



HAL
open science

Magrathea-Pathfinder: a 3D adaptive-mesh code for geodesic ray tracing in N-body simulations

Michel-Andrès Breton, Vincent Reverdy

► **To cite this version:**

Michel-Andrès Breton, Vincent Reverdy. Magrathea-Pathfinder: a 3D adaptive-mesh code for geodesic ray tracing in N-body simulations. *Astronomy and Astrophysics - A&A*, 2022, 662, pp.A114. 10.1051/0004-6361/202142661 . hal-03462933

HAL Id: hal-03462933

<https://hal.science/hal-03462933v1>

Submitted on 28 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Magrathea-Pathfinder: a 3D adaptive-mesh code for geodesic ray tracing in N -body simulations

Michel-Andrès Breton^{1,2,3} and Vincent Reverdy^{4,5}

¹ Aix Marseille Univ, CNRS, CNES, LAM, Marseille, France
e-mail: breton@ice.csic.es

² Institute of Space Sciences (ICE, CSIC), Campus UAB, Carrer de Can Magrans, s/n, 08193 Barcelona, Spain

³ Institut d'Estudis Espacials de Catalunya (IEEC), Carrer Gran Capità 2-4, 08193 Barcelona, Spain

⁴ National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: vince.rev@gmail.com

⁵ Laboratoire Univers et Théories, Observatoire de Paris, Université PSL, Université de Paris, CNRS, 92190 Meudon, France

Received 14 November 2021 / Accepted 4 March 2022

ABSTRACT

We introduce MAGRATHEA-PATHFINDER, a relativistic ray-tracing framework that can reconstruct the past light cone of observers in cosmological simulations. The code directly computes the 3D trajectory of light rays through the null geodesic equations, with the weak-field limit as its only approximation. This approach offers high levels of versatility while removing the need for many of the standard ray-tracing approximations such as plane-parallel, Born, or multiple-lens. Moreover, the use of adaptive integration steps and interpolation strategies based on adaptive-mesh refinement grids allows MAGRATHEA-PATHFINDER to accurately account for the non-linear regime of structure formation and fully take advantage of the small-scale gravitational clustering. To handle very large N -body simulations, the framework has been designed as a high-performance computing post-processing tool relying on a hybrid parallelization that combines MPI tasks with C++11 `std::threads`. In this paper, we describe how realistic cosmological observables can be computed from numerical simulation using ray-tracing techniques. We discuss in particular the production of simulated catalogues and sky maps that account for all the observational effects considering first-order metric perturbations (such as peculiar velocities, gravitational potential, integrated Sachs-Wolfe, time-delay, and gravitational lensing). We perform convergence tests of our gravitational lensing algorithms and conduct performance benchmarks of the null geodesic integration procedures. MAGRATHEA-PATHFINDER introduces sophisticated ray-tracing tools to make the link between the space of N -body simulations and light-cone observables. This should provide new ways of exploring existing cosmological probes and building new ones beyond standard assumptions in order to prepare for the next generation of large-scale structure surveys.

Key words. large-scale structure of Universe – methods: numerical – gravitation – gravitational lensing: weak

1. Introduction

One of the main challenges of modern science is understanding the dark sector of the Universe which dominates its energy content (according to the cosmological concordance model). Since the discovery of the accelerated expansion of our Universe (Perlmutter et al. 1998; Riess et al. 1998), cosmologists have for decades searched for new probes to understand its properties and have successfully confronted the Λ CDM model with them (Planck Collaboration VI. 2020; Abbott et al. 2019; Scolnic et al. 2018; Riess et al. 2019; Wong et al. 2019; Freedman et al. 2019; Alam et al. 2021). All of these probes share a commonality in that they consist in information coming from photons, using which we can observe the Universe. It is only very recently that we have begun to use other messengers such as gravitational waves to get complementary information (Holz & Hughes 2005; Caprini & Tamanini 2016). Nevertheless, photons remain our principal source of information, and one can ask how the properties of light modify our perception of the Universe.

The propagation of photons in a non-homogeneous Universe leads to mainly two effects, which have received a lot of attention over the years and are accounted for to interpret observational data. First, gravitational lensing

(Schneider et al. 1992; Bartelmann & Schneider 2001) modifies the apparent position of sources but also alters their observed properties (shape, luminosity) with respect to the case where photons would propagate in a homogeneous Friedmann–Lemaître–Robertson–Walker (FLRW) universe. Second, the position of the emission lines of sources are shifted due to their own motion, which in return leads to an error in estimating distances when assuming a fiducial cosmological model: this is called redshift-space distortions (RSD, Kaiser 1987; Hamilton 1992). The image we have of our Universe is therefore altered due to these two phenomena. However, these effects also leave distinct imprints on various cosmological observables, which in turn can help us infer the properties of the Universe. Various current and future missions such as *Euclid* (Laureijs et al. 2011) or DESI (DESI Collaboration 2016) aim at studying these effects through the shape of distant galaxies or their observed spatial distribution.

Due to the increasing quality of data, it is becoming necessary to model the mapping from a statistically homogeneous and isotropic universe to the observed one more accurately. This was done for example regarding the galaxy number counts, accounting for all the effects at first order in metric perturbations within linear theory (Yoo et al. 2009;

Challinor & Lewis 2011; Bonvin & Durrer 2011). However, theoretical prescriptions are limited because they usually cannot properly model the non-linear regime of structure formation. Instead, the use of numerical simulations becomes mandatory to fully understand the clustering of matter, from linear to non-linear scales.

Numerical simulations and, more precisely, dark-matter (DM) N -body simulations (Hockney & Eastwood 1988) have been widely used in cosmology to study the large-scale structure of the Universe beyond analytical methods. A lot of work has been done to develop optimised N -body codes (Kravtsov et al. 1997; Couchman 1991; Knebe et al. 2001; Teyssier 2002; Springel 2005; Bryan et al. 2014; Aubert et al. 2015; Garrison et al. 2021) which allow us to have access to good spatial resolution (small scales) and beat cosmic variance (large scales), while limiting shot noise. Cosmological N -body simulations run on supercomputers and use high-performance computing (HPC) techniques, so that today the largest numerical simulations reach trillions of DM particles (Potter et al. 2017; Heitmann et al. 2019; Ishiyama et al. 2021).

N -body simulations compute the evolution of the matter density field from initial perturbations at high redshift up to now. However, it is not sufficient to accurately model what we actually observe as one still needs to construct past light cones for some given observers and compute the trajectory of light. This is usually done through the multiple lens formalism (Blandford & Narayan 1986), and ray-tracing post-processing tools have been developed mostly to model the effect of gravitational lensing (Fluke et al. 1999; Jain et al. 2000; Fosalba et al. 2008; Hilbert et al. 2009; Metcalf & Petkova 2014; Giocoli et al. 2015; Fabbian et al. 2018; Gouin et al. 2019). However, these methods often compute the lens equation on 2D planes and rely on multiple approximations. Alternatively, several authors have developed ray-tracing algorithms which instead directly compute the geodesic equations in 3D; however the applications are usually still restricted to gravitational lensing (Killedar et al. 2012; Barreira et al. 2016) or distance measurements (Koksbang & Hannestad 2015; Giblin et al. 2016).

Ideally, ray tracing should be general enough to accurately reconstruct the past light cone of an observer and produce a wide range of cosmological observables. This approach has been gaining more and more attention recently (Reverdy 2014; Borzyszkowski et al. 2017; Breton et al. 2019; Adamek et al. 2019; Lepori et al. 2020; Breton & Fleury 2021) as it allows for an in-depth study of subtle effects which are currently neglected but could play an important role in future surveys. As evidence of this, Breton et al. (2019) used ray tracing to find the null geodesic connecting an observer to various sources for the first time in order to estimate the impact of relativistic effects on the clustering of haloes. These authors found that the dipole of the correlation function could be used to probe their gravitational potential in next-generation surveys (Saga et al. 2022) and therefore test the nature of gravity (Bonvin & Fleury 2018; Saga et al. 2021).

The goal of this paper is to present the MAGRATHEA-PATHFINDER framework initially developed in Reverdy (2014) and further developed in several directions since then. It is designed to model the observed Universe as accurately as possible. To this end, we developed ray-tracing techniques that allow us to construct various cosmological observables beyond standard assumptions. In this article, we briefly present the basics of MAGRATHEA-PATHFINDER (Reverdy 2014) and focus on recent developments and code validation. In Sect. 2, we review the main features of our ray-tracing code. In Sect. 3, we perform convergence tests and conclude in Sect. 4.

2. Numerical methods

MAGRATHEA-PATHFINDER is a post-processing numerical framework¹ to propagate photons throughout light cones produced by N -body astrophysics simulations. The framework is built on top of MAGRATHEA (Multi-processor Adaptive Grid Refinement Analysis for THEoretical Astrophysics), a high-performance library developed to provide highly optimised building blocks for the construction of AMR-based astrophysics applications (Reverdy 2014). More specifically, the ray-tracing framework leverages MAGRATHEA's generic N -dimensional hyperoctree abstraction and the associated numerical methods to simplify the handling of the adaptive mesh refinement while ensuring the highest level of performance. Internally, the adaptive-mesh refinement (AMR) structure is flattened, each cell being associated with a unique binary index that encodes information about its exact location in the tree. All canonical hyperoctree operations such as finding parents, children, or neighbour cells as well as calculating inter-cell distances are implemented in terms of bit manipulation operations. Each cell is also associated with a data tuple to store the physics quantities attached to a particular location. One of the key features of MAGRATHEA is the heavy use of C++ template metaprogramming approaches to guide the compiler through the optimisation process and ensure the highest level of performance regarding the manipulation of indices and physics data (Reverdy & Alimi 2015). In that sense, MAGRATHEA acts as an active library running at compile-time to pre-process MAGRATHEA-PATHFINDER's code. The exact implementation of the hyperoctree and low-level numerical algorithms can be found in Reverdy (2014) and will be presented in more details in an upcoming paper focusing on the MAGRATHEA library. For the present paper, we focus on the ray-tracing algorithms in MAGRATHEA-PATHFINDER made possible by the above-mentioned library.

