Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computation of the Initial Matter Power Spectrum $P(k)$ with PyCosmo

Semester Thesis

Department of Physics, ETH Zürich

Written by
Caterina Prior

Supervised by
Dr. Andreas Adelmann, PSI / ETH
Sonali Mayani, PSI / ETH

Abstract

This thesis examines the integration of the PyCosmo Python package for generating the initial matter power spectrum $P(k)$ into the IPPL Structure Formation mini-app, providing an alternative to the existing N-GenIC initial condition generator. This work aims to evaluate whether PyCosmo can feasibly and effectively generate a tailored power spectrum to use in the initial condition generation N-body simulations of large-scale cosmic structure formation. The study compares the initial conditions and final structures produced using PyCosmo's BBKS-based power spectrum versus N-GenIC's internal Efstathiou parametrisation. A custom script was developed to generate a PyCosmo power spectrum compatible with N-GenIC. Simulations were run with 32^3 particles in a 64 Mpc h^{-1} box from a starting redshift of $z = 11.5$. Results indicate statistically significant differences in the initial particle distributions. The N-GenIC-generated power spectrum, which has a higher amplitude, leads to the earlier formation of more pronounced cosmic structures and larger halo densities compared to the PyCosmo-initialized simulation. This work demonstrates that the choice of power spectrum generation method has a tangible impact on simulation outcomes, highlighting the importance of accurate initial condition assumptions in cosmological simulations. Future work should explore these differences over longer evolutionary timescales and with more complex physics.

July 21, 2025

Contents

1 Introduction

The ΛCDM model was recognised as the standard cosmological model nearly two decades ago [1]. This paradigm combines a cosmological constant (Λ) with cold dark matter (CDM) to account for the universe's accelerated expansion and thermal evolution. Present research aims to test and refine the ΛCDM model [2]. Indeed, this theory has overcome many observational tests and is supported by measurements of the Cosmic Microwave Background (CMB), Type Ia supernovae, Baryon Acoustic Oscillations (BAO), weak gravitational lensing, and large-scale galaxy surveys [3].

Current and upcoming surveys continue to measure cosmological probes which can be used to test this concordance model. These experiments include DES [4], DESI [5], LSST [6], EUCLID [7], and WFIRST [8]. However, these data need to be compared to predictions generated by the ΛCDM model to test its validity and constrain cosmological parameters. Therefore, observational data must be compared with predictions of how the initial conditions evolve, which can be obtained through large-scale N-body simulations of cosmic structure formation.

Accurate comparison between theory and observation requires precise theoretical predictions. This strongly depends on the choice of initial conditions, which have a measurable impact on the accuracy of large-scale structure simulations [9]. A key initial condition is the matter power spectrum $P(k)$, which describes how the variance in matter density is distributed across spatial scales and is typically expressed as a function of the comoving wavenumber k, (where $k \equiv |\boldsymbol{k}|$). At any given redshift, the power spectrum encodes the statistical distribution of matter fluctuations, incorporating both the primordial perturbations and the effects of nonlinear structure formation. This makes it a fundamental initial condition for N-body simulations by allowing the generation of initial density and velocity fields required for the subsequent evolution of cosmic structure.

Initial-condition generators such as the publicly available, MPI-parallelised `N-GenIC` [10] code, can produce particle displacements and velocities that serve as inputs for N-body simulation codes like `Gadget-2` [11], which model the nonlinear evolution of cosmic structure. `N-GenIC` can internally generate the initial matter power spectrum, but alternatively, a tabulated spectrum corresponding to a custom cosmology can be provided as input. To calculate a matter power spectrum, generation typically involves solving multi-species Boltzmann equations, which can be done efficiently with tools like `CLASS` [12] and `CAMB` [13]. These numerical Einstein-Boltzmann solvers compute the evolution of linear perturbations in the early universe and derive cosmological observables such as the linear matter power spectrum. An alternative approach is to use the `PyCosmo` Python package, which predicts cosmological observables by numerically solving the Einstein–Boltzmann equations within a Python-based framework. It delivers precise predictions for ΛCDM cosmological parameters via an intuitive interface, available both locally and online [14]. Tests have revealed that `PyCosmo` outputs are in good agreement with other codes [15].

Once the initial density and velocity fields have been generated, structure formation can be simulated. Simulating large-scale structure formation with high galaxy-level resolution demands computational power attainable only through exascale-capable computing infrastructures. While advances in computing have enhanced simulation capabilities, the increasing diversity of modern hardware (combining nodes, CPUs, and GPUs) creates a need for flexible software that can be adapted with minimal changes. IPPL [1], a `C++`-based and performance-portable framework for Particle-In-Cell methods, was developed to meet this need. It utilises the Message Passing Interface (MPI) and Kokkos (a portable programming model optimized for high-performance computing) to manage communication in parallel tasks and supports execution on CPUs and GPUs. The Accelerator Modelling and Advanced Simulations (AMAS) Group at the Paul Scherrer Institute (PSI) have developed a set of mini-apps (Alpine [16]) using IPPL for various plasma physics applications. As part of extending the mini-app framework to cosmology, the Structure Formation mini-app was developed [17] to simulate large-scale dark matter structure formation and evolution. Focusing on dark matter dynamics is advantageous, as it simplifies the evolution equations to forms analogous to those in electrostatics and gravity. This mini-app computes forces using a Particle-Mesh algorithm based on Fourier techniques for periodic boundary conditions, where time integration is handled via the Leapfrog scheme. Currently, Structure Formation uses `N-GenIC` to generate initial conditions. However, its internally generated power spectrum differs from that produced by `PyCosmo` under identical initial conditions (see Appendix A for more details). The differences in the initial linear matter power spectra generated by these codes are expected to naturally propagate into variations in the resulting large-scale structures. It is therefore important to assess the feasibility and potential benefits of integrating `PyCosmo` into the Structure Formation pipeline. Unlike `N-GenIC`, which has limited flexibility in selecting the parametrisation of the linear theory input spectrum, `PyCosmo` offers a more customizable frame-

[1] `https://github.com/IPPL-framework/ippl.git`

work, including support for a broader range of transfer functions and full Boltzmann solvers. Incorporating such flexible initial-condition generation into N-body simulations supports the development of more accurate and theoretically consistent predictions, ultimately improving the comparison between ΛCDM model predictions and observational data.

In response to this need, the aim of this thesis is to integrate `PyCosmo`'s power spectrum generation into IPPL's Structure Formation mini-app and to evaluate the effect of this change on the simulation of large-scale structure, focusing on dark-matter dynamics. This will involve a comparison of the initial conditions generated by `N-GenIC` when the matter power spectrum is computed internally versus when it is provided as an externally tabulated input generated by `PyCosmo`, as well as a comparison of its effects on large-scale structure formation when simulated using Structure Formation. The thesis is structured as follows: Section 2 outlines the theoretical models underlying the power spectrum and its initialization. In Section 3, the process for generating the initial power spectra using `PyCosmo` and `N-GenIC` is detailed. Section 4 presents a comparison of the differences in the generated initial conditions and the resulting large-scale structures formed after simulation evolution. Section 5 provides a summary of the findings and discusses potential directions for future work. Finally, additional results, along with a detailed user guide for generating power spectra using the PyCosmo implementation and a link to the source code, are provided in the appendices.

