EFFICIENT METHOD FOR SOLVING THE POISSON EQUATION IN SPHERICAL POLAR COORDINATES

Jeonghyeon Ahn, Woong-Tae Kim, Yonghwi Kim

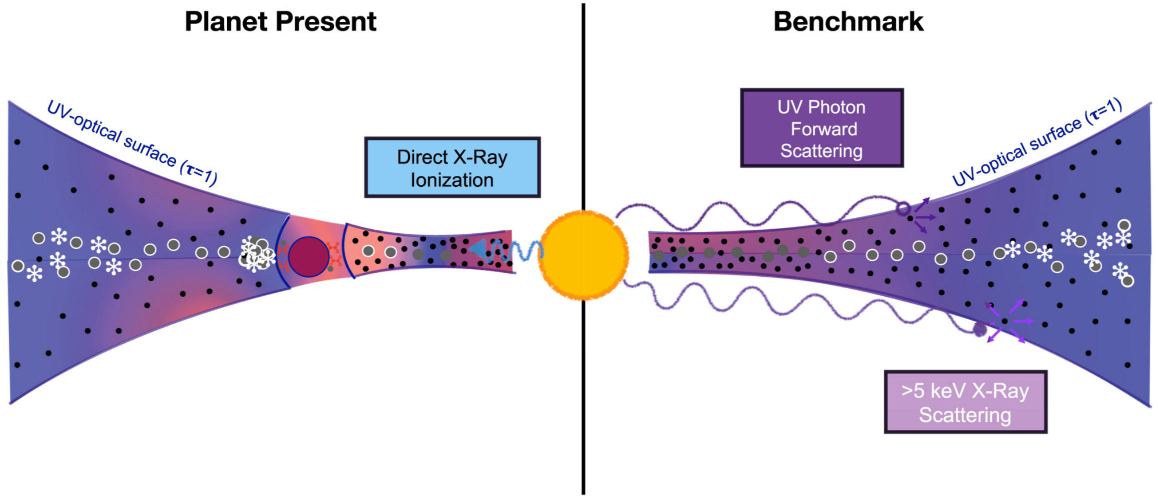
Department of Physics & Astronomy, Seoul National University

Significance of the study

- Poisson equation

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

- Protoplanetary disk – flared structure



- The flared structure makes it difficult to resolve cells near the central star.

Discrete operator in spherical polar coordinate

- Discrete operator for spherical polar coordinates

$$\nabla_r^2 \Phi_{i,j,k} = \frac{3}{r_{i+1/2}^3 - r_{i-1/2}^3}\left(\frac{\Phi_{i+1,j,k} - \Phi_{i,j,k}}{r_{i+1} - r_i} r_{i+1/2}^2 - \frac{\Phi_{i,j,k} - \Phi_{i-1,j,k}}{r_i - r_{i-1}} r_{i-1/2}^2\right)$$

$$\nabla_\theta^2 \Phi_{i,j,k} = \frac{3(r_{i+1/2}^2 - r_{i-1/2}^2)}{2r_i(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})}\left(\frac{\Phi_{i,j+1,k} - \Phi_{i,j,k}}{\theta_{j+1} - \theta_j}\sin\theta_{j+1/2} - \frac{\Phi_{i,j,k} - \Phi_{i,j-1,k}}{\theta_j - \theta_{j-1}}\sin\theta_{j-1/2}\right)$$

$$\nabla_\phi^2 \Phi_{i,j,k} = \frac{3\left(r_{i+1/2}^2 - r_{i-1/2}^2\right)\delta\theta/\delta\phi}{2r_i \sin\theta_j\,(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})}\left(\frac{\Phi_{i,j,k+1} - \Phi_{i,j,k}}{\phi_{k+1} - \phi_k} - \frac{\Phi_{i,j,k} - \Phi_{i,j,k-1}}{\phi_k - \phi_{k-1}}\right)$$

$$\nabla_r^2 \Phi_{i,j,k} + \nabla_\theta^2 \Phi_{i,j,k} + \nabla_\phi^2 \Phi_{i,j,k} = 4\pi G \rho_{i,j,k}$$

