CMBXC Likelihood: Status Report

Massimiliano Lattanzi - INFN Ferrara

(on behalf of the CMBXC likelihood group)

CMBXC Meeting
March 19th, 2020

CMBXC Likelihood – people

Bologna

- A. Gruppuso
- N. Mauri
- L. Patrizii
- G. Sirri
- M. Tenti

Ferrara

- S. Alvi
- M. Gerbino
- M. Lattanzi
- P. Natoli
- L. Pagano

Milano

- M. Archidiacono

Padova

- S. Dusini
- A. Renzi
- C. Sirignano
- L. Stanco
- G. Verza

Roma

- M. Migliaccio

Work Packages

WP1: Simulations for CMBX

WP1.1 ray tracing for CMB lensing and for ISW

WP1.2 Nbody simulations

WP1.3 Monte-Carlo and/or Correlated sims for covariance

WP2: Integrated Sachs-Wolfe effect

WP2.1 ISW with Xcorel galaxies

WP2.2 ISW with stacking of voids/super-cluster

WP2.3 ISW reconstruction

WP3: Correlation of CMB lensing with LSS Tracers

WP3.1 galaxy-CMB lensing

WP3.2 galaxy lensing-CMB lensing

WP4: Correlation of tSZ with LSS Tracers

WP4.1 tSZ Xcorrel with galaxy/shear and (super) clusters

WP4.2 kSZ and bulk flows with Xcorrel with spectro galaxies

WP5: Correlation of CIB with LSS Tracers

WP5.1 CIB Xcorrelation with galaxy distribution

WP5.2 CIB Xcorrelation with galaxy lensing

WP6: Estimators & Likelihood

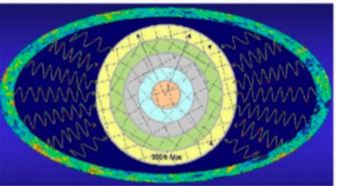
WP6.1 Pseudo-Cl estimators

WP6.2 Hybrid estimators

+ potentially other estimators

WP6.3 Combined Xcorrelation likelihoods from WP2, 3, 4, 5

WP6.4 Data processing validation



CMBXC Estimators & Likelihood

Two main (related but independent) activities

- Development and validation of spectral estimators
- Development and validation of a likelihood package (report in this talk)

CMBXC Likelihood - Activities

- Aim: build, test and validate a likelihood approach for Euclid CMBXC (focusing on ISW for the moment)
- Joint activity in this respect between the Bologna, Ferrara, Padova groups started in 2019 with regular meetings to start development of the likelihood code
- Plans for 2020: two meetings/month, one remote and one face to face (this is obviously going to change, at least for the near future….)
- More frequent tcons between restricted groups of people for more specific tasks (e.g. hands-on coding)
- Status report at the last CMBXC meeting in Bologna

CMBXC Likelihood - Activities

- Evaluation of available tools and agreement on a framework for the ISW likelihood – i.e., fixed some implementation choices:
- Harmonic-space likelihood
- Boltzmann code+MC engine: CLASS + MontePython (J. Lesgourgues)
- TG Xspectra to be calculated in the likelihood code (as opposed to the Boltzmann solver)

$$C_\ell^{cT} = \frac{3\Omega_m \mathrm{H}_0^2}{c^3\left(l+\frac{1}{2}\right)^2}\int dz\, b_c(z)\frac{dN}{dz}H(z)D(z)\frac{d}{dz}\left(\frac{D(z)}{a(z)}\right)$$

$$\times P\left(k=\frac{l+\frac{1}{2}}{\chi(z)}\right).$$

- (uses the Limber approximation for the moment)
- Exact likelihood (for the moment)

