

GRGADGET: an N -body TreePM relativistic code for cosmological simulations

Eduardo Quintana-Miranda,^{1,2,3*} Pierluigi Monaco^{1,2,3,4} and Luca Tornatore^{1,2,3}

¹*Dipartimento di Fisica, Sezione di Astronomia, via G.B. Tiepolo 11, I-34143 Trieste, Italy*

²*INAF – Istituto Nazionale di Astrofisica, Osservatorio Astronomico di Trieste, via G.B. Tiepolo 11, I-34143 Trieste, Italy*

³*IFPU – Institute for the Fundamental Physics of the Universe, via Beirut 2, I-34100 Trieste, Italy*

⁴*INFN – Istituto Nazionale di Fisica Nucleare, Via Valerio 2, I-34127 Trieste, Italy*

Accepted 2023 April 13. Received 2023 April 6; in original form 2023 January 27

ABSTRACT

We present the merging of the Particle-Mesh (PM) relativistic GEVOLUTION code with the TreePM GADGET-4 code, with the aim of studying general relativity effects in cosmology. Our code, called GRGADGET, is able to track the evolution of metric perturbations in the weak field limit by using GEVOLUTION’s implementation of a relativistic PM in the Poisson gauge. To achieve this, starting from GEVOLUTION, we have written a C++ library called LIBGEVOLUTION, which allows a code to access and use the same abstractions and resources that GEVOLUTION uses for its PM-only N -body simulations. The code works under the assumption that particle interactions at short distances can be approximated as Newtonian, so that we can combine the forces computed with a Newtonian Tree with those computed with a relativistic PM. The result is a TreePM simulation code that represents metric perturbations at the scales where they are relevant while resolving non-linear structures. We validate our code by closely matching GADGET-4 forces, computed with the Tree switched off, with those computed with LIBGEVOLUTION in the Newtonian limit. With GRGADGET, we obtain a matter power spectrum that is compatible with Newtonian GADGET-4 at small scales and contains GR features at large scales that are consistent with results obtained with GEVOLUTION. We demonstrate that, due to the better resolution of the highly non-linear regime, the representation of the relativistic fields sampled on the mesh improves with respect to the PM-only simulations.

Key words: large-scale structure of Universe – cosmology: theory.

1 INTRODUCTION

The state of the art of precision cosmology provides a standard cosmological model, Λ CDM, that is consistent with most observational evidence on large scales, but relies on the existence of a dark sector populated by dark matter (DM) and dark energy (DE). The first is responsible for the formation of cosmological structures such as galaxies and their large-scale density field, while the second causes the observed accelerated expansion of the universe in the present epoch. Their physical nature is an open problem, since the only evidence of their existence comes from their gravitational interaction with visible matter. A possible explanation is that the dark sector is due to a misrepresentation of gravity, which on large scales does not follow Einstein’s general relativity (GR), at the basis of the Λ CDM model.

This fact has triggered a wave of interest in modifications of GR, which can lead to extra terms that explain DE or DM (see e.g. Silvestri & Trodden 2009; Capozziello & De Laurentis 2012, and references therein). Such modifications must be significant only on large scales or low density, because GR is very accurate in predicting planetary orbits, light deflection, and Doppler effects in solar system tests and has more recently been successfully tested with the detection

of gravitational waves (Abbott et al. 2016) and the direct imaging of black hole event horizons (Event Horizon Telescope Collaboration 2019).

In order to characterize DE in the age of its dominance, many projects have been planned to survey large parts of the sky and probe the large-scale distribution of matter using galaxy clustering and galaxy lensing, both from the ground (DES¹, Krause et al. 2017; DESI², DESI Collaboration 2016; Rubin’s LSST³, Ivezić et al. 2019; SKAO⁴ surveys) and from space (Euclid⁵, Laureijs et al. 2011; Roman⁶, Spergel et al. 2015; SphereX⁷, Doré et al. 2014). Some of these surveys have already started to produce a flood of data that will soon lead to a precise characterization of the galaxy and matter density fields. A comparison of these observations to model predictions, either using summary statistics or field-level inference, will lead to unprecedented tests not only of the cosmological model but also of the gravity theory behind it. With precision being guaranteed by the amount of available high-quality data, accuracy

¹www.darkenergysurvey.org

²www.desi.lbl.gov

³www.lsst.org

⁴www.skao.int

⁵sci.esa.int/web/euclid

⁶roman.gsfc.nasa.gov

⁷spherex.caltech.edu

* E-mail: eduardo.quintana@pm.me

will be achieved only by rigorous control of systematics, both in the data and in theory predictions.

The highly non-linear nature of the observed density field and the non-locality of gravity make cosmological simulations necessary to compare the predictions of current theories with the observations at an increasing level of accuracy. Yet most of the widely adopted simulation codes, like e.g. GADGET-4 (Springel et al. 2021), use Newtonian dynamics for the evolution of matter perturbations. This is not the ideal configuration to pass from the unobservable distribution of matter in a periodic comoving box to the observable distribution of light in the past light cone. Relativistic corrections can be added *a posteriori* by post-processing Newtonian simulation outputs; one specific example of this approach is the modelling of lensing due to the distortion of null geodesics (Bartelmann & Schneider 2001), while a more comprehensive approach to adding relativistic effects is presented by Borzyszkowski, Bertacca & Porciani (2017). However, even though the biases introduced by this approach are expected to be small, a fully self-consistent approach is necessary to convincingly demonstrate our ability of controlling theory systematics. For instance, galaxy clustering is affected by magnification bias due to lensing, and neglecting this effect induces a non-negligible bias in parameter estimation (Lepori et al. 2020; Alam et al. 2021). This is even more true when modified gravity theories are used: extensions of gravity are typically derived in a full relativistic context, and while they influence the Newtonian limit of gravity, the small but measurable relativistic effects may provide smoking-gun signals of a specific class of gravity theories. In this sense, restricting to the treatment of the Newtonian limit of modified gravity theories (as e.g. in Li et al. 2012; Puchwein, Baldi & Springel 2013; Llinares, Mota & Winther 2014; Hernández-Aguayo et al. 2021) may leave out crucial observable signatures, which may arise from the metric degrees of freedom (e.g. vector and tensor modes), their interaction with the matter sector and the coupling of the scale factor with the evolution equations.

Two examples of fully relativistic N -body codes for the evolution of cosmic perturbations, which integrate Einstein's equations to follow the motion of massive particles along their geodesics, are the adaptive mesh refinement (AMR) code GRAMES (Barrera-Hinojosa & Li 2020) and the Particle-Mesh (PM) code GEVOLUTION (Adamek et al. 2016). These have proven to be precious tools to produce accurate cosmological predictions, like a self-consistent treatment of massive neutrinos (Adamek et al. 2022), and to explore phenomena that were previously overlooked, like the strength of the frame dragging field acting on DM haloes (Barrera-Hinojosa et al. 2021). These codes sample the fields in a mesh that fills the simulated volume, but while GRAMES uses an AMR scheme to increase resolution only where it is needed, PM schemes working on a single non-adaptive mesh are well known to be limited by memory, so they are unable to achieve the large dynamic range required, e.g. to resolve DM haloes in large cosmological volumes. The integration of Newtonian particle trajectories has historically been addressed with the introduction of an oct-tree data structure (Barnes & Hut 1986) that provides a $M \log N$ scaling for the computation of gravity without compromising its accuracy. Because the integration of large-scale perturbations is very slow in this scheme, such an oct-tree is used to compute short-range forces and is complemented by a PM code on large scales. The resulting algorithm is commonly called TreePM, and it is the standard gravity solver for GADGET-4.

As we will show in next section, deviations from a pure Newtonian approach become significant on scales that are comparable with the Hubble horizon, so a Newtonian treatment of small-scale clustering,

performed by the Tree algorithm, would introduce a negligible error if large scales are treated by a fully relativistic gravity solver. This can be achieved, in a TreePM scheme, by using a relativistic PM code for large-scale gravity, where relativistic potentials are sampled on a small enough mesh so as to be effectively Newtonian on the scales where the Tree code gets in.

In this paper, we present an implementation of GADGET-4 that uses a PM library, based on GEVOLUTION relativistic code, as the PM part of the TreePM solver. This is a step toward the construction of an ecosystem of codes and post-processing tools to perform end-to-end simulations of future surveys, with the aim of achieving optimal control of all systematics, including theoretical ones. The paper is organized as follows: Section 2 gives an overview of the theory of relativistic perturbations, with a focus on the approach used in GEVOLUTION. Section 3 gives a description of the GADGET-4 and GEVOLUTION codes and describes the implementation of LIBGEVOLUTION and GRGADGET. Section 4 presents the tests performed to validate GRGADGET, while Section 5 gives our conclusions.

2 THEORY OF RELATIVISTIC PERTURBATIONS

The success of Newtonian simulations in describing the large-scale structure of the universe follows from the fact that, for an observer at rest with respect to the CMB, the metric of spacetime is very close to Friedmann–Lemaître–Robertson–Walker's (FLRW). Deviations from the Newtonian approach are expected to be significant, albeit small, on scales near the Hubble horizon, or when the energy-momentum tensor has relativistic components like radiation or fast massive neutrinos. Deviations from FLRW metric are expected to be strong in the proximity of compact objects, but this happens on scales that are far smaller than the resolution that can be afforded in simulations of large comoving volumes. It is thus fair to assume that the perturbations to the metric are small and can be described in a weak-field regime. This does not imply that deviations of the components of the energy-momentum tensor from homogeneity are assumed to be small, density perturbations can be highly non-linear: what we require is that the size of self-gravitating objects is much larger than their gravitational radius.

The GEVOLUTION code (see Adamek et al. 2016) models the spacetime metric with a perturbed FLRW metric in the weak field regime. In the *Poisson gauge*, the metric can be written as:

$$ds^2 = a^2 \left(-c^2 d\tau^2 (1 + 2\Psi) - 2c d\tau dx^i B_i + dx^i dx^j (\gamma_{ij} (1 - 2\Phi) + h_{ij}) \right), \quad (1)$$

where $a(\tau)$ is the scale factor of the FLRW background, τ is the conformal time, and x^i are the space coordinates. It is possible to exploit the residual degrees of freedom of the metric to impose the conditions $B_{ij}{}^i = 0$, $h_i{}^i = 0$, and $h_{ij}{}^j = 0$. In our notation, repeated latin indexes denote Einstein's summation over the spatial coordinates 1,2,3 and the vertical bar subscript, e.g. B_{ij} , denotes a covariant derivative with respect to the affine connection that emerges from the background spatial metric γ_{ij} .

The choice of the Poisson gauge is convenient because the two potentials Ψ and Φ are the gauge-invariant Bardeen potentials, and in the Newtonian limit, the field Ψ can be interpreted as the gravitational potential. In other words, this is the gauge in which the standard N -body solver is integrating the right equations of motion in the Newtonian limit (Chisari & Zaldarriaga 2011).

2.1 Field equations

The background, characterized by $a(\tau)$, is by construction a solution of the Einstein's equations in the presence of a homogeneous and isotropic energy-momentum tensor $\bar{T}^\mu{}_\nu$:

$$\bar{G}^\mu{}_\nu = -\frac{8\pi G}{c^4} \bar{T}^\mu{}_\nu, \quad (2)$$

where $\bar{G}^\mu{}_\nu$ is Einstein's tensor constructed from the metric (1) with the perturbations Ψ , Φ , B_i , and h_{ij} set to zero. Applying equation (2) to the FLRW metric, one obtains Friedmann's equations.

To solve for the perturbations of the metric, the usual procedure consist in subtracting (2) from the full Einstein's equations:

$$G^\mu{}_\nu - \bar{G}^\mu{}_\nu = -\frac{8\pi G}{c^4} (T^\mu{}_\nu - \bar{T}^\mu{}_\nu). \quad (3)$$

The right-hand side now contains the perturbation of the energy-momentum tensor due to inhomogeneities in mass and energy distributions, while the left-hand side is a very complicated non-linear expression containing the potentials Ψ , Φ , B_i , and h_{ij} and their space-time derivatives up to second order.

