)\xi(\bar{z},\mu,r) = (2\ell+1)\int_{0}^{1} dx\, P_\ell(2x-1)\xi(\bar{z},2x-1,r)\,.$$

Depending on the type of correlation function (nonintegrated vs. integrated), we use either standard 1D integration (`gsl_integration_qag`), or various Monte Carlo methods:

`monte_<plain/miser/vegas>`,

in either 2 or 3 dimensions.
The user can alternatively use the `CUBA` library described below which uses cubature rules, and is typically much faster.

2.5.4 z-averaged Multipoles $\Xi_\ell(r)$

This corresponds to `output_type=3`. The $\Xi_\ell(r)$ are computed from the following integral:

$$\Xi_\ell(r,z_1,z_2) = \frac{1}{z_2-z_1}\int_{z_1}^{z_2} d\bar{z}\,\frac{2\ell+1}{2}\int_{-1}^{1} d\mu\, P_\ell(\mu)\frac{\xi(\bar{z},\mu,r)}{\mathcal{H}(\bar{z})(1+\bar{z})}$$

$$= (2\ell+1)\int_{0}^{1} dy \int_{0}^{1} dx\, P_\ell(2x-1)\frac{\xi((z_2-z_1)y+z_1,2x-1,r)}{\mathcal{H}((z_2-z_1)y+z_1)(1+(z_2-z_1)y+z_1)}\,.$$

Note that z_1 and z_2 are computed from the input `z_min` and `z_max` as:

$$z_1 = z\left[\chi(z_{min}) + \frac{r}{2}\right]$$
$$z_2 = z\left[\chi(z_{max}) - \frac{r}{2}\right]$$

2.5.5 Covariance matrices $\mathrm{cov}_{\ell\ell'}(r_i, r_j)$

This corresponds to `output_type=4` or `5`. The covariance is built from eqs. (2.51), (2.52) of the COFFE paper. Note that for thick redshift bins the covariance computed with the GSL method (`covariance_integration_method = 1`) might not be positive definite because of numerical fluctuations.
The output is given as a table $[r_i \,|\, r_j, |\, \mathrm{cov}(r_i, r_j))]$.

2.5.6 2D correlation function $\xi(r_\parallel, r_\perp)$

This corresponds to `output_type=6`. Here we compute the correlation function for a predefined grid of parallel and transverse separations $r_\parallel$ and $r_\perp$ up to $300\,\mathrm{Mpc}/h$. Given section 2.5.2 we simply translate the $\xi(r, \mu, \bar{z})$ into $\xi(r_\parallel, r_\perp, \bar{z})$ with

$$r_\parallel = r\mu\,, \quad r_\perp = \sqrt{r^2 - r_\parallel^2}\,.$$

The output is given as a table $[r_\parallel \,|\, r_\perp \,|\, \xi(r_\parallel, r_\perp)]$.

2.6 Background functions

We report here the list of background and time-dependent functions relevant for the code and how they are computed:

$$\begin{aligned}
a(z) &= \frac{1}{1+z} \\
H(z) &= H_0\sqrt{\Omega_m^0(1+z)^3 + \Omega_\gamma^0(1+z)^4 + \Omega_{\rm DE}^0 \exp\left\{3\int_0^z \frac{1+w(z')}{1+z'}dz'\right\}} \\
&\stackrel{*}{=} H_0\sqrt{\Omega_m^0(1+z)^3 + \Omega_\gamma^0(1+z)^4 + \Omega_{\rm DE}^0(1+z)^{3(1+w_0+w_a)}\exp\left\{-3w_a\frac{z}{1+z}\right\}} \\
\mathcal{H}(z) &= a(z)H(z) \\
\dot{\mathcal{H}}(z) &= -\frac{H_0^2}{2(1+z)^2}\left((2(1+z)\Omega_\gamma^0 + \Omega_m^0)(1+z)^3 + (1+3w(z))\Omega_{\rm DE}^0\exp\left\{3\int_0^z\frac{1+w(z')}{1+z'}dz'\right\}\right) \\
&\stackrel{*}{=} -\frac{H_0^2}{2}\left(\Omega_{\rm DE}e^{-3w_a\frac{z}{z+1}}(w_a z + 3w_0(z+1) + z + 1)(z+1)^{3(w_a+w_0)} + (z+1)(\Omega_m + 2\Omega_\gamma(z+1))\right) \\
D_1(z) \Rightarrow D'' &+ \frac{3}{2}\left[1 - \frac{w(a)}{1+X(a)}\right]\frac{D'}{a} - \frac{3}{2}\frac{X(a)}{1+X(a)}\frac{D}{a^2} = 0; \quad X(a) = \frac{\Omega_m^0}{1-\Omega_m^0}e^{-3\int_a^1 d(ln\,a')w(a')} \\
g(z) &= (1+z)D(z) \\
f(z) &= \frac{a(z)}{D(z)}\frac{dD}{da}
\end{aligned}$$