2 Theoretical background

`PyCosmo` is a general-purpose cosmology code for generating predictions of key cosmological quantities. It currently supports the standard ΛCDM model, which includes dark matter (Ω_m), baryons (Ω_b), the Hubble parameter (H_0), the spectral index (n_s), the fluctuation amplitude (σ_8), and a dark energy component with a barotropic parameter of $w = -1$ [15]. Key assumptions of the ΛCDM model include spatial flatness, general relativity, and a positive cosmological constant. The model also assumes that galaxy redshifts are due to cosmic expansion, that primordial nucleosynthesis produced light elements, and that the CMB originates from the early universe [18].

In the ΛCDM framework, structure forms through the gravitational instability of collisionless dark matter perturbations. This process is hierarchical: small-scale fluctuations collapse first, gradually building up larger structures over cosmic time [19]. The growth of these linear perturbations is governed by the Einstein-Boltzmann equations, which `PyCosmo` solves symbolically using the `Sympy` Python library [20]. `PyCosmo` outputs include quantities describing both the background expansion and the evolution of perturbations. However, only the linear matter power spectrum is required to initialise cosmological N-body simulations. This is because simulations start at high redshift, where density fluctuations δ are small ($\delta \ll 1$) and the universe is still well described by linear theory.

The comoving mass density field at position $\boldsymbol{x} \in \mathbb{R}^3$ and time t is expressed as

$$\rho(\boldsymbol{x}, t) = \bar{\rho}(t)[1 + \delta(\boldsymbol{x}, t)], \tag{1}$$

where $\bar{\rho}(t)$ is the mean background density and $\delta(\boldsymbol{x}, t)$ is the overdensity field. In practice, working in Fourier space is more convenient as it simplifies the statistical description of non-Gaussianities [21]. The matter power spectrum $P(k)$ is defined via the two-point correlation of Fourier modes

$$\left\langle \tilde{\delta}(k)\, \tilde{\delta}^*(k') \right\rangle = (2\pi)^3 P(k)\, \delta^{(3)}(k - k'), \tag{2}$$

where the angular brackets denote the ensemble average (which is equivalent to the spatial average under the implicit assumption of a homogeneous and isotropic universe), k' denotes an independent Fourier mode to k, $\delta^*(k')$ is the complex conjugate of $\delta(k')$, and $\delta^{(3)}$ denotes the three-dimensional Dirac delta function. The Dirac delta function ensures that only modes with $k = k'$ contribute due to statistical homogeneity.

The power spectrum is connected to different redshifts by the growth factor $D_+(a)$, which describes how density perturbations evolve over time as a function of the scale factor a. `PyCosmo` calculates the linear growth factor $D_+(a)$ by solving

$$\frac{d^2\delta_m}{da^2} + \left(\frac{d\ln H}{da} + \frac{3}{a}\right)\frac{d\delta_m}{da} - \frac{3\Omega_m H_0^2}{2a^5 H^2}\delta_m = 0 \tag{3}$$

for matter-overdensity δ_m. Here, $\Omega_m = \frac{\rho_{m,0}}{\rho_{0,crit}}$, $H = \frac{\dot{a}}{a}$, H_0, $H_0^2 = \frac{8\pi G}{3}\rho_{0,crit}$ and $\rho_{0,crit}$ is the critical density today. The solution $\delta_m(a)$, normalised such that $\delta_m(a = 1) = 1$, defines the growth factor $D_+(a)$, which equals a during matter domination and satisfies $D_+(a = 1) = 1$ today. This expression is a simplification of the Einstein-Boltzmann equation in the sub-horizon ($k \ll \eta^{-1}$, where η is the comoving causal horizon), late time ($a \gg a_{eq}$, where a_{eq} is the scale factor at the time of matter-radiation equality) regime.

Based on the assumption that the primordial perturbations are nearly Gaussian with zero mean, the linear matter power spectrum can be written as

$$P(k,a) \propto \frac{k^{n_s}}{\Omega_m^2} D_+^2(a) T^2(k), \tag{4}$$

where $T(k)$ is the matter transfer function [22]. This determines the scale dependence of the linear matter power spectrum and encodes the evolution of density perturbations as they enter the horizon and through the transition from radiation to matter domination.

To avoid computationally expensive numerical solutions, approximate analytical fitting functions are often employed. One widely used example is the Bardeen–Bond–Kaiser–Szalay (BBKS) fitting formula, which is valid under the assumption that $\Omega_m \gg \Omega_b$ [21]. The BBKS formula simplifies the matter transfer function by neglecting the contributions from baryons and radiation. As a result, it eliminates effects such as Silk damping (where density fluctuations in the early universe are suppressed due to interactions between photons and baryons [23]), acoustic oscillations, and the suppression of growth before matter-radiation equality. The BBKS approximation to the transfer function is given by

$$\begin{aligned} T_{c,\mathrm{BBKS}}(k) \equiv & \frac{\ln(1+2.34q)}{2.34q}[1+3.89q \\ & +(16.1q)^2+(5.46q)^3+(6.71q)^4\big]^{-1/4}, \end{aligned}$$

in which

$$q(k) \equiv \frac{k\theta^{1/2}}{(\omega_m-\omega_b)\,\mathrm{Mpc}^{-1}}, \quad \theta \equiv \frac{\rho_r}{1.68\rho_\gamma}, \tag{5}$$

where $\omega_X \equiv \Omega_X h^2$ is the reduced density parameter, with Ω_X being the reduced density parameter of the species X, h is the Hubble constant, and ρ_X is the background density [24]. Here, $X = b, c, m, r, \nu, \gamma$ denotes Baryons, CDM, pressureless matter, radiation, neutrinos, and photons, respectively. Our simulations are limited to dark matter structure formation, thus this approximation is suitable for our needs. This fit is used to compute angular power spectra [15], employing the Limber approximation for further simplification [25].

However, some cosmological code solvers continue to use simplistic parametrisations of the power spectrum. For instance, `N-GenIC` gives the user the option to use the following parametric approximation of the power spectrum:

$$P(k) = \frac{Bk}{\left\{1+\left[ak+(bk)^{3/2}+(ck)^2\right]^v\right\}^{2/\nu}} \tag{6}$$

where B is a normalisation constant set by COBE measurements [26], $a = (6.4/\Gamma)h^{-1}$Mpc, $\Gamma = \Omega_0 h$, Ω_0 is the matter-density parameter in the present epoch, $b = (3.0/\Gamma)h^{-1}$Mpc, $c = (1.7/\Gamma)h^{-1}$Mpc, and $\nu = 1.13$ [27]. This formula is known as the "Efstathiou parametrisation". Its tunable parameters allow it to fit a variety of CDM models and show good agreement with power spectrum data. For instance, if $\Gamma = h$ and Ω_b is small, Equation 6 has been found to accurately model the linear power spectrum [28]. However, the Efstathiou parametrisation is rarely used in current literature after being superceded by Boltzmann solvers within the last decade.

