








# A Hamiltonian, post-Born, three-dimensional, on-the-fly ray tracing algorithm for gravitational lensing

Alan Junzhe Zhou <sup>a,b</sup>, Yin Li <sup>c</sup>, Scott Dodelson <sup>a,b</sup>, Rachel Mandelbaum <sup>a,b</sup>,  
Yucheng Zhang <sup>b,c</sup>, Xiangchong Li <sup>a,b</sup> and Giulio Fabbian <sup>d,e,f,g</sup>

<sup>a</sup>Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

<sup>b</sup>McWilliams Center for Cosmology and Astrophysics, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

<sup>c</sup>Department of Mathematics and Theory, Peng Cheng Laboratory, Shenzhen, Guangdong 518000, China

<sup>d</sup>Institute of Astronomy, Madingley Road, Cambridge CB3 0HA, U.K.

<sup>e</sup>Kavli Institute for Cosmology Cambridge, Madingley Road, Cambridge CB3 0HA, U.K.

<sup>f</sup>School of Physics and Astronomy, Cardiff University, The Parade, Cardiff, CF24 3AA, U.K.

<sup>g</sup>Center for Computational Astrophysics, Flatiron Institute, New York, New York 10010, U.S.A.

E-mail: [ajzhou@uchicago.edu](mailto:ajzhou@uchicago.edu), [eelregit@gmail.com](mailto:eelregit@gmail.com), [dodelson@fnal.gov](mailto:dodelson@fnal.gov),  
[rmandelb@andrew.cmu.edu](mailto:rmandelb@andrew.cmu.edu), [yucheng.zhang@nyu.edu](mailto:yucheng.zhang@nyu.edu), [xli6@bnl.gov](mailto:xli6@bnl.gov),  
[FabbianG@cardiff.ac.uk](mailto:FabbianG@cardiff.ac.uk)

**ABSTRACT:** The analyses of the next generation cosmological surveys demand an accurate, efficient, and differentiable method for simulating the universe and its observables across cosmological volumes. We present Hamiltonian ray tracing (HRT) — the first post-Born (accounting for lens-lens coupling and without relying on the Born approximation), three-dimensional (without assuming the thin-lens approximation), and on-the-fly (applicable to any structure formation simulations) ray tracing algorithm based on the Hamiltonian formalism. HRT performs symplectic integration of the photon geodesics in a weak gravitational field, and can integrate tightly with any gravity solver, enabling co-evolution of matter particles and light rays with minimal additional computations. We implement HRT in the particle-mesh library pmwd, leveraging hardware accelerators such as GPUs and automatic differentiation capabilities based on JAX. When tested on a point-mass lens, HRT achieves sub-percent accuracy in deflection angles above the resolution limit across both weak and moderately strong lensing regimes. We also test HRT in cosmological simulations on the convergence maps and their power spectra.

**KEYWORDS:** cosmological simulations, dark matter simulations, gravitational lensing, weak gravitational lensing

ARXIV EPRINT: [2405.12913](https://arxiv.org/abs/2405.12913)

---

**Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Hamiltonian dynamics of light rays</b>	<b>3</b>
<b>3</b>	<b>Symplectic ray tracing</b>	<b>5</b>
3.1	Equations of motion in discrete time	5
3.2	Time resolution correction	6
<b>4</b>	<b>Weak lensing observables</b>	<b>6</b>
<b>5</b>	<b>Implementation</b>	<b>8</b>
5.1	Efficient dark matter and photons co-evolution	8
5.2	The kick operator	8
5.3	Adaptive ray mesh	10
<b>6</b>	<b>Validations</b>	<b>10</b>
6.1	Lensing by a point mass	10
6.2	Post-Born weak lensing in a cosmological volume	12
<b>7</b>	<b>Conclusions</b>	<b>15</b>
<b>A</b>	<b>Hamiltonian dynamics of photons lensed by weak gravitational field</b>	<b>17</b>
<b>B</b>	<b>Symplectic integrator</b>	<b>19</b>
<b>C</b>	<b>Adaptive ray mesh</b>	<b>20</b>
<b>D</b>	<b>Other considerations and convergence tests for point mass lensing</b>	<b>21</b>
<b>E</b>	<b>Convergence tests for weak lensing power spectrum</b>	<b>22</b>

---

**1 Introduction**

Next-generation galaxy, lensing, and Cosmic Microwave Background (CMB) surveys promise percent-level or better constraints on cosmological and astrophysical parameters. This discovery potential, however, hinges on our ability to quickly and accurately simulate the universe and its observables over cosmological volumes. For example, computing covariance matrices for two-point cosmological analyses using mock catalogs requires a large number of mock catalogs [1–4]. Higher-order statistical methods, such as peak and void counts, require many high-fidelity simulations to model how observables and their covariances depend on cosmology [5, 6]. Simulation-based inference (SBI) needs a large number of simulations for its training set [7]. Finally, field-level inference (FLI) demands a fast and differentiable forward

model to simulate the universe and to construct observables thereof [8–26]. Perhaps not surprisingly, photons are the primary carriers of information for most cosmological observables. A photon conveys information from both its source (e.g., the temperature or polarization of the primordial CMB, or the positions of galaxies) as well as the intervening matter structures through which it traverses. In this work, we propose a new ray tracing algorithm called Hamiltonian ray tracing (HRT) that 1) accounts for the actual light trajectory and lens-lens couplings (post-Born effects in short); 2) is differentiable and computed on-the-fly; and 3) incurs minimal computational overhead when integrated with a background N-body algorithm.

The Born approximation assumes that lensing observables (e.g., weak lensing convergence  $\kappa$  or the CMB deflection angle) accumulate along a photon’s unperturbed path and consider subsequent lensing events as independent from each other. Post-Born corrections effectively accounts for the ray deflection during a photons propagation and for the so-called lens-lens coupling, which describes how gravitational lenses at different redshifts can interact to generate rotational modes in the observable fields [27–29]. Post-Born effects introduce additional non-Gaussianities into lensing observables besides those generated by the non-linear clustering of the matter. Thus, it is important to include them in the modeling of higher-order statistics for both CMB and galaxy lensing. In particular, in the case of CMB lensing, non-linear and post-Born effects have similar amplitude [30, 31]. For example, post-Born effects significantly change the higher-order moments or peaks statistics of galaxy weak lensing convergence maps [32, 33]. These effects translate into cosmological biases for analysis that targets the aforementioned higher-order statistics. Moreover, post-Born corrections affect FLI through the galaxy delensing effect. As demonstrated by ref. [34], galaxies at  $z = 1$  are typically deflected on an arc minute scale, which is the typical pixel size in an FLI analysis. If the post-Born effects are unaccounted for, the FLI’s forward model will generate observable fields that are coherently shifted with respect to the truth. However, since all existing lensing FLI models to date assume the Born approximation [24, 25, 35] and there does not yet exist a forward model that captures post-Born effects, the resulting cosmological biases are not well understood. Post-Born effects are less disruptive for two-point statistics, although the corrections could still be measured for the highest multipoles of the shear and lensing convergence power spectra at the level of sensitivity of future CMB lensing experiments and galaxy surveys [27, 28, 33, 36–41]. Similar conclusions apply for cross-correlations between CMB lensing and galaxy survey probes [42].

Weak lensing simulations typically capture post-Born effects using the multiple lens plane (MLP) formalism [37, 43–45], which has been widely applied in many studies [31, 32, 41, 46–54]. Our new HRT formalism improves upon the MLP method in ways that enable new cosmological analysis methods. *First*, whereas MLP deflects photons only by the gravitational potential gradients instantaneously at a lens plane, HRT utilizes the gravitational potential of the entire simulated volume without the thin-lens approximation, a feature known as three-dimensional ray tracing [43, 55–58]. *Second*, the MLP algorithm requires initially running an N-body simulation, storing its time steps (called snapshots), preparing the density field on the light cone in mass shells, and finally accumulating the convergence fields backward in time. This multi-step process decouples ray tracing from the N-body simulations, complicating implementation when varying cosmology and introducing computational overhead in terms of

data storage and transfer. Our HRT method performs ray tracing on-the-fly by co-evolving the light rays with the matter particles. The physics and the time-stepping of the rays are tightly synchronized with the N-body simulation and subject to the same cosmological model, thereby eliminating the need for data storage. Refs. [56, 57]’s algorithm performs three-dimensional ray tracing on-the-fly, but ignore post-Born effects and lack differentiability and other features listed below. *Third*, FLI requires that the observable fields be differentiable with respect to all cosmological parameters and the initial conditions of the background N-body simulation. Partly because of the multi-step process, current MLP codes are not differentiable. HRT formalism leverages a combination of automatic differentiation routines in JAX and variational equation to achieve efficient differentiation with respect to all parameters of interest. *Fourth*, MLP typically uses Kaiser-Squires algorithm [37] or finite-difference [45] methods to compute lensing observables, which could suffer from numerical instability. As we shall see, since most lensing observables are defined in terms of derivatives of the photon trajectory with respect to its initial position, and our new methods are differentiable, we can calculate these observables by direct differentiation. *Fifth*, unlike MLP which approximates the total lensing deflection as a Riemann sum over the radial comoving space, HRT solves each photon’s (light ray’s) equations of motion (EOMs) using a highly stable and accurate symplectic integrator. This provides a systematic way to correct for the finite time-resolution of the snapshots. *Finally*, MLP implementations do not currently leverage hardware accelerators while HRT is implemented on both CPUs and GPUs. Moreover, on-the-fly ray-tracing is particularly suited for GPU applications since GPU-based N-body simulations are already memory limited and cannot afford to store snapshots and post-process them later into shear maps.

The paper is organized in the following way. We first review the Hamiltonian formulation of the photon dynamics in a gravitational field in section 2. Next, we discretize the EOMs to construct a symplectic integrator for the photons’ trajectories in section 3. Once we have the trajectories, we construct an algorithm that calculates the weak lensing observables along the light path in section 4. We discuss an efficient implementation of the algorithm in section 5, and test the algorithm for point mass deflection and weak lensing maps in cosmological volumes in section 6. We conclude in section 7.

## 2 Hamiltonian dynamics of light rays

In post-Born ray tracing, the trajectories of photons are deflected by the large-scale structure. Consequently, it is not known a priori if a photon emitted by a source will reach the observer. The core idea of HRT is to evolve both light rays (from its observed position on the image plane, denoted by  $\theta_0$ ) and the matter particles (from their displacements and velocities at today) backward in time in an N-body simulation<sup>1</sup> The light ray EOMs are reversible once we describe them with Hamiltonian dynamics, which we then implement numerically using symplectic integration. This approach involves three parts: constructing the spacetime metric with gravitational perturbations, defining the photon’s Hamiltonian using this metric, and

---

<sup>1</sup>We can backward evolve the dark matter particles by reversing the time variable in their EOMs. Table 2 of ref. [8] has shown that both the positions and velocities of the dark matter particles can be recovered to high accuracy across different simulation configurations in pmwd when the EOMs are solved with a symplectic integrator.

finally, solving Hamilton's equations. We only motivate the results in this section and leave the detailed derivations to appendix A.

Let us parameterize spacetime with conformal time  $\tau$  and comoving spatial coordinates  $\mathbf{x}$ . On scales smaller than the cosmic horizon, the matter density contrast  $\delta(\mathbf{x}, \tau)$  sources a gravitational potential field  $\Phi(\mathbf{x}, \tau)$  via Poisson's equation,

$$\nabla^2 \Phi(\mathbf{x}, \tau) = \frac{3}{2} \frac{\Omega_m H_0^2}{a(\tau)} \delta(\mathbf{x}, \tau), \quad (2.1)$$

where the gradient (unless specified) is with respect to the comoving coordinates. Here,  $a$ ,  $\Omega_m$ , and  $H_0$  represent the scale factor, matter fraction, and Hubble parameter, respectively. The potential  $\Phi(\mathbf{x}, \tau)$  perturbs the otherwise homogeneous and isotropic Friedmann-Lemaître-Robertson-Walker metric, given by

$$ds^2 = -a^2 \left(1 + 2 \frac{\Phi}{c^2}\right) d\tau^2 + a^2 h_{ij} dx^i dx^j \quad (2.2)$$

$$\equiv -a^2 \left(1 + 2 \frac{\Phi}{c^2}\right) d\tau^2 + a^2 \left(1 - 2 \frac{\Phi}{c^2}\right) (r^2 d\boldsymbol{\theta}^2 + d\chi^2), \quad (2.3)$$

for  $i = 1, 2, 3$ .  $r$  and  $\chi$  are the transverse and radial comoving distances, respectively, and  $r = \chi$  in a flat universe. We parameterize the sky under the flat-sky approximation with angular coordinates, i.e.,  $\mathbf{x} = (\boldsymbol{\theta}, \chi)$ .

