An Accurate P3M Algorithm for Gravitational Lensing Studies in Simulations

Kun Xu $^{1}$ and Yipeng Jing $^{1,2}$

$^{1}$Department of Astronomy, School of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai, 200240, China
$^{2}$Tsung-Dao Lee Institute, and Shanghai Key Laboratory for Particle Physics and Cosmology, Shanghai Jiao Tong University, Shanghai, 200240, China

(Received xxxx xx, xxxx; Revised xxxx xx, xxxx; Accepted xxxx xx, xxxx)

Submitted to ApJ

ABSTRACT

We present a two-dimensional (2D) Particle-Particle-Particle-Mesh (P3M) algorithm with an optimized Green function and adaptive softening length for gravitational lensing studies in N-Body simulations. The analytical form of the optimized Green function $\hat{G}(k)$ is given. The softening schemes ($S$) are studied for both the PM and the PP calculations in order for accurate force calculation and suppression of the particle discreteness effect. Our method is two orders of magnitude more accurate than the simple PM algorithm with the poor man’s Green function ($\propto 1/k^2$) at a scale of a few mesh cells or smaller. The force anisotropy is also much smaller than the conventional PM calculation. We can achieve a force accuracy better than 0.1 percent at all scales with our algorithm, which makes it an ideal (accurate and fast) algorithm for micro lensing studies. When we apply the algorithm to computing weak and strong lensing quantities in N-Body simulations, the errors are dominated by the Poisson noise caused by particle discreteness. The Poisson noise can be suppressed by smoothing out the particle distribution, which can be achieved by simply choosing an adaptive softening length in the PP calculation. We have presented a criterion to set the adaptive softening length. Our algorithm is also applicable to cosmological simulations. We provide a Python implementation P3Mlens for this algorithm.

Keywords: Gravitational lensing (670); N-body simulations (1083); Algorithms (1883)

1. INTRODUCTION

Nowadays gravitational lensing becomes a powerful tool to investigate the distribution of dark matter, both for dark matter halos and for larger scale structures (Treu 2010; Mandelbaum 2018). With the advent of ongoing or planned large imaging and spectroscopic surveys such as LSST (Ivezić et al. 2019), DESI (Dey et al. 2019), HSC-SSP (Aihara et al. 2018) and PFS (Takada et al. 2014), the statistical uncertainties of observational measurement for gravitational lensing will be very small, but the accuracy will be limited by remaining systematics in the observation.

Accurate theoretical modeling will be required to properly interpret the lensing observations. A widely used way for modeling is to perform ray-tracing computation on the basis of high resolution N-body simulations (Wambsganss et al. 1998; Jain et al. 2000; Takada & Jain 2003a). In nearly all the ray-tracing methods, single or multiple plane approximation is used (Bartelmann & Weiss 1994; Heymans et al. 2006; Fosalba et al. 2008), where the 3D matter distribution around each plane is projected onto the plane (the 2D distribution). Thus, the main step in the ray-tracing computation is to accurately calculate the deflection angle of light for the lens plane, which can be reduced to a physical problem of solving a 2D Poisson’s equation.

In the literature, 2D Poisson’s equation is usually solved using a Particle-Mesh (PM) algorithm. At first, mass of particles is assigned to a grid used for Fast Fourier Transform (FFT). Then, potential is obtained by solving Poisson’s equation in Fourier space with a Green function where $\hat{G}(k) \propto 1/k^2$ is often used. Differential is approximated by finite-difference to obtain the
force field or the lensing parameters on the grid from the potential, and finally the lensing quantities on the grid are interpolated to the whole space. Although the Green function is correct analytically, because the calculation is done on the grid with finite interval in the PM, errors are inevitably generated due to under-sampling (alias) and anisotropies in each step from mass assignment to force interpolation. This is usually called the poor man’s Poisson Solver. Hockney & Eastwood (1981) provides a method to minimize the error in the whole PM process by regarding the Green function as a free parameter and optimizing it. The force has to be softened on small scale in the PM calculation even with the optimized Green function. Thus, the PP algorithm is employed to compensate for the force softening in the PM calculation, and the force can be calculated to a very high precision at all scales. The method is usually called the Particle-Particle-Particle-Mesh (P³M) algorithm.

Another source of error in ray-tracing simulations is Poisson noise. Since particle distribution in N-body simulation is just a Monte Carlo sampling of the underlining density distribution, Poisson noise is unavoidable. Smoothing is the main method to reduce the noise and recover the density distribution, which has been investigated in many studies (Bradač et al. 2004; Li et al. 2006; Yi-bin et al. 2009). Kernels of Gaussian or other shapes with an adaptive smoothing length are usually used in PM algorithm when performing mass assignment. In P³M algorithm, as we will show, the smoothing can be fully incorporated in the PP calculation by transforming point sources to shaped sources. This is another advantage of the P³M algorithm to predict lensing quantities in N-Body simulations. Furthermore, we can easily adopt an adaptive smoothing length in the algorithm.

In this paper, we present a P³M algorithm for lensing studies in N-Body simulation. We study the accuracy of the computed lensing quantities, and investigate how the errors are related to the accuracy of the force calculation and to the Poisson noise of the discrete particle distribution. We give an analytical form of the optimized Green function and determine the best choice of the free parameters in PM. We also give an analytical form for the force between 2D shaped particles in real space, which is used both for the PP calculation and for the smoothing of the particle distribution. Our recommendation for the two softening (smoothing) lengths is given. Finally, we will briefly introduce our PYTHON implementation P³MLens of the 2D P³M algorithm.

