

Computing the gravitational potential on nested meshes using the convolution method[★]

Eduard I. Vorobyov¹, James McKevitt¹, Igor Kulikov², and Vardan Elbakyan^{3,4}

¹ University of Vienna, Department of Astrophysics, Türkenschanzstrasse 17, Vienna 1180, Austria
e-mail: eduard.vorobiev@univie.ac.at

² Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Lavrentieva ave., 6, Novosibirsk 630090, Russia

³ Research Institute of Physics, Southern Federal University, Stachki Ave. 194, Rostov-on-Don 344090, Russia

⁴ School of Physics, University of Leicester, Leicester, LE1 7RH, UK

Received 5 August 2022 / Accepted 6 January 2023

ABSTRACT

Aims. Our aim is to derive a fast and accurate method for computing the gravitational potential of astrophysical objects with high contrasts in density, for which nested or adaptive meshes are required.

Methods. We present an extension of the convolution method for computing the gravitational potential to the nested Cartesian grids. The method makes use of the convolution theorem to compute the gravitational potential using its integral form.

Results. A comparison of our method with the iterative outside-in conjugate gradient and generalized minimal residual methods for solving the Poisson equation using nonspherically symmetric density configurations has shown a comparable performance in terms of the errors relative to the analytic solutions. However, the convolution method is characterized by several advantages and outperforms the considered iterative methods by factors 10–200 in terms of the runtime, especially when graphics processor units are utilized. The convolution method also shows an overall second-order convergence, except for the errors at the grid interfaces where the convergence is linear.

Conclusions. High computational speed and ease in implementation can make the convolution method a preferred choice when using a large number of nested grids. The convolution method, however, becomes more computationally costly if the dipole moments of tightly spaced gravitating objects are to be considered at coarser grids.

Key words. methods: numerical – gravitation – hydrodynamics

1. Introduction

Gravity, together with magnetic fields and radiation, is a key mechanism that determines the formation and evolution of many astrophysical objects from planets to clusters of galaxies. On rare occasions the gravity force can be found analytically (Binney et al. 1987). In practice, various numerical methods are often employed to compute the gravitational potential using the Poisson equation (e.g., Binney et al. 1987; Press et al. 1992; Bodenheimer et al. 2007). The gravity force can then be calculated from the gradient of the potential using discretization schemes with various levels of complexity (e.g., Stone & Norman 1992; Kulikov & Vorobyov 2016; Wang & Yen 2020). The required computational resources are often substantial and may take up a large portion of the entire runtime of a three-dimensional numerical hydrodynamics code. Therefore, fast methods such as the fast Fourier transform (FFT) are often a method of choice if the geometry of the system permits regular meshes with equidistant cells (Press et al. 1992).

In many astrophysical applications, the FFT method is difficult to apply because of the requirement for irregular computational meshes that can better resolve the spatial regions where matter accumulates and density increases. For example, the gravitational collapse of interstellar clouds involves an increase in the gas density in the cloud interiors by many orders of magnitude (e.g., Masunaga et al. 1998; Larson 2003). Extensions of the

FFT algorithm to nonequidistant meshes exist (Potts & Tasche 2008) but they are usually much slower than their classical counterparts. In such situations, nested meshes or, more generally, adaptive meshes are often used to achieve a better numerical resolution in the innermost regions (e.g., Matsumoto & Hanawa 2003; Machida et al. 2005; Tomida et al. 2015; Hennebelle 2018; Lebreuilly et al. 2020).

Finding the gravitational force on nested meshes is a complicated task. A contemporary approach involves solving for the Poisson equation first on the coarsest grid. The result is then used to calculate the gravitational potential on the refined grid using, for the boundary conditions, the potential at the coarsest grid (e.g., Matsumoto & Hanawa 2003; Guillet & Teyssier 2011; Wang & Yen 2020). Multigrid methods are often used on each level to improve the accuracy, especially when tightly spaced gravitating objects are present on the deepest levels of refinement (Matsumoto & Hanawa 2003; Matsumoto 2007; Truelove et al. 1998; Stone et al. 2020). Moving in this fashion from the coarsest to the most refined grid, the gravitational potential on the entire nested mesh can be retrieved. However, this “outside-in” approach has a potentially serious drawback – this algorithm is difficult to parallelize because computations on a refined grid cannot be done until the potential on a coarse grid is found. According to our experience, calculating the gravity force in numerical hydrodynamics equations on nested meshes with a moderate base resolution $N \leq 64^3$ may take a large fraction of the entire computational time. Therefore, developing alternative methods is worthwhile.

[★] Code can be viewed at <https://github.com/jamesmckevitt/CM4NG>

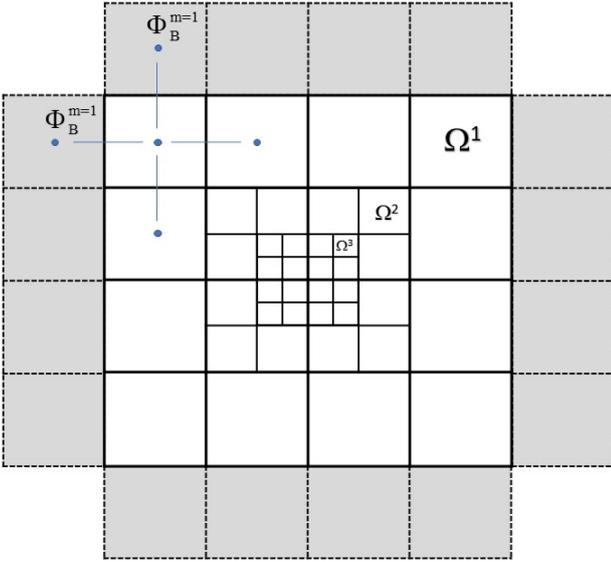


Fig. 1. Example of a nested mesh with three levels of refinement: Ω^1 , Ω^2 , and Ω^3 . The gray-shaded cells represent boundary layers for the coarsest grid; the gravitational potential at these boundary layers has to be found a priori.

Here, we present a conceptually different method of computing the gravitational force on nested meshes, which employs the integral form of the gravitational potential and the convolution theorem to solve the resulting triple sum. To the best of our knowledge, this method has not been explicitly described in the literature (but see [Ruffert & Mueller 1990](#)), although its application to regular (non-nested) meshes in astrophysics is well known (e.g., [Müller et al. 2012](#); [Vorobyov et al. 2020](#); [Stücker et al. 2020](#)). The layout of this work is as follows. In Sect. 2 we briefly outline the outside-in solution procedure using the conjugate gradient method and provide the test results on two nonspherically symmetric mass distributions. In Sect. 3 we review the general convolution method and in Sect. 4 we describe its extension to nested meshes. The comparison of models is provided in Sect. 5. The analysis of runtimes is given in Sect. 6. Our main conclusions are summarized in Sect. 8.

2. Outside-in conjugate gradient solution procedure

In this section, we briefly outline the outside-in solution method for the gravitational potential Φ on nested meshes. This solution procedure is used merely for comparison with an alternative method that we propose later in Sect. 3. In general, the gravitational potential of an arbitrary mass distribution can be found by solving for the Poisson equation

$$\nabla \cdot \nabla \Phi = \Delta \Phi = 4\pi G \rho, \quad (1)$$

where Δ is the Laplace operator, G is the gravitational constant, and ρ is the mass volume density. Figure 1 presents a simple example of three nested meshes on the Cartesian two-dimensional grid (x, y) . For simplicity we show only two dimensions. However, a third dimension will be added in practical tests. By default, the outer coarsest grid Ω^1 encompasses the entire computational domain with a size of $[\omega_L : \omega_R] \times [\omega_D : \omega_T]$. The inner finer grids Ω^2 and Ω^3 provide a better numerical resolution in the inner regions of the computational domain.

The physical size of each nested grid in the x - and y -coordinate directions is determined as $(\omega_R - \omega_L)/2^{m-1}$ and $(\omega_T - \omega_D)/2^{m-1}$, where m takes values from 1 to 3 in our simple example. Furthermore, each nested grid is subdivided into $N = 4$ square cells. Such a mesh refinement is often needed to better resolve the central denser regions of astrophysical objects (e.g., gravitationally collapsing interstellar clouds of gas).

The outside-in solution procedure consists of a sequence of steps starting from the calculation of the gravitational potential on the coarsest $\Omega^{m=1}$ grid. We discretize the Laplace operator using a simple five-point stencil

$$\Delta \Phi \approx \frac{1}{dx_m^2} (\Phi_{i+1,j}^m - 2\Phi_{i,j}^m + \Phi_{i-1,j}^m) + \frac{1}{dy_m^2} (\Phi_{i,j+1}^m - 2\Phi_{i,j}^m + \Phi_{i,j-1}^m), \quad (2)$$

where dx_m and dy_m are the sizes of computational cells in x - and y -coordinate direction defined as $dx_m = (\omega_R - \omega_L)/(2^{m-1}N)$ and $dy_m = (\omega_T - \omega_D)/(2^{m-1}N)$, respectively. We note that we adopt $dx_m = dy_m$ throughout the paper. A similar algorithm applies to the third z -coordinate dimension in practical tests. The discretization of the Laplace operator introduces boundary values of the potential Φ_B for each nested domain. The corresponding boundary ghost zones for the coarsest Ω^1 grid are shown in Fig. 1 as additional layers of gray-colored cells. These boundary values can be found using the multipole expansion of the Laplace equation $\Delta \Phi = 0$ in spherical harmonics (note that by our model setup the volume density outside the computational domain is equal zero). The pertaining formulae can be found in, for example, [Jackson & Fox \(1999\)](#). The higher-order harmonics are added until a desired accuracy in $\Phi_B^{m=1}$ is reached. This method usually converges fast for compact density configurations when most of the mass is concentrated in the center of the computational domain, but may become computationally costly or even diverge for arbitrary mass distributions with large amounts of mass present near outer boundaries.

Once the boundary values $\Phi_B^{m=1}$ of the gravitational potential are found, the potential on the coarsest grid $\Phi^{m=1}$ can be computed using one of the methods of linear algebra. In this work, we employ the conjugate gradient method described in, for example, [Press et al. \(1992\)](#). As with any iterative method, a truncation error has to be established a priori. The completion of the iterative process in the conjugate gradient method occurs when the norm of the relative residuals falls below $\epsilon = 10^{-6}$. In practice, we substitute the calculated grid values of the potential into the numerical scheme for approximating the Poisson equation (see Eq. (2)) and use the resulting data to form a residual difference vector with the grid values of the right-hand side ($4\pi G \rho$). The norm of this residual vector should not exceed 10^{-6} relative to the norm of the right-hand side vector. It is important to note that the achieved accuracy is not the value of the relative error in the gravitational potential. The relative error in the gravitational potential is improved only when the grid is refined. In the iterative process, it is only guaranteed that the grid values of the potential, when substituting into the scheme for the Poisson equation, restore the right-hand side with a chosen truncation error of $\epsilon = 10^{-6}$. The dependence of the solution on ϵ is investigated in Sect. 5. The potentials Φ^m on the finer grids can then be found by taking the corresponding potentials on the coarser grids Φ^{m-1} as the boundary values (which are known from the previous step). Moving in this manner from the outer coarsest grid toward the inner finest grid the gravitational potential in the entire computational domain can be computed. In the following

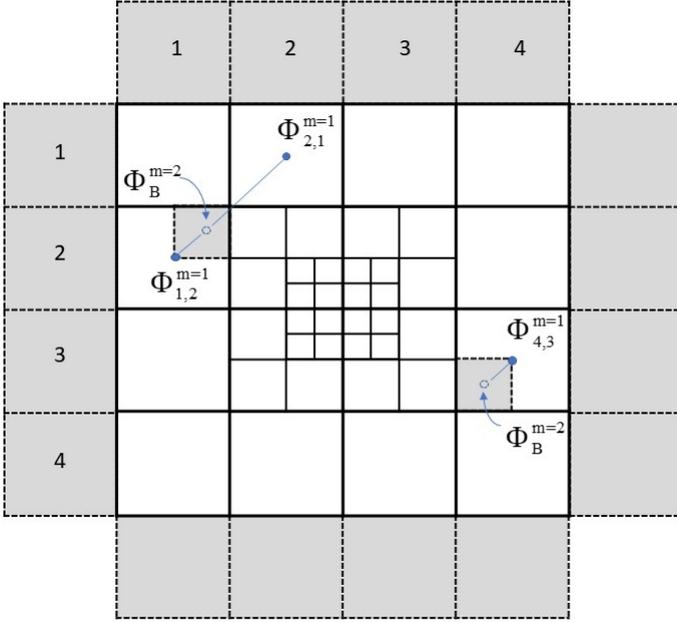


Fig. 2. Examples of finding the boundary potentials for the $m = 2$ nested grid. The bottom-right example presents a simple extrapolation of the nearest potential at the $m = 1$ grid. The top-left example shows a more sophisticated linear interpolation scheme, which involves two neighboring potentials at the $m = 1$ level.

text, the outlined solution procedure is denoted as the outside-in conjugate gradient method (hereafter, OiCG method).

