






A Self-gravity Module for the PLUTO Code

Ankush Mandal¹ , Dipanjan Mukherjee¹ , and Andrea Mignone² ¹ Inter-University Centre for Astronomy and Astrophysics, Pune 411007, India; ankushm@iucaa.in² Dipartimento di Fisica, Università di Torino via Pietro Giuria 1 (I-10125) Torino, Italy

Received 2023 May 29; revised 2023 August 1; accepted 2023 August 1; published 2023 September 14

Abstract

We present a novel implementation of an iterative solver for the solution of Poisson’s equation in the PLUTO code for astrophysical fluid dynamics. Our solver relies on a relaxation method in which convergence is sought as the steady-state solution of a parabolic equation, whose time discretization is governed by the Runge–Kutta–Legendre (RKL) method. Our findings indicate that the RKL-based Poisson solver, which is both fully parallel and rapidly convergent, has the potential to serve as a practical alternative to conventional iterative solvers such as the Gauss–Seidel and successive overrelaxation methods. Additionally, it can mitigate some of the drawbacks of these traditional techniques. We incorporate our algorithm into a multigrid solver to provide a simple and efficient gravity solver that can be used to obtain the gravitational potentials in self-gravitational hydrodynamics. We test our implementation against a broad range of standard self-gravitating astrophysical problems designed to examine different aspects of the code. We demonstrate that the results match excellently with analytical predictions (when available), and the findings of similar previous studies.

Unified Astronomy Thesaurus concepts: [Computational methods \(1965\)](#); [Gravitation \(661\)](#); [Hydrodynamical simulations \(767\)](#)

1. Introduction

Elliptical partial differential equations (PDEs) are crucial in numerous mathematical and physical problems. One prime example is Poisson’s equation, which emerges naturally in many contexts such as gravity, incompressible flows, electromagnetism, thermal conduction, the divergence-cleaning procedure in magnetohydrodynamics, and several others. In this sense, the development of efficient and fast solvers for such PDEs represents one of the primary goals in computational physics. In particular, solving Poisson’s equation for the gravitational potential for a given mass distribution is of great interest in those astrophysical environments where self-gravitational forces play a crucial role, such as the formation of stars (Ostriker et al. 2001; McKee & Ostriker 2007), planet formation (Boss 1997; Rice et al. 2005), supernova explosions (Nordhaus et al. 2010; Couch et al. 2013), the evolution of galaxies (Mo et al. 2010), and the structure formation (Bertschinger 1998) in the Universe. Thus, the accurate representation of self-gravity is critical for achieving realistic and reliable results in the numerical modeling of such systems.

The need for solving Poisson’s equation has led to the development of various algorithms with different levels of complexity and applicability. One can rely on direct solvers such as the fast Fourier transform (FFT) method to achieve a solution accurate up to the round-off error. On the other hand, the traditional relaxation methods such as Gauss–Seidel (G-S) or successive overrelaxation (SOR) are also of great interest due to their simplicity of implementation (see Press et al. 1992 for details). Over the past two decades, there has been an increasing interest in these iterative methods due to the advancement of massively parallel computers. Since one generally does not need the solution to be accurate up to the round-off error (as provided by the FFT method), iterative methods are preferred for practical purposes as iterations can be stopped early when the iteration error

of the solution drops below the truncation error. Although these iterative solvers suffer from a slow convergence rate for long-wavelength modes (Press et al. 1992), the usage of a multigrid scheme can improve the convergence rate drastically (Brandt 1977; Briggs et al. 2000). The multigrid method uses a hierarchy of grids with decreasing spatial resolution to correct the error in the guess solution on the finest grid. The usage of coarser grids ensures that the long-wavelength modes of the error on the finer grids appear short on the coarser ones, and thus can be damped down efficiently. Thus, the multigrid method consists of three primary elements: (i) a smoothing operator on the finer grids that damps small-wavelength (high-frequency) modes, smoothing out the solution; (ii) some restriction and prolongation operators to map different quantities between grids; and (iii) an efficient Poisson solver on the coarsest grid, which is necessary for solving the error equation (Trottenberg et al. 2000).

While the G-S and SOR methods are commonly used in several codes for the coarsest grid solver, both pose certain limitations. In particular, the G-S algorithm converges very slowly, whereas the convergence rate of the SOR method heavily depends on the optimal choice of the overrelaxation parameter (ω), which is typically problem dependent and difficult to derive analytically (Press et al. 1992). Furthermore, none of the methods achieves full parallelization, meaning that all grid points within the computational domain cannot be updated simultaneously during a single iteration step. Rather, the updates are accomplished in an alternating odd–even pattern, requiring twice the amount of parallel communication between processors compared to a fully parallel algorithm, which incurs additional computational overheads (Trottenberg et al. 2000).

In this study, we present a new Poisson solver for the PLUTO code³ (Mignone et al. 2007, 2012) for self-gravitational astrophysical fluids. The method is implemented within a multigrid V-cycle algorithm structure and it takes advantage of the accelerated super-time-stepping (STS) technique based on the Runge–Kutta–Legendre (RKL) method used for solving



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

³ PLUTO is freely available at <http://plutocode.ph.unito.it>.

parabolic PDEs (Meyer et al. 2012, 2014; Vaidya et al. 2017), by viewing Poisson’s equation as the steady-state solution of a parabolic equation. RKL time-stepping methods are optimized for stability rather than accuracy, so that a single time step, consisting s number of substages, can be taken as much larger compared to the time step permitted by a standard explicit method, and it increases monotonically with s (Meyer et al. 2014). In theory, a sufficiently high s can produce a steady-state solution in a single iteration, but the convergence rate does not necessarily increase monotonically with increasing s . In fact, we demonstrate that there exists an optimal value of s that produces the maximum convergence rate, which depends only on the domain size and boundary conditions and it can be analytically derived. An optimized RKL-based Poisson solver exhibits a much higher convergence rate than the G-S method and can become competitive with the SOR scheme. Moreover, it eliminates the uncertainties associated with the optimal overrelaxation parameter (ω) of the SOR algorithm. In addition, the RKL method is fully parallelizable, allowing all grid points to be updated simultaneously during a single iteration, which results in a lower parallel communication overhead, making the scheme well suited for massively parallel systems.

The article is organized as follows. In Section 2, we describe the basic V-cycle multigrid algorithm, its individual components, and the boundary conditions. Section 3 outlines how gravity is coupled with hydrodynamics. In Section 4, we present the accuracy of the Poisson solver and the gravity module as a whole for a wide variety of standard self-gravitating astrophysical problems. The chosen problems are designed to showcase the accuracy of various aspects of the code, including energy conservation, variation in the equation of state (EOS), momentum conservation, and boundary conditions. Finally, we summarize our findings in Section 5.

2. The Algorithm

2.1. The Iterative Multigrid Algorithm

The core idea of the multigrid technique is to use a hierarchy of grids with geometrically decreasing mesh sizes. The method solves a system of discrete equations on a given grid iteratively, through constant interactions with a series of coarser grids. In this method, the coarser grids are used to find the error in the approximate solution which can be used to obtain a better guess solution for the next iteration, traditionally called the correction scheme multigrid method (Briggs et al. 2000; Trottenberg et al. 2000). In the following, we briefly summarize the scheme.

Suppose we wish to solve the following elliptic equation,

$$\mathcal{L}\Phi = f, \quad (1)$$

where \mathcal{L} is a linear elliptic operator and f is the source function defined in some domain Ω . Let us assume, our domain is to be discretized into N_x , N_y , and N_z grid points with a grid spacing of h . This discretized domain is denoted as Ω^h , where the grid points are represented by $x_i = ih$, $y_j = jh$, and $z_k = kh$. Similarly, if we have a discretization of Ω with $N_x/2$, $N_y/2$, and $N_z/2$ points and a grid spacing of $2h$, we represent it as Ω^{2h} . Then, the discretization of Equation (1) on Ω^h can be written as

$$\mathcal{L}_h\Phi_h = f_h. \quad (2)$$

Let us assume that the guess to the solution of Equation (2) at the first iteration is Ψ_h . Then the error can be found as

$$e_h = \Phi_h - \Psi_h, \quad (3)$$

where Φ_h is the exact solution of Equation (2). Now, the residual can be defined as

$$r_h = \mathcal{L}_h\Psi_h - f_h. \quad (4)$$

Interestingly, since \mathcal{L}_h is a linear operator, the error e_h satisfies

$$\mathcal{L}_he_h = -r_h. \quad (5)$$

Thus, instead of relaxing the original solution Φ_h with an arbitrary guess solution, we can relax Equation (5) with a specific guess $e_h = 0$.

Now, if we expand e_h into a discrete Fourier series, it can be shown that the long-wavelength modes of e_h converge very slowly, whereas the oscillatory or short-wavelength modes of e_h can be reduced more easily (Press et al. 1992; Briggs et al. 2000; Trottenberg et al. 2000). Thus, if we restrict the error to a coarser grid (Ω^{2h}) from the original grid (Ω^h), the long-wavelength modes of the error will appear as short-wavelength modes on the coarser grid, which can be reduced by a large factor with each iteration (Briggs et al. 2000). Once, we get the solution for e_{2h} on Ω^{2h} , it can be interpolated back to the fine grid to calculate the approximated solution \tilde{e}_h of the error. Then the guess to the solution is updated by

$$\Psi_h^{\text{new}} = \Psi_h + \tilde{e}_h. \quad (6)$$

Thus, we can utilize these concepts recursively across a hierarchy of grids with geometrically decreasing mesh sizes to carry out an entire multigrid cycle. In order to transfer data between grids with different mesh spacings, we need to define restriction and prolongation operators. Fine-to-coarse grid interpolation is done through the restriction operator (\mathcal{R}) while the inverse process is achieved using the coarse-to-fine prolongation operator (\mathcal{P}). We will discuss these in detail in Section 2.2.

One of the simplest forms of multigrid cycles is the V-cycle, which is schematically shown in Figure 1. The V-cycle algorithm initially approximates the solution by performing N_{pre} smoothing iterations on the finest grid level ($l=0$). The residual is then restricted to the next coarse level, where another N_{pre} smoothing iterations are carried out. This process of restriction followed by smoothing is repeated until the coarsest level is reached (downstroke), where we solve the correction equation with a fast iterative solver. Subsequently, the correction solution is prolonged to the next fine level, which is utilized to adjust the solution. Using the modified solution, N_{post} number of smoothing operations are performed. The sequence of prolongation and smoothing steps persists until the correction is incorporated into the solution at the finest grid level (upstroke). The whole procedure forms a V-cycle, which is repeated until the desired convergence level is reached. In the following, we give the V-cycle multigrid algorithm that contains L number of levels (see Briggs et al. 2000; Trottenberg et al. 2000, for details).

do {

1. At $l=0$, perform N_{pre} relaxation on $\mathcal{L}^0\Phi^0 = f^0$.
2. for ($l=1$ to $l < L$) {
 - 2a. Compute the residual: $r^{l-1} = \mathcal{L}^{l-1}\Phi^{l-1} - f^{l-1}$.
 - 2b. Calculate f^l by performing restriction on r^{l-1} : $f^l = \mathcal{R}(r^{l-1})$.
 - 2c. Set the guess solution for this level to zero: $\delta\Phi^l = 0$.
 - 2d. If $l < L - 1$
- i. Perform N_{pre} relaxation on $\mathcal{L}^l\delta\Phi^l = -f^l$, with $\delta\Phi^l = 0$ on $\partial\Omega^l$.
 - ii. $l \leftarrow l + 1$

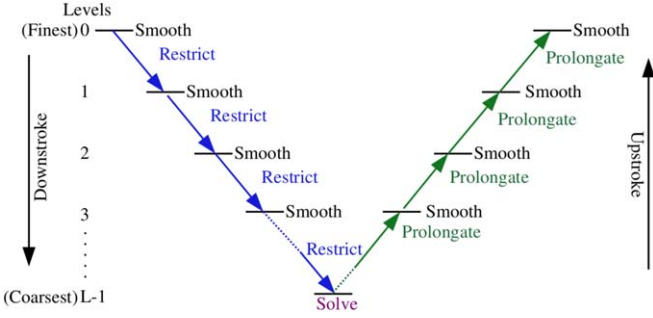


Figure 1. A schematic diagram of the multigrid V-cycle algorithm.

