An improved dynamical Poisson equation solver for self-gravity

Maeda et al. 2024, MNRAS, 527, 471

Tohoku Univ.
Ryunosuke Maeda

Collaborator

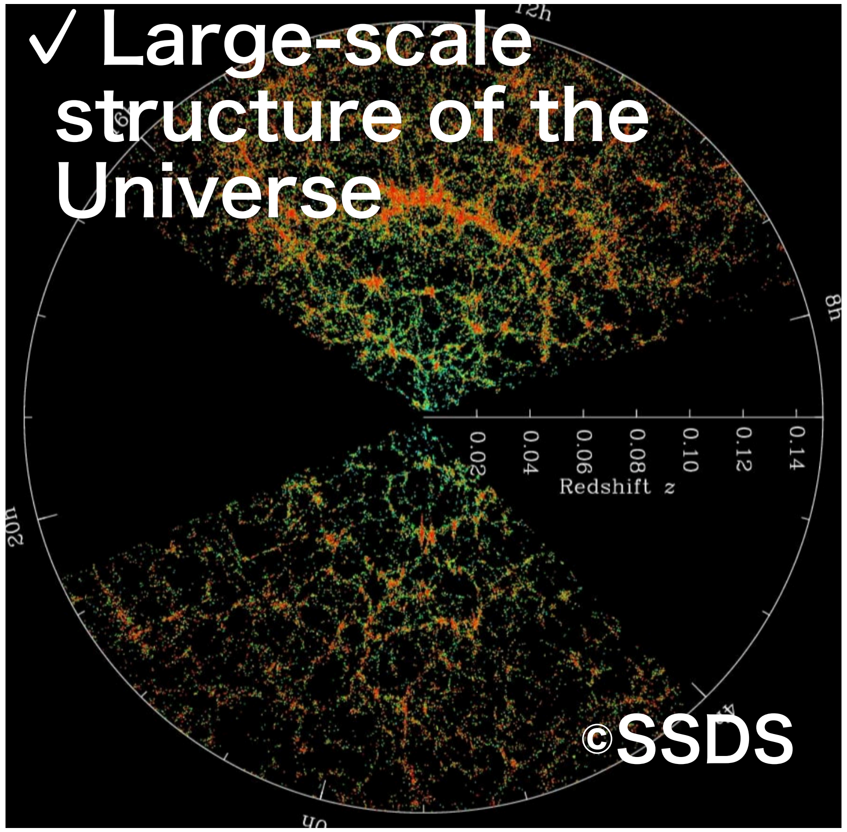
Tsuyoshi Inoue (Konan Univ.)
Shu-ichiro Inutsuka (Nagoya Univ.)

EANAM10 2025.09.15

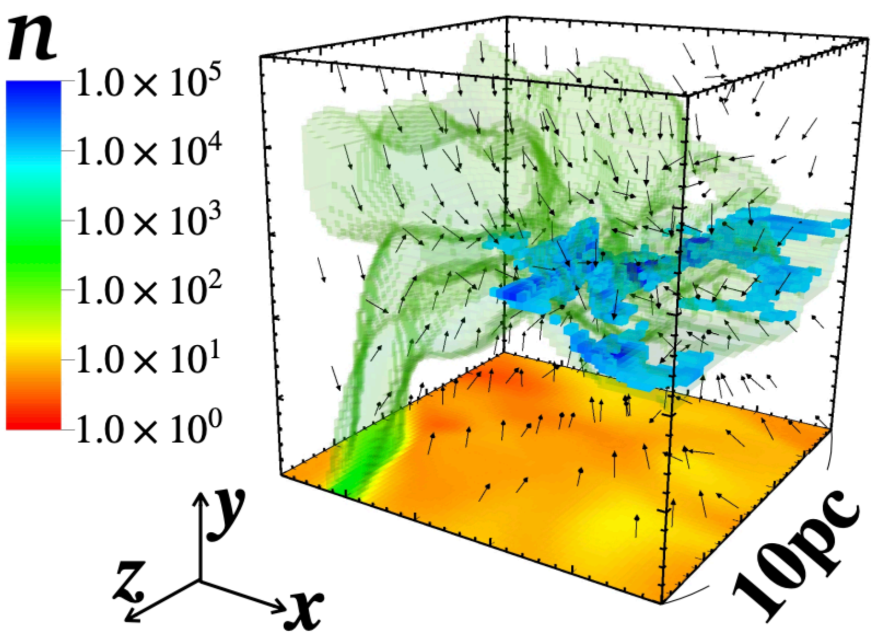
Importance of Self-gravity for Astrophysics

scale

✓ Large-scale structure of the Universe

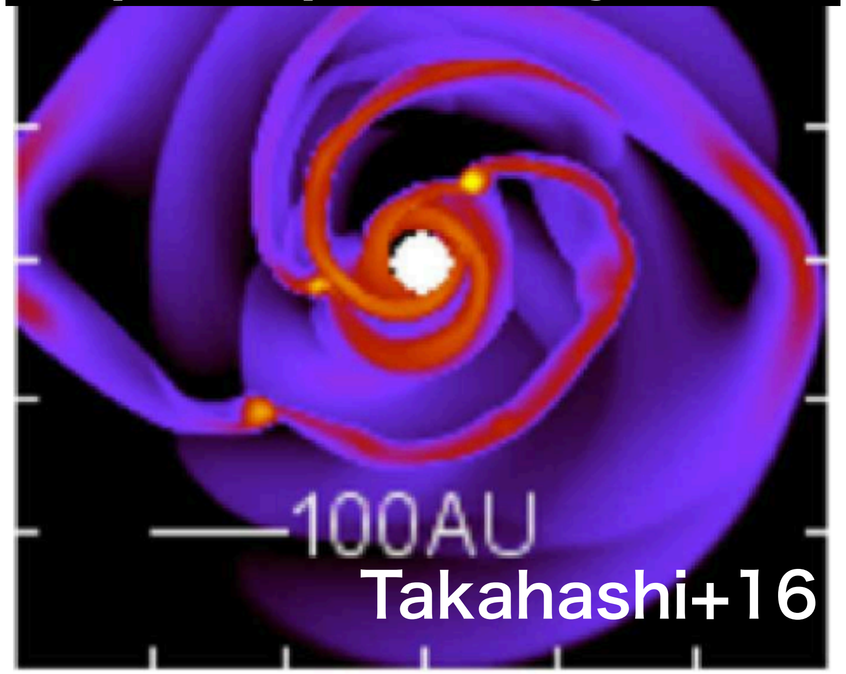


✓ Star cluster formation



Maeda+21, 24a

✓ Planet formation in protoplanetary disks



- ✓ Understanding the role of self-gravity across various scales is key to comprehending the Universe.
- ✓ Numerical simulations have been used to elucidate the complex effects of self-gravity.
- ✓ Higher-resolution (massive parallel) simulations help better understand of the Universe.

A gravitational solver with high parallelization efficiency is essential for large-scale simulations.