An important prerequisite for the OiCG method is a proper interpolation of the potentials on a coarser grid to be used as the boundary values on a finer grid (Matsumoto & Hanawa 2003; Wang & Yen 2020). Figure 2 shows two examples for the choice of the boundary values $\Phi_B^{m=2}$ on the second nested grid $m = 2$. The first choice shown in the bottom-right corner consists in taking the nearest value of the potential on the corresponding coarser grid $m = 1$, so that $\Phi_B^{m=2} = \Phi_{4,3}^{m=1}$. The subscripts (4,3) refer to the cell number on the two-dimensional (x, y) mesh. This choice, however, is not optimal and will lead to degraded accuracy of the solution, as we subsequently demonstrate below. A better accuracy can be achieved by taking a linear interpolation of the potentials on the coarser $m = 1$ grid to the center of the corresponding ghost zone of the $m = 2$ mesh. The algorithm is illustrated in the upper-left corner of Fig. 2 and for the left-hand side interface between the nested grids ($i = 1, x$ -coordinate) can be described as follows

$$\Phi_{B,(i,k,l)}^{m+1} = 0.75 \Phi_{ib,kb,lb}^m + 0.25 \Phi_{ib+1,kb+1,lb+1}^m, \quad \text{if } \text{mod}(k, 2) \neq 0 \text{ and } \text{mod}(l, 2) \neq 0, \quad (3)$$

$$\Phi_{B,(i,k,l)}^{m+1} = 0.75 \Phi_{ib,kb,lb}^m + 0.25 \Phi_{ib+1,kb+1,lb-1}^m, \quad \text{if } \text{mod}(k, 2) \neq 0 \text{ and } \text{mod}(l, 2) = 0, \quad (4)$$

$$\Phi_{B,(i,k,l)}^{m+1} = 0.75 \Phi_{ib,kb,lb}^m + 0.25 \Phi_{ib+1,kb-1,lb+1}^m, \quad \text{if } \text{mod}(k, 2) = 0 \text{ and } \text{mod}(l, 2) \neq 0, \quad (5)$$

$$\Phi_{B,(i,k,l)}^{m+1} = 0.75 \Phi_{ib,kb,lb}^m + 0.25 \Phi_{ib+1,kb-1,lb-1}^m, \quad \text{if } \text{mod}(k, 2) = 0 \text{ and } \text{mod}(l, 2) = 0. \quad (6)$$

Here, the indices k and l take values from 1 to $N + 2$ and ib, kb, lb are related to i, k, l as $ib = N/4 + i/2 + 1, kb = N/4 + k/2 + 1, lb = N/4 + l/2 + 1$, respectively. The algorithm can be generalized to other grid interfaces in a similar manner. More

sophisticated methods involving the introduction of additional boundary layers can be found in, e.g., Wang & Yen (2020).

To test the accuracy of the OiCG method, we computed the gravitational potential and gravitational acceleration for two mass configurations with known analytical solutions: an oblate homogeneous ellipsoid and two nonintersecting homogeneous spheres. The former has a ratio of semi-major axes $a_1 : a_2 : a_3 = 0.5 : 1.0 : 1.0$. The latter is composed of two homogeneous spheres, both having a radius of 0.2. Their centers lie in the $x - y$ plane at -0.5 and 0.5 with respect to the coordinate center. This latter test model mimics a binary star. The computational domain has dimensions of $[-2.25:2.25]$ in each coordinate direction. The ratio of masses of the two spheres is 2:1. The analytic solution for the gravitational potential of an oblate ellipsoid was derived by Chandrasekhar (1969) and is provided in Stone & Norman (1992). The analytic solution for the binary star is the sum of analytic solutions for the two homogeneous spheres, which can also be found in Stone & Norman (1992).

The left pair of columns in Fig. 3 presents the relative errors in the gravitational potential and gravitational acceleration for an ellipsoid with the ratio of semi-major axes $a_1 : a_2 : a_3 = 0.5 : 1.0 : 1.0$. The errors in the gravitational potential and acceleration are calculated with respect to the analytical solutions as follows

$$\text{err}_\Phi = \left| \frac{\Phi_{\text{num}} - \Phi_{\text{an}}}{\Phi_{\text{an}}} \right|; \quad \text{err}_g = \left| \frac{|g|_{\text{num}} - |g|_{\text{an}}}{|g|_{\text{an}}} \right|, \quad (7)$$

where the subscripts “num” and “an” refer to the numerical and analytical values, respectively.

The modulus of gravitational acceleration $|g|$ was calculated using a simple six-point stencil for the central discretization scheme, so that the x -component of the gravitational acceleration is written as

$$g_x = -\frac{\partial \Phi}{\partial x} \approx \frac{\Phi_{i+1,j,k}^m - \Phi_{i-1,j,k}^m}{2 \, dx_m}. \quad (8)$$

The other y - and z -components are expressed accordingly. This simple six-point stencil has to be modified at the cells adjacent to the boundary between fine and coarse grids. When calculating the gravitational acceleration at the cells of a finer grid that are adjacent to a coarser grid, we use the interpolated values of the gravitational potential as described in Eqs. (3)–(6). The opposite case of the gravitational acceleration on a coarser grid is more subtle. A straightforward method would be to use the potentials in the adjacent cells on a finer grid, arithmetically averaged to the geometrical center of the underlying coarser cell. A more physically motivated method makes use of the Gauss theorem and requires that the gravitational acceleration at the coarse-fine grid interface be equal when computed using the fine and coarse grid cells, respectively. We review these cases in Appendix A and note that we used the first simple method in this paper but in real simulations the second method may be more appropriate. The boundary conditions on the gravitational potential for the outermost mesh $m = 1$ were calculated using the analytical solution. Two mesh configurations were considered: $m = 1$ (no mesh refinement) and $m = 4$ (four nested grids). The number of grid cells for each mesh was set equal to $N = 128$ in each coordinate direction. Clearly, the solution with a simple extrapolation of the boundary potentials to the nearest cell on the coarse grid (as shown in the bottom row of Fig. 2) produces a notably higher relative error in both Φ and $|g|$ than the solution that employs a more sophisticated linear interpolation scheme (middle row

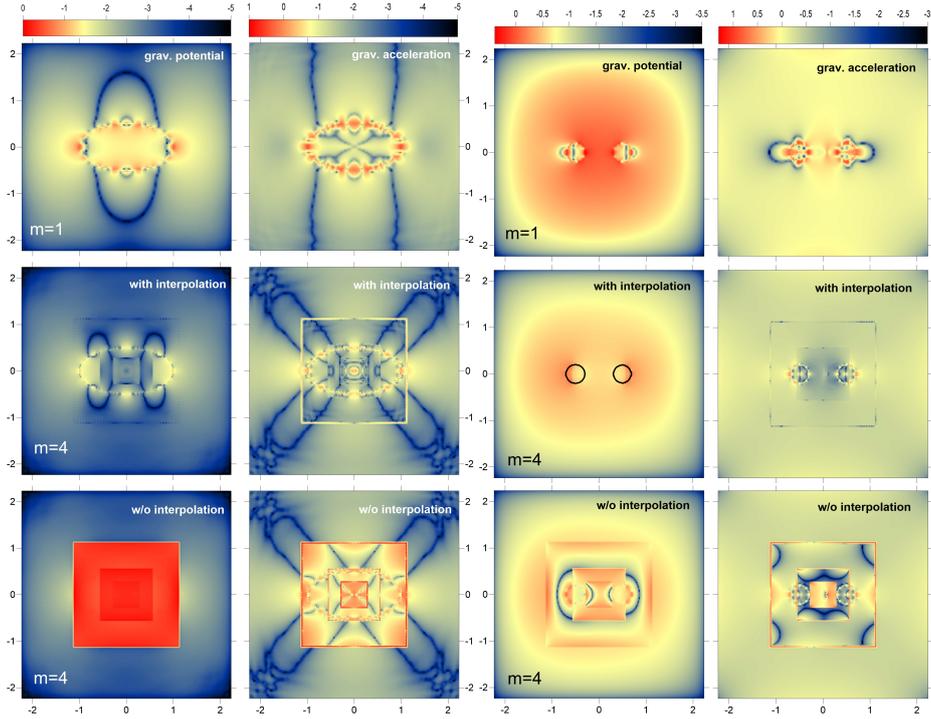


Fig. 3. Gravitational potential and acceleration on nested grids in the OiCG method. Left pair of columns: relative error in the calculation of the gravitational potential (first column) and gravitational acceleration (second column) for an oblate ellipsoid with the semi-major axis ratio 0.5:1.0:1.0. The horizontal bars present the relative errors in percent (log units). The top row illustrates the errors obtained with no mesh refinement ($m = 1$). The middle and bottom rows show the errors for the case of four nested grids ($m = 4$). In particular, the middle row corresponds to a linear interpolation of the boundary potentials, whereas the bottom row presents the errors when a simple extrapolation is used. Right pair of columns: relative error in the gravitational potential (first column) and gravitational acceleration (second column) for a binary star, the spatial location of which is outlined by two circles. The left component is twice as massive as the right one.

Table 1. Errors in gravitational potential: ellipsoid (left) and binary star (right).

m	Max. err. [%]	Mean err. [%]	σ [%]	Max. err. [%]	Mean err. [%]	σ [%]
1	0.25	0.01	0.017	1.6	0.08	0.16
4	0.85/0.06	0.42/0.003	0.25/0.0045	0.84/1.0	0.23/0.25	0.14/0.09

Table 2. Errors in gravitational acceleration: ellipsoid (left) and binary (right).

m	Max. err. [%]	Mean err. [%]	σ [%]	Max. err. [%]	Mean err. [%]	σ [%]
1	2.6	0.058	0.145	9.95	0.86	0.47
4	25/1.23	0.52/0.042	1.45/0.08	55.5/3.66	1.1/0.37	1.7/0.23

in Fig. 2). In the latter case, the mesh interfaces can still be noticed as locations where the relative errors experience jumps or discontinuities.

The relative errors in the gravitational potential and gravitational acceleration for a binary star are shown in the right pair of columns in Fig. 3. This example is useful because it tests the method on a nonaxisymmetric mass distribution. The resulting trends are nevertheless similar – mesh refinement and linear interpolation of the boundary potentials help to improve the accuracy and reduce the errors. The corresponding errors relative to the analytic solution for both density configurations, as well as the standard deviation in the relative errors σ , are provided in Tables 1 and 2 for comparison. The numbers

separated by the slash correspond to the cases without and with a linear interpolation following Eq. (6), respectively, and σ represents the standard deviation in the errors.

3. Integral form of the gravitational potential and the convolution method

An alternative approach to computing the gravitational potential is to use the integral representation

$$\Phi(\mathbf{r}) = -G \iiint_V \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}', \quad (9)$$

where the integration is performed over the volume V occupied by the computational domain. On a discrete Cartesian computational mesh the gravitational potential $\Phi(x_i, y_j, z_k)$ in a cell (i, j, k) can be written in the following form

$$\begin{aligned}\Phi(x_i, y_j, z_k) &= -G \sum_{i', j', k'} \frac{M(x_{i'}, y_{j'}, z_{k'})}{\sqrt{(x_i - x_{i'})^2 + (y_j - y_{j'})^2 + (z_k - z_{k'})^2}} \\ &= -G \sum_{i', j', k'} M(x_{i'}, y_{j'}, z_{k'}) \mathcal{G}(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'}),\end{aligned}\quad (10)$$

where $M(x_{i'}, y_{j'}, z_{k'})$ is the mass in a numerical cell (i', j', k') with coordinates $(x_{i'}, y_{j'}, z_{k'})$, $\mathcal{G}(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'})$ is the inverse distance between cells (i, j, k) and (i', j', k') , and the summation is performed over all numerical cells.