- 2e. Else if $l=L-1$, solve $\mathcal{L}^l \delta\Phi^l = -f^l$ with a fast iterative solver.
 - }
 3. for ($l=L-2$ to $l>0$) {
 - 3a. Perform prolongation of $\delta\Phi^{l+1}$ to correct $\delta\Phi^l$: $\delta\Phi^l \leftarrow \delta\Phi^l + \mathcal{P}(\delta\Phi^{l+1})$.
 - 3b. Perform N_{post} relaxations on $\mathcal{L}^l \delta\Phi^l = -f^l$, with $\delta\Phi^l = 0$ on $\partial\Omega^l$.
 - 3c. $l \leftarrow l-1$.
 - }
 4. At $l=0$,
 - 4a. Correct the finest grid solution: $\Phi^0 \leftarrow \Phi^0 + \mathcal{P}(\delta\Phi^1)$
 - 4b. Perform N_{post} relaxation on $\mathcal{L}^0 \Phi^0 = f^0$.
 - }
- while (residual is above the specified tolerance level)

From the above steps, we see that, at the coarsest level of the multigrid hierarchy, we need a very efficient iterative solver to estimate the solution of the error equation, which is sometimes called the “kernel” of the iterative multigrid algorithm. We discuss this in detail in Section 2.2.4.

2.2. Operator Definitions

Spatial discretization can be based on either staggered or cell-centered meshes, but the structure of different operators and the treatment of boundary conditions change accordingly (Trottenberg et al. 2000). For application purposes, we adopt in what follows a multigrid algorithm that makes use of a cell-centered discretization, essential in devising the iterative multigrid algorithm, as discussed in Section 2.1.

2.2.1. Discretization of the Laplace Operator

In Section 2.1, we discussed how the iterative multigrid algorithm works for any general linear elliptic PDE. However, in this paper, our primary goal is to implement a multigrid solver for Poisson’s equation for gravitational potential, which has the following form,

$$\nabla^2 \Phi = 4\pi G\rho, \quad (7)$$

where ∇^2 is the Laplace operator. We use a second-order accurate five-point (in 2D) or seven-point (in 3D) stencil for the discretization of the Laplace operator on a particular grid level L . For two dimensions, the discretization is given by

$$\begin{aligned} \nabla^2 \Phi_{ij}^L \approx & \frac{1}{\Delta x_L^2} (\Phi_{i-1,j}^L - 2\Phi_{ij}^L + \Phi_{i+1,j}^L) \\ & + \frac{1}{\Delta y_L^2} (\Phi_{i,j-1}^L - 2\Phi_{ij}^L + \Phi_{i,j+1}^L), \end{aligned} \quad (8)$$

and for three dimensions,

$$\begin{aligned} \nabla^2 \Phi_{i,j,k}^L \approx & \frac{1}{\Delta x_L^2} (\Phi_{i-1,j,k}^L - 2\Phi_{i,j,k}^L + \Phi_{i+1,j,k}^L) \\ & + \frac{1}{\Delta y_L^2} (\Phi_{i,j-1,k}^L - 2\Phi_{i,j,k}^L + \Phi_{i,j+1,k}^L) \\ & + \frac{1}{\Delta z_L^2} (\Phi_{i,j,k-1}^L - 2\Phi_{i,j,k}^L + \Phi_{i,j,k+1}^L), \end{aligned} \quad (9)$$

where Δx_L , Δy_L , and Δz_L are the grid spacings along x -, y -, and z -directions on the L th level.

2.2.2. Restriction Operator

As discussed in Section 2.1, a restriction operator (\mathcal{R}_{2h}^h) is needed that maps or restricts the error from the fine grid (Ω^h) to the coarse grid (Ω^{2h}). Here, we adopt the simple second-order accurate four-point average for two dimensions (Trottenberg et al. 2000). Application of \mathcal{R}_{2h}^h to a fine grid (Ω^h) function $r_h(x, y)$ at coarse grid point $(x, y) \in \Omega^{2h}$ yields

$$\begin{aligned} r_{2h}(x, y) &= \mathcal{R}_{2h}^h r_h(x, y) \\ &= \frac{1}{4} \left[r_h\left(x - \frac{h}{2}, y - \frac{h}{2}\right) + r_h\left(x + \frac{h}{2}, y - \frac{h}{2}\right) \right. \\ &\quad \left. + r_h\left(x - \frac{h}{2}, y + \frac{h}{2}\right) + r_h\left(x + \frac{h}{2}, y + \frac{h}{2}\right) \right]. \end{aligned} \quad (10)$$

In Figure 2(a), we show a simple stencil of the fine (black lines) and coarse grids (red lines). The values are defined at the center of the cells. From Equation (10), the coarse grid value E is simply given by

$$E = \frac{1}{4}(h + g + e + f), \quad (11)$$

where g , h , e , and f are the fine grid values at the specified location in Figure 2(a).

In three dimensions, the restriction operator can be easily extended from the 2D counterpart by considering an eight-point average, which reads

$$\begin{aligned} r_{2h}(x, y, z) &= \mathcal{R}_{2h}^h r_h(x, y, z) \\ &= \frac{1}{8} \left[r_h\left(x - \frac{h}{2}, y - \frac{h}{2}, z - \frac{h}{2}\right) \right. \\ &\quad + r_h\left(x + \frac{h}{2}, y - \frac{h}{2}, z - \frac{h}{2}\right) \\ &\quad + r_h\left(x - \frac{h}{2}, y + \frac{h}{2}, z - \frac{h}{2}\right) \\ &\quad + r_h\left(x - \frac{h}{2}, y - \frac{h}{2}, z + \frac{h}{2}\right) \\ &\quad + r_h\left(x + \frac{h}{2}, y - \frac{h}{2}, z + \frac{h}{2}\right) \\ &\quad + r_h\left(x - \frac{h}{2}, y + \frac{h}{2}, z + \frac{h}{2}\right) \\ &\quad + r_h\left(x + \frac{h}{2}, y + \frac{h}{2}, z - \frac{h}{2}\right) \\ &\quad \left. + r_h\left(x + \frac{h}{2}, y + \frac{h}{2}, z + \frac{h}{2}\right) \right]. \end{aligned} \quad (12)$$

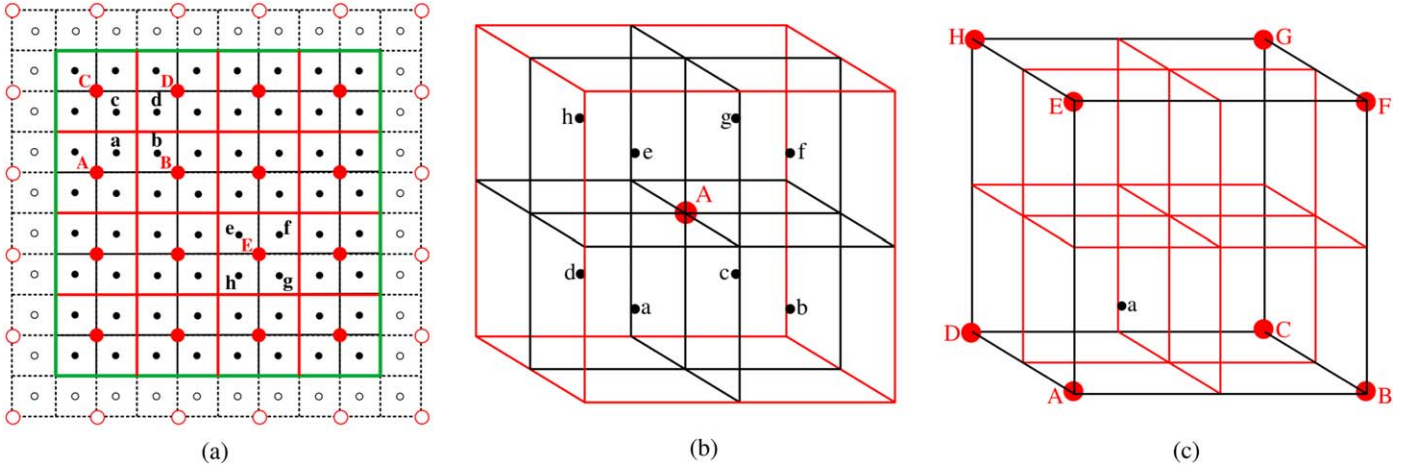


Figure 2. Visual representations of the restriction (\mathcal{R}) and prolongation (\mathcal{P}) operators in 2D (a) and 3D (b) and (c). The uppercase letters in red represent the value cell-centered values of the coarse grid and the lowercase black letters are the same for the fine grid. If we consider panel (a), in order to calculate the coarse grid value E , we simply take the average of the fine grid values e, f, g , and h that are closest to E (Equation (11)). Conversely, the fine grid value a can be estimated by taking the weighted sum of the four nearest coarse grid values A, B, C , and D (Equation (14)). Likewise, the 3D restriction operator (depicted in panel (b)) performs an averaging operation on the eight closest fine grid values (as expressed in Equation (13)). On the other hand, the 3D prolongation operator (illustrated in panel (c)) calculates the weighted sum of the eight nearest coarse grid points (as indicated in Equation (15)).

From Figure 2(b), the coarse grid value A can be obtained by

$$A = \frac{1}{8}(a + b + d + c + e + f + h + g). \quad (13)$$

2.2.3. Prolongation Operator

In order to transfer the coarse grid correction to the fine grid, we use the second-order accurate cell-centered prolongation operator shown in Figure 2. In two dimensions, with reference to Figure 2(a), the values at fine grid points are calculated as

$$\begin{aligned} a &= \frac{1}{16}(9A + 3B + 3C + D), \\ b &= \frac{1}{16}(3A + 9B + C + 3D), \\ c &= \frac{1}{16}(3A + B + 9C + 3D), \\ d &= \frac{1}{16}(A + 3B + 3C + 9D), \end{aligned} \quad (14)$$

where a, b, c , and d are the fine grid values at the cell centers and A, B, C , and D are the coarse grid cell-centered values as depicted in Figure 2(a). In three dimensions, we can easily extend the prolongation operator, by inspecting Figure 2(c), where the cell-centered fine grid value a can be obtained as follows,

$$a = \frac{1}{64}(27A + 9B + 9D + 9E + 3C + 3F + 3H + G), \quad (15)$$

where A, B, C, D, E, F, G , and H are the cell-centered coarse grid values.