Self-Gravitational Solver: Direct Method

- Numerical methods for the Poisson equation for self-gravity

$$\nabla^2 \Phi = 4\pi G \rho$$

Self-Gravitational Solver: Direct Method

» Numerical methods for the Poisson equation for self-gravity

$$\nabla^2 \Phi = 4\pi G \rho$$

differentiation

$$\frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta x^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2} + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{\Delta z^2} = 4\pi G \rho_{i,j,k}$$

$$\downarrow$$

$$A \begin{pmatrix} \phi_{1,1,1} \\ \vdots \\ \phi_{i,j,k} \\ \vdots \end{pmatrix} = 4\pi G\, \Delta^2 \begin{pmatrix} \rho_{1,1,1} \\ \vdots \\ \rho_{i,j,k} \\ \vdots \end{pmatrix}$$

Direct method (ex, LU decomposition, Gaussian elimination)

✓ Matrix inversion (solving as a system of linear equations)

✓ In 3D, the computational time and memory requirements become enormous, making it impractical.

The direct method is challenging for 3D numerical simulations

->The Poisson equation is solved by slightly modifying the original equation.

Self-Gravitational Solver: Iterative Method

» Numerical methods for the Poisson equation for self-gravity

$$\frac{\partial}{\partial t}\Phi - \nabla^2\Phi = -4\pi G\rho$$

(Steady state: $\nabla^2\Phi = 4\pi G\rho$)

Iterative method (ex, Jacobi, Gauss–Seidel, Multigrid method)

✓ Iterate until it reaches a steady state.

✓ Since it is a diffusion process, the convergence of long wavelengths is slow. ($t \propto L^2$: diffusion)

Self-Gravitational Solver: Iterative Method

» **Numerical methods for the Poisson equation for self-gravity**

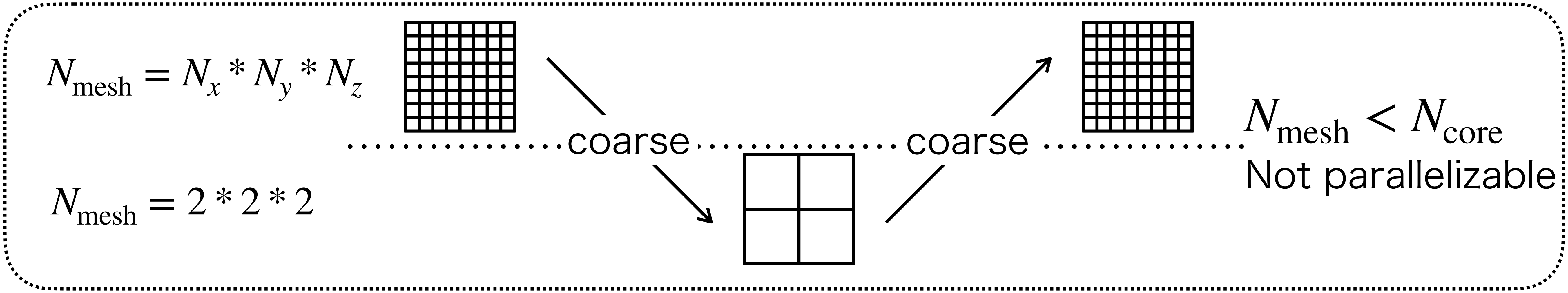
$$\frac{\partial}{\partial t}\Phi - \nabla^2\Phi = -4\pi G\rho$$

(Steady state: $\nabla^2\Phi = 4\pi G\rho$)

Multigrid method

✓ Using coarse resolution to speed up calculations (Multigrid).

✓ In the multigrid method, when one mesh becomes smaller than one core, parallelization breaks down, creating a bottleneck.



$N_{\mathrm{mesh}} = N_x * N_y * N_z$

$N_{\mathrm{mesh}} = 2*2*2$

$N_{\mathrm{mesh}} < N_{\mathrm{core}}$

Not parallelizable

Parallelization efficiency becomes poor at about 10,000 cores. (Nakajima 2012, 2014)

Self-Gravitational Solver: Hyperbolic Method

» **Numerical methods for the Poisson equation for self-gravity**

$$\frac{1}{c_{\mathrm{g}}^2}\frac{\partial^2}{\partial t^2}\Phi - \nabla^2\Phi = -4\pi G\rho$$

(Steady state: $\nabla^2\Phi = 4\pi G\rho$)

Hyperbolic solver (Hirai et al. 2016)

✓ A sufficiently large c_{g} must be used compared to the fluid velocity.

✓ Parallelization efficiency is high due to the nature of hyperbolic equations.

✓ The error is large for periodic boundary conditions due to the lack of wave decaying.

Hyperbolic methods have high parallel efficiency, but the waves do not dissipate, leading to large errors.

New Solver: Telegraph Type

» **Numerical methods for the Poisson equation for self-gravity**

$$\frac{1}{c_{\mathrm{g}}^2}\frac{\partial^2}{\partial t^2}\Phi + \frac{2\kappa}{c_{\mathrm{g}}^2}\frac{\partial \Phi}{\partial t} - \nabla^2\Phi = -4\pi G\rho$$

(Steady state: $\nabla^2\Phi = 4\pi G\rho$)

c_{g} Phase velocity of gravity

κ Diffusion coefficient

Telegraph solver (Maeda+24b)

✓ Parallelization efficiency is high due to the hyperbolic equation.

✓ Diffusion helps reduce errors even under periodic boundary conditions.

✓ κ represents how far a wave propagates before diffusion takes effect. Which κ is best?

$$\left(\frac{\partial^2}{\partial t^2}\Phi \Big/ 2\kappa\frac{\partial \Phi}{\partial t} \sim \frac{c_{\mathrm{g}}}{L}\Big/\kappa\right)$$

This method achieves both high parallel efficiency and low error.

This Work

$$\frac{1}{c_g^2}\frac{\partial^2}{\partial t^2}\Phi + \frac{2\kappa}{c_g^2}\frac{\partial \Phi}{\partial t} - \nabla^2\Phi = -4\pi G\rho$$

→High parallelization efficiency & low error for periodic boundary conditions.

Verify the usefulness of this method

» Test for usefulness of telegraph method

✓ Converge test

Does it converge to its analytical solution when a static density field is applied?

✓ Time evolution test

Can errors in the gravitational field be suppressed in a time-evolving density field?

✓ Performance test

Is the parallelization performance of our method good?