Finding the gravitational potential for a numerical mesh with N^3 cells by direct summation requires $(2N)^6$ operations, which becomes numerically expensive for N greater than a few tens. Therefore, the convolution theorem is often invoked to compute the potential given by Eq. (10). The convolution theorem states that the sum of the following form (Binney et al. 1987)

$$h_k = \frac{1}{\sqrt{2K}} \sum_{k'=-N}^{k'=N-1} f_{k-k'} g_{k'} \quad (11)$$

can be computed by taking the Fourier transforms separately for functions $f_{k-k'}$ and $g_{k'}$, multiplying them in the Fourier space to form $\hat{h}_p = \hat{f}_p \hat{g}_p$, and finally taking the inverse Fourier transform of \hat{h}_p . For a two-dimensional grid, for example, this requires $2N[6 \log_2(2N) + 1]$ additions and multiplications to be compared to $(2N)^2$ operations for a direct evaluation of the sum (11), see Binney et al. (1987). We note that the classic Fourier transformation method to solve the Poisson equation (e.g., Press et al. 1992) is inapplicable to Eq. (10). This is because this method is used to turn a system of discretized Poisson equations into an algebraic expression in the Fourier space, whereas Eq. (10) is already an algebraic expression, the value of which can be found most efficiently using the convolution theorem.

To apply the convolution theorem to the gravitational potential given by Eq. (10), certain manipulations with the mass M and inverse distance \mathcal{G} have to be made. The convolution theorem is applicable to periodic functions, whereas M and \mathcal{G} may not be periodic. This problem can be solved by doubling the computational domain in each coordinate direction and assigning zero values to M on this extended mesh (note that M on the original domain is kept intact). Both \mathcal{G} and M can then be made periodic on the extended domain by rearranging the extended mesh as shown in Fig. 4 (see also, e.g., Hockney & Eastwood 1988). We note that the zero-padding of the ghost domains makes the mass distribution in the active domain effectively isolated, meaning that this method can be only applied to isolated mass distributions and not to periodic ones.

Furthermore, one can notice that the inverse distance \mathcal{G} formally turns into infinity when $i = i'$, $j = j'$, and $k = k'$. This problem can be solved by introducing a smoothing length ϵ_{sm} , so that the inverse distance takes the following form

$$\mathcal{G} = \frac{1}{\sqrt{(x_i - x_{i'})^2 + (y_j - y_{j'})^2 + (z_k - z_{k'})^2 + \epsilon_{\text{sm}}^2}}. \quad (12)$$

However, there is always an uncertainty as to what value is to assign to ϵ_{sm} (Müller et al. 2012). A way to circumvent the problem is to evaluate the input to the potential in the cell $(i = i'$,

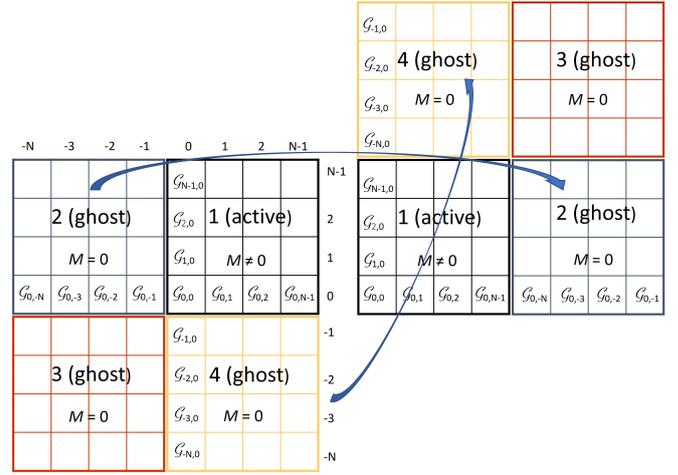


Fig. 4. Schematic illustration of the mesh doubling and rearrangement in the convolution method. The original active region 1 is doubled in each coordinate direction to form three ghost regions (2–4) that are filled with zero masses ($M = 0$). The doubled mesh is rearranged as shown by the arrows to make the inverse distance G and mass M periodic functions on the new domain.

$j = j'$, $k = k'$) from the mass distribution within the cell itself. If we assume that the density ρ is constant within the cell, then the corresponding potential can be found as (see Eq. (9))

$$\begin{aligned}\Phi(x_i = x_{i'}, y_j = y_{j'}, z_k = z_{k'}) \\ = -G \rho(x_i, y_j, z_k) \int_{-\frac{dx}{2}}^{\frac{dx}{2}} \int_{-\frac{dy}{2}}^{\frac{dy}{2}} \int_{-\frac{dz}{2}}^{\frac{dz}{2}} \frac{dx' dy' dz'}{\sqrt{x'^2 + y'^2 + z'^2}},\end{aligned}\quad (13)$$

where dx , dy , and dz are the sizes of the cell in each coordinate direction. We also used variable substitutions of the following form: $x_i - x_{i'} = x'$, $y_j - y_{j'} = y'$, and $z_k - z_{k'} = z'$. The integral in Eq. (13) can be found either numerically or analytically (Macmillan 1985; Stücker et al. 2020). We note that the assumption of constant ρ may be relaxed and the input to the potential from the cell itself can also be computed for piecewise-linear density profiles (Stücker et al. 2020).

We emphasize that the convolution method does not require us to precompute the boundary values of the gravitational potential at the coarsest grid (using, e.g., the multipole expansion) because of the absence of spatial derivatives in Eq. (10). This can be considered as a clear advantage compared to numerical solutions of the Poisson equation. Nevertheless, the practical applicability of the method described in this section has been limited to Cartesian meshes with grid cells of similar size in each coordinate direction (equidistant grids). This limitation is dictated by the fast Fourier transform (FFT), which was developed for the case of equidistant grids. There are extensions of the FFT methods to nonequidistant grids (e.g., Potts & Tasche 2008), but in our experience they usually work notably slower than the classic FFT methods. The convolution method can be also applied to finding the gravitational potential of disk-like configurations on the polar grid (r, ϕ) with a logarithmic scaling in the radial direction. In this case, the introduction of a new variable $u = \ln r$ helps to map the polar grid onto the equidistant Cartesian grid, so that the classic convolution theorem can be used (Binney et al. 1987). However, such cases are rather specific and for a three-dimensional cylindrical grid (z, r, ϕ) the above described method of variable substitution does not work. For 3D cylindrical coordinates, the convolution theorem can be applied only along the ϕ

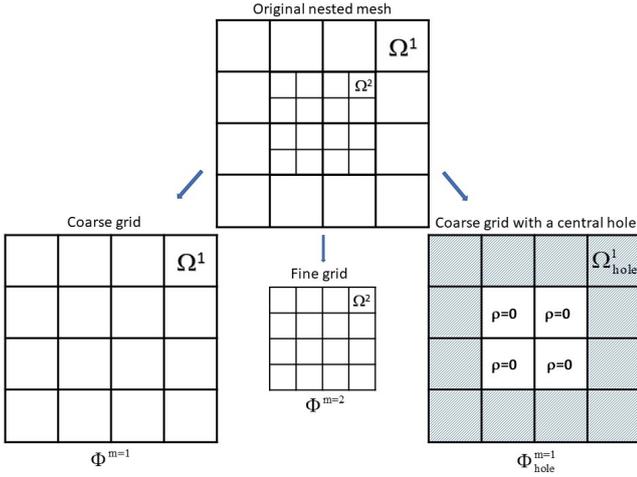


Fig. 5. Decomposition of two nested grids (top) into three individual grids with cells of equal size to which the classic FFT method can be applied: a coarse grid (Ω^1 , bottom left), refined grid (Ω^2 , bottom center), and coarse grid with a cavity (Ω^1_{hole} , bottom right).

and z -directions, while in the radial direction the direct summation has to be used, which considerably decelerates the method. For 3D spherical coordinates the situation is even worse since the convolution theorem works only along the ϕ -coordinate.

4. Modification of the convolution method for nested grids

In this section, we describe a modification to the convolution method that can be applied to nested Cartesian grids. A nested grid with only two levels of refinement ($\Omega^1 : \Omega^2$) is used for simplicity to illustrate the method but the procedure can easily be generalized to any level of refinement.

Figure 5 presents an example of the mesh decomposition, which is used to compute the gravitational potential on nested grids. The original mesh (top) is split into three regular grids to which the FFT method can be applied: a coarse $m = 1$ grid (Ω^1 , bottom left), a refined $m = 2$ grid (Ω^2 , bottom center), and a coarse $m = 1$ grid with a central cavity (Ω^1_{hole} , bottom right). The physical size of the cavity is equal to that of the refined $m = 2$ grid. We note that the density ρ in the cavity is set equal to zero. The grid with a central cavity accounts for the gravitational potential in the refined $m = 2$ grid created by the matter that lies in the hatched cells outside of this grid. Indeed, if we calculate the potentials on the coarse and refined grids, and then simply add them up, the procedure would not be complete. This is because the calculation of the potential on the refined grid does not take into account the contribution of the matter that lies outside this grid.

The solution procedure consists of three steps: (1) calculating the potential $\Phi^{m=2}$ on the refined $m = 2$ grid, (2) calculating the potential $\Phi^{m=1}$ on the coarse $m = 1$ grid, and (3) calculating the potential $\Phi^{m=1}_{\text{hole}}$ on the coarse $m = 1$ grid with a central cavity. An additional substep includes an interpolation of $\Phi^{m=1}_{\text{hole}}$ to the centers of the refined $m = 2$ grid as, for example, shown in Fig. 6. The resulting interpolated potential $\tilde{\Phi}^{m=1}_{\text{hole}}$ is used in the construction of the final solution. Other more sophisticated interpolation procedures can also be devised and may improve the accuracy of the final solution.

The potential on the original nested mesh is then constructed as illustrated in Fig. 7. The gravitational potential on the coarse

Interpolation of $\Phi^{m=1}_{\text{hole}}$ onto the refined Ω^2 grid

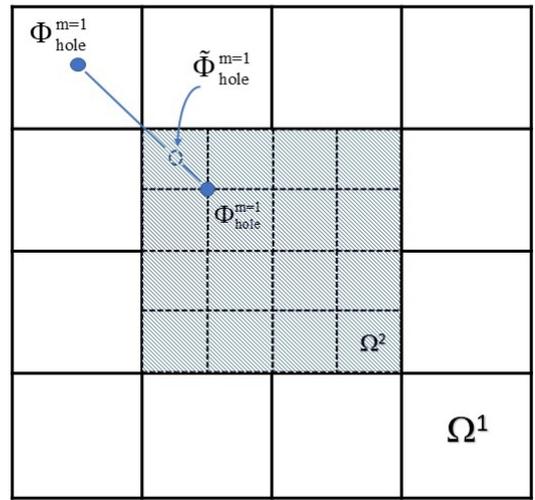


Fig. 6. Interpolation of the gravitational potential $\Phi^{m=1}_{\text{hole}}$, obtained on the coarse $m = 1$ grid with a central cavity, on the refined $m = 2$ grid highlighted with hatches. The resulting potential $\tilde{\Phi}^{m=1}_{\text{hole}}$ is shown with the arrow.

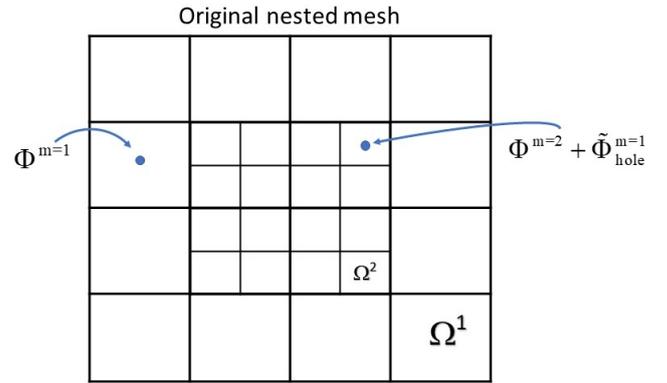


Fig. 7. Construction of the final solution on the original nested mesh ($\Omega^1 : \Omega^2$).