This paper is organized as follows. In Section 2, we briefly introduce the gravitational lensing theory and the general P³M algorithm. In Section 3, our realization of the 2D P³M algorithm is provided. In Section 4, we test the accuracy of the algorithm. The PYTHON implementation is briefly introduced in Section 5 and our conclusions are summarized in Section 6. Throughout this paper, Fourier transform of a function \( f \) is denoted as \( \hat{f} \), and the convention of Fourier transform we use, as example for one dimension, is

\[
f(x) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \hat{f}(k)e^{ikx}
\]

\[
\hat{f}(k) = \int_{-\infty}^{\infty} dx f(x)e^{-ikx}
\]

2. METHOD

2.1. Gravitational lensing for thin lens

The deflection angle of light by a gravitational lens is the key quantity calculated in the ray tracing simulations. For a thin lens, according to General Relativity (GR), the deflection angle \( \alpha \) depends on the gravitational potential \( \Phi \) as follows:

\[
\alpha = \frac{2}{c^2} \int \nabla_2 \Phi \, dl = \frac{2}{c^2} \nabla_2 \int \Phi \, dl ,
\]

where \( \nabla_2 \) is the 2D gradient operator in the lens plane. Using Poisson’s equation for 3D gravitational potential, we can easily prove,

\[
\nabla_2^2 \int \Phi \, dl = \int \nabla^2 \Phi - \frac{\partial^2 \Phi}{\partial l^2} \, dl = 4\pi G \int \rho \, dl - \frac{\partial \Phi}{\partial l} \bigg|_{\infty} = 4\pi G \Sigma .
\]

In the above equation, \( \Sigma \) is the surface density of the lens. The second term on the rhs vanishes due to the zero boundary condition for the potential. It’s convenient to define 2D potential \( \psi = \int \Phi \, dl \) and field \( E = -\nabla_2 \psi \). We have \( \alpha = -\frac{D_l}{r} E \). Thus, the goal of ray-tracing simulations is to solve the 2D Poisson’s equation from the known matter distribution of an N-body simulation,

\[
\nabla_2^2 \psi = 4\pi G \Sigma
\]

It’s easy to derive the 2D force field for a point source:

\[
E = -\frac{2Gm r}{r^2} ,
\]

where \( m \) is the mass of the point source.

For a single thin lens, gravitational lensing parameters like shear (\( \gamma_1, \gamma_2 \)), convergence (\( \kappa \)) and magnification (\( \mu \)) can be calculated if \( \alpha \) is known in the lens plane.

\[
\gamma_1 = \frac{D_1D_{ls}}{2D_s} \left( \frac{\partial \alpha_x}{\partial y} - \frac{\partial \alpha_y}{\partial x} \right)
\]
\[ \gamma_2 = \frac{D_l D_{ls}}{2D_s} \left( \frac{\partial \alpha_x}{\partial y} + \frac{\partial \alpha_y}{\partial x} \right) \] (8)

\[ \gamma = \gamma_1 + i\gamma_2 \] (9)

\[ \kappa = \frac{D_l D_{ls}}{2D_s} \left( \frac{\partial \alpha_x}{\partial x} + \frac{\partial \alpha_y}{\partial y} \right) \] (10)

\[ \mu = \frac{1}{(1 - \kappa)^2 - |\gamma|^2} \] (11)

where \( D_l, D_s \) and \( D_{ls} \) are the angular diameter distances from the observer to the lens, the observer to the source and from the lens to the source.

### 2.2. General multi-dimensional P\(^3\)M algorithm

\( \text{P}^3\text{M} \) algorithms are a class of hybrid algorithms developed decades ago to simulate plasma systems using particles. They have been used to simulate the evolution of large-scale structures in the Universe (e.g., Efstathiou et al. (1985); Couchman (1991); Jing et al. (1994)).

The short-range force on a particle is computed by directly summing up particle-particle (PP) pair force, and the smoothly varying long-range force is approximated by the particle-mesh (PM) force calculation. In this subsection, we briefly introduce the main idea of the \( \text{P}^3\text{M} \) algorithm and we recommend Hockney & Eastwood (1981) for details. The main steps of the \( \text{P}^3\text{M} \) algorithm include: PM (mass assignment, potential solving, potential difference and force interpolation) and PP (force splitting and summation) calculations.

#### 2.2.1. Mass assignment

To take advantage of the Fast Fourier Transform (FFT) algorithm, the density field is assigned to a regular grid. Assignment schemes commonly adopted are Nearest Grid Point (NGP), Cloud In Cell (CIC), Triangular Shape Cloud (TSC) and Piecewise Cubic Spline (PCS), which corresponds to the zeroth, first, second and third order piecewise polynomial functions \( W^{(p)} \) with \( p = 0,1,2,3 \) respectively. For each of the schemes, we have a one-dimensional function:

\[ W^{(0)}(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \] (12)

\[ W^{(1)}(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \] (13)

\[ W^{(2)}(x) = \begin{cases} \frac{3}{4} - x^2 & |x| < \frac{1}{2} \\ \frac{1}{2} \left( \frac{3}{2} - |x| \right)^2 & \frac{1}{2} \leq |x| < \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \] (14)

\[ W^{(3)}(x) = \begin{cases} \frac{1}{6} (4 - 6x^2 + 3|x|^3) & 0 \leq |x| < 1 \\ \frac{1}{2} (2 - |x|)^3 & 1 \leq |x| < 2 \\ 0 & \text{otherwise} \end{cases} \] (15)