Discrete operator in spherical polar coordinate

- FFT in ϕ axis

$$\mathcal{P}_j^m = \exp\left[\frac{2\pi\sqrt{-1}mj}{N_\phi}\right]$$

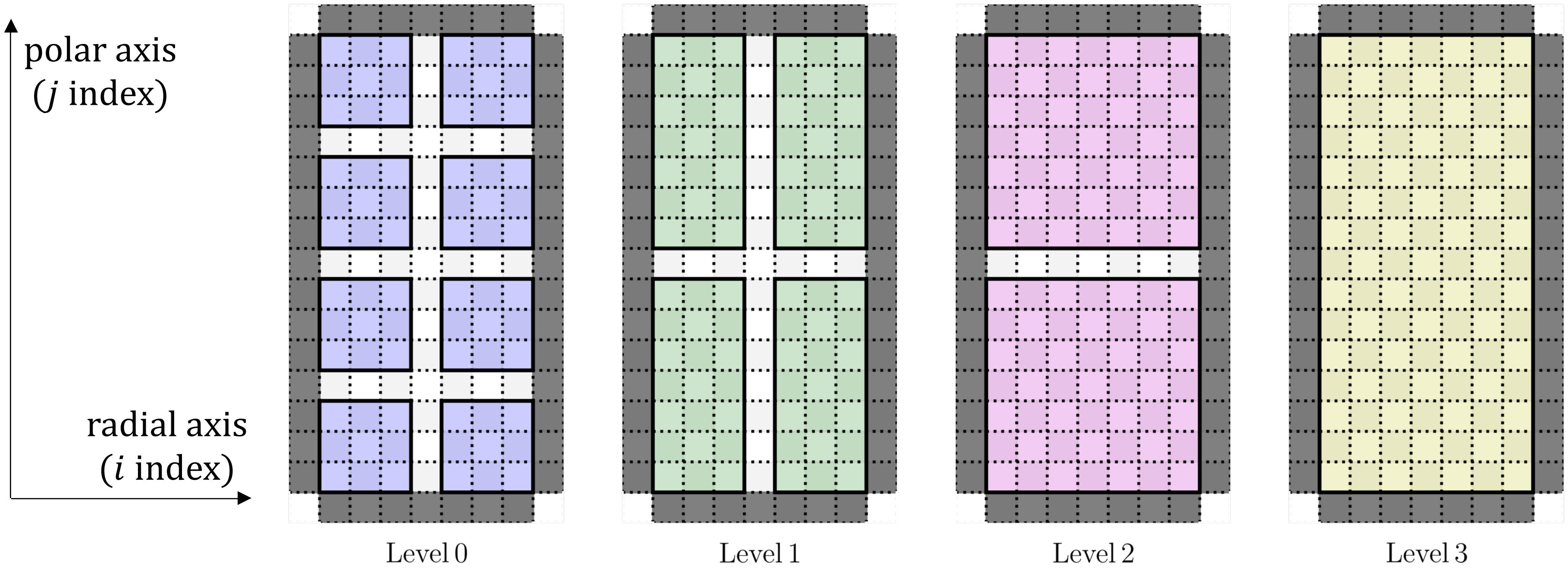
$$\lambda_{i,j}^m = -\frac{3\left(r_{i+1/2}^2 - r_{i-1/2}^2\right)\delta\theta}{2r_i \sin\theta_j \left(r_{i+1/2}^3 - r_{i-1/2}^3\right)\left(\cos\theta_{j-1/2} - \cos\theta_{j+1/2}\right)}\left[\sin\frac{m\pi}{N_\phi}\middle/\frac{\pi}{N_\phi}\right]^2$$

$$\nabla_r^2 \widetilde{\Phi}_{i,j}^m + \nabla_\theta^2 \widetilde{\Phi}_{i,j}^m + \lambda_{i,j}^m \widetilde{\Phi}_{i,j}^m = 4\pi G \rho_{i,j}^m$$