$m = 1$ grid is set equal to $\Phi^{m=1}$, while the potential on the refined $m = 2$ grid is set equal to the sum of $\Phi^{m=2}$ and $\tilde{\Phi}^{m=1}_{\text{hole}}$. We note that each of the three steps is independent and can be done in parallel, unlike the outside-in solution procedure outlined in Sect. 2. This procedure can be extended to any number of nested grids as follows

$$\begin{aligned} \Phi(\Omega^1) &= \Phi^{m=1}, \\ \Phi(\Omega^2) &= \Phi^{m=2} + \tilde{\Phi}^{m=1}_{\text{hole}}, \\ \Phi(\Omega^3) &= \Phi^{m=3} + \tilde{\Phi}^{m=1}_{\text{hole}} + \tilde{\Phi}^{m=2}, \\ \Phi(\Omega^n) &= \Phi^{m=n} + \sum_{k=1}^{n-1} \tilde{\Phi}^{m=k}_{\text{hole}}, \end{aligned} \quad (14)$$

where $\Phi(\Omega^n)$ is the gravitational potential on the n -th nested grid. In the following section, we use four nested grids to test the proposed convolution method.

5. Comparison of the models

In this section, we carry out a comparison of the gravitational potential and acceleration calculated using the two considered

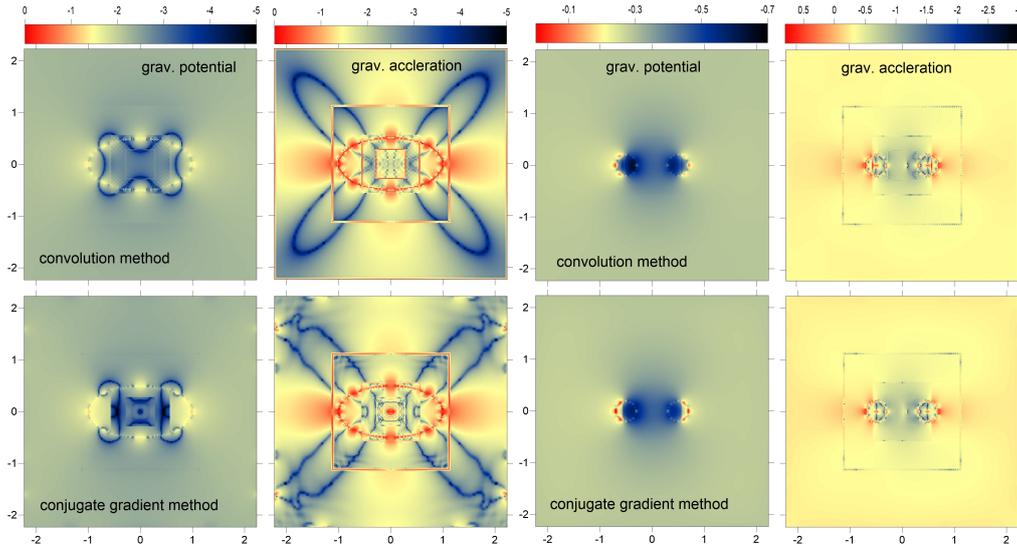


Fig. 8. Comparison of the relative errors in the gravitational potential and acceleration for an oblate ellipsoid (left pair of columns) and binary star (right pair of columns). The top row shows the results for the convolution method, while the bottom row presents the relative errors for the OiCG method. Four nested grids and 128 grid zones in each coordinate direction are used.

Table 3. Errors in gravitational potential: ellipsoid (left) and binary star (right).

Method	Max. err. [%]	Mean err. [%]	σ [%]	Max. err. [%]	Mean err. [%]	σ [%]
conj. gr.	0.06	0.003	0.0045	1.0	0.25	0.09
convolution	0.045	0.004	0.004	0.85	0.43	0.078

Table 4. Errors in gravitational acceleration: ellipsoid (left) and binary (right).

Method	Max. err. [%]	Mean err. [%]	σ [%]	Max. err. [%]	Mean err. [%]	σ [%]
conj. gr.	1.23	0.04	0.08	3.66	0.37	0.23
Convolution	1.74	0.05	0.12	9.4	0.38	0.29

models: the convolution method and OiCG method. As in Sect. 2, we consider an oblate ellipsoid and a binary star. We note, however, that here we use the multipole expansion with up to 6 multipole moments to find the gravitational potential at the outer boundary of the coarsest $m = 1$ grid. This makes the comparison more realistic because in real simulations the analytic solution is rarely available. In both cases, linear interpolation of the boundary potentials at the interfaces between the nested grids is applied. In the convolution method, the interpolation is only used to obtain the gravitational acceleration, while in the OiCG method the interpolation is also required to calculate the boundary potentials at the finer grids for the outside-in procedure to work.

We found that both methods are characterized by comparable accuracy, as can be seen from Fig. 8 for the case of four nested grids. The relative errors are provided in Tables 3 and 4 for convenience. For instance, the maximum error in the gravitational potential does not exceed 1.0% in the case of the binary star and is notably smaller for the ellipsoid. The mean errors are a few per mille (in %) in the case of the ellipsoid and just a fraction of a percent for the binary. As expected, the calculation of the gravitational acceleration shows larger relative errors and

the maximum values can reach 10% for the binary. Nevertheless, the mean errors are smaller than 1%. The standard deviation is also low, implying that large errors are rare. A closer look at Fig. 8 indicates that the interfaces between the nested grids are hardly seen in the errors of the gravitational potential, whereas the errors in the gravitational acceleration are often the highest at the position of the interfaces. This means that more accurate schemes for calculating the gravitational acceleration from the gravitational potential (rather than a simple six-point stencil) may be needed to reduce the error and improve the accuracy (e.g., Wang & Yen 2020).

Figure 9 presents the relative errors (in %) for the oblate ellipsoid as a function of radial distance from the coordinate center in the vertical $x - z$ plane. The errors have a wide spread but are mostly below 0.01% for the potential and 0.1% for the acceleration. There is little systematic difference between the two considered methods.

As another comparison test, we calculate the maximum and mean errors in the gravitational potential and acceleration of the oblate ellipsoid for nested grids ranging from $m = 1$ to $m = 10$. The results are shown in Fig. 10. The errors in the vertical $x - z$ plane passing through the coordinate center are only considered.

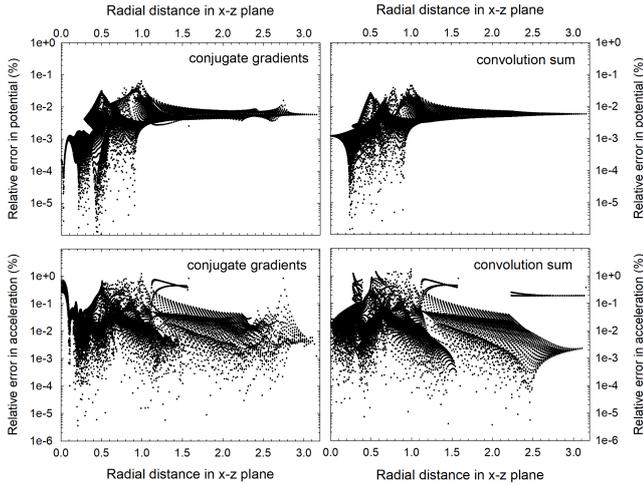


Fig. 9. Relative errors in the gravitational potential (top row) and gravitational acceleration (bottom row) as a function of radial distance from the coordinate center in the vertical $x-z$ plane. The case of the oblate ellipsoid is shown. Left and right columns present the results for the OiCG and convolution method, respectively.

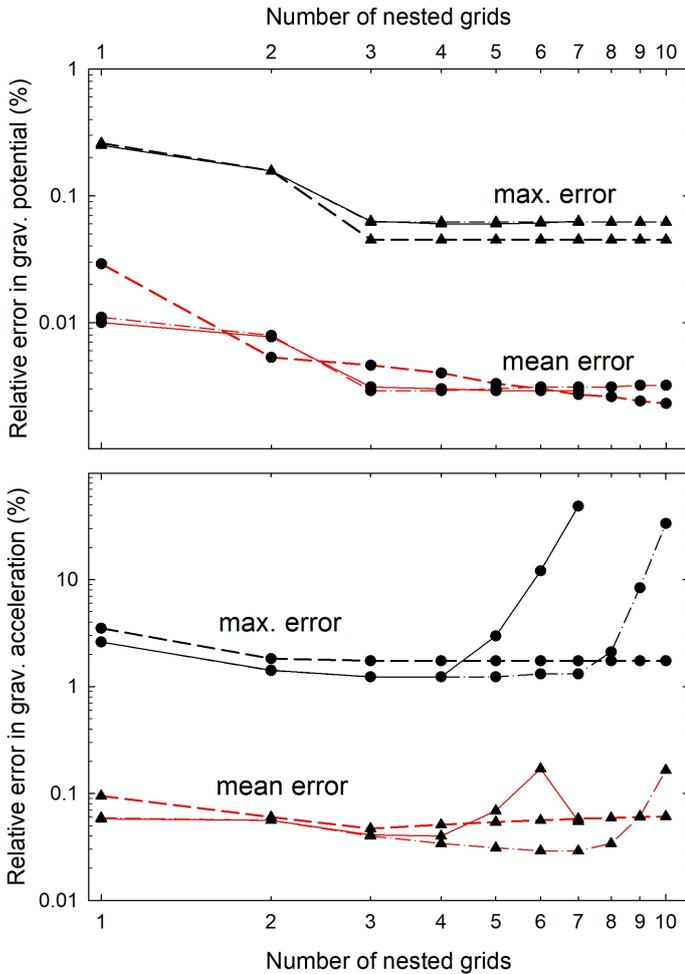


Fig. 10. Relative errors for the gravitational potential (top panel) and gravitational acceleration (bottom panel) for the oblate ellipsoid as a function of the number of nested grids. The black and red lines present the maximum and mean errors, respectively. The dashed lines correspond to the convolution method, while the solid and dash-dotted lines show the results for the OiCG method with the tolerance ϵ set equal to 10^{-6} and 10^{-8} , respectively.

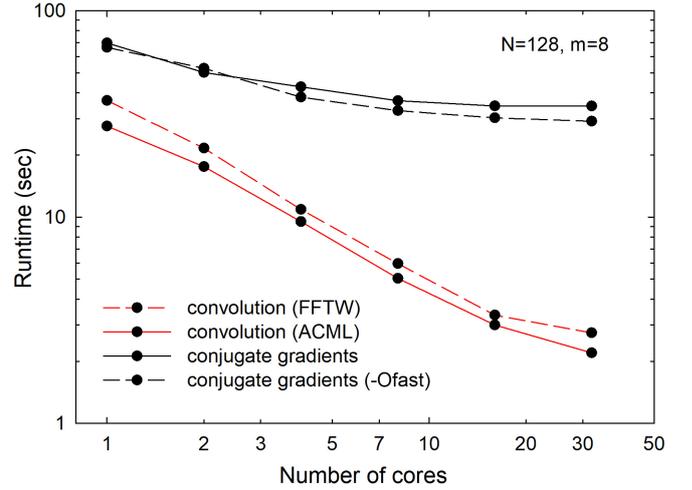


Fig. 11. Runtime vs. the number of cores in computing the gravitational potential of an oblate ellipsoid on $m = 8$ nested grids. Red and black lines correspond to the convolution method and OiCG method, respectively.

In the case of the OiCG method, we also vary the tolerance ϵ with which the solution is found by setting it to 10^{-6} and 10^{-8} (see Sect. 2). We note that the convolution method is not iterative and this procedure is not needed. Figure 10 indicates that both methods are characterized by errors in the gravitational potential of comparable magnitude. The convolution method produces relative errors in Φ that decline with increasing m , while the corresponding errors in the OiCG method saturate already at $m \geq 3$. An important difference between the two considered methods can be seen in the behavior of relative errors for the gravitational acceleration. While the errors in the convolution method saturate with increasing m , the errors in the OiCG method begin to grow. The turning point depends on the tolerance ϵ with which the iterations are controlled. We found that the deterioration occurs in the very central regions of the ellipsoid. This test problem reveals a potentially dangerous behavior of the OiCG method and requires further investigation in the future.

6. Performance comparison

In this section, we compare the two methods for computing the gravitational potential in terms of its performance on the Intel Xeon Platinum 8174 processors with 48 physical cores. We run the test in a loop that has 20 full cycles of the potential calculation in order to reduce the associated overheads and then divide the obtained runtime by a factor of 20. Figure 11 presents the results of our tests for the number of OpenMP threads ranging from 1 to 32, increasing by a factor of 2. The basic tests use the Intel/19 FORTRAN compiler without explicit optimization. Additionally, we run tests with the `-Ofast` optimization key. For the test configuration, we have chosen an oblate ellipsoid as in Sect. 5. The resolution is $N = 128$ cells in each direction and the number of nested grids is $m = 8$. Both values are based on our practical simulations of cloud core collapse and protoplanetary disk formation in the Solar metallicity environment, resulting in a numerical resolution of ≈ 1 au near the forming star. The tolerance ϵ for the OiCG method is set equal to 10^{-8} . To compute the FFTs, we used two external libraries: the AMD Core Math Library (ACML) and the FFTW¹ library.