Now, consider a photon. The general relativistic covariant Hamiltonian of a photon is given by

$$H(x^j, p_i, \tau) = c \sqrt{h^{ij} p_i p_j} \left(1 + 2 \frac{\Phi}{c^2}\right), \quad (2.4)$$

where  $p_i$  is the momentum one-form [59]. The EOMs of the photon are solutions to Hamilton's equations. The results are (see detailed proof and discussion in appendix A; ref. [59] also derives a similar result albeit with a different parameterization of the image plane)

$$\frac{d\boldsymbol{\theta}}{d\chi} = -\frac{\boldsymbol{\eta}}{r^2}, \quad (2.5)$$

$$\frac{d\boldsymbol{\eta}}{d\chi} = \frac{2}{c^2} \frac{\partial \Phi}{\partial \boldsymbol{\theta}}, \quad (2.6)$$

$$\frac{d\tau}{d\chi} = -\frac{1}{c} \left(1 - 2 \frac{\Phi}{c^2} + \frac{\boldsymbol{\eta}^2}{2r^2}\right), \quad (2.7)$$

where we have defined the conjugate momentum

$$\boldsymbol{\eta} = \frac{r\mathbf{v}}{c}, \quad (2.8)$$

and  $\mathbf{v}$  is the *transverse* peculiar velocity of the photon. Eq. (2.5) is simply a geometric relation in the deflection tangent:  $-rd\boldsymbol{\theta}/d\chi = \mathbf{v}/c$ . The second and third terms in eq. (2.7) correspond to the Shapiro and the geometric time delay, respectively.  $\boldsymbol{\theta}$  has the unit of angle while  $\boldsymbol{\eta}$  has the unit of length.<sup>2</sup> The initial conditions are given by

$$\boldsymbol{\theta}(\chi_0) = \boldsymbol{\theta}_0, \quad \boldsymbol{\eta}(\chi_0) = 0, \quad \chi_0 = 0. \quad (2.9)$$

---

<sup>2</sup>They are actually conjugate variables in a separable Hamiltonian and relates to the Fermat action as shown in appendix B. In fact, the time delay terms can also be seen easily from the separable Hamiltonian.

To arrive at these EOMs, we have used the flat-sky (small angle) approximation, kept  $\Phi/c^2$ ,  $\nabla\Phi/c^2$ , and assumed  $(\mathbf{v}/c)^2$  is of  $\mathcal{O}(\Phi/c^2)$ . We have also used the comoving distances instead of conformal time to parameterize the photon trajectory. The two are related by eq. (2.7), where the second term is the Shapiro time delay. However, we will not consider the time delay correction in our study.

### 3 Symplectic ray tracing

#### 3.1 Equations of motion in discrete time

So far, we have derived the EOMs for photons in angular coordinates and conjugate momenta using the Hamiltonian formalism. We now introduce a second-order symplectic integrator to integrate these EOMs and trace the photons' trajectories. A symplectic integrator advances the equations of motion in discrete time steps, each of which preserves the mathematical structure of the Hamiltonian. As such, a symplectic integrator maintains the long-term behavior of the particle's motion. Symplectic integrators have been widely used in cosmological simulations and achieve superior accuracy compared to other integration methods of the same or even higher orders [60–62].

In this work, we utilize the kick-drift-kick (KDK) integrator, also known as the velocity Verlet method [63]. We will only sketch out the key ideas here, and leave a more detailed analysis of the integrator to appendix B. The KDK integration scheme operates by decomposing each integration step of the EOMs,  $\Delta\chi$ , into three consecutive stages, updating either the positions or the momenta at a time. Initially, we update  $\boldsymbol{\eta}$  for a half time step ( $\Delta\chi/2$ ), leaving  $\boldsymbol{\theta}$  unchanged (the first kick). Next, we update  $\boldsymbol{\theta}$  for a full time step ( $\Delta\chi$ ), while keeping  $\boldsymbol{\eta}$  unchanged (the drift stage). Finally, we apply another half time step update to  $\boldsymbol{\eta}$  to complete the integration (the second kick).

To implement this integration scheme, we configure the time steps in our simulations as follows. We perform ray tracing while evolving the N-body simulation backward in time. Light is traced from the observer's position at  $\chi = 0$  and  $\tau = \tau_0$ , towards the light source at  $\chi_s > 0$  and  $\tau = \tau_0 - \chi_s/c$ . We define a series of lens in mass shells perpendicular to the main line of sight, labeled by the subscript  $n$ , with  $n = 0$  at the observer and  $n = s$  at the source. The lens shells are determined by the time stepping of the forward simulation itself, with each shell sandwiched between the light fronts at two consecutive time steps, and the  $n$ -th shell's midpoint labeled by  $n + 1/2$ . With this set up, we can now integrate the EOMs iteratively via (see appendix B for derivation and error analysis)

$$\boldsymbol{\eta}_{n+1/2} = \boldsymbol{\eta}_n + K_n^{n+1/2}(\boldsymbol{\theta}_n, \chi_n), \quad (3.1)$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + D_n^{n+1}(\boldsymbol{\eta}_{n+1/2}, \chi_{n+1/2}), \quad (3.2)$$

$$\boldsymbol{\eta}_{n+1} = \boldsymbol{\eta}_{n+1/2} + K_{n+1/2}^{n+1}(\boldsymbol{\theta}_{n+1}, \chi_{n+1}), \quad (3.3)$$

where

$$K_a^b(\boldsymbol{\theta}_c, \chi_c) = -\frac{2}{c} \int_{\chi_a}^{\chi_b} d\chi r(\chi) \nabla_{\perp} \Phi(\mathbf{x}_c, \tau(\chi)), \quad (3.4)$$

$$D_a^b(\boldsymbol{\eta}_c, \chi_c) \simeq \left(\frac{\boldsymbol{\eta}_c}{cr_c^2}\right)(\chi_b - \chi_a), \quad (3.5)$$

are the kick and the drift factors, and  $\nabla_{\perp} \equiv r^{-1} \nabla_{\theta}$  denotes the gradient with respect to the transverse comoving coordinates. Note that the traditional velocity Verlet algorithm typically approximates the first kick operator in eq. (3.1) as  $\frac{1}{2} K_{n-1/2}^{n+1/2}(\theta_n, \chi_n)$ . The proposed iteration thus provides a better comoving space (time) resolution.

### 3.2 Time resolution correction

Eq. (3.4) updates light ray's conjugate momenta by integrating the *time-evolving*  $\nabla_{\perp} \Phi(\mathbf{x}, \tau(\chi))$  along their trajectories. In practice, we derive  $\nabla_{\perp} \Phi(\mathbf{x}, \tau(\chi))$  from the forces in N-body simulation, at discrete time steps often referred to as “snapshots”, say at  $\tau(\chi_c)$ . If each time step is small, we can approximate the growth of the matter structure using

$$\Phi(\mathbf{x}, \tau(\chi)) = \Phi(\mathbf{x}, \tau_c) \frac{D_1(\tau) a(\tau_c)}{D_1(\tau_c) a(\tau)}, \quad (3.6)$$

where  $D_1$  is the linear growth function,  $\tau = \tau(\chi)$ , and  $\tau_c = \tau(\chi_c)$ . Using this correction, eq. (3.4) becomes

$$K_a^b(\theta_c, \chi_c) = -\frac{2}{c} \int_{\chi_a}^{\chi_b} d\chi r(\chi) \nabla_{\perp} \Phi(\mathbf{x}_c, \tau) \frac{D_1(\tau) a(\tau_c)}{D_1(\tau_c) a(\tau)}. \quad (3.7)$$

## 4 Weak lensing observables

In section 3, we worked out the trajectories of the photons. We now aim to construct observable maps on the image plane using these photon trajectories. Most cosmological observables  $O$  can be written as line-of-sight integrals along the light path,

$$O(\theta) = \int_{\theta(\chi)} d\chi P(\chi) Q(\mathbf{x}, \tau(\chi)), \quad (4.1)$$

where  $P$  is the line-of-sight kernel, and  $Q$  is any cosmological field. Examples include weak lensing distortions, the galaxy density, the Sunyaev-Zel'dovich effects, the integrated Sachs-Wolfe effect, and the dispersion measure [57, 64, 65]. Here, we focus on weak lensing and develop an efficient and accurate algorithm for computing the convergence, shear, and rotation maps.

We begin with the distortion matrix  $\mathbf{A}$ , which characterizes the lensing effect of an object at source  $\chi$ ,

$$\mathbf{A}(\chi) = \frac{\partial \theta(\chi)}{\partial \theta_0}. \quad (4.2)$$

$\mathbf{A}$  can be decomposed into the product of a rotation and a shear matrix:<sup>3</sup>

$$\mathbf{A} = \begin{pmatrix} \cos \omega & \sin \omega \\ -\sin \omega & \cos \omega \end{pmatrix} \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix}. \quad (4.3)$$

The convergence  $\kappa$ , shear  $\gamma$ , and rotation  $\omega$  are the primary weak lensing observables. Since  $\omega$  is orders of magnitude smaller than  $\kappa$  and  $\gamma$ , we solve for the observables in terms of

---

<sup>3</sup>Omitting  $\chi$  for clarity.

the entries of  $\mathbf{A}$  keeping  $\omega$  to first order,

$$\begin{cases} \kappa = 1 - C_1, & \gamma_1 = C_3 + C_4 \omega, \\ \omega = \frac{C_2}{1 - \kappa}, & \gamma_2 = -C_4 + C_3 \omega, \end{cases} \quad (4.4)$$

where

$$\begin{cases} C_1 = \frac{A_{11} + A_{22}}{2}, & C_3 = \frac{A_{22} - A_{11}}{2}, \\ C_2 = \frac{A_{12} - A_{21}}{2}, & C_4 = \frac{A_{12} + A_{21}}{2}. \end{cases} \quad (4.5)$$

Finally, going back to eq. (4.1), the line-of-sight kernel  $P$  is the normalized galaxy density distribution  $n_g(\chi)$ , and  $Q$  is either  $\kappa$ ,  $\gamma$ , or  $\omega$ .

Past works have computed these observables using either the Kaiser-Squires inversion [37] or finite differences [45, 57]. However, the former is sensitive to boundary conditions and the latter can induce numerical instability. We propose a new algorithm that computes the observable maps via direct (forward-mode) differentiation of eq. (4.2), which is convenient to implement in our framework using JAX automatic differentiation.

The main idea is to decompose  $A_n$ ,<sup>4</sup> as chained products of Jacobian matrices for time steps  $n' < n$ . Differentiating the discretized EOM in eq. (3.2) with respect to the initial ray position using the chain rule,

$$\begin{aligned} \frac{\partial \boldsymbol{\theta}_{n+1}}{\partial \boldsymbol{\theta}_0} &= \frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\theta}_0} + \frac{\partial D_n^{n+1}}{\partial \boldsymbol{\eta}_{n+1/2}} \frac{\partial \boldsymbol{\eta}_{n+1/2}}{\partial \boldsymbol{\theta}_0} \\ &= \frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\theta}_0} + \frac{\partial D_n^{n+1}}{\partial \boldsymbol{\eta}_{n+1/2}} \left( \frac{\partial \boldsymbol{\eta}_n}{\partial \boldsymbol{\theta}_0} + \frac{\partial K_n^{n+1/2}}{\partial \boldsymbol{\theta}_n} \frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\theta}_0} \right) \end{aligned} \quad (4.6)$$

Using the helper function

$$\mathbf{B}(\chi) = \frac{\partial \boldsymbol{\eta}(\chi)}{\partial \boldsymbol{\theta}_0}, \quad (4.7)$$

we can then rewrite eq. (4.6) as an iteration,

$$\mathbf{B}_{n+1/2} = \mathbf{B}_n + \frac{\partial K_n^{n+1/2}}{\partial \boldsymbol{\theta}_n} \mathbf{A}_n, \quad (4.8)$$

$$\mathbf{A}_{n+1} = \mathbf{A}_n + \frac{\partial D_n^{n+1}}{\partial \boldsymbol{\eta}_{n+1/2}} \mathbf{B}_{n+1/2}, \quad (4.9)$$

$$\mathbf{B}_{n+1} = \mathbf{B}_{n+1/2} + \frac{\partial K_{n+1/2}^{n+1}}{\partial \boldsymbol{\theta}_{n+1}} \mathbf{A}_{n+1}, \quad (4.10)$$

with the initial conditions (for each photon),

$$\mathbf{A}_0 = \mathbb{I}_{2 \times 2}, \quad \mathbf{B}_0 = \mathbb{O}_{2 \times 2}. \quad (4.11)$$

Here,  $\mathbb{I}$  and  $\mathbb{O}$  are the identity and the zero matrices, respectively. We compute the iterations for  $\mathbf{A}$  and  $\mathbf{B}$  while calculating the trajectories of the photons. Each  $\mathbf{A}$  and  $\mathbf{B}$  iteration

---

<sup>4</sup>We use the subscript  $n$  to denote observables computed at  $\chi_n$ , e.g.,  $A(\chi_n) = A_n$ .



step is a Jacobian-vector product, allowing us to calculate them automatically using the forward differentiation in **JAX**, with minimal computational and memory overhead. Finally, we accumulate the observables by

$$O = \sum_n n_g(\chi_n) O_n, \quad (4.12)$$

where  $O_n \in \{\kappa_n, \gamma_n, \omega_n\}$ .

## 5 Implementation

### 5.1 Efficient dark matter and photons co-evolution

In this section, we propose an implementation of the HRT algorithm that co-evolves the dark matter particles and light rays.<sup>5</sup> This implementation only incurs minimal computational overhead compared to evolving the dark matter particles alone. The HRT algorithm is applicable to all numerical simulations of structure formation, and particularly so to differentiable ones. Specifically, we have implemented it atop the pmwd library, which offers a differentiable, fast, and memory-efficient particle mesh-based dark matter simulation [8, 9].