To reach a tractable set of equations that we can interpret and solve numerically, we apply the weak field assumption. The perturbations Ψ , Φ , B_i , and h_{ij} are assumed to be of order $\epsilon \ll 1$. Spatial derivatives are known to increase their amplitude by a factor of $\epsilon^{-1/2}$, accounting for the presence of shortwave fluctuations induced by the non-linear structure in the energy-momentum tensor, while time derivatives are assumed to preserve the perturbation order. Then one can expand $G^\mu{}_\nu - \bar{G}^\mu{}_\nu$ in terms of the metric perturbations, neglecting contributions with order higher than ϵ . For example, Φ is a term of order $\mathcal{O}(\epsilon)$, $\Phi_{,i}$ has order $\mathcal{O}(\epsilon^{1/2})$, $\Phi_{|n}^n$ is a leading term (order 1, because of the second derivative), quadratic terms like $\Phi_{,n}\Phi_{,n}$ are $\mathcal{O}(\epsilon)$, and a term like $\Phi_{,00}$ is considered as $\mathcal{O}(\epsilon)$. This type of expansion is known as the *shortwave correction* (Adamek, Durrer & Kunz 2014).

Furthermore, experience has shown that the scalar perturbations Φ and Ψ are generally larger than the vector and tensor perturbations B_i and h_{ij} . Indeed, the scalar potentials, which are sourced by the density perturbation ΔT^{00} , become the Newtonian potential in the Newtonian limit, while the vector perturbation B_i is sourced by ΔT^{0i} , that is small by a factor of v/c for non-relativistic matter perturbations, and h_{ij} by ΔT^{ij} , that is suppressed by a $(v/c)^2$ factor. Hence, it is fair to drop quadratic terms of B_i and h_{ij} in this weak field limit approximation.

In this approximation, from equation (3), it descends that its time-time component yields a Poisson-like equation for the scalar Φ :

$$\begin{aligned} \Phi_{|n}^n (1 + 4\Phi) - 3\frac{\mathcal{H}}{c^2} \Phi_{,0} + 3\frac{\mathcal{H}^2}{c^2} (\chi - \Phi) + \frac{3}{2} \Phi_{|n} \Phi_{|n} \\ = \frac{4\pi G a^2}{c^4} \Delta T^0{}_0, \end{aligned} \quad (4)$$

where $\mathcal{H} = a^{-1} \frac{da}{d\tau}$ and $\chi = \Phi - \Psi$. From the time-space section of equation (3), we obtain:

$$-\frac{B_{i|n}^n}{4c} - \frac{\Phi_{,i0}}{c^2} - \frac{\mathcal{H}}{c^2} (\Phi_{,i} - \chi_{,i}) = -\frac{4\pi G a^2}{c^4} \Delta T^0{}_i, \quad (5)$$

that, taking advantage of the condition $B_{n|n} = 0$, can be reduced to:

$$-\frac{B_{i|n}^n}{4c} = -\frac{4\pi G a^2}{c^4} P_\perp \Delta T^0{}_i, \quad (6)$$

where P_\perp is a linear operator that selects from a vector field its divergenceless component.

The traceless part of the spatial section of equation (3) leads to:

$$\begin{aligned} \left(\delta^j{}_b \delta^a{}_i - \frac{1}{3} \delta^a{}_b \delta^j{}_i \right) \left[\chi_{|j}{}^i - 2\Phi_{|j}{}^i \chi + 4\Phi \Phi_{|j}{}^i + 2\Phi_{|j} \Phi_{|i} \right. \\ \left. + \frac{1}{2c^2} h^i{}_{j,00} + \frac{\mathcal{H}}{c^2} h^i{}_{j,0} - \frac{1}{2} h^i{}_{j|n}{}^n \right. \\ \left. + \frac{1}{2c} \left(\frac{\partial}{\partial \tau} + 2\mathcal{H} \right) (B^i{}_{|j} + B_{j|}{}^i) \right] \\ = \left(\delta^j{}_b \delta^a{}_i - \frac{1}{3} \delta^a{}_b \delta^j{}_i \right) \left(-\frac{8\pi G}{c^4} \Delta T^i{}_j \right), \end{aligned} \quad (7)$$

from which we can determine the rest of the metric degrees of freedom χ and h_{ij} . Since the source of χ and h_{ij} are the perturbation of the energy-momentum tensor $\Delta T^i{}_j$, their amplitude in a matter-dominated universe is suppressed by a factor $(v/c)^2$. That is equivalent to say: since DM is non-relativistic, χ and h_{ij} must be very small with respect to Φ or even B_i .

As a matter of fact, V1.2 of GEVOLUTION implements an improved expansion of the metric perturbations, which has been presented in Adamek, Durrer & Kunz (2017). For our tests, we used the implementation of the original expansion, the one presented above. However, the improved expansion has been ported to LIBGEVOLUTION and will be used when analysing result on the past light cone. We do not expect the results presented in this paper to depend on the specific expansion used.

2.2 Motion of particles along geodesics

Massive particles move along geodesics, whose equation can be expressed as:

$$\begin{aligned} \frac{dx^i}{d\tau} = \frac{cp^i}{\sqrt{(mca)^2 + p^2}} + cB^i \\ + \frac{cp^i}{\sqrt{(mca)^2 + p^2}} \left(\Psi + \Phi \frac{2(mac)^2 + p^2}{(mac)^2 + p^2} \right), \end{aligned} \quad (8)$$

$$\frac{dp_i}{d\tau} = -c \left(p^n B_{n|i} + \Psi_{,i} \sqrt{(mca)^2 + p^2} + \frac{p^2 \Phi_{,i}}{\sqrt{(mca)^2 + p^2}} \right), \quad (9)$$

where p^i is the space part of the particle momentum and p its norm. The right-hand side in the last equation is the generalized force acting on the particles. The term proportional to $\Psi_{,i}$ becomes the Newtonian force in the limit of small velocities, while $p^n B_{n|i}$ represent the corrections due to *frame dragging* and the third term in parenthesis is a further relativistic correction.

The energy-momentum tensor is constructed from the knowledge of particle positions and momenta, but its computation depends on the perturbed metric. This means that, in equations (4), (6), and (7), the source terms on the right-hand sides depend on the potentials themselves. These implicit equations may be solved starting from the potentials at the previous time-step and solving the equations iteratively until convergence. The integration scheme that GEVOLUTION implements is simpler: at each time-step, the energy momentum tensor is computed using the potentials from the previous step, then the Poisson equations are solved to find the updated potentials, that will be used in the next time-step to compute the energy-momentum tensor.

The Newtonian limit is recovered when we consider Fourier modes larger than \mathcal{H}/c , and we further neglect B_i and consider $\Phi \ll 1$; then equation (4) becomes:

$$\Phi_{|n}^n = \frac{4\pi G a^2}{c^4} \Delta T^0{}_0 \quad (10)$$

while (8) and (9) become:

$$\frac{dx^i}{d\tau} = \frac{p^i}{ma}, \quad (11)$$

$$\frac{dp_i}{d\tau} = -\Phi_{,i} mc^2 a. \quad (12)$$

3 ALGORITHMS AND CODE INFRASTRUCTURE

3.1 GEVOLUTION

GEVOLUTION⁸ (Adamek et al. 2016) is an N -body relativistic cosmological code, written in C++ and parallelized with the MPI paradigm. The physical theory behind this code has been described at length in Section 2. Numerically, this code implements a PM scheme to follow the evolution of energy-momentum tensor perturbations. As in PM codes, the advantage of working with a single grid and using Fast Fourier Transforms (FFTs) to solve the Poisson-like equations for the fields is paid with a high cost in memory, of $\mathcal{O}(N^3)$, where N is the number of grid points per dimension.

GEVOLUTION can run in either *Newton* or *General Relativity* modes. The Newtonian gravity solver inverts the Laplace operator in the Poisson equation for the Newtonian potential, equation (10). When running the GR mode, the code solves equations (4), (6), and (7), which require the computation of the perturbed energy-momentum tensor. This is performed using a Cloud-In-Cell (CIC) scheme both for the density and for particle velocities; details are given in the presentation paper. Then the Hamiltonian forces to which particles are subjected are computed from equations (8) and (9).

GEVOLUTION solves the field equations in Fourier space, using a C++ library, called LATFIELD2 to operate FFTs on classical fields in massively parallel applications with distributed memory. LATFIELD2 provides a programming interface to perform operations on the fields, either in their real or Fourier space representations. This library implements FFTs of 3D fields whose memory is distributed among parallel processes, following a 2D uniform decomposition of space, in which each process owns in memory a portion of the grid with a *rod* shape (Daverio, Hindmarsh & Bevis 2015). In this way, LATFIELD2 overcomes the scaling limitations of a simpler 1D domain (*slab*) decomposition provided by the mainstream FFTW3 library.⁹ FFTW3 is used, however, to compute 1D FFTs.

3.2 GADGET-4

GADGET-4¹⁰ is a state-of-the-art TreePM N -body hydrodynamical cosmological code written in C++ (see Springel et al. 2021); it is massively parallelized in a distributed-memory paradigm using MPI.

As in most N -body codes, gravity in GADGET-4 is represented in the Newtonian limit, but the equations of motion are modified to take into account the Universe expansion, obtained by integrating the Friedmann equations separately. As mentioned above, this approach is consistent with GR in the Poisson gauge and gives the leading-order term of weak field expansion. This amounts to neglecting the metric degrees of freedom B_i , χ and h_{ij} and is valid on scales much smaller than the Hubble horizon. In a typical configuration that is convenient

for large cosmological volumes, the code solves for the forces acting on each particle, representing them as the sum of two contributions, one due to the interactions with nearby particles, computed with a Tree algorithm, and one due to long-range interactions, computed with a PM algorithm.¹¹

The Tree algorithm works by partitioning the space into cubic cells, called nodes; in turn, each node is recursively partitioned into eight children nodes down to a pre-determined maximum refinement level. A tree structure tracks the list of particles that are located within each node. This structure is used to speed up the computation of gravitational force on a particle: in a particle-particle (PP) integration scheme, this force is computed by adding up a series of $\vec{r} m/r^3$ terms, one for each particle pair, but we know that the accuracy of force evaluation does not depend strongly on the small-scale distribution of distant particles, so in the Tree scheme, the evaluation of gravity is performed by grouping particles that belong to the same node, under the condition that the node subtends a given aperture angle θ . PP computation is then used only for the nearest neighbours. This is equivalent to considering the leading order in a multipole expansion of the gravity force from particles belonging to a distant cell. While the construction of the Tree is expensive in terms of computing time, it allows to achieve $\mathcal{O}(N_p \log N_p)$ scaling for the force computation, where N_p is the total number of particles in the simulation. Thus the Tree is able to compute with high accuracy the short-wavelength modes of the gravitational interaction, while keeping the computational time low for large simulations. However, the Tree code is slow in integrating particle motions near the initial conditions when the departures from homogeneity are small. This is why it is often coupled with a PM code to speed up the first time-steps of a cosmological box.

The PM algorithm represents gravity through the gravitational potential field Φ , evaluated on a Cartesian cubic mesh of fixed size. The potential is found from the density field by solving the Poisson equation in Fourier space, while the force is computed from the gradient of the potential, obtained with a finite differences scheme. According to the Nyquist–Shannon theory, this implies that the information handled by the PM is limited to the long modes, up to the Nyquist frequency.