Exact likelihood

$$\mathcal{P}(\hat{\mathbf{C}}_\ell|\mathbf{C}_\ell) \propto \frac{\left|\hat{\mathbf{C}}_\ell\right|^{(\nu-p+1)/2} \exp\left[-\mathrm{Tr}\left(\nu\,\mathbf{C}_\ell^{-1}\hat{\mathbf{C}}_\ell/2\right)\right]}{|\mathbf{C}_\ell|^{\nu/2}}$$

$$\hat{\mathbf{C}}_\ell = \begin{pmatrix} \hat{C}_\ell^{TT} & \hat{C}_\ell^{TG_1} & \dots & \hat{C}_\ell^{TG_{n_\mathrm{b}}} \\ \hat{C}_\ell^{TG_1} & \hat{C}_\ell^{G_1G_1} & \dots & \hat{C}_\ell^{G_1G_{n_\mathrm{b}}} \\ \dots & \dots & \dots & \dots \\ \hat{C}_\ell^{TG_{n_\mathrm{b}}} & \hat{C}_\ell^{G_1G_{n_\mathrm{b}}} & \dots & \hat{C}_\ell^{G_{n_\mathrm{b}}G_{n_\mathrm{b}}} \end{pmatrix}$$

$$\mathbf{C}_\ell = \begin{pmatrix} C_\ell^{TT} & C_\ell^{TG_1} & \dots & C_\ell^{TG_{n_\mathrm{b}}} \\ C_\ell^{TG_1} & C_\ell^{G_1G_1} & \dots & C_\ell^{G_1G_{n_\mathrm{b}}} \\ \dots & \dots & \dots & \dots \\ C_\ell^{TG_{n_\mathrm{b}}} & C_\ell^{G_1G_{n_\mathrm{b}}} & \dots & C_\ell^{G_{n_\mathrm{b}}G_{n_\mathrm{b}}} \end{pmatrix}$$