¹ <https://fftw.org>

Table 5. Different device-host configurations.

Name	Device			Host	
	CUDA cores	GFLOPs/s (FP64)	CUDA version	Name	Physical cores
NVIDIA A40	10752	1169	11.6	AMD EPYC 7252 (×2)	16
NVIDIA V100	5120	7000	11.2	Intel Xeon E5-2650 v3 (×2)	20

Analysis of Fig. 11 indicates that the convolution method outperforms the OiCG method by more than an order of magnitude on multicore OpenMP applications. Even without parallelization, the convolution method is faster than the OiCG method by a factor of several. The optimization key `-Ofast` marginally improves the performance of the OiCG method but has little effect on the convolution method. The ACML library is found to be somewhat faster than that of FFTW.

An even better performance on the convolution method can be achieved when graphics processing units (GPUs) are utilized. While an extension of the Intel FORTRAN compiler can offer an interface with GPUs (hereafter, devices) through OpenMP target offload, CUDA by NVIDIA is widely accepted as faster in most cases and thus was the chosen method of implementation (Gayatri et al. 2019). This required the adoption of the NVIDIA FORTRAN compiler². Various NVIDIA devices were used to accelerate the computations of the gravitational potential on each nested mesh, and this was coupled with CPU (hereafter, host) calculations of the boundary conditions and the final assembly of the gravitational potential from potentials on each individual mesh (see Eq. (14)). The results presented here are for the NVIDIA A40 and NVIDIA V100, the parameters for which can be found in Table 5. While the host of the NVIDIA V100, the Intel Xeon E5-2650 v3, offers a maximum of 20 physical cores, for consistency and in keeping with the dimensions of the computational domain, results are presented to a maximum of 16 physical cores. To compute the FFTs, the CUDA library `cufft` and the `iso_c_binding` module for FORTRAN to C interoperability were used. As in the example shown in Fig. 11, an oblate ellipsoid was chosen, with a numerical resolution of $N = 128$ grid zones in each coordinate direction and $m = 8$ nested meshes. The results are presented in Fig. 12. The CPU-GPU coupled version can be seen to achieve acceleration by a factor of 10 compared to the pure CPU model when at full OpenMP host parallelization. During operations accelerated by the device, 75% of the time is spent performing FFT computation, 20% on transferring the data between the host and the device, and 5% on other auxiliary computations. In total when using GPUs, a factor of 200 times acceleration can be achieved when compared to the original OiCG method with CPU-only implementation (see the black lines in Fig. 11).

Finally, we tested the performance of the Generalized Minimal Residual method (GMRes) when computing the gravitational potential using the discretized form of the Poisson equation (see, Eqs. (1) and (2)). In our particular case, we applied the OiCG method outlined in Sect. 2 to find the potential on each grid. Intel/19 FORTRAN compiler and the MKL Intel library were used for this test problem. No explicit optimization was used. We did this additional test to illustrate how our method can outperform the “standard” methods offered by frequently used

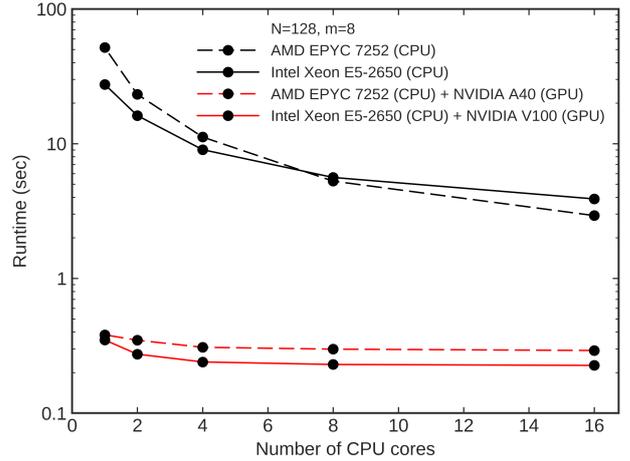


Fig. 12. Runtime vs. the number of CPU cores in computing the gravitational potential of an oblate ellipsoid on $m = 8$ nested grids. The red lines show the results of CPU plus GPU computations, while the black lines present the CPU model only.

numerical libraries. GMRes is an iterative method and, similar to the conjugate gradient method, it requires presetting a desired accuracy when calculating the gravitational potential. We set the tolerance in the relative residuals to $\epsilon = 10^{-6}$ to be consistent with the OiCG method. We found, however, that GMRes does not converge on $N = 128$ grid zones unless we reduce the tolerance ϵ . Therefore, we decreased the number of grid zones in each direction to $N = 64$. GMRes also required too many iterations to converge for nested grid levels $m \geq 7$, which may be related to the problem of divergence discussed in the context of Fig. 10. We, therefore, limited our test runs to $m = 6$. The runtime for the GMRes method for an oblate ellipsoid (again, using 20 loops of full potential calculations to reduce the overheads) is 29 s and this number weakly depends on the OpenMP parallelization. This value is comparable to the best performance of the OiCG method but is much worse than the performance of the convolution method. Considering that the test was done on a twice lower resolution and with a smaller number of nested grids, the GMRes methods of the Intel MKL library is clearly inferior to both methods presented in this work. The GMRes method can be accelerated by reducing the tolerance ϵ , but we found that this leads to degraded accuracy in the calculation of the potential on the innermost nested grid and is therefore not recommended.

7. The dipole problem and the modified convolution method

The original convolution method described in Sects. 3 and 4 can be modified if specific cases are encountered. We illustrate this using the following idealized example problem. We consider an electric dipole with a dipole moment $p = qd$, where d is the distance between unit charges $+q$ and $-q$. For simplicity, we place the dipole on a nested mesh with only two nested grids Ω^1 and Ω^2 . Each grid has a numerical resolution of $N = 32^3$. The unit charges are located near the geometrical center of the fine Ω^2 grid and their spatial position is chosen so as to cancel each other on the course Ω^1 grid. This means that on the coarse grid, the net charge is zero. The resulting electric field geometry is illustrated in Fig. 13. While the field lines show a regular and correct pattern on the fine Ω^2 grid, the field is artificially suppressed on the course Ω^1 grid because of the charge cancellation. The problem

² <https://developer.nvidia.com/cuda-fortran>

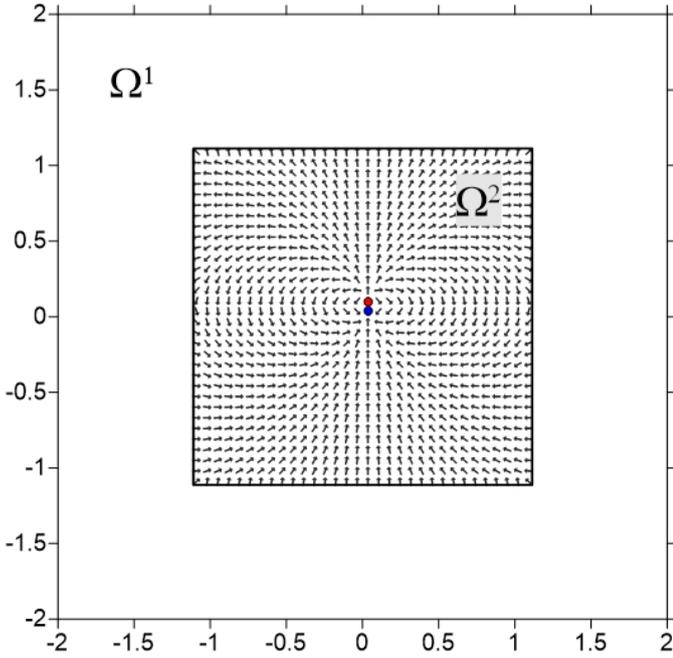


Fig. 13. Electric field vectors of a dipole schematically shown by the red and blue circles near the coordinate center. The original convolution method is applied to construct the field. We note that the field is present and regular on the fine Ω^2 grid but is absent on the coarse Ω^1 grid. This occurs because of the cancellation of the charges when they are projected from the fine to the coarse grid in the original convolution method.

was outlined and successfully addressed in [Matsumoto \(2007\)](#); [Truelove et al. \(1998\)](#); [Stone et al. \(2020\)](#).

Our augmentation of the convolution method is described in detail below. We do not consider possible updates to the OiCG method as this method is used mainly for comparison purposes. Figure 14 presents an example of the mesh decomposition, which can be used to solve the aforementioned dipole problem on two nested grids. In particular, the solution procedure is split into three steps. The first two steps are identical to the original method: the potential on a fine $m = 2$ grid (Ω^2 , bottom center) and on a coarse $m = 1$ grid with a central cavity (Ω_{hole}^1 , bottom right) is found. The third step, however, involves expanding the fine grid to the physical size of the coarse grid as shown in the bottom left panel. The resulting doubled fine grid Ω_{dbl}^2 has the same physical size as the coarse grid Ω^1 but the size of individual numerical cells is similar to that of Ω^2 . As a result, Ω_{dbl}^2 has twice as many cells in each coordinate direction. The additional layers of cells are then filled with zeros (white cells in the bottom left panel), while the original physical size of the fine $m = 2$ grid (indicated by the light-blue shade) retains the original values from the fine Ω^2 grid. The described algorithm can be written as follows

$$\begin{aligned}\Phi(\Omega^1) &= \Phi_{\text{hole}}^{m=1} + \tilde{\Phi}_{\text{dbl}}^{m=2}, \\ \Phi(\Omega^2) &= \tilde{\Phi}_{\text{hole}}^{m=1} + \Phi^{m=2}.\end{aligned}\quad (15)$$

The introduction of the doubled fine grid allows us to avoid the charge cancellation on the coarse grid and retrieve the correct behavior of the electric field lines across the grid interface as illustrated in the top panel of Fig. 15. Now, the field lines are continuous and are present on the entire nested mesh (and not only on the fine mesh as was in the original method). Finding the potential on the coarse grid Ω^1 is now a more complex

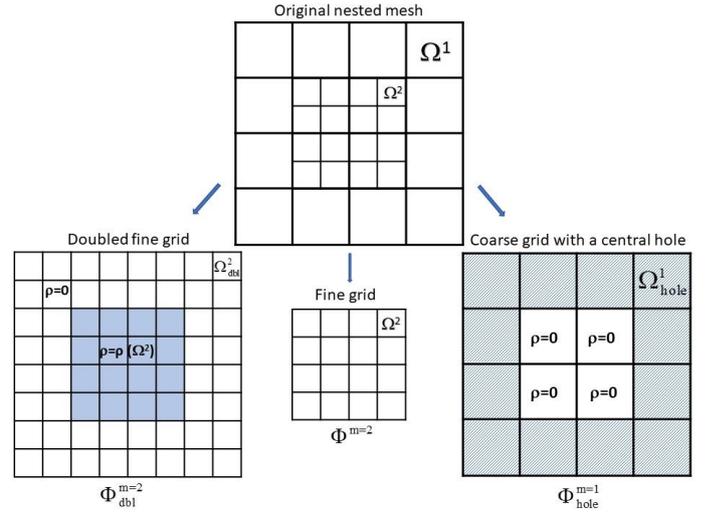


Fig. 14. Decomposition of two nested grids (top) into three individual grids in the modified convolution method: a fine $m = 2$ grid (Ω^2 , bottom center), a coarse $m = 1$ grid with a central cavity (Ω_{hole}^1 , bottom right), and a doubled fine grid $m = 2$ (Ω_{dbl}^2 , bottom left). The density in the latter grid is set to zero everywhere except the original physical size of the fine $m = 2$ grid (shown with a light blue shade), where the density inherits the values from the fine $m = 2$ grid, $\rho = \rho(\Omega^2)$.

procedure than it was in the original method (see Eq. (14)). The potential on the doubled fine grid $\Phi_{\text{dbl}}^{m=2}$ should further be reduced to the coarse $m = 1$ grid (this mathematical operation is denoted by tilde) and added to the potential on the coarse grid with a hole $\Phi_{\text{hole}}^{m=1}$. We note that the above procedure can in principle be extended to any number of nested grids (see Appendix B). For instance, the bottom panel in Fig. 15 presents the electric field of a dipole on three nested grids. As can be seen, the field lines are present and smooth across all grids. However, the modified convolution method suffers from computational overheads and we discuss this issue in more detail in Appendix B.