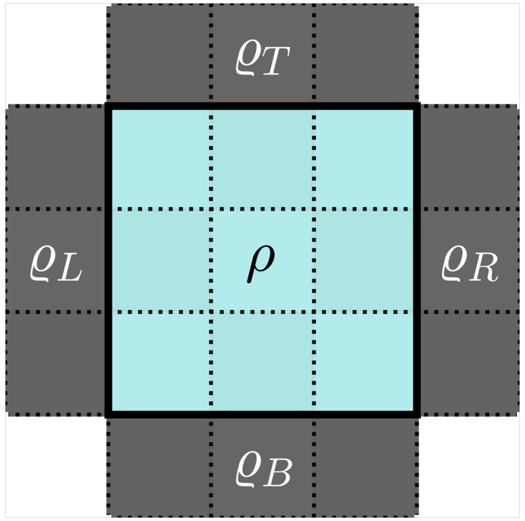
Divide & Conquer approach

- Our strategy is to compute the potential on small grids and then combine them recursively to obtain the potential on larger grids.

$$\nabla_r^2 \widetilde{\Phi}_{i,j}^m + \nabla_\theta^2 \widetilde{\Phi}_{i,j}^m + \lambda_{i,j}^m \widetilde{\Phi}_{i,j}^m = 4\pi G \rho_{i,j}^m$$



James's algorithm after FFT

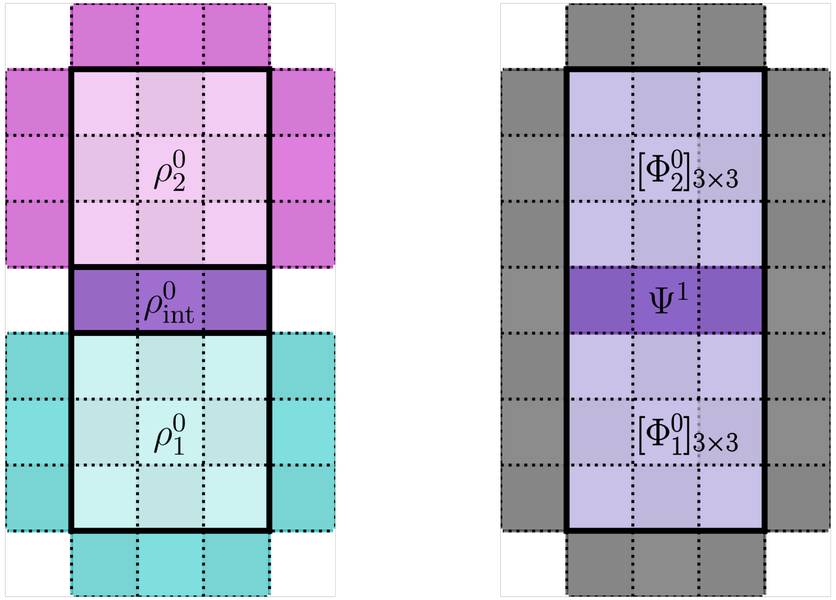


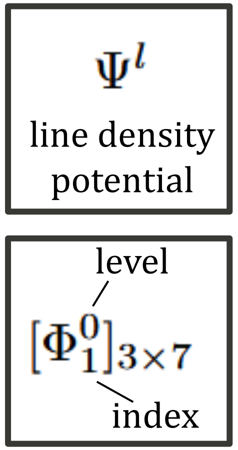
- Zero-boundary potential [Φ] induce the screening density $\varrho_T, \varrho_B, \varrho_L, \varrho_R$.

- Then, isolated potential Φ is calculated by

$$\Phi = [\Phi] + \Theta_T + \Theta_B + \Theta_L + \Theta_R$$

Potential merging





- Using the James algorithm, we can merge two zero-boundary potentials.

$$[\Phi^1]_{3\times7} = [\Phi_1^0]_{3\times3} + [\Phi_2^0]_{3\times3} + \Psi^1$$

Ψ^l

line density potential

level

$[\Phi_1^0]_{3\times7}$

index

Potential merging

- The general formula for level k is

$$[\Phi^k] = \sum_i^{2^k} [\Phi_i^0] + \sum_{l=1}^{k} \sum_{i=1}^{2^{k-l}} \Psi_i^l$$

- From this, we can determine the zero-boundary potential of the entire grid.

- According to James's algorithm, the screening density can be derived from the zero-boundary potential, and then this can be used to calculate the boundary potential Θ.