$$\chi(z) = \frac{1}{H_0}\int \frac{dz}{\sqrt{\Omega_m^0(1+z)^3 + \Omega_\gamma^0(1+z)^4 + \Omega_{\rm DE}^0 \exp\left\{3\int_0^z \frac{1+w(z')}{1+z'}dz'\right\}}} \quad \text{(in units Mpc/h)}$$

$$r_p(z) = a(z)\chi(z) \quad \text{(in units Mpc/h)}$$

$$G(z) = \frac{\dot{\mathcal{H}}}{\mathcal{H}^2} + \frac{2-5s}{\chi\mathcal{H}} + 5s - f_{\rm evo}$$

where $\overset{*}{=}$ means we have made use of the parametrisation

$$w(z) \equiv w_0 + w_a \frac{z}{1+z}\,.$$

3 FAQ

The settings I save after running the program are different from the settings I input!
This is a bug in the `libconfig` library, see https://github.com/hyperrealm/libconfig/issues/58 for clarification. In short, an earlier version of the library has a precision-related bug when saving the input. The only solution is to upgrade to the most recent one.

What is the `CUBA` library for?
`CUBA` is a library for multidimensional numerical integration (more details on http://www.feynarts.de/cuba/). It can optionally be used to compute the double integrated terms in less time and higher precision than the GSL Monte Carlo methods. Currently the `Cuhre` cubature method is being used.
To use it in the code, you need to add `--enable-cuba` when running `./configure`.

There's something wrong with the compiled version!/The output doesn't make sense!

If you think there is a bug in COFFE itself, please open an issue on Github or contact me directly on my email address.
If not, but the output is still not what it's supposed to be, you may opt for using the `Docker` version. If the problem persists, please open an issue on Github.

4 The `functions.c` list

The relevant functions in `functions.c` are defined as:

$$X_0^0\big|_{\rm den} = b_1 b_2\,,$$
$$X_0^0\big|_{\rm rsd} = f_1 f_2 \frac{1+2\cos^2\theta}{15}\,,$$