The mass assignment weight function is given by

\[ W(x) = \prod_j W^{(p)}(x^j/H), \]

where \( x_i \) is the position of particle \( i \), \( N_P \) is the number of particles and \( \delta(x) \) is the Dirac delta function. The mass density after assignment can be described by:

\[ \rho(x) = \frac{m}{V_g} \int n(x') W(x - x') dx' \]

where \( m \) is the particle mass and \( V_g \) is the volume of the grid. The mass density grid used for FFT can be generated by a sampling process described by the sampling function:

\[ \text{III}(x) = \sum_n \delta(x - n) \]

The Green function \( G(x) \) describes the potential from a unit mass point source. The solution of Poisson’s equation can be written as:

\[ \psi(x) = G(x) \ast \rho'(x) \] (21)

The equation is usually calculated in Fourier space \( (k) \) in which the convolution is transformed to a simple multiplication product. In the literature, there are many choices for the Green function, e.g., the “poor man’s” Green function \( G(k) \propto 1/k^2 \). However, here we will treat it as a free parameter which will be optimized after we take into account of all the errors in the PM calculation.
The force field is calculated from the potential by $E = -\nabla \psi$. However, the differential is approximated by a difference in PM:

$$E(x) = -D(x) * \psi(x)$$  \hfill (22)

The first order approximation for the differential is the two-point finite-difference,

$$D(x) = \frac{\delta(x + H) - \delta(x - H)}{2H}$$  \hfill (23)

$$D(x) = \sum_{j=1}^{d} D(x^j) \delta^j$$  \hfill (24)

The error of the difference is at the order of $O(H^2)$. And the next order approximation is the four-point finite-difference,

$$D(x) = \frac{\beta \delta(x + H) - \delta(x - H)}{2H}$$
$$+ (1 - \beta) \frac{\delta(x + 2H) - \delta(x - 2H)}{4H}$$  \hfill (25)

Setting $\beta = 4/3$ leads to a cancellation of $O(H^2)$ term and the accuracy of order $O(H^4)$ can be achieved. Higher order approximations can be more accurate, but they are also more time-consuming in the calculation since more points are used for the difference. The force field grid $E'$ calculated by PM is also a sampling of the underlining force field $E$:

$$E'(x) = \Pi(\frac{x}{H}) D(x) * \psi(x).$$  \hfill (26)

Finally, the force field should be interpolated to the whole space. We use the same interpolation function for the force as we did for the mass assignment,

$$F(x) = \frac{m}{V_g} W(x) * E'(x).$$  \hfill (27)

When the two interpolation functions are taken the same, the momentum of the system is conserved.

### 2.2.3. Optimized Green function

Though the Green function ($\hat{G}(k) \propto 1/k^2$) is accurate in poor man’s Poisson Solver, errors are generated in every step of the PM calculation. For example, in the mass assignment, error called alias is generated during sampling if $\hat{\rho}(k)$ is non-zero beyond the Nyquist frequency:

$$\hat{\rho}(k) = \sum_{n=-\infty}^{+\infty} \hat{\rho}(k - nk_g)$$  \hfill (28)

with $k_g = 2\pi/H$. Using a finite-difference to replace differential also generates error, and so does the interpolation. Thus, the Green function is better regarded as a free parameter to be optimized to compensate for all the errors arising in the PM calculation.

The Green function is optimized to minimize the average force error between two particles. The force on a unit mass particle at $x_2$ from another unit mass particle at $x_1$ is given by:

$$F(x_2) = \int \frac{dk}{(2\pi)^d} \hat{F}(k) e^{ik \cdot x_2}$$  \hfill (29)

From equations 20-22 and 26-28, we get

$$\hat{F}(k) = \frac{1}{V_g} \hat{W} \hat{E}'$$
$$= -\frac{1}{V_g} \hat{W} \hat{D} \hat{\psi}$$
$$= -\frac{1}{V_g} \hat{W} \hat{D} \hat{G} \hat{\rho}'$$
$$= -\frac{1}{V_g^2} \hat{W} \hat{D} \hat{G} \sum_n \hat{W}(k - nk_g) e^{-ik \cdot x_1} e^{-ik \cdot x_2}$$  \hfill (30)

The last step uses the Fourier transform of the delta function that $\delta(k) = e^{-ik \cdot x_1}$. Thus, the force between two particles in a periodic system becomes:

$$F(x_2; x_1) = \frac{1}{V_b} \sum_l \frac{\hat{W}}{V_g} \hat{D} \hat{G} \sum_n \frac{\hat{W}(kn)}{V_g} e^{-ik_n \cdot x_1} e^{-ik \cdot x_2}$$  \hfill (31)

Where $l$ is the wavenumber of the Fourier series for the periodic system, $V_b$ is the box volume and $k_n = k - nk_g$.