Discrete Green's function

Ψ^l

line density potential

level

$[\Phi_1^0]_{3\times 7}$

index

- To calculate the boundary potential, we use the Fourier transformed DGF.

$$(\Delta_r^2 + \Delta_\theta^2 + \lambda_{i,j}^m)\mathcal{G}_{i,i',j,j'}^m = 4\pi G \frac{\delta_{ii'}\delta_{jj'}}{V_{i',j'}}$$

- Fourier-transformed isolated potential is

$$\Phi_{i,j}^m = \sum_{i'=1}^{N_r}\sum_{j'=1}^{N_\theta} \mathcal{G}_{i,i',j,j'}^m \rho_{i',j'}^m V_{i',j'}$$

- Then, we get Θ_b.

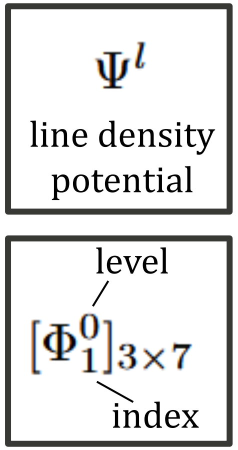
Ψ^l
line density potential

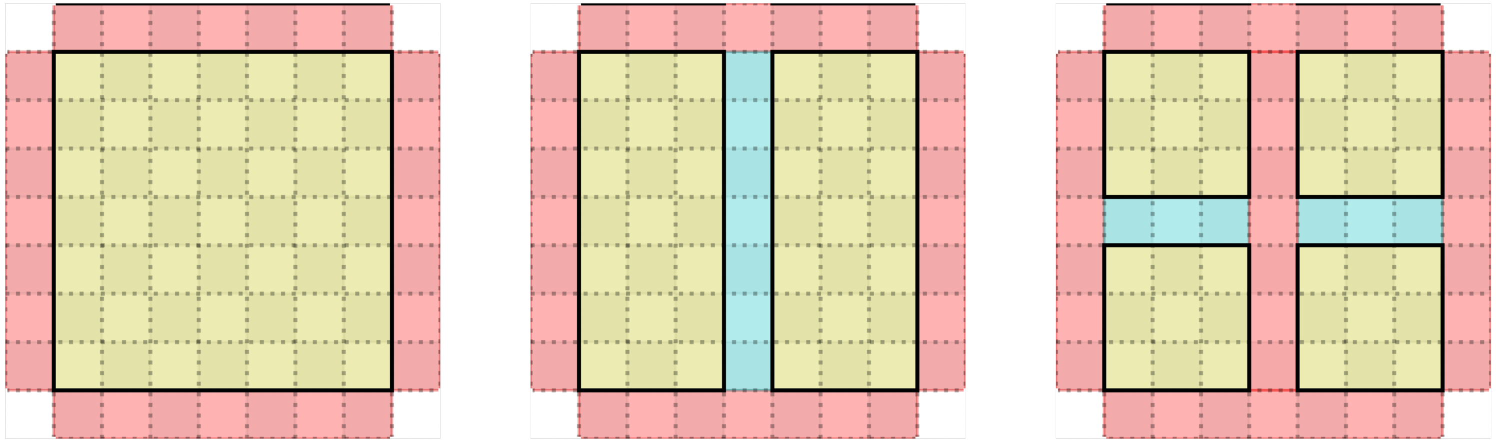
level
$[\Phi_1^0]_{3\times 7}$
index

Discrete Green's function

- James's algorithm is $[\Phi] + \Theta = \Phi$, but we know only Θ_b.
- How can we efficiently determine Θ in the inner region?
- We address this by applying the divide-and-conquer approach once again.

Source-free potential





- We calculate the potential only along the middle line. We repeat this.

- In this way, the total computational cost is reduced to logarithmic scale.