Let us first evolve a dark matter distribution from the initial condition to  $a = 1$  where the observer is. To start ray tracing, we first initialize the rays on a uniform grid representing the pixels (of size  $\mu_{2D}$ ; table 1 lists the mesh variables in this section) on the image plane at  $a = 1$ . To simplify our discussion, we will focus on tracing a single ray. The actual implementation parallelizes trivially across all the rays since there are no interactions between them. We use  $\mathcal{P}_P$  to denote the state (position and momentum) of this ray and  $\mathcal{P}_{DM}$  to represent the state of all the dark matter particles. We co-evolve the matter particles and the light rays backward in time as described in algorithm 1. At each time step, we compute the potential gradient  $\nabla_{\perp}\Phi$  on the 3D mesh. We then use  $\nabla_{\perp}\Phi$  to first evolve the dark matter particles (`nbody_reverse_step` in algorithm 1, [8]) backward in time and then to integrate the ray's EOMs via the KDK integrator (section 3).

The most computationally expensive operation in the simulation is to calculate  $\nabla_{\perp}\Phi$  via the Fast Fourier Transform (FFT) [8]. Algorithm 1, however, only requires  $\nabla_{\perp}\Phi$  to be calculated once per time step. The extra computation is either in 2D or on a thin shell of 3D field (e.g., the computation of the kick operator as detailed below in section 5.2), and thus is negligible compared to the 3D work load already done by the gravity solvers.

### 5.2 The kick operator

In a particle mesh (PM)-based N-body simulation, we first compute the gravitational force field on a 3D mesh and then interpolate this force onto particle positions. Similarly, we compute the force on light rays (in the kick operator,  $K_a^b(\boldsymbol{\theta}_c, \chi_c)$  in eq. (3.7), where  $\boldsymbol{\theta}_c$  is the position of the ray at the integration step) using the PM method. Our algorithm achieves this using 3 meshes/arrays: a 3D particle mesh covering the entire simulation volume with resolution  $l_{3D}$  and coordinates  $\boldsymbol{x}_{3D}$ , a 2D array of rays covering the image plane with angular

<sup>5</sup>Each ray (bundle) is defined as the collection of photons observed in the same pixel on the image plane.

---

```

procedure RAYTRACING( $\mathcal{P}_P, \mathcal{P}_{DM}, n_g$ )
   $A \leftarrow \mathbb{I}_{2 \times 2}, B \leftarrow \mathbb{O}_{2 \times 2}$ 
   $[\kappa, \gamma_1, \gamma_2, \omega] \leftarrow \mathbf{0}$ 
   $a \leftarrow 1$ 
  while  $a \geq a_{\min}$  do ▷ ray-trace backward in time to  $a_{\min}$ 
     $\nabla_{\perp} \Phi \leftarrow \mathcal{P}_{DM}$ 
     $\mathcal{P}_{DM} \leftarrow \text{nbody\_reverse\_step}(\mathcal{P}_{DM}, \nabla_{\perp} \Phi)$  ▷ Evolve dark matter backward [8]
     $\mathcal{P}_P \leftarrow \text{kick}(\mathcal{P}_P, \nabla_{\perp} \Phi)$  ▷ eq. (2.5)
     $B \leftarrow \text{iterate\_B}(\mathcal{P}_P, A, B, \nabla_{\perp} \Phi)$  ▷ eq. (4.8)
     $\mathcal{P}_P \leftarrow \text{drift}(\mathcal{P}_P)$  ▷ eq. (2.6)
     $A \leftarrow \text{iterate\_A}(\mathcal{P}_P, A, B)$  ▷ eq. (4.9)
     $\mathcal{P}_P \leftarrow \text{kick}(\mathcal{P}_P, \nabla_{\perp} \Phi)$  ▷ eq. (2.5)
     $B \leftarrow \text{iterate\_B}(\mathcal{P}_P, A, B, \nabla_{\perp} \Phi)$  ▷ eq. (4.10)
     $\kappa, \gamma_1, \gamma_2, \omega \leftarrow \text{observe}(A, n_g, \kappa, \gamma_1, \gamma_2, \omega)$  ▷ eqs. (4.4) and (4.12)
     $a \leftarrow a - \Delta a$ 
  end while
  return  $\kappa, \gamma_1, \gamma_2, \omega$ 
end procedure

```

---

**Algorithm 1.** Reverse-time co-evolution of dark matter and a single ray.

Variable	Purpose/Definition
$l_{3D}$	3D particle mesh comoving resolution
$\mathbf{x}_{3D}$	3D positions of particle mesh grid points
$\mu_{2D}$	2D ray spacing/pixel resolution on the image plane
$\boldsymbol{\theta}_0, \boldsymbol{\theta}$	2D ray positions on the image plane and during ray tracing, respectively
$\nu_{2D}$	2D ray mesh resolution during ray tracing
$\boldsymbol{\theta}_{2D}$	2D positions of ray mesh grid points during ray tracing

**Table 1.** Definitions of mesh and ray variables. The white rows define the resolution of meshes or rays, while the light gray ones define their corresponding position vectors. The top two rows are for the 3D particle mesh, on which the gravitational forces are computed. The middle two rows specify the 2D configuration of rays. And the bottom two rows describe the 2D ray mesh on which we interpolate and transfer the transverse gravitational forces, from the 3D particle mesh and to the 2D ray positions.

resolution  $\mu_{2D}$  and coordinates  $\boldsymbol{\theta}_0$ , and an intermediate 2D ray mesh transferring forces from the first mesh to the second with angular resolution  $\nu_{2D}$  and coordinates  $\boldsymbol{\theta}_{2D}$ .

We first evaluate the integrand of eq. (3.7) on the 3D mesh. For mesh points within the lens plane (having comoving coordinates  $\chi_a \leq \chi < \chi_b$ ), we calculate their projected angles, then interpolate and accumulate their values onto the ray mesh using cloud-in-cell (CIC), or trilinear, interpolation [66]. This process evaluates the kick operator line-of-sight integration on the ray mesh,  $K_a^b(\boldsymbol{\theta}_{2D}, \chi_c)$ . To account for the smoothing effect introduced by the CIC

interpolation, we deconvolve  $K_a^b(\boldsymbol{\theta}_{2D}, \chi_c)$  in Fourier space as follows,

$$\tilde{K}_a^b(\mathbf{k}, \chi_c) \rightarrow \tilde{K}_a^b(\mathbf{k}, \chi_c) \operatorname{sinc}\left(\frac{\mathbf{k}_x \nu_{2D}}{2}\right)^{-4} \operatorname{sinc}\left(\frac{\mathbf{k}_y \nu_{2D}}{2}\right)^{-4}, \quad (5.1)$$

where  $\tilde{\cdot}$  denotes the Fourier space quantity and  $\mathbf{k}$  is the wave vector. And the powers of -4 on the sinc functions accounts for both the interpolation from the 3D particle mesh to the 2D ray mesh, and that from the latter to the rays. To account for the finite width of the ray bundle, we apply Gaussian smoothing on  $K_a^b(\boldsymbol{\theta}_{2D}, \chi_c)$  at the resolution

$$\lambda_{\text{lim}} = \max\left(\frac{2 l_{3D}}{r_a + r_b}, \mu_{2D}\right), \quad (5.2)$$

by applying

$$\tilde{K}_a^b(\mathbf{k}, \chi_c) \rightarrow \tilde{K}_a^b(\mathbf{k}, \chi_c) \exp\left(\frac{1}{2}(\mathbf{k}_x^2 + \mathbf{k}_y^2)\lambda_{\text{lim}}^2\right). \quad (5.3)$$

Finally, we use CIC interpolation to map  $K_a^b(\boldsymbol{\theta}_{2D}, \chi_c)$  onto the positions of the rays, resulting in  $K_a^b(\boldsymbol{\theta}_c, \chi_c)$ .

### 5.3 Adaptive ray mesh

Unlike the 3D mesh which has a fixed comoving resolution, an angle on the 2D mesh corresponds to a comoving length that varies with the redshift. For example, at low  $z$ ,  $\nu_{2D}\chi(z) \ll l_{3D}$ . As a result, a 2D ray mesh samples forces from the 3D particle mesh at a much higher resolution the latter doesn't offer. So maintaining a high resolution 2D mesh at low redshift is both computationally inefficiency. Therefore, we employ an adaptive ray mesh, where we progressively coarsen  $\nu_{2D}$  towards low redshifts. The choice of  $\nu_{2D}$  depends on the 3D mesh resolution, pixel size, and efficient FFT sizes (including padding). Further details and convergence tests are discussed in appendix C.

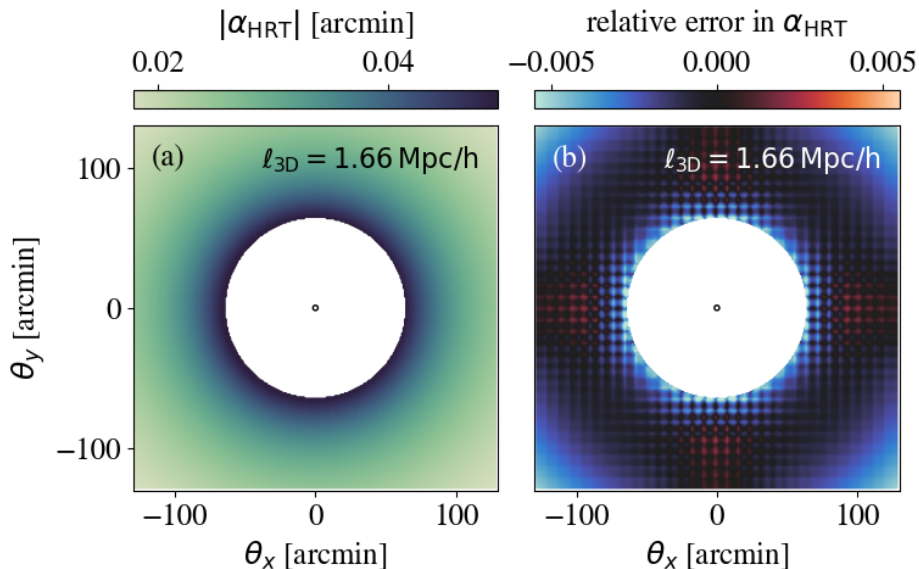
## 6 Validations

### 6.1 Lensing by a point mass

We test our ray tracing algorithm on the classic problem of gravitational lensing by a point mass in a flat, static universe. The observer is positioned at  $\chi = 0$ , with a lens mass  $M = 1.3 \times 10^{15} M_\odot$  (with an Einstein radius of  $\theta_E = 1.8'$ , representative of a massive cluster) at  $\chi_1 = 350 \text{ Mpc}/h$ , and the source plane at  $\chi_s = 850 \text{ Mpc}/h$  (or  $z_s \approx 0.3$ ). The theoretical prediction for the deflection angle  $\boldsymbol{\alpha}_{\text{theory}} \equiv \boldsymbol{\theta}_{\text{theory}} - \boldsymbol{\theta}_0$  is provided by [67]:

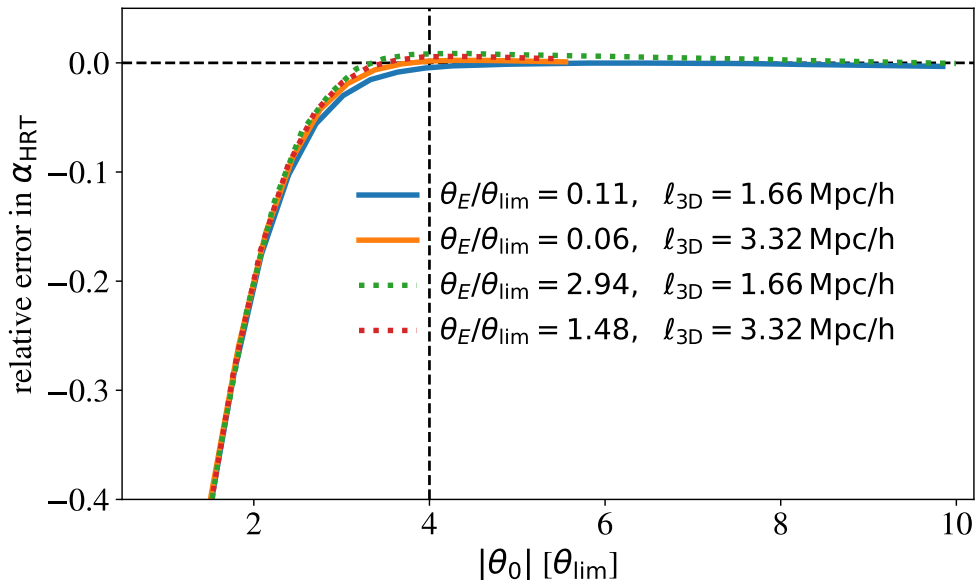
$$\boldsymbol{\alpha}_{\text{theory}} = \frac{\chi_s - \chi_l}{\chi_s} \left[ \hat{\alpha} + \frac{15\pi}{4} \hat{\alpha}^2 + \mathcal{O}(\hat{\alpha}^3) \right] \frac{\boldsymbol{\theta}_0}{|\boldsymbol{\theta}_0|}, \quad (6.1)$$

where  $\hat{\alpha} \equiv 4GM/(bc^2)$  and  $b = \chi_1 |\boldsymbol{\theta}_0|$  is the impact parameter. We solve the same problem using HRT. We define a 3D mesh of size (256, 256, 512) with a resolution of 1.66 Mpc/h, and an image plane spanning  $256' \times 256'$  with a pixel resolution of  $1'$ . We initialize rays on a uniform grid ( $\boldsymbol{\theta}_0$ ) at  $z = 0$  and trace them to  $z_s$  in 45 time steps. Panel (a) of figure 1 shows the deflection angle  $\boldsymbol{\alpha}_{\text{HRT}}$  obtained via HRT.



