# COFFE v1.0
## User guide

Goran Jelic-Cizmek and Vittorio Tansella

June 29, 2018

# Contents

## 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` 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 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*", [arXiv:1806.11090].

## Copyright

# 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: 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, go to https://github.com/JCGoran/coffe/releases/tag/1.0-docker and download the coffe-1.0.tar.gz file.
- Open a terminal window, in the directory where you downloaded the file, and type:

```
docker load -i coffe-1.0.tar.gz
```

- You now need to launch a copy of the image (called a container) by running:

```
docker run -ti coffe-docker:v1.0
```

- The command prompt now that starts with '/#': it means you're running the container, and can interact with it. The COFFE binary, along with all the config. files, is in

```
cd /build
```

  To run the program

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

  or

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

  and <number of cores> defaults to 1.

After COFFE has successfully run, there are two things that you probably want to do:

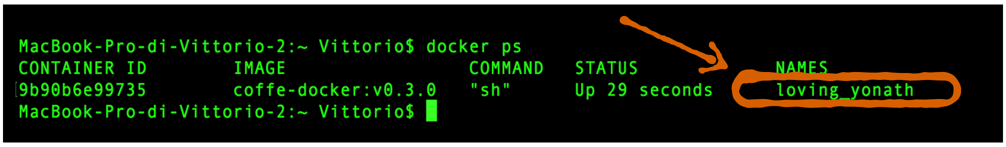
- Copy the output to your local machine, or copy something to the coffe directory. To copy the output on your hard drive, while still running the container, go to another terminal window, and do:

```
docker cp <CONTAINER_ID>:<SOURCE_DIRECTORY> <DESTINATION_DIRECTORY>
```

where <SOURCE_DIRECTORY> is the directory on the Docker container you copy from (i.e. /build/results), and <DESTINATION_DIRECTORY> is the directory on your local machine where you copy to. The <CONTAINER_ID> is the name of the Docker container: to obtain it open another terminal window (while still running the container) and type

```
docker ps
```

You will find the <CONTAINER_ID> here:




The ordering is reverse if you are copying from the local machine to the running Docker container:

```
docker cp <SOURCE_DIRECTORY> <CONTAINER_ID>:<DESTINATION_DIRECTORY>
```

Note that if you exit the Docker container (you do this by running exit on the terminal running the container), without copying the output to your machine, all of the data obtained by running COFFE will be lost, and if you run another container, you'll find that there's nothing to find in the results directory.

- Change the settings file. In the /build directory, use the command vi to edit the settings

  ```
  vi settings.cfg
  ```

  (i to enter insert mode and after your modifications are done, :wq to save and quit).

## 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)

Then run

```
./configure && make
```

If you encounter errors with library dependencies while running `configure`, make sure that the script can find them. When you have successfully compiled, to run the program

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

or

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

and `<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`.

**(1.b)** The linear power spectrum $P(k)$ is read from the file parsed with `input_power_spectrum`. Note that $k$ must be in $h$/Mpc and $P(k)$ in units $(\mathrm{Mpc}/h)^3$, 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).

**(1.c)** Matter and radiation density parameters today

$$\Omega_m \to \texttt{omega\_m},$$

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

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 $\Omega_m = \Omega_b + \Omega_{DM}$.

**(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)

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

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

$$f_{\text{evo}} \rightarrow \texttt{evolution\_bias1}, \texttt{evolution\_bias2}.$$

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 $(h/\text{Mpc})^3$) and the sky coverage $f_{\text{sky}}$. To optimise the runtime you can compute the covariance for a list of redshift bins with mean redshift $\bar{z}_i$ and thickness $\delta z_i$:

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

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

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

$$f_{\text{sky}} \rightarrow \texttt{covariance\_fsky} = \{f_{\text{sky},1}, f_{\text{sky},2}, ...\}.$$