Computational complexity

- step 1. Calculate all $[\Phi^0]_{3\times3}$ - $O(N_\phi N^2)$
- step 2. Calculate all Ψ - $O(N_\phi N^2 \log N)$
- step 3. Calculate entire boundary potential using DGF - $O(N_\phi N^2)$
- step 4. Calculate all source-free potential - $O\left(N_\phi N^2 \log N\right)$
- Fast Fourier Transform (at the start & end) - $O(N_\phi N^2 \log N_\phi)$

Ψ^l

line density potential

level

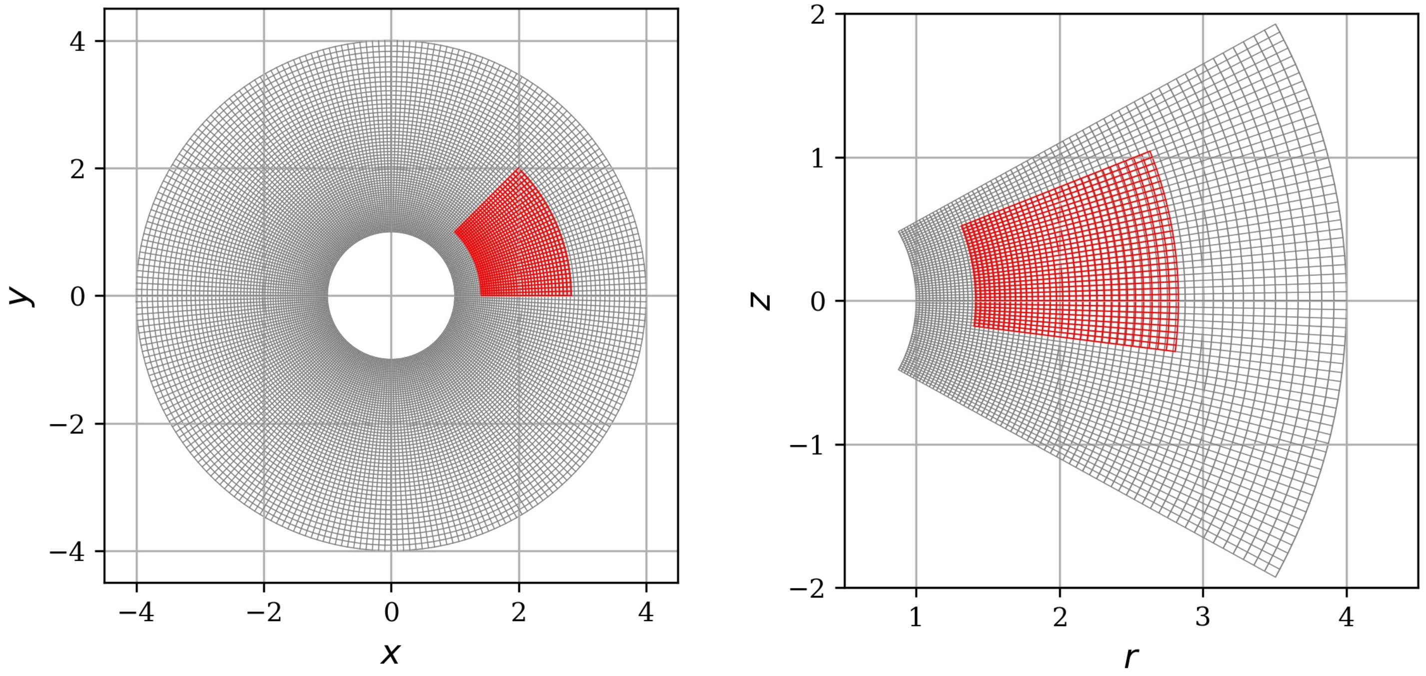
$[\Phi_1^0]_{3\times7}$

index

Accuracy test