$$X_2^0\big|_{\rm rsd} = -\frac{f_1 f_2}{21}\left[1+11\cos^2\theta+\frac{18\cos\theta(\cos^2\theta-1)\chi_1\chi_2}{r^2}\right],$$
$$X_4^0\big|_{\rm rsd} = f_1 f_2\left[\frac{4(3\cos^2\theta-1)(\chi_1^4+\chi_2^4)}{35r^4}+\chi_1\chi_2(3+\cos^2\theta)\frac{3(3+\cos^2\theta)\chi_1\chi_2-8(\chi_1^2+\chi_2^2)\cos\theta}{35r^4}\right],$$
$$X_0^2\big|_{\rm d1} = \mathcal{H}_1\mathcal{H}_2 f_1 f_2 G_1 G_2\frac{r^2\cos\theta}{3},$$
$$X_2^2\big|_{\rm d1} = -\mathcal{H}_1\mathcal{H}_2 f_1 f_2 G_1 G_2\left((\chi_2-\chi_1\cos\theta)(\chi_1-\chi_2\cos\theta)+\frac{r^2\cos\theta}{3}\right),$$
$$X_0^4\big|_{\rm d2} = (3-f_{\rm evo1})(3-f_{\rm evo2})\,r^4\mathcal{H}_1^2\mathcal{H}_2^2 f_1 f_2,$$
$$X_0^4\big|_{\rm g1} = \frac{9\,r^4\Omega_m^2}{4a_1a_2}(1+G_1)(1+G_2)\mathcal{H}_0^4,$$
$$X_0^4\big|_{\rm g2} = \frac{9\,r^4\Omega_m^2}{4a_1a_2}(5s_1-2)(5s_2-2)\mathcal{H}_0^4,$$
$$X_0^4\big|_{\rm g3} = \frac{9\,r^4\Omega_m^2}{4a_1a_2}(f_1-1)(f_2-1)\mathcal{H}_0^4$$
$$X_0^0\big|_{\rm den\text{-}rsd} = \frac{b_1 f_2}{3},$$
$$X_2^0\big|_{\rm den\text{-}rsd} = -b_1 f_2\left(\frac{2}{3}-(1-\cos^2\theta)\frac{\chi_1^2}{r^2}\right),$$
$$X_1^1\big|_{\rm den\text{-}d1} = -b_1 f_2\mathcal{H}_2 G_2(\chi_1\cos\theta-\chi_2),$$
$$X_0^2\big|_{\rm den\text{-}d2} = (3-f_{\rm evo2})\,r^2 b_1 f_2\mathcal{H}_2^2,$$
$$X_0^2\big|_{\rm den\text{-}g1} = -b_1\frac{3\Omega_m}{2a_2}(1+G_2)r^2\mathcal{H}_0^2,$$
$$X_0^2\big|_{\rm den\text{-}g2} = -b_1\frac{3\Omega_m}{2a_2}(5s_2-2)r^2\mathcal{H}_0^2,$$
$$X_0^2\big|_{\rm den\text{-}g3} = -b_1\frac{3\Omega_m}{2a_2}(f_2-1)r^2\mathcal{H}_0^2,$$
$$X_1^1\big|_{\rm rsd\text{-}d1} = f_1 f_2\mathcal{H}_2 G_2\frac{(1+2\cos^2\theta)\chi_2-3\chi_1\cos\theta}{5},$$
$$X_3^1\big|_{\rm rsd\text{-}d1} = f_1 f_2\mathcal{H}_2 G_2\frac{(1-3\cos\theta)\chi_2^3+\cos\theta(5+\cos^2\theta)\chi_2^2\chi_1-2(2+\cos\theta^2)\chi_2\chi_1^2+2\chi_1^3\cos\theta}{5r^2},$$
$$X_0^2\big|_{\rm rsd\text{-}d2} = \frac{3-f_{\rm evo2}}{3}f_1 f_2 r^2\mathcal{H}_2^2,$$
$$X_2^2\big|_{\rm rsd\text{-}d2} = -(3-f_{\rm evo2})f_1 f_2\mathcal{H}_2^2\left(\frac{2}{3}r^2-(1-\cos^2\theta)\chi_2^2\right),$$
$$X_0^2\big|_{\rm rsd\text{-}g1} = -\frac{\Omega_m}{2a_2}f_1(1+G_2)r^2\mathcal{H}_0^2,$$