Since `N-GenIC` samples the power spectrum in logarithmic wavenumber ($\ln(k)$) space, the dimensionless power spectrum Δ^2 is a more practical input than the dimensional $P(k)$. Δ^2 represents the contribution to the variance of the density field per logarithmic interval in wave number and is fully normalised. This makes it more intuitive, particularly when generating Gaussian random fields with a specific variance spectrum [29]. Additionally, working with dimensionless quantities improves numerically stability [30]. In initial condition generators like `N-GenIC`, the dimensionless power spectrum is defined as

$$\Delta^2 = 4\pi k^3 P(k). \tag{7}$$

Initial particle velocities and displacements can be generated using the Zel'dovich approximation. This provides a physically motivated (though approximate) framework for modelling non-linear structure formation through the growth of initial perturbations, offering insight into the origin of the filamentary features of the cosmic web. It is an extrapolation of a Lagrangian framework beyond the range of its formal applicability [31]. Formally, it is only valid in the linear regime [32], which makes it appropriate for simulating structure formation on large-scales.

3 Methodology

This section outlines how the custom power spectrum generator, built using `PyCosmo`, produces a tabulated linear power spectrum compatible with `N-GenIC` for generating initial conditions for cosmological simulations. The generator accepts the same input format as the IPPL Structure Formation code, though only a subset of parameters is required by `PyCosmo` to compute the power spectrum. These parameters—defining the fiducial cosmology and simulation resolution—are listed in Table I and ensure consistency between the theoretical spectrum and the resulting particle distribution. With these inputs, the generator is configured as follows:

- Reads to relevant parameter file.

- Extracts variables and passes them to the custom power spectrum-generating class.
- Uses the custom class to initialise a fiducial ΛCDM cosmology according to the input parameters.
- Determines the range of comoving wave numbers k according to the simulation volume and particle resolution.
- Uses the inbuilt `PyCosmo` 'Background' class to compute the Hubble parameter and comoving distances
- Computes the linear power spectrum over the range of wave numbers.
- Tabulates the resulting power spectrum where each entry corresponds to $ln(k)$ and $ln(\Delta^2)$.
- The tabulated spectrum is stored in an output directory for user access.

The output is a plain two-column ASCII file, matching the input format required by the `N-GenIC` code, which expects a tabulated linear matter power spectrum. Specifically, the file must list pairs of values where each line corresponds to $\log_{10}(k)$ and $\log_{10}(\Delta^2)$, with k in units of $h\,\mathrm{Mpc}^{-1}$ and Δ^2 denoting the dimensionless power spectrum. The power spectrum output by `PyCosmo` serves as a direct input to `N-GenIC`, selected for its capability to efficiently generate initial conditions for large-scale, high-particle-count simulations.

The initial positions and velocities of dark matter particles are then generated using `N-GenIC`. When run with the appropriate parameter file and input power spectrum, `N-GenIC` constructs a Gaussian random density field consistent with the given power spectrum and applies Lagrangian perturbation theory (typically the Zel'dovich approximation) to displace particles from a uniform grid, thus setting up the initial conditions for the subsequent N-body simulation. The procedure can be summarised as follows:

- A Cartesian grid is created to match the simulation volume, with periodic boundary conditions.
- The perturbation field is computed by applying the inverse Fourier transform to the power spectrum.
- Velocity perturbations are determined using the Zel'dovich approximation.
- These perturbations are then applied to the grid.

If a tabulated power spectrum is not provided, the user must choose an analytic parameterisation of the linear theory spectrum. Two options are available: the Eisenstein & Hu fitting function and Efstathiou parameterisation. While Eisenstein & Hu accommodates various cosmologies, including baryons and radiation [33], it is unnecessarily complex for the dark-matter-only model considered here. The simpler Efstatitou function is more appropriate and computationally efficient for this case.

The complete workflow, summarised in Figure 1, ensures that the initial particle distribution is physically consistent with the specified cosmological model, providing reliable starting conditions for large-scale structure formation simulations.

4 Results

The following results were consistent across simulations with varying particle numbers and box sizes. For clarity and brevity, we present one representative case: a simulation with 32^3 particles in a

Parameter	**Value**	**Physical Meaning**
`np`	32	Defines the Nyquist frequency and sets the maximum wavenumber, chosen such that `np`^{3} equals the total number of particles.
`box_size`	64.0 Mpc/h	Physical size of one edge of the simulation cube given in code units.
`z_in`	50	The redshift at which the power spectrum should be computed (i.e. the starting redshift of the simulation)
`hubble`	0.7	Hubble parameter today
`Omega_m`	0.3	Total dark matter density parameter at $z = 0$ (Ω_m)
`Omega_bar`	0.0	Total baryon density parameter at $z = 0$ (Ω_b)
`sigma_8`	0.8	Velocity dispersion coefficient σ_8 of the ΛCDM model

Table I: PyCosmo input parameters required for linear power spectrum generation.

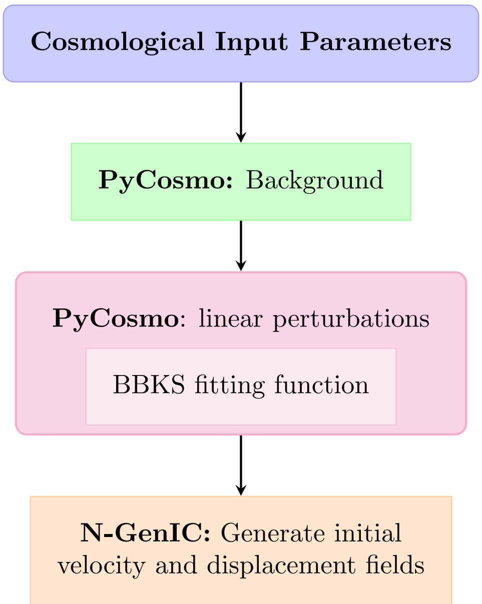


Figure 1: Workflow for generating initial conditions using a PyCosmo-generated power spectrum as input to N-GenIC. Cosmological parameters are first read from the specified input file. The "Background" class in PyCosmo computes the Hubble parameter and comoving distances. The "linear perturbations" module calculates the matter power spectrum. In this case, the linear fitting function used is BBKS. This spectrum is then passed to N-GenIC, which generates the initial displacement and velocity fields for the simulation.

64.0 Mpc box at redshift $z = 11.5$. This redshift was chosen to avoid numerical issues in PyCosmo, which occur at higher redshifts and are currently under investigation by its development team.

We begin by statistically comparing the initial conditions through histograms of the displacement and velocity fields along the x, y, and z axes. This is followed by both visual and quantitative analyses of the resulting large-scale structures produced using the IPPL Structure Formation mini-app.

