


# Adaptive hierarchical tree algorithm for gravitational lensing wave effect

XIKAI SHAN <sup>1, 2, 3</sup> AND GUOLIANG LI<sup>1</sup>

<sup>1</sup>*Purple Mountain Observatory, Chinese Academy of Sciences, Nanjing, Jiangsu, 210023, China*

<sup>2</sup>*Institute for Frontier in Astronomy and Astrophysics, Beijing Normal University, Beijing, 102206, China*

<sup>3</sup>*Department of Astronomy, Beijing Normal University, Beijing, 100875, China*

## ABSTRACT

*Keywords:* Gravitational lensing — Diffractional integral — Algorithm — Gravitational wave

### 1. INTRODUCTION

### 2. METHOD

#### 2.1. Diffraction integral

In the context of the scalar wave approximation, the gravitational wave (GW) lensing effect can be quantified using the diffraction integral (Schneider et al. 1992; Nakamura & Deguchi 1999; Takahashi & Nakamura 2003) represented by the equation:

$$F(\omega, \mathbf{y}) = \frac{2GML(1+z_L)\omega}{\pi c^3 i} \int -\infty^\infty d^2x \exp[i\omega t(\mathbf{x}, \mathbf{y})], \quad (1)$$

where  $F(\omega, \mathbf{y})$  denotes the magnification factor, and  $\omega$  and  $\mathbf{y}$  represent the GW's circular frequency and its position (normalized by the Einstein radius) in the source plane, respectively.  $M_L$  is the lens mass,  $z_L$  is the lens redshift, and  $\mathbf{x}$  is the coordinate (normalized by the Einstein radius) in the lens plane.  $t(\mathbf{x}, \mathbf{y})$  is the time delay function given by:

$$t(\mathbf{x}, \mathbf{y}) = \frac{4GML(1+z_L)}{c^3} \left[ \frac{1}{2} |\mathbf{x} - \mathbf{y}|^2 - \psi(\mathbf{x}) + \phi_m(\mathbf{y}) \right]. \quad (2)$$

Here,  $\psi(\mathbf{x})$  represents the dimensionless deflection potential, and  $\phi_m(\mathbf{y})$  is employed to correct the zero point of the time delay.

However, due to the oscillatory nature of the integrand, conventional Gaussian numerical integration methods are inadequate. To address this limitation, Ulmer & Goodman (1995) (referred to as UG95 hereafter) proposed a novel integration algorithm. In detail, this algorithm Fourier transforms the amplification factor into the time domain:

$$\tilde{F}(t, \mathbf{y}) \equiv \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \exp(-i\omega t) \frac{F(\omega, \mathbf{y})}{C_\omega}. \quad (3)$$

where  $\tilde{F}(t, \mathbf{y})$  is the time domain amplification factor and  $C_\omega$  is the coefficient before the integrand of Eq. (1):

$$C_\omega = \frac{2GML(1+z_L)\omega}{\pi c^3 i}. \quad (4)$$

Upon substituting Eq.(1) into Eq.(3), it is evident that the time domain amplification factor is proportional to the time delay probability density function (PDF):

$$\tilde{F}(t, \mathbf{y}) = \int_{-\infty}^{\infty} d^2x \delta[t(\mathbf{x}, \mathbf{y}) - t] = \frac{|dS|}{dt}, \quad (5)$$

which can be determined by calculating the surface area  $dS$  in the time delay interval  $dt$ . The subsequent step is to inversely Fourier transform  $\tilde{F}(t, \mathbf{y})$  to recover  $F(\omega, \mathbf{y})$ . One can find that this algorithm eliminates the integral oscillation problem.

One direct and easy realization method (hereafter referred to as Direct method) for the calculation of the time domain amplification factor is to pixelate the lens plane and then accumulate the area of the pixel within the same time delay interval (Diego et al. 2019). The advantage of this method is that the calculation time is independent of the number of microlenses and does not need to find the image point at first, compared with the contour integral method (Ulmer & Goodman 1995; Mishra et al. 2021). However, the Direct method needs an extremely high-resolution pixelized lens plane to avoid numerical error induced by the time delay variation in the inner region of the pixel. In other words, if the maximum time delay difference in the pixel is greater than the time delay resolution, rudely arranging the whole pixel area into one time delay interval will introduce numerical error.

To mitigate uncontrollable numerical errors and expedite the calculation, we propose a trapezoid-approximation-based adaptive hierarchical tree algorithm to more accurately compute the diffraction integral.

## 2.2. Trapezoid approximation in pixel

The time-domain amplification factor within a pixel can be evaluated through the contour integral:

$$\frac{|dS|}{dt} = \oint_C \frac{ds}{|\nabla_{\mathbf{x}} t(\mathbf{x}, \mathbf{y})|} \quad (6)$$

Here,  $ds$  represents the line element, and  $C$  is the length of the iso-time delay curve. The upper panel of Fig. 1 illustrates that when the pixel size is sufficiently small, the gradient of the time delay  $\nabla_{\mathbf{x}} t(\mathbf{x}, \mathbf{y})$  remains constant. This panel displays the gradient of the time delay (black dashed curve with an arrow, indicating the direction of the gradient) and the iso-time delay curves (black solid curves orthogonal to the dashed curve, labeled with different red tags representing different time delays ( $t_{1 \sim 4}$ )) in a square pixel.  $t_0$  is the time delay in the middle of the square pixel, and  $\alpha$  is the angle between the direction of the gradient and the diagonal of the pixel. The contour length between  $t_1$  and  $t_2$  is constant, and the contour length between  $t_1$  and  $t_3$  and between  $t_2$  and  $t_4$  monotonically decreases.