2.1. From simulations to MAGRATHEA octree

As ray-tracing simulations are performed as an independent and subsequent phase of dynamic cosmological simulations, it is possible to leverage the particular geometry of the problem to optimise parallelization schemes and minimise inter-node communications. In practice, the very first step consists in generating a 3D light cone containing all the information relative to the gravitational field and converting it into MAGRATHEA's 3D-octree data structure. Usually, the light cone is built from an N -body simulation. If the simulation uses a particle-mesh (PM) method, then the identification between the cells of the simulations and that of MAGRATHEA can be trivially achieved through the conversion of Cartesian positions into MAGRATHEA indices. The same applies to AMR-based simulation. However, when cosmological simulations rely on another method to compute the gravitational field one has to first interpolate the gravitational information available at particle locations onto a fixed or adaptive grid that can then be processed by MAGRATHEA's hyperoctree engine.

Once the light cone is converted into MAGRATHEA's octree format, geometrical inconsistencies are checked to ensure that the ray-tracer can work on clean data. One typical problem that may arise during concentric shell extraction happening at the cosmological simulation level is the production of sparse AMR cells at the boundary of two shells. In this case, the geometrical

¹ Available at <https://github.com/vreverdy/magrathea-pathfinder>

checker adds cells interpolated from a coarser level to build a full tree with either zero or eight children per cell that preserves the original gravitational information. The operation is repeated at every level, from the coarsest to the most refined.

Because the light cone of high-resolution simulations can reach several terabytes of data, it is often not possible to load it on a single node. In this case, a conic domain decomposition of the light cone with overlaps is performed which allows photons to be propagated almost without the need of cross-node communications for the most part of the ray-tracing phase. On top of the conic domain, every computational node gets a copy of a small spherical region centered on the observer to ensure proper independent propagation at very low redshift. The hybrid MPI/C++ thread parallelization approach of MAGRATHEA allows to the user maximise the size of geometrical subdomains and minimise the total size of overlaps while still ensuring a maximal exploitation of computational resources with many threads working at the same time on the same shared memory, each one of them taking care of a particular photon. Once the octree containing all the relevant gravitational information is built, checked, and distributed across computational nodes, the main ray-tracing phase can begin.

2.2. Geodesic integration and light propagation

MAGRATHEA-PATHFINDER propagates photons on the null geodesics of the weakly perturbed FLRW metric in Newtonian gauge

$$ds^2 = a^2(\eta) \left[- \left(1 + 2 \frac{\Phi}{c^2} \right) c^2 d\eta^2 + \left(1 - 2 \frac{\Phi}{c^2} \right) dx^2 \right], \quad (1)$$

where η and \mathbf{x} are respectively the conformal time and comoving coordinates, $a(\eta)$ is the scale factor, Φ the gravitational potential, and c the speed of light. The null geodesic equations are

$$\frac{d^2\eta}{d\lambda^2} = - \frac{2a'}{a} \frac{d\eta}{d\lambda} \frac{d\eta}{d\lambda} - \frac{2}{c^2} \frac{d\Phi}{d\lambda} \frac{d\eta}{d\lambda} + 2 \frac{\partial\Phi}{\partial\eta} \left(\frac{d\eta}{d\lambda} \right)^2, \quad (2)$$

$$\frac{d^2x^i}{d\lambda^2} = - \frac{2a'}{a} \frac{d\eta}{d\lambda} \frac{dx^i}{d\lambda} + \frac{2}{c^2} \frac{d\Phi}{d\lambda} \frac{dx^i}{d\lambda} - 2 \frac{\partial\Phi}{\partial x^i} \left(\frac{d\eta}{d\lambda} \right)^2, \quad (3)$$

where a prime denotes a derivative with respect to conformal time, and λ is an affine parameter. We perform backward ray tracing, meaning that we start the propagation of the photons at the observer today towards the past. The geodesic equations are solved using a fourth-order Runge-Kutta integrator (RK4) with \mathcal{N} steps per AMR cell ($\mathcal{N}=4$ in common settings), where the photons are initialised using $k^\nu k_\nu = 0$ (with $k^\nu = dx^\nu/d\lambda$) given the initial direction of the photon k^i , and $k^0 = 1$, meaning that at the observer conformal time and affine parameter coincides.

To compute Eqs. (2) and (3), we have access to η , x^i , k^0 and k^i that are given by the integrator, $a(\eta)$ and $a'(\eta)$ by external pre-computed tables in the fiducial cosmology of the simulation given η , and $\partial\Phi/\partial x^i$ by the simulation itself. The last subtlety lies in the use of $d\Phi/d\lambda$ or $\partial\Phi/\partial\eta$, knowing that these terms are simply related by

$$\frac{d\Phi}{d\lambda} = \frac{\partial\Phi}{\partial x} \frac{dx}{d\lambda} + \frac{\partial\Phi}{\partial y} \frac{dy}{d\lambda} + \frac{\partial\Phi}{\partial z} \frac{dz}{d\lambda} + \frac{\partial\Phi}{\partial\eta} \frac{d\eta}{d\lambda}. \quad (4)$$

Usually, N -body simulations do not provide $\partial\Phi/\partial\eta$ and therefore it is easier to rewrite the geodesic equations in terms of $d\Phi/d\lambda$

only, because it can be estimated by differentiating between two steps of the integration as

$$\frac{d\Phi^i}{d\lambda} = \frac{\Phi^i - \Phi^{i-1}}{\lambda^i - \lambda^{i-1}}, \quad (5)$$

where the superscripts refer to the integration step. The main drawback of this method is that it strongly depends on the procedure used to build the light cone. For example, if one uses the onion-shell method (Fosalba et al. 2008), then some artefacts will appear on $d\Phi/d\lambda$ at the crossing between shells. It is important to note that these subtleties are only important when studying very small effects (such as ISW or time delay).

Ideally, $\partial\Phi/\partial\eta$ should be provided by the simulation. It is the case for example with the RayGal simulations (Rasera et al. 2021) where the authors use a double-layer strategy, meaning that two light cones slightly shifted in time are stored. In this case, the time derivative of the potential can be straightforwardly computed for every cell using

$$\frac{\partial\Phi}{\partial\eta} = \frac{\Phi_2 - \Phi_1}{a_2 - a_1} \frac{da}{d\eta}, \quad (6)$$

where the subscripts denote which light cone is used. The main advantage of this method is that it is non-perturbative and therefore accurate even at non-linear scales. Last, we note that in practice $\partial\Phi/\partial\eta$ could also be computed from the density and velocity fields (Cai et al. 2010).

All of the components needed to compute the geodesic equations are available at the location of cells. This means that we need to interpolate them at the photon position at each step (and each sub-step in the RK4 integrator), while accounting for the AMR structure of the light cone. We implemented three different interpolation schemes, namely the nearest-grid point (NGP), cloud-in-cell (CIC), and triangular-shaped cloud (TSC) interpolations, of which the first two were already present in Reverdy (2014). For consistency, the interpolation scheme must be the same as the one used to produce the gravity grid (see Sect. 2.1). The weighting functions associated to NGP, CIC, and TSC are

$$W_{\text{NGP}}(r_i) = \begin{cases} 1 & \text{for } |r_i| < 0.5, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

$$W_{\text{CIC}}(r_i) = \begin{cases} 1 - |r_i| & \text{for } |r_i| < 0.5, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$W_{\text{TSC}}(r_i) = \begin{cases} 0.75 - r_i^2 & \text{for } |r_i| < 0.5, \\ (1.5 - |r_i|)^2 / 2 & \text{for } 0.5 < |r_i| < 1.5, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where r is the separation between a cell and the location at which we interpolate, normalised by the cell size.

For NGP, the interpolation is trivial as we take the gravity information from the most refined cell which contains the photon. For CIC and TSC, the interpolation procedure is as follows:

1. Estimate the refinement level of the most refined cell containing the photon;
2. Check if there are 8 (27) neighbouring cells to perform the CIC (TSC) interpolation in three dimensions;

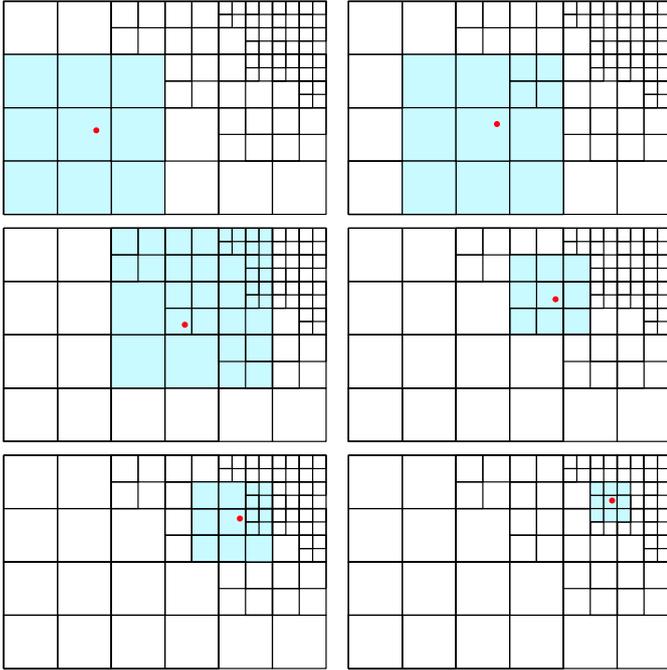


Fig. 1. Illustration of the TSC interpolation scheme at a given location in an AMR grid in 2D. Red points refer to some locations on the grid (which can be for example the position of a photon during its propagation) and the cells used to perform the interpolation are highlighted in blue. The interpolation is done at fixed level, meaning that when the interpolation level is set, we do not use the information from coarser or finer cells.

3. If all the neighbouring cells exist in the octree, compute the interpolation;
4. If at least one of the neighbouring cells does not exist in the octree, repeat the full procedure at a coarser level;
5. If there are not enough neighbours even at coarse level (meaning that we reach the edges of the numerical light cone), the integration is stopped.