The displacement-averaged total squared deviation of the PM calculated force $F(x = x_2 - x_1; x_1)$ from the reference interparticle force $R(x)$ is defined as:

$$Q = \frac{1}{V_g} \int_{V_g} \int_{V_g} d x_1 d x_2 |F(x; x_1) - R(x)|^2$$  \hfill (32)

Using Fourier transform, $Q$ can be written as:

$$Q = \frac{1}{V_b} \sum_l \left\{ |\hat{D}|^2 \hat{G}^2 \left[ \sum_n \hat{U}(kn) \right]^2 \right. - 2 \hat{G} \hat{D} \cdot \sum_n \hat{R^*}(kn) \hat{G}^2 + \left. \sum_n |\hat{R}|^2 \right\}$$  \hfill (33)

where $\hat{U} = \hat{W}/V_g$. Minimizing $Q$ with respect to $\hat{G}$ gives the optimized Green function:

$$\hat{G}(k) = \frac{\hat{D}(k) \cdot \sum_n \hat{U}^*(kn) \hat{R^*}(kn)}{|\hat{D}(k)|^2 \left[ \sum_n \hat{U}^2(kn) \right]^2}$$  \hfill (34)
2.2.4. PP force calculation

Due to the very nature of the PM calculation, the PM force is both softened and anisotropic on small scale (smaller than a few $H$) compared to the reference force between two point particles. The PP algorithm is usually used to accurately calculate the short-range force, in order to compensate for the PM softening. To suppress the error from the anisotropy and to smoothly transit from the PM force to the PP force, a shape $S(r)$ is assigned to each particle for the PM, reflected in the softened $\hat{R}$:

$$\hat{R} = \hat{R}_p \hat{s}^2$$

(35)

where $\hat{R}_p$ is the Fourier transform of the force between two point sources. The shapes with polynomial forms are usually chosen for convenient analytical calculation, such as:

$$S_1(r; a) = \begin{cases} 
\frac{128}{\pi a^4} (a^{2/4} - r^2) & r < a/2 \\
0 & \text{otherwise}
\end{cases}$$

(36)

$$S_2(r; a) = \begin{cases} 
\frac{24}{\pi a^3} (a/2 - r) & r < a/2 \\
0 & \text{otherwise}
\end{cases}$$

(37)

in the 2D case. The smoothing length is controlled by $a = a_{pm}$, which also influences the volume to perform PP. Larger $a_{pm}$ can better suppress the total error at the cost of many more particle pairs in PP ($O(a_{pm}^p)$). The force used for PP is simply $F_{pp} = R_{tot} - R_{pm}$, where $R_{tot}$ and $R_{pm}$ are the reference total force and PM force respectively.

3. 2D P³M ALGORITHM FOR GRAVITATIONAL LENSING

First of all, the analytical form of the optimized Green function should be obtained. The analytic forms of $D$ and $\hat{U}$ are provided in Hockney & Eastwood (1981):

$$\dot{D}_x(k) = i\beta \sin \left( \frac{k_x H}{2H} \right) + i(1 - \beta) \sin \left( \frac{2k_x H}{2H} \right)$$

$$\dot{D}_y(k) = i\beta \sin \left( \frac{k_y H}{2H} \right) + i(1 - \beta) \sin \left( \frac{2k_y H}{2H} \right)$$

(38)

$$\hat{U}(k) = \left( \frac{\sin \left( \frac{k_x H}{2H} \right) \sin \left( \frac{k_y H}{2H} \right)}{\frac{k_x H}{2H} \frac{k_y H}{2H}} \right)^{p+1}$$

(39)

where $p$ is the order of the piecewise polynomial function used for mass assignment and force interpolation.

Obtaining $\hat{R}$ in 2D is more complicated than in 3D. We first calculate $\hat{R}_p$, and then consider $\hat{S}$ for a selected particle shape. For unit mass, $R_p(r) = 2Gr/r^2$, the 2D Fourier transform of the x component is:

$$\hat{R}_{px}(k) = 2G \int_0^\infty \int_0^\infty \frac{x}{x^2 + y^2} e^{-i(k_x x + i k_y y)} dx dy$$

$$= -2\pi G i \int_0^\infty \text{sgn}(k_x) e^{-|k_x| - ik_y y} dy$$

$$= -4\pi G \frac{\text{sgn}(k_x) k_x^2}{|k_x|^2 + k_y^2}$$

(37)

$$= -4\pi G \frac{ik_x}{k_x^2 + k_y^2}$$

(40)

Where $\text{sgn}$ is the sign function. Similarly we have:

$$\hat{R}_{py}(k) = -4\pi G \frac{ik_y}{k_x^2 + k_y^2}$$

(41)

Two forms of particle shape, Eqs. 36 and 37, are considered and compared here. $S_2$ is widely used in 3D simulations, which is however not preferred in the 2D case as will be shown. The Fourier transform of $S_1$ is,

$$\hat{S}_1(k; a) = \int_0^{2\pi} e^{-ikr \cos \theta} d\theta \int_0^\infty \frac{2}{\pi a^2} \left[ \frac{a^2}{4} - r^2 \right] dr ,$$

(42)

where $\theta$ is the angle between $r$ and $k$. Using Jacobi–Anger expansion,

$$e^{i z \cos \theta} = J_0(z) + 2 \sum_{n=1}^{\infty} i^n J_n(z) \cos(n \theta) ,$$

(43)

where $J_n$ is the $n$-th order Bessel functions of the first kind, we get

$$\hat{S}_1(k; a) = \frac{64}{a^2} \int_0^{a^2} \left[ a^2 - r^3 \right] J_0(kr) dr$$

$$= 128 \left[ \frac{k a}{2} \right]^2 J_1 \left( \frac{k a}{2} \right) - 32 \frac{a^2}{k^2 a^2} J_0 \left( \frac{k a}{2} \right)$$

(44)

Similarly we can derive $\hat{S}_2$,

$$\hat{S}_2(k; a) = \frac{12}{k^2 a^2} \left[ \frac{ka}{2} \right] H_0 \left( \frac{ka}{2} \right) - \frac{32}{k^2 a^2} J_0 \left( \frac{ka}{2} \right)$$

(45)

where $H_n$ is the $n$-th Struve function.

