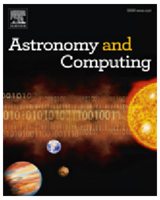
Astronomy and Computing 14 (2016) 35–42

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom



Full length article

CosmoBolognaLib: C++ libraries for cosmological calculations[☆]

F. Marulli[a,b,c,*], A. Veropalumbo[a], M. Moresco[a,b]

[a] Dipartimento di Fisica e Astronomia - Università di Bologna, viale Berti Pichat 6/2, I-40127 Bologna, Italy
[b] INAF - Osservatorio Astronomico di Bologna, via Ranzani 1, I-40127 Bologna, Italy
[c] INFN - Sezione di Bologna, viale Berti Pichat 6/2, I-40127 Bologna, Italy

ARTICLE INFO

Article history:
Received 10 November 2015
Accepted 24 January 2016
Available online 3 February 2016

Keywords:
Cosmology: theory
Cosmology: observations
Cosmology: large-scale structure of Universe
Methods: numerical
Methods: statistical

ABSTRACT

We present the CosmoBolognaLib, a large set of Open Source C++ numerical libraries for cosmological calculations. CosmoBolognaLib is a *living project* aimed at defining a common numerical environment for cosmological investigations of the large-scale structure of the Universe. In particular, one of the primary focuses of this software is to help in handling astronomical catalogues, both real and simulated, measuring one-point, two-point and three-point statistics in configuration space, and performing cosmological analyses. In this paper, we discuss the main features of this software, providing an overview of all the available C++ classes implemented up to now. Both the CosmoBolognaLib and their associated doxygen documentation can be freely downloaded at https://github.com/federicomarulli/CosmoBolognaLib. We provide also some examples to explain how these libraries can be included in either C++ or Python codes.

1. Introduction

Numerical tools for cosmological calculations are one of the crucial ingredients in the increasingly ambitious investigations of the large-scale structure of the Universe. Several public libraries for astronomical calculations are nowadays available, in different languages, such as e.g. CfunBASE (Taghizadeh-Popp, 2010), CosmoPMC (Kilbinger et al., 2011), AstroML (VanderPlas et al., 2012), CUTE (Alonso, 2012), Astropy (Astropy Collaboration et al., 2013), Cosmo++ (Aslanyan, 2014), CosmoloPy,[1] NumCosmo (Dias Pinto Vitenti and Penna-Lima, 2014), TreeCorr (Jarvis, 2015).

Aiming at defining a common environment for handling extragalactic source catalogues, performing statistical analyses and extracting cosmological constraints, we implemented a large set of C++ libraries, called CosmoBolognaLib (hereafter CBL), specifically focused on numerical computations for cosmology, thus complementing the available software. In particular, the CBL provide highly optimised algorithms to measure two-point (2PCF) and three-point correlation functions (3PCF), exploiting a specifically designed parallel chain-mesh algorithm to count pairs and triplets. Several types of correlation functions can be computed, such as the angle-averaged 2PCF, the 2D 2PCF in both Cartesian and polar coordinates and its multipole moments, the angular, projected and deprojected 2PCF, the clustering wedges, the filtered 2PCF, and the connected and reduced 3PCF (see Section 4.1). Moreover, a large set of methods are provided to construct random catalogues, to estimate errors and to extract cosmological constraints from clustering analyses (see Section 4.3). These features represent the main novelty of the presented libraries.

The CBL are fully written in C++. They can be included either in C++ codes or, alternatively, in high-level scripting languages through wrapping. We provide an example code that shows how to include the CBL in Python scripts in Appendix B.3.