We illustrate this procedure in Fig. 1. In the top left panel, we see that there are the nine neighbouring cells to perform the 2D TSC interpolation. In this case the gravity information is easily interpolated at the photon location to compute the geodesic equations. The photon location is updated in the top right panel, where there are also enough neighbours at coarse level. Here, even if the top-right cell also contains more refined cells, they do not contribute to the interpolation and we rather use the coarser cell. In the middle left panel, the photon is in a more refined cell than previously. The code tries to interpolate at this finer level, but cannot find the associated neighbouring cells. It therefore performs the interpolation at a coarser level, where all the neighbours exist in the grid. In the middle right panel, the photon is in a new cell at the same level as previously, the only difference being that now there are enough neighbouring cells to perform the interpolation at this finer level. The bottom panels show a similar behaviour as previously but with more refinement. The CIC procedure is similar to that of TSC, except for the fact that in this case we need four neighbouring cells in two dimensions. A nice feature about TSC compared to CIC is that for the latter, the neighbours depend on the position of the photon within a cell, while for the former it is not the case. We take advantage of this by keeping in memory all the neighbours of a given photon at each step, and if the next step of the photon is in the same cell and the TSC interpolation performed at the same level, then

we do not have to search again for the neighbouring cells in the octree and therefore gain in performance. It should be noted that this interpolation procedure does not prevent discontinuities at the crossing between AMR levels; however we expect this effect to be very faint, especially for the commonly used CIC and TSC interpolation schemes, as we show in Sect. 3.2.

At each step of the propagation, we keep in memory the step number, scale factor, conformal time, comoving position, affine parameter, redshift, wavevector k^ν , level of the most refined cell, density, gravitational potential and its derivatives with respect to conformal time, affine parameter, and comoving coordinates (i.e. the force). It is also possible to estimate these quantities along the propagation of an unperturbed light ray (this is the so-called ‘Born approximation’) by setting the gravity to zero in the geodesic equations.

Now that we have seen the methodology to propagate photons on null geodesics within the 3D AMR structure of the light cone, we shall turn to the implementation of gravitational lensing and simulated catalogues.

2.3. Distortion matrix and gravitational lensing

One direct consequence of light propagation in an inhomogeneous universe is the modification of the apparent position of a source, as well as the modification of its properties (shape and observed luminosity). To derive these properties, we start from the lens equation

$$\beta = \theta - \alpha, \quad (10)$$

where β and θ are the comoving and observed angular position on the sky of a source, and α the deflection angle. We note that θ is the photon direction at the observer which is directly given by k^i at this location. The relevant information for weak gravitational lensing is encoded in the Jacobian matrix \mathcal{A} which describes the mapping from $d^2\beta$ to $d^2\theta$. This gives

$$\mathcal{A} = \frac{d^2\beta}{d^2\theta} = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix}, \quad (11)$$

where κ , $\gamma = \gamma_1 + i\gamma_2$, and ω are respectively the convergence, shear, and rotation of an image. The magnification μ is the change in observed flux and size of an image (due to the conservation of surface brightness), and is defined as the inverse determinant of \mathcal{A} .

Usually, ray-tracing codes are designed to compute κ and γ (and sometimes μ). However, these often make use of several approximations such as Born, plane-parallel, and multiple lens. We now show how to implement the computation of weak-lensing quantities using the 3D propagation on null geodesics described in Sect. 2.2. To do so, we use two methods which we refer to as ‘infinitesimal’ and ‘finite’ beams.

2.3.1. Infinitesimal light beams

First, we consider the usual definition of the distortion matrix, which describes the behaviour of infinitesimal light beams. Formally, Eq. (11) is given by

$$\mathcal{A}_{ab} = \delta_{ab} - \frac{2}{c^2} \int_0^{\chi_s} d\chi \frac{(\chi_s - \chi)\chi}{\chi_s} \nabla_a \nabla_b \Phi[\eta(\chi), \mathbf{x}(\chi)], \quad (12)$$

where the subscripts refer to the angular coordinates, δ_{ab} is the Kronecker delta, χ is the comoving distance of the photon during its propagation, and χ_s is the distance at the source.

We note that here the derivatives are performed along angular spherical coordinates. Because spherical derivatives are not straightforward to compute, we find it easier to first perform derivatives along the 3D Cartesian coordinates, and then rewrite them in terms of spherical ones. At any location on the light cone, we have access to the gravitational field $-\nabla_i\Phi(\mathbf{x}) \equiv F_i(\mathbf{x}) = \{F_x(\mathbf{x}), F_y(\mathbf{x}), F_z(\mathbf{x})\}$. It is possible to compute its Cartesian derivatives by differentiating as

$$-\nabla_i\nabla_j\Phi(\mathbf{x}) = \frac{F_j(\mathbf{x} + h\mathbf{e}_i) - F_j(\mathbf{x} - h\mathbf{e}_i)}{2h}, \quad (13)$$

where $\mathbf{e}_i = \{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ is a unit vector, and h is the derivation step. We found that choosing a derivation step equal to the size of the most refined AMR cell (and therefore highest AMR level) the photon is in, that is $h = 2^{-\text{level}}$, is the optimal choice to compute these derivatives (see also Sect. 3.5). From a numerical perspective, $\nabla_i\nabla_j \neq \nabla_j\nabla_i$ when $i \neq j$. However, the difference is so small that we consider them equal so as to avoid the costly computation of all the possible permutations. The last step is to go from Cartesian to spherical derivatives. Our method is similar to that of Barreira et al. (2016), except that we do not resort to the Born approximation and we compute ray-tracing as post-processing. The components of the Laplacian in Eq. (12) are given by

$$\nabla_1\nabla_1\Phi = \sin^2\varphi \nabla_x\nabla_x\Phi + \cos^2\varphi \nabla_y\nabla_y\Phi - \sin 2\varphi \nabla_x\nabla_y\Phi, \quad (14)$$

$$\begin{aligned} \nabla_2\nabla_2\Phi &= \cos^2\varphi \cos^2\vartheta \nabla_x\nabla_x\Phi + \sin^2\varphi \cos^2\vartheta \nabla_y\nabla_y\Phi \\ &+ \sin^2\vartheta \nabla_z\nabla_z\Phi + \sin 2\varphi \cos^2\vartheta \nabla_x\nabla_y\Phi \\ &- \sin\varphi \sin 2\vartheta \nabla_y\nabla_z\Phi - \cos\varphi \sin 2\vartheta \nabla_x\nabla_z\Phi, \end{aligned} \quad (15)$$

$$\begin{aligned} \nabla_1\nabla_2\Phi &= \cos\vartheta \cos\varphi \sin\varphi (\nabla_y\nabla_y\Phi - \nabla_x\nabla_x\Phi) \\ &+ (\cos^2\varphi - \sin^2\varphi) \cos\vartheta \nabla_x\nabla_y\Phi \\ &+ \sin\varphi \sin\vartheta \nabla_x\nabla_z\Phi - \cos\varphi \sin\vartheta \nabla_y\nabla_z\Phi, \end{aligned} \quad (16)$$

where (φ, ϑ) is the angular position of the photon in spherical coordinates at each step. Last, we use $\nabla_1\nabla_2\Phi = \nabla_2\nabla_1\Phi$, meaning that we consider that there is no rotation of the image. We note that the option to use the Born approximation has been implemented by using this method along an FLRW trajectory.

2.3.2. Finite beams

In reality, sources are not infinitesimal but are rather extended. The usual weak-lensing formalism is therefore an approximation of the more accurate finite-beam formalism (Fleury et al. 2017, 2019a,b). In this case we can compute the lensing distortion matrix by launching a beam composed of several close-by light rays which are all integrated on null geodesics independently. The idea to consider several rays to characterise a light beam is similar to the ‘ray-bundle method’ proposed by Fluke et al. (1999) and Fluke & Lasky (2011). First, we launch a reference photon in the direction in which we want to compute the lensing matrix. This photon is used as a reference to know where to stop the beam. For example, we might want to know the lensing quantities at some parameter p_0 where $p = \{a, \eta, \chi, z, \lambda\}$ (see Sect. 2.2), but we still have the choice to stop the beam light rays at some other parameter $\tilde{p}_0 = \tilde{p}(p_0)$ for the reference photon. We note that all of the parameters are equivalent in an FLRW universe, but this is no longer true when accounting for inhomogeneities.

To compute the distortion matrix, we therefore need to know p_0 , ε , which is the beam semi-aperture, and $\theta = (\theta_1, \theta_2)$, which is the photon direction at the observer.

In Cartesian coordinates, the direction of the target is $\hat{\mathbf{r}} = (\cos\theta_1 \sin\theta_2, \sin\theta_1 \sin\theta_2, \cos\theta_2)$ where a hat denotes a unit vector. We can define a screen perpendicular to this direction with two orthogonal vectors $\mathbf{e}_1 = (-\sin\theta_1, \cos\theta_1, 0)$ and $\mathbf{e}_2 = (-\cos\theta_1 \cos\theta_2, -\sin\theta_1 \cos\theta_2, \sin\theta_2)$. Now we can launch four rays denoted A, B, C, and D (see also Fig. 5 in Breton & Fleury 2021), with initial directions

$$\hat{\mathbf{r}}_A = \hat{\mathbf{r}} + \tan(\varepsilon) \mathbf{e}_1 \times \mathbf{u}, \quad (17)$$

$$\hat{\mathbf{r}}_B = \hat{\mathbf{r}} - \tan(\varepsilon) \mathbf{e}_1 \times \mathbf{u}, \quad (18)$$

$$\hat{\mathbf{r}}_C = \hat{\mathbf{r}} - \tan(\varepsilon) \mathbf{e}_2 \times \mathbf{u}, \quad (19)$$

$$\hat{\mathbf{r}}_D = \hat{\mathbf{r}} + \tan(\varepsilon) \mathbf{e}_2 \times \mathbf{u}, \quad (20)$$

where $\mathbf{u} = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. Each ray is propagated on the light cone until \tilde{p}_0 so that their final position is given by ξ_A, ξ_B, ξ_C and ξ_D . To compute the lensing distortion matrix, we differentiate between the positions of the light rays of the beam. Taking advantage of the fact that the beam is supposed to be small, we can write $\Delta\beta = \Delta\xi/\chi_0 = \hat{\mathcal{A}}\Delta\theta$, where χ_0 is the comoving distance of the reference ray at p_0 and $\hat{\mathcal{A}}$ the finite-beam distortion matrix.