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

$$L_p \rightarrow \texttt{covariance\_pixelsize}.$$

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

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

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

## 2.2 Output

The desired output can be selected in the settings file: `settings.cfg`

**(2.a)** Output path and name: `output_path`, `output_prefix`.

**(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"];
```

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

$$\begin{aligned} \texttt{0} &= \text{angular correlation function } \xi(\theta, \bar{z}), \\ \texttt{1} &= \text{correlation function } \xi(r, \mu, \bar{z}), \end{aligned}$$

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

**(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 $\Xi_\ell$ use `zmin`, `zmax`.

**(2.f)** Only relevant if `output_type=1`. The value of $\mu$: `mu`

**(2.g)** For `output_type=2,3,4,5`, the multipoles you want to compute:

`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 $\ell_i\le\ell_j$ (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", ...];
```

## 2.3 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. You can switch with `integration_method` being equal to:

$$
\begin{aligned}
0 &= \text{standard random sampling}\,, \\
1 &= \texttt{MISER} \text{ algorithm, based on recursive stratified sampling}\,, \\
2 &= \texttt{VEGAS} \text{ algorithm, based on importance sampling}\,.
\end{aligned}
$$

The integration sampling is given with `integration_sampling`. Note that, while `integration_sampling` can be tuned both in the `Docker` version and in the compiled one, `integration_method` is only used the compiled version. The `Docker` version integration method is `Cuhre` from the `CUBA` library.

**(3.e)** Integration range for the $I_\ell^n$:

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

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

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

$$\begin{aligned} \texttt{1} &= \text{linear}\,, \\ \texttt{2} &= \text{polynomial}\,, \\ \texttt{3} &= \text{cubic spine with natural boundary condition}\,, \\ \texttt{4} &= \text{cubic spine with periodic boundary condition}\,, \\ \texttt{5} &= \text{non-rounded Akima spline with natural boundary conditions}\,, \\ \texttt{6} &= \text{non-rounded Akima spline with periodic boundary conditions}\,, \\ \texttt{7} &= \text{monotone cubic spline}\,. \end{aligned}$$

In the following section we give some details on the different possible `output_type`s. 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.

### 2.3.1 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 translate $r$ in angular separations with

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

### 2.3.2 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 tag 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 $\mu$, parallelized using the openMP standard.
For the integrated terms, we use the following rescaling:

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

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.

#### 2.3.3 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.

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

### 2.3.4 $z$-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:

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

### 2.3.5 Covariance matrices $\mathbf{cov}_{\ell\ell'}(r_i, r_j)$

This corresponds to output_type=4 or 5. The covariance is built from eqs. (2.51),(2.52) of the COFFE paper. The challenging part of the computation are clearly the integrals $\mathcal{D}_{\ell\ell'}$ and $\mathcal{G}_{\ell\ell'}$. As the 2-FAST algorithm is not optimised to compute covariances[2], it is (at the moment) too slow to be implemented in the public version of the code. We therefore choose to release the first version of COFFE with the covariance implemented in GSL, which is much faster but less precise. Note that this trade off of precision for speed has important drawbacks: for thick redshift bins the GSL covariance might not be positive definite because of numerical fluctuations. In future versions of COFFE we will optimise 2-FAST for covariance calculation and release it to the public.
The output is given as a table $[r_i \,|\, r_j, \,|\, \mathrm{cov}(x_i, x_j))]$.

### 2.3.6 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 2.3.2 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]To be precise, 2-FAST allows for the computation of integrals with two Bessel functions such as $\mathcal{D}_{\ell\ell'}$ and $\mathcal{G}_{\ell\ell'}$. However the algorithm is structured to output them for a list of $x_i$ but fixed $R = x_j/x_i$. In the covariance we however need $N_p^2$ pairs of $(x_i, x_j)$, where $N_p = r_{\max}/L_p$ is the number of pixels in the covariance. To get them, with no modification of the algorithm, we need to run 2-FAST $N_p^2$ times, with a runtime not suitable for a public code.