To combine the forces provided by the PM and Tree codes, the gravitational potential is split into the sum of two fields:

$$\Phi = \Phi^{(L)} + \Phi^{(S)}, \quad (13)$$

where $\Phi^{(L)}$ represents long-range modes from the PM, and $\Phi^{(S)}$ represents short-range modes from the Tree. Written in Fourier space (tilde on top of symbols denotes a Fourier transform), the Poisson equation reads:

$$\tilde{\Phi}_k = -\frac{4\pi}{k^2} \tilde{\rho}_k, \quad (14)$$

where ρ denotes the mass density. We can split the density as a sum of short-range and long-range terms, using Gaussian filters:

$$\tilde{\Phi}_k = -\frac{4\pi}{k^2} \tilde{\rho}_k (1 - \exp(-k^2 r_a^2)) - \frac{4\pi}{k^2} \tilde{\rho}_k \exp(-k^2 r_a^2). \quad (15)$$

The scale r_a is the one at which we split long- and short-range modes. We can obtain $\Phi^{(S)}$ by solving the modified Poisson equation for short

⁸<https://github.com/gevolution-code>

⁹<http://fftw.org/>

¹⁰<https://wwwmpa.mpa-garching.mpg.de/gadget4>

¹¹The code can work in other configurations (a non-cosmological volume, switching off the PM, enhancing the Tree part using multipole expansion) that are however not relevant for this paper.

modes:

$$\tilde{\Phi}_k^{(S)} = -\frac{4\pi}{k^2} \tilde{\rho}_k (1 - \exp(-k^2 r_a^2)), \quad (16)$$

and $\Phi^{(L)}$ by solving the modified Poisson equation for long modes

$$\tilde{\Phi}_k^{(L)} = -\frac{4\pi}{k^2} \tilde{\rho}_k \exp(-k^2 r_a^2). \quad (17)$$

The long-mode Poisson equation (17) is solved by the PM in Fourier space, so the convolution with the kernel is a simple multiplication. The Tree, on the other hand, works in real space, hence equation (16) has to be transformed; this can be done analytically, yielding:

$$\Phi^{(S)}(\vec{x}) = -G \sum_i \frac{m_i}{|\vec{x} - \vec{r}_i|} \operatorname{erfc}\left(\frac{|\vec{x} - \vec{r}_i|}{2r_a}\right) \quad (18)$$

3.3 GRGADGET

3.3.1 LIBGEVOLUTION library

In order to have a relativistic PM code working in GADGET-4, we developed a library that implements both the Newtonian and the relativistic PM algorithms of the monolithic GEVOLUTION code. This was done by forking the GEVOLUTION github repository into LIBGEVOLUTION, a library that is publicly available on GITHUB¹² under MIT license.

The rationale behind the development of LIBGEVOLUTION is to encapsulate GEVOLUTION's resources and methods into abstract objects. This yields several benefits. First, GEVOLUTION maintenance is eased by the logical modularization of the code, i.e. instead of a monolithic code with a unique workflow, we can divide GEVOLUTION into components (C++ classes and/or namespaces) with well-defined purposes. Secondly, we are allowed to re-use GEVOLUTION components within other applications, such as we do within GADGET-4 in the present paper.

We give here an overview of the library; the precise signature of all the defined functions, methods, and data structures is described in the technical documentation of the code. LIBGEVOLUTION is based on three cornerstones: (i) a particle container implemented through the class `Particles_gevolution`; (ii) a PM data structure named `particle_mesh`, templated on the particle container type, which can be used either as a `relativistic_pm` or a `newtonian_pm`; (iii) an executable application that uses the previous components to produce N -body simulations as the original code does. `particle_mesh` has to be understood as a container that is aware of the parallelization of the tasks and distribution of memory; it holds the gravitational fields, and it allows the user to compute the forces acting on the simulation particles. The user interface declared in `particle_mesh` consists of the following functions:

(i) `sample(...)` that builds the sources (density field or energy-momentum tensor) by sampling particle properties in the mesh;

(ii) `compute_potential(...)` that solves Poisson equations to compute the potential fields;

(iii) `compute_forces(...)` that computes the forces acting on particles.

`particle_mesh` is specialized to solve the Newtonian problem or the General Relativistic problem using class inheritance; Fig. 1 il-

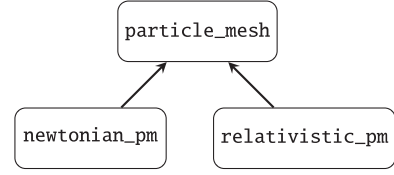


Figure 1. PM class hierarchy in LIBGEVOLUTION.

lustrates the class hierarchy of LIBGEVOLUTION's `particle_mesh`. The expert user will be able to specialize `particle_mesh` to his/her own needs, for example by deriving a PM that solves a modified gravity problem.

`newtonian_pm` is the specialization of `particle_mesh` that contains a real `Latfield2::Field` scalar field Φ_{Newton} and its complex `Latfield2::Field` Fourier transform $\tilde{\Phi}_{\text{Newton}}$, plus a `Latfield2::PlanFFT` that connects Φ_{Newton} with $\tilde{\Phi}_{\text{Newton}}$ through discrete Fourier transform. `relativistic_pm` is the specialization of `particle_mesh` that contains the above quoted degrees of freedom of the perturbed FLRW metric, Φ , B_i and χ . These are represented as real `Latfield2::Field`, with complex `Latfield2::Field` counterparts to represent their Fourier transforms and a `Latfield2::PlanFFT` for each field.

As a first testing phase, we run LIBGEVOLUTION, called with a simple wrapper, and the native GEVOLUTION code, applying them to the same set of initial conditions, checking that the results were identical both in the Newtonian and relativistic cases. Then we stripped down GADGET-4 by switching off the Tree code, and compared its results to the Newtonian results of LIBGEVOLUTION. It is necessary that this comparison gives nearly identical results if we want LIBGEVOLUTION to substitute the native PM code of GADGET-4 without loss of accuracy. To achieve a satisfactory match of the two PM codes, we had to change the GEVOLUTION scheme in a few points.

We started from V1.2 of GEVOLUTION, which implemented a first-order version of finite differences instead of the fourth-order scheme of GADGET-4. This resulted in a difference with GADGET-4 run on the same initial conditions, and in a per cent-level offset of the matter power spectrum on large scales at low redshift. We upgraded the computation of spatial derivatives to fourth order, in parallel with the GEVOLUTION developers that had noticed the same problem; our implementation is equivalent the most recent issue of GEVOLUTION (used, e.g. in Adamek et al. 2022). The upgrade is the following: let's consider the gravitational potential along one direction of the mesh, and let's call its values Φ_i , where the index i denotes its position along that direction. Its first derivative is computed with finite differences at the first order as:

$$\frac{\partial \Phi_i}{\partial x} = \frac{\Phi_{i+1} - \Phi_i}{h} + \mathcal{O}(h), \quad (19)$$

where h is the size of the mesh cell. Fourth-order Taylor expansion gives:

$$\frac{\partial \Phi_i}{\partial x} = 8 \frac{\Phi_{i+1} - \Phi_{i-1}}{12h} - \frac{\Phi_{i+2} - \Phi_{i-2}}{12h} + \mathcal{O}(h^4). \quad (20)$$

This has a smaller error of order $\mathcal{O}(h^4)$, so it achieves higher precision than (19) with the little cost of knowing the potential value at the second-nearest cell, which implies a negligible communication overhead.

Another improvement with respect to V1.2 of GEVOLUTION, which follows an implementation of GADGET-4, was the application of

¹²<https://github.com/Grgadget/gevolution-1.2>

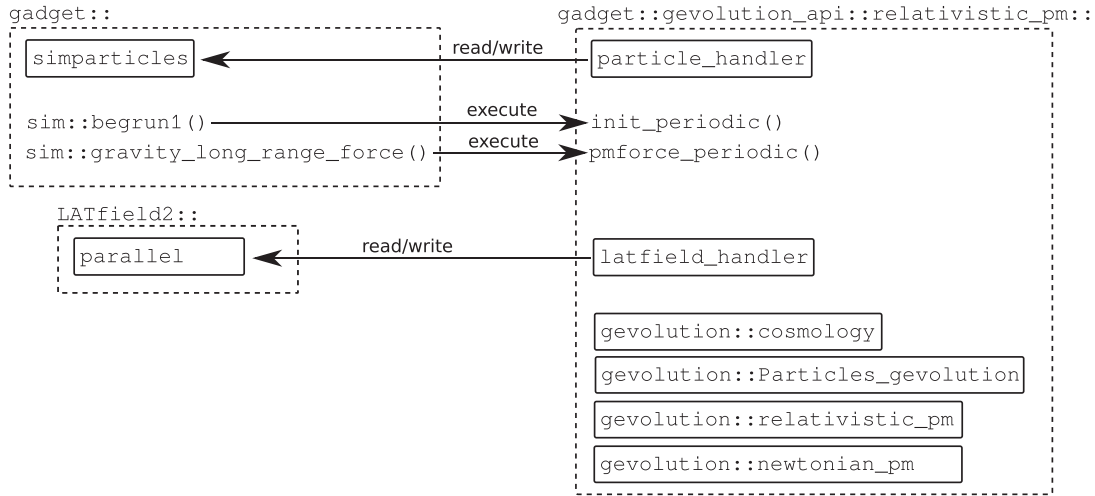


Figure 2. Diagram of resource ownership and relations for LIBGEVOLUTION integrated into GADGET-4’s workflow. Each solid box represent a memory resource (an instantiation of a variable type), while the dashed boxes indicate ownership. The newly developed code, represented in the right part of the diagram denoted with the namespace `gadget::gevolution_api`, consists of a class named `relativistic_pm` that owns a `particle_handler` object that reads and writes directly into `gadget::simparticles`, a `latfield_handler` that takes care of setting up and inspect the state of `LATfield2::parallel`, and some types defined in LIBGEVOLUTION, that are defined in `gevolution` namespace, like `cosmology`, `Particles_gevolution`, and `relativistic_pm`. The methods `sim::begrun1()` and `sim::gravity_long_range_force()` in `gadget::` interact with the `relativistic_pm` through their interface `init_periodic()` and `pmforce_periodic()`.

correcting filters to the density in Fourier space to compensate for CIC interpolation. Indeed, as discussed e.g. in Springel (2005) or Sefusatti et al. (2016), CIC interpolation at some finite order leads to some loss of power that can be compensated for in Fourier space using suitable kernels. This was applied both to the computation of the density and to the computation of energy-momentum tensor components in the relativistic case.

Lastly, to make the Newtonian PM scheme equivalent to that of GADGET-4, we changed the form of the discrete Laplacian operator in the Poisson equation solver from its original form

$$\nabla^2 \rightarrow -\frac{4N^2}{L^2} \left(\sin^2 \frac{\pi k_x}{N} + \sin^2 \frac{\pi k_y}{N} + \sin^2 \frac{\pi k_z}{N} \right), \quad (21)$$

described in Adamek et al. (2016), equation (C.5), to the form used in GADGET-4:

$$\nabla^2 \rightarrow -\frac{4\pi^2}{L^2} \left(k_x^2 + k_y^2 + k_z^2 \right). \quad (22)$$

3.3.2 Calling LIBGEVOLUTION from GADGET-4

The implementation of LIBGEVOLUTION in GADGET-4 was performed as follows. We created a new PM class with a similar interface as the original one in GADGET-4, so that it is initialized and executed with the same functions as GADGET-4, i.e. `init_periodic()` and `pmforce_periodic()`. A new class `relativistic_pm` was implemented within an `gadget::gevolution_api` namespace, avoiding to use the wider `gadget` namespace to make a clear distinction of purpose between the original GADGET-4 code and our additional features. This `relativistic_pm` class acts much like a mediator taking information in and out of `gadget` simulation particles, processing the correct units conversion and calling the methods on `gevolution` namespace. Fig. 2 shows a diagram that summarizes the contents of this PM class, its relation with GADGET-4’s resources and the entry points for `gevolution`’s api.

`relativistic_pm` consists of:

(i) A variable of type `simparticle_handler` that acts as a wrapper for providing particle information from GADGET-4’s `simparticles` global variable and writing back the data produced by `gevolution`’s PM.

(ii) A variable of type `latfield_handler` that takes care of correctly initializing LATFIELD global state. Indeed, while GADGET-4 can run with any number of MPI processes, LATFIELD has limitations that depend on the number of grid points in the PM. `latfield_handler` also takes care of creating a sub-communicator from GADGET-4’s MPI global communicator that satisfies the constraints set by LATFIELD.

(iii) A variable of type `gevolution::cosmology` that contains the parameters for the background evolution.

(iv) A container of type `gevolution::Particles_gevolution` that holds particle information, stored according to their location on the PM grid.

(v) Variables of type `gevolution::relativistic_pm` and `gevolution::newtonian_pm` that perform the actual PM computations, i.e. construct the sources, either density or the components of the energy-momentum tensor, compute the gravitational potential or the metric perturbation fields and the forces that act upon the particles.