4.1 Comparison of the initial conditions

We now compare the initial conditions (the three-dimensional particle positions and velocities) produced by N-GenIC when using a PyCosmo-generated input power spectrum versus when generating the power spectrum internally. These initial conditions are illustrated as histograms in Figure 2. Initially, the distributions appear broadly consistent in both cases. The velocity histograms show similar Gaussian profiles, and the magnitudes of both position and velocity remain within comparable orders. This suggests that N-GenIC is able to interpret and apply both externally supplied and internally generated power spectra in a manner that yields similar initial particle distributions. However, a chi-squared analysis yields a p-value of < 0.05 for each pair of distributions, indicating statistically significant differences in the initial conditions.

While the initial conditions show broad similarity, notable differences remain. In the coordinate distributions, for example, both datasets exhibit similar overall shapes; however, starting from the left, the odd-numbered bins are predominantly populated by data from the PyCosmo-generated spectrum, whereas the even-numbered bins are dominated by data from the N-GenIC-generated spectrum. This alternating pattern suggests that differences in the input power spectra influence how structure is initially seeded in the simulation. This can be understood by examining how N-GenIC constructs initial conditions from the power spectrum. Since N-GenIC randomly selects Fourier phases for each mode and scales them according to the square root of the power spectrum amplitude, variations in the shape and amplitude of the input spectrum can affect the initial positions and velocities of the particles. Furthermore, in the Zel'dovich approximation, particle positions are offset from a regular grid using displacements proportional to the potential gradient. Since the potential depends on $P(k)/k^2$, small differences in the shape of the power spectrum could cause cumulative position shifts. Furthermore, N-GenIC generates the initial conditions using a discrete mesh and a fast Fourier transform (FFT). Mapping the continuous power spectrum onto a discrete grid introduces quantisation and aliasing effects, which are particularly significant when the power spectra differ near the Nyquist limit.

The coordinate distribution integrals differ only slightly (on the order of 10^{-5}), while the velocity integrals are consistently larger for the N-GenIC dataset. This trend aligns with the initial power spectra presented in Section A, where the N-GenIC-generated spectrum exhibits systematically higher amplitudes at relevant scales. Since the velocity field is derived from the gradient of the gravitational potential—which itself is sourced from the matter power spectrum weighted by factors of k, a power spectrum with higher amplitude naturally leads to larger velocities. This results in the observed increase in the integral of the velocity distribution in the N-GenIC case.

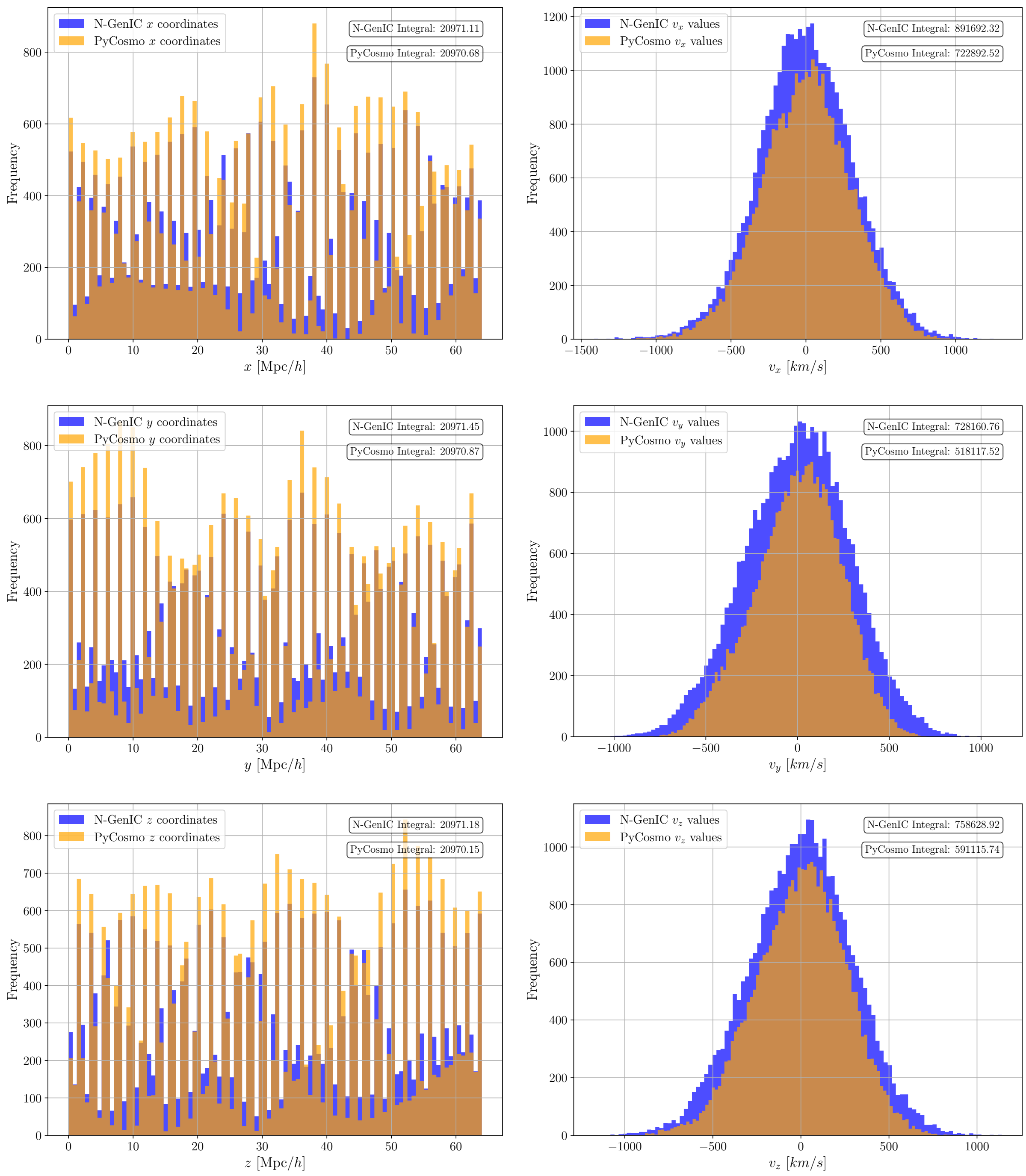


Figure 2: Histograms of initial conditions at $z = 11.5$ for a simulation with 32^3 particles in a box with a side length of 64 Mph h^{-1}. The plots compare initial conditions generated by `N-GenIC`, with one dataset generated using its internal power spectrum (blue) versus the dataset generated with an input `PyCosmo` power spectrum (orange). Left: distributions of initial x, y, and z coordinates. Right: distributions of initial velocities v_x, v_y, and v_z.

4.2 Comparison of final structures

The simulation was performed using the IPPL Structure Formation code [17], which also enabled visualisation of the resulting large-scale structures. Figures 3b and 3a show the final structures evolved from initial conditions generated using the `N-GenIC` and `PyCosmo` power spectra, respectively.