### 2.4 Background functions

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

$$
\begin{aligned}
a(z) &= \frac{1}{1+z}\\
H(z) &= H_0\sqrt{\Omega_m^0(1+z)^3+\Omega_\gamma^0(1+z)^4+\Omega_{\rm DE}^0\exp\left\{3\int_0^z\frac{1+w(z')}{1+z'}dz'\right\}}\\
&\stackrel{*}{=} H_0\sqrt{\Omega_m^0(1+z)^3+\Omega_\gamma^0(1+z)^4+\Omega_{\rm DE}^0(1+z)^{3(1+w_0+w_a)}\exp\left\{-3w_a\frac{z}{1+z}\right\}}\\
\mathcal{H}(z) &= a(z)H(z)\\
\dot{\mathcal{H}}(z) &= -\frac{H_0^2}{2(1+z)^2}\left((2(1+z)\Omega_\gamma^0+\Omega_m^0)(1+z)^3+(1+3w(z))\Omega_{\rm DE}^0\exp\left\{3\int_0^z\frac{1+w(z')}{1+z'}dz'\right\}\right)\\
&\stackrel{*}{=} -\frac{H_0^2}{2}\left(\Omega_{\rm DE}e^{-3w_a\frac{z}{z+1}}(w_az+3w_0(z+1)+z+1)(z+1)^{3(w_a+w_0)}+(z+1)(\Omega_m+2\Omega_\gamma(z+1))\right)\\
D_1(z) &\Rightarrow D''+\frac{3}{2}\left[1-\frac{w(a)}{1+X(a)}\right]\frac{D'}{a}-\frac{3}{2}\frac{X(a)}{1+X(a)}\frac{D}{a^2}=0;\quad X(a)=\frac{\Omega_m^0}{1-\Omega_m^0}e^{-3\int_a^1 d(ln\,a')w(a')}\\
g(z) &= (1+z)D(z)\\
f(z) &= \frac{a(z)}{D(z)}\frac{dD}{da}\\
\chi(z) &= \frac{1}{H_0}\int\frac{dz}{\sqrt{\Omega_m^0(1+z)^3+\Omega_\gamma^0(1+z)^4+\Omega_{\rm DE}^0\exp\left\{3\int_0^z\frac{1+w(z')}{1+z'}dz'\right\}}}\qquad\text{(in units Mpc/h)}\\
r_p(z) &= a(z)\chi(z)\quad\text{(in units Mpc/h)}\\
G(z) &= \frac{\dot{\mathcal{H}}}{\mathcal{H}^2}+\frac{2-5s}{\chi\mathcal{H}}+5s-f_{\rm evo}
\end{aligned}
$$

where $\stackrel{*}{=}$ 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 sav-

ing the input. The only solution is to upgrade to the most recent one. Future versions of our code may directly save the file using the standard library to avoid this issue.

**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 `-DHAVE_CUBA` when running `make`, include the library with `-lcuba -lm`, and specify the path to the library and header file `cuba.h`.

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

Below is the software used to build COFFE:

- Linux 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86_64 GNU/Linux
- gcc (Debian 6.3.0-18+deb9u1) 6.3.0 20170516
- autoconf (GNU Autoconf) 2.69
- automake (GNU automake) 1.15
- `FFTW` 3.3.7
- `libconfig` 1.5.0
- `GSL` 2.3
- `CUBA` 4.2

We recommend using a system as close as possible to the above when building from source to ensure proper functionality. Alternatively, you may opt for using the `Docker` version instead.

# 4 The `functions.c` list

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

$$
\begin{aligned}
X_0^0\big|_{\mathrm{den}} &= b_1 b_2\,, \\
X_0^0\big|_{\mathrm{rsd}} &= f_1 f_2 \frac{1+2\cos^2\theta}{15}\,, \\
X_2^0\big|_{\mathrm{rsd}} &= -\frac{f_1 f_2}{21}\left[1 + 11\cos^2\theta + \frac{18\cos\theta(\cos^2\theta - 1)\chi_1\chi_2}{r^2}\right]\,,
\end{aligned}
$$

$$X_4^0\big|_{\rm rsd} = f_1 f_2 \left[\frac{4(3\cos^2\theta-1)(\chi_1^4+\chi_2^4)}{35r^4} + \chi_1\chi_2(3+\cos^2\theta)\frac{3(3+\cos^2\theta)\chi_1\chi_2 - 8(\chi_1^2+\chi_2^2)\cos\theta}{35r^4}\right] ,$$

$$X_0^2\big|_{\rm d1} = \mathcal{H}_1\mathcal{H}_2 f_1 f_2 G_1 G_2 \frac{r^2\cos\theta}{3} ,$$