Numerical Methods

$$\frac{1}{c_{\mathrm{g}}^2}\frac{\partial^2}{\partial t^2}\Phi + \frac{2\kappa}{c_{\mathrm{g}}^2}\frac{\partial \Phi}{\partial t} - \nabla^2\Phi = -4\pi G\rho$$

≫ The equation is split as follows:

① $$\frac{\partial \Phi}{\partial t} = -c_{\mathrm{g}}\frac{\partial \Phi}{\partial x} - c_{\mathrm{g}}\frac{\partial \Phi}{\partial y} - c_{\mathrm{g}}\frac{\partial \Phi}{\partial z} + \kappa(\Psi - \Phi)$$

② $$\begin{aligned}\frac{\partial \Psi}{\partial t} = {} & c_{\mathrm{g}}\frac{\partial \Psi}{\partial x} + c_{\mathrm{g}}\frac{\partial \Psi}{\partial y} + c_{\mathrm{g}}\frac{\partial \Psi}{\partial z} + \kappa(\Phi - \Psi) \\ & -\frac{1}{\kappa}2c_{\mathrm{g}}^2\frac{\partial}{\partial x}\frac{\partial}{\partial y}\Phi - \frac{1}{\kappa}2c_{\mathrm{g}}^2\frac{\partial}{\partial y}\frac{\partial}{\partial z}\Phi - \frac{1}{\kappa}2c_{\mathrm{g}}^2\frac{\partial}{\partial z}\frac{\partial}{\partial x}\Phi - \frac{4\pi G c_{\mathrm{g}}^2}{\kappa}\rho\end{aligned}$$

c_{g}	Phase velocity of gravity
κ	Diffusion coefficient

🟨 MUSCL method Yamamoto & Daiguji (1993)

🟦 Piecewise exact solution Inoue & Inutsuka (2008)

🟥 Explicit solver ··········> **Operator splitting**

Setups: Convergence of Gravitational Fields

➢ Initial conditions

- Initial gravitational field $\Phi_{\mathrm{init}} = 0.0$
- Density distributions

  Uniform sphere $r = 0.04,\ 0.2,\ 0.45$

  Dipole, Quadrupole
- Diffusion coefficients

  $\tilde{\kappa} = 5.0 \times 10^{-6},\ 5.0 \times 10^{-5},\ 0.5,\ 1.0,\ 1.7,\ 2.5,\ 3.3,\ 5.0,\ 10$
- Resolutions

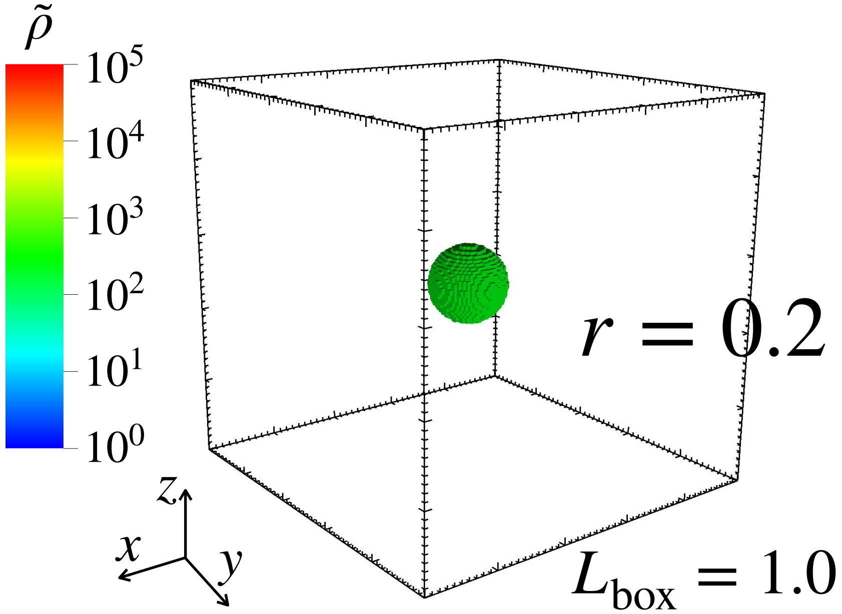
  $64^3,\ 128^3,\ 256^3$ cells

➢ Boundary conditions

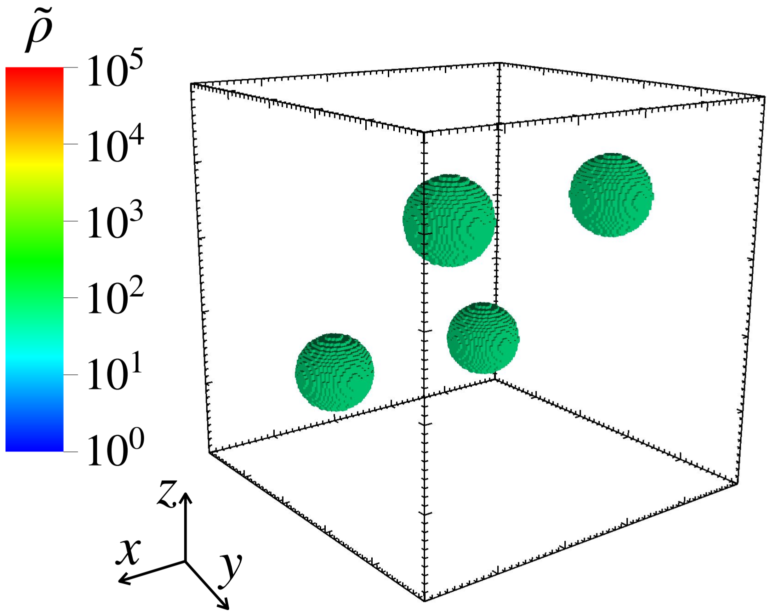
$$\Phi = \Phi_{\mathrm{exa}}$$

$$\Psi = \frac{1}{\kappa}\left(c_{\mathrm{g}}\frac{\partial \Phi_{\mathrm{exa}}}{\partial x} + c_{\mathrm{g}}\frac{\partial \Phi_{\mathrm{exa}}}{\partial y} + c_{\mathrm{g}}\frac{\partial \Phi_{\mathrm{exa}}}{\partial z}\right) + \Phi_{\mathrm{exa}}$$