**Figure 1.** We apply HRT to the problem of point mass lensing as described in section 6.1, using a  $256 \times 256 \times 512$  box with a mesh resolution  $l_{3D} = 1.66 \text{ Mpc}/h$ . We show  $\alpha_{\text{HRT}}$  in panel (a). For a given  $l_{3D}$ , the point mass angular resolution limit is  $\theta_{\text{lim}} = 4l_{3D}/\chi_1$  (see figure 2). We mask light rays within this threshold where the potential field of the point mass lens is not well resolved. The black circle indicates the Einstein radius  $\theta_E$ . Panel (b) illustrates the relative error between  $\alpha_{\text{HRT}}$  and  $\alpha_{\text{theory}}$ . HRT achieves accuracy within  $< 0.5\%$  for both configurations above the resolution limit.

The HRT result aligns with theoretical predictions with very high accuracy. We plot the relative error on the image plane, defined by  $\epsilon = |\alpha_{\text{HRT}} - \alpha_{\text{theory}}|/|\alpha_{\text{theory}}|$ , in panel (b) of figure 1. HRT consistently achieves accuracy within 0.5% across the image plane where the 3D mesh resolution is adequate. Specifically, we mask pixels within  $\theta_{\text{lim}} = 4l_{3D}/\chi_1$ , where the potential field generated by the point particle cannot be clearly resolved due to the finite resolution of the 3D mesh. A lower resolution mesh dampens the potential field at the mesh scale, thereby suppressing  $\alpha_{\text{HRT}}$  near the lens mass. Figure 2 illustrates  $\epsilon$  for the  $M = 1.3 \times 10^{15} M_{\odot}$  case above as a function of  $\theta_{\text{lim}}$  for two configurations with different resolutions:  $l_{3D} = 1.66 \text{ Mpc}/h$  and  $3.32 \text{ Mpc}/h$ . These two configurations test HRT in the weak field limit, where the Einstein radius  $\theta_E \ll \theta_{\text{lim}}$ . As expected, for pixels falling within  $\theta_{\text{lim}}$ ,  $\alpha_{\text{HRT}}$  is systematically lowered (figure 2, solid blue line). Halving the 3D mesh resolution doubles  $\theta_{\text{lim}}$ , but  $\alpha_{\text{HRT}}$  still demonstrates percent-level accuracy outside  $\theta_{\text{lim}}$  and is suppressed within it (figure 2, solid orange line). We also test HRT in a stronger gravitational field by increasing  $M$  such that  $\theta_E \approx \theta_{\text{lim}}$ . The result for the high and low resolution cases are shown in dashed green and red lines in figure 2, respectively. In general, we find the definition of  $\theta_{\text{lim}}$  serves as a robust and universal threshold to characterize the accuracy of HRT regardless of  $l_{3D}$  and  $M$ . This also shows that HRT is accurate as long as the 3D mesh resolution is sufficiently high. The accuracy of HRT also weakly depends on the resolution and boundary conditions of the ray mesh, as well as the number of time steps. We characterize these effects in appendix D.



**Figure 2.** Relative error of  $\alpha_{\text{HRT}}$  versus  $|\theta_0|$  (in unit of  $\theta_{\text{lim}}$ ) for two different 3D mesh resolutions and two different lens masses. The low and high mass cases are shown in solid and dashed lines, respectively, with their masses represented in terms of their Einstein radii  $\theta_E$ . We observe a universal relationship between the accuracy and  $|\theta_0|/\theta_{\text{lim}}$ . HRT maintains sub-percent accuracy beyond  $\theta_{\text{lim}}$ , demonstrating that mesh resolution is the primary determinant of accuracy.

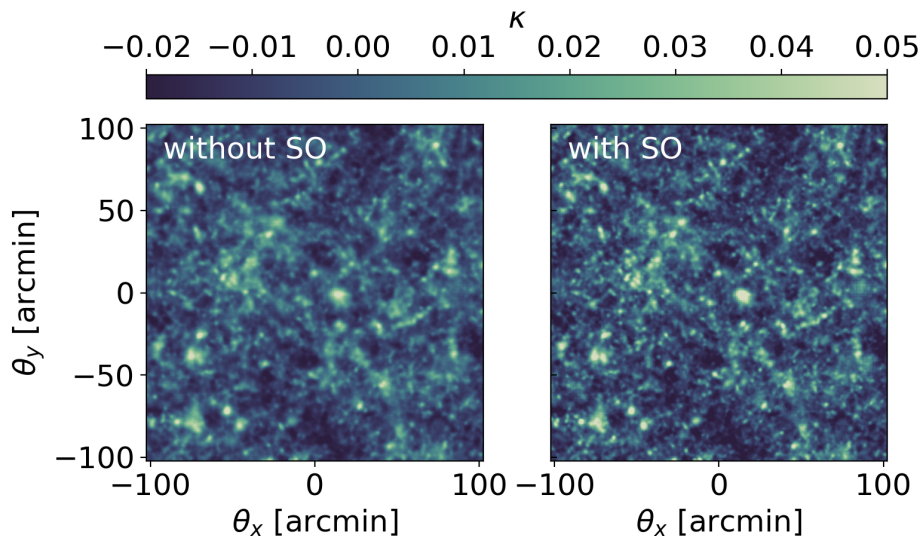
## 6.2 Post-Born weak lensing in a cosmological volume

We use the HRT algorithm to perform ray tracing over a cosmological volume and study the statistics of weak lensing observables. Throughout this paper, we adopt the *Planck* 2015 cosmology [68]. This cosmology also underpins the  $\kappa$ TNG simulations [50] and the lensing forward model by ref. [24], with which we compare our results. The  $\kappa$ TNG simulations are obtained by post-processing the higher resolution dark matter-only TNG300-1-Dark simulations (hereafter TNG-Dark) using the MLP ray tracing algorithm [50, 69–73]. Ref. [24] constructed weak lensing convergence maps with PM simulation using the Born approximation. The main purpose of this paper is to present the HRT algorithm itself; we will focus on power spectrum recovery here and leave detailed higher-order statistics analysis for a future work.

Our fiducial results are obtained using a  $512 \times 512 \times 512$  simulation box with a particle spacing of  $0.4 \text{ Mpc}/h$  and a mesh spacing of  $0.4 \text{ Mpc}/h$ . We first evolve the particles from initial condition to  $a = 1/64$  using 2nd-order Lagrangian perturbation theory. We then simulate the gravitational interaction using the PM algorithm from  $a = 1/64$  to today in 64 time steps.<sup>6</sup> From there, we perform ray tracing back to  $z_s = 1.034$  across 30 time steps. The lens distribution is the entire density field in the simulation, and the source distribution is a Dirac delta distribution at  $z_s$ . The image plane spans  $205' \times 205'$ ; it includes  $1024 \times 1024$  pixels with a  $0.2'$  pixel size. The size and resolution of our simulation box are constrained by the memory capacity of the GPU.<sup>7</sup> Consequently, we do not yet have the hardware capability to conduct ray tracing within a single, monolithic simulation box. Instead, we tile our past light

<sup>6</sup>This choice differs from ref. [24]’s, who seeds the initial condition at  $a = 1/7$ .

<sup>7</sup>For this test, we use a H100 NVL GPU on the Vera cluster at the Pittsburgh Supercomputing Center.

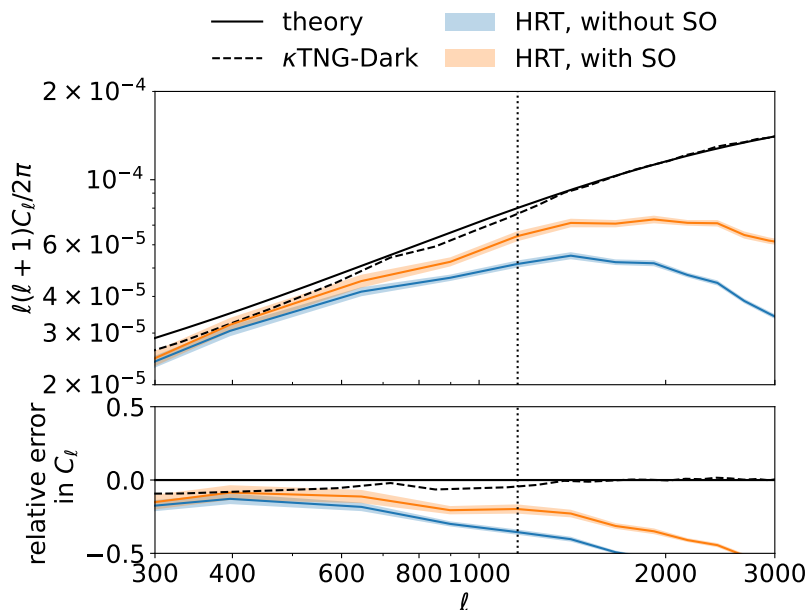


**Figure 3.** Convergence ( $\kappa$ ) maps generated using the HRT algorithm on a cosmological simulation. The left panel shows the  $\kappa$  map derived from a density field evolved using the standard PM gravity solver. The right panel illustrates the  $\kappa$  map derived from the same initial cosmological conditions but evolved using the PM algorithm enhanced with the spatial optimization (SO) gravity solver, that sharpens the convergence peaks compared to the regular PM.

cone by replicating the snapshots 12 times, each modified by a random translation and rotation along the three spatial axes. This tiling strategy, extensively studied in ref. [74], can produce  $10^4$  independent realizations of weak lensing power spectra and peak statistics with even a single snapshot. This process has been used in many weak lensing mocks including the  $\kappa$ TNG simulations [50]. An example of our ray-traced  $\kappa$  map is shown in the left panel of figure 3. We run 50 independent simulations and present the distribution of  $C_\ell^{\text{HRT}}$  along with its comparison to theory (black, solid) and  $C_\ell^{\kappa\text{TNG}}$  (black, dashed) in the top and bottom panels of figure 4 (in blue). The plot shows that  $C_\ell^{\text{HRT}}$  aligns with  $C_\ell^{\kappa\text{TNG}}$  within 20% for  $\ell < 800$  and within 30% for  $\ell < 1200$ .  $C_\ell^{\text{HRT}}$  is suppressed on the large scale because our simulation box is small and can only include a limited number of large-scale modes. However, the amount of deficit in power is consistent with the result  $C_\ell^{\kappa\text{TNG}}$  and the results in obtained by ref. [24] who performed lensing simulation using the same simulation volume. We also observe  $C_\ell^{\text{HRT}}$  to be suppressed at high  $\ell$ 's. This is because the PM gravity solver cannot accurately resolve gravitational interactions at the mesh scale. This result is comparable to the ‘‘DLL without correction’’ analysis in ref. [24], who recovers lensing  $C_\ell$  up to  $\ell \approx 300$  compared to  $C_\ell^{\kappa\text{TNG}}$ . We recover  $C_\ell$  over a larger  $\ell$  range because of the finer mesh size used in the present work.