2.2.4. Smoothing Operator and Kernel Solver

The two most important components of the iterative multigrid algorithm are the smoothing operator which is used to smooth out the solution error on the finer grids, and the coarsest grid solver (or the kernel solver), which solves the

error equation on the coarsest level for the given boundary conditions. For the smoothing operations, the G-S relaxation method with red-black ordering (GS-RB) has been shown to be one of the best methods available in the literature (Press et al. 1992; Trottenberg et al. 2000) and it will also be our method of choice. The GS-RB solver has been used in several (magneto-) hydrodynamic codes where a multigrid algorithm is used for solving Poisson's equation (Ziegler 2005; Ricker 2008; Almgren et al. 2010; Guillet & Teyssier 2011; Wang & Yen 2020). In the GS-RB scheme, cell values are updated in two passes, with each pass covering half of the computational domain according to the colors of a checkerboard pattern (red or black).

For the coarsest grid direct solver, there are various algorithms ranging from iterative solvers to the conjugate gradient method and FFT. Traditionally, iterative methods are employed as kernel solvers owing to their simplicity of implementation. Still, one important criterion for choosing a particular solver is the degree of parallelization as it can degrade the overall performance of a highly parallelized code. The traditional relaxation ω -Jacobi scheme (Trottenberg et al. 2000) is fully parallel (i.e., the operator can be applied to all grid points simultaneously and new values are mutually exclusive) but has a very poor convergence rate.

The GS-RB method has a relatively better convergence rate than the ω -Jacobi method, but it becomes inefficient as the domain becomes larger and it is also half-parallel.⁴ The ω -GS-RB or the SOR method can be designed to improve the convergence rate by choosing an optimal value of ω . However, the efficiency of this method heavily depends on the over-relaxation parameter (ω), which is typically problem dependent. A slight deviation from the true value of ω can degrade the convergence rate considerably.

In this work, we propose a novel approach for the solution of Poisson's equation based on the RKL method for solving parabolic differential equations (Meyer et al. 2012, 2014;

⁴ In the GS-RB method, one iteration consists of two half-steps. Thus, only half of the total grid points inside the computational domain can be updated simultaneously. This implies the method is half-parallel (see Trottenberg et al. 2000 for details).

Vaidya et al. 2017). The RKL method is a class of STS methods that have been developed to deal with the restrictive time step of an explicit scheme for parabolic equations of the form $\partial u/\partial t = \mathbf{M}u$. This allows us to consider a larger time step (τ) than the maximum permissible explicit parabolic time step (Δt_{par}) without affecting the stability of the solution. For the s -stage RKL method, the maximum possible time step (τ_{max}) can be shown to be,

$$\tau_{\text{max}} = \Delta t_{\text{par}} \frac{s^2 + s}{2}, \quad (16)$$

where $\Delta t_{\text{par}} = 2/|\lambda_{\text{max}}|$ is the explicit parabolic time step (λ_{max} is the maximum eigenvalue of the elliptic operator \mathbf{M}). As Poisson's equation can be sought as the steady-state manifestation of the following inhomogeneous parabolic equation,

$$\frac{\partial \Phi}{\partial t} = \nabla^2 \Phi - \rho, \quad (17)$$

we can easily take advantage of the RKL technique to solve for the gravitational potential Φ in Equation (17) in the $t \rightarrow \infty$ limit with much larger time steps than allowed by explicit methods. While this could (in principle) be obtained within a single time step τ using a very large value of s , it does not ensure a monotonically increasing convergence rate with growing value of s .

In fact, although increasing the value of s improves the convergence rate of the RKL operator, we find that there exists an optimal value of s (s_{opt}) for which the convergence rate of an RKL-based Poisson solver is the highest, and any further increment of s degrades the convergence rate. This value of s_{opt} depends only on the domain size and the boundary conditions, which remove the uncertainty of the optimal overrelaxation parameter ω in the SOR algorithm. In Appendix A, we derive the convergence property of the RKL-based Poisson solver in detail. Another advantage of the RKL scheme is that it is fully parallel, i.e., the calculation of the cell values at the current time step entirely depends on the previous step, thus a single communication between processors is enough to update all the points during a single iteration. Therefore, being fully parallel, highly optimized, and specifically not fine tuned, the RKL-based Poisson solver can be an alternative to traditional relaxation methods for solving Poisson's equation. For instance, in Appendix B, we demonstrate the comparative performance of the G-S, SOR, and RKL methods for a 3D problem. We find that the RKL scheme not only achieves a given residual level in the least amount of time but also exhibits a significantly lower parallel communication overhead compared to the G-S and SOR methods. Hence, in the multigrid algorithm described in this paper, we use the RKL-based Poisson solver for the coarsest grid kernel solver.

2.3. Boundary Conditions

For the cell-centered multigrid method, the boundary points need special attention as for the cell-centered discretization, with no grid points laying along the boundary (see the green line in Figure 2(a), which lies along the cell edges). For a Dirichlet-type boundary condition, linear extrapolation is used to specify solution values at the centers of ghost cells. For example, for the two boundaries at the start and end points of x -

axis, we have

$$\Phi_{0,j,k} = 2\Phi_{\frac{1}{2},j,k} - \Phi_{1,j,k}, \quad (18)$$

$$\Phi_{N_x+1,j,k} = 2\Phi_{N_x+\frac{1}{2},j,k} - \Phi_{N_x,j,k}, \quad (19)$$

where $\Phi_{\frac{1}{2},j,k}$ and $\Phi_{N_x+\frac{1}{2},j,k}$ are the potential values at the edges of the first and last active cells, respectively, corresponding to the physical boundaries, which are specified by the user. The ghost cells along the y - and z -axes can be filled in a similar way. Notice that the boundary conditions must be specified at the beginning of the relaxation procedure only on the finest level ($l=0$). As we are solving the error equations on levels other than the finest one, the error along the boundary should be zero for a Dirichlet-type boundary condition. Thus the ghost cell values at the levels with $l > 0$ will be

$$e_{0,j,k} = -e_{1,j,k}, \quad \text{and} \quad e_{N_x+1,j,k} = -e_{N_x,j,k}. \quad (20)$$

In the case of periodic boundary conditions (let us say along the x -axis), left ghost cells are filled by copying the last active cell values along the x -axis and vice-versa for the right ghost cells,

$$\Phi_{0,j,k} = \Phi_{N_x,j,k}. \quad (21)$$

The same periodic boundary conditions (Equation (21)) are also employed at the coarser levels ($l > 0$).

The periodic boundary condition for gravitational potential, however, requires some modifications of Poisson's equation (Equation (7)). Let Φ be the solution of Equation (7) in the domain Ω with periodic boundary conditions. Direct application of Gauss' theorem upon integrating yields

$$\int_{\Omega} \nabla^2 \Phi dV = \int_{\partial\Omega} \frac{\partial \Phi}{\partial \mathbf{n}} dS. \quad (22)$$

Now, for a periodic boundary condition on Φ , the right-hand side of Equation (22) vanishes, which implies

$$\int_{\Omega} \nabla^2 \Phi dV = 0. \quad (23)$$

Thus, if Φ satisfies Equation (7), we must have

$$\int_{\Omega} \rho dV = 0. \quad (24)$$

However, the mass density ρ can never be less than 0 physically.⁵ Thus, it is impossible to satisfy Equation (24) without modifying the source term. The usual way to overcome this issue is to subtract the mean value of the density from the source term, as any infinite homogeneous density distribution does not affect the potential (Binney & Tremaine 1987; Kiessling 1999; Falco et al. 2013). Hence, we can subtract the mean homogeneous density and calculate the potential due to the fluctuation. This leads to the following modification of the source term in Equation (7),

$$\rho \rightarrow \rho - \bar{\rho}, \quad \text{where} \quad \bar{\rho} = \frac{\int_{\Omega} \rho dV}{\int_{\Omega} dV}. \quad (25)$$

⁵ This is analogous to the well-known contradiction in the linear stability analysis of such systems, popularly known as Jeans' Swindle (Binney & Tremaine 1987).

Equation (25) ensures that Equation (24) is satisfied, thus implying the convergence of the multigrid algorithm is guaranteed. We also subtract the average from the residual on all the coarser levels ($l > 0$) explicitly (Ricker 2008). This ensures that the convergence rate of the multigrid algorithm is not affected by the nonzero uniform component of the residuals for the periodic boundary condition.

3. Coupling Gravity with Hydrodynamics

The effect of self-gravity is generally treated by adding source terms in Euler's equations, which are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (26)$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot [\rho \mathbf{v} \mathbf{v} + p \mathbf{I}] = \rho \mathbf{g}, \quad (27)$$

$$\frac{\partial E_t}{\partial t} + \nabla \cdot [(E_t + p) \mathbf{v}] = \rho \mathbf{v} \cdot \mathbf{g}, \quad (28)$$

$$\nabla^2 \Phi = 4\pi G \rho, \quad (29)$$

where ρ is the mass density, \mathbf{v} is the velocity, p is the thermal pressure, $\mathbf{g} = -\nabla \Phi$ is the acceleration due to gravity and E_t is the total energy density given by

$$E = \rho e + \frac{\rho v^2}{2}. \quad (30)$$

The above equations are combined with an EOS $\rho e = \rho e(p, \rho)$ to provide the closure. Thus, the change in momentum and energy due to self-gravity is contributed by the source terms in the momentum (Equation (27)) and energy (Equation (28)) equations, respectively. In order to compute gravitational acceleration \mathbf{g} from the potential (Φ), we use a fourth-order finite difference approximation, e.g.,

$$\begin{aligned} (g_x)_{i,j,k} &= -\nabla_x \Phi \\ &= -\frac{4}{3} \frac{\Phi_{i+1,j,k} - \Phi_{i-1,j,k}}{2\Delta x} \\ &\quad + \frac{1}{3} \frac{\Phi_{i+2,j,k} - \Phi_{i-2,j,k}}{4\Delta x} + \mathcal{O}((\Delta x)^4). \end{aligned} \quad (31)$$

There have been recent developments for not treating gravity as a source term but in a fully conservative fashion where the momentum and energy equations are evolved by the divergence of a gravitational stress tensor and a gravitational energy flux, as the traditional source-term-based approach does not guarantee the explicit conservation of energy and momentum (e.g., Jiang et al. 2013). This approach explicitly conserves the energy and the momentum of the system to machine accuracy. However, some studies (Springel 2010; Katz et al. 2016; Hanawa 2019; Mullen et al. 2021) argue that if the equations are formulated in a conservative manner the source-term-based approach can give a similar level of accuracy. Nonetheless, all of these schemes require that the error in the gravitational potential must be on the order of the round-off error (Mullen et al. 2021), which can be achieved by the FFT algorithm but is generally not possible for an iterative Poisson solver like the multigrid method. The small amount of residual in the solution obtained by an iterative method cannot ensure the round-off error level accuracy of the energy and momentum. Moreover, the conservative approach requires solving Poisson's equation

in multiple stages along with an underlying time-stepping scheme (e.g., RK2 or RK3), which can become computationally expensive. Therefore, in this work, we follow the traditional source-term-based approach with some level of approximation.