(vi) The methods `pm_init_periodic` and `pm_force_periodic` for initialization and execution of the PM, respectively.

3.3.3 Kick and drift operators

In order to keep the Hamiltonian character of the equations of motion in GADGET-4, we have to describe the state of each particle through its position and momentum, not velocity. Following a leap-frog scheme, the momentum should be updated with a *kick* operation using the full relativistic equations (8) and (9). However, velocities in GADGET-4 are to be interpreted as momenta (per unit mass) of non-relativistic particles in the Newtonian limit. Then we redefine the GADGET-4 *kick*

and *drift* operators assuming non-relativistic matter, $p \ll mca$, and further neglecting the very small contribution coming from χ :

$$\frac{dx^i}{d\tau} = \frac{p^i}{ma} (1 + 3\Phi) + cB^i, \quad (23)$$

$$\frac{dp_i}{d\tau} = -cp^n B_{n|i} - \Phi_{,i} mc^2 a. \quad (24)$$

The right hand side of (24) is what we call *force*.

3.3.4 Adding long-range and short-range forces

To combine the forces computed with the relativistic PM and GADGET-4's Newtonian Tree, we have extended the idea of the TreePM coupling. From equation (13), one obtains that the force acting on a particle in a TreePM scheme consists of two terms:

$$\vec{F} = S_{r_a} \left[\vec{F}_{\text{Newton}}^{\text{Tree}} \right] + L_{r_a} \left[\vec{F}_{\text{Newton}}^{\text{PM}} \right]. \quad (25)$$

The first term is the force computed using the Tree on which an exponential high-pass filter S_{r_a} is applied, leaving short-wavelength modes. The second term corresponds to the PM force on which the complementary low-pass filter L_{r_a} is applied to leave long-wavelength modes. The symbols S_a and L_a formally denote these linear operators:

$$S_{r_a}[f](\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \tilde{f}_{\vec{k}} (1 - \exp(-k^2 r_a^2)) \exp(-i\vec{k} \cdot \vec{r}), \quad (26)$$

and

$$L_{r_a}[f](\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \tilde{f}_{\vec{k}} \exp(-k^2 r_a^2) \exp(-i\vec{k} \cdot \vec{r}). \quad (27)$$

The *grid smoothing scale* r_a scales with the PM mesh size, and its value is optimized in GADGET-4, in a way that will be tested below, to minimize the impact of the two different treatments of the gravitational force.

In order to account for the relativistic dynamics while preserving the match between tree and PM contributions that is valid in the Newtonian case, we choose the following strategy: GADGET-4 calls both `newtonian_pm` and `relativistic_pm`, the Newtonian value of the force is added to the Tree force as in a standard Newtonian simulation, while the difference between the Newtonian and the relativistic forces is added on top as a correction, but filtered on a different scale r_b , that we call *gr-smoothing scale*. Equation (25) then becomes:

$$\vec{F} = S_{r_a} \left[\vec{F}_{\text{Newton}}^{\text{Tree}} \right] + L_{r_a} \left[\vec{F}_{\text{Newton}}^{\text{PM}} \right] + L_{r_b} \left[\vec{F}_{\text{GR}}^{\text{PM}} - \vec{F}_{\text{Newton}}^{\text{PM}} \right]. \quad (28)$$

The case $r_a = r_b$ would correspond to simply adding the relativistic force to the Tree:

$$\vec{F} = S_{r_a} \left[\vec{F}_{\text{Newton}}^{\text{Tree}} \right] + L_{r_b} \left[\vec{F}_{\text{GR}}^{\text{PM}} \right]. \quad (29)$$

However, while the size of r_a , which regulates the match between Newtonian Tree and PM forces, is very well tested within GADGET-4, the optimal value of r_b is to be found; we will show in the next section that using r_b larger than r_a allows us to achieve percent accuracy at small scales.

The grid smoothing scale r_a has a direct impact on the code performance because the short-range forces decay exponentially with the distance to the source as shown in equation (18); this fact is exploited by the tree code that ignores contributions to the force coming from sources at distances much larger than r_a (Springel 2005). On the contrary, the *gr-smoothing scale* r_b is implemented

Table 1. Cosmological simulation configurations used to validate GRGADGET.

Name	N_p (particles)	N (PM grid points)	L (box size)
N64	64^3	64	1 Gpc h^{-1}
N256	256^3	256	1 Gpc h^{-1}
high_res	512^3	512	500 Mpc h^{-1}

only with a function convolution in the PM; thus it has no direct impact on the speed of the calculations. However, the value of r_b must be within an upper bound set by the physical scale where the Newtonian forces are numerically similar to GR in our gauge and a lower bound around $3L/N$ set to reduce the numerical noise produced by the PM discretization (where L is the boxsize and N is the number of grid points per dimension of the PM; these constraints will be discussed in details in the following sections). Therefore, if r_b is fixed, the mesh resolution must be fine enough (for example increasing the value of N) to allow $3L/N \leq r_b$; in this way, r_b has an indirect impact on performance.

4 VALIDATION

The GRGADGET code has been validated by running it on a few realizations of initial conditions, listed in Table 1. These were generated with GADGET-4's `ngenic` code at $z = 19$, starting from a linear power spectrum generated with CAMB¹³ and with cosmological parameters consistent with Planck 2018 result (Planck Collaboration VI 2020): $\Omega_b h^2 = 0.0223$, $\Omega_c h^2 = 0.120$, $H_0 = 67.3 \text{ km s}^{-1} \text{ Mpc}^{-1}$, $A_s = 2.097 \times 10^{-9}$, and $n_s = 0.965$.

4.1 GEVOLUTION and GADGET-4 original codes

As already discussed in Section 3.3.1, the `newtonian_pm` implementation in V1.2 of GEVOLUTION computes the Newtonian forces differently from those obtained with GADGET-4's PM. Before implementing LIBGEVOLUTION as the PM engine of GADGET-4, we need to make the two algorithms work in the same way.

To this aim, we have run a set of simulations with the configuration N64 (described in Table 1) with a small number of particles $N_p = 64^3$ to be able to compute forces using a straightforward PP scheme, which can be taken as the true force that we are trying to approximate. The same initial conditions at $z = 19$ have been fed to both GADGET-4 (with Tree either on or switched off to have a pure PM run) and GEVOLUTION (in Newtonian mode) codes. At later times, $z = 8$ and $z = 0$, we have written snapshots of the forces that the simulation particles experience, separating the PM and the TreePM components; we have then compared those to the true Newtonian force computed with the PP scheme. The data we have obtained are summarized in the plots shown in Fig. 3. We have binned particles according to the value of the true force; then for each bin, we have computed the mean (coloured lines) and standard deviation (shaded regions) of the difference between the force computed with approximate methods (PM or TreePM) and the true value. Forces are given in GADGET-4's default units, which is actually acceleration, measured in units of $10H_0 \text{ km/s} = h \text{ km}^2 \text{ s}^{-2} \text{ kpc}^{-1}$. The green line shows the PM result using the original GEVOLUTION code (the true force is anyway computed with GADGET-4 and matched particle by

¹³<https://camb.info/>

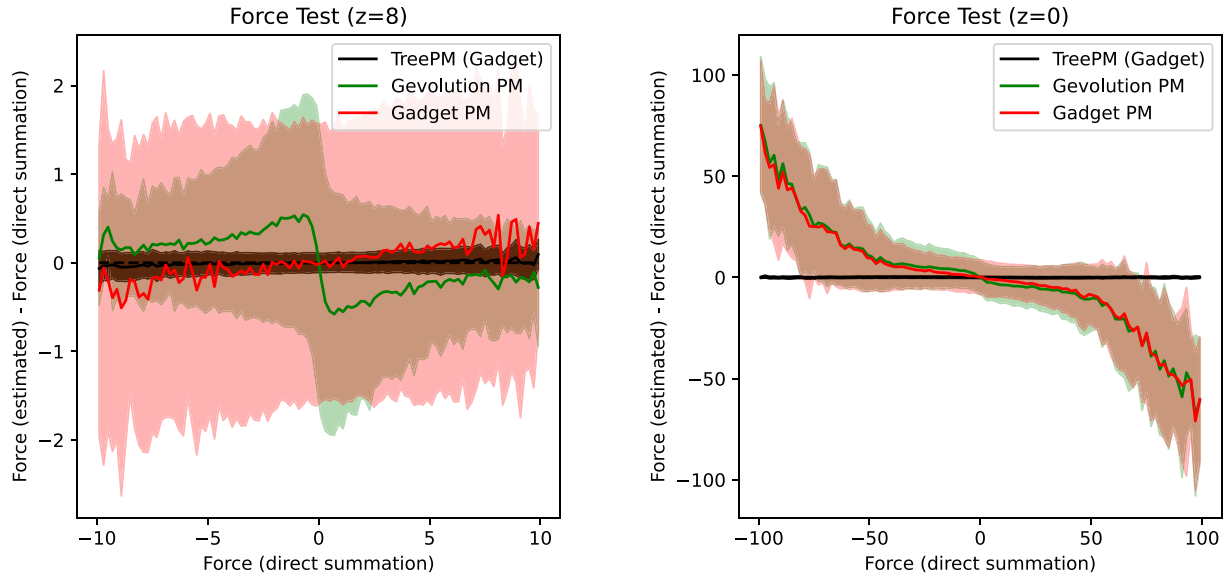


Figure 3. Difference of gravity force with respect to the true PP value, binned according to the true force, for $N64$ initial conditions, at $z = 8$ (left-hand panel) and $z = 0$ (right-hand panel). Lines represent the mean value of force difference in the bins, with colours explained in the legend; the shaded regions give the standard deviation of the corresponding force difference.

particle), while the red line is obtained from a pure PM using GADGET-4's original code. The black line gives the TreePM method precision, obtained using GADGET-4.

Looking at the red and green lines (and their shaded areas), we find two known results. First, the TreePM method produces far less bias and dispersion when estimating forces; for instance, in the left-hand panel of Fig. 3 the error is of the order¹⁴ of $0.1 h \text{ km}^2 \text{ s}^{-2} \text{ kpc}^{-1}$, while in the right-hand panel, it is larger but barely visible when compared with the other curves. Second, while the PM force has low bias but a much larger variance than the TreePM one at high redshift, at low redshift, i.e. at higher level of non-linearity, it underestimates the value of the Newtonian force as its magnitude increases. This underestimation is due to the failure of PM in resolving interaction at scales smaller than the grid resolution.

When comparing GEVOLUTION PM and true forces, we notice an S-shaped feature in the plot, much more visible at high redshift. As anticipated in Section 3.3.1, this is mostly due to the first-order interpolation used to find the gradient of the potential in the code version that we tested.

In Fig. 4, we show the matter power spectra¹⁵ obtained at $z = 0$ from a set of larger simulations with the configuration N256 (see Table 1). The red solid line shows the result obtained with the original GADGET-4 code with its TreePM method, while the red dotted line shows the results obtained by switching off the Tree so that the forces are computed using the PM alone. The green lines show results obtained with the latest *develop* version of GEVOLUTION that implements higher-order schemes for finite differences; the dotted line gives results obtained with `GRADIENT_ORDER = 1` and is identical to the result obtained with V1.2 of GEVOLUTION; the green solid line uses `GRADIENT_ORDER = 2` that corresponds to

¹⁴This quantification is in code units; we can take this value as a reference for a high accuracy gravity solver.

¹⁵In this paper, all particle power spectra were computed using `Power14` code presented in Sefusatti et al. (2016). Unless otherwise stated, all power spectra are computed up to the Nyquist frequency of the PM mesh.