The time-domain amplification factor within the pixel is described in detail by the equation:

$$\frac{|ds|}{dt'} = \begin{cases} \text{const} \times \frac{t' - t_3}{t_1 - t_3} & (t_3 \leq t' \leq t_1) \\ \text{const} & (t_1 < t' < t_2) \\ \text{const} \times \frac{t - t_2}{t_4 - t_2} & (t_2 \leq t' \leq t_4) \\ 0 & (\text{else}) \end{cases} \quad (7)$$

where  $\text{const} = \frac{h}{\cos(\pi/4 - \alpha) \times |\nabla_{\mathbf{x}} t| \times \text{coeff}}$ . The lower panel in Fig. 1 illustrates this theoretical distribution of the time-domain amplification factor in the square pixel.

The numerical algorithm for calculating the time-domain amplification factor at  $t_0$  in the total lens plane is expressed as:

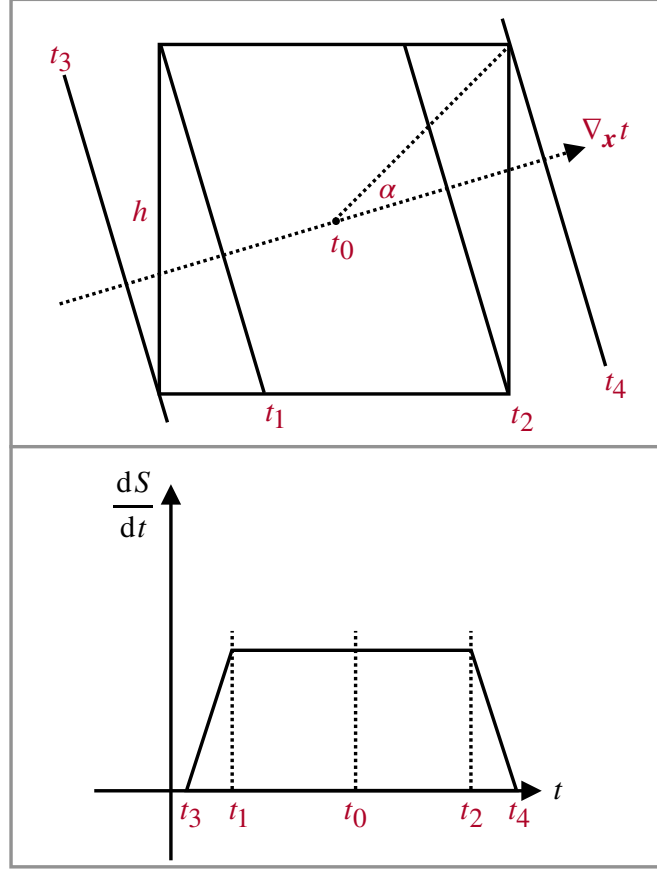
$$\frac{|dS|}{dt} \Big|_{t_0} = \frac{\sum_{i=0}^{N_{\text{pixel}}} \int_{t_0 - \Delta t/2}^{t_0 + \Delta t/2} \frac{|ds|}{dt'} \Big|_i dt'}{\Delta t} \quad (8)$$

where  $i$  represents the  $i$ th pixel,  $N_{\text{pixel}}$  is the total number of pixels in the image plane.

The accuracy of the time-domain amplification factor calculation using the method in Eq. (8) depends on the accuracy of the trapezoid approximation. In other words, it relies on the constant approximation accuracy of the gradient of the time delay within the pixel. To quantitatively describe the trapezoid approximation accuracy, we introduce an artificial precision parameter  $\epsilon$ . Thus, one can control the approximation accuracy by ensuring that the relative change rate of the gradient within the pixel  $|\Delta \nabla_{\mathbf{x}} t|/|\nabla_{\mathbf{x}} t|$  is less than  $\epsilon$ .

## 2.3. Adaptive hierarchical algorithm for diffraction integral

The time delay of microlenses embedded in the lens galaxy/galaxy cluster, influenced by macro convergence ( $\kappa$ ) and shear ( $\gamma$ ), is given by (Wambsganss 1990; Schneider et al. 1992; Chen et al. 2021):



**Figure 1.** Deflection angle perturbation in the lens plane.  $\kappa = \kappa_- = 0.6$ ,  $\gamma = 0.1$ ,  $N = 238$ , with an equal mass of  $1M_\odot$ .

$$t(\mathbf{x}, \mathbf{x}^i, \mathbf{y} = 0) = \underbrace{\frac{k}{2} ((1 - \kappa + \gamma)x_1^2 + (1 - \kappa - \gamma)x_2^2)}_{t_{\text{smooth}}(\kappa, \gamma, \mathbf{x})} - \underbrace{\left[ \frac{k}{2} \sum_i^{N_*} \ln(\mathbf{x}^i - \mathbf{x})^2 + k\phi_-(\mathbf{x}) \right]}_{t_{\text{micro}}(\mathbf{x}, \mathbf{x}^i)} \quad (9)$$

where  $k = 4GM_{\text{microlens}}(1 + z_L)/c^3$ ,  $\mathbf{x}^i$  is the coordinate of the  $i$ th microlens, and  $N_*$  is the number of microlenses. Notably, the macro image point serves as the coordinate origin ( $y = 0$ ).