If we consider Equations (26)–(28) in the following form,

$$\frac{\partial \mathbf{U}}{\partial t} = \mathcal{L}, \quad \text{where } \mathcal{L} = -\nabla \cdot \mathbf{T} + \mathbf{S}, \quad (32)$$

then the semidiscrete RK2 method advances Equation (32) as

$$\mathbf{U}^* = \mathbf{U}^n + \Delta t \mathcal{L}^n, \quad (33)$$

$$\mathbf{U}^{n+1} = \frac{1}{2}(\mathbf{U}^n + \mathbf{U}^* + \Delta t \mathcal{L}^*), \quad (34)$$

$$= \mathbf{U}^n + \frac{\Delta t}{2}(\mathcal{L}^* + \mathcal{L}^n), \quad (35)$$

where \mathbf{U}^n is the solution of the conservative variables at the n th step and \mathcal{L}^n is the corresponding right-hand side array.

Note that the calculation of \mathcal{L}^* at the last stage requires the gravitational potential at the predictor stage (*). This demands solving Poisson's equation twice during a single RK2 step, which will be computationally expensive. Some codes use an extrapolation technique to compute the gravitational potential at the intermediate stage by linearly extrapolating the value of the potential at t^n and t^{n-1} (e.g., Bryan et al. 1995; Fryxell et al. 2000). However, in the PLUTO implementation, we provide the user with the option to choose between two alternatives: (i) solving Poisson's equation at each intermediate step, as explained earlier, or (ii) assuming that the potential remains constant within a single time step and using the acceleration \mathbf{g}^n at all intermediate stages to advance the solution to the $(n+1)$ th step. The later approximation is valid if the change in density during a time step is not dramatic and may fail for a region undergoing gravitational collapse toward a singularity. However, the benchmarks considered in Section 4 demonstrate excellent agreement with analytical predictions (when available) and with many prior studies by various authors, even for a cloud undergoing gravitational collapse to a singularity.

4. Test Results

In this section, we verify our implementation against standard astrophysical problems involving self-gravity. A list of different tests with the primary goals of each of the problems is summarized in Table 1. While some of these tests have been investigated by means of adaptive mesh refinement (AMR) techniques, here we focus attention on the convergence property of the multigrid RKL solver on a static grid.

4.1. Accuracy Tests for a 3D Model Problem

We first begin with the accuracy of the Poisson solver described in Section 2. To accomplish this, we consider a 3D model problem with a known analytical solution, namely

$$\nabla^2 \Phi = 4\pi \rho, \quad \text{where } \rho(r) = \begin{cases} \rho_0 \left(1 - \frac{r^2}{r_0^2}\right)^2, & \text{if } r \leq r_0, \\ 0, & \text{if } r > r_0, \end{cases} \quad (36)$$

Table 1
List of Numerical Tests Considered Here to Examine the Accuracy of the Self-gravity Module

Test Name	Primary Goal	Resolution
Accuracy Test	Test for the numerical accuracy of the Poisson solver	16^3 – 1024^3
Jeans Instability	Test for self-gravity with periodic boundary condition	128^3
Adiabatic Collapse of a Sphere	Check for energy conservation, shock formation, and virialization	1024^3
Collapse of an isothermal nonrotating uniform sphere	Test for the accuracy of the self-gravity module for a sphere undergoing gravitational collapse to a singularity	512^3
Collapse of an isothermal rotating uniform sphere	Check for angular momentum conservation and disk formation	512^3
Collapse of an isothermal nonrotating sphere with azimuthal density perturbation	Test for cloud fragmentation and binary structure formation	512^3

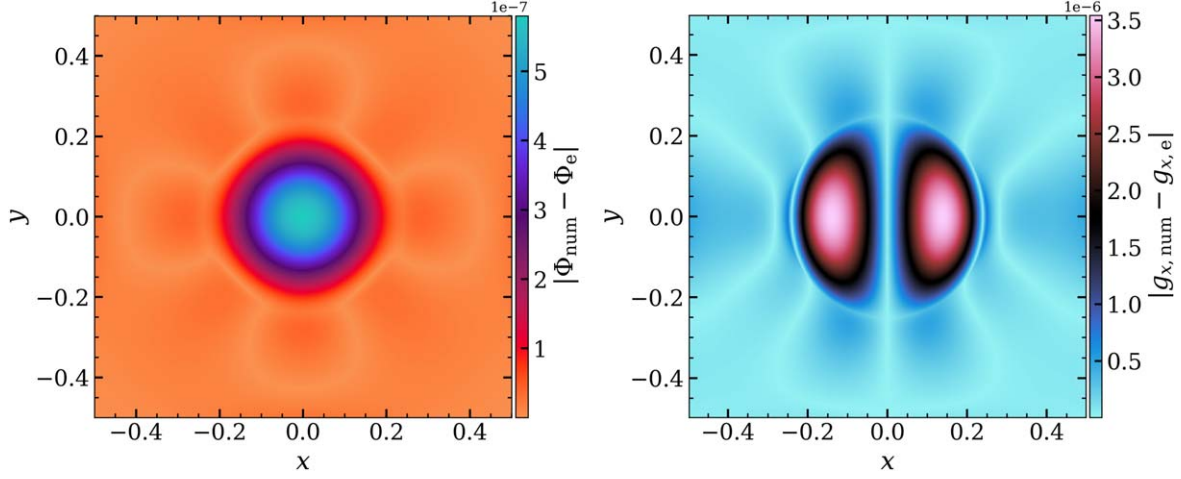


Figure 3. Error in the numerical result of the potential (left) and x -component of the acceleration (right) for the 3D model problem (Section 4.1). In both instances, we observe that the error magnitude is approximately 10^{-6} at the considered resolution of 1024^3 , which aligns with the discretization error ($\mathcal{O}(\Delta x^2)$).

where $\rho_0 = 1$ and $r_0 = 0.25$. For simplicity, we set $G = 1$. The analytical solution of Equation (36) can be derived as

$$\Phi_e(r) = \begin{cases} -\frac{2}{3}\pi\rho_0 r_0^2 + 4\pi\rho_0 \left(\frac{r^2}{6} - \frac{1}{10} \frac{r^4}{r_0^2} + \frac{1}{42} \frac{r^6}{r_0^4} \right), & \text{if } r \leq r_0, \\ -M/r, & \text{if } r > r_0, \end{cases} \quad (37)$$

where $M = 32\pi\rho_0 r_0^3/105$ is the mass of the sphere. The gravitational acceleration (\mathbf{g}) corresponding to this potential (Equation (37)) is given by

$$\mathbf{g}(r) = \begin{cases} -4\pi\rho_0 \left(\frac{r}{3} - \frac{2}{5} \frac{r^3}{r_0^2} + \frac{1}{7} \frac{r^5}{r_0^4} \right) \hat{\mathbf{r}}, & \text{if } r \leq r_0, \\ -(M/r^2) \hat{\mathbf{r}}, & \text{if } r > r_0. \end{cases} \quad (38)$$

We consider the computational domain of $(x, y, z) \in [-0.5, 0.5]$ with different resolutions ($N = 16, 32, \dots, 1024$ along each dimension) at the finest level ($l = 0$) and boundary conditions at the finest level specified using the exact solution (Equation (37)). We employ the maximum possible number of coarse levels L (including the finest level) such that $N/(2^{L-1})$ is an even number and each processor contains at least one active cell (excluding the ghost cells) of the coarsest level ($l = L - 1$). For the best performance, we use $N_{\text{pre}} = 2$ and $N_{\text{post}} = 1$ in the multigrid algorithm presented in Section 2.1.

In Figure 3, we show the error in the numerical solution of the potential (left) and the x -component of the acceleration vector for a

1024^3 box calculation of the problem described above. The acceleration has been calculated from the potential using Equation (31). In both cases, we observe that the error is of the order of the discretization error ($\mathcal{O}(\Delta x^2) \approx 10^{-6}$). To be more quantitative, we define the L^2 norm ($\|\cdot\|_2$) of the quantity $\mathcal{F}_{i,j,k}$ as

$$\|\mathcal{F}\|_2 = \left[\frac{1}{V} \sum_{i,j,k} (\mathcal{F}_{i,j,k})^2 dV_{i,j,k} \right]^{1/2}, \quad (39)$$

where $dV_{i,j,k}$ is the volume of the cell with index (i, j, k) and V is the total volume of the computational domain. In the left panel of Figure 4, we show the L^2 norm of the absolute error, defined as $(\|\Phi_{\text{num}} - \Phi_e\|) / \|\Phi_e\|$, in the numerical solution of the potential (red square) as a function of the finest grid resolution (the black dashed line represents the second-order predicted accuracy). As expected, our numerical results match the second-order convergence for the considered resolutions. The right panel of Figure 4 displays the L^2 norm of the residual of the numerical solution at the m th V-cycle (blue circles) as a function of the number of V-cycle iterations (m) for the 512^3 case. We also show the L^2 norm of the absolute error ($\|\Phi^m - \Phi_e\| / \|\Phi_e\|$) as a function of iteration number (red squares). It is worth noting that every multigrid V-cycle diminishes the error norm by approximately one order of magnitude, i.e., the convergence factor is about 10. Likewise, the absolute error is also reduced by a similar factor initially and after 4–5 V-cycles, it quickly attains the truncation error value. Thus, we only need just a few numbers of V-cycles for a

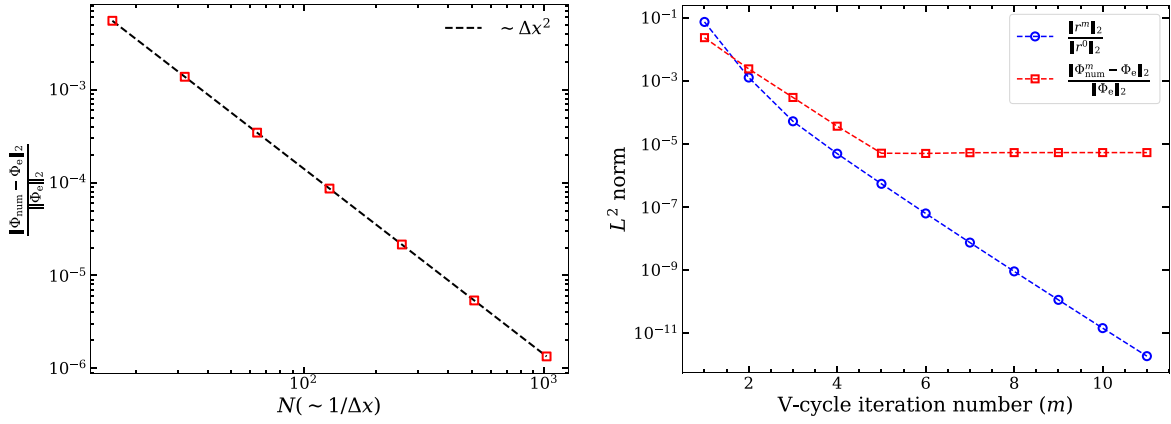


Figure 4. Left: the L^2 norm of the absolute error in the numerical solution of the potential as a function of the resolution of the finest grid level. The black dashed line is $\mathcal{O}((\Delta x)^2)$. Right: the L^2 norm of the residual (blue circles) at m th (scaled to the L^2 norm of the initial residual assuming $\Phi = 0$) V-cycle iteration and the L^2 norm of the absolute error at the m th iteration step (red square) as a function of the number of V-cycles (m). The residual at different steps have been calculated using Equation (4).

problem to attain accuracy of the order of truncation. Furthermore, if the solution at the previous step is used as the guess for the current step, it just takes 2–3 V-cycles to converge to an error level of $\max_{\Omega} |\Phi^m - \Phi^{m-1}| < 10^{-6}$, where Φ^m is the solution at the m th V-cycle.