A last subtlety is the choice of screen onto which we compute the finite differences. From Eq. (11), a natural choice for this screen is the one orthogonal to $\beta = (\beta_1, \beta_2)$, defined as $\tilde{\mathbf{e}}_1 = (-\sin\beta_1, \cos\beta_1, 0)$ and $\tilde{\mathbf{e}}_2 = (-\cos\beta_1 \cos\beta_2, -\sin\beta_1 \cos\beta_2, \sin\beta_2)$. Alternatively, it is possible to use the more physically motivated Sachs screen, which is orthogonal to the central (reference) photon direction. In this case, the screen is defined with $\tilde{\mathbf{e}}_1 = (-\sin\zeta_1, \cos\zeta_1, 0)$ and $\tilde{\mathbf{e}}_2 = (-\cos\zeta_1 \cos\zeta_2, -\sin\zeta_1 \cos\zeta_2, \sin\zeta_2)$, with $\zeta_1 = \arctan(\hat{k}_y/\hat{k}_x)$ and $\zeta_2 = \arccos(\hat{k}_z)$, where $\hat{\mathbf{k}} \equiv \hat{\mathbf{k}}(p_0) = (k_x, k_y, k_z)$ is the direction of the central photon at p_0 . Finally, the lensing distortion matrix is computed as

$$\hat{\mathcal{A}} \equiv \frac{1}{2\chi_0 \tan(\varepsilon)} \begin{bmatrix} (\xi_A - \xi_B) \times \tilde{\mathbf{e}}_1 & (\xi_C - \xi_D) \times \tilde{\mathbf{e}}_1 \\ (\xi_A - \xi_B) \times \tilde{\mathbf{e}}_2 & (\xi_C - \xi_D) \times \tilde{\mathbf{e}}_2 \end{bmatrix}. \quad (21)$$

Moreover, instead of stopping the light rays of the beam at \tilde{p}_0 , we also implemented the possibility to stop them directly on the screen of interest.

Using finite beams to compute the Jacobian matrix allows us to make an accurate treatment of extended sources, which smooths the effect of gravitational lensing on the scale of the beam (Fleury et al. 2017, 2019a,b). This impacts the convergence and shear angular power spectra with respect to the infinitesimal case. A nice agreement between theoretical prediction and numerical estimation was found in Breton & Fleury (2021) (see also Appendix B for a visualisation of the finite-beam effect on convergence maps).

As this method is purely geometrical and depends on the differential deflection of photons within a beam, we cannot adapt it with the Born approximation. Also, this method does not assume that the off-diagonal terms of the Jacobian matrix are equal, meaning that we have access to the image rotation. Having implemented the tools to propagate photons and compute the weak gravitational lensing quantities, we now turn to the production of cosmological observables: maps and simulated catalogues.

2.4. Producing Healpix maps

First, we consider ‘direction-averaged’ observables (Kibble & Lieu 2005), which relate to observations in random directions

of the sky that are especially relevant for the cosmic microwave background. Usually, ray tracing is performed in a pencil beam where all the photons propagate almost in parallel towards the pixels of a plane. In MAGRATHEA-PATHFINDER, all the photons start from the observer and in this case we find that the most natural frame to homogeneously sample the sky is to use Healpix (Górski et al. 2005). Given a resolution level N_{side}^2 , Healpix gives the position of the centre of the pixels. These positions are then used to initialise the photon direction k^i at the observer.

In practice, we first assign the pixels to the different MPI subdomains (see Sect. 2.1) to ray-trace the different parts of the sky in parallel. Within one MPI task, we use C++11 `std::thread` multithreading to propagate light rays towards the pixels simultaneously. The user must then specify `z_stop_min` (minimum redshift), `z_stop_max` (maximum redshift), and `nb_z_maps` (number of redshifts at which we compute the maps), which sets (roughly) the redshifts z_n of the output maps. More precisely, the maps are computed at some iso-parameter p surfaces (using the keyword `stop_ray`, see also Sect. 2.3.2), and evaluated at $p_n = p(z_n)$, where p_n is the stop criterion computed by launching an FLRW light ray in a very refined homogeneous grid.

Now, we need to specify which quantities we want to estimate. In `map_components`, the user can write a list of keywords to output several maps containing the following information:

- `lensing`: The code computes the weak-lensing quantities κ , γ_1 , γ_2 and $1/\mu$ using either the infinitesimal method (`jacobianteype=infinitesimal`, see Sect. 2.3.1) or the finite-beam method (`jacobianteype=bundle`, see Sect. 2.3.2). For the latter, an additional map is written which contains the image rotation, and one must specify the stop criterion of `bundle` (that is, \tilde{p} , see Sect. 2.3.2) using `stop_bundle`.

- `lensing_born`: Here we propagate an FLRW light ray in the pixel directions. We then use the infinitesimal method along these trajectories to estimate the weak-lensing quantities.

- `deflection`: The deflection angle, computed as $\alpha = \theta - \beta$, where θ is given by Healpix and β by the photon position at the map.

- `dens`: The density (computed in the N -body solver) interpolated at the iso- p surfaces.

- `dens_max`: The maximum density probed by the photon during its trajectory until the maps.

- `phi`: The gravitational potential interpolated at the iso- p surfaces.

- `isw`: Integrated Sachs-Wolfe/Rees-Sciama maps.

- `steps`: The number of integration steps for the photon.

- `Relative differences with respect to their FLRW counterpart for various quantities: χ , λ , η , a , z` , with the keywords `dr`, `dl`, `dt`, `da`, `dz` respectively.

These are the map types currently implemented in MAGRATHEA-PATHFINDER, and in the future this number will be easily expanded to add more functionalities. The only limitation regarding the number of map components and redshifts at some Healpix resolution is the memory available.

Last, there is one subtlety regarding the computation of the redshift: as we estimate the redshift of the photon at each step of integration, we only perturb the redshift with gravity information (local and integrated terms). Indeed, we do not have access to the velocity which is only available at the position of DM particles (or haloes). However, by adding the compile flag `-DVELOCITYFIELD`, MAGRATHEA-PATHFINDER computes the velocity field at the position of the AMR grid by interpolating (using either CIC or TSC) the velocity from all the particles

available in the light cone shells around z_n in the subdomain of interest. This means that we need to add data slots on the octree to store $\{v_x, v_y, v_z\}$ at each cell, which produces a heavier octree. This is interesting in particular when we want to compute a map at some constant-redshift surface, where the redshift is notably impacted by the Doppler contributions.

2.5. Geodesics finder and relativistic catalogues

Alternatively, we can produce ‘source-averaged’ observables: these are simulated catalogues which relate to observations at the direction of sources on the sky (such as galaxy or supernovae surveys). As we observe sources thanks to photons that propagated between their emission location and us, we must reproduce the same procedure numerically to construct realistic simulated catalogues. MAGRATHEA-PATHFINDER already integrates the trajectory of light rays on null geodesic; the only remaining element is to find the appropriate initial condition to link the observer to sources on the light cone.

As described in Breton et al. (2019), we start from the comoving position of a source r , with comoving angular position β . The goal is to find θ so that the photon angular position at the comoving distance of the source is very close to β . To do so, we iterate over θ and use a Newton-like method to find the null geodesic which connects the observer to the source. This reads

$$\theta_{i+1} = \theta_i - \mathcal{A}^{-1}(\beta_i - \beta), \quad (22)$$

where the subscripts refer to the iteration of the root-finding method, and β_i is the photon angular position at χ with initial direction at the observer θ_i . To avoid any problems due to the system of coordinates, we make all the calculations on a screen orthogonal to θ_i . We consider that our method has converged when $|\beta_i - \beta| < \epsilon$, with ϵ being the convergence criterion set with the keyword `cat_accuracy`. To speed up the calculation, for the first three iterations we impose $\mathcal{A} = \mathcal{I}$, with \mathcal{I} being the identity matrix. This should be a good enough approximation when there are no large gravitational fields along the photon trajectory. If the iterations do not converge, we then estimate \mathcal{A} with the infinitesimal method (which is faster than the finite-beam one) for two iterations. If convergence is still not achieved, we use a finite-beam method to compute the Jacobian matrix. If after ten iterations convergence is still not achieved (which in our tests represents about one part per million), the sources are saved in some separate files, which we can decide to use if $|\beta_i - \beta|$ is small enough, or we can run an alternative root finder where we re-run the ray tracing on sources with higher resolution. In this case, we launch photons in the direction of a regular grid centred on θ_{10} , and compute θ_{11} from the grid pixel which gives the best agreement. We repeat this process until we achieve the desired accuracy. The size of the grid decreases at every iteration of this procedure. While this last method is slow, it should in the end converge for all the remaining sources. We note that our root-finder algorithm stops whenever one image per source is detected, which corresponds to the weak-lensing regime. For multiply imaged sources, the principal image is likely to be the one to be detected first, and we plan to develop a multiple-image finder in the future to study strong-lensing in more detail.

Finally, for each source we therefore have β , θ , \mathcal{A} (computed when θ is known) using either the infinitesimal (with or without the Born approximation) or finite-beam method, as well as various redshifts containing local and integrated terms, depending on the contributions we are interested in. These read

$$z_0 = \frac{a_0}{a} - 1, \quad (23)$$

² The total number of pixels on the full sky is $N_{\text{pix}} = 12 \times N_{\text{side}}^2$.

$$z_1 = z_0 + \frac{a_0}{a} \frac{[\Phi_o - \Phi_s]}{c^2}, \quad (24)$$

$$z_2 = z_1 + \frac{a_0}{a} \frac{[(\mathbf{v}_s - \mathbf{v}_o) \times \mathbf{n}]}{c}, \quad (25)$$

$$z_3 = z_2 + \frac{1}{2} \frac{a_0}{a} \frac{[|\mathbf{v}_s|^2 - |\mathbf{v}_o|^2]}{c^2}, \quad (26)$$

$$z_4 = z_3 - \frac{2a_0}{c^2 a} \int_{\eta_s}^{\eta_o} \frac{\partial \Phi}{\partial \eta} d\eta, \quad (27)$$

$$z_5 = \frac{(g_{\mu\nu} k^\mu u^\nu)_s}{(g_{\mu\nu} k^\mu u^\nu)_o} - 1, \quad (28)$$

where z_0 is the FLRW redshift, and z_1 to z_4 contain the added contribution of the gravitational potential, Doppler effect, Transverse Doppler effect, and ISW. The scale factor today is given by a_0 , the subscripts ‘ o ’ and ‘ s ’ refer to evaluations at the observer and at the source respectively, and $g_{\mu\nu} k^\mu u^\nu = -ak^0 \left[1 + \Phi/c^2 + \mathbf{v} \times \mathbf{n}/c + \frac{1}{2} |\mathbf{v}|^2/c^2 \right]$. In practice, z_5 is the true observed redshift (at first order in metric perturbations), while the redshift decomposition from z_0 to z_4 is particularly interesting to study these effects either in isolation or in combination (see also Breton et al. 2019 for an analysis of these effects on the dipole of the correlation function). For example, by combining z_0 , z_1 , and z_2 we can infer a measure of redshift that is only perturbed by peculiar velocities, which is the usual framework for RSD studies.