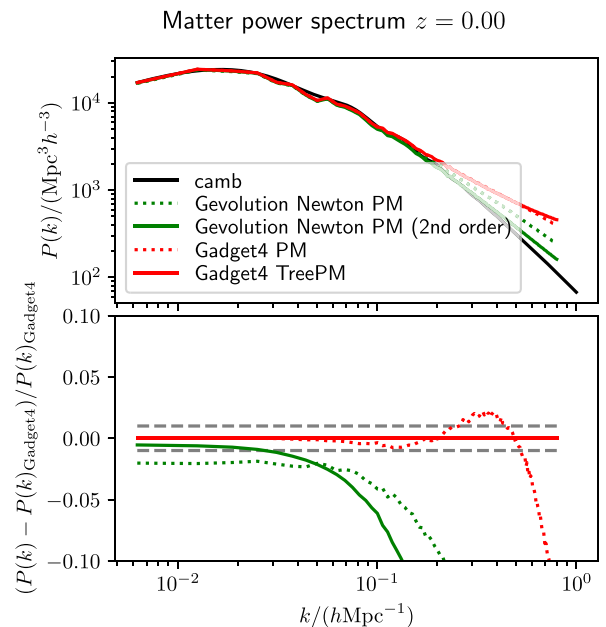


Figure 4. Matter power spectrum of N256 cosmological simulations. The lower panel shows residuals with respect to GADGET-4's original code (in red) used as baseline. The black line shows the linear power spectrum obtained with CAMB. Red lines show results obtained with GADGET-4, with the Tree part on (solid line) or switched off (dotted line). Green lines show results obtained with GEVOLUTION in Newtonian configuration, with finite differences at first order (dotted line) or second order (solid line).

a second-order scheme. These power spectra show that the matter distribution in GEVOLUTION using first-order gradients loses power in what seems to be a uniform trend for large-scale modes. This is a behaviour that is not inherent to the PM nature of the code, since that type of numerical approximation should predict very well the linear evolution at large scales; indeed, the higher-order scheme

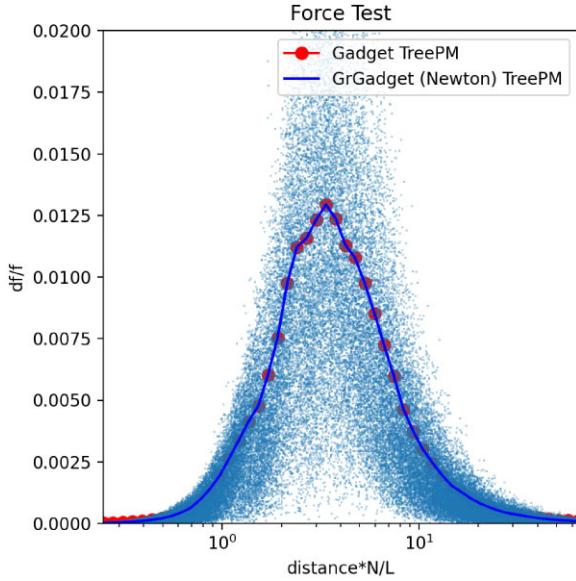


Figure 5. Forces due to a point source: the points are test particles located at different distances (in units of the mesh resolution L/N) from the source, and the lines represent the RMS of the difference between real and TreePM forces in different distance bins. The red line corresponds to GADGET-4 original TreePM, while the blue line was obtained with GRGADGET in Newtonian mode. As for the grid smoothing scale, the default value was used: $r_a = 1.25L/N$. For this test, we have used $N = 256$ and $L = 1 \text{ Gpc } h^{-1}$.

recovers power on large scales to sub-per cent accuracy. Conversely, GADGET-4’s PM and TreePM agree very well at wavenumbers below $k \sim 0.1 h \text{ Mpc}^{-1}$ scale.

The higher-order differentiation worsens the loss of power of GEVOLUTION for high values of k , which is not present in GADGET-4. This can be explained as a consequence of the particle-to-mesh sampling and mesh-to-particle interpolation described in Section 3.3.1. As discussed there, GADGET-4’s PM corrects for these effects, resulting in a power spectrum that degrades only at very high values of k as we approach the Nyquist frequency, while producing a ~ 2 per cent overcorrection at $k \sim 0.4 h \text{ Mpc}^{-1}$.

After implementing the higher-order differentiation scheme, the correction for the loss of power discussed above and the change in the discrete Laplacian operator (Section 3.3.1), the results of native GADGET-4 and LIBGEVOLUTION PMs become indistinguishable.

4.2 Newtonian forces

We have tested our implementation of the GRGADGET code by running a standard test in GADGET-4: we create an N -body configuration in which there is a single massive particle in the entire simulation box, while other massless test particles are placed at different distances from the first. In this setting, the exact value of the force on each particle is known, hence one can compare the numerical results coming from the TreePM algorithm to the analytical solution.

The results are shown in Fig. 5, where each dot represents a test particle. The x-axis gives the distance to the massive particle that sources the gravitational field in units of the PM resolution (L/N), while the y-axis gives the corresponding absolute value of the relative difference of the true and estimated forces acting on the test particle. The red and blue lines correspond to the mean value of force residuals, for particles binned into distance bins; the red line

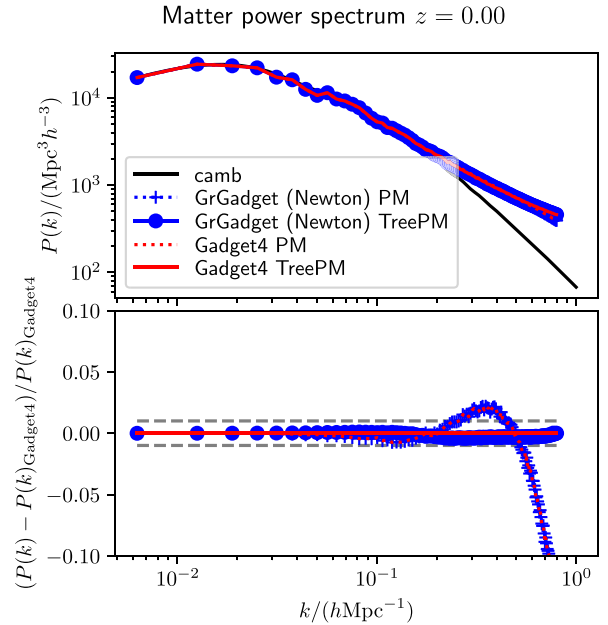


Figure 6. Matter power spectrum of four simulations starting from the same initial conditions `high_res`: blue lines give results for GADGET-4 original code, and red lines give results for GRGADGET. In both cases, dotted lines refer to runs with PM-only, and solid lines refer to runs with full TreePM.

denotes the statistics obtained from a simulation using GADGET-4’s original TreePM implementation, and the blue line was produced using GRGADGET, in this case with the Newtonian gravity engine.

This figure shows that the accuracy with which the TreePM code reproduces the gravitational force is at worst at per cent level on scales of a few mesh cells, corresponding to the scale where the PM and Tree contributions are matched, and gets very accurate in the limits where either the Tree (small scales) or the PM (large scales) dominates.

GADGET-4’s and GRGADGET’s Newtonian PMs show basically the same accuracy, even though their PM implementations are very different.

In Fig. 6, we show the matter power spectra of a set of N_{256} simulations (see Table 1). In this case, we are comparing the matter clustering of GRGADGET, in blue (with Newtonian forces for testing purposes), against GADGET-4, in red. In agreement with the previous test of force differences, we find that both codes produce the same matter power spectrum up to floating-point errors. This is verified both in the case of simulations computing forces using a pure PM and in the case of TreePM.

4.3 Relativistic simulations with GRGADGET

We present here results obtained by running GRGADGET with `relativistic_pm`, comparing them with the corresponding relativistic version of GEVOLUTION. We expect that the power spectrum of the matter density displays some relativistic features at large scales due to terms preceded by \mathcal{H} in the field equation (4), while at small scales, results should be compatible with GADGET-4’s Newtonian simulations. However, the matter power spectrum shown here is not an observable quantity, so this comparison is just meant to give a first validation of the results. A more thorough comparison of observables reconstructed on the past light cone will be presented in a future paper.

Fig. 7 shows the matter power spectra for a series of N_{256}

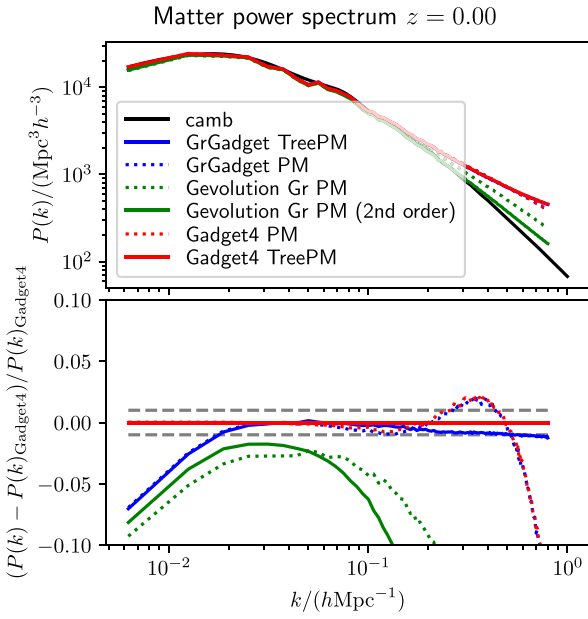


Figure 7. Matter power spectrum of GADGET-4, GEVOLUTION, and GRGADGET runs, the last code being run in relativistic mode. The upper panel shows the absolute value and the lower panel the relative difference with respect to GADGET-4’s TreePM. The black line gives the linear matter power spectrum; red and blue lines give GADGET-4 and GRGADGET results, with full TreePM forces (solid lines) or with the Tree switched off (dotted lines). Green lines give GEVOLUTION results, dotted line referring to first-order finite differences (GRADIENT_ORDER = 1) and solid line referring to second-order calculation (GRADIENT_ORDER = 2).

simulations (see Table 1). In this case, GEVOLUTION and GRGADGET are run in GR mode. The parameter that regulates the scale of the relativistic correction [equations (28) and (29)] is set to $r_b = 6 L/N \approx 23 \text{ Mpc } h^{-1}$, i.e. the relativistic corrections of the PM method are smoothed at distances below six grid cells. The plot shows that relativistic PM-only simulations, GRGADGET (blue dotted

line) and GEVOLUTION (green lines) are compatible on large scales ($k < 0.03h \text{ Mpc}^{-1}$) up to a small per cent-level difference that it is likely caused by the use of different orders for finite-difference gradient; indeed, going from first- to second-order differences (from dotted to solid green line), the power spectrum gets nearer to GRGADGET’s fourth-order one. The plot also confirms that our combination of Tree and PM forces in the relativistic weak field limit with GRGADGET (blue solid line) reproduces the Newtonian non-linear features to sub-per cent level at small scales, that is for $k > 0.1h \text{ Mpc}^{-1}$; here, GADGET-4 (red solid line) is again our reference for the non-linear clustering.

Being designed for the use of Fourier methods from the beginning, LIBGEVOLUTION offers an interface for the computation of the power spectrum of the fields defined through the library’s interface. Thus, we can also extract and analyse the power spectra of the individual components of the metric perturbations from the relativistic simulations. Figs 8 and 9 show the power spectra of the relativistic potentials, Φ , B_i , and χ , for a high-resolution configuration `high_res` (see Table 1). These plots show a comparison of PM (blue lines) and TreePM (red lines) simulations. The power spectrum of the gravitational potentials converge for both methods on large scales. However, below $1 \text{ Mpc } h^{-1}$, the PM-only simulation loses power with respect to the TreePM one; the differences can reach up to 40 per cent as we approach the Nyquist frequency. This pattern is equally found for the scalar fields Φ and χ , as well as for the individual components of B_i .