4.2. Jeans Instability Test

There exist very few problems with periodic gravity that have analytical solutions and the Jeans instability test is one of them (Hubber et al. 2006, 2018). This problem can be used to test the accuracy of the periodic boundary conditions imposed for the gravitational potential. We briefly describe the problem below (see Hubber et al. 2006 for a detailed description).

We start from an infinite uniform medium with density ρ_0 and isothermal sound speed c_s , initially at rest. By linearizing the isothermal Euler equations (i.e., Equations (26)–(29), and replacing Equation (28) with $p = c_s^2 \rho$) with a small density perturbation $\rho_1(\mathbf{r}, t) = A\rho_0 e^{i(\mathbf{k} \cdot \mathbf{r} \pm \omega t)}$, one finds that there exists a critical value of the wavenumber (k_J) below which ω becomes imaginary and the perturbations grow exponentially in time by the action of gravity. The corresponding wavelength (λ_J) associated with k_J is called the Jeans length, given by

$$\lambda_J = \sqrt{\frac{\pi c_s^2}{G\rho_0}}, \quad (40)$$

which divides the oscillating short-wavelength perturbations from collapsing long-wavelength modes. The time period of the oscillating stable modes ($\lambda < \lambda_J$) can be derived as (Hubber et al. 2006)

$$T_{\lambda}^{\text{osc}} = \left(\frac{\pi}{G\rho_0} \right)^{1/2} \frac{\lambda}{(\lambda_J - \lambda)^{1/2}}. \quad (41)$$

Similarly, the collapse timescale (defined as the time taken by the perturbation to grow by a factor $\cosh(1)$ of the unstable wavelengths ($\lambda > \lambda_J$) is given by Hubber et al. (2006):

$$T_{\lambda}^{\text{col}} = \left(\frac{1}{4\pi G\rho_0} \right)^{1/2} \frac{\lambda}{(\lambda - \lambda_J)^{1/2}}. \quad (42)$$

For the simulation setup, we consider a box with 128^3 zones, extending from -1 to 1 along each direction with a uniform

background density ρ_0 superimposed with a sinusoidal perturbation of amplitude $A\rho_0$ along the x -direction,

$$\rho(\mathbf{r}) = \rho_0 \left[1 + A \cos\left(\frac{2\pi x}{\lambda}\right) \right]. \quad (43)$$

The medium is set to be at rest initially with $\rho_0 = 1$ and $A = 0.01$. With this numerical setup, we perform several simulations with different values of λ/λ_J covering both oscillatory and collapsing modes. We use the same value of the wavelength ($\lambda = 2$) of the perturbation for all the simulations and adjust the sound speed which changes the Jeans length (Equation (40)), yielding an effective range of values of λ/λ_J . For computing the oscillation timescale from the simulations, we first find the time evolution of the density at point $(x, y, z) = (0, 0, 0)$, which is a sinusoidal function in time. Then we calculate the period of oscillation by finding the consecutive local minima or maxima of the sinusoidal function. We consider three such periods of oscillation and take the average value, which gives the oscillation timescale. For the collapsing modes, the timescales are estimated by calculating the time it takes the density at $(x, y, z) = (0, 0, 0)$ to increase from the initial value $A\rho_0$ to a value of $A\rho_0 \cosh(1)$.

In the left panel of Figure 5, we show the results from our simulation with different values of λ/λ_J . The purple stars and red circles represent the oscillation timescales and characteristic collapse timescale for the oscillating modes ($\lambda < \lambda_J$) and growing modes ($\lambda > \lambda_J$), respectively. The blue lines are the corresponding analytical solutions (Equations (41) and (42)). The black dashed line marks the $\lambda = \lambda_J$ threshold, where the oscillation/collapse timescale tends to infinity. All the timescales are normalized to the freefall time (t_{ff}), which is given by

$$t_{\text{ff}} = \sqrt{\frac{3\pi}{32\rho_0 G}}. \quad (44)$$

It is evident that our simulation results exhibit strong agreement with the analytical prediction across the entire range, with an error of less than 1% (as shown in the right panel). It is important to acknowledge that the analytical solution to this problem strictly holds for linearized Euler’s equations. Therefore, the amplitude of the perturbations (defined by the

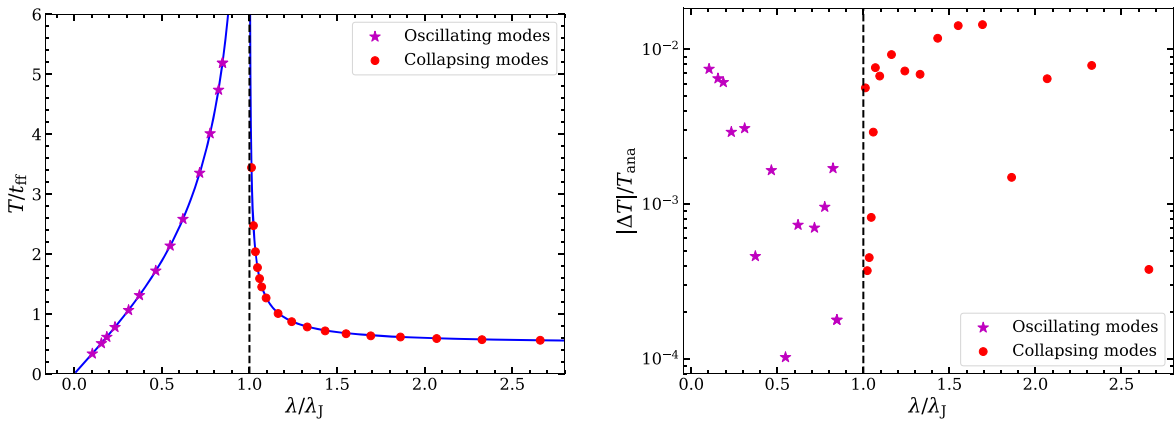


Figure 5. Left: the characteristic timescales for the evolution of a sinusoidal perturbation in the Jeans instability test. The x -axis represents the wavelength (λ) of the plane-wave perturbation in units of Jeans length (λ_J) and the y -axis is the corresponding timescale (or collapse) in units of the freefall time (t_{ff}). The blue line in the $\lambda < \lambda_J$ region represents the oscillation timescale (Equation (41)) as a function of λ of the oscillating modes and in the $\lambda > \lambda_J$ region it shows the characteristic collapse timescale (Equation (42)) of the collapsing modes. The black dashed line marks the $\lambda = \lambda_J$ point, where the oscillation/collapse timescale tends to infinity. The purple stars and the red circles are the corresponding results from the simulations, which agree very well with the analytical prediction. Right: the errors of the numerical results. The quantity ΔT is defined as $\Delta T = T_{num} - T_{ana}$, where T_{ana} and T_{num} are the analytical and numerically calculated timescales, respectively. We find that the error in the numerical result for the considered range of λ/λ_J is less than 1%.

parameter A) must be significantly smaller than the background density. For a large-amplitude perturbation, for instance, $A = 0.1$ as considered by Hubber et al. (2006, 2018), some terms in Euler’s equations cannot be neglected. Thus, the timescale calculated from the simulations will deviate from the analytical solutions (Equations (41) and (42)).

4.3. Collapse of an Adiabatic Gas Sphere (Evrard Collapse)

One very interesting problem that includes self-gravity is the collapse of an initially cold gas cloud, originally proposed by Evrard (1988). This test problem has been considered for validating the implementation of self-gravity and energy conservation in smoothed particle hydrodynamics (SPH) codes by many authors (e.g., Hernquist & Katz 1989; Springel et al. 2001; Wadsley et al. 2004; Springel 2005, 2010; Hopkins 2015; Grudić 2021, and many more) but has been rarely used in grid codes. Here, we use our grid-based method to simulate the problem. The cloud is initially at rest and the density profile is given by

$$\rho = \begin{cases} M/(2\pi R^2 r), & \text{if } r \leq R, \\ 0, & \text{otherwise,} \end{cases} \quad (45)$$

where $M = 1$ and $R = 1$. Although in the original SPH formulation the density of the background medium is set to 0, we employ here a value of 10^{-4} to avoid vacuum conditions in a grid-based code like PLUTO. The initial thermal energy per unit mass of the cloud is set to $u_{th} = 0.05$, which is very small compared to the gravitational binding energy (u_g) of the cloud ($u_g = 0.6$, assuming $G = 1$). The adiabatic index of the gas is set to be $\gamma = 5/3$.

As the initial thermal pressure support is negligible compared to the gravitational force, the gas freely falls toward the center of the sphere. The adiabatic EOS results in a pressure buildup in the central region. Eventually, a bounce back occurs, where a strong shock propagates outwards through the infalling outer part of the sphere. If a zero-gradient boundary condition is applied, this shock exits the computational domain, leading to a reduction in the total energy due to the outflowing matter. However, as we are interested in the conservation of energy, we

employ reflective boundary conditions at all the boundaries. This choice ensures that the shock remains within the boundaries and gets reflected back whenever it reaches a boundary. Furthermore, to safeguard the virialized sphere from the impact of the reflected shock, we confine our computational domain to the range of $(x, y, z) \in [-2, 2]$. We perform the simulation up to $t = 3$ and confirm that the reflected shock does not reach the virialized sphere until the end of the simulation.

The formation of the shock is shown in Figure 6, where a slice cut of the density (left) and pressure (right) distributions in the x - y plane at $t = 1.1$, overlaid with the local velocity field (white arrows), is represented. As the shock sweeps throughout the entire domain, the system virializes and eventually settles into a hydrostatic equilibrium. The problem involves the conservation of gravitational to kinetic and thermal energy and it is thus particularly sensitive to the code’s ability to conserve total energy.