Uniform sphere

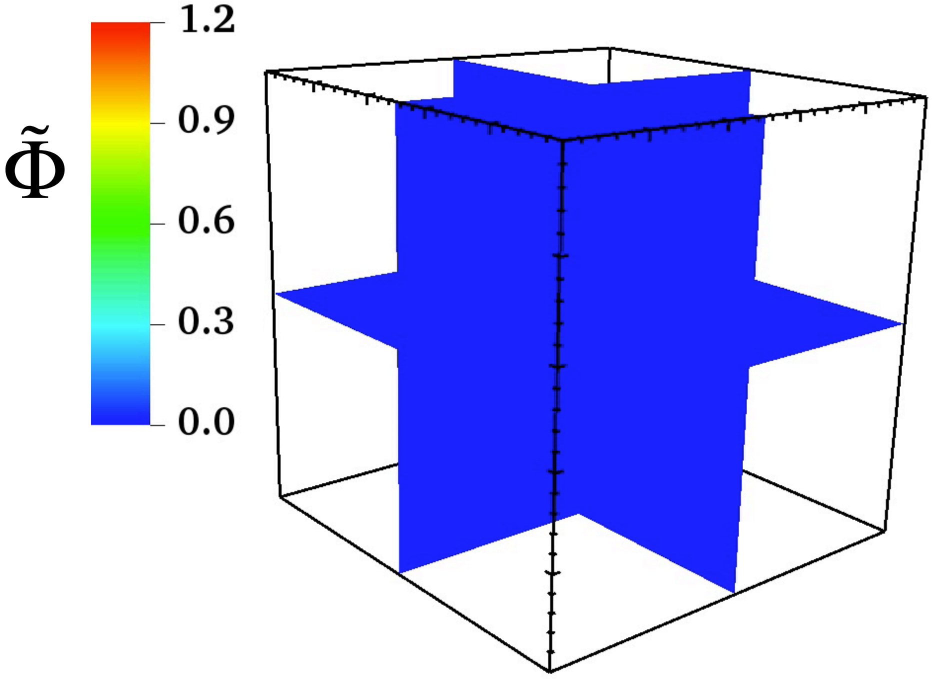


Quadrupole

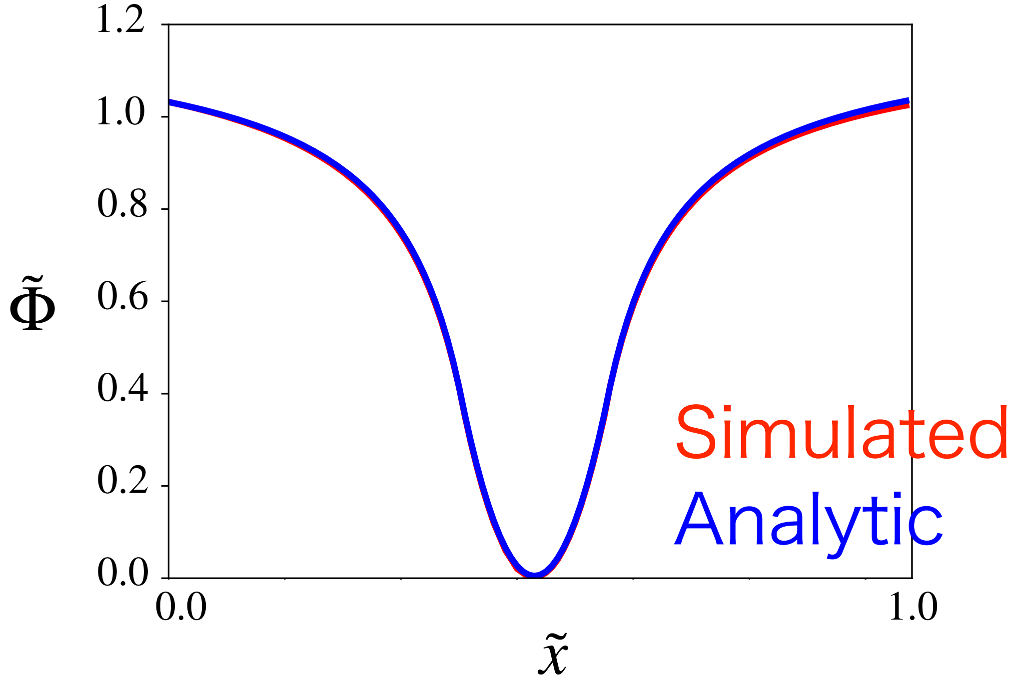


Convergence to Analytic Solution

- Evolution of Gravitational field



Converged field (1D)



Converged gravity field agrees with analytical solution

✓ Parameters

Non-dimensional telegraph equation

$$\frac{\partial^2\tilde{\Phi}}{\partial\tilde{t}^2} + 2\tilde{\kappa}\frac{\partial\tilde{\Phi}}{\partial\tilde{t}} - \tilde{\nabla}^2\tilde{\Phi} = -\tilde{\rho}$$

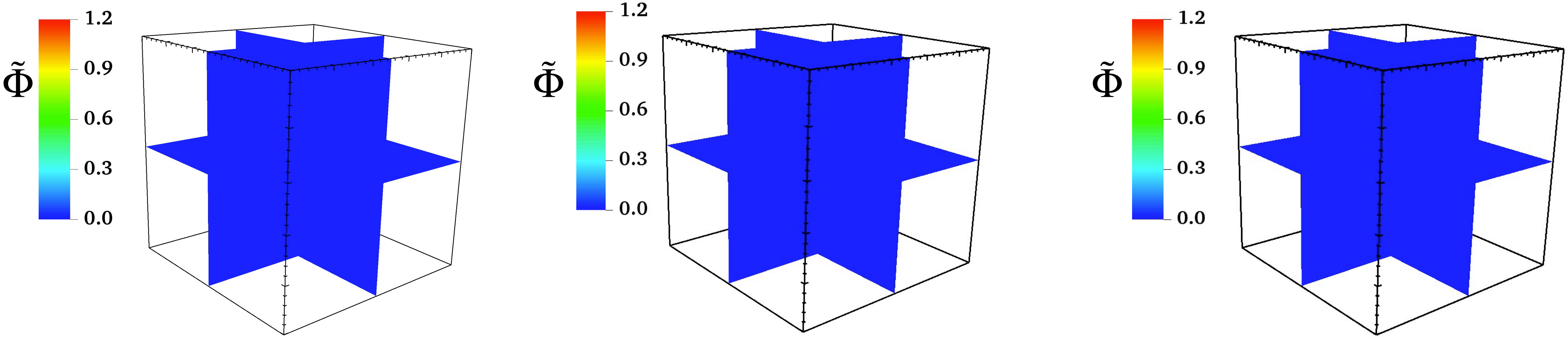
$\tilde{\kappa} = \kappa L/c_{\mathrm{g}}$ $\quad \tilde{t} = tc_{\mathrm{g}}/L$ $\quad \tilde{x} = x/L$

$\tilde{\rho} = \rho L^3/M$ $\quad \tilde{\Phi} = \Phi L/4\pi GM$

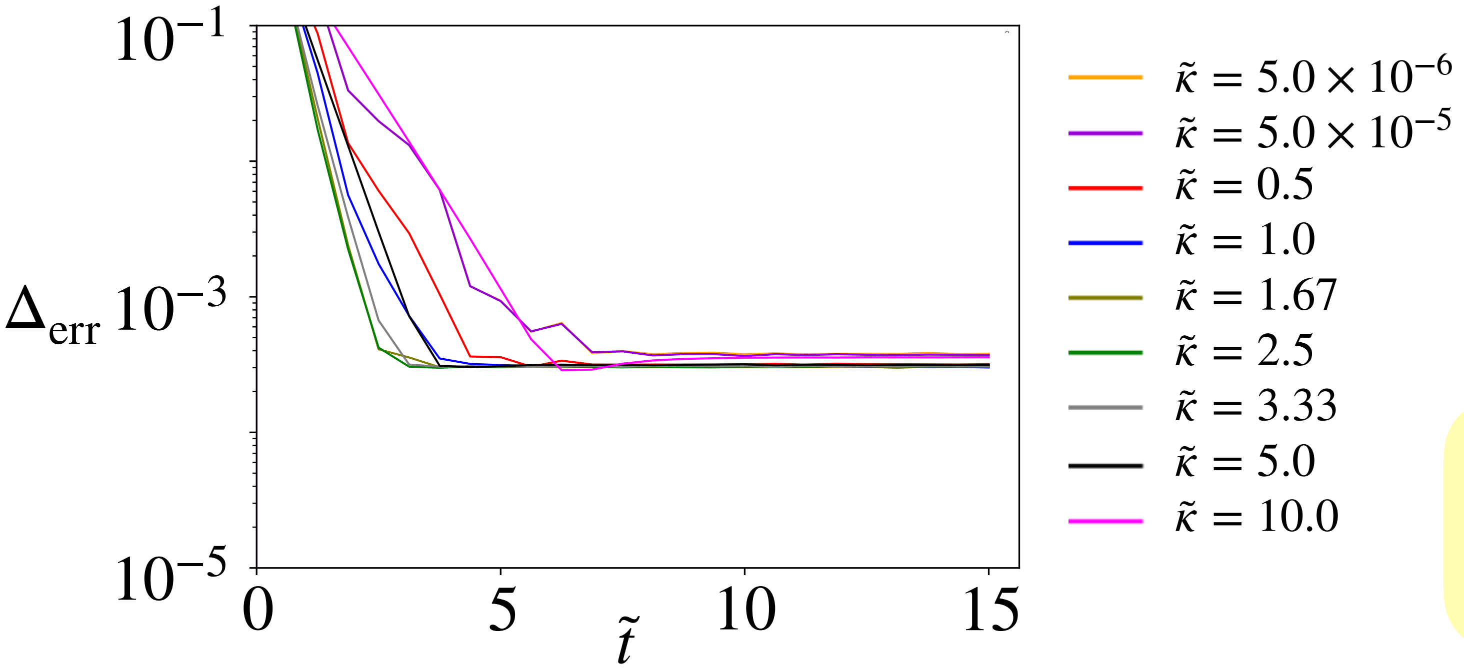
Most efficient $\tilde{\kappa}$ value?