The right plot in Fig. 8 helps to understand the reason behind this result. Generally speaking, energy density, momentum density, and their respective density current (the components of the Energy-Momentum tensor) are sources of the metric perturbations. Even though those quantities, as fields, are found at discrete positions of space defined by the mesh, their values are computed by sampling the energy and momentum carried by the particle distribution, which contain information on the clustering due to the short-range interactions (through the Tree) that goes well below the mesh resolution L/N . Therefore, TreePM simulations, having power on scales well smaller than the PM mesh, give a better representation of the source of metric perturbation, and thus allow to recover power at frequency modes right below Nyquist. Fig. 8 highlights the particular case of T^0_0 (the matter density) as a source for Φ ; by comparing T^0_0

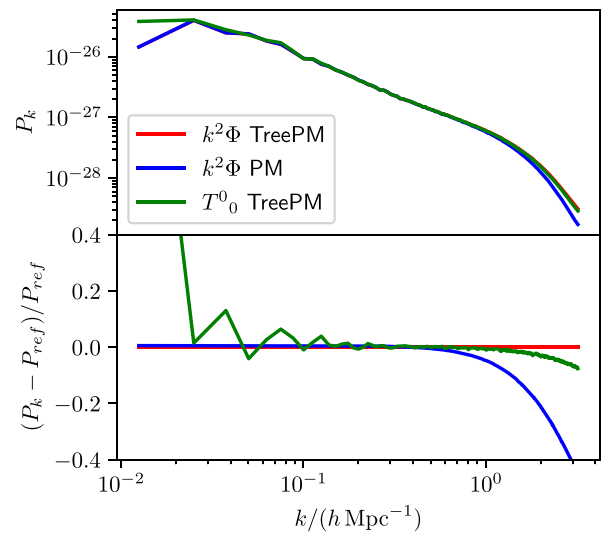
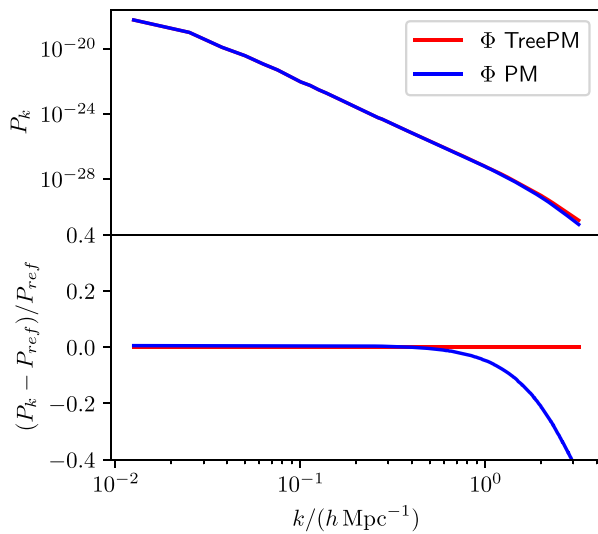


Figure 8. In the left plot: power spectrum of the metric perturbation Φ in a `high_res` simulation obtained with GRGADGET. In the right plot: power spectrum of $k^2\Phi$ and T^0_0 . For modes well below the Hubble horizon and small perturbations, it should be verified that $k^2\Phi \approx \hat{T}^0_0$.

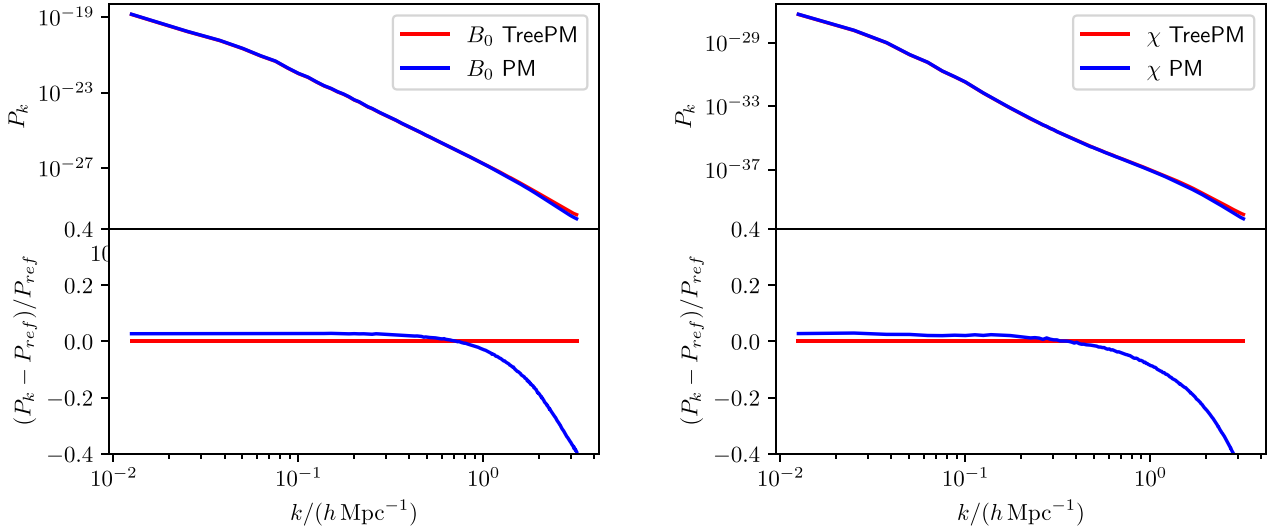


Figure 9. In the left plot: power spectrum of the metric perturbation B_i (the x component) in a `high_res` simulation obtained with GRGADGET. In the right plot: power spectrum of χ .

with $k^2\Phi$, we are verifying the Poisson equation $k^2\tilde{\Phi} \approx \tilde{T}^0_0$ that is valid for wavelengths below the Hubble horizon. This confirms that the presence of small-scale clustering in the particle distribution propagates to the gravitational fields up to the maximum resolution that the PM allows. The same thing is visible in the vector modes B_i and in χ (Fig. 9), where we also notice a small, few-per-cent mismatch on large scales. These fields are known to give sub-per cent effects on observables, so this difference, that is likely due to some degree of numerical mode coupling, is non considered as a problem.

In Fig. 10, we show how the matter power spectrum obtained using GRGADGET is affected by the choice of the gr-smoothing scale parameter r_b . We have used an `N256` box configuration to perform this test and tested values of $r_b = 1.5, 3,$ and 6 in units of $L/N \approx 4 \text{ Mpc } h^{-1}$. We find that large-scale power is independent of the value of r_b parameter; structures on scales below the PM resolution are resolved by the Tree algorithm, hence for $k > k_{\text{Nyquist}}$, there is a convergence of all simulations to a common non-linear power spectrum tail. It is in the medium to small scales $k_{\text{Nyquist}} > k > 0.2 \text{ Mpc}^{-1} h$ that we notice differences in the power spectrum above the ~ 1 per cent (dashed grey line). For small values of r_b ($\sim 1.5 L/N$), we obtain discrepancies in the power spectrum at $k \sim 0.5 \text{ Mpc}^{-1} h$ that can be as large as 5 per cent and indicate the limitations of our force summation scheme [equation (28)]. A value of $r_b = 3 L/N$ or possibly higher is needed to obtain a good compatibility of GRGADGET and GADGET-4 for all modes greater than $0.1 \text{ Mpc}^{-1} h$, where relativistic features in the matter clustering is negligible.

The last test we present here regards the convergence of the numerical results for increasing resolution. Fig. 11 shows the matter power spectrum obtained from running GADGET-4's TreePM (red lines), GRGADGET with PM-only (blue dotted lines) and GRGADGET with TreePM (blue continuous line). These various code configurations were run with different combinations of the number of grid points per dimension $N = 256, N = 512$ and box length $L = 250, 500, 1000,$ and $2000 \text{ Mpc } h^{-1}$; the number of particles was fixed as $N_p = N^3$. In all cases, we have set the PM smoothing scale to $r_a = 1.5 L/N$ and the gr-smoothing scale to $r_b = 3 L/N$. It can be observed with the finest resolution, in the top plots, that there is a matching between GR and Newtonian dynamics in the small scales. Then as the mesh

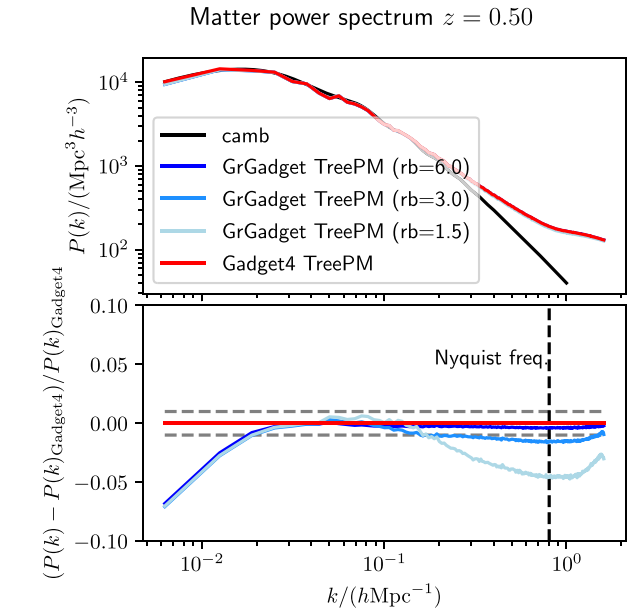


Figure 10. Power spectrum of matter density for GADGET-4 and GRGADGET, on a `N256` simulation configuration. The upper panel shows the absolute value and the lower panel the relative difference with respect to GADGET-4's TreePM. Different shades of blue indicate different values of the gr-smoothing scale parameter $r_b = 1.5, 3,$ and 6 in units of L/N . The PM smoothing scale is $r_a = 1.5 L/N$. The power spectra in this plot are computed beyond the Nyquist frequency to show the convergence of the matter distribution correlations for distances below the grid resolution, the Tree regime.

size becomes coarser, in the middle plots, some discrepancies in the power spectrum start to appear which become more evident for even coarser meshes, in the bottom plots. This mismatch may be caused by $r_b = 3 L/N$ moving towards larger scales, so that the assumption that PM forces are Newtonian on the small-scale breaks. Indeed, while with $L/N = 1 h^{-1} \text{ Mpc}$ ($r_b = 3 h^{-1} \text{ Mpc}$), the scales where relativistic effects become evident in the matter power spectrum and the scales where the pure PM prediction starts to deviate from TreePM are well

