Astronomy and Computing 25 (2018) 38–43

Contents lists available at ScienceDirect

Astronomy and Computing

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

Full length article

PyCosmo: An integrated cosmological Boltzmann solver

A. Refregier [a,*], L. Gamper [a], A. Amara [a], L. Heisenberg [b]

[a] Department of Physics, ETH Zurich, Wolfgang Pauli Strasse 27, 8093 Zurich, Switzerland
[b] Institute for Theoretical Studies, ETH Zurich, Clausiusstrasse 47, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 16 August 2017
Accepted 4 August 2018
Available online 13 August 2018

Keywords:
Cosmology
Boltzmann equation
Differential equations
Python

ABSTRACT

As wide-field surveys yield ever more precise measurements, cosmology has entered a phase of high precision requiring highly accurate and fast theoretical predictions. At the heart of most cosmological model predictions is a numerical solution of the Einstein–Boltzmann equations governing the evolution of linear perturbations in the Universe. We present PyCosmo, a new Python-based framework to solve this set of equations using a special purpose solver based on symbolic manipulations, automatic generation of C++ code and sparsity optimisation. The code uses a consistency relation of the field equations to adapt the time step and does not rely on physical approximations for speed-up. After reviewing the system of first-order linear homogeneous differential equations to be solved, we describe the numerical scheme implemented in PyCosmo. We then compare the predictions and performance of the code for the computation of the transfer functions of cosmological perturbations and compare it to existing cosmological Boltzmann codes. While PyCosmo does not yet have all the features of other codes, our approach is complementary to other fast cosmological Boltzmann solvers and can be used as an independent test of their numerical solutions. The symbolic representation of the Einstein–Boltzmann equation system in PyCosmo provides a convenient interface for implementing extended cosmological models. We also discuss how the PyCosmo framework can also be used as a general framework to compute cosmological quantities as well as observables for both interactive and high-performance batch jobs applications. Information about the PyCosmo package and future code releases are available at http://www.cosmology.ethz.ch/research/software-lab.html.

1. Introduction

In order to address the fundamental questions raised by the nature of Dark Matter, Dark Energy and large scale gravity, a number of cosmological experiments are currently underway or in the planning (e.g. DES,[1] DESI,[2] LSST,[3] Euclid,[4] WFIRST[5]). These experiments aim to achieve the high precision required to tackle these questions and will thus require highly accurate predictions of observables for a wide set of cosmological models. At the heart of most cosmological model prediction is a numerical solution of the Einstein–Boltzmann equations (see Ma and Bertschinger, 1995 and reference therein) governing the linear evolution of perturbations in the Universe. Several codes have thus been developed to produce fast and accurate solutions to this set of first-order linear homogeneous differential equations, such as COSMICS (Bertschinger, 1995a), CMBFAST (Seljak and Zaldarriaga, 1996), CMBEASY (Doran, 2005), CAMB (Lewis et al., 2000), CLASS (Lesgourgues, 2011a; Blas et al., 2011; Lesgourgues, 2011b), with only the latter two being currently maintained. The predictions of these codes are then compared to measurements from cosmological surveys to derive constrains on the parameters of the cosmological model using Monte-Carlo Markov Chain techniques (e.g. Lewis and Bridle, 2002; Akeret et al., 2012). Given the central importance of Boltzmann codes to our current constraints on the cosmological model and the well known numerical difficulty to solve this set of equations (see e.g. Nadkarni-Ghosh and Refregier, 2016 for a mathematical analysis, and references therein), it is important to explore different numerical schemes for the solutions of the differential equations.

In this paper, we present PyCosmo, a new Python-based framework to solve the Einstein–Boltzmann equations using a special purpose solver based on symbolic manipulations, automatic generation of C++ code and sparsity optimisation. The code uses a consistency relation of the field equations to adapt the time step and does not rely on physical approximations for speed-up. We study the accuracy and performance of PyCosmo for the

* Corresponding author.
E-mail address: alexandre.refregier@phys.ethz.ch (A. Refregier).