This effort can be considered as a *living project*, started a few years ago and intended to be continued in the forthcoming years. The following is the list of scientific publications that have been fully or partially performed using the presented libraries: Marulli et al. (2011, 2012a,b, 2013, 2015), Giocoli et al. (2013), Villaescusa-Navarro et al. (2014), Moresco et al. (2014), Veropalumbo et al. (2014, 2015), Sereno et al. (2015), Moresco et al. (in preparation) and Petracca et al. (2015). Thanks mainly to the adopted object-oriented programming technique, the CBL are flexible enough to be significantly extended. In this paper, we present the main features of the current version of the CBL, that is fully publicly

---

[☆] This code is registered at the ASCL with the code entry ascl:1511.019.

[*] Corresponding author at: Dipartimento di Fisica e Astronomia - Università di Bologna, viale Berti Pichat 6/2, I-40127 Bologna, Italy.
E-mail address: federico.marulli3@unibo.it (F. Marulli).
URL: https://www.unibo.it/sitoweb/federico.marulli3 (F. Marulli).

[1] http://roban.github.com/CosmoloPy/.

http://dx.doi.org/10.1016/j.ascom.2016.01.005
2213-1337/

available, [2] together with the documentation obtained with doxygen.[3] A set of sample codes, that explain how to use these libraries in either C++ or Python software, is provided at the same webpage.

The paper is organised as follows. In Section 2 we describe the CBL class for cosmological computations. In Section 3 we present the classes implemented for handling catalogues of extragalactic sources. 2PCF and 3PCF can be measured and modelled with specific classes that are described in Section 4. Section 5 presents the CBL methods for statistical analyses. In Section 6 we provide a brief description of the other CBL functions used for several generic calculations. Finally, in Section 7 we draw our conclusions. Compiling instructions and a few sample codes are reported in Appendices A and B, respectively.

2. Cosmology

All the cosmological functions defined in the CBL are implemented as public members of the class cosmobl::Cosmology.[4] The private parameters of this class are the following: the matter density, that is the sum of the density of baryons, cold dark matter and massive neutrinos (in units of the critical density) at $z = 0$, $\Omega_{\rm matter}$; the density of baryons at $z = 0$, $\Omega_{\rm baryon}$; the density of massive neutrinos at $z = 0$, Ω_ν; the effective number of relativistic degrees of freedom, $N_{\rm eff}$; the number of massive neutrino species; the density of dark energy at $z = 0$, $\Omega_{\rm DE}$; the density of radiation at $z = 0$, $\Omega_{\rm radiation}$; the Hubble parameter, $h = H_0/100$; the initial scalar amplitude of the power spectrum, A_s; the primordial spectral index, $n_{\rm spec}$; the two parameters of the dark energy equation of state in the Chevallier–Polarski–Linder parameterisation (Chevallier and Polarski, 2001; Linder, 2003), w_0 and w_a; the non-Gaussian amplitude, $f_{\rm NL}$; the non-Gaussian shape—local, equilateral, enfolded, orthogonal (Fedeli et al., 2011); the model used to compute distances (used only for some specific interacting dark energy models, see Marulli et al. 2012a); a variable called *unit*, used to choose between physical units or cosmological units (that is in unit of h). If the above parameters are not specified when creating an object of this class, default values from Planck cosmology will be used (Planck Collaboration et al., 2014). In any case, each cosmological parameter can be set individually, when required.

Once the cosmological model has been chosen by setting the parameters described above, a large set of cosmological functions can then be used. We provide here a brief overview of the main functions of the class. The full explanation of the whole set of class members can be found in the doxygen documentation at the CBL webpage.

Several functions are available to estimate the redshift evolution of all the relevant cosmological parameters, to compute the lookback and cosmic times, to estimate cosmological distances and volumes, and to convert redshifts into comoving distances and vice versa. There are methods to estimate the number density and mass function of dark matter haloes. Specifically, the code implements the following equation (see e.g. Marulli et al., 2011):

$$\frac{MdM}{\bar\rho}\frac{dn(M,z)}{dM} = \zeta f(\zeta)\frac{d\zeta}{\zeta}, \tag{1}$$