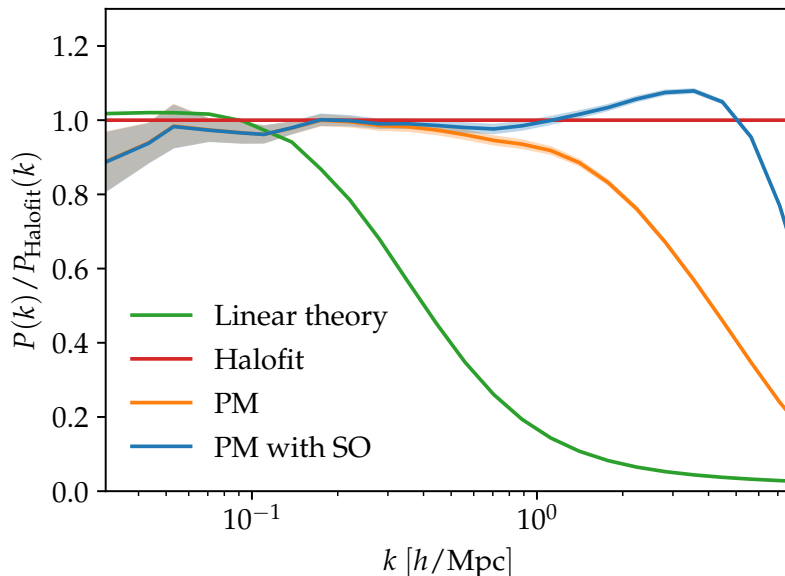
Next, we explore the potential of recovering the convergence power spectrum at higher  $\ell$  ranges by extending the regular PM gravity solver to smaller scales with a spatial optimization (SO) algorithm that sharpens the PM forces near the mesh scale [75]. In Fourier space, the gravitational force is proportional to  $(\mathbf{k}/k^2)\delta(\mathbf{k})$ . SO modifies the Fourier force kernel by a nonlinear function  $g$ :

$$\frac{\mathbf{k}}{k^2} \rightarrow \frac{\mathbf{k}}{k^2} g(\mathbf{k}; \vartheta), \quad (6.2)$$



**Figure 4.** The top panel shows the distribution of the convergence power spectra obtained by ray tracing through cosmological simulations using HRT. The bottom panel shows the relative error compared to theory (black, solid) for the same cosmology and  $n(z)$ . The  $\kappa$ TNG results are shown in black, dashed lines for comparison. Simulations where gravity is solved using the standard PM method are depicted in blue, while those using the PM enhanced with the spatial optimization (SO) force-sharpening method are shown in orange. The HRT and the  $\kappa$ TNG  $C_\ell$ 's are similarly suppressed at low  $l$ 's because of the finite simulation box size and the lack of large-scale modes. The HRT results align with  $\kappa$ TNG at large scales but are suppressed at smaller scales since PM methods cannot resolve interactions at or below the mesh resolution. Enhancing PM with SO mitigates this issue and extends  $C_\ell^{\text{HRT}}$  to higher  $l$ 's. For the PM+SO result (where the redshift distribution is a delta function at the same redshift,  $z = 1.034$ , as the  $\kappa$ TNG lensing map),  $C_\ell^{\text{HRT}}$  agrees with  $C_\ell^{\kappa\text{TNG}}$  to within 10% for  $l < 800$  and 15% for  $l < 1200$ , where  $l = 1200$  is marked by the vertical dotted line.

where  $\vartheta$  includes cosmological parameters and simulation configurations. We implement  $g$  using symmetry-preserving neural networks, which are trained to align the PM+SO simulations with the GADGET-4 [62] simulations across a wide range of cosmologies and simulation configurations. As illustrated in figure 5, SO boosts the small-scale  $P(k)$  compared to the regular PM algorithm for  $k > 1h/\text{Mpc}$ , though still slightly lower than the halofit [76] predictions. We run HRT simulations under the same settings as previously discussed but integrate SO with the same initial conditions. An example  $\kappa$  map, shown in the right panel of figure 3, indicates that SO indeed sharpens the small-scale fluctuations compared to the regular PM simulations. The distribution of power spectra with SO is displayed in figure 4 in orange. SO consistently improves the convergence power spectrum at all scales, but is especially helpful at high  $l$ 's.  $C_\ell^{\text{HRT}}$  agrees with  $C_\ell^{\kappa\text{TNG}}$  to within 10% for  $l < 800$  and 15% for  $l < 1200$ . In general, the maximum  $l$  at which we can recover  $C_\ell$  depends on the 3D mesh resolution  $l_{3\text{D}}$  and the shape of the lensing kernel. Empirically, we find this scale to be approximately  $l \approx \frac{\ell_{\text{lim}}}{16} \approx 1200$  for  $z_s = 1.034$  when SO is applied (vertical dotted line



**Figure 5.** Matter power spectrum at  $z = 0$  with the regular PM solver (orange) and with PM + spatial optimization (SO, blue). They are compared to the prediction of the Halofit model in red and the linear theory in green. The shaded regions represent the standard error of the mean estimated with 50 independent simulations. The simulation has  $512^3$  particles with a  $0.4 \text{ Mpc}/h$  particle spacing. SO corrects for the small-scale interactions between dark matter particles that are ignored by the regular PM, boosting power at high- $k$ .

in figure 4). We define  $\ell_{\text{lim}} = \pi / \left( \frac{\chi(z_s)}{2l_{3\text{D}}} \right)$ , since  $\frac{\chi(z_s)}{2l_{3\text{D}}}$  is the angular scale of the 3D mesh at the radial comoving distance where the lensing kernel has the most sensitivity. More generally, when the source distribution is not a delta function, the denominator of  $\ell_{\text{lim}}$  should be similarly approximated by the distance to the peak of the lensing kernel.

We anticipate that higher mesh resolution and stronger force-sharpening effects will enable  $C_\ell^{\text{HRT}}$  to reach even smaller scales, which is potentially achievable when larger GPU memory becomes accessible in the future. We also conduct convergence tests with varying HRT hyper-parameters in appendix E.

## 7 Conclusions

We have presented a post-Born, three-dimensional, on-the-fly ray tracing algorithm based on the Hamiltonian dynamics of light rays. This method, termed Hamiltonian ray tracing (HRT), includes the lens-lens coupling and does not assume the Born approximation. Additionally, HRT deflects photons based on the gravitational potential generated by the entire cosmological volume, not just that within a single lens plane. HRT also performs ray tracing on-the-fly. As a result, it integrates well with any gravity solver and computes light ray trajectories and lensing observable maps with minimal computational overhead compared to running the gravity solver alone. We implemented HRT using the pmwd library on a GPU platform, demonstrating its accuracy and limitations both for point mass lensing and in generating convergence maps and power spectra for dark matter simulations. For point-mass lensing,



HRT yields deflection angles accurate to sub-percent levels above the resolution limit. When applied to cosmological simulations using the standard PM gravity solver with a  $0.4 \text{ Mpc}/h$  particle spacing, HRT generates lensing convergence maps whose power spectrum aligns with the fiducial  $\kappa^{\text{TNG}}$  results to within 20% of  $C_\ell^{\kappa^{\text{TNG}}}$  for  $\ell < 800$  and 30% for  $\ell < 1200$ . Extending the PM with small-scale force-sharpening method (SO) enables recovery of  $C_\ell$  to within 10% of  $C_\ell^{\kappa^{\text{TNG}}}$  for  $\ell < 800$  and 15% for  $\ell < 1200$ . Future works could extend the tests to other scenarios, such as lensing by compensated black holes in an evolving cosmological setting [77] or different weak lensing statistics paired with various redshift distributions.

While HRT should work with any simulation of structure formation, its implementation is particularly easy in frameworks with automatic differentiation capability, such as pmwd which is based on JAX. The forward mode in automatic differentiation helps to co-evolve the lensing observables, such as cosmic shear, with the light ray deflections, since the Jacobian of the latter involves the former. On the other hand, we would also need the reverse-mode differentiation to compute the likelihood gradient, for example, in FLI applications that involve Hamiltonian Monte Carlo. Automatic reverse-mode differentiation through the whole simulation can be extremely memory consuming, and the adjoint method [8] has been introduced to obtain memory-efficient gradients. The same method can be applied to HRT, which we leave for future development.

The accuracy of HRT is primarily limited by the mesh resolution and the precision of the gravity solver at small comoving scales. If we assume that the smallest scale at which the gravity solver is accurate is  $k_{\text{max}} \propto l_{3\text{D}}^{-1}$ , and that the smallest angular scale at which HRT is accurate is  $\ell_{\text{max}} \propto l_{3\text{D}}^{-1}$ , then  $\ell_{\text{max}} \propto k_{\text{max}}$ . For the current generation of weak lensing surveys like HSC,  $\ell_{\text{max}} \approx 1800$  [78], about 2 times higher than the  $\ell$  limit we achieve here. To bridge the gap between simulation and observational data, future work could extend the SO framework to push  $k_{\text{max}}$  above  $5 h/\text{Mpc}$  where baryonic effects also becomes important. Alternatively, increasing the resolution of the cosmological simulation by two or three-folds, which could be achieved by parallelizing the PM code and the HRT algorithm across multiple GPU devices or nodes, may also prove equally effective.

The HRT algorithm empowers cosmological analysis in several ways. HRT can quickly generate cosmology-dependent ray tracing shear maps, making it suitable for training machine learning models for simulation-based inference. It can help establishing the connection between cosmology and higher-order statistics in a data-driven manner and aid in estimating cosmology-dependent covariance matrices for summary statistics analyses. As a differentiable ray tracing algorithm, HRT also enables field-level inference that accounts for post-Born effects. While this work focuses on the algorithm and its implementation, future studies could extend this discussion with a detailed analysis of higher-order statistics in the convergence maps simulated by HRT. Additionally, optimizing the differentiation of HRT through the adjoint method could make it more memory-friendly in field-level inference applications.

**Code availability.** pmwd is open-source on GitHub (<https://github.com/eelregit/pmwd>). The ray tracing feature will be made available in that repository after code cleaning, including the source files and scripts of this paper (<https://github.com/eelregit/pmwd/tree/master/docs/papers/hrt>).

## Acknowledgments

AZ thanks Yuuki Omori for discussion on post-born validations on weak lensing. YL thanks Sukhdeep Singh for helpful discussion. AZ, XL, and RM were partially supported by a grant from the Simons Foundation (Simons Investigator in Astrophysics, Award ID 620789). YL and YZ are supported by The Major Key Project of PCL. YZ is further supported by the China Postdoctoral Science Foundation under award number 2023M731831. GF acknowledges the support of the European Research Council under the Marie Skłodowska Curie actions through the Individual Global Fellowship No. 892401 PiCOGAMBAS and of the Simons Foundation for the initial stages of this work. This work is supported by the Bridges-2 supercomputer at the Pittsburgh Supercomputing Center under the NSF ACCESS Explore allocation PHY230147. We thank the Columbia Lensing group (<http://columbialensing.org>) for making their simulations available. The creation of these simulations is supported through grants NSF AST-1210877, NSF AST-140041, and NASA ATP-80NSSC18K1093.

## A Hamiltonian dynamics of photons lensed by weak gravitational field

In this section, we derive the EOMs in eqs. (2.5)–(2.7) from the Hamiltonian principle. Although the procedure is similar to that of ref. [59], we employ a different metric and present the derivations in full detail. These details, not included in ref. [59], may prove helpful to some readers.

The general relativistic covariant Hamiltonian for a photon is given by eq. (2.4) and copied here:

$$H(x^j, p_i, \tau) = c\sqrt{h^{ij}p_i p_j} \left(1 + 2\frac{\Phi}{c^2}\right). \quad (\text{A.1})$$

The photon's EOMs are given by Hamilton's equations

$$\frac{dx^i}{d\tau} = \{x^i, H\} = \frac{\partial H}{\partial p_i}, \quad (\text{A.2})$$

$$\frac{dp_i}{d\tau} = \{p_i, H\} = -\frac{\partial H}{\partial x^i}, \quad (\text{A.3})$$

which we will solve explicitly. The time derivatives of position and momentum are

$$\frac{1}{c} \frac{dx^j}{d\tau} = \frac{\partial \sqrt{h^{kl}p_k p_l}}{\partial p_j} \left(1 + 2\frac{\Phi}{c^2}\right) = \frac{h^{kl}}{2p} \left(\frac{\partial p_k}{\partial p_j} p_l + \frac{\partial p_l}{\partial p_j} p_k\right) \left(1 + 2\frac{\Phi}{c^2}\right) = n^j \left(1 + 2\frac{\Phi}{c^2}\right). \quad (\text{A.4})$$

$$\frac{1}{c} \frac{dp_j}{d\tau} = -2p \frac{\partial_j \Phi}{c^2} - \frac{p_k p_l \partial_j h^{kl}}{2p} \left(1 + 2\frac{\Phi}{c^2}\right) = -p \left[2 \frac{\partial_j \Phi}{c^2} + \frac{n_k n_l \partial_j h^{kl}}{2} \left(1 + 2\frac{\Phi}{c^2}\right)\right], \quad (\text{A.5})$$

where the unit momentum vector  $n^i$  is

$$n^i \equiv \frac{p^i}{p} = \frac{p^i}{\sqrt{h^{ij}p_i p_j}}. \quad (\text{A.6})$$

$n^i$  is the most important dynamical variable in ray-tracing. To explicitly derive its EOM, we express its time dependence in those of the two independent variables  $p_i$  and  $x^j$ , and consider the following expansion

$$\frac{1}{c} \frac{dn^i}{d\tau} = \frac{1}{c} \frac{d}{d\tau} \left(\frac{p^i}{p}\right) = \frac{\partial(p^i/p)}{\partial p_j} \frac{1}{c} \frac{dp_j}{d\tau} + \frac{\partial(p^i/p)}{\partial x^j} \frac{1}{c} \frac{dx^j}{d\tau}. \quad (\text{A.7})$$

The first term becomes

$$\begin{aligned} \frac{\partial(p^i/p)}{\partial p_j} \frac{1}{c} \frac{dp_j}{d\tau} &= \frac{1}{c} \frac{dp_j}{d\tau} \left( \frac{h^{ik}}{p} \frac{\partial p_k}{\partial p_j} - \frac{p^i}{p^2} \frac{\partial p}{\partial p_j} \right) = \frac{1}{c} \frac{dp_j}{d\tau} \frac{h^{ij} - n^i n^j}{p} \\ &= \left[ -2 \frac{\partial_j \Phi}{c^2} - \frac{n_k n_l \partial_j h^{kl}}{2} \left( 1 + 2 \frac{\Phi}{c^2} \right) \right] (h^{ij} - n^i n^j). \end{aligned} \quad (\text{A.8})$$