$$
\begin{aligned}
X_2^2\big|_{\text{rsd-g1}} &= \frac{3\Omega_m}{2a_2} f_1(1+G_2)\mathcal{H}_0^2 \left(\frac{2}{3}r^2 - (1-\cos^2\theta)\chi_2^2\right) , \\
X_0^2\big|_{\text{rsd-g2}} &= -\frac{\Omega_m}{2a_2} f_1(5s_2-2) r^2 \mathcal{H}_0^2 , \\
X_2^2\big|_{\text{rsd-g2}} &= \frac{3\Omega_m}{2a_2} f_1(5s_2-2)\mathcal{H}_0^2 \left(\frac{2}{3}r^2 - (1-\cos^2\theta)\chi_2^2\right) , \\
X_0^2\big|_{\text{rsd-g3}} &= -\frac{\Omega_m}{2a_2} f_1(f_2-1) r^2 \mathcal{H}_0^2 , \\
X_2^2\big|_{\text{rsd-g3}} &= \frac{3\Omega_m}{2a_2} f_1(f_2-1)\mathcal{H}_0^2 \left(\frac{2}{3}r^2 - (1-\cos^2\theta)\chi_2^2\right) , \\
X_1^3\big|_{\text{d1-d2}} &= -(3-f_{\text{evo2}})\mathcal{H}_1\mathcal{H}_2^2 f_1 f_2\, r^2(\chi_2\cos\theta - \chi_1) , \\
X_1^3\big|_{\text{d1-g1}} &= \frac{3\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1 f_1(1+G_2)\, r^2(\chi_2\cos\theta - \chi_1) , \\
X_1^3\big|_{\text{d1-g2}} &= \frac{3\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1 f_1(5s_2-2)\, r^2(\chi_2\cos\theta - \chi_1) , \\
X_1^3\big|_{\text{d1-g3}} &= \frac{3\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1 f_1(f_2-1)\, r^2(\chi_2\cos\theta - \chi_1) , \\
X_0^4\big|_{\text{d2-g1}} &= -\frac{3(3-f_{\text{evo1}})\, r^4\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1^2 f_1(1+G_2) , \\
X_0^4\big|_{\text{d2-g2}} &= -\frac{3(3-f_{\text{evo1}})\, r^4\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1^2 f_1(5s_2-2) , \\
X_0^4\big|_{\text{d2-g3}} &= -\frac{3(3-f_{\text{evo1}})\, r^4\Omega_m}{2a_2}\mathcal{H}_0^2\mathcal{H}_1^2 f_1(f_2-1) , \\
X_0^4\big|_{\text{g1-g2}} &= \frac{9\, r^4\Omega_m^2}{4a_1a_2}\mathcal{H}_0^4(1+G_1)(5s_2-2) , \\
X_0^4\big|_{\text{g1-g3}} &= \frac{9\, r^4\Omega_m^2}{4a_1a_2}\mathcal{H}_0^4(1+G_1)(f_2-1) , \\
X_0^4\big|_{\text{g2-g3}} &= \frac{9\, r^4\Omega_m^2}{4a_1a_2}\mathcal{H}_0^4(5s_1-2)(f_2-1) .
\end{aligned}
$$

where

$$
G(z) = \frac{\dot{\mathcal{H}}}{\mathcal{H}^2} + \frac{2-5s}{\chi\mathcal{H}} + 5s - f_{\text{evo}} . \tag{2}
$$

The full list of Z_ℓ^n is given:

$$
Z\big|_{\text{len}} = \frac{9\Omega_m^2}{4}\mathcal{H}_0^4(2-5s_1)(2-5s_2)\int_0^1 dx_1 \int_0^1 dx_2 \frac{(1-x_1)(1-x_2)}{x_1x_2}\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}\left\{\frac{2}{5}(\cos^2\theta-1)\lambda^2\lambda'^2 I_0^0(r)\right.
$$

$$+\frac{4r^2\cos\theta\lambda\lambda'}{3}I_0^2(r)+\frac{4\cos\theta\lambda\lambda'(r^2+6\cos\theta\lambda\lambda')}{15}I_1^1(r)+\frac{2(\cos^2\theta-1)\lambda^2\lambda'^2(2r^2+3\cos\theta\lambda\lambda')}{7r^2}I_2^0(r)$$
$$+\frac{2\cos\theta\lambda\lambda'\left(2r^4+12\cos\theta r^2\lambda\lambda'+15(\cos^2\theta-1)\lambda^2\lambda'^2\right)}{15r^2}I_3^1(r)$$
$$+\frac{(\cos^2\theta-1)\lambda^2\lambda'^2\left(6r^4+30\cos\theta r^2\lambda\lambda'+35(\cos^2\theta-1)\lambda^2\lambda'^2\right)}{35r^4}I_4^0(r)\Bigg\},$$

$$Z\big|_{\rm g4}=9\Omega_m^2\mathcal{H}_0^4(2-5s_1)(2-5s_2)\int_0^1 dx_1\int_0^1 dx_2\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}r^4I_0^4(r),$$

$$Z\big|_{\rm g5}=9\Omega_m^2\mathcal{H}_0^4G_1G_2\chi_2\chi_2\int_0^1 dx_1\int_0^1 dx_2\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}\mathcal{H}(\lambda)\mathcal{H}(\lambda')(f(\lambda)-1)(f(\lambda')-1)r^4I_0^4(r),$$