In Figure 7, we show radial profiles (blue circles) of the density (left), radial velocity (middle), and entropy (right) at $t = 0.8$ (after the formation of the outward shock). For reference, we depict results from a very-high-resolution (8192 grid points) spherically symmetric 1D piecewise parabolic method (PPM) calculation of the problem (red solid lines), similar to the computation done by Steinmetz & Mueller (1993), which has been used as a common benchmark by numerous authors. In our 1D calculations, we compute the gravitational potential (Φ) by explicitly integrating the density field at each time step and imposing the boundary condition $\Phi(r) = 0$ as $r \rightarrow \infty$. Thus, the high-resolution 1D PPM calculation can be considered as a reference solution to the problem as there is no analytical solution available. We see that the results obtained from the 3D simulation conducted in Cartesian geometry exhibit very good agreement with the 1D results. The shock structure in every plot is captured correctly as well as the pre- and postshock regions.

Figure 8 presents the temporal evolution of thermal (green), kinetic (red), gravitational (blue), and total (black) energy densities as the sphere goes through different phases of evolution. We can readily observe the conversion of gravitational energy to kinetic energy (first) and then to the thermal energy of the gas. Throughout the entire evolution, the total energy remains well conserved, with an error of less than 0.4%.

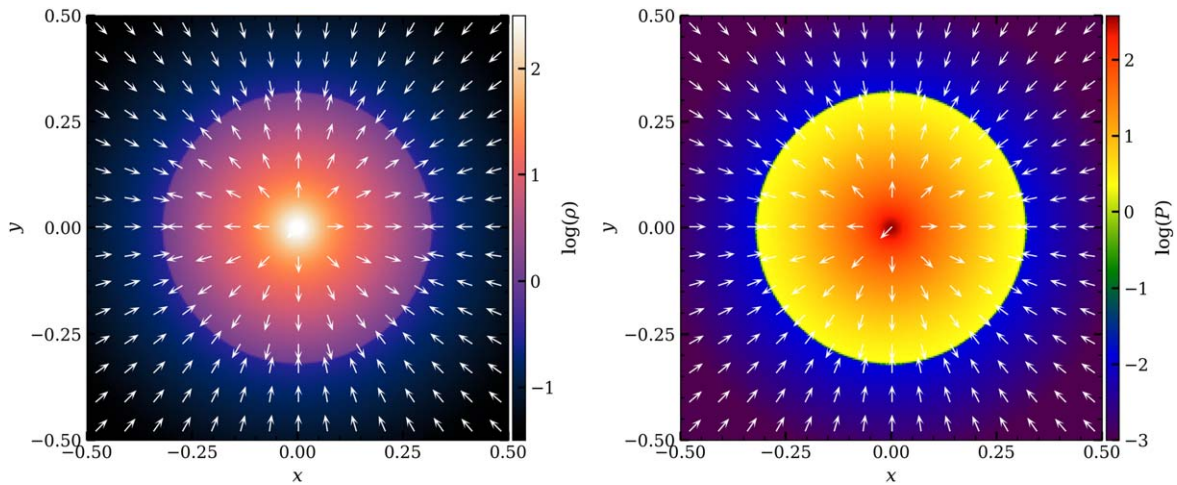


Figure 6. A slice of the density (left) and pressure (right) distributions in the x - y plane at $t = 1.1$. The white arrows represent the local velocity field. We can clearly see the shock in both the density and pressure structures, which is going outward through the still infalling outer part.

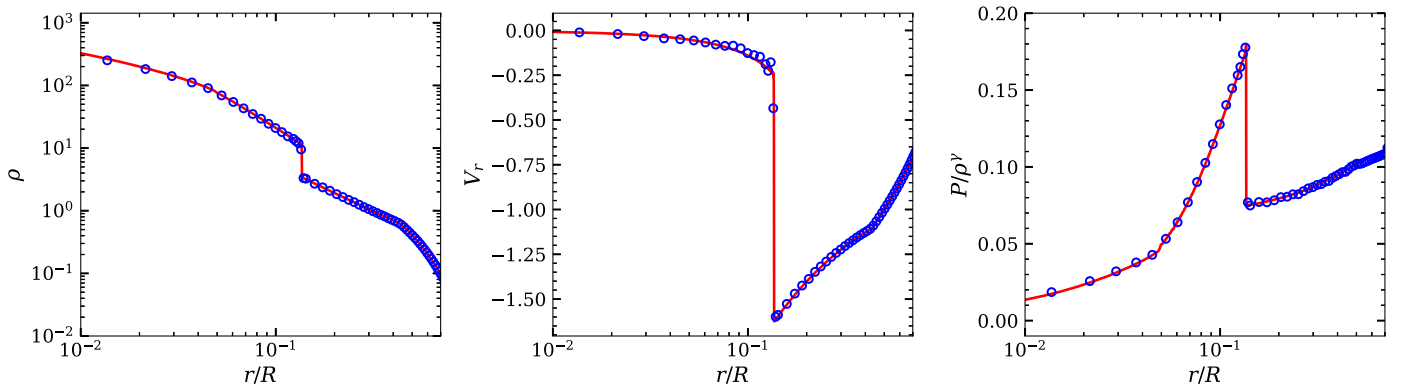


Figure 7. Radial profile of the density (left), velocity (middle), and entropy function (right, defined as P/ρ^γ) for the adiabatic collapse problem (Evrard collapse) at time $t = 0.8$. The solid red lines in each plot are the results from a very-high-resolution (4096 grid points) spherically symmetric 1D PPM calculation of the same problem, similar to the calculation by Steinmetz & Mueller (1993), which can be taken as a reference solution of the problem. We see that the results from the 3D calculation in Cartesian geometry (blue circles) excellently match with the 1D PPM results in both the pre- and postshock regions, and the shock profile is well recovered in the 3D simulation.

Upon virialization of the sphere, the kinetic energy reaches zero, and the total energy is shared between the thermal and gravitational energies.

4.4. Isothermal Collapse Tests

We now consider the collapse of an isothermal gas sphere using various initial configurations, i.e., a nonrotating uniform cloud (Section 4.4.1), a rotating uniform cloud (Section 4.4.2), and a rotating cloud with an azimuthal perturbation (Section 4.4.3). While the dynamical evolution of these systems is best described within the AMR framework, we still show that—at moderate resolution (512^3)—our gravity module successfully reproduces the main features of the results. We use a 3D cubic box for all the simulations and the gravitational potential at the boundary is calculated using multipole expansion up to $l = 4$ (Mandal et al. 2021) that is supplied to the Poisson solver as a Dirichlet-type boundary condition. For the hydrodynamics, outflow boundary conditions are used at all boundaries.

4.4.1. Nonrotating Uniform Sphere

First, we consider the collapse of a uniform nonrotating cold gas sphere, originally proposed by Truelove et al. (1998) and

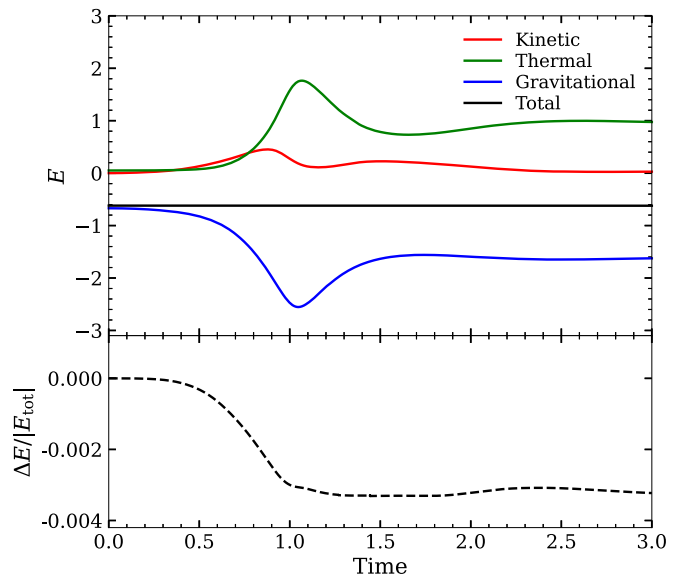


Figure 8. Top: evolution of the kinetic (red), gravitational (blue), thermal (green), and total (black) energies with time for the adiabatic collapse problem. Dimensionless units are considered for both energy and time. Bottom: the fractional error of the total energy as a function of time. Conservation of the total energy is quite good during the whole evolution, with an error $\lesssim 0.4\%$.

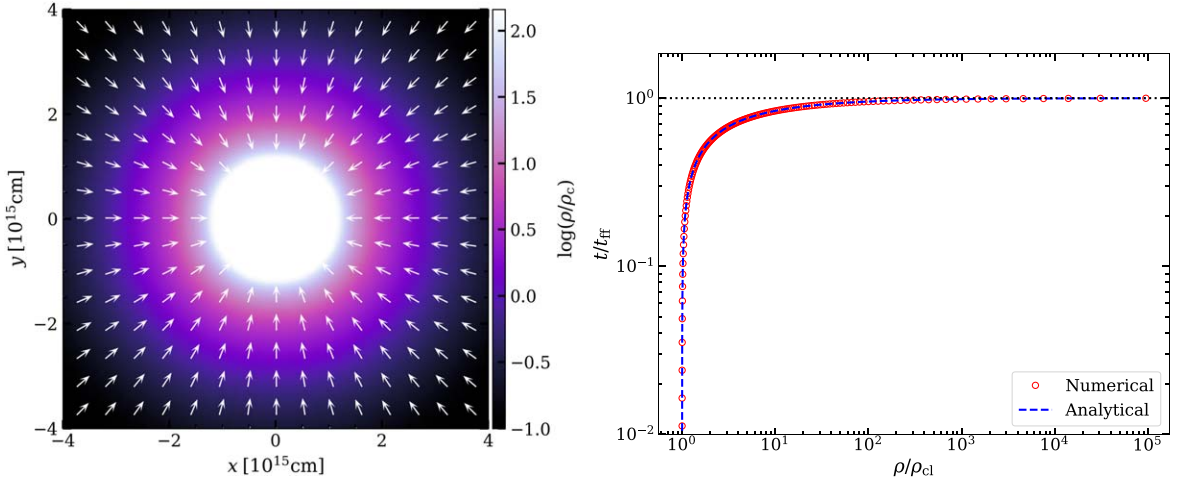


Figure 9. Left: midplane slice of the density distribution in the x - y plane for the nonrotating isothermal collapse problem at $t = 0.963t_{\text{ff}}$. The white arrows show the local velocity field. The density values are normalized to the initial cloud density (ρ_c). Right: time to reach density ρ for the nonrotating isothermal collapse test (red circles). The blue dashed line represents the analytical solution (Equation (46)).