We want to make the following notes in this context. The electric dipole problem considered above is an extreme case, which helped us to highlight the problem. A close analog of an electric dipole in astrophysics is a binary star. However, masses are never cancelled out but add up to produce the common gravitational field. We have already considered a wide-separation binary system in Sect. 3 and our original convolution method has revealed good performance on this nonaxisymmetric test problem (see also Appendix C), but the situation with close binaries may be different. The dipole component of the gravity force of a close binary star falls off with distance faster ($\sim 1/r^3$) than the basic field component ($\sim 1/r^2$). This means that special care should be taken when computing the potential of a binary star in its vicinity, where the dipole moments are important, but at distances much greater than the binary separation the dipole contribution may be insignificant.

We illustrate the problem in Fig. 16 by computing the gravitational field of a close binary in the original and modified convolution methods. We consider a tight binary with companions located next to each other on the finest grid (Ω^2 for two nested grids and Ω^3 for three nested grids). The densities in the neighboring cells occupied by the binary are set to $\rho = 1$ and $\rho = 2$, implying the companion mass ratio 1:2, and are zeroed everywhere else in the computational domain. We then calculate the relative error (in per cent) between the two methods, keeping in mind that the modified method computes the dipole moments accurately, but the original method does not. The top panel

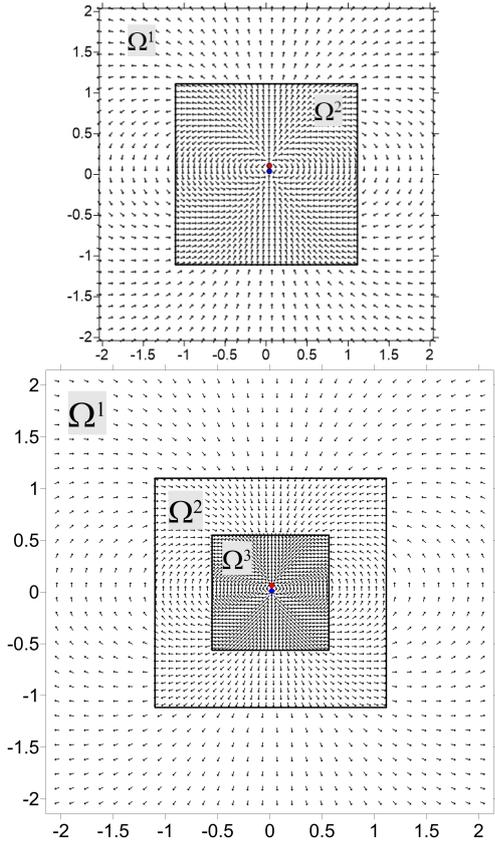


Fig. 15. Electric field vectors of a dipole. The dipole is located at the center of the finest grid and is shown schematically by the red and blue circles. The modified convolution method is applied to construct the field (see Fig. 13 for comparison with the original method). The top panel presents the case of two nested grids, while the bottom panel corresponds to three nested grids. The black squares identify the grid interfaces. Note that the field is present and continuous across all nested grids Ω^1 , Ω^2 and Ω^3 .

corresponds to the binary placed in the center of the finest grid. The bottom panel presents the case when the binary is located near the upper interface between Ω^1 and Ω^2 when two nested grids are considered and near the interface of Ω^2 and Ω^3 for the case of three nested grids. Each level of refinement has $N = 128^3$ grid cells.

We first consider two nested grids shown in the left column of Fig. 16. Both methods produce identical solutions on the refined Ω^2 grid and, hence, the relative error is zero there. The deviations between the two methods occur on the coarse Ω^1 grid, for which the original method does not reproduce correctly the dipole moments, and the errors depend on the spatial location of the binary. The errors are highest when the binary is near the grid interface (bottom panel) and can reach several per cent on the coarse grid. On the other hand, the errors stay below 1% if the binary is in the center of the fine grid (top panel). The case of three nested grids is shown in the right column of Fig. 16. Here, the comparison is made between the modified convolution method that correctly computes the dipole moments only on the finer Ω^2 and Ω^3 grids, but not on the coarsest Ω^1 grid, and the modified convolution method that correctly computes the dipole moments on all three nested grids. Both cases produce identical potentials in the Ω^2 and Ω^3 grids and the relative error is zero there, but they deviate in the outermost Ω^1 grid. However, the errors stay low and do not exceed 0.35%, regardless of the spatial position of the binary. If we degrade the numerical

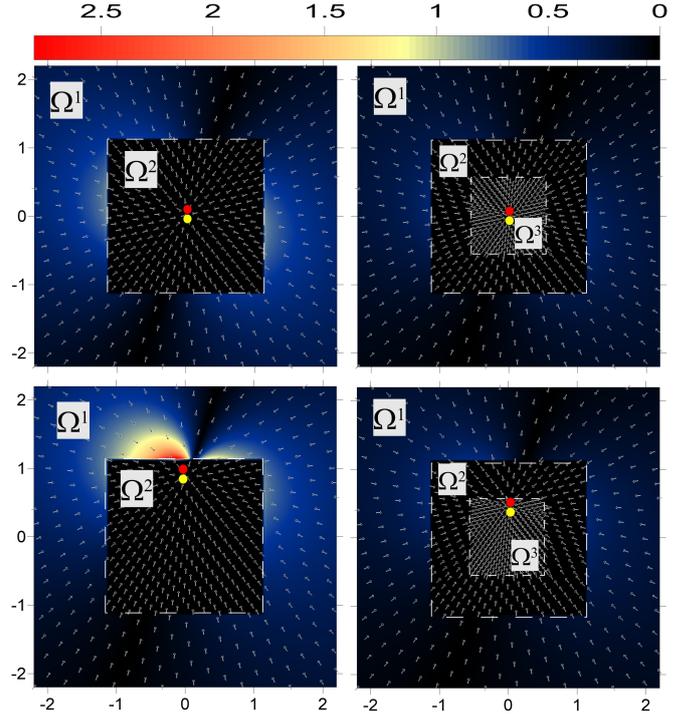


Fig. 16. Relative errors between the gravitational potentials of a close binary computed by the original and modified convolution methods. The horizontal bar shows the errors in percent. Two mesh setups are used for this test problem: left column – two nested grids and right column – three nested grids. The grid interfaces are shown by the dashed lines for convenience. The binary position is schematically indicated by the red and yellow circles. The arrows show the gravitational field in the modified convolution method. Note that the field lines are continuous across the grid interfaces.

resolution to $N = 64^3$ and $N = 32^3$, the maximum errors slightly increase to 0.7 and 1.5%, which can be expected with the overall decrease in accuracy of the method as the resolution drops.

These test examples demonstrate the utility of the updated convolution method in correctly reproducing the dipole field of a binary across the coarse-fine grid interfaces. But it also shows that the error quickly drops with distance from the binary. A similar behavior was found when we varied the separation between the binary components by several grid cells. Consideration between the associated numerical costs and desired accuracy is needed if the dipole moments over a larger number of grid interfaces are to be computed. For instance, the convolution method was originally developed for astrophysical problems involving the gravitational collapse of an individual prestellar core. The star(s) and the circumstellar disk that form in the process are located within two-three innermost grids and the rest is filled with a rarified infalling envelope. The correct calculation of the dipole moments over the three innermost nested grids should be sufficient in this case. We also note that when applied to a smooth density distribution, e.g., an oblate ellipsoid, or wide separation binaries we did not find significant differences between the original and modified convolution methods (see Appendix C for details).

Finally, we performed a convergence test on the modified convolution method, in which we varied the resolution of each nested grid from $N = 16$ to $N = 128$, but kept the number of nested grid fixed at $m = 4$. An oblate ellipsoid with the ratio of semi-axes 0.5:1.0:1.0 was used for this purpose. The results are shown in Fig. 17. When the mean numerical errors are

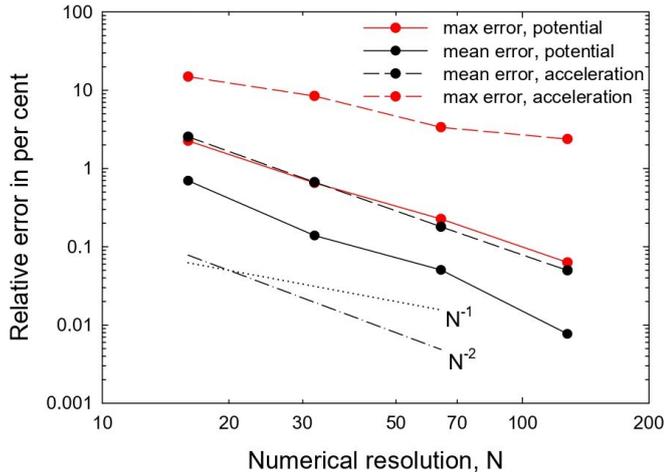


Fig. 17. Convergence test in the modified convolution method. Shown are the mean (black lines) and maximum (red lines) errors in the gravitational potential and acceleration as a function of numerical resolution N . The dotted and dash-dotted lines show the linear and quadratic convergence for convenience.

considered, the method demonstrates second-order accuracy in both the gravitational potential and acceleration. The maximum error is characterized by second-order convergence only for the gravitational potential, while the maximum error in acceleration reveals first-order convergence. We attribute this decrease in accuracy to boundary effects between the nested grids, where the adopted linear interpolation procedure of the gravitational potential is only first-order accurate. The development of more accurate interpolation schemes across the grid interfaces should be our next task in future works.

8. Conclusions

In this paper, we presented an extension of the convolution method for calculating the gravitational potential on regular and equidistant grids, which can be applied to the practical case of nested grids. We compared the performance of the convolution method against the OiCG method using two test problems: an oblate ellipsoid and a binary star, for which the analytic solution of the gravitational potential and acceleration are known. We found that both methods are characterized by comparable computational errors, which do not exceed a few percent for the gravitational acceleration and are notably lower for the gravitational potential.

The convolution method, however, has several clear advantages over the iterative OiCG method. First, the convolution method does not require the precalculation of the gravitational potential at the boundaries of the coarsest $m = 1$ grid. Second, the convolution method permits decomposition of the nested grids into individual subgrids, followed by efficient computation of the potential on each subgrid in a parallel mode. Third, the convolution method does not require a preset tolerance ϵ , with which the iterations in the OiCG method are controlled. We found that the OiCG method reveals an increase in the errors with increasing m , and the magnitude of this effect depends on the predefined value of ϵ .

A comparison of runtimes of the OiCG and convolution methods demonstrates a clear advantage of the latter method, which can outperform the former by more than a factor of 10 on the CPUs with OpenMP parallelization. When GPUs are utilized, the convolution method can be accelerated by another

factor of 10. The GMRes method offered by the Intel MKL library is found to be inferior in speed performance as compared to both methods presented in this work. The convolution method revealed an overall second-order convergence, except for the maximum error at the coarse-fine grid interfaces where the convergence is linear, in accordance with the adopted linear interpolation of the gravitational potential across the grid interface.

The disadvantage of the convolution method lies in the requirement to interpolate the gravitational potential from the coarse grids onto the refined grids. This step can be improved by devising more sophisticated interpolation procedures. When tightly spaced gravitating sources are present at the deepest levels of refinement, for example, close binary/multiple stars surrounded by a circumstellar disk, a modification to the convolution method is required to correctly calculate the dipole moments on the coarser grids. The modification introduces computational overheads, which may be substantial for a large number of nested grids, but acceptable accuracy can already be reached when the method is applied to each pair or triplet of multiple nested grids.

Acknowledgements. We are thankful to the referee for the comments and suggestions that helped to improve the manuscript. This work was supported by the FWF project I4311-N27 (E. I. V. and J. M.) and RFBR project 19-51-14002 (I. K. and V. E.). The simulations were performed on the VSC Vienna Scientific Cluster. We thank Oliver Hahn for pointing to an analytic solution of Eq. (13). The code that uses the convolution theorem for calculating the gravitational potential on CPUs and GPUs can be employed on reasonable request.