The analytic form of $\hat{G}$ can be derived with equations 38-39,41,44 and 45. The summation $\sum_n$ can be done within $|n| < 2$, for very fast decay of $\hat{U}^2$ and $\hat{R}$ with respect to $|k|$.

Free parameters $p$, $a_{pm}$ and the shape form remain to be determined. The main improvement with higher
order mass assignment $p$ is to suppress the anisotropy (Efstathiou et al. 1985). Thus, in the later part of this paper, we set $p = 2$ by default with the TSC mass assignment scheme. As we will show, with this scheme, we can achieve a sufficiently good isotropy with a reasonable choice of $a_{pm}$.

The error $Q$ (equation 33) in PM with different choices of particle shape and $a_{pm}$ can be calculated for the optimized Green function $\hat{G}$. To investigate the sources of the error, $Q$ can be split into two parts,

$$Z = \int dx |\langle F(x; x_1) \rangle - R(x)|^2$$

$$= \frac{1}{V_6} \sum_i \left\{ |\hat{D}|^2 \hat{G}^2 \left[ \sum_n \hat{U}^4 \right] - 2 \hat{G} \cdot \left[ \sum_n \hat{R}^* \hat{U}^2 \right] + \sum_n |\hat{R}|^2 \right\},$$  

and

$$P = \frac{1}{V_6} \sum_i dx_1 \int_{V_6} dx |F(x; x_1) - \langle F(x; x_1) \rangle|^2$$

$$= \frac{1}{V_6} \sum_i |\hat{D}|^2 \hat{G}^2 \left\{ \left[ \sum_n \hat{U}^2 \right]^2 - \sum_n \hat{U}^4 \right\}.$$  

In the above equations, $Z$ is the total squared deviation of the displacement-averaged mesh force from the reference force, which describes the deviation of the mean force at separation $r$ from the reference force, and $P$ is the total squared deviation of the mesh force from its displacement average, which reflects the anisotropy of the PM force. By definition, we have $Q = Z + P$. To construct the dimensionless error estimator, we define

$$Q^+ = \frac{Q}{(\frac{2\pi}{H})^2 (\pi H^2)}$$

$$Z^+ = \frac{Z}{(\frac{2\pi}{H})^2 (\pi H^2)}$$

$$P^+ = \frac{P}{(\frac{2\pi}{H})^2 (\pi H^2)}.$$  

The dependence of $\sqrt{Q^+}$, $\sqrt{Z^+}$ and $\sqrt{P^+}$ on $a_{pm}$ is shown in Figure 1. Results for the two different particle shapes are also presented. In general, $S_1$ performs better than $S_2$. And considering that the calculation of $J_n$ is much faster than $H_n$, we prefer $S_1$ in our 2D lensing calculation. As expected, with increasing $a_{pm}$, the error is decreasing. At small $a_{pm}$, the error is mainly from the deviation of the mean force from the reference force ($Z$). Gradually, the anisotropy ($P$) becomes a dominant part of the error as $a_{pm}$ increases. The decrease of the error is much slower once $a_{pm}$ reaches $6H$. Considering the computational cost increases with $a_{pm}$ in the PP calculation, we recommend $a_{pm} \approx 6H$. Moreover, using higher order finite-difference approximations may also reduce the error, which we don’t investigate further here, as the PM force is accurate enough (better than 0.1 percent on all scales, cf. Fig.3) with the above parameters.

To perform PP on the basis of PM, an analytic form of $F_{pp}(r)$, determined by $R_{tot}$ and $R_{pm}$, is needed. $R_{tot}$ should also be smoothed to recover the underlining matter distribution from the Monte Carlo sampling particles. Hence, real space force $R(r; a)$ between two particles of unit mass with shape $S_1$ for any $a$ should be known. Actually, knowing the magnitude of the force $R(r; a)$ at distance $r$ is enough since $R$ doesn’t depend on the direction of $r$ and the direction of $R$ can be easily calculated from the particle positions. The Fourier transform of the $x$ component of $R(r; a)$ is:

$$\hat{R}_x(k_x, k_y) = -4\pi G \frac{ik_x \hat{S}^2_1(k; a)}{k^2}.$$  

Transforming it to real space,

$$R_x(x, y) = -\frac{4\pi G}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{ik_x \hat{S}^2_1(k)}{k^2} e^{ik_x x + ik_y y} dk_x dk_y.$$  

Figure 1. The dependence of $\sqrt{Q^+}$, $\sqrt{Z^+}$ and $\sqrt{P^+}$ on $a_{pm}$. The solid lines show the results for particle shape $S_2$ and dashed ones for $S_1$. 
we can get \( R(r; a) \) from \( R_x(x, y) \) by setting \( x = r \) and \( y = 0 \),

\[
R(r; a) = R_x(r, 0) = \frac{G}{\pi} \int_0^\infty \int_0^{2\pi} i\cos \theta e^{ikr \cos \theta} d\theta dk_y
\]

\[
= \frac{G}{\pi} \int_0^\infty \int_0^{2\pi} i\cos \theta S_1^2(k) e^{ikr \cos \theta} d\theta dk_y
\]

(53)