The contribution from a negative mass sheet, denoted by  $\phi_-(\mathbf{x})$ , is included to ensure the total convergence  $\kappa$  remains unchanged when adding microlenses (Wambsganss 1990; Chen et al. 2021; Zheng et al. 2022).

The calculation time  $t_{\text{consume}}$  for the microlensing field diffraction integral is proportional to  $N_{\text{pixel}} \times N_*$ , where  $N_{\text{pixel}}$  is the number of pixels in the lens plane. To mitigate prolonged computation times resulting from a large pixel count, we propose an adaptive hierarchical tree algorithm based on the trapezoid approximation and the hierarchical tree algorithm introduced in Wambsganss (1990); Chen et al. (2021); Zheng et al. (2022).

This algorithm involves two fixed grids, named Level 1 and Level 2, and one adaptive grid, named Level  $n$ . Specifically, the Level 1 grid stores the indices of the microlenses, facilitating the construction of a lookup table for quick microlens identification. Here, we set the resolution of Level 1 as  $L_1 = \min L_0/10, \sqrt{L_0^2/N_*}$ , where  $L_0$  is the length of the total lens plane. This ensures that each Level 1 grid has at least one microlens when the number of microlenses is not very small.

The Level 2 grid stores the deflection angle of the far-field microlenses. As outlined in Wambsganss (1990); Chen et al. (2021); Zheng et al. (2022), the evolution of the lens potential induced by far-field microlenses within the Level 2 grid is smoothing, and the distance between these microlenses and the center of the Level 2 grid is significantly larger than the resolution of the Level 2 grid. Consequently, the time delay and deflection angle of far-field microlenses within the Level 2 grid (in other words, Level  $n$  grid) can be effectively interpolated using the stored deflection angle. Additional details are provided in Appendix A. Hence, for a more accurate grid Level  $n$  within Level 2, there is no need to recalculate the far-field contribution, and the far-field microlenses can be treated as a group. Thus, it is sufficient to only sum over the near-field microlenses to obtain the final result. This leads to a substantial reduction in the total computing time, approximating  $t'_{\text{consume}} \simeq N_{L_2} \times N_* + N_{\text{pixel}} \times N_{*,\text{near}}$ , where, in a typical microlensing system,  $N_{L_2} \ll N_{\text{pixel}}$  and  $N_{*,\text{near}} \ll N_*$ .

The resolution of the Level 2 grid used is the same as in Zheng et al. (2022), which is  $L_2 = L_1/20$ . Far-field microlenses are the lenses outside the nearest nine  $L_1$  grids surrounding the  $L_1$  grid in which the  $L_2$  grid resides.

The Level  $n$  grid is an adaptive refined grid within Level 2. As established in section 2.2, the time domain amplification factor within the pixel can be approximated by a trapezoid-shaped distribution. The accuracy of this approximation relies on the relative changing rate of the gradient of the time delay. If this relative changing rate  $|\Delta \nabla_{\mathbf{x}} t|/|\bar{\nabla}_{\mathbf{x}} t|$  is less than a tiny value  $\epsilon$ , the approximation is accurate. Therefore, based on this property, we propose an adaptive refinement algorithm for the Level  $n$  grid. In detail, we set a precision parameter  $\epsilon$  and calculate  $e_2 = |\Delta \nabla_{\mathbf{x}} t|/|\bar{\nabla}_{\mathbf{x}} t|$  in the Level 2 grid. If  $e_2 < \epsilon$ , the trapezoid approximation in the Level 2 grid is accurate enough; otherwise, we reduce the grid length by half, and calculate  $e_n = |\Delta \nabla_{\mathbf{x}} t|/|\bar{\nabla}_{\mathbf{x}} t|$  in each subpixel (where  $n > 2$  is the resolution level number, and the number of  $e_n$  is  $4^{n-2}$ ). The refinement stops when all  $e_n$  are less than  $\epsilon$ . Here, we set a maximum resolution level number as  $n_{\text{max}}$  to avoid infinite refinement at the microlensing image point and the location of microlenses.