$$\nu = 2\ell + 1$$

$$p = n_{\mathrm{bin}} + 1$$

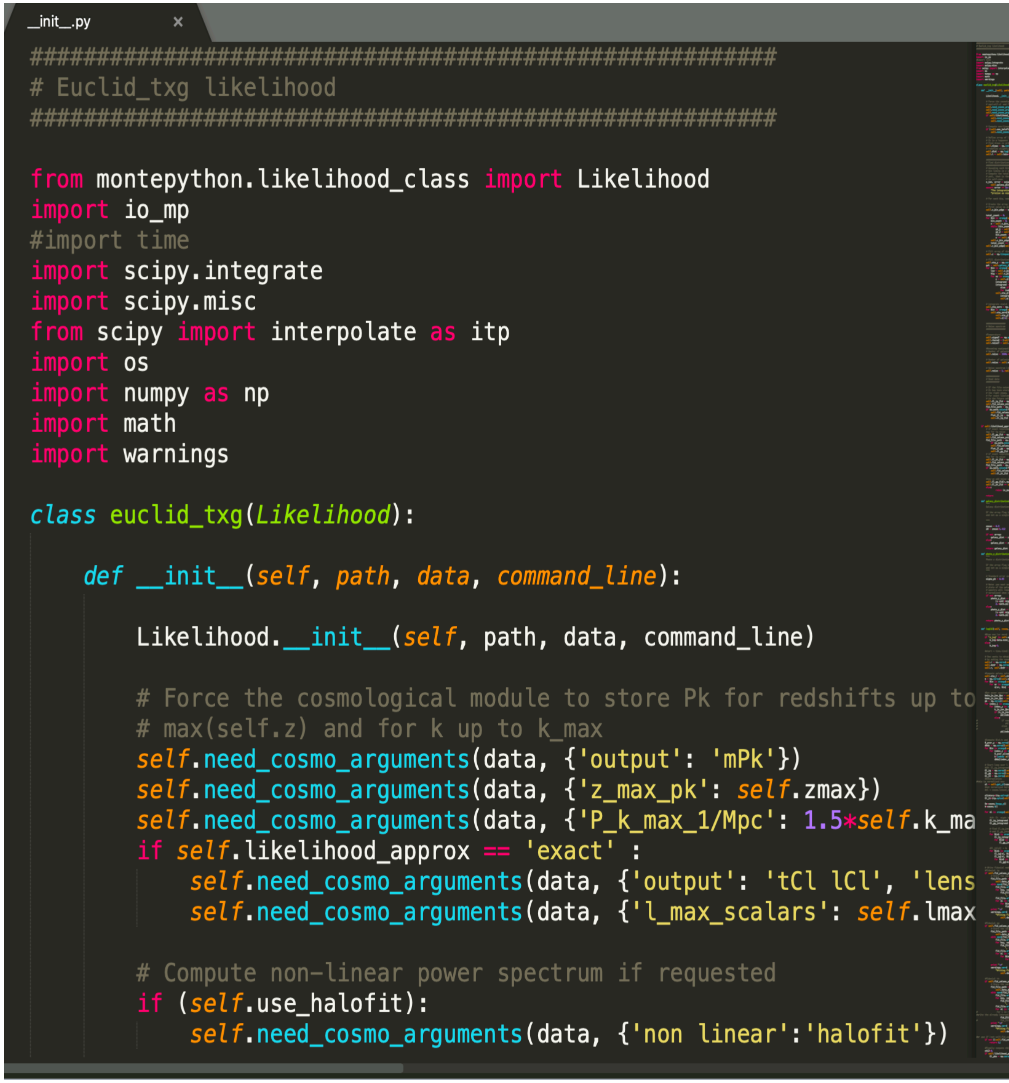
Current Status

First milestone: working version of a MontePython module for the TXG likelihood

- Allows for tomographic analysis
- Accuracy in the sampling can be changed by the user
- Uses Limber
- Exact likelihood

Code is available on INFN GitLab:

https://baltig.infn.it/paganol/euclid_txg.git

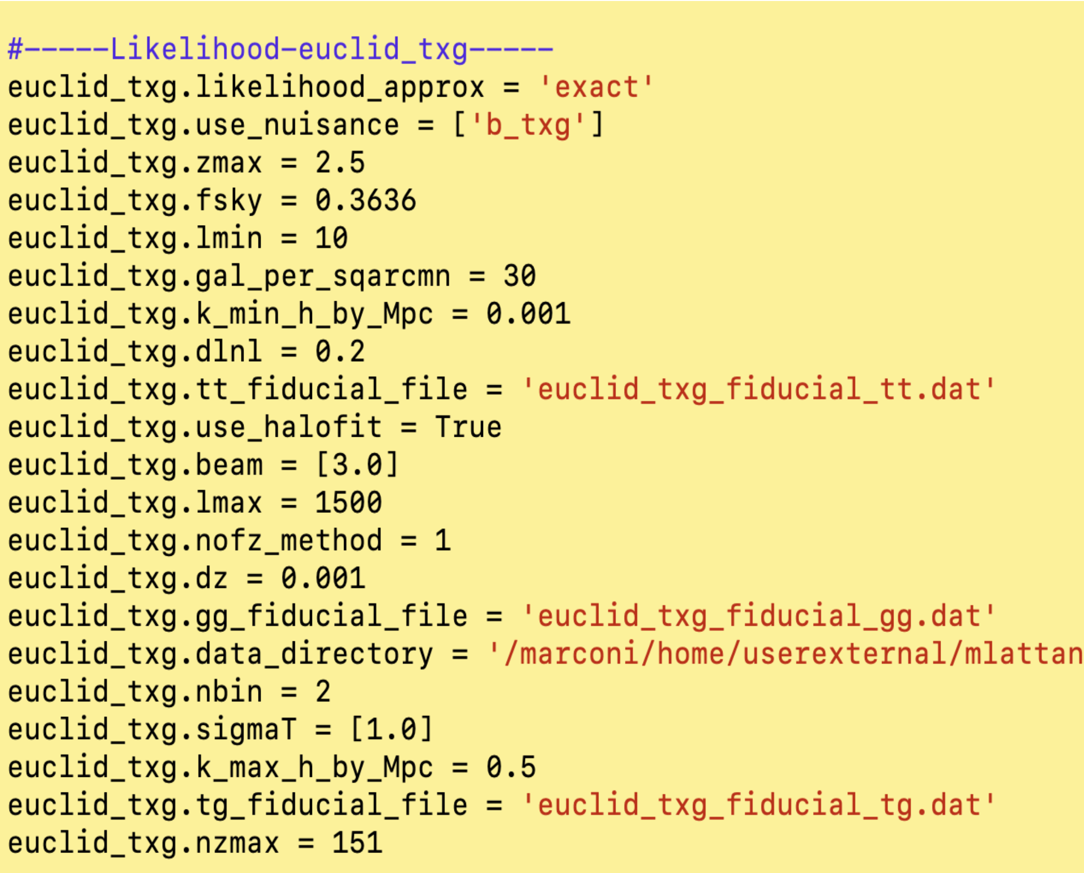


```
########################################################
# Euclid_txg likelihood
########################################################

from montepython.likelihood_class import Likelihood
import io_mp
#import time
import scipy.integrate
import scipy.misc
from scipy import interpolate as itp
import os
import numpy as np
import math
import warnings

class euclid_txg(Likelihood):

    def __init__(self, path, data, command_line):

        Likelihood.__init__(self, path, data, command_line)

        # Force the cosmological module to store Pk for redshifts up to
        # max(self.z) and for k up to k_max
        self.need_cosmo_arguments(data, {'output': 'mPk'})
        self.need_cosmo_arguments(data, {'z_max_pk': self.zmax})
        self.need_cosmo_arguments(data, {'P_k_max_1/Mpc': 1.5*self.k_ma
        if self.likelihood_approx == 'exact' :
            self.need_cosmo_arguments(data, {'output': 'tCl lCl', 'lens
            self.need_cosmo_arguments(data, {'l_max_scalars': self.lmax

        # Compute non-linear power spectrum if requested
        if (self.use_halofit):
            self.need_cosmo_arguments(data, {'non linear':'halofit'})
```