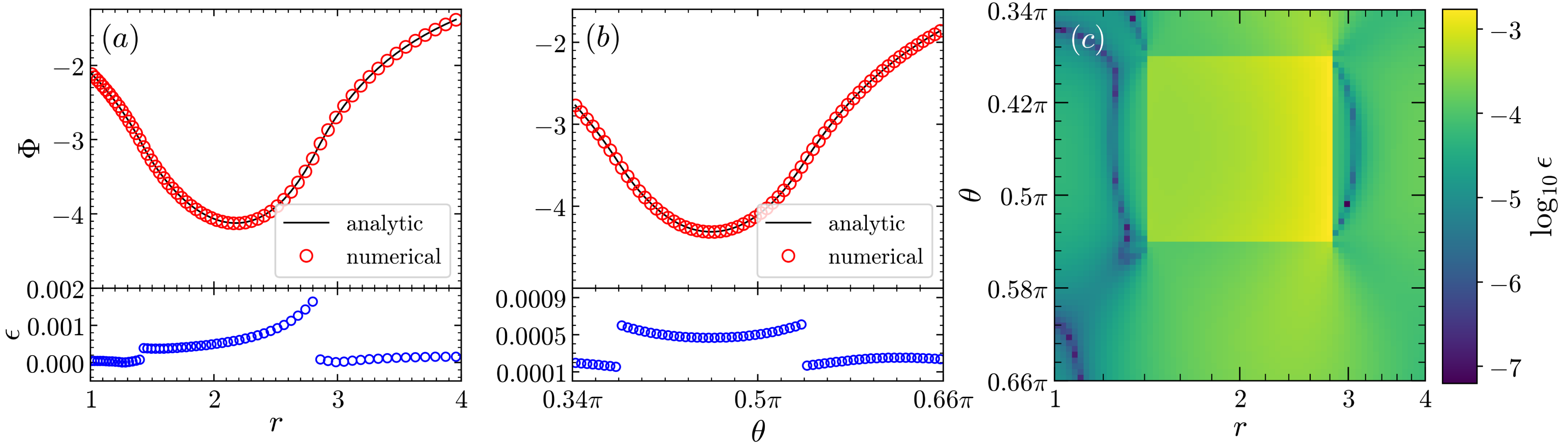
- Relative error ε is given by

$$\epsilon \equiv \left| \frac{\Phi - \Phi_r}{\Phi_r} \right|$$



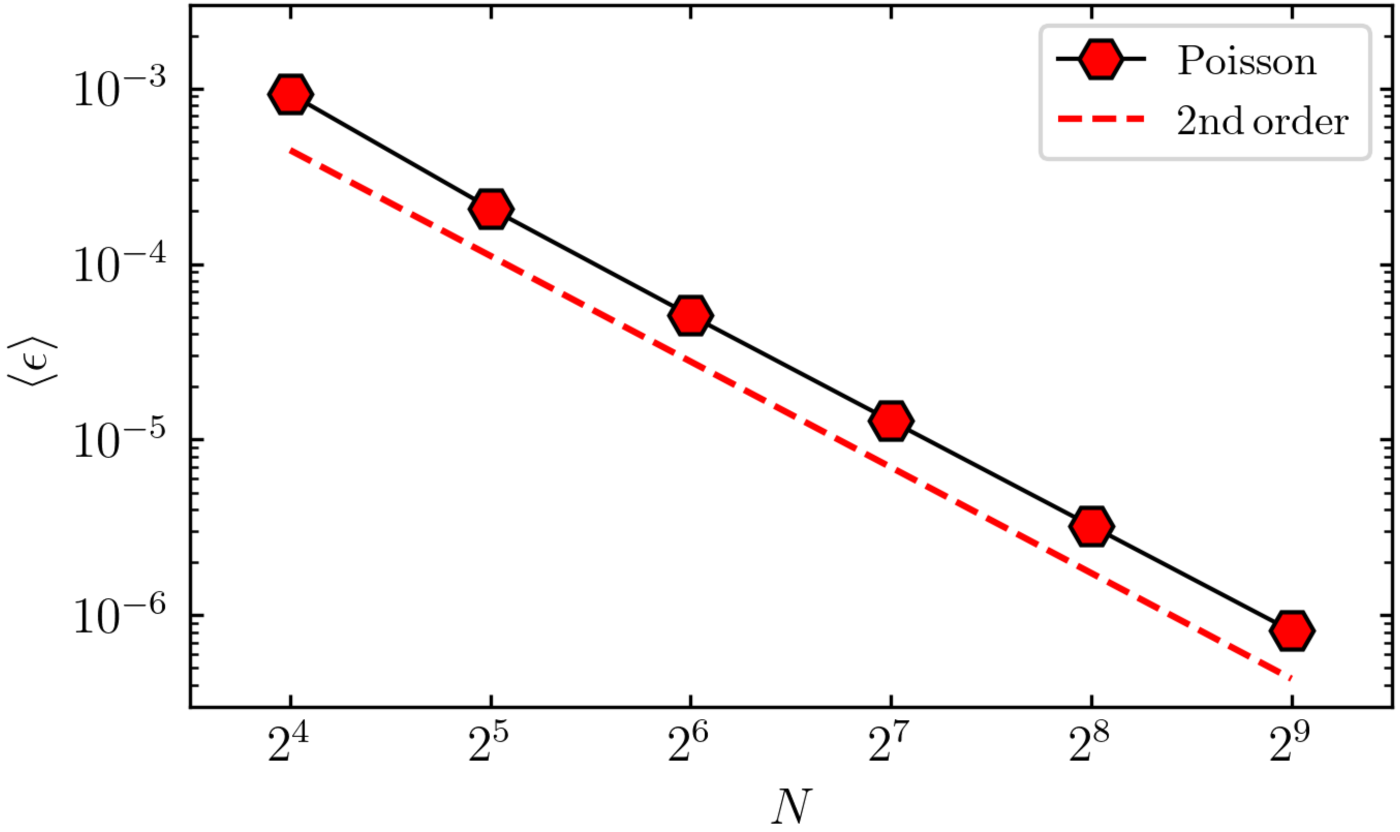
Accuracy test

	r	θ	ϕ
Domain size	$[1.0, 4.0]$	$[0.34\pi, 0.66\pi]$	$[0.0, 2.0\pi]$
sector size	$[\sqrt{2}, 2\sqrt{2}]$	$[0.38\pi, 0.54\pi]$	$[0.0\pi, 0.25\pi]$
Resolution	64	64	256



Convergence test

- The slope of the test results is approximately -2.0, which indicates that our solver achieves second-order convergence.