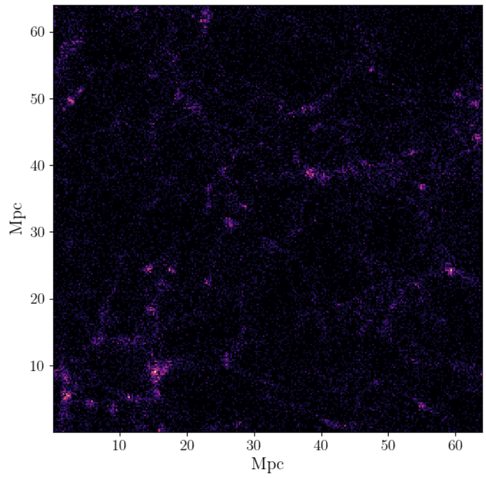
Figure 3 shows that the final structures produced by the two methods exhibit broadly similar filamentary patterns. This agreement is further supported by the histograms in Figure 4, which indicate that, despite differences in the initial power spectra, both approaches seed similar large-scale distributions. However, the simulation using `N-GenIC's` internally generated power spectrum displays higher densities at filament intersections (haloes), as evidenced by the more intense colouration in these regions. A primary cause likely lies in the distinct matter transfer functions employed: the empirical Efstathiou parametrisation used by `N-GenIC` versus the BBKS function implemented in `PyCosmo`. The BBKS transfer function is based on analytic approximations to the Boltzmann equations for cold dark matter perturbations, incorporating radiation-to-matter domination transitions [21], whereas the Efstathiou parametrisation is a purely empirical fit without derivation from first principles. Thus, the results are consistent with expectations: the higher amplitude of `N-GenIC`'s linear and dimensionless power spectra leads to the earlier and more pronounced formation of massive cosmic structures.

5 Conclusion

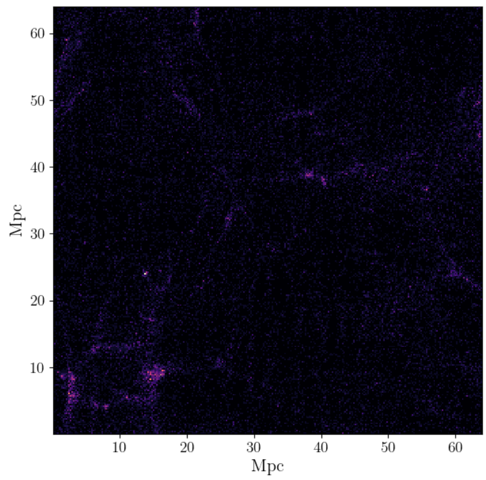
5.1 Summary and Discussion

This thesis demonstrated how `PyCosmo` can be integrated into IPPL's Structure Formation mini-app to generate initial conditions from a custom linear power spectrum, illustrating the impact of different spectrum generation methods on large-scale structure formation. As cosmological simulations grow in complexity and precision, understanding the sensitivity of outcomes to initial conditions becomes increasingly important. By examining how variations in the input power spectrum affect structure evolution, this work offers valuable insight into the role of assumptions made during the initialisation of the linear matter power spectrum.

A script for producing the linear matter power spectrum using the same inputs as Structure Formation was developed. Initial conditions were generated by passing the different power spectra to `N-GenIC`. A simulation with box side-length of 64 Mpc h^{-1} and 32^3 particles initialised at redshift $z = 11.5$ was run to generate results.



(a) 2D heat map of the final structure resulting from the evolution of initial conditions generated using the `N-GenIC` internal power spectrum and the Zel'dovich approximation to create initial conditions.



(b) 2D heat map of the final structure resulting from the evolution of initial conditions generated using the `PyCosmo` power spectrum and the Zel'dovich approximation in `N-GenIc` to create initial conditions.

Figure 3: Heat maps of the final large-scale structure from a 64 Mpc h^{-1} simulation with 32^3 particles. Each image shows a projection along one axis, with pixel colours indicating the logarithm of the particle count accumulated along the third axis.

The results demonstrate that large-scale structure evolution is sensitive to the method used for initializing the linear matter power spectrum. This sensitivity is evident in the initial condition histograms, where significant differences between distributions were confirmed through integral compar-