Differences in Convergence Due to Different Diffusion Coefficients

Large κ ⟷ Small κ



- ➢ Small diffusion coefficients do not decay the wave and oscillations can be seen.
- ➢ Large diffusion coefficients lead to slow convergence because of the difficulty in wave propagation.

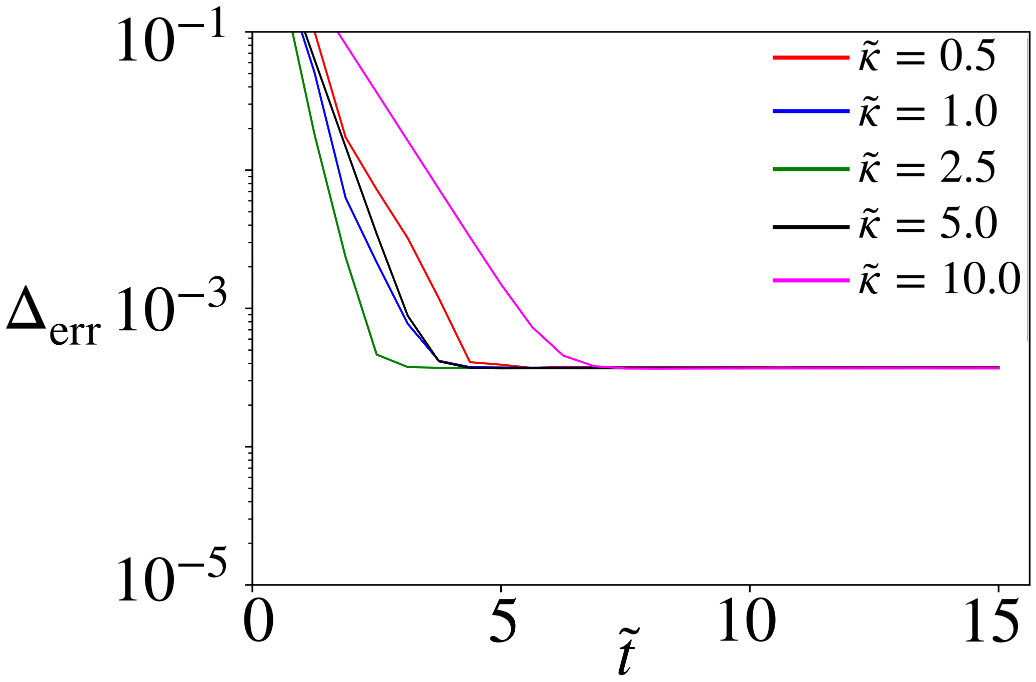


$$\Delta_{\mathrm{err}} = \frac{\sqrt{\frac{1}{n}\Sigma(\Phi - \Phi_{\mathrm{exa}})^2}}{\Phi_{\mathrm{exa,max}} - \Phi_{\mathrm{exa,min}}}$$

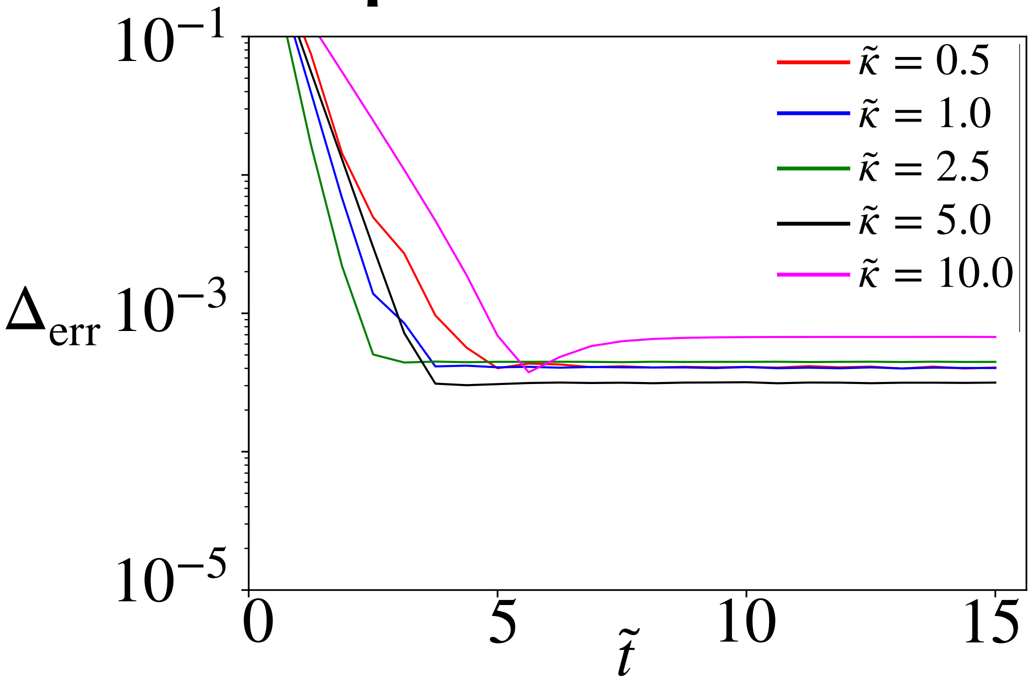
$\tilde{\kappa} = 2.5$ is best parameter.