1 http://www.darkenergysurvey.org.
2 http://desi.lbl.gov.
3 http://www.lsst.org.
4 http://sci.esa.int/euclid/.
5 http://wfirst.gsfc.nasa.gov.

https://doi.org/10.1016/j.ascom.2018.08.001
2213-1337/

computation of the transfer functions of cosmological perturbations in the newtonian gauge in a flat universe, and compare it to existing codes. Our approach is complementary to existing cosmological Boltzmann solvers that are based on more general differential equation solvers and physical approximations, and can be used as an independent test of their numerical solutions. We discuss how the symbolic representation of the Einstein–Boltzmann equation system in PyCosmo provides a convenient interface for theorists to rapidly implement new cosmological models. We also discuss how the PyCosmo framework can also be used as a general framework to compute cosmological quantities as well as observables for both interactive and for high-performance batch jobs applications, drawing upon the earlier IDL cosmological package ICosmo (Refregier et al., 2011).

The paper is organised as follows. In Section 2, we describe the set of Einstein–Boltzmann equations describing the linear growth of cosmological structures. Section 3 describes our implementation scheme for deriving numerical solutions to this set of equations. In Section 4 we study the performance of PyCosmo in terms of numerical precision and speed and compare it to existing codes. Our conclusions are described in Section 5.

2. Einstein–Boltzmann equations

2.1. Linear perturbations

After solving the evolution of the scale factor a and the Hubble parameter H using the Friedmann equation (see e.g. Dodelson, 2003), we can consider the linear evolution of scalar perturbations. For this purpose, we choose the newtonian gauge in a flat ΛCDM cosmology. The evolution of perturbations are thus governed by the Einstein–Boltzmann equations which, in this case and to linear order, are given by (Ma and Bertschinger, 1995 with the conventions of Dodelson, 2003)

$$\dot{\delta} = -ku - 3\dot{\Phi} \tag{1}$$

$$\dot{u} = -\frac{\dot{a}}{a}u + k\Psi \tag{2}$$

$$\dot{\delta_b} = -ku_b - 3\dot{\Phi} \tag{3}$$

$$\dot{u_b} = -\frac{\dot{a}}{a}u_b + k\Psi + kc_s^2\delta_b + \frac{\dot{\tau}}{R}[u_b - 3\Theta_1] \tag{4}$$

$$\dot{\Theta}_0 = -k\Theta_1 - \dot{\Phi} \tag{5}$$

$$\dot{\Theta}_1 = \frac{k}{3}[\Theta_0 - 2\Theta_2 + \Psi] + \dot{\tau}\left[\Theta_1 - \frac{u_b}{3}\right] \tag{6}$$

$$\dot{\Theta}_2 = \frac{k}{5}[2\Theta_1 - 3\Theta_3] + \dot{\tau}\left[\Theta_2 - \frac{\Pi}{10}\right] \tag{7}$$

$$\dot{\Theta}_l = \frac{k}{2l+1}[l\Theta_{l-1} - (l+1)\Theta_{l+1}] + \dot{\tau}\Theta_l \quad , l > 2 \tag{8}$$

$$\dot{\Theta}_{Pl} = \frac{k}{2l+1}\left[l\Theta_{P(l-1)} - (l+1)\Theta_{P(l+1)}\right] + \dot{\tau}\left[\Theta_{Pl} - \frac{\Pi}{2}(\delta_{l,0} + \frac{\delta_{l,2}}{5})\right] \tag{9}$$

$$\dot{\mathcal{N}}_0 = -k\mathcal{N}_1 - \dot{\Phi} \tag{10}$$

$$\dot{\mathcal{N}}_1 = \frac{k}{3}[\mathcal{N}_0 - 2\mathcal{N}_2 + \Psi] \tag{11}$$

$$\dot{\mathcal{N}}_l = \frac{k}{2l+1}[l\mathcal{N}_{l-1} - (l+1)\mathcal{N}_{l+1}], \quad l > 1 \tag{12}$$

$$k^2\Phi + 3\frac{\dot{a}}{a}\left(\dot{\Phi} - \frac{\dot{a}}{a}\Psi\right) = 4\pi G a^2[\rho_m\delta_m + 4\rho_r\Theta_{r0}], \tag{13}$$