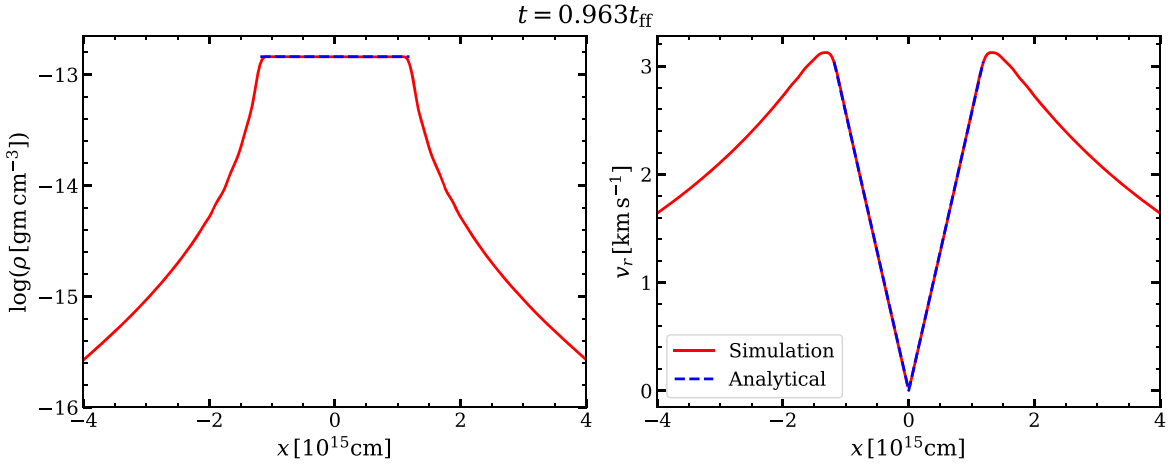


Figure 10. Cuts of the density (left) and the radial velocity field (right) along the x -axis of the left panel of Figure 9. The red solid lines are the results of the simulation. The blue dashed lines show the analytical solutions, which are only valid inside the rarefaction front.

used to test the accuracy of the grid-based code NIRVANA (Ziegler 2005). The initial conditions consist of a gas sphere of radius $R = 7.8 \times 10^{15}$ cm with density $\rho_{\text{cl}} = 10^{-15}$ g cm $^{-3}$ placed in a background medium of density $\rho_b = 0.01\rho_{\text{cl}}$. The isothermal sound speed inside the gas is set to $c_s = 0.167$ km s $^{-1}$. This is an excellent problem as it provides an analytic solution in the pressureless ($T=0$) case, which consists of self-similar evolution. However, as we cannot model a pressureless and infinite medium, the solution consists of a rarefaction wave traveling radially inward from the cloud surface superimposed on a self-similar collapse. At any time, the region inside the rarefaction wave remains uniform and the density increases self-similarly, which can be described by the analytical solution (Truelove et al. 1998). This is illustrated in the left panel of Figure 9, where we show a slice of the density distribution in the x - y plane and the velocity field (white arrows) at $t = 0.963t_{\text{ff}}$. The central density plateau is clearly visible, which still follows the pressureless solution and is unaffected by the rarefaction front. The medium with a radially decreasing density beyond the central plateau represents the outer part of the sphere which has been affected by the inward rarefaction wave that initially appeared at the cloud surface. At

$t = t_{\text{ff}}$, it approaches a singular state of infinite density, and the temperature of the medium is chosen in such a way that the collapse occurs before the rarefaction wave reaches the cloud center. The time to build up a density ρ inside the rarefaction wave can be derived analytically (see Truelove et al. 1998),

$$\frac{t_\rho}{t_{\text{ff}}} = \frac{2}{\pi} \left[\eta + \frac{1}{2} \sin(2\eta) \right], \quad \text{where } \eta = \arccos \left[\left(\frac{\rho}{\rho_{\text{cl}}} \right)^{-1/6} \right]. \quad (46)$$

In the right panel of Figure 9, we compare the analytical expectation (Equation (46), blue dashed line) with our numerical results (red circles). It is evident that our numerical results align remarkably well with the analytical prediction across the entire range of density and time. In the simulation, the density increases by more than five orders of magnitude at $0.9982t_{\text{ff}}$ and the code breaks at $0.99994t_{\text{ff}}$ as we cannot numerically reach the singular state. The primary difference between our results and those of Truelove et al. (1998) is that their solution features a time delay with respect to the analytic solution. This is due to the image boundary conditions they used for the gravitational potential,

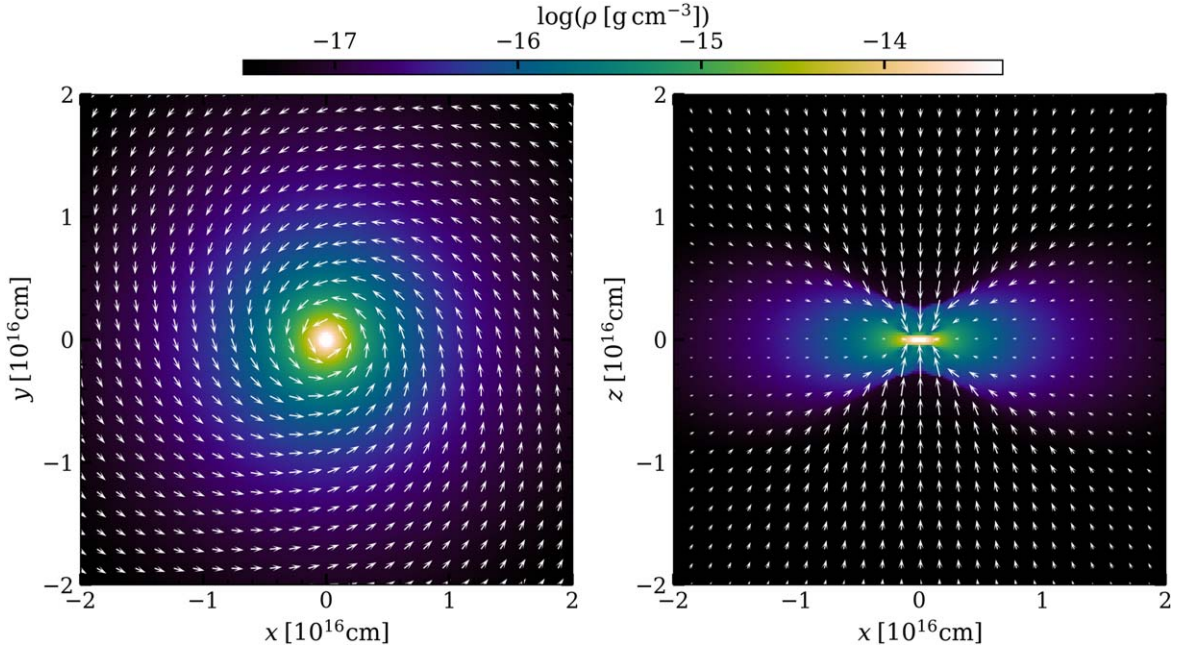


Figure 11. Slice of the density distribution in the x - y (left) and x - z (right) planes for the rotating uniform sphere test at $t = 1.26t_{\text{ff}}$. Arrows in each plot represent the velocity field.

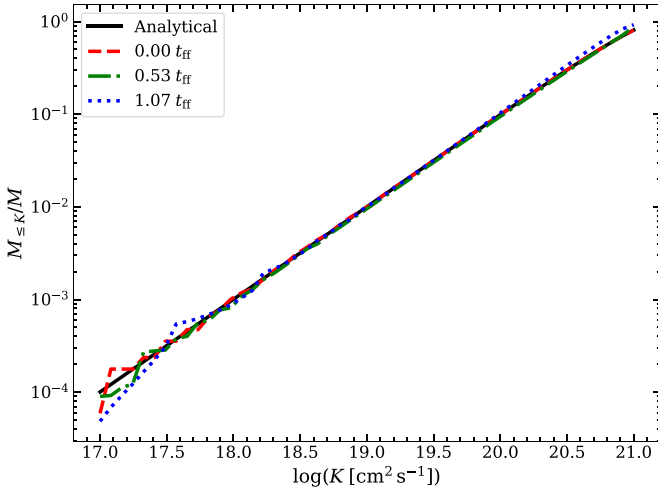


Figure 12. The SAM (K) spectrum at different times for the rotating uniform cloud collapse test. The y-axis represents the total mass inside the cloud with SAM less than or equal to K normalized to the initial total mass of the cloud. The red, green, and blue lines are the spectrum calculated from the simulation at $t = 0, 0.53,$ and $1.07t_{\text{ff}}$. The black line show the theoretical estimate (Equation (50)).

which mimics the existence of neighboring clouds that oppose the collapse by the attracting force. As the image cloud separation is increased in their study, the time delay decreases. In our simulation, on the other hand, we use the Dirichlet boundary conditions for the gravitational potential, which are calculated from multipole expansion considering $\Phi(r) \rightarrow 0$ as $r \rightarrow \infty$. This better represents an isolated system and eventually leads to better agreement with the analytical solution. A similar result is also reported by Ziegler (2005), where they have also used multipole expansion to specify the boundary conditions and found no time delay compared to the analytical solution.

We can also derive the mass (M_{rf}), size (r_{rf}), and the radial velocity ($v(r, \rho)$) of the unaffected region inside the rarefaction

wave as functions of central density (see Truelove et al. 1998 for the derivation). The mass inside the rarefaction wave at a density ρ is given as

$$M_{\text{rf}} = M \left\{ 1 - 2 \frac{c_s}{v_{\text{ff}}} \arctan \left[\left(\frac{\rho}{\rho_{\text{cl}}} \right)^{1/3} - 1 \right]^{1/2} \right\}^3, \quad (47)$$

where M is the mass of the cloud and $v_{\text{ff}} = \sqrt{2GM/R}$ is the freefall velocity. Thus, from Equation (47), we can easily calculate the radius of the central region inside the rarefaction front (r_{rf}) to be

$$r_{\text{rf}} = R \left\{ 1 - 2 \frac{c_s}{v_{\text{ff}}} \arctan \left[\left(\frac{\rho}{\rho_{\text{cl}}} \right)^{1/3} - 1 \right]^{1/2} \right\}. \quad (48)$$

From energy conservation, the flow velocity can be expressed as a function of the radius r and of the density ρ (Truelove et al. 1998),

$$v(r, \rho) = -v_{\text{ff}} \frac{r}{r_{\text{rf}}}, \left[\left(\frac{\rho}{\rho_{\text{cl}}} \right)^{1/3} - 1 \right]^{1/2} \quad \text{for } r \leq r_{\text{rf}}. \quad (49)$$

Figure 10 displays cuts of the density (left) and radial velocity (right) along the x -axis at $t = 0.963t_{\text{ff}}$. The red solid lines in both panels represent the results from our simulation while the blue dashed lines are the corresponding analytical solutions, valid inside the rarefaction front. For both cases, the simulation results agree well with the analytical prediction inside the rarefaction front.