Performance test – weak scaling

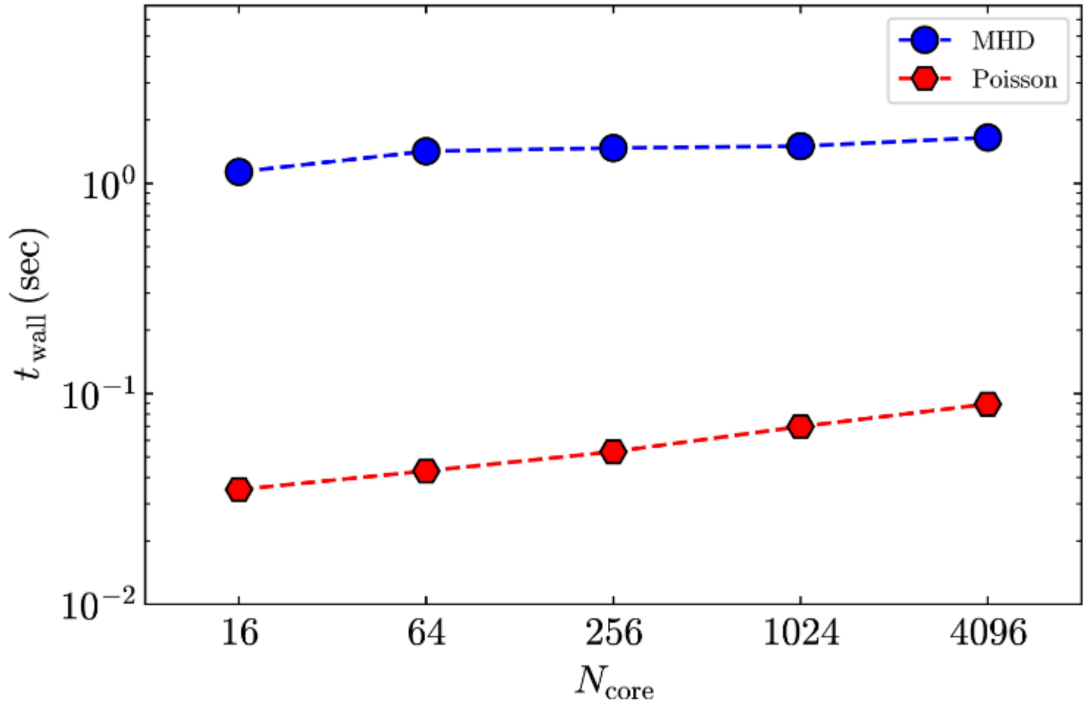
- Each core – 16×16×1024 grid
- We averaged the wall-clock time over 100 runs

domain size

core	16	64	256	1024	4096
N_r	64	128	256	512	1024
N_θ	64	128	256	512	1024
N_ϕ	1024	1024	1024	1024	1024

Table 2. Results of the weak scaling test, listing the wallclock time per cycle and the update time per cell for both the Poisson solver and the MHD solver in FARGO3D for different numbers of cores.