Using Jacobi–Anger expansion, and integrating \( \theta \) component, we get

\[
\int_0^{2\pi} \cos \theta e^{ikr \cos \theta} d\theta
\]

\[
= 2 \sum_{n=1}^{\pm \infty} \int_0^{2\pi} i^n J_n(kr) \cos(n\theta) \cos \theta d\theta
\]

(54)

Only the \( n = 1 \) component survives in \( \int_0^{2\pi} \cos(n\theta) \cos \theta d\theta \), thus

\[
\int_0^{2\pi} \cos \theta e^{ikr \cos \theta} d\theta = 2 \int_0^{2\pi} iJ_1(kr) \cos^2 \theta d\theta
\]

\[
= 2\pi iJ_1(kr)
\]

(55)

Finally we have

\[
R(r; a) = 2G \int_0^\infty J_1(kr) S_1^2(k; a) dk
\]

(56)

The integration is hard to calculate analytically, so we provide an accurate (< 10^{-3}, Figure 2) fitting formula for fast calculation:

\[
\begin{align*}
\frac{1}{\pi} & \begin{pmatrix}
0.743081 \xi^4 - 1.832992 \xi^3 - 0.057116 \xi^2 \\
+2.672707 \xi - 0.000082
\end{pmatrix} \\
& +2G \begin{pmatrix}
-0.524104 \xi^4 + 4.081230 \xi^3 - 11.856245 \xi^2 \\
+15.107021 \xi - 6.823192 + 1.539967/\xi
\end{pmatrix}
\end{align*}
\]

(57)

where \( \xi = 2r/a \). Thus,

\[
F_{pp}(r) = R(r; a_{pp}) - R(r; a_{pm})
\]

(58)

The PP calculation for each point of interest should be performed within the space range \( \max(a_{pm}, a_{pp}) \) since \( F_{pp} \) vanishes beyond, so the computational cost highly depends on the choice of these two parameters. We have discussed the choice of \( a_{pm} \) already, and we will determine \( a_{pp} \) in next section.

The purpose of this paper is to accurately calculate the 2D potential field \( E \), which is proportional to the deflection angle \( \alpha \). Although we only discussed about
the calculation of $R_{\text{tot}}$, it can be converted to $E$ by simply dividing it by the particle mass.

4. TEST

In this section, we test the accuracy of our algorithm and determine the smoothing length for gravitational lensing studies in N-body simulations.

4.1. Point source

We first calculate the 2D force field from a point source with our algorithm, and compare the performance with that of the poor man’s PM Poisson solver ($G(k) \propto 1/k^2$). We generate particles with different positions in the same grid cell, and calculate force field for different directions at a particular $r$. Errors of our algorithm and of the poor man’s PM are shown in Figure 3. The test is done with $H = 1$ kpc/$h$, $a_{pm} = 6H$, $a_{pp} = 0$, particle mass $m = 5 \times 10^8 M_\odot/h$ and box size 10000 kpc/$h$. The large box is used to reduce the effect of the periodic images. The mean deviation from the theoretical force field and the standard deviation in different directions of our algorithm are both two orders of magnitude smaller than the PM at $r$ smaller than $a_{pm}$, which can be attributed to our optimized PM and PP calculations. At large $r$, the mean deviation of the two algorithms is nearly the same but the standard deviation is slightly smaller in our P3M algorithm due to optimized $G$. Overall, the force between the two particles is calculated with the error smaller than $10^{-3}$ in our P3M algorithm at all scales. At small $r$, the PM algorithm with optimized Green function also performs better than the conventional one (green lines). However, in P3M algorithm, since the short-range force is dominated by PP calculation, the improvement in accuracy at small $r$ is mainly due to PP.

The precise calculation of forces between point sources at all $r$ makes P3M an ideal and elegant algorithm for micro lensing simulations. The calculation is much faster than the pure PP algorithm and the accuracy is guaranteed.

4.2. Halo with NFW profile

As shown in the previous subsection, we can achieve a high precision in the force calculation. However, dark matter distribution is represented by particles in N-body simulations, which inevitably leads to a Poisson noise in the lensing quantities. To suppress the Poisson noise, one needs to smooth out the density distribution, which can be done in our algorithm by just choosing a proper softening length $a_{pp}$ in the PP calculation. The softening length needs to be large enough to smooth the density field but be small enough to maintain the underlying density distribution. We adopt an adaptive softening length $a_{pp}$ for each point of interest as the distance to the $N_{nb}$-th nearest neighbor particle. We calculate lensing quantities $\kappa$, $\gamma$ and $\mu$ for NFW (Navarro et al. 1997) halos, and compare the results with the theoretical prediction. From the comparison, we test the performance for different $N_{nb}$ and try to give a recommendation for the choice of $N_{nb}$.

We test our algorithm with an NFW halo of the virial mass $M_{\text{vir}} = 10^{14.0} M_\odot/h$ and the virial radius $R_{\text{vir}} = 709$ kpc/$h$ ($z = 0.5$). We use the mass-concentration relation from Dutton & Macciò (2014). Analytical forms of the lensing quantities for an NFW halo have been provided by Takada & Jain (2003a) and Takada & Jain (2003b). We generate Monte Carlo particles for such a halo with particle mass of $5 \times 10^8 M_\odot/h$, which is the typical mass resolution ($10^8 \sim 10^9 M_\odot/h$) for cosmological simulations. The calculation is done with $H = 1$ kpc/$h$, $a_{pm} = 6H$ and box size $bR_{\text{vir}}$. Redshift is set to be 1.0 and 0.5 for the source and lens planes respectively.