Meanwhile, the second term becomes

$$\begin{aligned} \frac{\partial(p^i/p)}{\partial x^j} \frac{1}{c} \frac{dx^j}{d\tau} &= \frac{1}{c} \frac{dx^j}{d\tau} \left( \frac{p_k}{p} \partial_j h^{ik} - \frac{p^i}{p^2} \partial_j p \right) = \frac{1}{c} \frac{dx^j}{d\tau} \left( n_k \partial_j h^{ik} - \frac{1}{2} n^i n_k n_l \partial_j h^{kl} \right) \\ &= \left( n^j n_k \partial_j h^{ik} - \frac{1}{2} n^i n^j n_k n_l \partial_j h^{kl} \right) \left( 1 + 2 \frac{\Phi}{c^2} \right). \end{aligned} \quad (\text{A.9})$$

Putting the two terms together,

$$\begin{aligned} \frac{1}{c} \frac{dn^i}{d\tau} &= -2 \frac{\partial_j \Phi}{c^2} (h^{ij} - n^i n^j) - \frac{n_k n_l \partial_j h^{kl}}{2} \left( 1 + 2 \frac{\Phi}{c^2} \right) (h^{ij} - n^i n^j) \\ &\quad + \left( n^j n_k \partial_j h^{ik} - \frac{1}{2} n^i n^j n_k n_l \partial_j h^{kl} \right) \left( 1 + 2 \frac{\Phi}{c^2} \right) \\ &= -2 \frac{\partial_j \Phi}{c^2} (h^{ij} - n^i n^j) - \frac{h^{ij} n_k n_l \partial_j h^{kl}}{2} \left( 1 + 2 \frac{\Phi}{c^2} \right) + n^j n_k \partial_j h^{ik} \left( 1 + 2 \frac{\Phi}{c^2} \right). \end{aligned} \quad (\text{A.10})$$

It turns out that we can simplify the last two terms, because

$$\begin{aligned} -\frac{1}{2} h^{ij} n_k n_l \partial_j h^{kl} + n^j n_k \partial_j h^{ik} &= \frac{1}{2} h^{ij} n^k n^l \partial_j h_{kl} - h^{il} n^j n^k \partial_j h_{kl} \\ &= \frac{1}{2} h^{il} n^j n^k (\partial_l h_{jk} - 2 \partial_j h_{kl}) \\ &= -\frac{1}{2} h^{il} n^{(j} n^{k)} (\partial_j h_{kl} + \partial_k h_{jl} - \partial_l h_{jk}) \\ &= -\Gamma_{jk}^i n^j n^k, \end{aligned} \quad (\text{A.11})$$

where we have used  $\partial_j h^{ik} = -h^{il} h^{km} \partial_j h_{lm}$  in the first equality, swapped  $j$  and  $l$  in the (first term of the) second one, symmetrized  $j$  and  $k$  in the third, and reduced the metric derivatives to the Christoffel symbol  $\Gamma_{jk}^i$  at last. With the result above, we can simplify  $dn^i/d\tau$  as follows

$$\frac{1}{c} \frac{dn^i}{d\tau} = -2 \frac{\partial_j \Phi}{c^2} (h^{ij} - n^i n^j) - \Gamma_{jk}^i n^j n^k \left( 1 + 2 \frac{\Phi}{c^2} \right), \quad (\text{A.12})$$

which agrees with the result in [59].

We are now ready to derive the EOMs of a photon. Until this point, we have not enforced any specific metric on the EOMs. We now select the metric in eq. (2.3) with coordinates  $\mathbf{x} = (\boldsymbol{\theta}, \chi)$ . Here, we assume small angle approximations such that  $\sin^2(\theta) \approx 1$ . The unit momentum vector in this coordinate system is then

$$\mathbf{n} = \left[ \frac{\mathbf{v}}{cr}, -1 + \mathcal{O}\left(\frac{\mathbf{v}^2}{c^2}\right) \right], \quad (\text{A.13})$$

where the first term can be viewed as the normalized peculiar angular velocity on the sky, with  $\mathbf{v}$  being the 2D transverse peculiar velocity. We shall work in the limits of weak fields

and small angles, which corresponds to first order in  $\Phi/c^2$ ,  $\nabla\Phi/c^2$ , and  $\mathbf{v}/c$ . The nonzero Christoffel symbols are

$$\Gamma_{13}^1 = \Gamma_{31}^1 = \Gamma_{23}^2 = \Gamma_{32}^2 = \frac{r'}{r}, \quad \Gamma_{11}^3 = \Gamma_{22}^3 = -rr', \quad (\text{A.14})$$

where  $'$  denotes derivative with respect to  $\chi$ . We have omitted corrections of  $\mathcal{O}(\nabla\Phi/c^2)$  because the Christoffel symbol only appears in the expression  $\Gamma_{jk}^i n^j n^k (1 + 2\Phi/c^2)$ , and always pairs with at least one  $\mathcal{O}(v/c)$  term. Taking time derivatives of  $n^i$  for  $i = 1, 2$ , we obtain

$$\frac{1}{c} \frac{dn^i}{d\tau} = \frac{1}{c^2 r} \frac{dv^i}{d\tau} - \frac{v^i}{c^2 r^2} \frac{dr}{d\tau} \quad (\text{A.15})$$

Consider  $i = 1$  of the  $n^i$  EOM:

$$\begin{aligned} \frac{1}{c} \frac{dn^1}{d\tau} &= \frac{1}{c^2} (-2h^{11} \partial_1 \Phi + 2n^1 n^1 \partial_1 \Phi + 2n^1 n^2 \partial_2 \Phi + 2n^1 n^3 \partial_3 \Phi) - 2\Gamma_{13}^1 n^1 n^3 \left(1 + 2\frac{\Phi}{c^2}\right) \\ &\simeq -2 \frac{\partial_1 \Phi}{c^2 r^2} + 2 \frac{v^1 r'}{c r^2}. \end{aligned} \quad (\text{A.16})$$

The general case follows similarly by symmetry:

$$\frac{1}{c} \frac{dn^i}{d\tau} = -2 \frac{\partial_i \Phi}{c^2 r^2} + 2 \frac{v^i r'}{c r^2}, \quad (\text{A.17})$$

where we ignore the potential and its gradient terms since they are higher order corrections (while more generally  $h^{ij} - n^i n^j$  projects in the transverse direction of  $n^i$ ). Comparing this with the above equation, and approximating  $cd\tau \approx -d\chi$  by ignoring time delay correction of  $\mathcal{O}(\Phi/c^2)$  and  $\mathcal{O}(v^2/c^2)$ , we derive

$$\frac{d\eta^i}{d\chi} \equiv \frac{d(rv^i/c)}{d\chi} = \frac{rv^{i'} + r'v^i}{c} = 2 \frac{\partial_i \Phi}{c^2} = 2 \frac{1}{c^2} \frac{\partial \Phi}{\partial \theta^i}. \quad (\text{A.18})$$

## B Symplectic integrator

We have purposely written the EOMs in eqs. (2.5) and (2.6) using  $\boldsymbol{\theta}$  and  $\boldsymbol{\eta}$ , in unit of angle and length, respectively. Either from the Hamiltonian eq. (2.4) or from the EOMs, we see that the system, under the assumptions above, admits a separable Hamiltonian

$$\tilde{H}(\boldsymbol{\theta}, \boldsymbol{\eta}, \chi) = \underbrace{\frac{\boldsymbol{\eta}^2}{2r^2}}_T + 2 \underbrace{\frac{\Phi(\boldsymbol{\theta}, \chi)}{c^2}}_V, \quad (\text{B.1})$$

where  $\boldsymbol{\theta}$  and  $\boldsymbol{\eta}$  are the canonical coordinate and momentum,  $\chi$  serves the function of time, and  $T$  and  $V$  represent kinetic and potential energy. In fact,  $\tilde{H}$  also follows from the Fermat principle, and can be derived with Legendre transformation from the Lagrangian in the Fermat action (see [59, 79, 80], especially for the continuous version in the former). This allows us to decouple the effect of the kinetic and the potential energy and rewrite the EOMs in eqs. (2.5) and (2.6) as

$$\frac{d\boldsymbol{\theta}}{d\chi} = \{\boldsymbol{\theta}, T\}, \quad \frac{d\boldsymbol{\eta}}{d\chi} = \{\boldsymbol{\eta}, V\}, \quad (\text{B.2})$$

where  $\{\}$  is the Poisson bracket.

Formally, we can integrate<sup>8</sup> eq. (B.2) over the interval  $[\chi, \chi + \Delta\chi]$  by

$$\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\eta} \end{bmatrix} (\chi + \Delta\chi) = e^{(T+V)\Delta\chi} \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\eta} \end{bmatrix} (\chi). \quad (\text{B.3})$$

Since the EOMs of  $\boldsymbol{\theta}$  and  $\boldsymbol{\eta}$  are coupled together, we cannot solve them individually. The KDK integrator utilizes the Baker-Campbell-Hausdorff identity to approximate this evolution operator as the product

$$e^{(H+\tilde{H}_{\text{err}})\Delta\chi} = e^{V\frac{\Delta\chi}{2}} e^{T\Delta\chi} e^{V\frac{\Delta\chi}{2}} \quad (\text{B.4})$$

where  $\tilde{H}_{\text{err}}$  denotes the approximation error, which is second order in  $\Delta\chi$  [63]. Eq. (B.4) shows that we can decompose each  $\Delta\chi$  integration at each time step into three consecutive steps, each updating only either the position or the momentum variables.  $e^{V\Delta\chi}$  is termed a kick operator since it updates  $\boldsymbol{\eta}$  and leaves  $\boldsymbol{\theta}$  unchanged, while  $e^{T\Delta\chi}$  is called a drift operator since it updates  $\boldsymbol{\theta}$  and leaves  $\boldsymbol{\eta}$  unchanged.

### C Adaptive ray mesh

The 2D array of light rays is characterized by its pixel size  $\mu_{2\text{D}}$ , i.e. their spacing at  $z = 0$ , and the number of pixels  $M_{x,y}$ . In order to interpolate and transfer the PM forces from the 3D mesh to the rays, we need another intermediate 2D angular mesh of resolution  $\nu_{2\text{D}}$  and size  $N_{x,y}$ . We call it the ray mesh (and likewise the 3D mesh particle mesh), as explained in table 1.

Generally,  $\nu_{2\text{D}} \leq \mu_{2\text{D}}$  and  $N_{x,y} > M_{x,y}$ . Our adaptive ray mesh needs to address two problems: 1) we need to vary the 2D mesh resolution  $\nu_{2\text{D}}$  as a function of  $\chi$  (section 5.3) and 2) we need to add padding to the ray mesh. Padding is crucial for accurate smoothing and deconvolution in the computation of the kick operator, as it alleviates the effects of periodic boundary assumptions inherent to FFT.

A 3D mesh cell at  $\chi$  corresponds to an angular size  $l_{3\text{D}}/r(\chi)$ . Therefore, for a lens plane that spans  $[\chi_a, \chi_b]$ , we define the angular resolution limit  $\lambda_{\text{lim}}$  as in eq. (5.2):

$$\lambda_{\text{lim}} = \max\left(\frac{2l_{3\text{D}}}{r_a + r_b}, \mu_{2\text{D}}\right), \quad (\text{C.1})$$

in which the former dominates near the observer as limited by the PM force resolution, and the latter takes effect at early times. Let us fix the ray mesh spacing as

$$\nu_{2\text{D}} = \iota \lambda_{\text{lim}}, \quad (\text{C.2})$$

where  $0 < \iota \leq 1$  is an accuracy parameter. The smaller  $\iota$  is, the finer the ray mesh. Take the x-direction for example, we want

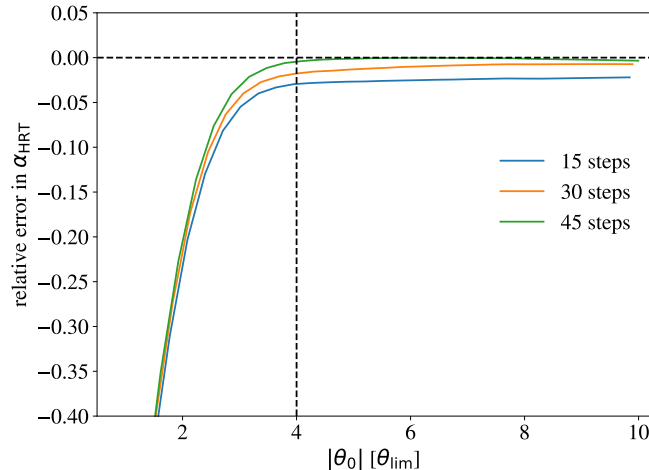
$$N_x \nu_{2\text{D}} \geq M_x \mu_{2\text{D}} + p_{\text{min}} \lambda_{\text{lim}}, \quad (\text{C.3})$$

where  $p_{\text{min}}$  is the minimum padding width in unit of  $\lambda_{\text{lim}}$ . The above conditions reduce to

$$N_x = \left\lceil M_x \frac{\mu_{2\text{D}}}{\nu_{2\text{D}}} + \frac{p_{\text{min}}}{\iota} \right\rceil_{\text{FFT}}, \quad (\text{C.4})$$

where we round up the mesh size to an integer that is efficient for FFT, e.g. powers of 2, while also saving the number of compilations that can affect JAX performance. The y-dimension follows accordingly.

<sup>8</sup>Here, for an operator  $A$  and a vector  $\boldsymbol{v}$ , we use the shorthand  $A\boldsymbol{v} = \{\boldsymbol{v}, A\}$ .