Differences in Convergence Due to Different Density Distribution

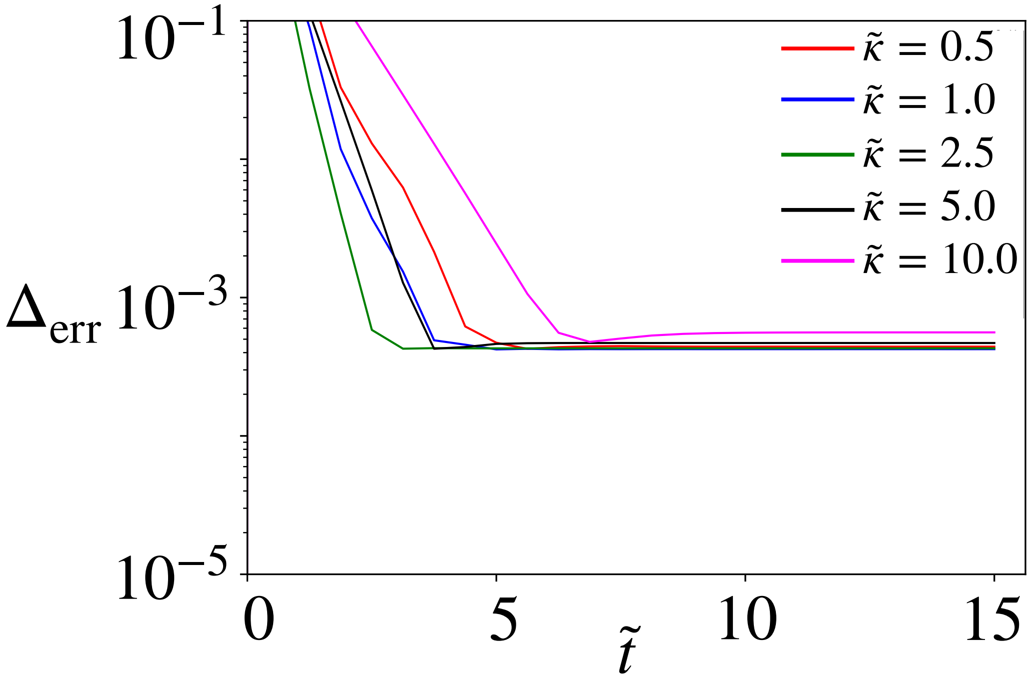
- Uniform sphere r=0.04



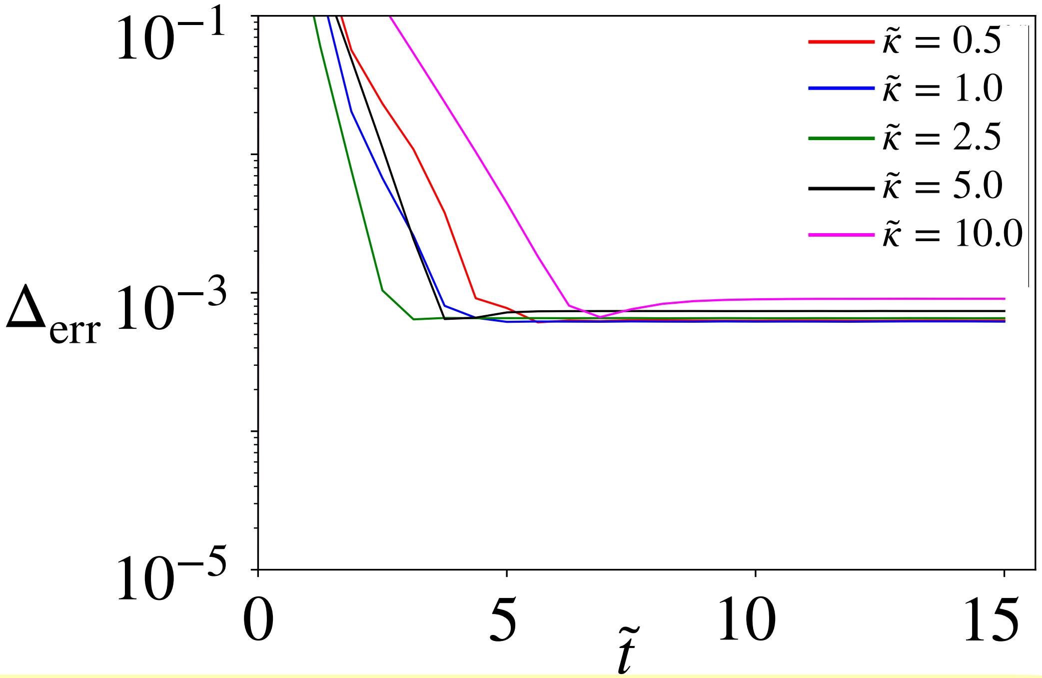
- Uniform sphere r=0.45



- Dipole