	Wall clock time (s)		Update time per cell (μs)	
Number of Cores	MHD	Poisson	MHD	Poisson
16	1.134	0.03503	4.325	0.1336
64	1.420	0.04292	5.418	0.1637
256	1.472	0.05300	5.616	0.2022
1024	1.502	0.06979	5.728	0.2662
4096	1.652	0.08929	6.304	0.3406



Summary

- We developed an efficient and accurate Poisson solver in spherical polar coordinates.

- Implemented using a divide-and-conquer approach, achieving $O(N^3 \log N)$ computational complexity.

Kernel matrix

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix} \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_n \end{pmatrix} \equiv \begin{pmatrix} \Phi_1 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_m \end{pmatrix}$$

$$K\rho = \Phi$$

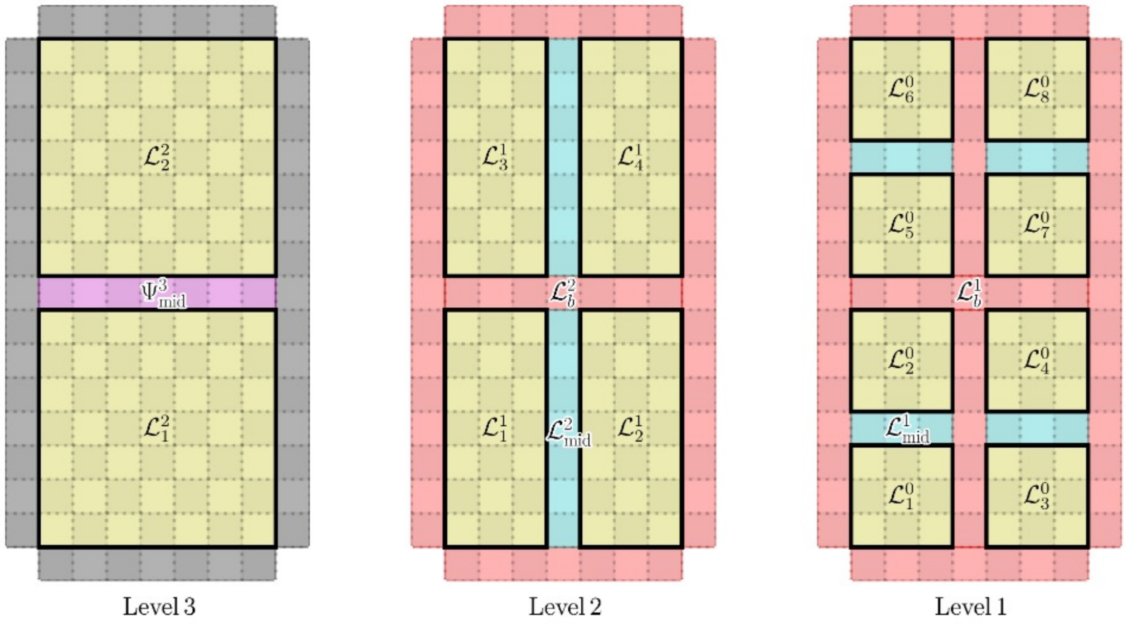
ρ_i - density of each cell, Φ_i - potential of each cell

To enable efficient computation, we devised a kernel matrix.

It can be thought of as the Discrete Green’s Function for various cases.

kernels were used throughout all parts of the code.

Line density potential Ψ



- Method for calculating source-free potential can be used to compute the line density potential Ψ.
- Since the density that generates Ψ is nonzero only along the middle line, by calculating only the potential along that line, remaining region becomes source-free.

Domain shape

- At the highest level, the equator consists of two cells.
- In addition, the boundary at $r = 2^N$ is included in the domain.
- As a result, the overall grid size takes the form of a power of two.