**Figure 6.** Relative error of  $\alpha_{\text{HRT}}$  versus  $\theta_{\text{lim}}$  as a function of the number of HRT integration steps used for a point mass lensing task. This figure is to be compared to the blue line in figure 2 which has the same experimental setup with 45 integration steps.

### D Other considerations and convergence tests for point mass lensing

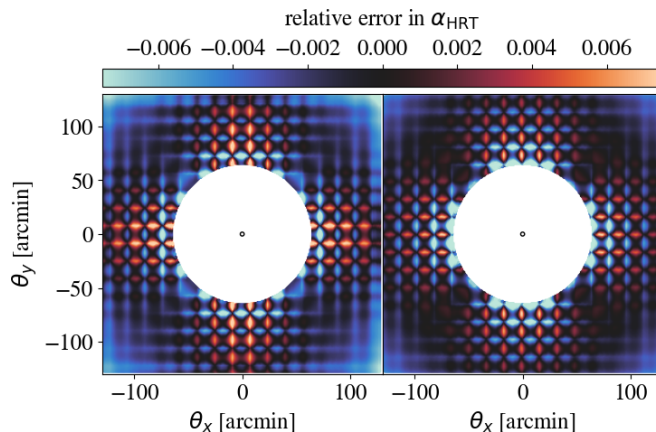
In the context of point mass lensing, we explore how the accuracy of HRT is influenced by the boundary conditions of the 3D mesh, the number of integration steps, the 2D ray mesh accuracy parameter  $\iota$ , and the 2D ray mesh padding size  $p_{\text{min}}$  (appendix C).

First, when we perform ray-tracing on a point mass lens, we compute the gravitational potential field using FFT with periodic boundary conditions. This is an approximation of the gravity solver and not the ray-tracing algorithm itself. To disentangle the error induced by the gravity solver from those due to HRT, we have compared  $\alpha_{\text{HRT}}$  against a theoretical result that incorporates periodic boundary conditions throughout this work. To model the effect of the periodic boundary condition, we assume there is not only one but also infinitely many periodic images of the point mass lens, at comoving positions  $\mathbf{m} \in \{(0, 0), (\pm L_x, 0), (0, \pm L_y), \dots\}$ , where  $L_{x,y}$  are the side lengths of the simulation box. The leading order deflection in eq. (6.1) is replaced by its periodic summation:

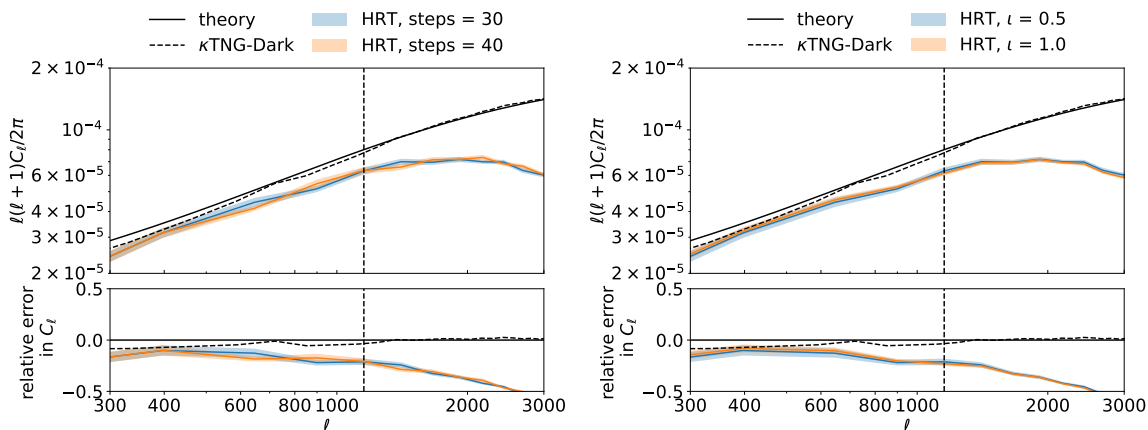
$$\hat{\alpha} \frac{\boldsymbol{\theta}_0}{|\boldsymbol{\theta}_0|} = \frac{4GM}{c^2} \frac{\mathbf{b}}{b^2} \longrightarrow \frac{4GM}{c^2} \sum_{\mathbf{m}} \frac{\mathbf{b} - \mathbf{m}}{|\mathbf{b} - \mathbf{m}|^2}, \quad (\text{D.1})$$

where  $\mathbf{b} = (\theta_{0x}\chi_1, \theta_{0y}\chi_1)$  is the impact parameter of the photons. Eq. (D.1) reduces to eq. (6.1) when only  $\mathbf{m} = (0, 0)$  is considered. Throughout this work, we account for the nearest  $32^2$  periodic images of the lens mass when computing  $\alpha_{\text{theory}}$ . The theoretical result converges within numerical precision.

The numerical experiments in section 6.1 use 45 integration steps and assume a padding of  $\iota = 0.5$  and  $p_{\text{min}} = 256$ . We utilize the same setup as in section 6.1 to evaluate how the accuracy of HRT depends on these hyper-parameters. Figure 6 illustrates that  $\alpha_{\text{HRT}}$  converges to the theoretical values as the number of integration steps increases. With too few integration steps,  $\alpha_{\text{HRT}}$  is typically lower than the truth. The two panels in figure 7 demonstrate the effect of the ray mesh hyper-parameters: the accuracy parameter  $\iota$  and the padding size  $p_{\text{min}}$ . The left panel uses ( $\iota = 1, p_{\text{min}} = 4$ ) and the right panel uses ( $\iota = 1, p_{\text{min}} = 256$ ). These are compared to panel (b) of figure 1, which has the same



**Figure 7.** The effect of ray mesh hyper-parameters: the accuracy parameter  $\iota$  and the padding size  $p_{\min}$ . The two panels show  $\alpha_{\text{HRT}}$ 's residual errors on the image plane for the point mass test. The left panel uses  $(\iota = 1, p_{\min} = 4)$  and the right panel uses  $(\iota = 1, p_{\min} = 256)$ . They are compared to panel (b) of figure 1 which uses  $(\iota = 0.5, p_{\min} = 256)$ .



**Figure 8.** Relative error of the SO-enhanced  $C_\ell$  as a function of the number of HRT integration steps (left) and of the HRT ray mesh accuracy parameter  $\iota$  (right). The shaded regions show the standard error of the mean over 25 independent simulations. Similar to figure 4, the black solid line shows the theory prediction, the dashed line shows the  $\kappa\text{TNG}$  simulation, and the vertical dotted line shows the  $\ell = 1200$  scale cut.

experimental setup but with  $(\iota = 0.5, p_{\min} = 256)$ . In both cases, HRT achieves sub-percent accuracy, showing that increased padding improves  $\alpha_{\text{HRT}}$  accuracy near the boundary and higher  $\iota$  results in more prominent discretization features of the particle and ray meshes. The result also shows that, in practice, we do not need to carry a  $p_{\min}$  that is as large.

## E Convergence tests for weak lensing power spectrum

We compute the convergence power spectrum by ray tracing to a source plane at  $z_s = 1.034$ . We tile the past light cone with simulation boxes of a  $512^3$  mesh and  $512^3$  particles with a particle/mesh spacing of  $0.4 \text{ Mpc}/h$ , similar to section 6.2. We vary the number of ray tracing time steps and the ray mesh accuracy parameter  $\iota$ , as shown in figure 8, and find that the power spectrum is not sensitive to these hyper-parameters.

## References

- [1] A. Kiessling, A.F. Heavens and A.N. Taylor, *SUNGLASS: A new weak lensing simulation pipeline*, *Mon. Not. Roy. Astron. Soc.* **414** (2011) 2235 [[arXiv:1011.1476](#)] [[INSPIRE](#)].
- [2] P. Fosalba, E. Gaztañaga, F.J. Castander and M. Crocce, *The MICE Grand Challenge light-cone simulation — III. Galaxy lensing mocks from all-sky lensing maps*, *Mon. Not. Roy. Astron. Soc.* **447** (2015) 1319 [[arXiv:1312.2947](#)] [[INSPIRE](#)].
- [3] R. Sgier, A. Réfrégier, A. Amara and A. Nicola, *Fast generation of covariance matrices for weak lensing*, *JCAP* **01** (2019) 044 [[arXiv:1801.05745](#)] [[INSPIRE](#)].
- [4] N. Tessore et al., *GLASS: Generator for Large Scale Structure*, *Open J. Astrophys.* **6** (2023) [[arXiv:2302.01942](#)] [[INSPIRE](#)].
- [5] Z. Li, J. Liu, J.M.Z. Matilla and W.R. Coulton, *Constraining neutrino mass with tomographic weak lensing peak counts*, *Phys. Rev. D* **99** (2019) 063527 [[arXiv:1810.01781](#)] [[INSPIRE](#)].
- [6] J. Liu et al., *Cosmology constraints from the weak lensing peak counts and the power spectrum in CFHTLenS data*, *Phys. Rev. D* **91** (2015) 063507 [[arXiv:1412.0757](#)] [[INSPIRE](#)].
- [7] K. Cranmer, J. Brehmer and G. Louppe, *The frontier of simulation-based inference*, *Proc. Nat. Acad. Sci.* **117** (2020) 30055 [[arXiv:1911.01429](#)] [[INSPIRE](#)].
- [8] Y. Li et al., *Differentiable Cosmological Simulation with the Adjoint Method*, *Astrophys. J. Suppl.* **270** (2024) 36 [[arXiv:2211.09815](#)] [[INSPIRE](#)].
- [9] Y. Li et al., *pmwd: A Differentiable Cosmological Particle-Mesh N-body Library*, [[arXiv:2211.09958](#)] [[INSPIRE](#)].
- [10] A.J. Zhou and S. Dodelson, *Field-level multiprobe analysis of the CMB, integrated Sachs-Wolfe effect, and the galaxy density maps*, *Phys. Rev. D* **108** (2023) 083506 [[arXiv:2304.01387](#)] [[INSPIRE](#)].
- [11] A.J. Zhou, X. Li, S. Dodelson and R. Mandelbaum, *Accurate field-level weak lensing inference for precision cosmology*, *Phys. Rev. D* **110** (2024) 023539 [[arXiv:2312.08934](#)] [[INSPIRE](#)].
- [12] J. Jasche and B.D. Wandelt, *Bayesian physical reconstruction of initial conditions from large scale structure surveys*, *Mon. Not. Roy. Astron. Soc.* **432** (2013) 894 [[arXiv:1203.3639](#)] [[INSPIRE](#)].
- [13] H. Wang et al., *ELUCID — Exploring the Local Universe with reConstructed Initial Density field I: Hamiltonian Markov Chain Monte Carlo Method with Particle Mesh Dynamics*, *Astrophys. J.* **794** (2014) 94 [[arXiv:1407.3451](#)] [[INSPIRE](#)].
- [14] U. Seljak, G. Aslanyan, Y. Feng and C. Modi, *Towards optimal extraction of cosmological information from nonlinear data*, *JCAP* **12** (2017) 009 [[arXiv:1706.06645](#)] [[INSPIRE](#)].
- [15] J. Jasche and G. Lavaux, *Physical Bayesian modelling of the non-linear matter distribution: new insights into the Nearby Universe*, *Astron. Astrophys.* **625** (2019) A64 [[arXiv:1806.11117](#)] [[INSPIRE](#)].
- [16] F. Schmidt et al., *A rigorous EFT-based forward model for large-scale structure*, *JCAP* **01** (2019) 042 [[arXiv:1808.02002](#)] [[INSPIRE](#)].
- [17] J. Alsing, A.F. Heavens and A.H. Jaffe, *Cosmological parameters, shear maps and power spectra from CFHTLenS using Bayesian hierarchical inference*, *Mon. Not. Roy. Astron. Soc.* **466** (2017) 3272 [[arXiv:1607.00008](#)] [[INSPIRE](#)].