In Figure 4, we show the relative deviation of the mean values from the theoretical predictions and the standard deviation normalized by the mean value for $\kappa$, $\gamma$ and $\delta \mu = \mu - 1$ at different $r/r_s$, where $r_s$ is the scale radius of the NFW profile. The former (the mean deviation) shows the systematic deviation caused by the smoothing, and the latter (the standard deviation) reflects the shot noise fluctuation left by the smoothing. As expected, the standard deviation is decreasing with increasing $N_{nb}$ for all lensing parameters at nearly all radius. If only the Poisson noise were considered, $N_{nb}$ should be as large as possible. However, the mean quantities will deviate from the predictions due to the smoothing. We take the adaptive softening length $a_{pp}$ at the center of the halo as $a_{min}$, which is also the smallest softening length for the halo. As shown in Figure 4, the mean deviation at the central region $r < a_{min}$ increases with the increase of $N_{nb}$. Thus, a compromise for the choice of $N_{nb}$ can be reached in order to make the mean deviation and the standard deviation to be comparable. From the test, we find that $N_{nb} = 400 \sim 800$ are recommended if the very inner part $r < a_{min}$ is not considered. Under this choice, the relative standard deviation is smaller than 7% for $\kappa$, 20% for $\gamma$ and 8% for $\delta \mu$, and the mean deviation is smaller than 4% for $\kappa$, 12% for $\gamma$ and 5% for $\delta \mu$ at $r > a_{min}$.

To apply our code to cosmological simulations, performance on smaller halos should also be investigated. In Figure 5, we show the results for halos with mass from $10^{12.0} M_\odot/h$ to $10^{14.0} M_\odot/h$. The concentration parameter is taken again from Dutton & Macciò (2014). $N_{nb}$ is set to 400. For the smaller halos, our code still performs...
Figure 4. The relative deviation of the mean value from the theoretical prediction (left) and the standard deviation normalized by the mean value (right) for $\kappa$ (top), $\gamma$ (middle) and $\delta\mu$ (bottom) as function of radius. The results for different smoothing lengths, denoted by $N_{nb}$, are shown in different colors. For each $N_{nb}$, a vertical dashed line is drawn to indicate the smallest smoothing length $a_{\text{min}}$ around the halo center.

Figure 5. The same as Figure 4 but for halos with a different mass. $N_{nb}$ is set to 400.
well for their outer parts \((r > a_{\text{min}})\), and our results does not depend on the halo mass.

Let us consider a halo simulated with different resolutions, and investigate the choice of \(N_{\text{nb}}\). We still use the NFW halo of the virial mass \(M_{\text{vir}} = 10^{14.0} M_\odot / h\). The higher resolution one has a particle mass \(10^7 M_\odot / h\), and the lower resolution one has the same resolution as before. The performance with different \(N_{\text{nb}}\) is shown in Figure 6. The standard deviation can be reduced roughly as \(N_{\text{nb}}^{-1/2}\) for all the three quantities at the radius \(r > a_{\text{min}}\). It is also interesting to note the standard deviation remains the same for the different resolutions when \(N_{\text{nb}}\) is taken the same. In contrast, the systematic deviations are only slightly reduced the higher resolution simulations even with the increase of \(N_{\text{nb}}\). Because the shear \(\gamma\) is a long-range interaction quantity, the smoothing at the central region of a halo can result in a significant systematic deviation at \(r\) smaller than a few \(a_{\text{min}}\), which in turn leads to the error of the magnification.

From the above tests, we can conclude that adaptive smoothing with \(N_{\text{nb}} \approx 400\) should work for most lensing studies. If one needs a smaller statistical deviation (noise), one may increase \(N_{\text{nb}}\) but at the expense of losing resolution at high density regions. The halo cores of \(r < a_{\text{min}}\) cannot be resolved, and the shear in the inner regions could be significantly underestimated by the smoothing.

Finally, we compare the conventional PM algorithm with our P3M. The PM algorithm needs to use an adaptive smoothing algorithm to get a smooth density field, otherwise the discreteness effect will be significant. In principle, the accuracy of the PM algorithm can always reach a similar level of our P3M if the grid length \(H\) is small enough. For example, for the halo of particle mass \(5 \times 10^6 M_\odot / h\), \(H\) should be taken \(\leq 1\) kpc/h for PM to get a similar accuracy as P3M. However, the grid length \(H\) may not be suitable for another halo that is of different mass or resolved by a different mass resolution. Especially if we want to generate a lensing map for a cosmological simulation where there many halos of different mass, it is difficult for the PM calculation achieve high accuracy for all halos in the simulation.

5. **P3MLens**: A PYTHON IMPLEMENTATION OF THE 2D P3M ALGORITHM

We provide a PYTHON script **P3MLens**\(^1\) (Xu 2021) that implements the 2D P3M algorithm for gravitational lensing. **P3MLens** construct a lens plane from input particles and return the quantities like \(E, \alpha, \kappa, \gamma, \mu\) for positions of interest. Taking full use of the numerical packages **numpy**\(^2\) and **scipy**\(^3\), calculations can be done in a very efficient way in PYTHON, even though it is a scripting language. Since the **for** loop of PYTHON is very slow, **numba**\(^4\), which translates PYTHON functions to optimized machine code at runtime using the industry-standard LLVM compiler library, is used to accelerate and parallelize the **for** loop when unavoidable. Taking advantage of these packages, **P3MLens** in PYTHON can approach the speeds of C or FORTRAN and be used in large ray-tracing simulations.