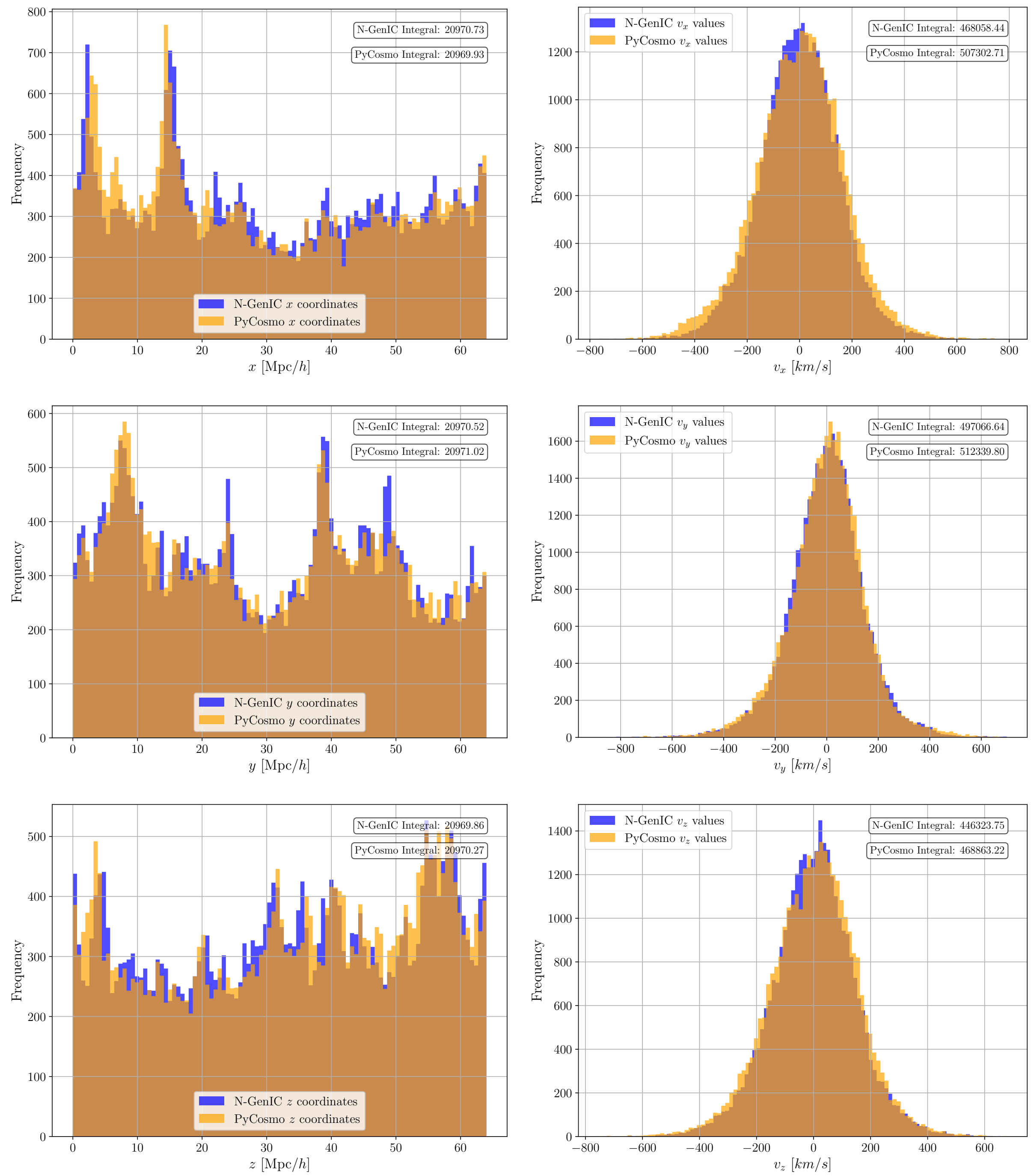


Figure 4: Histograms of large-scale structures generated by a simulation evolved from $z = 11.5$ to $z = 0$, where the histograms above show the conditions at $z \simeq 0$. The histograms show conditions from `N-GenIC` using its internal power spectrum (blue) versus a `PyCosmo`-generated input (orange). Left: distributions of initial x, y, and z coordinates. Right: distributions of initial velocities v_x, v_y, and v_z.

isons and chi-squared tests.

These differences carry through to the final dark matter distribution produced by IPPL's Structure Formation mini-app, where the impact of the initial power spectrum becomes visually and statistically evident. As shown in Figure 3b, the simulation using `N-GenIC's` power spectrum produces more developed structures, with clearer filaments and stronger clustering than the `PyCosmo`-initialized simulation in Figure 3a, which shows fainter structures and sparser haloes. The histograms of the final states in Figure 4 further support this finding. The discrepancies can be attributed to the different parametrisation methods used by `PyCosmo` and `N-GenIC`. While `PyCosmo` uses the BBKS transfer function, `N-GenIC` uses the Efstathiou parametrisation. BBKS is derived from first principles, whereas the Efstathiou function is formulated to correspond with empirical data and match observations.

Overall, this project shows that the integration of linear matter power spectrum generation into IPPL's Structure Formation would indeed be feasible. The results suggest that the choice of parametrisation for generating the linear matter power spectrum impacts the final state of cosmological structures. However, the specific advantages of using a PyCosmo-generated spectrum remain uncertain and warrant further investigation through comparison with expected simulation outcomes.

5.2 Future Work and Unresolved Issues

This investigation can be extended to better understand how the initial power spectrum shapes structure formation in N-body simulations and to assess the benefits of different parametrisations.

Firstly, extending the simulation to include baryonic physics would allow a more realistic assessment of how different power spectra initialisation methods influence structure formation. Baryonic processes, such as pressure and cooling, are essential for galaxy formation and can significantly modify the matter power spectrum, especially on small and intermediate scales. However, their inclusion requires a fitting function beyond BBKS, which does not account for baryonic effects.

One limitation of this investigation is the restricted redshift range, with a maximum redshift of $z = 11.5$, imposed by bugs in the PyCosmo power spectrum computation. This constraint prevents the study of structure formation from earlier cosmic times, limiting the conclusions about the accuracy of the PyCosmo-generated power spectrum to relatively low redshifts and a short evolutionary timescale. However, small differences in the initial conditions could lead to more significant discrepancies when evolved over longer periods. Therefore, further investigation into why the PyCosmo fitting functions fail at higher redshifts would be valuable, as resolving this issue would allow simulations to start at earlier times and offer a more complete picture of structure evolution. Future work could then repeat the statistical and visual comparisons performed here to assess whether the initial conditions generated at higher redshifts lead to any significant differences in structure evolution.

Additionally, this work focused on comparing the initial conditions generated by N-GenIC using two different methods. As a further validation step, it would be valuable to compare these results with initial conditions derived from analytical models or other cosmology codes to ensure consistency and accuracy across different approaches.

Future improvements to the structure formation code could enhance this investigation. As noted in [17], the current Particle-Mesh (PM) method is fast but lacks resolution in dense regions. Alternatives like Adaptive Mesh Refinement (AMR) and Particle-Particle-Particle-Mesh (P^3M) can increase resolution where needed by refining dense areas or combining long- and short-range force calculations. Better numerical resolution would improve modelling of small-scale structures sensitive to the initial power spectrum's shape at high k, especially when comparing transfer functions like BBKS and Eisenstein & Hu, which differ in their treatment of baryon acoustic oscillations and Silk damping. Comparing simulation outcomes—such as halo mass functions or two-point correlation functions—across different input spectra would provide valuable insights into the accuracy and physical impact of each fitting method.

These enhancements not only increase numerical accuracy but also broaden the scope for meaningful physical comparisons between initial conditions derived from different theoretical models.

References

[1] Phillip James Edwin Peebles. *Principles of physical cosmology.* Princeton university press, 2020.

[2] Nathalie Deruelle and Jean-Philippe Uzan. "The Lambda-CDM model of the hot Big Bang". In: *Oxford Scholarship Online* (2018).

[3] Horatiu Nastase. "Evidence for Dark Matter and the ΛCDM Model". In: *Fundamental Theories of Physics* (2019).

[4] T Abbott et al. "The Dark Energy Survey: more than dark energy–an overview". In: *Monthly Notices of the Royal Astronomical Society* 460.2 (2016), pp. 1270–1299.

[5] Amir Aghamousa et al. "The DESI experiment part I: science, targeting, and survey design". In: *arXiv preprint arXiv:1611.00036* (2016).

[6] Nora Elisa Chisari et al. "Core Cosmology Library: Precision Cosmological Predictions for LSST". In: *The Astrophysical Journal Supplement Series* 242.1 (2019), p. 2.

[7] Luca Amendola et al. "Cosmology and fundamental physics with the Euclid satellite". In: *Living reviews in relativity* 21 (2018), pp. 1–345.

[8] Olivier Dore et al. "WFIRST: the essential cosmology space observatory for the coming decade". In: *arXiv preprint arXiv:1904.01174* (2019).

[9] Benjamin L'Huillier, Changbom Park, and Juhan Kim. "Effects of the initial conditions on cosmological N-body simulations". In: *New Astronomy* 30 (2014), pp. 79–88.

[10] Volker Springel. *N-GenIC: Cosmological structure initial conditions.* Astrophysics Source Code Library, record ascl:1502.003. Feb. 2015.

[11] Volker Springel. "The cosmological simulation code GADGET-2". In: *Monthly Notices of the Royal Astronomical Society* 364.4 (Dec. 2005), pp. 1105–1134. arXiv: `astro-ph/0505010 [astro-ph]`.