$$X_2^2\big|_{\rm d1} = -\mathcal{H}_1\mathcal{H}_2 f_1 f_2 G_1 G_2 \left((\chi_2-\chi_1\cos\theta)(\chi_1-\chi_2\cos\theta) + \frac{r^2\cos\theta}{3}\right) ,$$

$$X_0^4\big|_{\rm d2} = (3-f_{\rm evo1})(3-f_{\rm evo2})\, r^4\mathcal{H}_1^2\mathcal{H}_2^2 f_1 f_2 ,$$

$$X_0^4\big|_{\rm g1} = \frac{9\, r^4\Omega_m^2}{4a_1a_2}(1+G_1)(1+G_2)\mathcal{H}_0^4 ,$$

$$X_0^4\big|_{\rm g2} = \frac{9\, r^4\Omega_m^2}{4a_1a_2}(5s_1-2)(5s_2-2)\mathcal{H}_0^4 ,$$

$$X_0^4\big|_{\rm g3} = \frac{9\, r^4\Omega_m^2}{4a_1a_2}(f_1-1)(f_2-1)\mathcal{H}_0^4$$

$$X_0^0\big|_{\rm den\text{-}rsd} = \frac{b_1 f_2}{3} ,$$

$$X_2^0\big|_{\rm den\text{-}rsd} = -b_1 f_2 \left(\frac{2}{3} - (1-\cos^2\theta)\frac{\chi_1^2}{r^2}\right) ,$$

$$X_1^1\big|_{\rm den\text{-}d1} = -b_1 f_2 \mathcal{H}_2 G_2 (\chi_1\cos\theta - \chi_2) ,$$

$$X_0^2\big|_{\rm den\text{-}d2} = (3-f_{\rm evo2})\, r^2 b_1 f_2 \mathcal{H}_2^2 ,$$

$$X_0^2\big|_{\rm den\text{-}g1} = -b_1\frac{3\Omega_m}{2a_2}(1+G_2)r^2\mathcal{H}_0^2 ,$$

$$X_0^2\big|_{\rm den\text{-}g2} = -b_1\frac{3\Omega_m}{2a_2}(5s_2-2)r^2\mathcal{H}_0^2 ,$$

$$X_0^2\big|_{\rm den\text{-}g3} = -b_1\frac{3\Omega_m}{2a_2}(f_2-1)r^2\mathcal{H}_0^2 ,$$

$$X_1^1\big|_{\rm rsd\text{-}d1} = f_1 f_2 \mathcal{H}_2 G_2 \frac{(1+2\cos^2\theta)\chi_2 - 3\chi_1\cos\theta}{5} ,$$

$$X_3^1\big|_{\rm rsd\text{-}d1} = f_1 f_2 \mathcal{H}_2 G_2 \frac{(1-3\cos\theta)\chi_2^3 + \cos\theta(5+\cos^2\theta)\chi_2^2\chi_1 - 2(2+\cos\theta^2)\chi_2\chi_1^2 + 2\chi_1^3\cos\theta}{5r^2} ,$$

$$X_0^2\big|_{\rm rsd\text{-}d2} = \frac{3-f_{\rm evo2}}{3} f_1 f_2 r^2\mathcal{H}_2^2 ,$$

$$X_2^2\big|_{\rm rsd\text{-}d2} = -(3-f_{\rm evo2}) f_1 f_2 \mathcal{H}_2^2 \left(\frac{2}{3}r^2 - (1-\cos^2\theta)\chi_2^2\right) ,$$

$$X_0^2\big|_{\rm rsd\text{-}g1} = -\frac{\Omega_m}{2a_2} f_1(1+G_2)r^2\mathcal{H}_0^2 ,$$

$$X_2^2\big|_{\rm rsd\text{-}g1} = \frac{3\Omega_m}{2a_2} f_1(1+G_2)\mathcal{H}_0^2 \left(\frac{2}{3}r^2 - (1-\cos^2\theta)\chi_2^2\right) ,$$