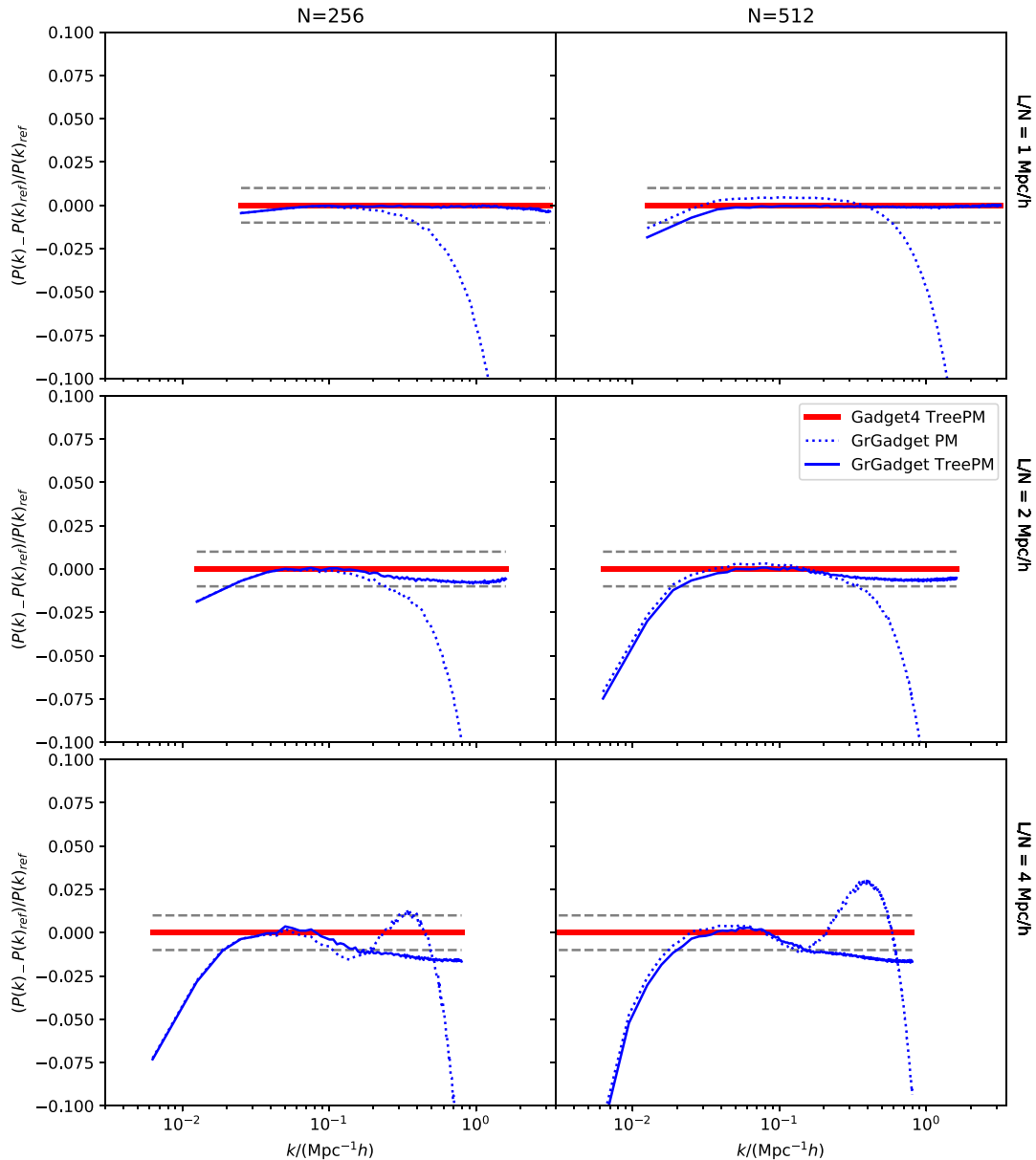


Figure 11. Matter power spectrum from cosmological simulations at $z = 0$ using GRGADGET (the blue lines) and compared to GADGET-4 (the red line) at $z = 0$. The dotted line is obtained with a simulation in which only the PM is used to compute forces. The plots show the relative difference with respect to the power spectrum obtained with GADGET-4. The left column corresponds to simulations with $N = 256$ grid points per dimension, while for the right column $N = 512$. The boxsize changes along the ranks so that for the top plots the resolution is the highest $L/N \approx 1 \text{ Mpc } h^{-1}$, in the middle $L/N \approx 2 \text{ Mpc } h^{-1}$ and the bottom plots correspond to $L/N \approx 4 \text{ Mpc } h^{-1}$. In all cases $r_a = 1.5 L/N$ and $r_b = 3 L/N$. The grey dashed line indicate a 1 per cent error.

separated, for larger L/N values the two scales get nearer, indicating that the assumption of pure Newtonian forces on the mesh scale may not be very good. This conclusion is apparently at variance with the discussion of Fig. 10, where a larger value of r_b was preferred; however, that figure refers to $L/N = 1$ and is shown at $z = 0.5$, where clustering is a bit weaker. We thus recommend to work with mesh sizes of $L/N \sim 1 \text{ Mpc } h^{-1}$.

5 CONCLUSIONS

We have constructed a relativistic TreePM code, which we call GRGADGET, where the large-scale contribution to the gravitational force is computed using the relativistic C++ PM library LIBGEVOLUTION,

based on GEVOLUTION code, while gravity coming from small scales is computed by the Tree code of GADGET-4. The code works under the assumption that, in the context of cosmological simulations, DM can be treated non-relativistically and then the equations of motion of tracer particles tend to the Newtonian limit at scales well below the Hubble horizon. Following the GEVOLUTION approach, we use a weak field approximation of GR, where the perturbations of the space-time metric with respect to FLRW background are encoded as fields and simulated by the PM.

With respect to the pure PM implementation of GEVOLUTION, the predictive power of GRGADGET gives an improvement even on the scales sampled by the mesh. This is due to the fact that the energy-momentum tensor, which sources the equations of the fields

that represent the perturbations of the metric, is computed from a fully non-linear distribution of particles, with gravity being resolved down to a much smaller softening length and not down to the mesh size. This may be very useful, e.g. when assessing the possibility of detecting the frame-dragging effect in the presence of vorticity (Bruni, Thomas & Wands 2014) that arises mainly from the orbit crossing of particles in the non-linear regime.

Comparing the matter power spectrum from GRGADGET simulations with that of original GADGET-4 and GEVOLUTION codes, we conclude that the code produces consistent results as long as the gr-smoothing parameter is greater than approximately three grid cells; i.e. $r_b \geq 3L/N$, see Fig. 10, this is a scale set by numerical stability. There is also a physical upper bound on r_b that arises when the Newtonian dynamics differs from the GR equations in our gauge, leading to a mismatch of GR forces computed in the PM and the Newtonian forces from the Tree. From the plots in Fig. 11, we can conclude that our scheme starts to break when $r_b \approx 6 \text{ Mpc } h^{-1}$. Hence we recommend setting $3L/N \leq r_b \leq 6 \text{ Mpc } h^{-1}$. This constraint sets a lower limit on the resolution of the PM: $L/N \leq 2 \text{ Mpc } h^{-1}$, meaning for example that a simulation with a boxsize of $L = 4 \text{ Gpc } h^{-1}$ must have at least $N = 2000$ grid cells per dimension.

Furthermore, this code is a development of the widely used GADGET-4 code, and because the PM sector of the code is called only by the computation of the gravity force, our code can be easily extended to simulations of galaxies or galaxy clusters by switching on the hydrodynamics, star formation, and feedback sectors. All the physics described by these sectors can safely be treated in the Newtonian limit; one should in principle add thermal energy of gas particles to the energy-momentum tensor, but while this extension is straightforward, it is likely to provide a negligible contribution.

This is, for our group, a further step in the construction of an ecosystem of simulation codes and post-processing tools for modelling the evolution of structure in the Universe, with the aim of making predictions for precision cosmology. Sub per cent accuracy in cosmological predictions, which matches the smallness of the statistical error that will be obtained with forthcoming galaxy surveys mentioned in the Introduction, can only be obtained taking into account relativistic effects (e.g. Lepori et al. 2020), and we can foresee that a self-consistent treatment of these effects (to within the required accuracy) will soon become the standard in cosmological simulations. These effects can also be added by post-processing Newtonian simulations, but a validation of these procedures requires the comparison against a more self-consistent approach. Conversely, a large community is developing GEVOLUTION in the direction of adding modifications of gravity (Hassani et al. 2019, 2022; Reverberi & Daverio 2019; Hassani & Lombriser 2020; Hassani, Adamek & Kunz 2021; Christiansen et al. 2023) whose formulation is typically worked out in a general relativistic context. This line of development, coupled with a Newtonian treatment of modified gravity in the Tree code, would be precious in the formulation of tests of gravity, because relativistic effects may hide smoking-gun features of specific classes of modified gravity theories.

ACKNOWLEDGEMENTS

We thank Julian Adamek for many fruitful discussions on GEVOLUTION, Volker Springel for his comments on an early draft, and Francesca Lepori, Marco Bruni, Marco Baldi, and Emilio Bellini for discussions. Simulations were performed with the HOTCAT system of INAF (Bertocco et al. 2020; Taffoni et al. 2020). PM acknowledges partial support by a *Fondo di Ricerca di Ateneo* grant of University of Trieste.

DATA AVAILABILITY

The simulation codes presented in this paper are publicly available on GITHUB in the following path: <https://github.com/Grgadget>.

REFERENCES

- Abbott B. P. et al., 2016, *Phys. Rev. Lett.*, 116, 061102
 Adamek J., Durrer R., Kunz M., 2014, *Class. Quantum Gravity*, 31, 234006
 Adamek J., Daverio D., Durrer R., Kunz M., 2016, *J. Cosmol. Astropart. Phys.*, 2016, 053
 Adamek J., Durrer R., Kunz M., 2017, *J. Cosmol. Astropart. Phys.*, 2017, 004
 Adamek J. et al., 2022, preprint (arXiv:2211.12457)
 Alam S. et al., 2021, *J. Cosmol. Astropart. Phys.*, 2021, 050
 Barnes J., Hut P., 1986, *Nature*, 324, 446
 Barrera-Hinojosa C., Li B., 2020, *J. Cosmol. Astropart. Phys.*, 2020, 007
 Barrera-Hinojosa C., Li B., Bruni M., Hua He J., 2021, *MNRAS*, 501, 5697
 Bartelmann M., Schneider P., 2001, *Phys. Rep.*, 340, 291
 Bertocco S. et al., 2020, in Pizzo R., Deul E. R., Mol J. D., de Plaa J., Verkouter H., eds, ASP Conf. Ser. Vol. 527, *Astronomical Data Analysis Software and Systems XXIX*. Astron. Soc. Pac., San Francisco, p. 303
 Borzyszkowski M., Bertacca D., Porciani C., 2017, *MNRAS*, 471, 3899
 Bruni M., Thomas D. B., Wands D., 2014, *Phys. Rev. D*, 89, 044010
 Capozziello S., De Laurentis M., 2012, *Ann. Phys.*, 524, 545
 Chisari N. E., Zaldarriaga M., 2011, *Phys. Rev. D*, 83, 123505
 Christiansen Ø., Hassani F., Jalilvand M., Mota D. F., 2023, preprint (arXiv:2302.07857)
 DESI Collaboration, 2016, preprint (arXiv:1611.00036)
 Daverio D., Hindmarsh M., Bevis N., 2015, *Latfield2: A c++ Library for Classical Lattice Field Theory*, preprint (arXiv:1508.05610)
 Doré O. et al., 2014, preprint (arXiv:1412.4872)
 Event Horizon Telescope Collaboration, 2019, *ApJ*, 875, L1
 Hassani F., Lombriser L., 2020, *MNRAS*, 497, 1885
 Hassani F., Adamek J., Kunz M., Vernizzi F., 2019, *J. Cosmol. Astropart. Phys.*, 2019, 011
 Hassani F., Adamek J., Kunz M., 2021, *MNRAS*, 500, 4514
 Hassani F., Adamek J., Durrer R., Kunz M., 2022, *A&A*, 668, A56
 Hernández-Aguayo C., Arnold C., Li B., Baugh C. M., 2021, *MNRAS*, 503, 3867
 Ivezić Ž. et al., 2019, *ApJ*, 873, 111
 Krause E. et al., 2017, preprint (arXiv:1706.09359)
 Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
 Li B., Zhao G.-B., Teyssier R., Koyama K., 2012, *J. Cosmol. Astropart. Phys.*, 2012, 051
 Linares C., Mota D. F., Winther H. A., 2014, *AAP*, 562, A78
 Lepori F., Adamek J., Durrer R., Clarkson C., Coates L., 2020, *MNRAS*, 497, 2078
 Planck Collaboration VI, 2020, *A&A*, 641, A6
 Puchwein E., Baldi M., Springel V., 2013, *MNRAS*, 436, 348
 Reverberi L., Daverio D., 2019, *J. Cosmol. Astropart. Phys.*, 2019, 035
 Sefusatti E., Crocce M., Scoccimarro R., Couchman H. M. P., 2016, *MNRAS*, 460, 3624
 Silvestri A., Trodden M., 2009, *Rep. Prog. Phys.*, 72, 096901
 Spergel D. et al., 2015, preprint (arXiv:1503.03757)
 Springel V., 2005, *MNRAS*, 364, 1105
 Springel V., Pakmor R., Zier O., Reinecke M., 2021, *MNRAS*, 506, 2871
 Taffoni G., Becciani U., Garilli B., Maggio G., Pasian F., Umana G., Smareglia R., Vitello F., 2020, in Pizzo R., Deul E. R., Mol J. D., de Plaa J., Verkouter H., eds, ASP Conf. Ser. Vol. 527, *Astronomical Data Analysis Software and Systems XXIX*. Astron. Soc. Pac., San Francisco, p. 307

APPENDIX A: CODE SCALING

The code we presented in this work is the merging of two codes whose behaviour in terms of run-time scaling is well-known and characterized; since we did not modify the underlying algorithms, it

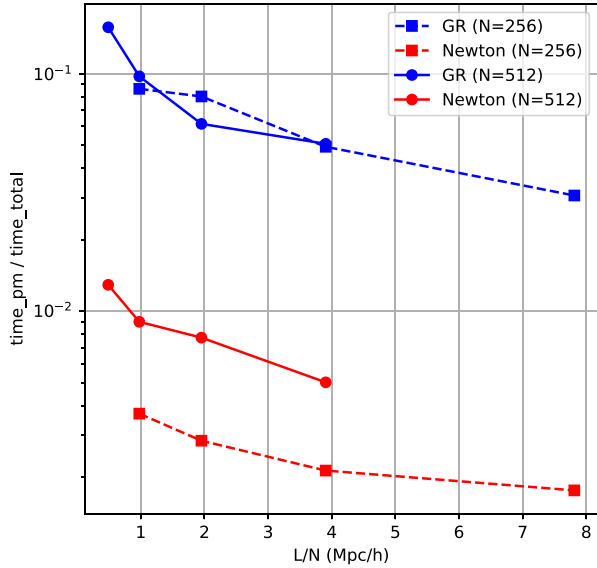


Figure A1. The fraction of PM time to the total running time. Relativistic runs are shown in blue, while Newtonian runs are shown in red, whereas symbols distinguish the value of grid points per dimension N (squares and circles for $N = 256$ and 512 , respectively). We plot the time fraction on the y -axis (log scale) versus the mesh resolution L/N on the x -axis.

is expected that the run-time scaling of our code follows that of the parent codes.