2.6. Light-ray statistics

Last, we implemented the possibility to propagate light rays and bundles in random directions on the sky, and save several relevant statistics along their trajectories. In each subdomain, the user sets the number of trajectories (i.e. the number of lines of sight). For each trajectory, MAGRATHEA-PATHFINDER ray traces a light bundle which contains one central ray, and N photons in a circular beam around it. The photons of the beam are evenly spaced on the circle, with semi-aperture set by the user, and each photon propagates on null geodesics independently. From this, MAGRATHEA-PATHFINDER can either output the full trajectory for all the photons (see Sect. 2.2 to see which kind of information is saved) or summary statistics for each subdomain or for the full light cone. Saving the full trajectories of all the bundles is interesting for a detailed study of what happens during light propagation at each step of integration, and the bundle method proposed here is more general than that of Sect. 2.3.2 which consists in only four surrounding rays. Furthermore, using a spherical bundle with an arbitrary number of photons in principle enables us to study higher order effects of gravitational lensing such as flexion with high accuracy.

3. Tests and convergence study

In this section we describe several tests that we used to check the convergence of the numerical methods described in Sect. 2. To this end, we used MAGRATHEA-PATHFINDER on the RayGal simulation³ (Rasera et al. 2021) which is based on the PM-AMR RAMSES code (Teyssier 2002). This simulation has evolved 4096^3 particles (and as many coarse cells) in a $(2.625 h^{-1} \text{Gpc})^3$ volume, with the Λ CDM, WMAP-7 year data best-fit parameters (Komatsu et al. 2011). The RayGal simulation outputs three light cones using the onion-shell method: a full-sky light cone

³ <https://cosmo.obspm.fr/public-datasets/>

Table 1. Ray-tracing run-time for a single integration step, averaged over 100 realisations of the same trajectory and depending on the type of interpolation.

Integration	Time per step (μs)
FLRW	0.55
NGP	1.10
CIC	6.0
TSC	14.6

Notes. We consider a photon trajectory with 4 steps per AMR cell on the narrow light cone of the RayGal simulation until $z = 10$. The total number of integration steps is roughly 5×10^4 .

and two narrow cones with 2500 and 400 deg^2 aperture which reach a maximum redshift of $z = 0.5, 2$ and 10 respectively.

3.1. Performance tests

First, we estimated the run-time performance of MAGRATHEA-PATHFINDER when propagating photons within the AMR structure of the RayGal light cone. As the MPI and multithreading parallelizations are almost ‘embarrassingly parallel’, that is there is little to no communication between tasks, we expect MAGRATHEA-PATHFINDER to scale almost linearly with the number of cores. Furthermore, the number of steps per photon depends on the stop criterion, the size of the light cone, and the number of integration steps per AMR cell chosen by the user. Therefore, the relevant quantity to estimate the performances of MAGRATHEA-PATHFINDER is the time needed to perform a single integration step. This run-time depends on the integrator (we implemented the Euler integration as well as RK4, however only the latter is used) but also on the type of integration. We can identify four types of trajectories, which will give different run times:

- FLRW: At any photon location, we assign the gravitational field of an FLRW universe to compute the geodesic equations.
- NGP, CIC, TSC: We use the NGP, CIC, and TSC interpolation schemes to estimate the gravitational field from the AMR grid.

To perform our tests, we run MAGRATHEA-PATHFINDER on the French supercomputer Irene, on the Skylake partition composed of Intel Xeon Platinum 8168 processors. We propagate a photon in a given direction of the narrow light-cone of RayGal (with 400 deg^2 aperture and maximum redshift $z = 10$) using a single CPU (and single thread). We show the results in Table 1. For an FLRW trajectory, MAGRATHEA-PATHFINDER takes roughly 0.55 μs . As we do not need to estimate the gravitational field, the run-time is mainly that needed to compute the geodesic equations in Sect. 2.2 with the RK4 integrator. For NGP, we see that it takes 1.10 μs , which is twice the time of FLRW. The additional time is that needed to obtain the index of the most refined cell the photon is in, find it in the octree, and get the associated data. We note that we search for an index in a sorted vector, meaning that in principle, the larger the octree vector, the longer it takes to find the index (in our case the octree of the subdomain we consider contains roughly 3×10^8 elements). When using CIC, MAGRATHEA-PATHFINDER takes 6 μs to perform one integration step. It is expected that the CIC interpolation schemes takes more time than NGP, and at first one could have expected $9 \times 0.55 = 5 \mu\text{s}$ to compute the geodesic equations and find the eight neighbouring cells. The slight discrepancy can

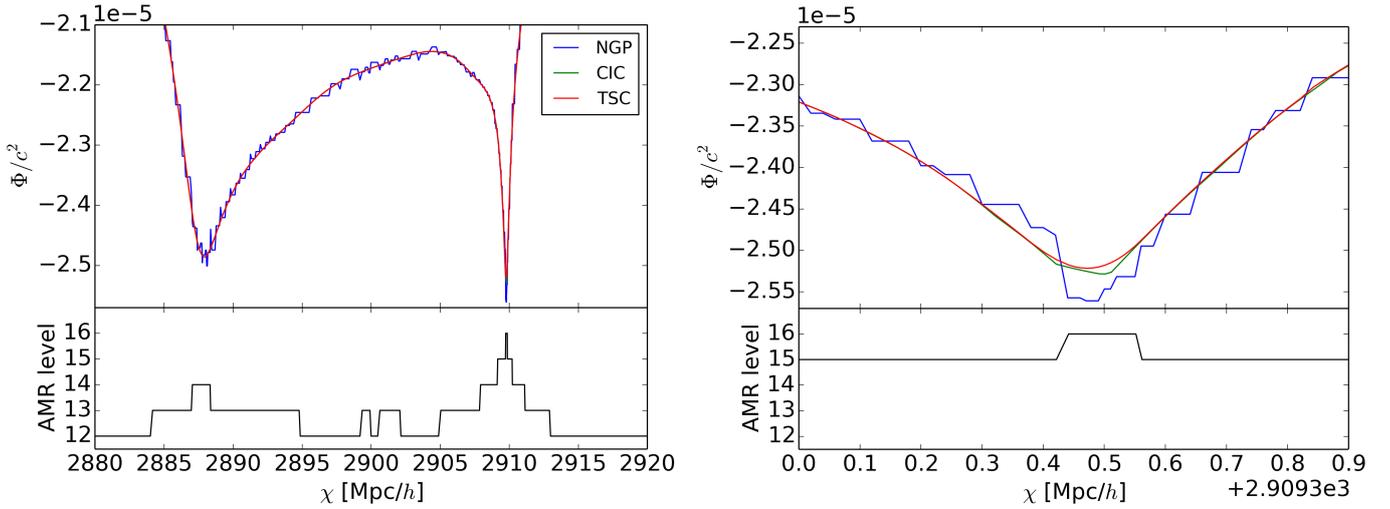


Fig. 2. Gravitational potential and AMR level along a photon trajectory, as a function of the comoving distance for different interpolation schemes. The *right panel* is a zoom onto the second potential well seen in the *left panel* around $2910 h^{-1}$ Mpc. The potential minima coincide with a higher clustering of matter, which refines the AMR grid (as seen in the subplots).

be explained by the fact that for CIC (and TSC) we need to perform the interpolation at some fixed level, and we loop over the coarser levels if an insufficient number of neighbours are found (see Sect. 2.2), which adds some additional run-time. Finally, we see that TSC takes $14.6 \mu\text{s}$, which is less than an optimistic expected run-time of $28 \times 0.55 + 1 = 16.4 \mu\text{s}$ from the 27 neighbouring cells (instead of 8 for CIC). This comes from the fact that we keep the neighbouring cells in memory, so that we do not need to find them again when the photon is in the same cell and we perform the interpolation at the same level as previously.

Last, we need to estimate how many steps N_{tot} are needed to reach a given comoving distance χ . Using the Λ CDM light cones of RayGal, with coarse cell size x_{coarse} and n_{steps} the number of steps per AMR cell (set by the user), we find that the total number of integration steps as a function of the distance is well fitted by

$$N_{\text{tot}}(\chi) \approx \left[1.33 - 2.37 \times 10^{-5} \chi \right] \frac{n_{\text{steps}}}{x_{\text{coarse}}} \chi, \quad (29)$$

between $z=0.1$ and $z=10$, where χ and x_{coarse} are in units of $h^{-1}\text{Mpc}$. If there was no AMR, we would expect $N_{\text{tot}}(\chi) = n_{\text{steps}} \chi / x_{\text{coarse}}$, which is different from Eq. (29). This difference comes from grid refinement (AMR), especially at low redshift where the late-time small-scale clustering is more important.

3.2. Accuracy of the interpolation schemes

One of the main features of AMR is the fact that we have access to very non-linear scales of structure formation in high-density regions. As our ray-tracing procedure adapts to the AMR level of the simulation light cone, we might wonder how well we recover the gravitational potential in these regions, and how it depends on the interpolation schemes described in Sect. 2.2. We use the methods in Sect. 2.6 to save the full trajectory of a single light ray for a given line of sight, with the three interpolation schemes previously described by setting the compile option, `-DORDER = 0, 1, and 2` for NGP, CIC, and TSC, respectively. We note that RayGal uses a TSC version of RAMSES to compute the potential, which means that for consistency we should use the same interpolation scheme. However, it can be interesting to visualise

the differences between these different types of interpolation and how it behaves with AMR.