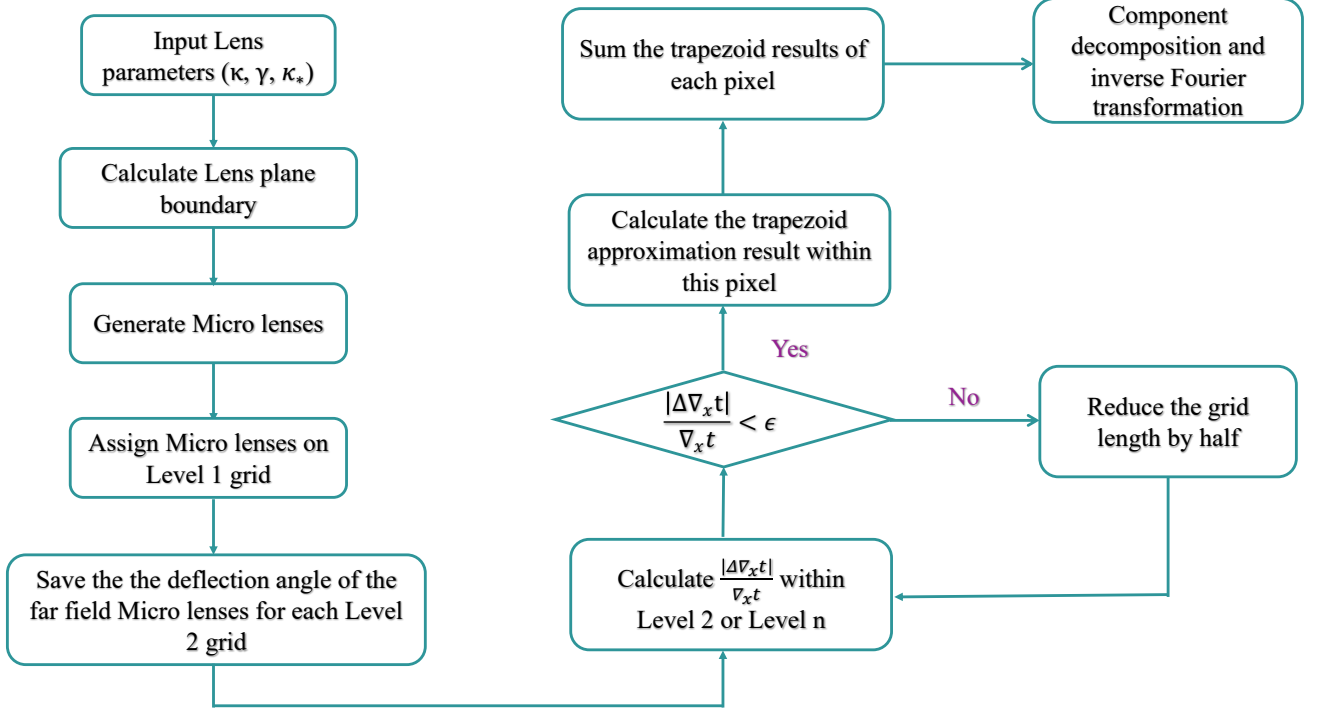
The completed adaptive hierarchical tree algorithm has now been constructed. In Fig. 2, we present the flow chart of the algorithm for calculating the diffraction integral of the microlensing field, encompassing the following steps:

- Input lens parameters, including macro lensing  $\kappa$ ,  $\gamma$ , and micro lensing  $\kappa_*$ .
- Use the method introduced in Shan et al. (2023) to calculate the microlensing field boundary  $L_0$ .
- Randomly place  $N_* = \kappa_* \times L_0^2/\pi$  microlenses within the microlensing field boundary.
- Set Level 1 grid with a length  $L_1 = \min\{L_0/10, \sqrt{L_0^2/N_*}\}$ , and assign each microlens to a Level 1 grid.
- Set Level 2 grid with a length  $L_2 = L_1/20$ , and store the deflection angle contributed by the far-field microlenses at the four vertices of the Level 2 grid.
- Calculate  $|\Delta \nabla_{\mathbf{x}} t|/|\bar{\nabla}_{\mathbf{x}} t|$  within the Level 2 or Level  $n$  grid. If  $|\Delta \nabla_{\mathbf{x}} t|/|\bar{\nabla}_{\mathbf{x}} t| < \epsilon$  or  $n > n_{\text{max}}$ , proceed to the next step; otherwise, reduce the grid length by half and  $n = n + 1$ .
- Calculate the trapezoid approximation result within the Level  $n$  pixel using Eq. 7.
- Sum over the trapezoid approximation results at all Level  $n$  grids using Eq. 8.
- Use the Component Decomposition algorithm proposed in Shan et al. (2023) to recover the frequency domain amplification factor  $F(\omega)$ .

### 3. RESULT

#### 3.1. Algorithm test using analytical result

In this section, we evaluate the performance of our trapezoid-approximation-based adaptive hierarchical tree algorithm by comparing its results with analytical solutions. We consider three distinct macro-lensing scenarios, and the corresponding macro and micro-lensing parameters are provided in Table 1. Specifically, for Type I and Type III macro-lensing images, we set the shear ( $\gamma$ ) value to 0. This choice is made because when a micro lens is embedded in external convergence without external shear, the diffraction has an analytical resolution described by:



**Figure 2.** Flow chart of the trapezoid-approximation-based adaptive hierarchical tree algorithm.

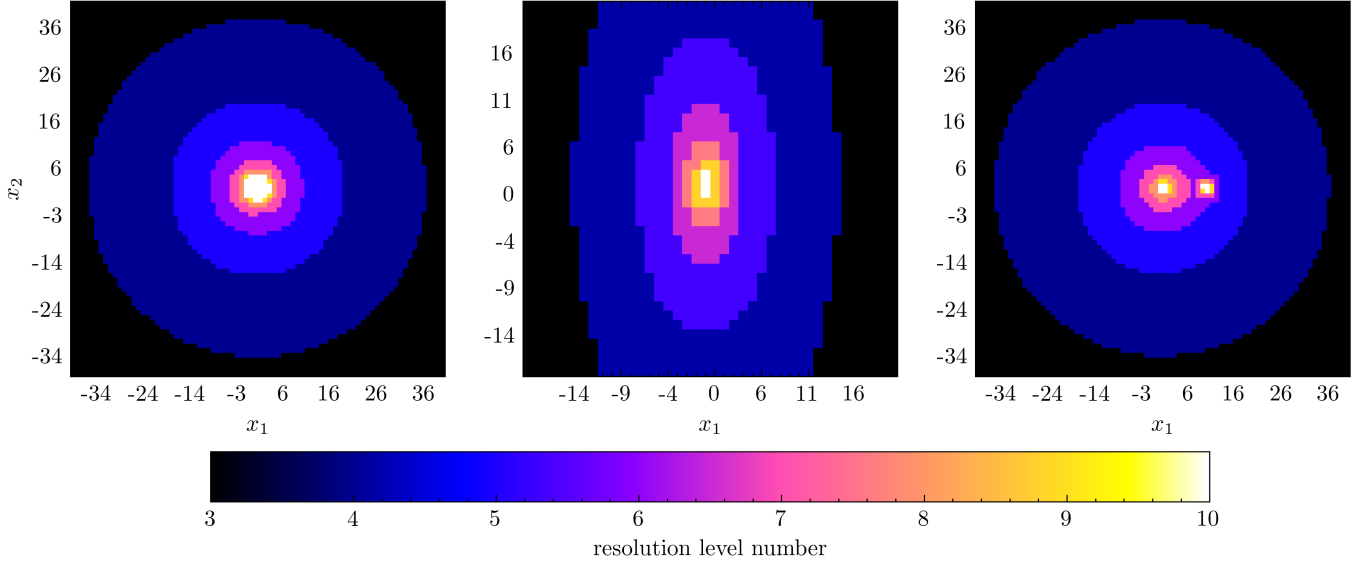
**Table 1.** This table listed the macro and micro parameters for algorithm testing.  $\kappa$  and  $\gamma$  are macro lensing convergence and shear, Micro  $x_1$  and Micro  $x_2$  are micro lens coordinate. Here, we set the mass of microlens as  $100 M_\odot$  for Type I and Type III scenario.

	$\kappa$	$\gamma$	Micro $x_1$	Micro $x_2$
Type I	0.7	0	0.1	0
Type II	0.875	0.325	none	none
Type III	1.3	0	10	0

$$\begin{aligned}
 F(w, \mathbf{x}) = & -\frac{1}{\lambda^2 w} 2^{-2-\frac{iw}{2}} |w|^{-1+\frac{iw}{2}} \Gamma\left(1 - \frac{iw}{2}\right) |\lambda|^{-1+\frac{iw}{2}} e^{\frac{iw}{2}(\phi_m + \lambda(x_1^2 + x_2^2))} \left((w + 2i) |\lambda|^3 |w|^3 (x_1^2 + x_2^2)\right. \\
 & \left. \left(\sinh\left(\frac{\pi w}{4}\right) \operatorname{sgn}(\lambda w) + \cosh\left(\frac{\pi w}{4}\right)\right) {}_2F_3\left(1 - \frac{iw}{4}, \frac{3}{2} - \frac{iw}{4}; 1, \frac{3}{2}, \frac{3}{2}; -\frac{1}{16} \lambda^2 w^2 (x_1^2 + x_2^2)^2\right)\right. \\
 & \left. - 4\lambda^2 w^2 \left(\cosh\left(\frac{\pi w}{4}\right) \operatorname{sgn}(\lambda w) + \sinh\left(\frac{\pi w}{4}\right)\right) {}_2F_3\left(\frac{1}{2} - \frac{iw}{4}, 1 - \frac{iw}{4}; \frac{1}{2}, \frac{1}{2}, 1; -\frac{1}{16} \lambda^2 w^2 (x_1^2 + x_2^2)^2\right)\right) \quad (10)
 \end{aligned}$$

For Type II macro-lensing images, we exclusively test our algorithm under a pure macro-lensing scenario. This choice is deliberate since logarithmic divergence is Type II's most important characteristic, and there are not analytical solutions when any micro lens is embedded.

Fig. 3 illustrates the resolution level number  $n$  (Level  $n$ ) in the lens plane for three different scenarios, with Type I on the left, Type II in the middle, and Type III on the right. Here, we set  $L_2 = 1.024$ ,  $\epsilon = 0.01$ , and  $n_{\max} = 12$ , resulting in a minimum pixel length of  $L_{12} = 0.001$ . In this figure, the inner region exhibits a higher resolution number, primarily due to macro lensing effects, as  $|\Delta \nabla_{\mathbf{x}} t| / |\nabla_{\mathbf{x}} t| \propto 1/|\mathbf{x}|$ , where  $|\mathbf{x}|$  is the distance from the center. Additionally, the third panel shows that at the micro lens points, the resolution level is very high, as this region is dominated by micro lensing effects and  $|\Delta \nabla_{\mathbf{x}} t| / |\nabla_{\mathbf{x}} t| \propto 1/|\mathbf{x} - \mathbf{x}_{\text{micro}}|$ , where  $\mathbf{x}_{\text{micro}}$  is the coordinate of the micro lenses.