$$Z\big|_{\rm den\text{-}len}=-\frac{3\Omega_m}{2}b_1\chi_2\mathcal{H}_0^2(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\left\{2\chi_1\cos\theta I_1^1(r)-\frac{\chi_1^2\lambda(1-\cos^2\theta)}{r^2}I_2^0(r)\right\},$$

$$Z\big|_{\rm rsd\text{-}len}=\frac{3\Omega_m}{2}f_1\chi_2\mathcal{H}_0^2(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Bigg\{\frac{1}{15}(\lambda-6\chi_1\cos\theta+3\lambda\cos2\theta)I_0^0(r)$$
$$-\frac{6\chi_1^3\cos\theta-\chi_1^2\lambda\left(9\cos^2\theta+11\right)+\chi_1\lambda^2\cos\theta(3\cos2\theta+19)-2\lambda^3(3\cos2\theta+1)}{21r^2}I_2^0(r)$$
$$-\frac{1}{35r^4}\Bigg[-4\chi_1^5\cos\theta-\chi_1^3\lambda^2\cos\theta(\cos2\theta+7)+\chi_1^2\lambda^3\left(\cos^4\theta+12\cos^2\theta-21\right)$$
$$-3\chi_1\lambda^4\cos\theta(\cos2\theta-5)-\lambda^5(3\cos2\theta+1)+12\chi_1^4\lambda\Bigg]I_4^0(r)\Bigg\},$$

$$Z\big|_{\rm d1\text{-}len}=\frac{3\Omega_m}{2}\chi_2\mathcal{H}_0^2\mathcal{H}_1f_1G_1(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Bigg\{\frac{2}{15}\Bigg(\cos\theta\left(\lambda^2-2\chi_1^2\right)$$
$$+\chi_1\lambda(2\cos2\theta-1)\Bigg)I_1^1(r)+\frac{2}{3}r^2\cos\theta I_0^2(r)$$
$$-\frac{4\chi_1^4\cos\theta-\chi_1^3\lambda\left(\cos^2\theta+9\right)+\chi_1^2\lambda^2\cos\theta\left(\cos^2\theta+5\right)-2\chi_1\lambda^3(\cos2\theta-2)-2\lambda^4\cos\theta}{15r^2}I_3^1(r)\Bigg\},$$

$$Z\big|_{\rm d2\text{-}len}=-\frac{3\Omega_m}{2}\chi_2(3-f_{\rm evo})f_1\mathcal{H}_1^2\mathcal{H}_0^2(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Bigg\{2\chi_1r^2\cos\theta I_1^3(r)$$
$$-\chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Bigg\},$$

$$Z\big|_{\text{g1-len}} = \frac{9\Omega_m^2}{4}\chi_2(1+G_1)\mathcal{H}_0^4(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Big\{2\chi_1 r^2\cos\theta I_1^3(r)$$
$$-\chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

$$Z\big|_{\text{g2-len}} = \frac{9\Omega_m^2}{4}\chi_2(5s_1-2)\mathcal{H}_0^4(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Big\{2\chi_1 r^2\cos\theta I_1^3(r)$$
$$-\chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

$$Z\big|_{\text{g3-len}} = \frac{9\Omega_m^2}{4}(f_1-1)\chi_2\mathcal{H}_0^4(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\Big\{2\chi_1 r^2\cos\theta I_1^3(r)$$
$$-\chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

$$Z\big|_{\text{g4-len}} = \frac{9\Omega_m^2}{2}\mathcal{H}_0^4(2-5s_1)(2-5s_2)\int_0^1 dx_1\int_0^1 dx_2\frac{1-x_2}{x_2}\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}\Big\{2\lambda\lambda' r^2\cos\theta I_1^3(r)$$
$$-\lambda^2\lambda'^2(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

$$Z\big|_{\text{g5-len}} = \frac{9\Omega_m^2}{2}\chi_1\mathcal{H}_0^4 G_1(2-5s_2)\int_0^1 dx_1\int_0^1 dx_2\mathcal{H}(\lambda)(f(\lambda)-1)\frac{1-x_2}{x_2}\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}\Big\{2\lambda\lambda' r^2\cos\theta I_1^3(r)$$
$$-\lambda^2\lambda'^2(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