In Fig. 2, we show the gravitational potential at the photon position for each integration step as a function of the comoving distance to the observer. Additionally, we also show the AMR level of the most refined cell at the photon location. In the left panel, we see two potential wells around 2887 and $2910 h^{-1}$ Mpc, which is roughly $z \approx 1.3$ in the fiducial cosmology of the simulation. These potential wells reach roughly -2.5×10^{-5} , which is slightly less than 10 km s^{-1} (the order of the gravitational redshift for galaxies and cluster). We can clearly see the difference between NGP and the other two interpolation schemes, as the former shows some sharp ‘jumps’ when the photon goes in a different cell while the others seem to exhibit a much smoother behaviour. At the same time, we see that the AMR level follows a similar pattern as the gravitational potential, starting from level 12 (which is the coarse level), and increases when the photon enters the potential wells. This is logical, because these potential wells come from small-scale clustering of matter, which also causes the N -body solver to refine the grid in these high-density regions. In particular, we see that for the second, deeper, and sharper potential well, the AMR level goes to 16 (while for the first well it only reaches level 14). The right panel shows a zoom of the second potential well in order to better visualise the differences between the interpolation schemes in these high-density regions. We indeed see the ‘stair’ behaviour of the NGP interpolation, while the CIC interpolation is very smooth. However, we can see that CIC seems to give a potential which linearly evolves by parts along the trajectory. This is expected, because CIC is a tri-linear interpolation and we use four steps per AMR cell. This means that the value of a field estimated by the same eight neighbours of the CIC scheme necessarily evolves linearly along any direction. Last, we see that the TSC scheme agrees very well with CIC far from the trough, while at the minimum of the potential it seems much smoother and would give better results when estimating its derivatives (which is important because weak lensing is sensitive to the second derivative of the gravitational potential) compared to CIC. This is also expected because TSC is higher order.

As a final note, we can see that although our method does not necessarily prevent discontinuities at the crossing between

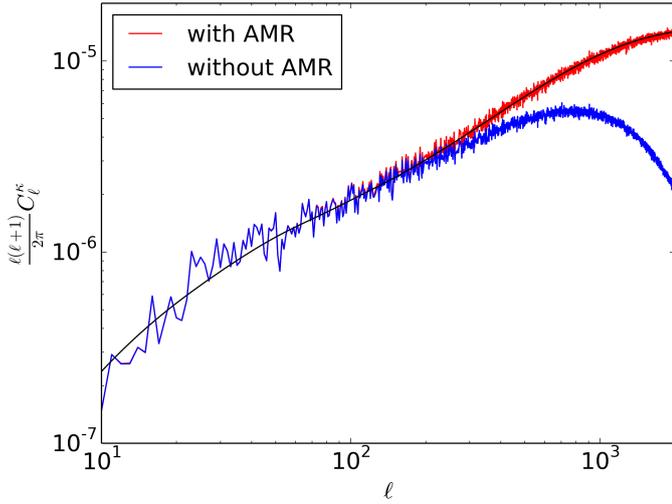


Fig. 3. Convergence power spectra with AMR (red) and without (blue) at $z=0.45$, using Healpix with $N_{\text{side}} = 2048$. The black line is the theoretical prediction computed using NICAIA (Kilbinger et al. 2017) with HALOFIT (Smith et al. 2003) parameters fitted on our simulation. When AMR is accounted for, we have a nice agreement between the numerical results and theoretical prediction. When it is not, the power spectrum experiences a damping at small angular scales (high ℓ), where the discrepancy becomes large already at $\ell \approx 300$.

AMR levels, we do not see sharp cuts in the gravitational potential when using either CIC or TSC. This suggests that our methodology should give good results even for very small scales.

3.3. The importance of AMR

Because most of the gravitational lensing power lies in small scales, we already expect AMR to be an important aspect of the ray-tracing procedure. In Fig. 3, we quantify its impact through the estimation of the angular power spectrum of the convergence computed with ANAFast on the full-sky light cone at $z=0.45$, between $\ell_{\text{min}} = 10$ and $\ell_{\text{max}} = 2000$ (because in that case $\ell_{\text{max}} \sim N_{\text{side}}$, which should give sufficient accuracy). First, we note that the angular power spectrum computed with AMR is in excellent agreement with the theoretical prediction. This validates our methodology (in the present case, that of the finite-beam method) to compute the lensing distortion matrix. Second, we see that the angular power spectrum without AMR (i.e. we only consider the coarse level for the ray tracing) departs early (around $\ell \sim 200\text{--}300$) from the AMR case and exhibits a strong damping at small scales. A similar trend was noted by Lepori et al. (2020). This shows that AMR is extremely important for recovering the correct statistical properties of gravitational lensing when modelling light propagation in a PM N -body simulation.

3.4. Propagation and number of steps per AMR cell

We now verify the appropriateness of our choice of making four integration steps per AMR cell. It was first shown in Reverdy (2014) that using four steps was ideal for correctly recovering the redshift up to double precision with respect to an analytical calculation when propagating an FLRW light ray with an RK4 integrator until $z=25$ in the DEUS-Full Universe Run simulations (Alimi et al. 2012; Rasera et al. 2014; Bouillot et al. 2015).

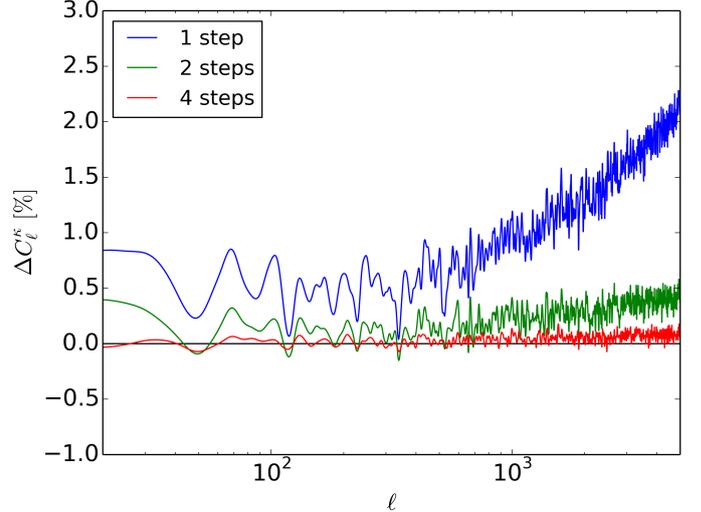


Fig. 4. Relative difference in the angular power spectrum of the convergence as a function of the number of steps per AMR cell during the propagation. The reference for the relative difference is taken as the angular power spectrum with eight steps per AMR cell. The convergence is computed using an infinitesimal method on a Healpix map with $N_{\text{side}} = 4096$ at $z=1.5$. We see that we achieve numerical convergence when using four steps per AMR cell.

However, it is unclear whether or not this choice is still relevant when accounting for an inhomogeneous universe.

In Fig. 4, we show the impact of taking one, two, or four steps on the convergence angular power spectrum with respect to the very conservative case where we use eight steps per AMR cell. To do so, we used a narrow light cone with 2500 deg^2 aperture, and produced Healpix maps (see Sect. 2.4) of the convergence at $z=1.5$ with different values of `nsteps`. To correctly estimate the power spectrum with a mask we use PolSpice (Szapudi et al. 2001; Chon et al. 2004). We see that the effect here is very small (at most a few percents). When taking one step per AMR cell, we see a roughly 0.5% bias at large angular scales, which increases up to 2% at $\ell = 5000$. For two steps, the angular power spectrum departs from the reference one around $\ell \approx 1000$ to reach 0.5% at most. Finally, we find that the convergence angular power spectrum with four steps is indistinguishable from its eight-steps counterparts. This is further evidence that there is no need to use more than four steps per AMR cell to propagate light rays.

We note that if subpercent precision is not needed, then two steps per AMR cell is sufficient. This should decrease the runtime by a factor of two and hence be very interesting for HPC. In any case, the user can decide to use an arbitrary number of steps by using the keyword `nsteps`.

3.5. Infinitesimal case, choice of the derivation step

Now we turn to the estimation of the lensing distortion matrix with the infinitesimal method as described in Sect. 2.3.1. The reason we only study the convergence of the infinitesimal method and not that of the finite-beam one is that the former depends on an arbitrarily chosen derivation step, while the latter does not depend on any arbitrary choice (except for the beam aperture at the observer, which is a parameter set by the user and is physically motivated, with its impact clearly understood from a theoretical perspective; see Fleury et al. (2019a); Breton & Fleury (2021)).

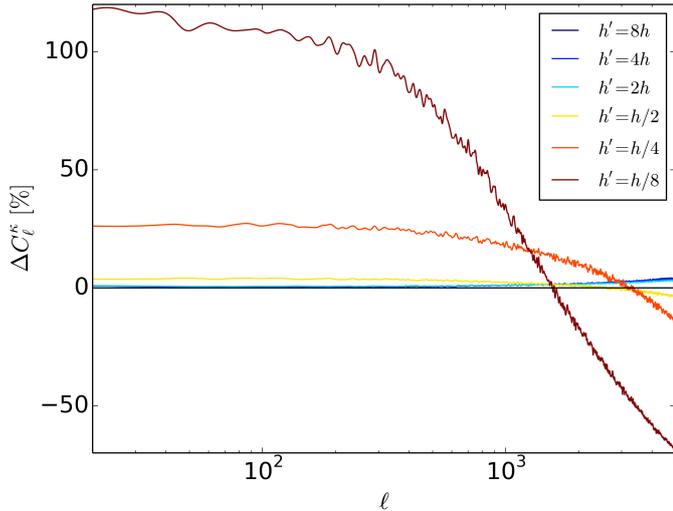


Fig. 5. Relative difference on the convergence angular power spectrum depending on the derivation step used in the infinitesimal method (see Sect. 2.3.1) to compute the lensing distortion matrix at $z = 1.9$, using Healpix with $N_{\text{side}} = 4096$. Using a larger derivation step considerably biases the estimation of the power spectrum, while smaller steps mostly affect the very small scales. Our choice for h seems ideal as it is stable for $\ell < 10^3$ and should not lead to any large bias at smaller scales.

The Laplacian of the gravitational potential along the line of sight in Eq. (12) is computed using finite differences of the force. To compute these differences, we need a derivation step h in Eq. (13), which is set to the size of the most refined cell the photon is in. In Fig. 5, we show the impact of the derivation step on the estimation of the convergence angular power spectrum. We multiplied the size of our step choice by a factor 8, 4, 2, 1/2, 1/4, and 1/8 to clearly test our method. For a derivation step smaller than our reference choice, we see that there is a large bias (several orders of magnitude larger than in Sect. 3.4) on the angular power spectrum even at linear scales (small ℓ), and then at damping at small scales (high ℓ). For derivation steps larger than the reference one, there is no noticeable difference at $\ell < 10^3$, while at $\ell > 10^3$ the angular power spectrum seems to be over-estimated with increasing h' . This overall behaviour shows that we find a ‘sweet spot’ when setting $h = 2^{\text{level}}$. This choice was also shown to lead to very good agreement with theoretical predictions from CLASS (Lesgourgues 2011) in Rasera et al. (2021).