6. CONCLUSION

In the paper, we introduce a 2D P3M algorithm with optimized Green function and adaptive softening length for gravitational lensing studies in cosmological simulations. The algorithm yields a precise calculation for the 2D force field between particles, which is two orders of magnitude more accurate than a simple PM algorithm at a small \(r\), and has a smaller anisotropy at large \(r\). Comparing the total error \((Q)\), mean deviation \((Z)\) and anisotropy \((P)\) for different smoothing schemes in the PM, we prefer \(a_{\text{pm}} \approx 6H\) and particle shape \(S_1\). From our tests of computing \(\kappa, \gamma\) and \(\delta\mu\) for a halo of the NFW profile, we found an adaptive smoothing with \(N_{\text{nb}} = 400\) in PP can largely suppress the Poisson noise in the cosmological simulation, though the high density cores of a mass corresponding to \(N_{\text{nb}}\) are smoothed out. One may reduce the Poisson noise further by increasing \(N_{\text{nb}}\) but at the expenses of losing resolution at high density regions. This algorithm is suitable for all micro, weak and strong lensing studies.

We will release the PYTHON implementation **P3MLens** of this algorithm once the paper is published.

ACKNOWLEDGMENTS

The work is supported by NSFC (11890691, 11621303, 11533006) and by 111 project No. B20019. We thank Guoliang Li and Chengliang Wei for useful discussion about micro lensing and PM algorithm. Xu thanks Wei Tian for useful discussion about analytical calculation of integration involving Bessel functions. This work made use of the Gravity Supercomputer at the Department of Astronomy, Shanghai Jiao Tong University.

\(^{1}\)https://github.com/kunxujsjt/P3MLens

\(^{2}\)https://numby.org

\(^{3}\)https://www.scipy.org

\(^{4}\)https://numba.pydata.org
Figure 6. Comparison of the performance for simulations with different mass resolution for a NFW halo of the virial mass $M_{\text{vir}} = 10^{14}\, M_\odot/ h$. The low resolution halo has a particle mass $5 \times 10^8\, M_\odot/ h$, and the high resolution one has a particle mass $10^7\, M_\odot/ h$. The results are presented for the high resolution halo with different $N_{\text{nb}} = 400, 400\sqrt{50}, 20000$, compared with the result for the low resolution halo with $N_{\text{nb}} = 400$.

REFERENCES

Aihara, H., Arimoto, N., Armstrong, R., et al. 2018, PASJ, 70, S4, doi: 10.1093/pasj/psx066
Bartelmann, M., & Weiss, A. 1994, A&A, 287, 1.
https://arxiv.org/abs/astro-ph/9311074
Bradač, M., Schneider, P., Lombardi, M., et al. 2004, A&A, 423, 797, doi: 10.1051/0004-6361:20040168
Couchman, H. M. P. 1991, ApJL, 368, L23, doi: 10.1086/185939
Dey, A., Schlegel, D. J., Lang, D., et al. 2019, AJ, 157, 168, doi: 10.3847/1538-3881/ab089d
Dutton, A. A., & Macciò, A. V. 2014, MNRAS, 441, 3359, doi: 10.1093/mnras/stu742
Efstathiou, G., Davis, M., White, S. D. M., & Frenk, C. S. 1985, ApJS, 57, 241, doi: 10.1086/191003
Fosalba, P., Gaztanaga, E., Castander, F. J., & Manera, M. 2008, MNRAS, 391, 435, doi: 10.1111/j.1365-2966.2008.13910.x
Heymans, C., White, M., Heavens, A., Vale, C., & van Waerbeke, L. 2006, MNRAS, 371, 750, doi: 10.1111/j.1365-2966.2006.10705.x
Hockney, R. W., & Eastwood, J. W. 1981, Computer Simulation Using Particles (crc Press)
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111, doi: 10.3847/1538-4357/ab042c
Jain, B., Seljak, U., & White, S. 2000, ApJ, 530, 547, doi: 10.1086/308384
Jing, Y. P., Mo, H. J., Boerner, G., & Fang, L. Z. 1994, A&A, 284, 703. https://arxiv.org/abs/astro-ph/9308017
Li, G.-L., Mao, S., Jing, Y. P., Kang, X., & Bartelmann, M. 2006, ApJ, 652, 43, doi: 10.1086/507695
Mandelbaum, R. 2018, ARA&A, 56, 393, doi: 10.1146/annurev-astro-081817-051928
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493, doi: 10.1086/304888
Takada, M., & Jain, B. 2003a, MNRAS, 340, 580, doi: 10.1046/j.1365-8711.2003.06321.x
—. 2003b, MNRAS, 344, 857, doi: 10.1046/j.1365-8711.2003.06868.x
Takada, M., Ellis, R. S., Chiba, M., et al. 2014, PASJ, 66, R1, doi: 10.1093/pasj/pst019
Treu, T. 2010, ARA&A, 48, 87, doi: 10.1146/annurev-astro-081309-130924
Wambsganss, J., Cen, R., & Ostriker, J. P. 1998, ApJ, 494, 29, doi: 10.1086/305174
Xu, K. 2021, P3MLens: First release of P3MLens, 1.0.0, Zenodo, doi: 10.5281/zenodo.4727920
Yi-bin, Y., Guo-liang, L., & Romeo, A. B. 2009, ChA&A, 33, 121, doi: 10.1016/j.chinastron.2009.03.002