**Figure 3.** Resolution level number  $n$  in the lens plane. Columns from left to right is Type I, Type II and Type III scenarios listed in Table. 1.

Fig. 4 illustrates the results of the diffraction integral for the three micro + macro/pure macro lensing configurations. Each column corresponds to a different macro lensing image, with Type I on the left, Type II in the middle, and Type III on the right. The top two rows display the absolute amplification factor and its relative error between numerical and analytical results. The bottom two rows exhibit the phase of the amplification factor and its error between numerical and analytical results. The blue curves represent our trapezoid-approximation-based adaptive hierarchical tree method, while the red curves represent the Direct Adding method. For the Direct Adding method, the resolution at the lens plane was set to 0.001, equivalent to the grid length of the maximum resolution level of the trapezoid-approximation-based adaptive hierarchical tree method. It is evident that the precision of our method is comparable to or more accurate than the Direct Adding method but requires less time due to the use of fewer pixels at the lens plane. The acceleration rate is approximately two orders of magnitude. However, we do not emphasize this number here, as this is a simple case primarily intended to test our method's accuracy.

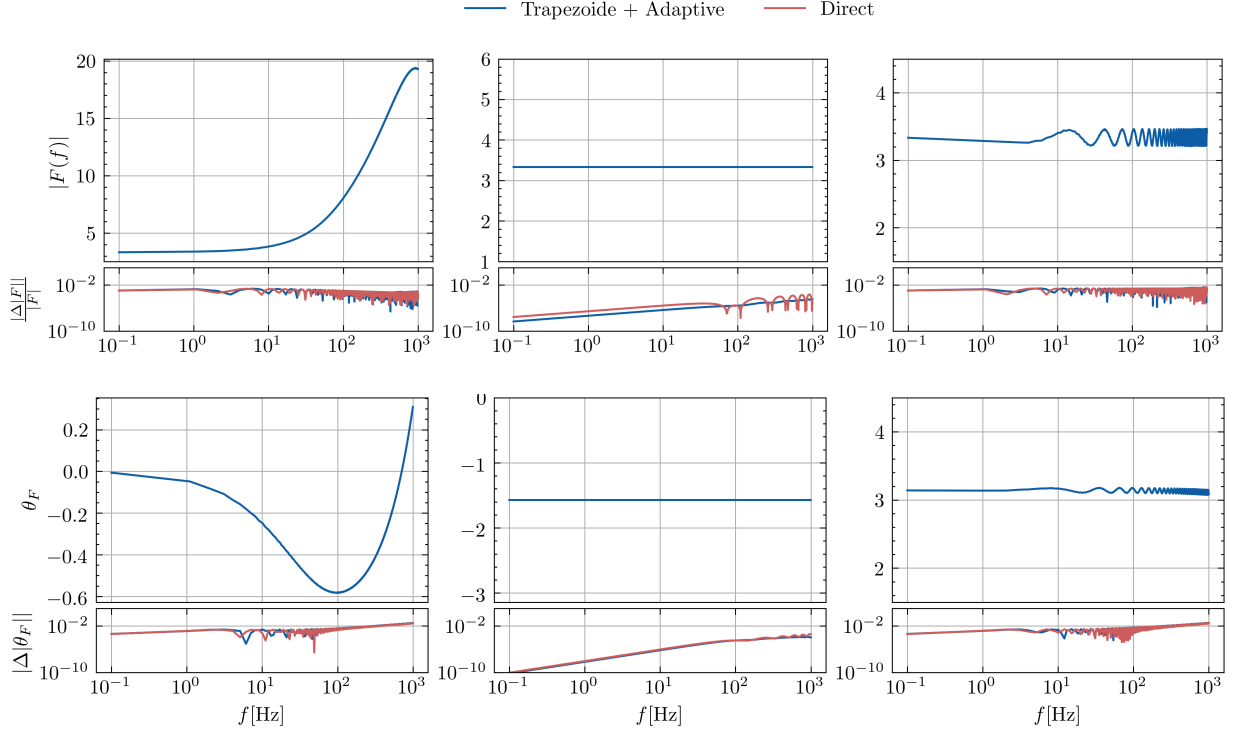
### 3.2. Adaptive Hierarchical tree results for microlensing field

In this section, we present the results of the diffraction integral for microlensing field scenarios, considering three different macro lensing image types: Type I, Type II, and Type III. The parameters used in the calculations are listed in Table.2. The lens plane boundary  $L_0$  is determined using the method introduced in Shan et al. (2023), and this method is retained in Appendix B. We choose a precision parameter  $\epsilon = 0.1$  and set the maximum grid level  $n_{\max} = 9$  (resulting in a resolution of  $L_9 = L_2/128$ ). This resolution ensures a two-order-of-magnitude higher resolution than Level 2 grid and a three-order-of-magnitude higher resolution than Level 1 grid.