References

- Binney, J., Tremaine, S., & Ostriker, J. 1987, *Galactic Dynamics, Princeton series in astrophysics* (Princeton: Princeton University Press)
- Bodenheimer, P., Laughlin, G. P., Rózycka, M., & Yorke, H. W. 2007, *Numerical Methods in Astrophysics: An Introduction* (Boca Raton: CRC Press)
- Chandrasekhar, S. 1969, *Ellipsoidal Figures of Equilibrium* (New Haven: Yale University Press)
- Feng, W., Guo, Z., Lowengrub, J. S., & Wise, S. M. 2018, *J. Comput. Phys.*, **352**, 463
- Gayatri, R., Yang, C., Kurth, T., & Deslippe, J. 2019, in *Accelerator Programming Using Directives*, eds. S. Chandrasekaran, G. Juckeland, & S. Wienke (Cham: Springer International Publishing), 75
- Guillet, T., & Teyssier, R. 2011, *J. Comput. Phys.*, **230**, 4756
- Hennebelle, P. 2018, *A&A*, **611**, A24
- Hockney, R. W., & Eastwood, J. W. 1988, *Computer Simulation Using Particles* (Boca Raton: CRC Press)
- Jackson, J. D., & Fox, R. F. 1999, *Am. J. Phys.*, **67**, 841
- Kulikov, I., & Vorobyov, E. 2016, *J. Comput. Phys.*, **317**, 318
- Larson, R. B. 2003, *Rep. Prog. Phys.*, **66**, 1651
- Lebreuilly, U., Commerçon, B., & Laibe, G. 2020, *A&A*, **641**, A112
- Machida, M. N., Matsumoto, T., Tomisaka, K., & Hanawa, T. 2005, *MNRAS*, **362**, 369
- Macmillan, W. D. 1985, *Theory of the Potential* (New York: Dover Publications Inc.)
- Masunaga, H., Miyama, S. M., & Inutsuka, S.-I. 1998, *ApJ*, **495**, 346
- Matsumoto, T. 2007, *PASJ*, **59**, 905
- Matsumoto, T., & Hanawa, T. 2003, *ApJ*, **583**, 296
- Müller, T. W. A., Kley, W., & Meru, F. 2012, *A&A*, **541**, A123
- Potts, D., & Tasche, M. 2008, *J. Comput. Appl. Math.*, **222**, 655
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical Recipes in FORTRAN. The Art of Scientific Computing* (Cambridge: Cambridge University Press)
- Ruffert, M., & Mueller, E. 1990, *A&A*, **238**, 116
- Stone, J. M., & Norman, M. L. 1992, *ApJS*, **80**, 753
- Stone, J. M., Tomida, K., White, C. J., & Felker, K. G. 2020, *ApJS*, **249**, 4
- Stücker, J., Hahn, O., Angulo, R. E., & White, S. D. M. 2020, *MNRAS*, **495**, 4943
- Tomida, K., Okuzumi, S., & Machida, M. N. 2015, *ApJ*, **801**, 117
- Truelove, J. K., Klein, R. I., McKee, C. F., et al. 1998, *ApJ*, **495**, 821
- Vorobyov, E. I., Khaibrakhmanov, S., Basu, S., & Audard, M. 2020, *A&A*, **644**, A74
- Wang, H.-H., & Yen, C.-C. 2020, *ApJS*, **247**, 2

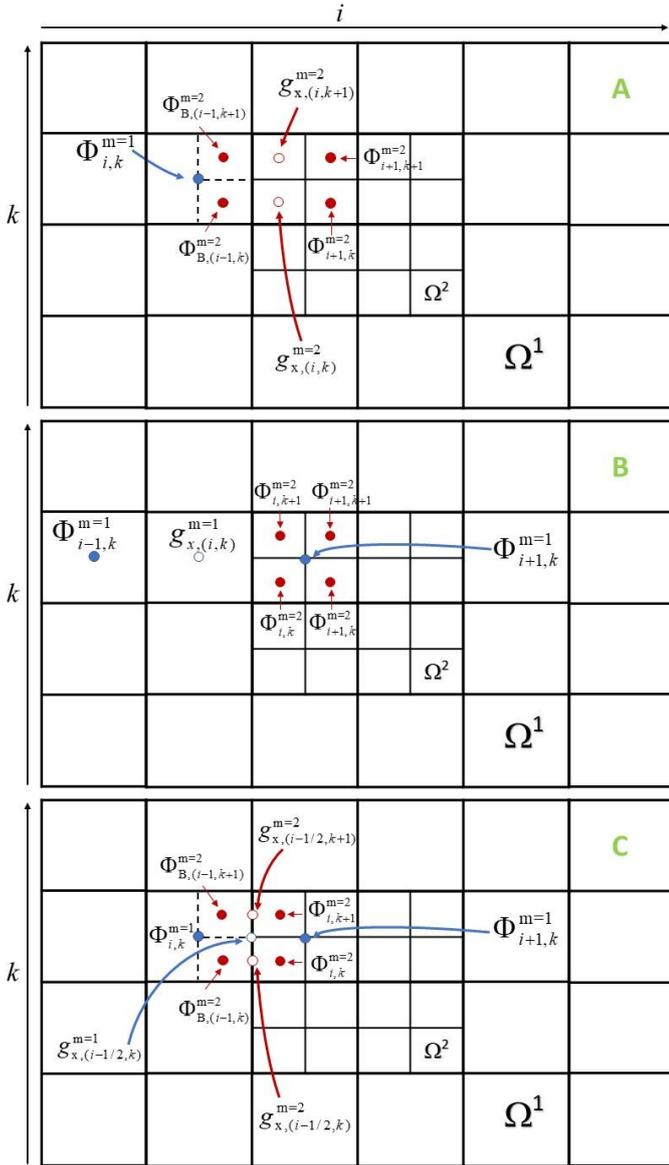


Fig. A.1. Three methods to calculate the gravitational acceleration at the cells adjacent to the coarse-fine grid interface. **Top panel.** Method A schematically illustrates a scheme to calculate the acceleration at the adjacent fine cells. **Middle panel.** Method B presents a scheme to calculate the acceleration at the adjacent coarse cells. **Bottom panel** Method C shows yet another scheme to calculate the acceleration at the adjacent coarse cells, which fulfills the Gauss theorem (see the text for details).

Appendix A: Gravitational acceleration at the coarse-fine grid interface

The components of gravitational acceleration \mathbf{g} across the coarse-fine grid interface can be calculated using three methods illustrated in Figure A.1. We take two nested grids, Ω^1 and Ω^2 , for simplicity and consider two dimensions, with the indices i and k corresponding to the x - and y -coordinates, respectively. The schemes can straightforwardly be extended to three dimensions.

Panel A graphically describes the case of the fine grid cells that are adjacent to the coarse grid (hereafter, method A). The x -component of the gravitational acceleration at the fine-to-coarse

grid interface is calculated as

$$g_{x,(i,k)}^{m=2} = -\frac{\Phi_{i+1,k}^{m=2} - \Phi_{B,(i-1,k)}^{m=2}}{2 dx_{m=2}}, \quad (\text{A.1})$$

where $\Phi_{B,(i-1,k)}^{m=2}$ are the boundary values of the potential on the fine grid found using the interpolation scheme described in Fig. 2 of Sect. 2 and $dx_{m=2}$ is the size of the fine grid cell.

Panel B presents the opposite case of the coarse grid cells that are adjacent to the fine grid (hereafter, method B). The gravitational acceleration at the coarse-to-fine grid interface can be calculated as

$$g_{x,(i,k)}^{m=1} = -\frac{\Phi_{i+1,k}^{m=1} - \Phi_{(i-1,k)}^{m=1}}{2 dx_{m=1}}, \quad (\text{A.2})$$

where $\Phi_{(i-1,k)}^{m=1}$ is found using an arithmetic average of $\Phi_{(i,k)}^{m=2}$, $\Phi_{(i+1,k)}^{m=2}$, $\Phi_{(i,k+1)}^{m=2}$, and $\Phi_{(i+1,k+1)}^{m=2}$. Here, $dx_{m=1}$ is the size of the coarse grid cells.

Panel C presents a more subtle and physically motivated method (hereafter, method C) for calculating the gravitational acceleration at the coarse-to-fine grid interface (see also Feng et al. 2018). Here, we first calculate the x -components of gravitational acceleration at the coarse-fine grid interface using the coarse and fine grid values separately

$$g_{x,(i-1/2,k)}^{m=1} = -\frac{\Phi_{i+1,k}^{m=1} - \Phi_{(i,k)}^{m=1}}{dx_{m=1}}, \quad (\text{A.3})$$

$$g_{x,(i-1/2,k)}^{m=2} = -\frac{\Phi_{i,k}^{m=2} - \Phi_{B,(i-1,k)}^{m=2}}{dx_{m=2}}, \quad (\text{A.4})$$

$$g_{x,(i-1/2,k+1)}^{m=2} = -\frac{\Phi_{i,k+1}^{m=2} - \Phi_{B,(i-1,k+1)}^{m=2}}{dx_{m=2}}. \quad (\text{A.5})$$

We note that these values of gravitational acceleration are defined at the cell interfaces and not at the cell centers. We then require that the gravitational accelerations at the coarse-fine grid interface be equal when using the coarse and fine grids separately

$$g_{x,(i-1/2,k)}^{m=1} = \frac{1}{2}(g_{x,(i-1/2,k)}^{m=2} + g_{x,(i-1/2,k+1)}^{m=2}). \quad (\text{A.6})$$

From the last expression it is straightforward to find the value of $\Phi_{i+1,k}^{m=1}$ and then employ Equation (A.2) to compute the gravitational acceleration $g_{x,(i,k)}^{m=1}$ at the coarse grid cells that are adjacent to the fine grid. This scheme is not fully self-consistent in the sense that the value of $\Phi_{i+1,k}^{m=1}$ depends on the boundary values on the fine grid $\Phi_{B,(i-1,k)}^{m=2}$ and $\Phi_{B,(i-1,k+1)}^{m=2}$, which in turn are found using the potentials on the coarse grid (see Eqs. 3–6). This inconsistency can be resolved through iterations when the value of $\Phi_{i+1,k}^{m=1}$ is substituted back to Eqs. (3)–(6) to find the boundary potentials $\Phi_{B,(i-1,k)}^{m=2}$ and $\Phi_{B,(i-1,k+1)}^{m=2}$ and the calculations involving Eqs. (A.3)–(A.6) are repeated. Our experience shows that convergence is achieved after several iterations.

The difference between methods B and C is illustrated in Figure A.2, which shows the relative error in the gravitational acceleration at the $x-z$ plane obtained for both methods. We consider two nested grids and an ellipsoid with the ratio of semi-axes 0.75:1.5:1.5. We note that for both cases, method A is used to calculate the acceleration in the cells of the fine grid that are adjacent to the coarse grid. We found that the span of errors is

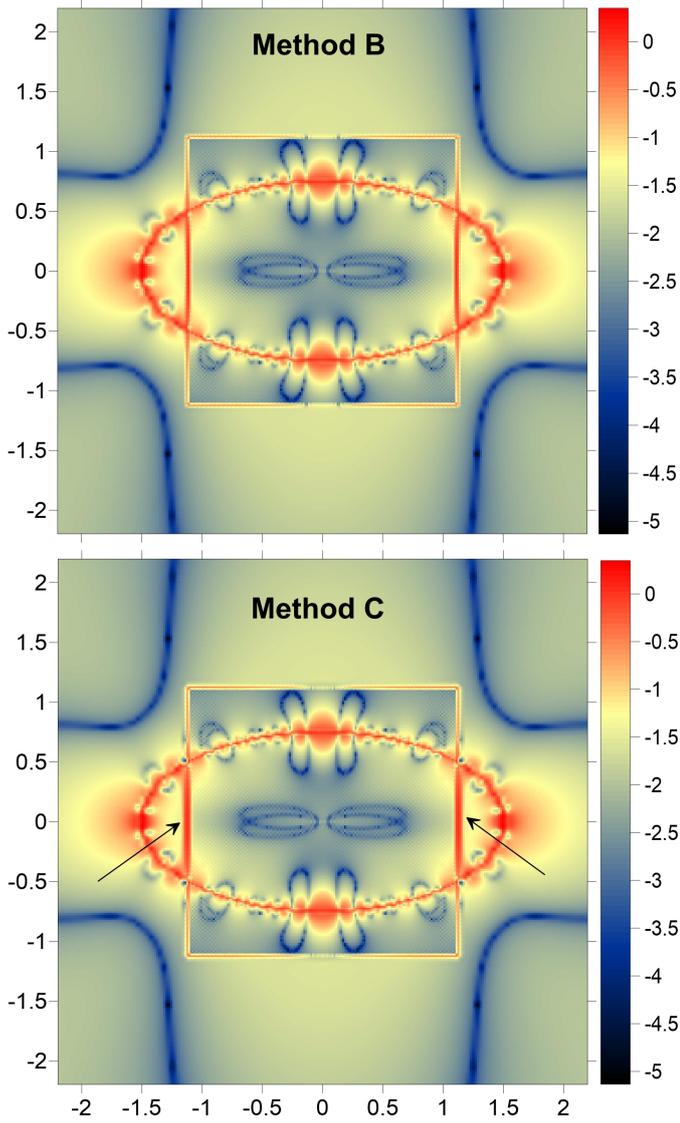


Fig. A.2. Relative error in the gravitational acceleration of an oblate ellipsoid. The top and bottom panels correspond to methods B and C for computing the gravitational acceleration at the coarse cells that are adjacent to the coarse-fine grid interface. The arrows illustrate the regions where differences between methods B and C are found. The color bars show errors relative to the analytic solution in per cent.

similar in both methods. In particular, the maximum errors occur in the vicinity of the ellipsoid surface and also at the coarse-fine grid interface. When method B is considered, the error is maximal at the adjacent fine cells and is much smaller at the adjacent coarse cells. When method C is considered, the error is similar and maximal on both sides of the coarse-fine grid interface. This effect is noticeable if the coarse-fine grid interface lies inside the ellipsoid, as illustrated by the arrows in Figure A.2. The corresponding region with the highest error appears thicker in method C than in method B. To summarize, method C is physically more accurate as it fulfills the Gauss theorem at the grid interfaces, but it comes at the expense of an increased error in the gravitational acceleration at the coarse cells that are adjacent to the fine grid.