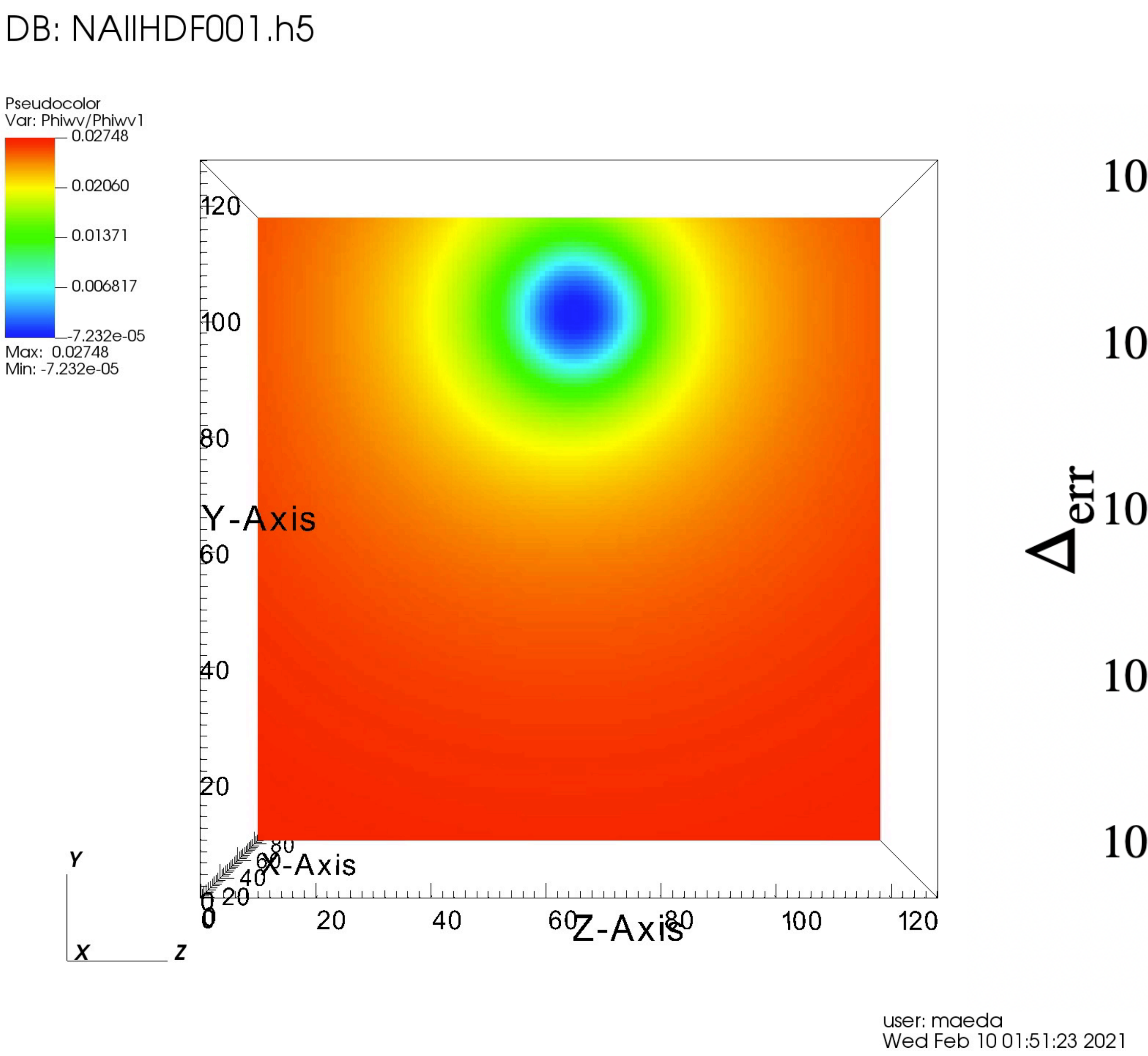


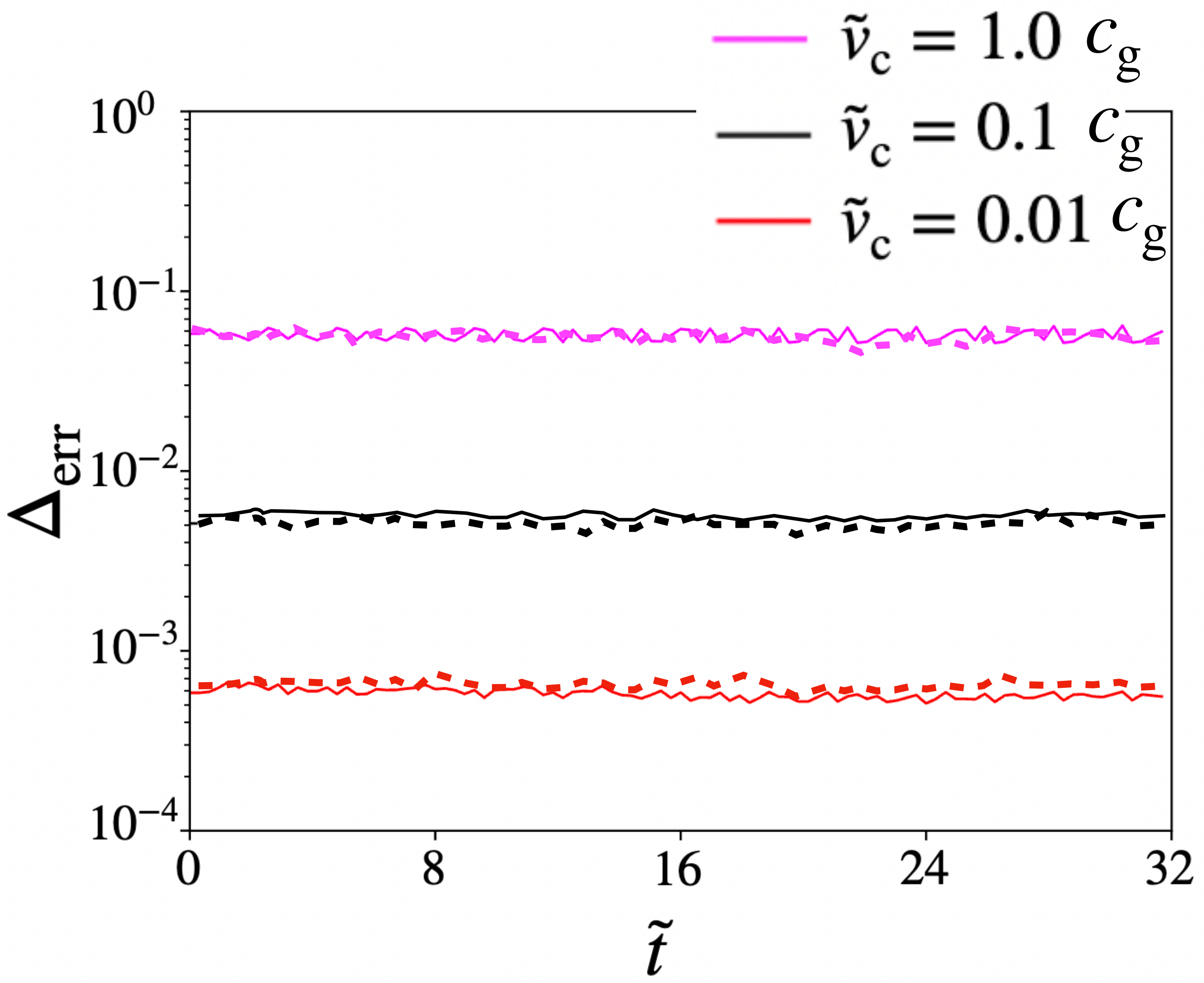
- Quadrupole



Our method reproduces the analytical solution.
$\tilde{\kappa} = 2.5$ is best parameter for all initial conditions.