3.6. Mean convergence and the Born approximation

Here, we estimate the impact of the commonly used Born approximation on weak-lensing quantities, and more precisely on the mean convergence. A standard test of the Born approximation with respect to real ray-tracing is to compute the convergence angular power spectrum and estimate the differences in both cases. In this particular configuration, the Born approximation is known to give results that are extremely close to real ray tracing (Hilbert et al. 2020), but this should be taken with caution as it can be deceiving when considering galaxy surveys. Indeed, to compute the angular power spectra, it is common to propagate light rays in the direction of homogeneously distributed pixels on a map (or plane). However, this procedure gives the same weight to each pixel, which is not what really happens in galaxy surveys where solid angles on the sky are magnified. This is the difference between direction-averaging and source-averaging

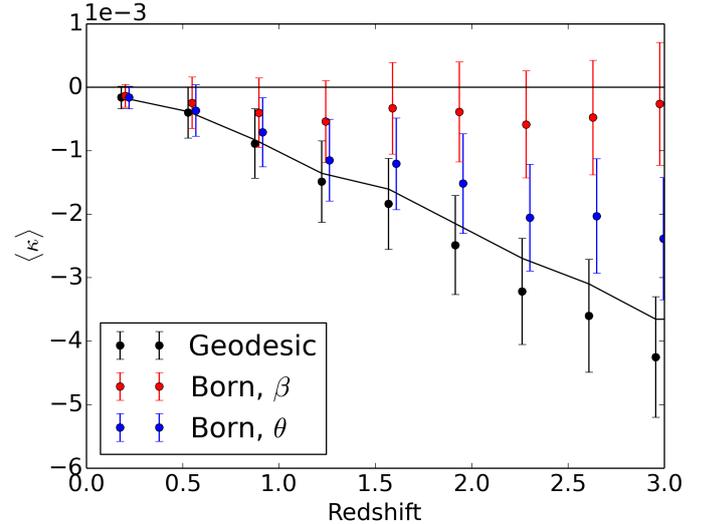


Fig. 6. Mean convergence on the deep narrow cone of RayGal (400 deg^2 aperture) as a function of redshift, when evaluating the distortion matrix along null geodesics (black) or using the Born approximation in the direction of sources (β , red) or images (θ , blue). The black solid line shows the expected value of the mean convergence when using null geodesics, that is $\langle \kappa \rangle = \frac{1}{2}(\langle \mu \rangle - 1) - 2 \langle \kappa \rangle^2$, where the averaged quantities are numerically evaluated. The error bars contain Poisson and super-sample variance (Breton & Fleury 2021). The red and black points are consistent with the expectation values for direction and source averaging procedures, respectively. The blue points, which come from a hybrid prescription, seem to lie between the former two.

(Kibble & Lieu 2005; Bonvin et al. 2015; Kaiser & Peacock 2016; Breton & Fleury 2021). When propagating light rays in random (or statistically isotropic) directions, it is expected that the Born approximation should give overall similar results to real ray tracing (Breton & Fleury 2021). In particular, in both cases, we expect $\langle \kappa \rangle = 0$ (which also implies $\langle \mu \rangle > 1$ at second order), because photons evenly sample the sky. However, when we compute the null geodesics between the observer and sources, it is known that $\langle \mu \rangle = 1$ (Weinberg 1976) and $\kappa < 0$. Indeed, photons tend to propagate in more under-dense regions between two fixed points for several realisations of the matter density field.

In Fig. 6, we compute the distortion matrix for a given simulated halo catalogue and estimate the averaged convergence in tomographic redshift bins. In any case, we use the infinitesimal method in Sect. 2.3.1, and distinguish three different types of trajectories: first we consider the null geodesics between the observer and sources; then we use the Born approximation where photons propagate in straight lines between the observer and the sources or the source images. We note that the last case is strictly theoretical, because to the best of our knowledge it is not necessarily used nor discussed, but is interesting from a pedagogical point of view to cover all the possible scenarios. We see that for the standard Born approximation, that is when photons propagate in straight lines in the source direction (β), the average convergence is consistent with zero within error bars, which is expected. For null geodesics, we find that the mean convergence is indeed negative and closely follows the theoretical prediction. Interestingly, when we use the Born approximation in the direction of images, the result seems to lie in between the first two cases.

There are two main implications of this result: first, a precise theoretical modelling of weak-lensing observables might need to account for the fact that $\langle \kappa \rangle \neq 0$ for galaxy surveys. Secondly

(and more importantly), one should take the distortion matrix evaluated using the Born approximation with caution when constructing realistic simulated catalogues, because in this case, the mean magnification is superior to unity (while in reality it should be strictly equal before the flux cut), and furthermore the magnification probability density function is that of the null-geodesic case multiplied by μ (this was also discussed in [Takahashi et al. 2011](#)). When emulating a flux-limited survey, one has to magnify the fluxes (or magnitude) with the magnification; however, if one uses the Born approximation, the final number count on the flux-limited sample will be larger than it should realistically be in comparison to observations.

3.7. Relativistic effects and galaxy clustering analyses

Last, we discuss the importance of the global relativistic treatment for light propagation. While ray-tracing codes usually only allow gravitational lensing studies, MAGRATHEA-PATHFINDER offers a unified framework which accounts for both gravitational lensing and redshift perturbations. For the latter, we implemented all the corrections at first order in metric perturbations (as seen in Sect. 2.5). This allows in particular to study the impact of relativistic effects for galaxy clustering analysis. The full relativistic number count was analytically estimated within linear theory by [Yoo et al. \(2009\)](#), [Bonvin & Durrer \(2011\)](#), and [Challinor & Lewis \(2011\)](#), and it was shown by [Bonvin et al. \(2014\)](#) that the dipole of the correlation function could be very sensitive to the gravitational potential and could therefore be a useful probe to test the nature of gravity ([Bonvin & Fleury 2018](#)). [Breton et al. \(2019\)](#) found a nice agreement at large scale with linear-theory-based predictions, which shows the accuracy of our method. However, it was noted that the small scales are completely dominated by the gravitational potential, which is far beyond the expectation from linear theory. This was later analytically modelled using non-linear prescriptions ([Di Dio & Seljak 2019](#); [Saga et al. 2020, 2022](#); [Beutler & Di Dio 2020](#)). The dipole of the correlation function of galaxies is expected to be detected with a high signal-to-noise ratio for next-generation galaxy surveys ([Saga et al. 2022](#)), for which it will be mandatory to perform a full relativistic treatment of light propagation.

4. Conclusion

In this paper, we present MAGRATHEA-PATHFINDER, a ray-tracing framework which post-processes numerical simulations to accurately rebuild the past light cone of an observer from linear to non-linear scales. This framework propagates light rays in the AMR structure of a simulation light cone by solving the null geodesic equations of a perturbed FLRW metric in the Newtonian gauge and does not resort to any further approximation. Moreover, MAGRATHEA-PATHFINDER is optimised for HPC and has already been run on very large N -body simulations such as DEUS- Full Universe Run ([Alimi et al. 2012](#)) and RayGal ([Rasera et al. 2021](#)).

Our code produces relativistic simulated catalogues (where the null geodesic between the observer and sources are identified), Healpix maps, and various light-ray statistics along their propagation. By accounting for all the effects at first order in metric perturbations, MAGRATHEA-PATHFINDER opens up a wide range of possible applications, some of which have already been studied in the literature:

- Weak gravitational lensing: [Rasera et al. \(2021\)](#) studied the impact of relativistic effects on various lensing-matter

angular power spectra. In particular, they emphasised the importance of peculiar velocities and magnification bias beyond linear order. Furthermore, the weak gravitational lensing properties of light beams with finite extension is different from those of an infinitesimal beam. This effect was successfully confronted to theoretical predictions for the convergence and shear angular power spectra in [Breton & Fleury \(2021\)](#).

- Galaxy clustering and relativistic effects: The simulated catalogues produced by MAGRATHEA-PATHFINDER contain both the comoving and apparent angular position, as well as the full redshift decomposition for a given source. From this, one can perform galaxy clustering analysis beyond standard assumptions (distant observer, redshift only perturbed by peculiar velocities, etc.). In particular, in [Breton et al. \(2021\)](#) the authors studied the impact of gravitational lensing (also known as magnification bias) on the estimation of the growth rate of structure when performing a standard RSD analysis and found that if lensing is not accounted for in the modelling, this leads to an underestimation of $f\sigma_8$. These catalogues also enabled for the first time the study of relativistic effects on the dipole of the correlation function at all scales ([Breton et al. 2019](#); [Taruya et al. 2020](#); [Saga et al. 2020](#)), which will be detectable in next-generation surveys with high signal-to-noise ratios ([Saga et al. 2022](#)). Using the same data, [Beutler & Di Dio \(2020\)](#) performed a similar analysis on the power spectrum dipole.
- Distance measures: Distance measures are crucial to interpreting observational data and cosmological inference. When it comes to the Hubble diagram, the average distance is often modelled by that of an FLRW universe. [Breton & Fleury \(2021\)](#) carried out a detailed study of the perturbations on the distance–redshift relation for a wide range of redshifts. They numerically tested Weinberg’s conjecture ([Weinberg 1976](#)), which states that the area of constant redshift is unaffected by inhomogeneities in the matter density field and showed that even a full non-linear treatment gives similar results to the theoretical predictions of [Kaiser & Peacock \(2016\)](#) regarding possible biases.
- Integrated Sachs-Wolfe effect: [Adamek et al. \(2020\)](#) used MAGRATHEA-PATHFINDER on the DEUS- Full Universe Run simulations up to $z = 25$ to study the ISW angular power spectra for various scenarios such as Λ CDM, phantom dark energy, and modified gravity. In any case, the authors found good agreement with theoretical predictions.

These works show the importance of an exhaustive modelling of the light cone in order to correctly interpret future surveys, which will probe the Universe at the largest scales with unprecedented precision. In particular, with increasing data accuracy, omissions in the model might lead to biases in the inference of cosmological parameters. The methods developed in MAGRATHEA-PATHFINDER will be useful, in coordination with theoretical predictions and observations, for refining existing probes and constructing new ones in order to shed light on the true nature of our Universe.