[12] Julien Lesgourgues and Thomas Tram. "The Cosmic Linear Anisotropy Solving System (CLASS) IV: efficient implementation of non-cold relics". In: *Journal of Cosmology and Astroparticle Physics* 2011.09 (2011), p. 032.

[13] Antony Lewis, Anthony Challinor, and Anthony Lasenby. "Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models". In: *The Astrophysical Journal* 538.2 (2000), p. 473.

[14] A. Refregier et al. "PyCosmo: An integrated cosmological Boltzmann solver". In: *Astronomy and Computing* 25 (2018), pp. 38–43.

[15] F. Tarsitano et al. "Predicting cosmological observables with PyCosmo". In: *Astronomy and Computing* 36 (2021), p. 100484.

[16] Sriramkrishnan Muralikrishnan et al. "Scaling and performance portability of the particle-in-cell scheme for plasma physics applications through mini-apps targeting exascale architectures". In: *Proceedings of the 2024 SIAM Conference on Parallel Processing for Scientific Computing (PP).* Society for Industrial and Applied Mathematics, 2024, pp. 26–38. eprint: `https://epubs.siam.org/doi/pdf/10.1137/1.9781611977967.3`.

[17] B. Crazzolara. "Cosmological Structure Formation with the Performance Portable IPPL Library". In: *ETH* (Dec. 2024).

[18] Martín López-Corredoira. "Tests and problems of the standard model in cosmology". In: *Foundations of Physics* 47.6 (2017), pp. 711–768.

[19] R. E. Smith et al. "Stable clustering, the halo model and non-linear cosmological power spectra". In: *Monthly Notices of the Royal Astronomical Society* 341.4 (June 2003), pp. 1311–1332. arXiv: `astro-ph/0207664 [astro-ph]`.

[20] U. Schmitt et al. "sympy2c: From symbolic expressions to fast C/C++ functions and ODE solvers in Python". In: *Astronomy and Computing* 42 (2023), p. 100666.

[21] Jian Qin et al. "Numerical investigation of non-Gaussianities in the phase and modulus of density Fourier modes". In: *Monthly Notices of the Royal Astronomical Society* 514.1 (2022), pp. 1548–1556.

[22] S Dodelson and F Schmidt. *Modern Cosmology, doi: 10.1016.* Tech. rep. C2017-0-01943-2, 2020.

[23] Axel Brandenburg, Kari Enqvist, and Poul Olesen. "The effect of Silk damping on primordial magnetic fields". In: *Physics Letters B* 392.3-4 (1997), pp. 395–402.

[24] Kazuhiro Yamamoto, Naoshi Sugiyama, and Humitaka Sato. "Evolution of small-scale cosmological baryon perturbations and matter transfer functions". In: *The Astrophysical Journal* 501.2 (1998), p. 442.

[25] Marilena LoVerde and Niayesh Afshordi. "Extended Limber approximation". In: *Phys. Rev. D* 78 (12 2008), p. 123506.

[26] George F Smoot. "COBE DMR observations of early universe physics". In: *AIP Conference Proceedings*. Vol. 272. 2. American Institute of Physics. 1992, pp. 1591–1601.

[27] G. Efstathiou, J. R. Bond, and S. D. M. White. "COBE background radiation anisotropies and large-scale structure in the Universe". In: *Monthly Notices of the Royal Astronomical Society* 258.1 (Sept. 1992), 1P–6P. eprint: `https://academic.oup.com/mnras/article-pdf/258/1/1P/2896039/mnras258-001P.pdf`.

[28] G. Efstathiou et al. "Numerical techniques for large cosmological N-body simulations". In: *Astrophysical Journal Supplement* 57 (Feb. 1985), pp. 241–260.

[29] Bruce N Miller and Jean-Louis Rouet. "Cosmology in one dimension: Fractal geometry, power spectra and correlation". In: *Journal of Statistical Mechanics: Theory and Experiment* 2010.12 (2010), P12028.

[30] Francisco M Fernández. "Dimensionless equations in non-relativistic quantum mechanics". In: *arXiv preprint arXiv:2005.05377* (2020).

[31] Ya. B. Zel'dovich. "Gravitational instability: An approximate theory for large density perturbations." In: 5 (Mar. 1970), pp. 84–89.

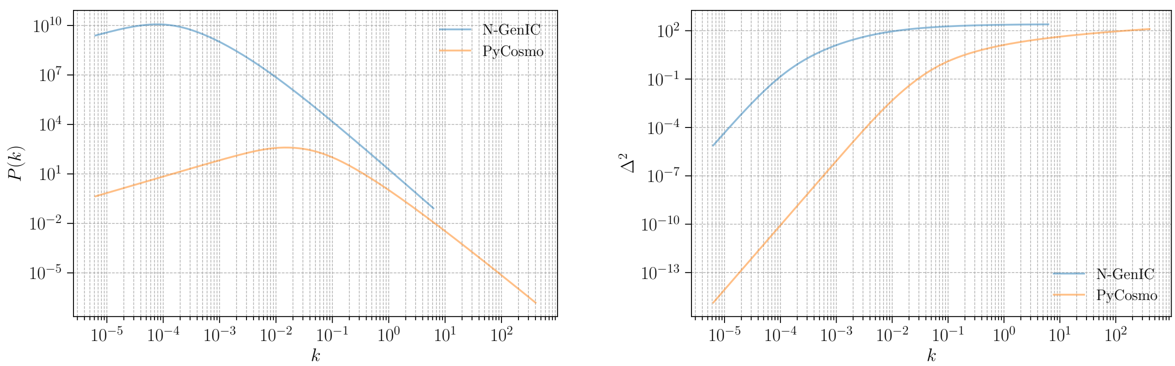
[32] Johan Hidding, Sergei F Shandarin, and Rien van de Weygaert. "The Zel'dovich approximation: key to understanding cosmic web complexity". In: *Monthly Notices of the Royal Astronomical Society* 437.4 (2014), pp. 3442–3472.

[33] Daniel J Eisenstein and Wayne Hu. "Power spectra for cold dark matter and its variants". In: *The Astrophysical Journal* 511.1 (1999), p. 5.

A Comparison of PyCosmo and N-GenIC Power Spectra

Differences in the underlying algorithms of `PyCosmo` and `N-GenIC` manifest in the linear and dimensionless matter power spectra they produce. Figure 5a presents the initial linear matter power spectra generated by both codes under identical cosmological conditions. There are distinct discrepancies are evident between the two. In particular, the peak of the `N-GenIC` power spectrum $P(k)$ appears at a larger spatial scale (corresponding to a smaller k-value) compared to that of `PyCosmo`. This indicates that `N-GenIC` predicts stronger density fluctuations over more extended regions of space. Physically, this suggests that at redshift $z = 11.5$, the simulation with `N-GenIC` initial conditions will model a universe dominated by larger, more coherent overdense and underdense regions.