with $\zeta \equiv [\delta_{\rm sc}(z)/\sigma(M)]^2$, where $\delta_{\rm sc}(z)$ is the overdensity required for spherical collapse at z, $\bar\rho = \Omega_{\rm matter}\rho_c$, ρ_c is the critical density of the Universe, and $dn(M,z)$ is the halo number density in the mass interval M to $M + dM$. The variance of the linear density field is given by

$$\sigma^2(M) = \int dk \frac{k^2 P_{\rm lin}(k)}{2\pi^2}|W(kR)|^2, \tag{2}$$

where the top-hat window function is $W(x) = (3/x^3)(\sin x - x\cos x)$, with $R = (3M/4\pi\bar\rho)^{1/3}$. At the moment, the implemented mass function models are the following: Press and Schechter (1974), Sheth and Tormen (1999), Jenkins et al. (2001), Warren et al. (2006), Shen et al. (2006), Reed et al. (2007), Pan (2007), Tinker et al. (2008) and Angulo et al. (2012).

Methods to estimate the effective linear bias of dark matter haloes are provided as well. The effective bias is computed through the following integral:

$$b(z) = \frac{\int_{M_{\rm min}}^{M_{\rm max}} n(M,z)b(M,z)dM}{\int_{M_{\rm min}}^{M_{\rm max}} n(M,z)dM}, \tag{3}$$

where $b(M,z)$ is the linear bias and $n(M,z)$ is the halo number density. The available parameterisations are: Sheth and Tormen (1999), Sheth et al. (2001) and Tinker et al. (2010).

A large set of functions is provided to estimate the real-space and redshift-space power spectra and 2PCF (see Appendix B.1), and to assess the cosmic mass accretion history (Giocoli et al., 2013). To estimate the dark matter power spectrum and all the derived quantities, such as the mass variance used to compute the mass function and bias, the user can choose between one of the following external codes: CAMB (Lewis et al., 2000), MPTbreeze (Crocce et al., 2012), CLASS (Lesgourgues, 2011; Blas et al., 2011), Eisenstein&Hu code (Eisenstein and Hu, 1998, 1999). The latter will be exploited automatically by the CBL via specific functions used to set the parameter files conveniently.

3. Catalogues

The class cosmobl::Catalogue is used to handle samples of astronomical objects. The present version of the CBL provides specific classes for galaxies, clusters of galaxies, dark matter haloes and generic mock objects. However, the code structure is sufficiently versatile to easily include new objects or to extend the present ones, e.g. by adding new properties. Once the catalogue is created, several operations can be performed, such as estimating the distribution of any property of the object members, dividing the catalogues in sub-samples, or creating a smoothed version of the original catalogue. Moreover, a catalogue can be passed to other objects as an input, e.g. to estimate 2PCF and 3PCF (see Section 4.1), or to assess errors through the jackknife or bootstrap techniques (see Section 4.2). Catalogues can also be added together, or they can be enlarged by adding new single objects.

For a fast spatial search of objects in the catalogues, we implemented a highly optimised chain-mesh method, specifically designed for counting object pairs and triplets in a specified range of scales. The algorithm implements a pixelization scheme, similar to the one described in Alonso (2012). First, the catalogue is divided into cubic cells, and the indexes of all the objects in each cell are stored in vectors. Then, to find all the objects close to a given one, the search is performed only on the cells in the chosen scale range, thus minimising the amount of useless counts of objects at too large separations. In this way, the efficiency of the method depends primarily on the ratio between the scale range of the searching region and the maximum separation between the objects in the catalogue. This is particularly useful when measuring 2PCF and 3PCF (see Section 4). For alternative searching algorithms, such as kd-tree and ball-tree methods, see e.g. Jarvis (2015).

The chain-mesh method is implemented in the four classes: cosmobl::ChainMesh, cosmobl::ChainMesh1D,

---

[2] https://github.com/federicomarulli/CosmoBolognaLib and http://apps.difa.unibo.it/files/people/federico.marulli3.

[3] www.doxygen.org.

[4] cosmobl is the global namespace of the CBL.