However, the LIBEVOLUTION’s PM is obviously different from GADGET-4’s, and we added the translation of particles data from the host code to the target relativistic PM. Both this facts require that we establish the overall scaling of GRGADGET in its fully relativistic configuration and the overhead associated to both the relativistic PM and the interface between the two codes.

In Fig. A1, we show the fraction of time spent in the PM in both the original and relativistic configurations as a function of the grid cell size (see the caption for details). The relativistic PM is an order of magnitude more expensive than the original GADGET-4’s Newtonian PM, although in absolute sense, it is still either negligible or secondary in the simulation sets that have been tested (it reaches a maximum value of 16 per cent at highest resolution, i.e. in the $N = 512$, $L = 250 \text{ Mpc } h^{-1}$). However, it scales with both the resolution and the grid number as the original Newtonian PM does.

Figs A2 and A3 report the scaling of run time in strong and weak scaling tests, respectively, for the total run time, the tree time, and the PM time (left-hand, middle, and right-hand panels in both figures; see the captions for details). As inferred from Fig. A1, the run-time and hence its scaling are dominated by the GADGET-4’s Tree section.

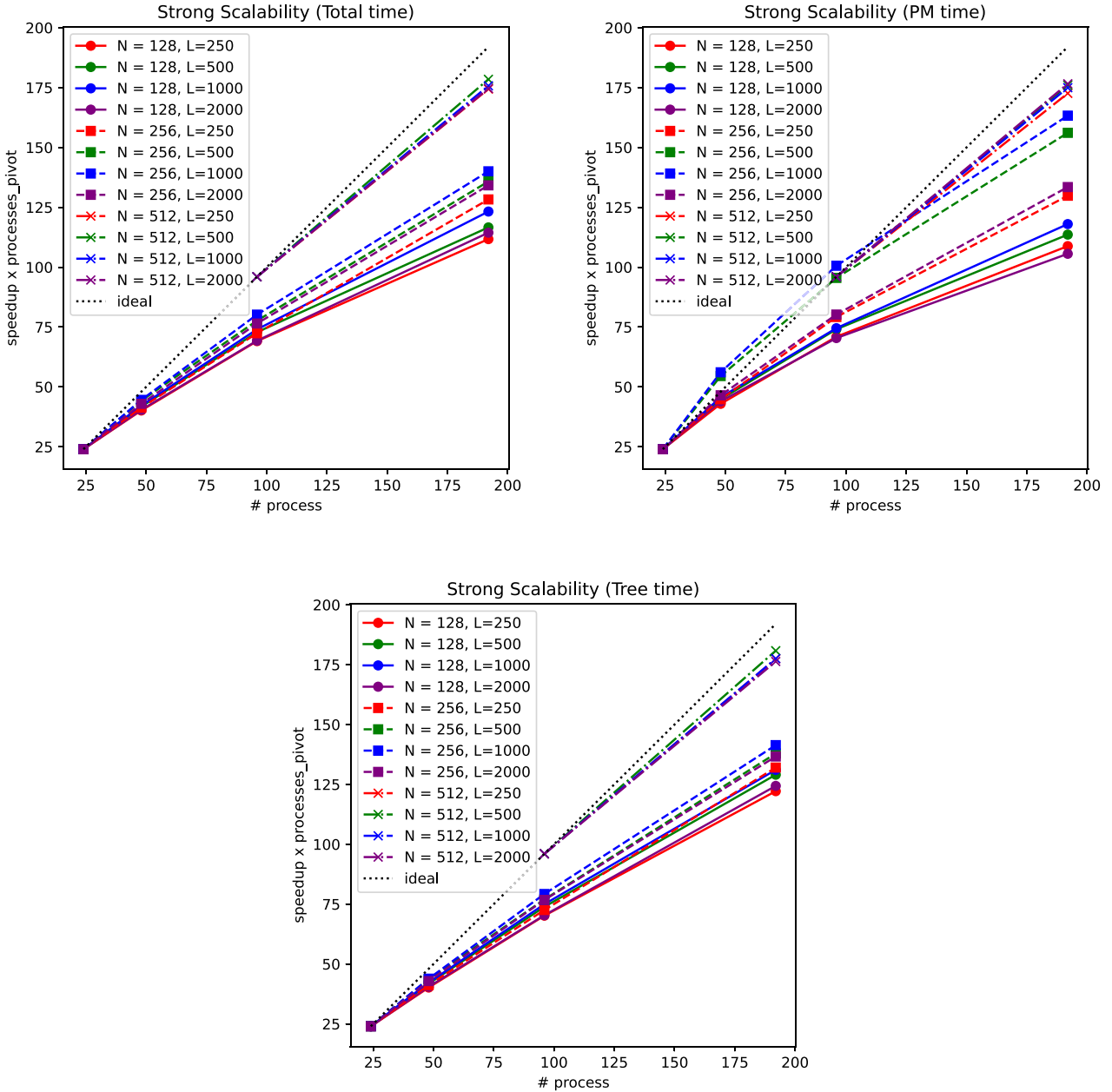


Figure A2. Strong-scaling test. We present the code scaling as the number P of MPI tasks is increased while running the same simulation set-up. All the results refer to GRGADGET, i.e. to the configuration with fully relativistic PM. On the x -axis, P increases from 24 to 192, by $\times 2$ steps. On the y -axis, we report the speed-up (normalized so that the ideal speed-up for $P = 1$ is 1) for the total running time, the time spent in the PM, and the time spent in the Tree on the left-hand, middle, and right-hand panels, respectively. Note that the ideal behaviour (black dotted line) would result in a linear speed-up. The PM data include the translation of particles data from GADGET-4 to LIBGEVOLUTION. We show the results for $N = 128, 256, \text{ and } 512$ (solid, dashed, and dot-dashed lines, respectively) for four different box sizes (i.e. mass resolutions), $L = 250, 500, 1000, \text{ and } 2000 \text{ Mpc } h^{-1}$ (circles, squares, and stars, respectively). See the discussion in Appendix A for details.

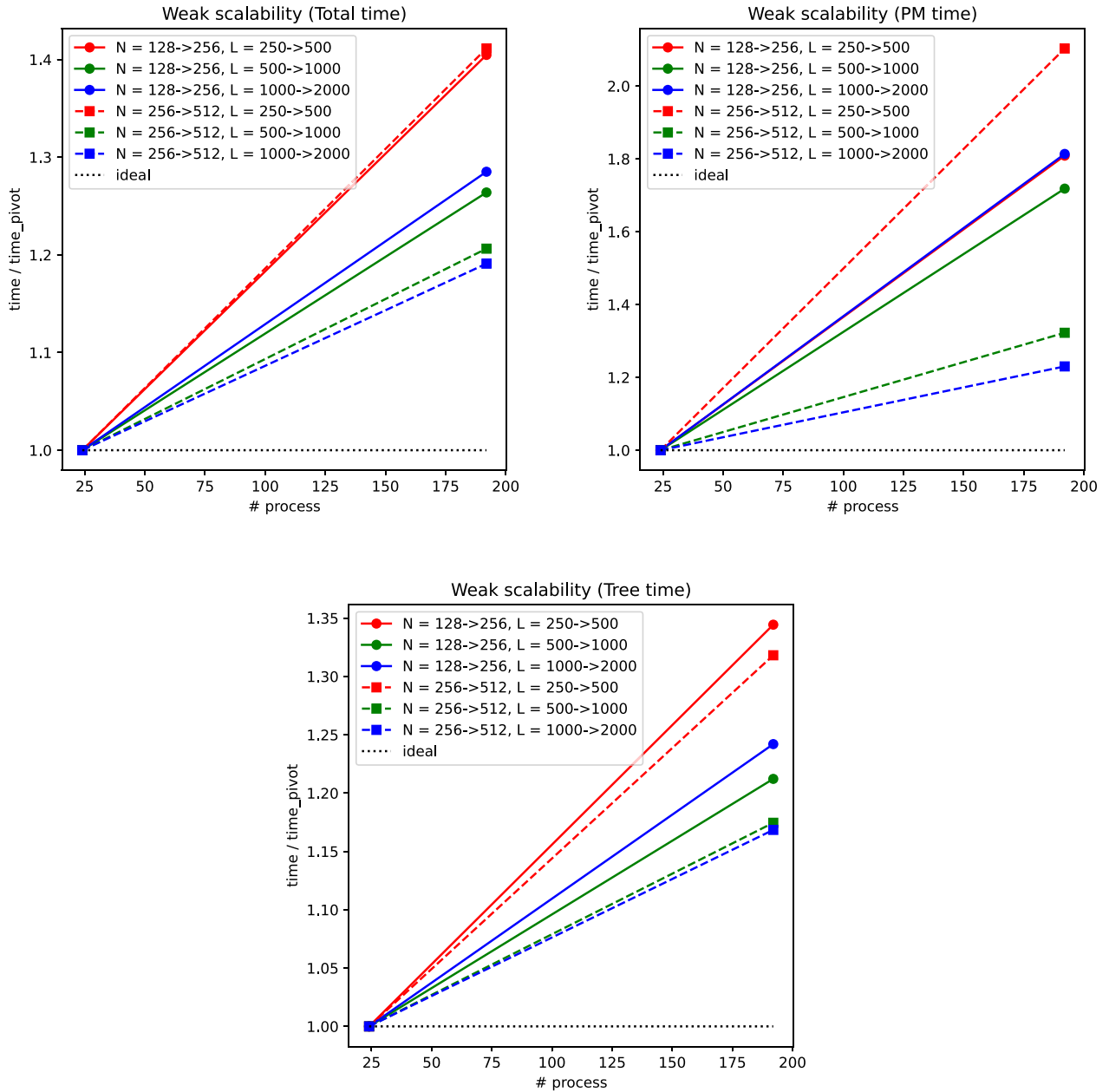


Figure A3. Weak-scaling test. We present the code scaling as the number P of MPI tasks is increased for a proportionally increasing problem, then keeping constant the particles-per-task occupancy. All the results refer to GRGADGET, i.e. to the configuration with fully relativistic PM. On the x -axis, P increases from 24 to 192 with only two test cases. On the y -axis, we report the speed-up for the total running time, the time spent in the PM, and the time spent in the Tree on the left-hand, middle, and right-hand panels, respectively. Note that the ideal behaviour would result in a constant running time (horizontal dotted black line). The PM data includes the translation of particles data from GADGET-4 to LIBGEVOLUTION. We show the results for two cases: from $N = 128$ to $N = 256$ (solid lines with circles), and from $N = 256$ to $N = 512$ (dashed lines with squares). Each of the two cases has been run for three different box sizes (i.e. mass resolutions): $L = 250 \rightarrow L = 500 \text{ Mpc } h^{-1}$, $L = 500 \rightarrow L = 1000 \text{ Mpc } h^{-1}$, and $L = 1000 \rightarrow L = 2000 \text{ Mpc } h^{-1}$ (red, green, and blue colours, respectively). See the discussion in Appendix A for details.

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.