$$Z\big|_{\text{den-g4}} = -3\Omega_m\mathcal{H}_0^2 b_1(2-5s_2)D_1(z_1)\int_0^1 dx\frac{D_1(\lambda)}{a(\lambda)}r^2 I_0^2(r)\,,$$

$$Z\big|_{\text{den-g5}} = -3\Omega_m\mathcal{H}_0^2 b_1\chi_2 G_2 D_1(z_1)\int_0^1 dx\mathcal{H}(\lambda)(f(\lambda)-1)\frac{D_1(\lambda)}{a(\lambda)}r^2 I_0^2(r)\,,$$

$$Z\big|_{\text{rsd-g4}} = 3\Omega_m\mathcal{H}_0^2 f_1(2-5s_2)D_1(z_1)\int_0^1 dx\frac{D_1(\lambda)}{a(\lambda)}\Big\{\left(\frac{2r^2}{3}+(\cos^2\theta-1)\lambda^2\right)I_2^2(r)-\frac{r^2}{3}I_0^2(r)\Big\}\,,$$

$$Z\big|_{\text{rsd-g5}} = 3\Omega_m\mathcal{H}_0^2 f_1\chi_2 G_2 D_1(z_1)\int_0^1 dx\mathcal{H}(\lambda)(f(\lambda)-1)\frac{D_1(\lambda)}{a(\lambda)}\Big\{\left(\frac{2r^2}{3}+(\cos^2\theta-1)\lambda^2\right)I_2^2(r)-\frac{r^2}{3}I_0^2(r)\Big\}\,,$$

$$Z\big|_{\text{d1-g4}} = 3\Omega_m \mathcal{H}_0^2 \mathcal{H}_1 f_1 (2-5s_2) D_1(z_1) \int_0^1 dx \frac{D_1(\lambda)}{a(\lambda)} \left\{ r^2 (\lambda \cos\theta - \chi_1) I_1^3(r) \right\},$$

$$Z\big|_{\text{d1-g5}} = 3\Omega_m \mathcal{H}_0^2 \mathcal{H}_1 f_1 \chi_2 G_2 D_1(z_1) \int_0^1 dx \mathcal{H}(\lambda)(f(\lambda)-1) \frac{D_1(\lambda)}{a(\lambda)} \left\{ r^2 (\lambda \cos\theta - \chi_1) I_1^3(r) \right\},$$

$$Z\big|_{\text{d2-g4}} = -3\Omega_m \mathcal{H}_0^2 (3 - f_{\text{evo}}) f_1 \mathcal{H}_1^2 (2-5s_2) D_1(z_1) \int_0^1 dx \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{d2-g5}} = -3\Omega_m \mathcal{H}_0^2 (3 - f_{\text{evo}}) f_1 \chi_2 \mathcal{H}_1^2 G_2 D_1(z_1) \int_0^1 dx \mathcal{H}(\lambda)(f(\lambda)-1) \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g1-g4}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 (1+G_1)(2-5s_2) D_1(z_1) \int_0^1 dx \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g1-g5}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 \chi_2 (1+G_1) G_2 D_1(z_1) \int_0^1 dx \mathcal{H}(\lambda)(f(\lambda)-1) \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g2-g4}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 (5s_1-2)(2-5s_2) D_1(z_1) \int_0^1 dx \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g2-g5}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 \chi_2 (5s_1-2) G_2 D_1(z_1) \int_0^1 dx \mathcal{H}(\lambda)(f(\lambda)-1) \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g3-g4}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 (f_1-1)(2-5s_2) D_1(z_1) \int_0^1 dx \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g3-g5}} = \frac{9\Omega_m^2}{2a_1} \mathcal{H}_0^4 \chi_2 (f_1-1) G_2 D_1(z_1) \int_0^1 dx \mathcal{H}(\lambda)(f(\lambda)-1) \frac{D_1(\lambda)}{a(\lambda)} r^4 I_0^4(r),$$

$$Z\big|_{\text{g4-g5}} = 9\Omega_m^2 \mathcal{H}_0^4 G_2 \chi_2 (2-5s_1) \int_0^1 dx_1 \int_0^1 dx_2 \mathcal{H}(\lambda')(f(\lambda')-1) \frac{D_1(\lambda) D_1(\lambda')}{a(\lambda) a(\lambda')} r^2 I_0^4(r).$$