- [18] E. Anderes, B. Wandelt and G. Lavaux, *Bayesian inference of CMB gravitational lensing*, *Astrophys. J.* **808** (2015) 152 [[arXiv:1412.4079](#)] [[INSPIRE](#)].
- [19] F.-S. Kitaura et al., *COSMIC BIRTH: Efficient Bayesian Inference of the Evolving Cosmic Web from Galaxy Surveys*, *Mon. Not. Roy. Astron. Soc.* **502** (2021) 3456 [[arXiv:1911.00284](#)] [[INSPIRE](#)].
- [20] M. Ata et al., *BIRTH of the COSMOS Field: Primordial and Evolved Density Reconstructions During Cosmic High Noon*, *Mon. Not. Roy. Astron. Soc.* **500** (2020) 3194 [[arXiv:2004.11027](#)] [[INSPIRE](#)].
- [21] P. Fiedorowicz et al., *KaRMMa — kappa reconstruction for mass mapping*, *Mon. Not. Roy. Astron. Soc.* **512** (2022) 73 [[arXiv:2105.14699](#)] [[INSPIRE](#)].
- [22] P. Fiedorowicz, E. Rozo and S.S. Boruah, *KaRMMa 2.0 — Kappa Reconstruction for Mass Mapping*, [arXiv:2210.12280](#) [[INSPIRE](#)].
- [23] M. Millea et al., *Optimal Cosmic Microwave Background Lensing Reconstruction and Parameter Estimation with SPTpol Data*, *Astrophys. J.* **922** (2021) 259 [[arXiv:2012.01709](#)] [[INSPIRE](#)].
- [24] LSST DARK ENERGY SCIENCE (LSST DESC) collaboration, *Forecasting the power of higher order weak-lensing statistics with automatically differentiable simulations*, *Astron. Astrophys.* **679** (2023) A61 [[arXiv:2305.07531](#)] [[INSPIRE](#)].
- [25] N. Porqueres et al., *Field-level inference of cosmic shear with intrinsic alignments and baryons*, [arXiv:2304.04785](#) [[INSPIRE](#)].
- [26] N.-M. Nguyen et al., *How much information can be extracted from galaxy clustering at the field level?*, [arXiv:2403.03220](#) [[INSPIRE](#)].
- [27] S. Dodelson et al., *Second order geodesic corrections to cosmic shear*, *Phys. Rev. D* **72** (2005) 103004 [[astro-ph/0503160](#)] [[INSPIRE](#)].
- [28] A. Cooray and W. Hu, *Second order corrections to weak lensing by large scale structure*, *Astrophys. J.* **574** (2002) 19 [[astro-ph/0202411](#)] [[INSPIRE](#)].
- [29] E. Krause and C.M. Hirata, *Weak lensing power spectra for precision cosmology: Multiple-deflection, reduced shear and lensing bias corrections*, *Astron. Astrophys.* **523** (2010) A28 [[arXiv:0910.3786](#)] [[INSPIRE](#)].
- [30] G. Pratten and A. Lewis, *Impact of post-Born lensing on the CMB*, *JCAP* **08** (2016) 047 [[arXiv:1605.05662](#)] [[INSPIRE](#)].
- [31] G. Fabbian, M. Calabrese and C. Carbone, *CMB weak-lensing beyond the Born approximation: a numerical approach*, *JCAP* **02** (2018) 050 [[arXiv:1702.03317](#)] [[INSPIRE](#)].
- [32] A. Petri, Z. Haiman and M. May, *Validity of the Born approximation for beyond Gaussian weak lensing observables*, *Phys. Rev. D* **95** (2017) 123503 [[arXiv:1612.00852](#)] [[INSPIRE](#)].
- [33] G. Fabbian, A. Lewis and D. Beck, *CMB lensing reconstruction biases in cross-correlation with large-scale structure probes*, *JCAP* **10** (2019) 057 [[arXiv:1906.08760](#)] [[INSPIRE](#)].
- [34] C. Chang and B. Jain, *Delensing Galaxy Surveys*, *Mon. Not. Roy. Astron. Soc.* **443** (2014) 102 [[arXiv:1405.1432](#)] [[INSPIRE](#)].
- [35] V. Böhm, Y. Feng, M.E. Lee and B. Dai, *MADLens, a python package for fast and differentiable non-Gaussian lensing simulations*, *Astron. Comput.* **36** (2021) 100490 [[arXiv:2012.07266](#)] [[INSPIRE](#)].
- [36] S. Dodelson, C. Shapiro and M.J. White, *Reduced shear power spectrum*, *Phys. Rev. D* **73** (2006) 023009 [[astro-ph/0508296](#)] [[INSPIRE](#)].

- [37] B. Jain, U. Seljak and S.D.M. White, *Ray tracing simulations of weak lensing by large scale structure*, *Astrophys. J.* **530** (2000) 547 [[astro-ph/9901191](#)] [[INSPIRE](#)].
- [38] C.M. Hirata and U. Seljak, *Reconstruction of lensing from the cosmic microwave background polarization*, *Phys. Rev. D* **68** (2003) 083002 [[astro-ph/0306354](#)] [[INSPIRE](#)].
- [39] S. Hilbert et al., *The Accuracy of Weak Lensing Simulations*, *Mon. Not. Roy. Astron. Soc.* **493** (2020) 305 [[arXiv:1910.10625](#)] [[INSPIRE](#)].
- [40] D. Beck, G. Fabbian and J. Errard, *Lensing Reconstruction in Post-Born Cosmic Microwave Background Weak Lensing*, *Phys. Rev. D* **98** (2018) 043512 [[arXiv:1806.01216](#)] [[INSPIRE](#)].
- [41] V. Böhm et al., *Effect of non-Gaussian lensing deflections on CMB lensing measurements*, *Phys. Rev. D* **98** (2018) 123510 [[arXiv:1806.01157](#)] [[INSPIRE](#)].
- [42] V. Böhm, C. Modi and E. Castorina, *Lensing corrections on galaxy-lensing cross correlations and galaxy-galaxy auto correlations*, *JCAP* **03** (2020) 045 [[arXiv:1910.06722](#)] [[INSPIRE](#)].
- [43] C. Vale and M.J. White, *Simulating weak lensing by large scale structure*, *Astrophys. J.* **592** (2003) 699 [[astro-ph/0303555](#)] [[INSPIRE](#)].
- [44] S. Hilbert, S.D.M. White, J. Hartlap and P. Schneider, *Strong lensing optical depths in a LambdaCDM universe*, *Mon. Not. Roy. Astron. Soc.* **382** (2007) 121 [[astro-ph/0703803](#)] [[INSPIRE](#)].
- [45] S. Hilbert, J. Hartlap, S.D.M. White and P. Schneider, *Ray-tracing through the Millennium Simulation: Born corrections and lens-lens coupling in cosmic shear and galaxy-galaxy lensing*, *Astron. Astrophys.* **499** (2009) 31 [[arXiv:0809.5035](#)] [[INSPIRE](#)].
- [46] M. Sato et al., *Simulations of Wide-Field Weak Lensing Surveys I: Basic Statistics and Non-Gaussian Effects*, *Astrophys. J.* **701** (2009) 945 [[arXiv:0906.2237](#)] [[INSPIRE](#)].
- [47] M.R. Becker, *CALCLENS: Weak Lensing Simulations for Large-area Sky Surveys and Second-order Effects in Cosmic Shear Power Spectra*, [arXiv:1210.3069](#) [[INSPIRE](#)].
- [48] A. Petri, *Mocking the Weak Lensing universe: the LensTools python computing package*, *Astron. Comput.* **17** (2016) 73 [[arXiv:1606.01903](#)] [[INSPIRE](#)].
- [49] R. Takahashi et al., *Full-sky Gravitational Lensing Simulation for Large-area Galaxy Surveys and Cosmic Microwave Background Experiments*, *Astrophys. J.* **850** (2017) 24 [[arXiv:1706.01472](#)] [[INSPIRE](#)].
- [50] K. Osato, J. Liu and Z. Haiman,  *$\kappa$ TNG: effect of baryonic processes on weak lensing with IllustrisTNG simulations*, *Mon. Not. Roy. Astron. Soc.* **502** (2021) 5593 [[arXiv:2010.09731](#)] [[INSPIRE](#)].
- [51] C. Wei et al., *Full-sky ray-tracing simulation of weak lensing using ELUCID simulations: exploring galaxy intrinsic alignment and cosmic shear correlations*, *Astrophys. J.* **853** (2018) 25 [[arXiv:1801.03941](#)] [[INSPIRE](#)].
- [52] K. Xu and Y. Jing, *An Accurate  $P^3M$  Algorithm for Gravitational Lensing Studies in Simulations*, [arXiv:2102.08629](#) [[DOI:10.3847/1538-4357/ac0249](#)] [[INSPIRE](#)].
- [53] J.J. Vicente and E. Mediavilla, *Fast Multipole Method for Gravitational Lensing: Application to High-magnification Quasar Microlensing*, *Astrophys. J.* **941** (2022) 80 [[arXiv:2211.00354](#)] [[INSPIRE](#)].
- [54] X. Suo, X. Kang, C. Wei and G. Li, *The Spherical Fast Multipole Method (sFMM) for Gravitational Lensing Simulation*, *Astrophys. J.* **948** (2023) 56 [[arXiv:2210.07021](#)] [[INSPIRE](#)].

- [55] H.M.P. Couchman, A.J. Barber and P.A. Thomas, *Measuring the three-dimensional shear from simulation data, with applications to weak gravitational lensing*, *Mon. Not. Roy. Astron. Soc.* **308** (1999) 180 [[astro-ph/9810063](#)] [[INSPIRE](#)].
- [56] B. Li, L.J. King, G.-B. Zhao and H.S. Zhao, *A Semi-analytic Ray-tracing Algorithm for Weak Lensing*, *Mon. Not. Roy. Astron. Soc.* **415** (2011) 881 [[arXiv:1012.1625](#)] [[INSPIRE](#)].
- [57] A. Barreira, C. Llinares, S. Bose and B. Li, *RAY-RAMSES: a code for ray tracing on the fly in N-body simulations*, *JCAP* **05** (2016) 001 [[arXiv:1601.02012](#)] [[INSPIRE](#)].
- [58] M. Killedar, P.D. Lasky, G.F. Lewis and C.J. Fluke, *Gravitational Lensing with Three-Dimensional Ray Tracing*, *Mon. Not. Roy. Astron. Soc.* **420** (2012) 155 [[arXiv:1110.4894](#)] [[INSPIRE](#)].
- [59] R. Bar-Kana, *Gravitational lensing as a probe of dark matter, the distance scale, and gravitational waves in the universe*, Ph.D. thesis, May, 1997.
- [60] T.R. Quinn, N. Katz, J. Stadel and G. Lake, *Time stepping N body simulations*, [astro-ph/9710043](#) [[INSPIRE](#)].
- [61] P. Saha and S. Tremaine, *Symplectic integrators for solar system dynamics*, *Astron. J.* **104** (1992) 1633.
- [62] V. Springel, R. Pakmor, O. Zier and M. Reinecke, *Simulating cosmic structure formation with the gadget-4 code*, *Mon. Not. Roy. Astron. Soc.* **506** (2021) 2871 [[arXiv:2010.03567](#)] [[INSPIRE](#)].
- [63] H. Yoshida, *Recent progress in the theory and application of symplectic integrators*, *Celest. Mech. Dyn. Astron.* **56** (1993) 27.
- [64] S. Dodelson, *Modern Cosmology*, Academic Press (2003) [[INSPIRE](#)].
- [65] P. Schneider, J. Ehlers and E.E. Falco, *Gravitational Lenses*, Springer (1992) [[DOI:10.1007/978-3-662-03758-4](#)] [[INSPIRE](#)].
- [66] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, CRC Press (2021) [[DOI:10.1201/9780367806934](#)].
- [67] R. Epstein and I.I. Shapiro, *Post-post-Newtonian deflection of light by the Sun*, *Phys. Rev. D* **22** (1980) 2947 [[INSPIRE](#)].
- [68] PLANCK collaboration, *Planck 2015 results. XIII. Cosmological parameters*, *Astron. Astrophys.* **594** (2016) A13 [[arXiv:1502.01589](#)] [[INSPIRE](#)].
- [69] J.P. Naiman et al., *First results from the IllustrisTNG simulations: a tale of two elements — chemical evolution of magnesium and europium*, *Mon. Not. Roy. Astron. Soc.* **477** (2018) 1206.
- [70] V. Springel et al., *First results from the IllustrisTNG simulations: matter and galaxy clustering*, *Mon. Not. Roy. Astron. Soc.* **475** (2018) 676 [[arXiv:1707.03397](#)] [[INSPIRE](#)].
- [71] F. Marinacci et al., *First results from the IllustrisTNG simulations: radio haloes and magnetic fields*, *Mon. Not. Roy. Astron. Soc.* **480** (2018) 5113 [[arXiv:1707.03396](#)] [[INSPIRE](#)].
- [72] D. Nelson et al., *First results from the IllustrisTNG simulations: the galaxy colour bimodality*, *Mon. Not. Roy. Astron. Soc.* **475** (2018) 624 [[arXiv:1707.03395](#)] [[INSPIRE](#)].
- [73] A. Pillepich et al., *First results from the IllustrisTNG simulations: the stellar mass content of groups and clusters of galaxies*, *Mon. Not. Roy. Astron. Soc.* **475** (2018) 648 [[arXiv:1707.03406](#)] [[INSPIRE](#)].
- [74] A. Petri, Z. Haiman and M. May, *Sample variance in weak lensing: how many simulations are required?*, *Phys. Rev. D* **93** (2016) 063524 [[arXiv:1601.06792](#)] [[INSPIRE](#)].

- [75] Y. Zhang et al., *Neural and symbolic optimization of cosmological particle-mesh simulation*, in preparation.
- [76] R. Takahashi et al., *Revising the Halofit Model for the Nonlinear Matter Power Spectrum*, *Astrophys. J.* **761** (2012) 152 [[arXiv:1208.2701](#)] [[INSPIRE](#)].
- [77] L. Hu, A. Heavens and D. Bacon, *Light bending by the cosmological constant*, *JCAP* **02** (2022) 009 [[arXiv:2109.09785](#)] [[INSPIRE](#)].
- [78] R. Dalal et al., *Hyper Suprime-Cam Year 3 results: Cosmology from cosmic shear power spectra*, *Phys. Rev. D* **108** (2023) 123519 [[arXiv:2304.00701](#)] [[INSPIRE](#)].
- [79] P. Schneider, *A new formulation of gravitational lens theory, time-delay, and Fermat's principle*, *Astron. Astrophys.* **143** (1985) 413.
- [80] R. Blandford and R. Narayan, *Fermat's principle, caustics, and the classification of gravitational lens images*, *Astrophys. J.* **310** (1986) 568 [[INSPIRE](#)].