Appendix B: Modified convolution method and reduction of overheads

In this section, we describe how the convolution method can be extended to tackle the dipole problem across multiple nested grids and highlight the associated numerical costs. First, we provide a scheme, in which the dipole moments are accurately computed for each pair of neighboring grids on a nested mesh with $m \geq 3$. As was demonstrated in Sect. 7, this allows notably reducing the error in the gravitational potential of a close binary near the coarse-fine grid interface. The corresponding algorithm can be written as

$$\Phi(\Omega^n) = \Phi_{\text{hole}}^{m=n} + \tilde{\Phi}_{\text{dbl}}^{m=n+1} + \sum_{k=1}^{n-1} \tilde{\Phi}_{\text{hole}}^{m=k}, \quad \text{for } n < M_{\text{depth}} \quad (\text{B.1})$$

$$\Phi(\Omega^n) = \sum_{k=1}^{n-1} \tilde{\Phi}_{\text{hole}}^{m=k} + \Phi^{m=n}, \quad \text{for } n = M_{\text{depth}}. \quad (\text{B.2})$$

Here, M_{depth} is the maximum number of nested grids. Each pair of neighboring grids $\Omega^n:\Omega^{n+1}$ has now its own auxiliary doubled grid $\tilde{\Phi}_{\text{dbl}}^{m=n+1}$. As usual, the tilde sign means proper interpolation from a coarser to a finer grid and vice versa. When this modified scheme is used on CPUs, the runtime increases by about a factor of four compared to the original method. When CPUs and GPUs are used together, the runtime increases by a factor of 2.

When the dipole moment are to be computed across three neighboring grids (as in the bottom panel of Fig. 15), the corresponding algorithm reads as

$$\Phi(\Omega^n) = \Phi_{\text{hole}}^{m=n} + \tilde{\Phi}_{\text{dbl,hole}}^{m=n+1} + \tilde{\Phi}_{\text{qdp}}^{m=n+2} + \sum_{k=1}^{n-1} \tilde{\Phi}_{\text{hole}}^{m=k}, \quad (\text{B.3})$$

$$\text{for } n < M_{\text{depth}} - 1 \quad (\text{B.4})$$

$$\Phi(\Omega^n) = \Phi_{\text{hole}}^{m=n} + \tilde{\Phi}_{\text{dbl}}^{m=n+1} + \sum_{k=1}^{n-1} \tilde{\Phi}_{\text{hole}}^{m=k}, \quad \text{for } n < M_{\text{depth}} \quad (\text{B.5})$$

$$\Phi(\Omega^n) = \sum_{k=1}^{n-1} \tilde{\Phi}_{\text{hole}}^{m=k} + \Phi^{m=n}, \quad \text{for } n = M_{\text{depth}}. \quad (\text{B.6})$$

Figure B.1 presents the corresponding mesh decomposition for the case of three nested grids. The solution procedure is now split into six steps. Three steps shown in the top row of the figure are similar to those of the original scheme (see Fig 5). We calculate the potential $\Phi^{m=3}$ on the finest Ω^3 grid, the potential $\Phi_{\text{hole}}^{m=1}$ on the coarsest Ω_{hole}^1 grid with a central hole, and the potential $\Phi_{\text{hole}}^{m=2}$ on the intermediate Ω_{hole}^2 grid with a central hole. The other three steps shown in the bottom row of Figure B.1 involve the calculation of the potentials on the auxiliary grids, which are introduced for the correct calculation of the dipole moments. We note that the doubled $\Omega_{\text{dbl,hole}}^2$ grid has now a central hole, which is filled with zero densities to avoid counting twice the mass distribution on the finest Ω^3 grid. The doubled Ω_{dbl}^3 grid is analogous to Ω_{dbl}^2 grid for the case of two nested meshes. Finally, the Ω_{qdp}^3 grid is introduced, which is an extension of the finest Ω^3 grid to the physical size of the coarsest Ω^1 grid. The additional cells are filled with zero densities. This grid allows us to correctly calculate the dipole moments at the coarsest Ω^1 grid. When the dipole moments are calculated across three neighboring grids, the runtime increases by a factor of 4.5 for $N = 64$ and by a factor of 8.5 for $N = 128$. An extension of the scheme to a larger number of neighboring grids can be done by analogy to Eqs. (B.1)–(B.6).

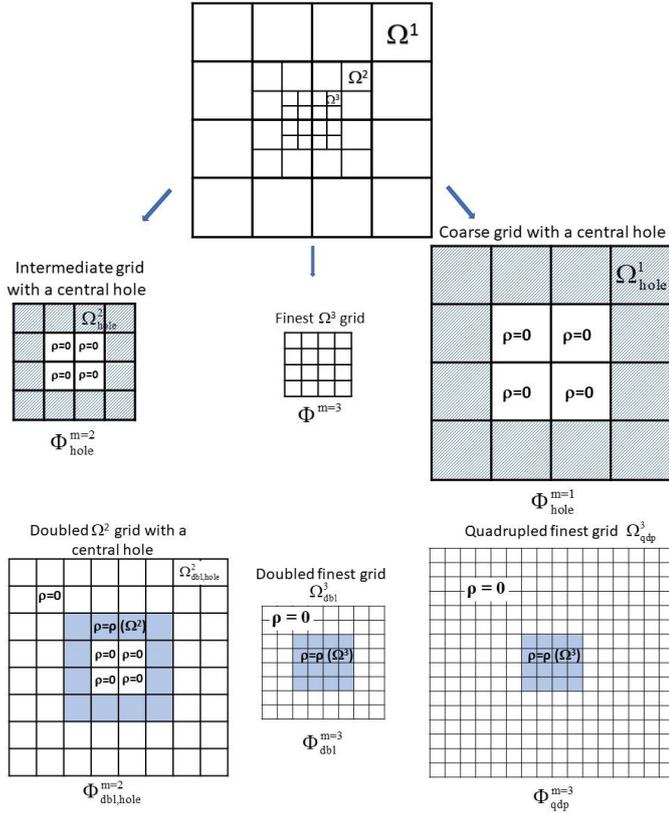


Fig. B.1. Decomposition of the $m = 3$ mesh in the modified convolution method accounting for the dipole moments. Six individual grids are now considered with the three grids in the bottom row representing the auxiliary steps needed to compute the dipole moments for three neighboring grids.

We note that the calculation of $\Phi_{qdp}^{m=3}$ comes at a substantial computational expense in both the computational time and the memory requirements. A solution to this problem may be in switching to direct summation when calculating the input of the mass distribution on the finest Ω^3 grid to the total gravitational potential on the coarsest Ω^1 grid. In this case, the calculation of the potential Φ_{qdp}^3 on the quadrupled grid Ω_{qdp}^3 becomes obsolete. Instead, the input of the finest Ω^3 grid to the potential on the coarsest Ω^1 grid is calculated as

$$\Phi_{d.s.}^{3 \rightarrow 1}(x_i, y_j, z_k) = -G \sum_{i', j', k'} M^{m=3}(x_{i'}, y_{j'}, z_{k'}) \times \mathcal{G}^{3 \rightarrow 1}(x_i - x_{i'}, y_j - y_{j'}, z_k - z_{k'}), \quad (\text{B.7})$$

where $M^{m=3}$ is the mass distribution on the finest Ω^3 grid and $\mathcal{G}^{3 \rightarrow 1}$ is the inverse distance between the cells on the Ω^1 grid with coordinates (x_i, y_j, z_k) and the cells on the Ω^3 grid with coordinates $(x_{i'}, y_{j'}, z_{k'})$. Our numerical experiments reveal that for a numerical resolution of $N \leq 32^3$ the direct summation is faster than the calculation of Φ_{qdp}^3 using the convolution theorem.

Appendix C: Further comparison of the modified and original convolution methods

Here, we perform an additional comparison between the original and modified convolution methods, the latter considering the dipole moments at every pair of neighboring grids (see

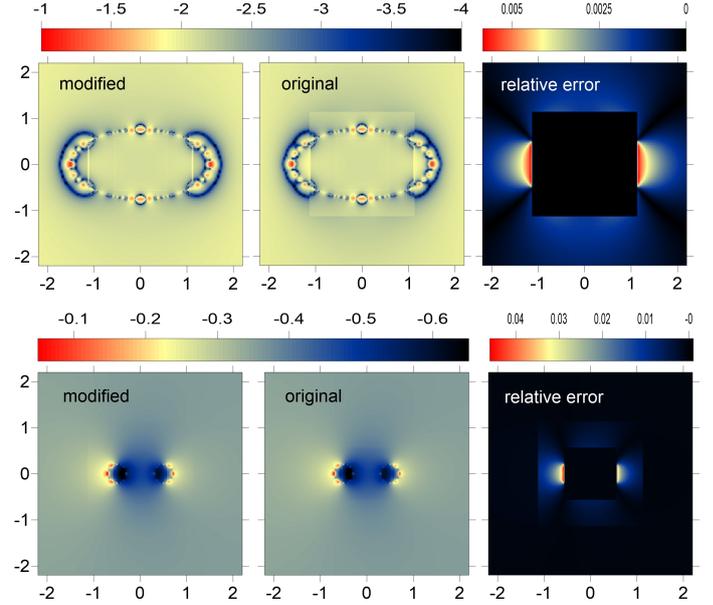


Fig. C.1. Comparison of the modified and original convolution method for an oblate ellipsoid (top row) and wide separation binary (bottom row). The errors in the gravitational potential compared to the analytical solution are shown in the left and middle columns (in log units), while the right column displays the relative error in the gravitational potential between the two methods in percent.

Eqs. B.1–B.2). Figure C.1 presents the errors in the gravitational potential relative to analytic solutions for each method and also the relative errors in the gravitational potential between the two methods. Four nested grids with a numerical resolution of $N = 128^3$ for each grid were used in this test problem. The mass distributions similar to those considered in Sect. 6 were used: an oblate ellipsoid and a wide separation binary. In particular, the ratio of semi-major axes of the ellipsoid was increased to 0.75:1.5:1.5 compared to the previous test problem (0.5:1.0:1.0) so the ellipsoid now covers all grid interfaces.

In the modified method, a smoother transition between the grid interfaces can be noted, but otherwise the potentials are similar. The relative error between the two methods for a smooth density distribution (ellipsoid, top row) is very small, $< 0.006\%$, indicating that the modification to the convolution method in this case is not really required. For a wide separation binary (bottom row), the error increased somewhat but still stays quite low, $< 0.045\%$. Therefore, the modified method is recommended for models where the formation of tightly spaced gravitating objects is expected, for example, close binary stars or gravitationally fragmenting protostellar disks.