Test Calculations for Moving Density Field

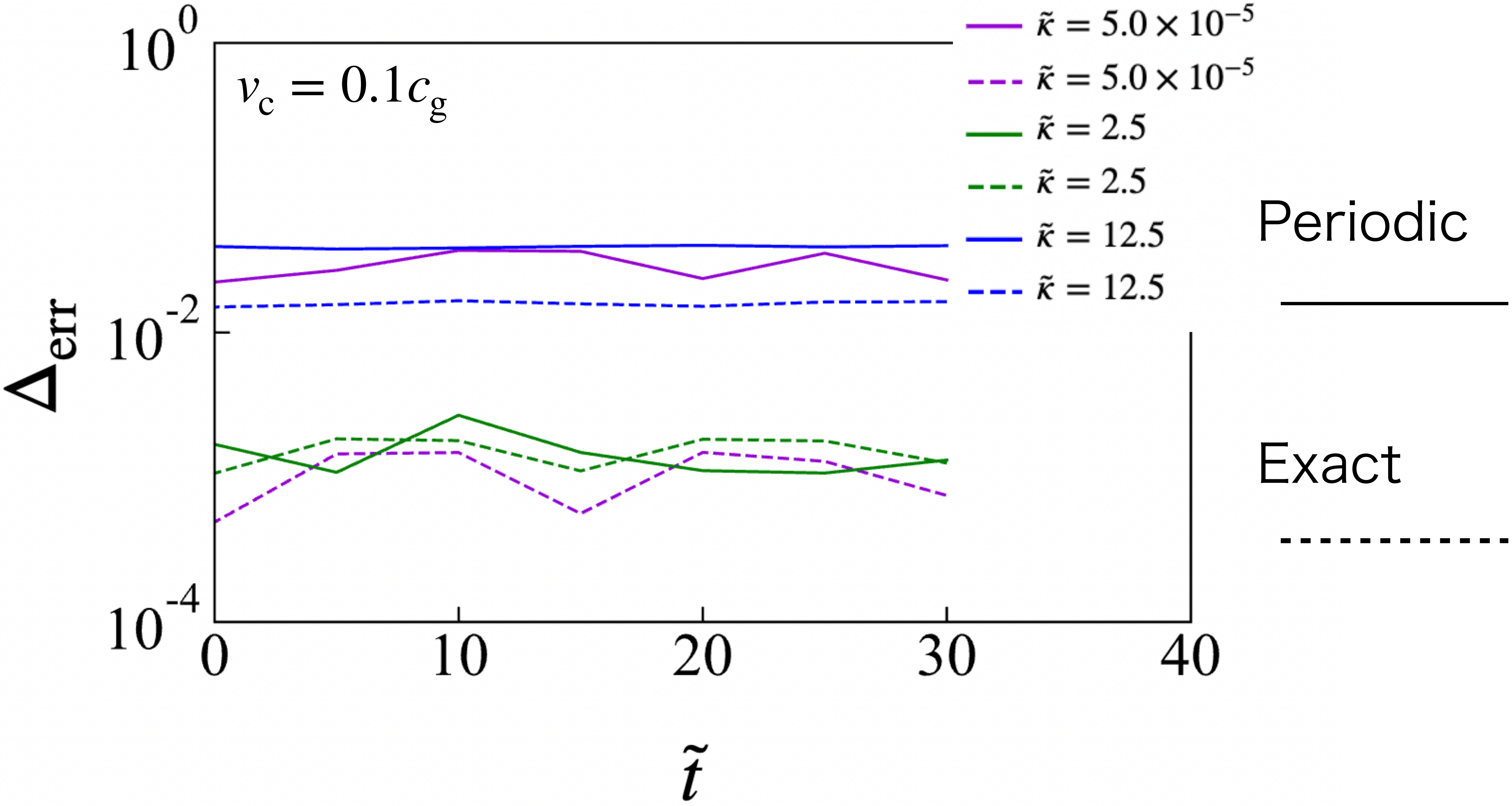




Solid 128 mesh
Dot 256 mesh

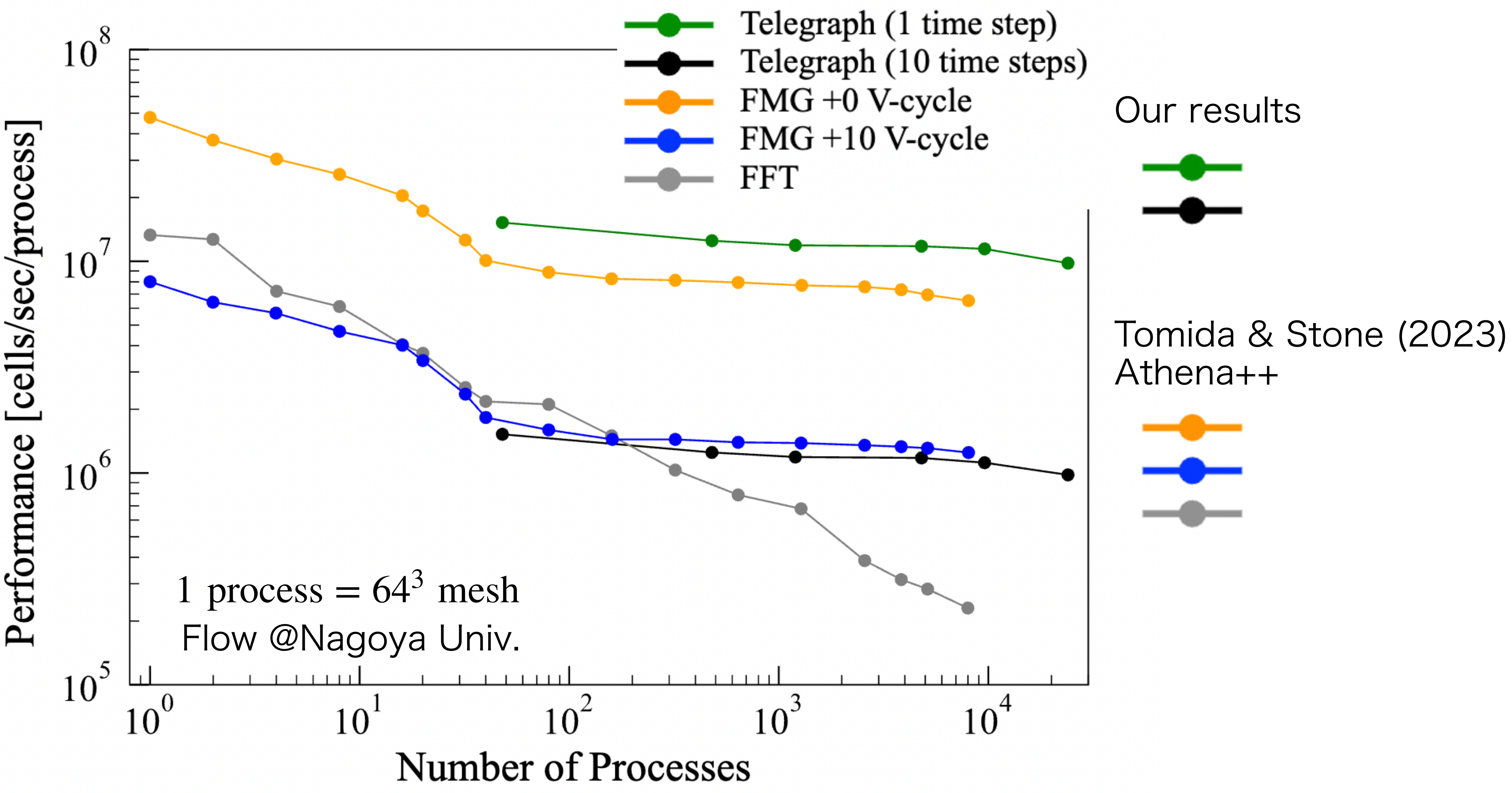
If $c_{\rm g} \gg v_{\rm fluid} + c_{\rm s}$ is used, the error in the calculation can be suppressed.

Error Evolution with Different Boundary Conditions



Errors can be reduced by introducing an appropriate κ in the periodic boundary condition.

Performance for Parallel Computation



Our results

Tomida & Stone (2023)
Athena++

✓ Maintain high parallel efficiency even in large-scale parallel computations.

✓ Our method is the most efficient when the computational timestep is set by cooling or chemical reactions.

Summary

✓ We developed a self-gravity solver using the telegraph equation which is a hyperbolic type with good parallelization efficiency.

✓ We determined the best diffusion coefficient for gravity calculations ($\tilde{\kappa} = 2.5$).

✓ The telegraphic equation can be calculated with reduced error, especially for periodic boundary conditions.

✓ Our method maintains high performance even for $> 10^4$ parallel calculations thanks to the nature of a hyperbolic equation.

✓ In particular, if the time step of the calculation is determined by heating/cooling or chemical reactions, etc., our method can significantly reduce the computational cost.