where dot denotes derivatives with respect to conformal time η, δ (δ_b) and u (u_b) are the density and velocity perturbations for the dark matter (baryons), Θ_l and Θ_{Pl} are the photon temperature and polarisation multipole moments, $\mathcal{N}_l$ are the multipole moments of the (massless) neutrino temperature, and $\Pi = \Theta_2 + \Theta_{P0} + \Theta_{P2}$. The baryon-to-photon ratio is given by $R = 3\rho_b/(4\rho_\gamma)$, τ is the Thomson scattering optical depth and c_s is the baryon sound speed. The subscripts r and m refer to the density-weighted sum of all radiation and matter components, and ρ_r and ρ_m are the mean density in each of these components.

The gravitational potential fields Φ and Ψ describe scalar perturbations to the metric $ds^2 = -(1+2\Psi)dt^2 + a^2(1+2\Phi)d\vec{x}^2$, and are related by the longitudinal traceless space–space parts of the Einstein equation by means of the algebraic equation

$$k^2(\Phi + \Psi) = -32\pi G a^2\rho_r\Theta_{r2}. \tag{14}$$

Note that an alternative to the time–time Einstein equation (Eq. (13)) is its time–space component given by

$$\dot{\Phi} - \frac{\dot{a}}{a}\Psi = -4\pi G\frac{a^2}{k}[\rho_m u_m + 4\rho_r\Theta_{r1}]. \tag{15}$$

2.2. Initial conditions

We assume initial conditions arising from inflation for which the primordial power spectrum of the potential Φ follows

$$P_\Phi(k) = \frac{50\pi^2}{9k^3}\left(\frac{k}{H_0}\right)^{n-1}\delta_H^2\left(\frac{\Omega_m}{D(a=1)}\right)^2, \tag{16}$$

where δ_H is a normalisation parameter, $D(a)$ is the late time growth factor (normalised to $D(a) = a$ in the matter era), n is the scalar spectral index, and H_0 is the present value of the Hubble parameter. For adiabatic initial conditions, the other fields at early times are given by Ma and Bertschinger (1995)

$$\begin{aligned} \Phi &= -\left(1 + \frac{2}{5}R_\nu\right)\Psi \\ \delta &= \delta_b = 3\Theta_0 = 3\mathcal{N}_0 = -\frac{3}{2}\Psi \\ u &= u_b = 3\Theta_1 = 3\mathcal{N}_1 = \frac{1}{2}k\eta\Psi \\ \mathcal{N}_2 &= \frac{1}{30}k\eta\Psi, \end{aligned} \tag{17}$$

where $R_\nu = (\rho_\nu + P_\nu)/(\rho + P) = \Omega_\nu/\left(\frac{3}{4}\Omega_m a + \Omega_r\right)$ is the neutrino ratio, and all the other perturbation fields are set to 0 at the initial time.

2.3. Practicalities

In practice, the hierarchy of moments for the photon, photon polarisation, and neutrino moments can be truncated to a maximum multipole $l_{\max}$ by replacing Eqs. (8)–(9) for $\dot{\Theta}_l$ at $l = l_{\max}$ with (Ma and Bertschinger, 1995)

$$\dot{\Theta}_l \simeq k\Theta_{l-1} - \frac{l+1}{\eta}\Theta_l + \dot{\tau}\Theta_l, \tag{18}$$

and similarly for the photon polarisation moments Θ_{Pl} and for the neutrino moments $\mathcal{N}_l$, but without the Thomson scattering term in the latter case.

The optical depth τ and the sound speed c_s can be pre-computed using public recombination codes such as RECFAST (Seager et al., 1999, 2000), RECFAST++ (Seager et al., 1999; Chluba and Sunyaev, 2010; Chluba et al., 2010; Rubiño-Martín et al., 2010; Chluba and Thomas, 2011) or COSMOSPEC (Chluba and Ali-Haïmoud, 2016). In PyCosmo, we have implemented an interface to RECFAST++, as well as the possibility of external input recombination variables for comparisons with other Boltzmann codes.