Acknowledgements. We thank Yann Rasera for helpful comments on the draft, and the Laboratoire Univers et Théories for its support and for providing a stimulating scientific environment from the early development of MAGRATHEA to the last raytracing analyses with MAGRATHEA-PATHFINDER. The design, development, and implementation of the foundational MAGRATHEA library as well as the initial version of MAGRATHEA-PATHFINDER were carried out at LUTH during VR’s PhD under the direction of Jean-Michel Alimi and Yann Rasera. The further developments of MAGRATHEA-PATHFINDER were carried out during MAB’s thesis under the direction of Yann Rasera. We thank Jean-Michel

Alimi and Yann Rasera for their scientific contributions and LUTH for its hospitality. MAB also thanks Julian Adamek for an early implementation of Healpix in the MAGRATHEA-PATHFINDER code. VR also thanks the countless dedicated developers of the software stack without which none of this would have been possible. This work was granted access to HPC resources of TGCC/CINES through allocations made by GENCI (Grand Équipement National de Calcul Intensif) under the allocation 2020-A0070402287.

References

- Abbott, T., Allam, S., Andersen, P., et al. 2019, *ApJ*, **872**, L30
- Adamek, J., Clarkson, C., Coates, L., Durrer, R., & Kunz, M. 2019, *Phys. Rev. D*, **100**, 021301
- Adamek, J., Rasera, Y., Corasaniti, P. S., & Alimi, J.-M. 2020, *Phys. Rev. D*, **101**, 023512
- Alam, S., Aubert, M., Avila, S., et al. 2021, *Phys. Rev. D*, **103**, 083533
- Alimi, J.-M., Bouillot, V., Rasera, Y., et al. 2012, ArXiv e-prints [arXiv:1206.2838]
- Aubert, D., Deparis, N., & Ocvirk, P. 2015, *MNRAS*, **454**, 1012
- Barreira, A., Linares, C., Bose, S., & Li, B. 2016, *JCAP*, **5**, 001
- Bartelmann, M., & Schneider, P. 2001, *Phys. Rep.*, **340**, 291
- Beutler, F., & Di Dio, E. 2020, *JCAP*, **2020**, 048
- Blandford, R., & Narayan, R. 1986, *ApJ*, **310**, 568
- Bonvin, C., & Durrer, R. 2011, *Phys. Rev. D*, **84**, 063505
- Bonvin, C., & Fleury, P. 2018, *JCAP*, **2018**, 061
- Bonvin, C., Hui, L., & Gaztañaga, E. 2014, *Phys. Rev. D*, **89**, 083535
- Bonvin, C., Clarkson, C., Durrer, R., Maartens, R., & Umeh, O. 2015, *JCAP*, **2015**, 040
- Borzyszkowski, M., Bertacca, D., & Porciani, C. 2017, *MNRAS*, **471**, 3899
- Bouillot, V. R., Alimi, J.-M., Corasaniti, P.-S., & Rasera, Y. 2015, *MNRAS*, **450**, 145
- Breton, M.-A. & Fleury, P. 2021, *A&A*, **655**, A54
- Breton, M.-A., Rasera, Y., Taruya, A., Lacombe, O., & Saga, S. 2019, *MNRAS*, **483**, 2671
- Breton, M.-A., de la Torre, S., & Piat, J. 2022, *A&A*, **661**, A154
- Bryan, G. L., Norman, M. L., O’Shea, B. W., et al. 2014, *APJS*, **211**, 19
- Cai, Y.-C., Cole, S., Jenkins, A., & Frenk, C. S. 2010, *MNRAS*, **407**, 201
- Caprini, C., & Tamanini, N. 2016, *JCAP*, **10**, 006
- Challinor, A. & Lewis, A. 2011, *Phys. Rev. D*, **84**, 043516
- Chon, G., Challinor, A., Prunet, S., Hivon, E., & Szapudi, I. 2004, *MNRAS*, **350**, 914
- Couchman, H. M. P. 1991, *ApJ*, **368**, L23
- DESI Collaboration (Aghamousa, A., et al.) 2016, ArXiv e-prints [arXiv:1611.00036]
- Di Dio, E., & Seljak, U. 2019, *JCAP*, **2019**, 050
- Fabbian, G., Calabrese, M., & Carbone, C. 2018, *JCAP*, **2018**, 050
- Fleury, P., Larena, J., & Uzan, J.-P. 2017, *Phys. Rev. Lett.*, **119**, 191101
- Fleury, P., Larena, J., & Uzan, J.-P. 2019a, *Phys. Rev. D*, **99**, 023525
- Fleury, P., Larena, J., & Uzan, J.-P. 2019b, *Phys. Rev. D*, **99**, 023526
- Fluke, C. J., & Lasky, P. D. 2011, *MNRAS*, **416**, 1616
- Fluke, C. J., Webster, R. L., & Mortlock, D. J. 1999, *MNRAS*, **306**, 567
- Fosalba, P., Gaztañaga, E., Castander, F. J., & Manera, M. 2008, *MNRAS*, **391**, 435
- Freedman, W. L., Madore, B. F., Hatt, D., et al. 2019, *ApJ*, **882**, 34
- Garrison, L. H., Eisenstein, D. J., Ferrer, D., Maksimova, N. A., & Pinto, P. A. 2021, *MNRAS*, **508**, 575
- Giblin, Jr., J. T., Mertens, J. B., & Starkman, G. D. 2016, *ApJ*, **833**, 247
- Giocoli, C., Metcalf, R. B., Baldi, M., et al. 2015, *MNRAS*, **452**, 2757
- Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, *ApJ*, **622**, 759
- Gouin, C., Gavazzi, R., Pichon, C., et al. 2019, *A&A*, **626**, A72
- Hamilton, A. J. S. 1992, *ApJ*, **385**, L5
- Heitmann, K., Finkel, H., Pope, A., et al. 2019, *ApJS*, **245**, 16
- Hilbert, S., Hartlap, J., White, S. D. M., & Schneider, P. 2009, *A&A*, **499**, 31
- Hilbert, S., Barreira, A., Fabbian, G., et al. 2020, *MNRAS*, **493**, 305
- Hockney, R. W., & Eastwood, J. W. 1988, *Computer Simulation Using Particles*, (Bristol: Adam Hilger)
- Holz, D. E., & Hughes, S. A. 2005, *ApJ*, **629**, 15
- Ishiyama, T., Prada, F., Klypin, A. A., et al. 2021, *MNRAS*, **506**, 4210
- Jain, B., Seljak, U., & White, S. 2000, *ApJ*, **530**, 547
- Kaiser, N. 1987, *MNRAS*, **227**, 1
- Kaiser, N., & Peacock, J. A. 2016, *MNRAS*, **455**, 4518
- Kibble, T. W. B., & Lieu, R. 2005, *ApJ*, **632**, 718
- Kilbinger, M., Heymans, C., Asgari, M., et al. 2017, *MNRAS*, **472**, 2126
- Killedar, M., Lasky, P. D., Lewis, G. F., & Fluke, C. J. 2012, *MNRAS*, **420**, 155
- Knebe, A., Green, A., & Binney, J. 2001, *MNRAS*, **325**, 845
- Koksbang, S. M. & Hannestad, S. 2015, *Phys. Rev. D*, **91**, 043508
- Komatsu, E., Smith, K. M., Dunkley, J., et al. 2011, *ApJS*, **192**, 18
- Kravtsov, A. V., Klypin, A. A., & Khokhlov, A. M. 1997, *ApJS*, **111**, 73
- Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, ArXiv e-prints [arXiv:1110.3193]
- Lepori, F., Adamek, J., Durrer, R., Clarkson, C., & Coates, L. 2020, *MNRAS*, **497**, 2078
- Lesgourgues, J. 2011, ArXiv e-prints [arXiv:1104.2932]
- Metcalf, R. B., & Petkova, M. 2014, *MNRAS*, **445**, 1942
- Perlmutter, S., Aldering, G., della Valle, M., et al. 1998, *Nature*, **391**, 51
- Planck Collaboration VI. 2020, *A&A*, **641**, A6
- Potter, D., Stadel, J., & Teyssier, R. 2017, *Comput. Astrophys. Cosmol.*, **4**, 2
- Rasera, Y., Corasaniti, P.-S., Alimi, J.-M., et al. 2014, *MNRAS*, **440**, 1420
- Rasera, Y., Breton, M.-A., Corasaniti, P.-S., et al. 2022, *A&A*, **661**, A90
- Reverdy, V. 2014, PhD thesis, Laboratoire Univers et Théories, <https://hal.archives-ouvertes.fr/tel-02095297/document>
- Reverdy, V., & Alimi, J.-M. 2015, *Achieving Genericity and Performance using Embedded Domain Specific Languages*
- Riess, A. G., Filippenko, A. V., Challis, P., et al. 1998, *AJ*, **116**, 1009
- Riess, A. G., Casertano, S., Yuan, W., Macri, L. M., & Scolnic, D. 2019, *ApJ*, **876**, 85
- Saga, S., Taruya, A., Breton, M.-A., & Rasera, Y. 2020, *MNRAS*, **498**, 981
- Saga, S., Taruya, A., Breton, M.-A., & Rasera, Y. 2021, ArXiv e-prints [arXiv:2112.07727]
- Saga, S., Taruya, A., Rasera, Y., & Breton, M.-A. 2022, *MNRAS*, **511**, 2732
- Schneider, P., Ehlers, J., & Falco, E. E. 1992, *Gravitational Lenses*, 112
- Scolnic, D. M., Jones, D. O., Rest, A., et al. 2018, *ApJ*, **859**, 101
- Smith, R. E., Peacock, J. A., Jenkins, A., et al. 2003, *MNRAS*, **341**, 1311
- Springel, V. 2005, *MNRAS*, **364**, 1105
- Szapudi, I., Prunet, S., Pogosyan, D., Szalay, A. S., & Bond, J. R. 2001, *ApJ*, **548**, L115
- Takahashi, R., Oguri, M., Sato, M., & Hamana, T. 2011, *ApJ*, **742**, 15
- Taruya, A., Saga, S., Breton, M.-A., Rasera, Y., & Fujita, T. 2020, *MNRAS*, **491**, 4162
- Teyssier, R. 2002, *A&A*, **385**, 337
- Weinberg, S. 1976, *ApJ*, **208**, L1
- Wong, K. C., Suyu, S. H., Chen, G. C. F., et al. 2019, *MNRAS*, **498**, 1420
- Yoo, J., Fitzpatrick, A. L., & Zaldarriaga, M. 2009, *Phys. Rev. D*, **80**, 083514