Current Status

First milestone: working version of a MontePython module for the TXG likelihood

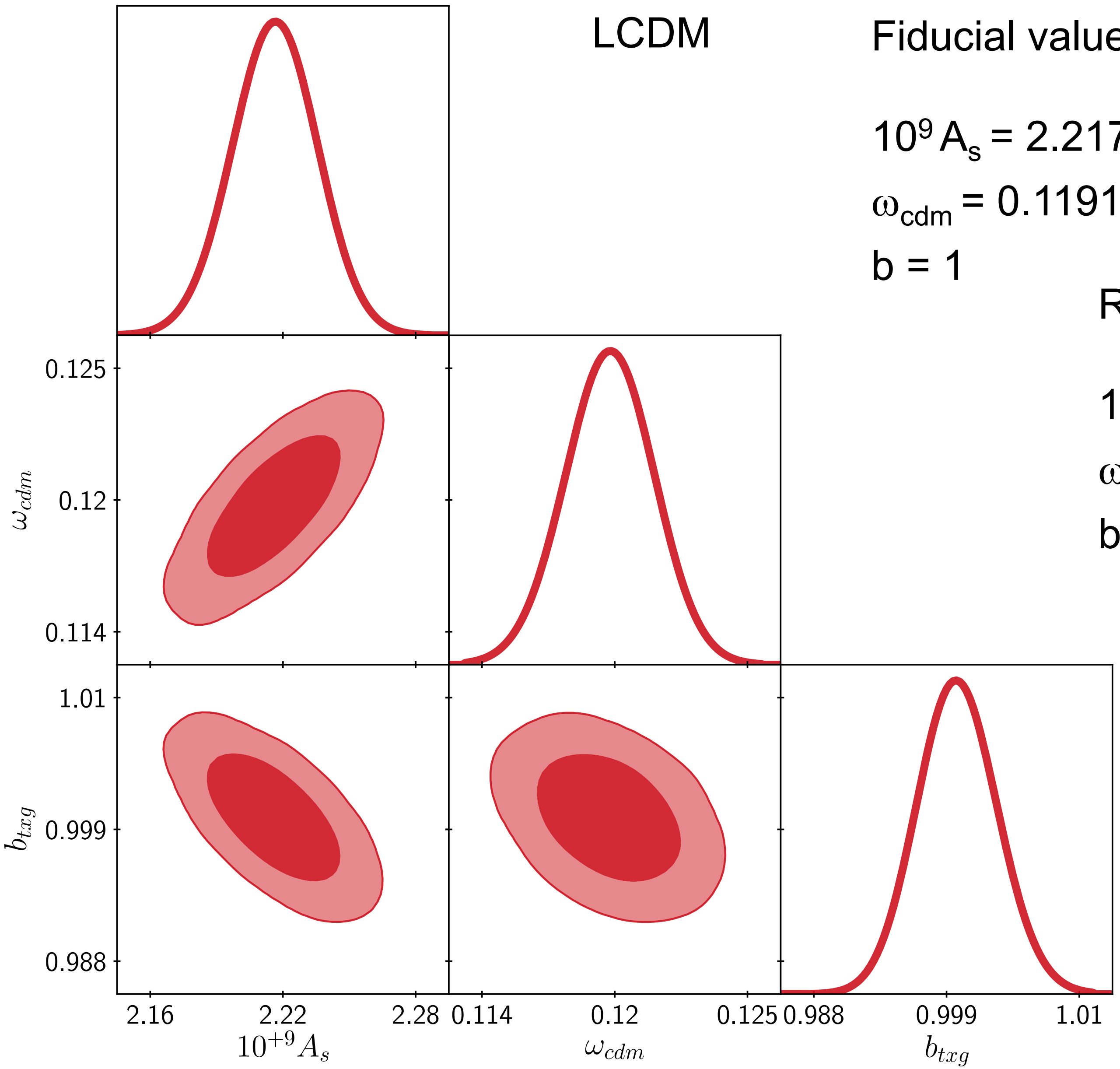
- Allows for tomographic analysis
- Accuracy in the sampling can be changed by the user
- Uses Limber
- Exact likelihood



```
#-----Likelihood-euclid_txg-----
euclid_txg.likelihood_approx = 'exact'
euclid_txg.use_nuisance = ['b_txg']
euclid_txg.zmax = 2.5
euclid_txg.fsky = 0.3636
euclid_txg.lmin = 10
euclid_txg.gal_per_sqarcmn = 30
euclid_txg.k_min_h_by_Mpc = 0.001
euclid_txg.dlnl = 0.2
euclid_txg.tt_fiducial_file = 'euclid_txg_fiducial_tt.dat'
euclid_txg.use_halofit = True
euclid_txg.beam = [3.0]
euclid_txg.lmax = 1500
euclid_txg.nofz_method = 1
euclid_txg.dz = 0.001
euclid_txg.gg_fiducial_file = 'euclid_txg_fiducial_gg.dat'
euclid_txg.data_directory = '/marconi/home/userexternal/mlattan
euclid_txg.nbin = 2
euclid_txg.sigmaT = [1.0]
euclid_txg.k_max_h_by_Mpc = 0.5
euclid_txg.tg_fiducial_file = 'euclid_txg_fiducial_tg.dat'
euclid_txg.nzmax = 151
```

Code is available on INFN GitLab:

https://baltig.infn.it/paganol/euclid_txg.git



LCDM

Fiducial values

$10^9 A_s = 2.2177$

$\omega_{cdm} = 0.11919$

b = 1

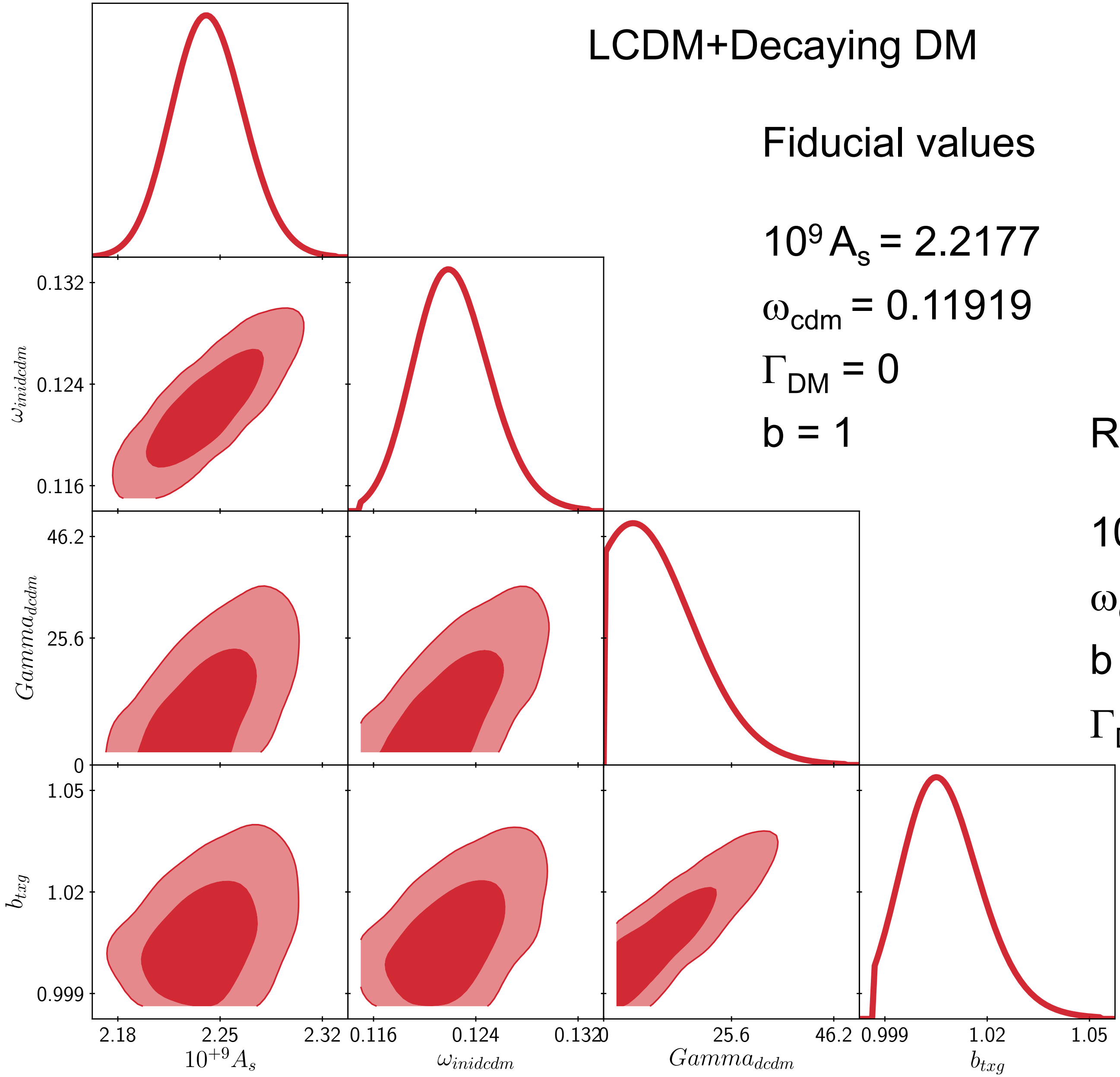
Recovered values

$10^9 A_s = 2.22 +/- 0.02$

$\omega_{cdm} = 0.119 +/- 0.002$

b = 1.000 +/- 0.003

LCDM+Decaying DM



Fiducial values

$10^9 A_s = 2.2177$

$\omega_{cdm} = 0.11919$

$\Gamma_{DM} = 0$

b = 1

Recovered values

$10^9 A_s = 2.24$ +/- 0.03

$\omega_{cdm} = 0.122$ +/- 0.003

b = 1.01 +/- 0.01

$\Gamma_{DM} < 15$ km s^{-1} Mpc^{-1}

CMBXC likelihood - Next steps and open issues

- Currently validating the code on w0-wa DE models.
- How to efficiently disentangle the contribution of TG from TT and GG?
- Go beyond Limber. Validate Limber approximation where used.
- Test and validate other likelihood approximations (e.g. Gaussian, Hamimeche-Lewis)

CMBXC likelihood - Next steps and open issues

- Is it really necessary to use QML for input spectra (except for the very lowest multipoles)?
- More realistic (Euclid-wise) outputs (e.g. covariances)
- Optimal binning and sampling
- Move to a more flexible framework (i.e., Cobaya)?