Differences also arise in the dimensionless power spectra produced by the two codes, as can be seen in Figure 5b. The `PyCosmo` spectrum exhibits a much deeper minimum near $k \approx 10^{-14}$ and follows an approximately linear slope up to $k \approx 10^{-2}$. In contrast, the `N-GenIC` spectrum displays a consistently positive curvature that gradually decreases with increasing k. Across the entire k-range considered, the `N-GenIC` dimensionless power spectrum remains higher than `PyCosmo`'s, indicating that `N-GenIC` predicts larger amplitudes of density fluctuations. The substantial differences between the power spectra suggest factors beyond numerical resolution or k-binning discrepancies.



(a) Comparison of the initial linear matter power spectrum generated by `PyCosmo` with that produced automatically by `N-GenIC` when no user-supplied power spectrum is provided. Both spectra correspond to a redshift of $z = 11.5$ for a simulation with 32^3 particles in a box of side length 50 Mpc.

(b) Comparison of the initial dimensionless power spectra generated by `PyCosmo` with that produced automatically by `N-GenIC` when no user-supplied power spectrum is provided. Both spectra correspond to a redshift of $z = 11.5$ for a simulation with 32^3 particles in a box of side length 50 Mpc.

Figure 5: Comparison of the initial linear (left) and dimensionless (right) matter power spectra generated by `PyCosmo` and `N-GenIC` at redshift $z = 11.5$ for a 32^3 particle simulation in a 50 Mpc box.

B Link to code

The code and plots can be found **here**. If the link doesn't work, please copy and paste the following link to the Github repository:

`https://github.com/caterina-prior/PyCosmo_SimGadget.git`

C Using the `PyCosmo_SimGadget` Repository

This repository sets up a Python-based environment for computing cosmological power spectra using a custom wrapper built on `PyCosmo`. The build is managed using `make`.

Project Structure

The table below gives an overview of the folder structure in the `PyCosmo_SimGadget` repository.

File/Folder	**Purpose**
PyCosmo_Power_Spectrum/	Contains code for generating power spectra using `PyCosmo`
Functions/	Helper functions for the power spectrum generation
initial_conditions/	.csv files of initial conditions generated by `N-GenIC`
outputted_power_spectrum/	Power spectrum files generated by `PyCosmo`
tests/	Unit and integration tests
test_k_values.py	Tests for k value computation correctness
test_power_spectrum_computation.py	Tests `PyCosmo` power spectrum generation
test_power_spectrum_plot.py	Tests power spectrum plotting
test_read_param_file.py	Ensures proper reading of parameter files
parameter_files/	Input parameter files used by `PyCosmo`
`PyCosmo`_plotting/	Plot generation code
animate_density.py	Visualizes structure evolution
initial_conditions_comparison.py	Generates histograms of initial conditions
final_conditions_comparison.py	Generates histograms of final conditions
plot_input_spectrum.py	Generates $P(k)$ and $\Delta^2(k)$ plots
`PyCosmo`hub.mplstyle	Standard `PyCosmo` matplotlib style file
generated_plots/	Stores outputted plots
structure_formation_data/	Contains data snapshots of structure formation
ngenic_data/	Snapshots without `PyCosmo` input
`PyCosmo`_data/	Snapshots with `PyCosmo` input
power_spectrum_generation/	Main power spectrum generation code
custom_power_spectrum.py	Passes parameter file data to `PyCosmo` class
power_spectrum_class.py	Automates power spectrum generation
ngenic/	Code for generating initial conditions using `N-GenIC`

Setup Instructions

1. Clone the Repository

```
git clone https://github.com/your-repo/power-spectrum-generator.git
cd power-spectrum-generator
```

2. Set Up `PyCosmo`

2.1 Set Up the `PyCosmo` Environment

```
cd \texttt{PyCosmo}_Power_Spectrum
source setup_pycosmo.sh
```

This will:

- Activate the Python virtual environment
- Set `LD_LIBRARY_PATH` correctly

2.2 Load System Dependencies
The project requires two different GSL versions loaded in separate environments.
Requirements:

- `make`
- System package manager (`apt` for Linux/WSL, `brew` for macOS)
- GSL (2.8 for `PyCosmo`, 2.7.1 for `N-GenIC`)
- GCC (12.2.0)
- OpenMPI (4.1.6)
- Python (3.9)
- TeX Live
- `virtualenv`

On ETH Euler Cluster, load modules:

```
module load stack/2024-06
module load gcc/12.2.0
module load gsl/2.7.1
module load openmpi/4.1.6
module load python/3.9
module load texlive
```

2.3 Run the `PyCosmo` Makefile

```
make
```

3. Set Up `N-GenIC`

3.1 Setup the `N-GenIC` Virtual Environment

```
cd ngenic
dos2unix setup_ngenic.sh
source setup_ngenic.sh
```

3.2 Load System Dependencies for `N-GenIC`

```
module load stack/2024-06
module load gcc/12.2.0
module load openmpi/4.1.6
module load gsl/2.7.1
```

3.3 Install FFTW 2.1.5

```
cd ngenic
wget https://www.fftw.org/fftw-2.1.5.tar.gz
tar -xvzf fftw-2.1.5.tar.gz
cd fftw-2.1.5
./configure --prefix=<path> --enable-shared --enable-mpi
make
make install
```

Replace `<path>` with desired install path, e.g. `--prefix=$(pwd)/../local/fftw-2.1.5` for a local install.
3.4 Compile `N-GenIC`
Add to `N-GenIC` Makefile:

```
FFTW_INCL = -I path/include
FFTW_LIBS = -L path/lib
```

Then run:

```
cd ngenic
make
```

4. Usage

4.1 Generating a Power Spectrum with `PyCosmo`
Run tests:

```
make test
make test_full
```

Generate power spectrum with a parameter file:

```
make run PARAM=PyCosmo_Power_Spectrum/parameter_files/pycosmo_input_32.param
```

This executes:

```
python -m power_spectrum_generation.custom_power_spectrum parameter_files/pycosmo_input_32.param
```

Outputs power spectrum for 32^3 particles with box side length 50000 Mpc into `outputted_power_spectrum/`.
Clean the `PyCosmo` environment:

```
make clean       # Remove artifacts and virtualenv
make clear_cache # Clear sympy2c cache
```

4.2 Creating Initial Conditions with `N-GenIC` Run:

```
make run ID=PARTICLES
```

This generates initial condition files:

- `lsf_32` (binary, GADGET-compatible)
- `lsf_32.csv` (readable CSV)

Files are saved in
`PyCosmo_Power_Spectrum/initial_conditions`.

5. Generating Comparison Plots

Use scripts in `pycosmo_plotting/` to generate histograms and compare `PyCosmo` vs. `N-GenIC` outputs.

6. Switching Between Environments

Activate `PyCosmo` environment:

```
source \texttt{PyCosmo}/setup_pycosmo.sh
```

To switch to `N-GenIC`:

```
deactivate
source ngenic/setup_ngenic.sh
```

Note: Do not activate both environments simultaneously. Always deactivate before switching.