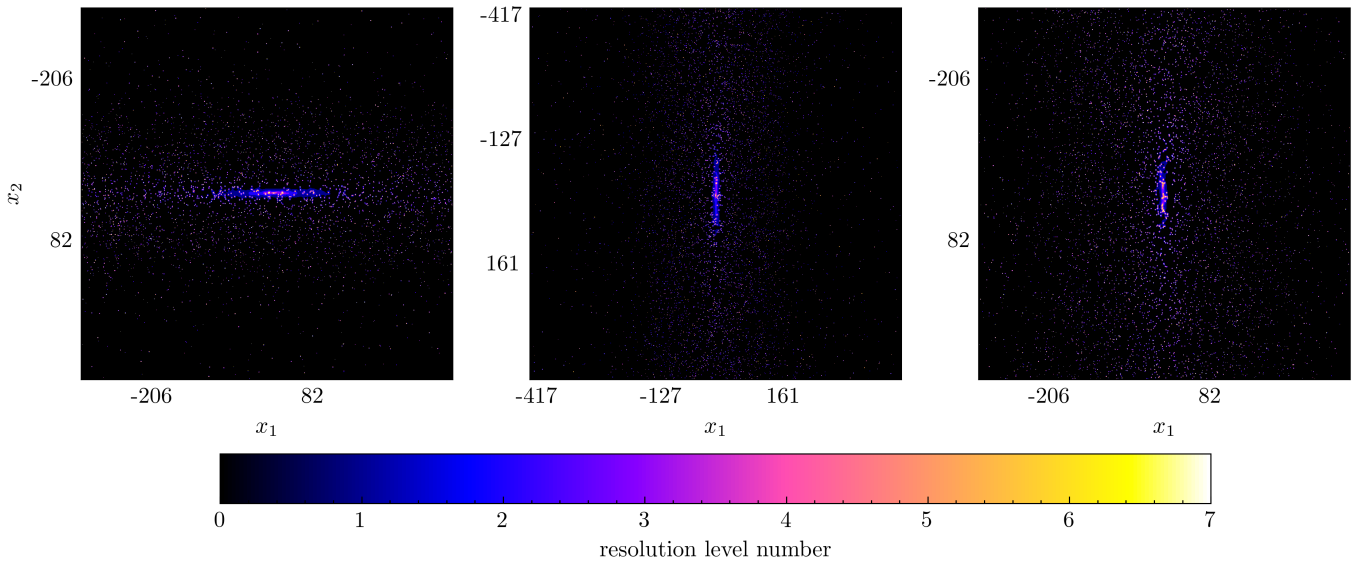
Figure 5 displays the resolution levels for these three macro + micro lensing configurations. Most grids are at Level 2 resolution (black region), with only a small part of the region requiring higher resolution. Consequently, our trapezoid-approximation-based adaptive hierarchical algorithm significantly accelerates the calculation speed. After counting the total number of pixels used in our algorithm and comparing it with the Direct Adding algorithm (with uniform resolution of  $L_2/128$ ), we find that our method achieves a speedup of two orders of magnitude.

Figure 6 compares the diffraction integral results of our trapezoid-approximation-based adaptive hierarchical algorithm with the Direct Adding method. Different columns represent different macro lensing image types, with Type I on the left, Type II in the middle, and Type III on the right. The top (bottom) two rows depict the absolute (phase) value of the amplification factor using our method and the differences between our method and the Direct Adding method. The differences between these two algorithms are less than 1%. Therefore, we conclude that our method can accelerate the diffraction integral by two orders of magnitude while maintaining a numerical error of less than 1%.

Fig4. adaptive result compare with uniform result (calculation time, resolution). Interpolation acceleration, adaptive hierarchical acceleration. In this step, more accelerate two order of magnitude.



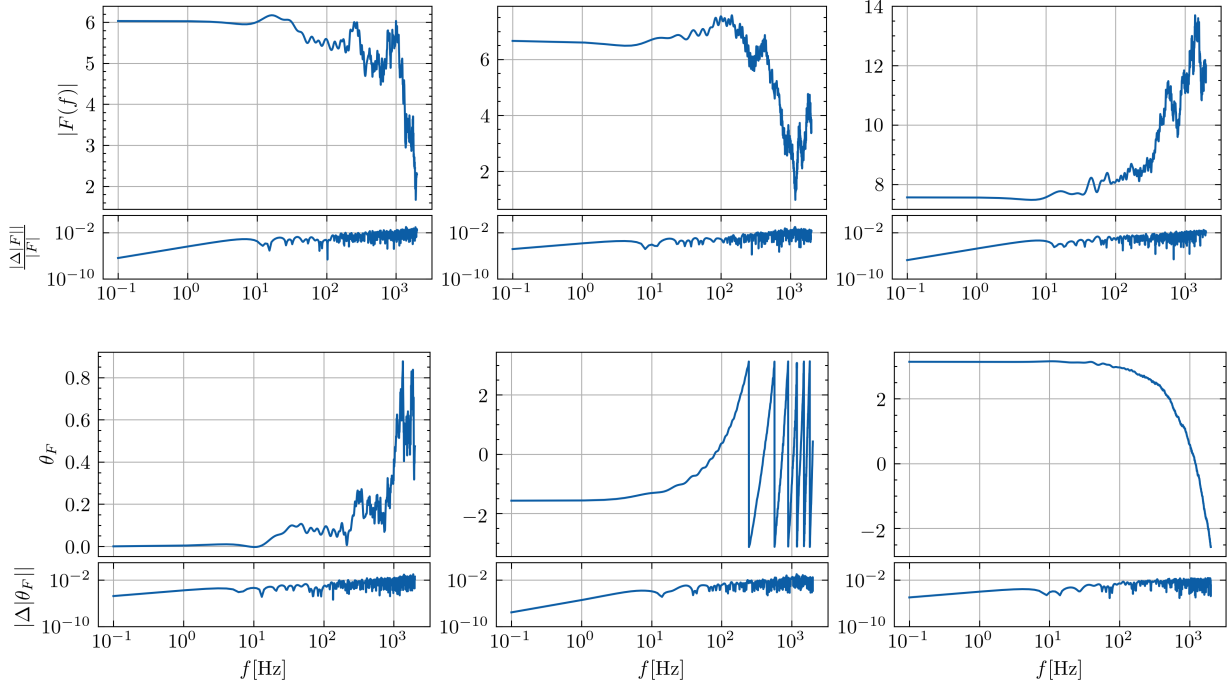
**Figure 4.** Diffraction integral result for tree micro + macro or pure macro lensing scenarios. Each column corresponds to a different macro lensing image, with Type I on the left, Type II in the middle, and Type III on the right. The top two rows display the absolute amplification factor and its relative error between numerical and analytical results. The bottom two rows show the phase of the amplification factor and its error between numerical and analytical results. The blue curves represent our trapezoid-approximation-based adaptive hierarchical tree method, while the red curves represent the Direct Adding method.