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

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 $Z_\ell^n$ is given:

$$
\begin{aligned}
Z\big|_{\text{len}} = \frac{9\Omega_m^2}{4}\mathcal{H}_0^4(2-5s_1)(2-5s_2)\int_0^1 dx_1 \int_0^1 dx_2 \frac{(1-x_1)(1-x_2)}{x_1x_2}\frac{D_1(\lambda)D_1(\lambda')}{a(\lambda)a(\lambda')}\Bigg\{\frac{2}{5}(\cos^2\theta-1)\lambda^2\lambda'^2 I_0^0(r) \\
+ \frac{4r^2\cos\theta\lambda\lambda'}{3}I_0^2(r) + \frac{4\cos\theta\lambda\lambda'(r^2+6\cos\theta\lambda\lambda')}{15}I_1^1(r) + \frac{2(\cos^2\theta-1)\lambda^2\lambda'^2(2r^2+3\cos\theta\lambda\lambda')}{7r^2}I_2^0(r)
\end{aligned}
$$

$$
+\frac{2\cos\theta\lambda\lambda'\left(2r^4+12\cos\theta r^2\lambda\lambda'+15(\cos^2\theta-1)\lambda^2\lambda'^2\right)}{15r^2}I_3^1(r)
$$

$$
\left.+\frac{(\cos^2\theta-1)\lambda^2\lambda'^2\left(6r^4+30\cos\theta r^2\lambda\lambda'+35(\cos^2\theta-1)\lambda^2\lambda'^2\right)}{35r^4}I_4^0(r)\right\},
$$

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

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

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

$$
Z\big|_{\rm rsd\text{-}len}=\frac{3\Omega_m}{2}f_1\chi_2\mathcal{H}_0^2(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\left\{\frac{1}{15}(\lambda-6\chi_1\cos\theta+3\lambda\cos2\theta)I_0^0(r)\right.
$$

$$
-\frac{6\chi_1^3\cos\theta-\chi_1^2\lambda\left(9\cos^2\theta+11\right)+\chi_1\lambda^2\cos\theta(3\cos2\theta+19)-2\lambda^3(3\cos2\theta+1)}{21r^2}I_2^0(r)
$$

$$
-\frac{1}{35r^4}\Big[-4\chi_1^5\cos\theta-\chi_1^3\lambda^2\cos\theta(\cos2\theta+7)+\chi_1^2\lambda^3\left(\cos^4\theta+12\cos^2\theta-21\right)
$$

$$
\left.-3\chi_1\lambda^4\cos\theta(\cos2\theta-5)-\lambda^5(3\cos2\theta+1)+12\chi_1^4\lambda\Big]I_4^0(r)\right\},
$$

$$
Z\big|_{\rm d1\text{-}len}=\frac{3\Omega_m}{2}\chi_2\mathcal{H}_0^2\mathcal{H}_1f_1G_1(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\left\{\frac{2}{15}\Big(\cos\theta\left(\lambda^2-2\chi_1^2\right)\right.
$$

$$
+\chi_1\lambda(2\cos2\theta-1)\Big)I_1^1(r)+\frac{2}{3}r^2\cos\theta I_0^2(r)
$$

$$
\left.-\frac{4\chi_1^4\cos\theta-\chi_1^3\lambda\left(\cos^2\theta+9\right)+\chi_1^2\lambda^2\cos\theta\left(\cos^2\theta+5\right)-2\chi_1\lambda^3(\cos2\theta-2)-2\lambda^4\cos\theta}{15r^2}I_3^1(r)\right\},
$$

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

$$
\left.-\chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\right\},
$$

$$
Z\big|_{\rm g1\text{-}len}=\frac{9\Omega_m^2}{4}\chi_2(1+G_1)\mathcal{H}_0^4(2-5s_2)D_1(z_1)\int_0^1 dx(1-x)\frac{D_1(\lambda)}{a(\lambda)}\left\{2\chi_1r^2\cos\theta I_1^3(r)\right.
$$

$$- \chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

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

$$- \chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

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

$$- \chi_1^2\lambda(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

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

$$- \lambda^2\lambda'^2(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

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

$$- \lambda^2\lambda'^2(1-\cos^2\theta)I_2^2(r)\Big\}\,,$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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