**Figure 5.** Resolution level numbers for microlensing field. Columns from left to right represent Type I, Type II, and Type III scenarios.

**Table 2.** This table lists the lens parameter values of Type I, Type II, and Type III macro images used in simulations.  $\kappa$  and  $\gamma$  are the macro convergence and shear.  $\kappa_*$  is the microlens convergence.  $L_0$  is the lens plane boundary.  $\epsilon$  is the precision parameter. Here, we set the mass of the microlens as  $1 \text{ M}_\odot$ .

Parameter	$\kappa$	$\gamma$	$\kappa_*$	$L_0$	$\epsilon$
Type I	0.7	-0.25	0.06	333.3	0.1
Type II	0.8	0.25	0.06	445.3	0.1
Type III	1.2	-0.15	0.06	333.3	0.1



**Figure 6.** Diffraction integral results for the microlensing field. Columns from left to right represent Type I, Type II, and Type III macro lensing scenarios. The top (bottom) two rows are the absolute (phase) value of the amplification factor using our trapezoid-approximation-based adaptive hierarchical tree algorithm and the differences between our method and the Direct Adding method.

## APPENDIX

### A. APPENDIX INFORMATION

#### REFERENCES

- 170 Chen, X., Shu, Y., Li, G., & Zheng, W. 2021, The  
 171 Astrophysical Journal, 923, 117,  
 172 doi: [10.3847/1538-4357/ac2c76](https://doi.org/10.3847/1538-4357/ac2c76)
- 173 Diego, J. M., Hannuksela, O. A., Kelly, P. L., et al. 2019,  
 174 Astronomy & Astrophysics, 627, A130,  
 175 doi: [10.1051/0004-6361/201935490](https://doi.org/10.1051/0004-6361/201935490)



176 Mishra, A., Meena, A. K., More, A., Bose, S., & Bagla,  
 177 J. S. 2021, Monthly Notices of the Royal Astronomical  
 178 Society, 508, 4869–4886, doi: [10.1093/mnras/stab2875](https://doi.org/10.1093/mnras/stab2875)  
 179 Nakamura, T. T., & Deguchi, S. 1999, Progress of  
 180 Theoretical Physics Supplement, 133, 137,  
 181 doi: [10.1143/PTPS.133.137](https://doi.org/10.1143/PTPS.133.137)  
 182 Schneider, P., Ehlers, J., & Falco, E. E. 1992, in  
 183 Gravitational Lenses (Springer), 467–515  
 184 Schneider, P., Ehlers, J., & Falco, E. E. 1992, Gravitational  
 185 Lenses, doi: [10.1007/978-3-662-03758-4](https://doi.org/10.1007/978-3-662-03758-4)

186 Shan, X., Li, G., Chen, X., Zheng, W., & Zhao, W. 2023,  
 187 Sci. China Phys. Mech. Astron., 66, 239511,  
 188 doi: [10.1007/s11433-022-1985-3](https://doi.org/10.1007/s11433-022-1985-3)  
 189 Takahashi, R., & Nakamura, T. 2003, The Astrophysical  
 190 Journal, 595, 1039–1051, doi: [10.1086/377430](https://doi.org/10.1086/377430)  
 191 Ulmer, A., & Goodman, J. 1995, The Astrophysical  
 192 Journal, 442, 67, doi: [10.1086/175422](https://doi.org/10.1086/175422)  
 193 Wambsganss, J. 1990, PhD thesis, -  
 194 Zheng, W., Chen, X., Li, G., & Chen, H.-z. 2022, An  
 195 Improved GPU-Based Ray-Shooting Code For  
 196 Gravitational Microlensing, arXiv,  
 197 doi: [10.48550/ARXIV.2204.10871](https://doi.org/10.48550/ARXIV.2204.10871)