4.4.2. Rotating Uniform Cloud

As the second test of the isothermal collapse problem, we consider the collapse of a rotating uniform cloud originally proposed by Norman et al. (1980) and later investigated by

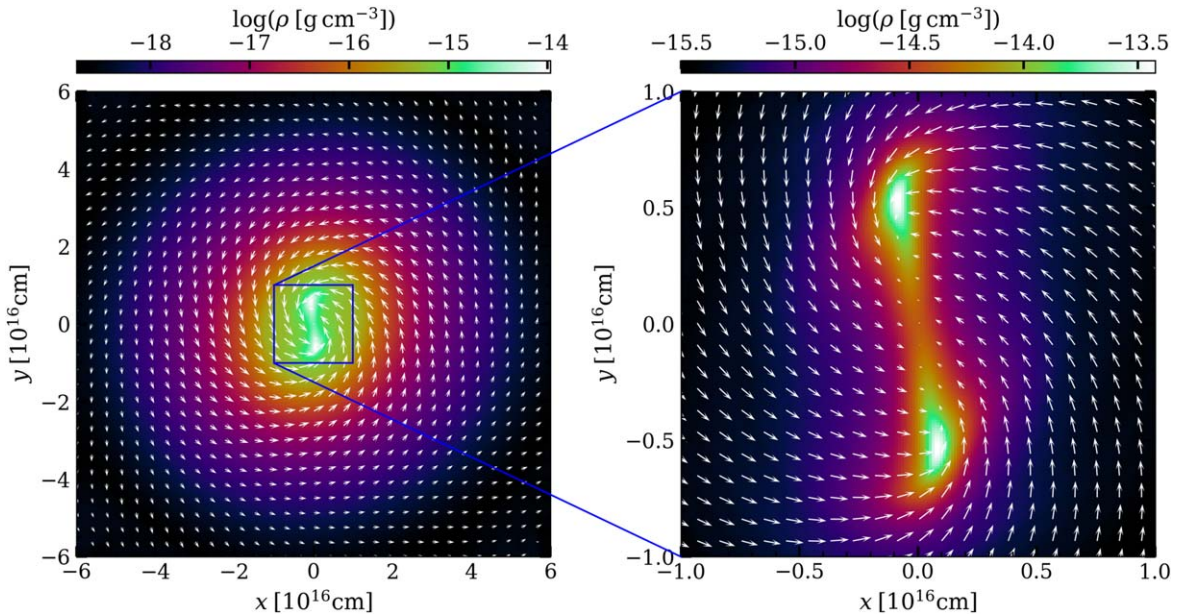


Figure 13. Left: slice of the log density distribution in the x - y plane for the rotating perturbed cloud problem at $t = 1.24t_{\text{ff}}$, roughly when the two structures become gravitationally bound and undergo gravitational collapse independently. Right: only the central 2×10^{16} cm region of the left panel just to clearly show the structures.

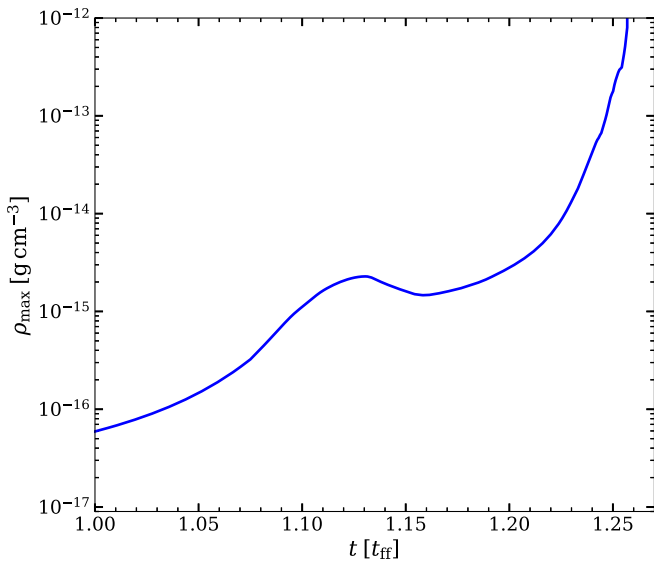


Figure 14. Maximum density in the simulation box as a function of time for the perturbed uniform cloud test.

various authors (Boss & Myhill 1992; Truelove et al. 1998; Ziegler 2005). This problem is particularly useful to verify the ability to conserve the angular momentum by numerical codes. The initial conditions comprise a uniform spherical gas sphere of mass $M = 1 M_{\odot}$ and radius $R = 7.01 \times 10^{16}$ cm, giving density $\rho_0 = 1.26 \times 10^{-18}$ g cm $^{-3}$. The cloud has a temperature of $T = 5$ K and rotates with uniform angular velocity $\Omega = 3.04 \times 10^{-13}$ rad s $^{-1}$. This yields an initial ratio of the thermal to gravitational energy of $\alpha = 0.54$ and a ratio of the rotational to gravitational energy of $\beta = 0.08$. The surrounding medium is initialized with a constant density of $\rho_b = 0.01 \rho_0$.

As shown by Norman et al. (1980), poor conservation of angular momentum can produce an artificial ring-like structure instead of a disk in the process of collapse. On the other hand, if the conservation is adequate, the sphere forms a disk with an

increasingly high central density as it evolves. In Figure 11, we show a slice of the density distribution in the x - y (left) and x - z (right) planes at $t = 1.26t_{\text{ff}}$. From a first inspection, our results are qualitatively comparable to those of Truelove et al. (1998) and Ziegler (2005). We can clearly see a disk has been formed and the velocity field in the x - z plane shows that the collapse is progressing toward a singular state. No evidence of a ring structure has been found until the end of the simulation. Thus, the angular momentum is conserved quite well, although it is not explicitly designed to be in our scheme. A better method to quantify the conservation of angular momentum is the calculation of the spectrum of specific angular momentum (SAM; symbol K) at different times as proposed by Norman et al. (1980), which is defined by the total mass inside the cloud less than or equal to a given value of K . For an ideal (inviscid) fluid, this distribution remains conserved and, for an initially rotating uniform cloud, the analytical form of the spectrum is given by Truelove et al. (1998),

$$M_{\leq K} = M \left[1 - \left(1 - \frac{K}{\Omega R^2} \right)^{3/2} \right], \quad (50)$$

where Ω is the uniform angular velocity of the cloud. Ideally, the minimum and maximum extent of K should be $K_{\min} = 0$ (corresponding to the rotation axis) and $K_{\max} = \Omega R^2$ (corresponding to the surface of the sphere). However, due to finite resolution, any structure below the resolution of the simulation Δx cannot be resolved, which corresponds to $K_{\min} \sim \Omega \Delta x^2 = 10^{18.2}$ cm 2 s $^{-1}$ for our setup. Note that, due to the inability to resolve the density and velocity structures at this small scale, the numerical result may not agree with the analytical solution. However, if the angular momentum is conserved, then the shape of the distribution should not change with time irrespective of the initial shape. Thus, we compare the distributions at different times in Figure 12, where we show the SAM spectrum from our simulation at $t = 0$ (red), $t = 0.53t_{\text{ff}}$ (green), and $t = 1.07t_{\text{ff}}$ (blue). For reference, the

corresponding analytical estimate (Equation (50)) is also plotted in black. We notice the SAM distributions at different times agree satisfactorily at larger values of K within $\lesssim 1\%$ error, except for the distribution at a later stage (blue), where the evolution is believed to be affected by the surrounding low-density medium of the cloud and the average fractional error in the numerical distribution is about $\sim 10\%$.

4.4.3. Rotating Cloud with Azimuthal Perturbation

Lastly, we consider a very demanding problem of fragmentation of an initially perturbed rotating cloud first proposed by Boss & Bodenheimer (1979) and employed by several authors (e.g., Burkert & Bodenheimer 1993; Truelove et al. 1997, 1998; Klein 1999; Springel 2005; Ziegler 2005; Hubber et al. 2018; Grudić 2021) in different versions. The original Boss & Bodenheimer test consisted of an isothermal uniform cloud subjected to a 50% initial azimuthal $m=2$ mode perturbation in density. However, Burkert & Bodenheimer (1993) introduced the same problem with a 10% initial perturbation in density and showed that the evolution and subsequent fragmentation of such a small perturbation are more challenging to simulate and constitute a good benchmark for a numerical code. This is the version we adopt here, which consists of an isothermal spherical cloud with sound speed $c_s = 0.167 \text{ km s}^{-1}$, mass $M = 1 M_\odot$, and with a radius of $R = 5 \times 10^{16} \text{ cm}$. The cloud rotates with an angular velocity of $\omega = 7.4 \times 10^{-13} \text{ rad s}^{-1}$, which gives energy ratios of $\alpha = 0.26$ and $\beta = 0.16$. The uniform density of the cloud is modified by an azimuthal $m=2$ mode perturbation in the following way,

$$\rho = \begin{cases} \rho_0[1 + 0.1 \cos(2\phi)], & \text{if } r \leq R, \\ 0, & \text{if } r > R, \end{cases} \quad (51)$$

where ϕ is the azimuthal angle around the rotation axis and $\rho_0 = 3.82 \times 10^{-18} \text{ g cm}^{-3}$ is the underlying constant density. The density of the medium outside the sphere is set to be $0.01\rho_0$.

Figure 13 illustrates an equatorial slice of the density distribution (left panel) and a close-up of the central structure (right panel) at $t = 1.24t_{\text{ff}}$. An elongated bar-like structure has formed connecting two high-density regions. The velocity field demonstrates that there are converging gas flows toward the center of mass of each structure as a result of gravitational collapse. This can be seen in Figure 14, where we show the maximum density inside the simulation box as a function of evolution time. Similar behavior has also been reported by Bate & Burkert (1997), Springel (2005), and Grudić (2021). However, we find that in our case the collapse proceeds at a somewhat slower rate when compared to Springel (2005) and Grudić (2021) owing to the limited resolution near the collapsing regions. The low-resolution runs of the same test problem in Springel (2005) also exhibit similar behavior.

It is worth noting that the problem at hand exhibits a significant range of dynamics. This necessitates the use of high resolution in the vicinity of the collapsing structure to witness the fragmentation collapse into a singular state, as illustrated in the works of Truelove et al. (1998) and Ziegler (2005). Here, the employment of a uniform Cartesian grid prevents the computations from reproducing the elongation and subsequent gravitational collapse to a singular state. From this perspective our results are fairly consistent with those of Bate & Burkert

(1997) and Springel (2005) and adequately reproduce the major features of the test problem despite the limited resolution.

5. Summary

We have presented a novel implementation of an iterative method for solving Poisson's equation based on the RKL time-stepping scheme developed for solving parabolic PDEs. Our solver utilizes a relaxation scheme to achieve convergence on the solution of an elliptic problem sought as the steady-state solution of a parabolic equation, whose time discretization is governed by the RKL scheme. The algorithm is part of a V-cycle multigrid method and it has been implemented as a simple and efficient gravity solver in the PLUTO code, to address self-gravitational hydrodynamics. We demonstrate that our approach offers a potential substitute to conventional iterative techniques like G-S or SOR, given its significantly superior convergence rate when compared to the G-S scheme, while also removing the uncertainties connected to the optimal overrelaxation parameter in the SOR method. Besides, being a fully parallel algorithm, our method incurs a lower communication overhead between processors compared to G-S or SOR, which are half-parallel (see Appendix B for the performance of these solvers for a model problem).

We have employed our methodology to perform a comprehensive set of astrophysical test problems, designed to assess different aspects of the code. In a 3D static problem, we have verified that the solver achieves second-order accuracy for both the potential and force, as expected. Our calculations of the timescales for the collapsing and oscillating modes of the Jeans instability test agree well with the theoretical expectations, with errors of less than 1%. The shock structures observed in the density, velocity, and entropy functions of the adiabatic collapse problem match closely with those obtained from a highly resolved 1D PPM calculation, which serves as a reference solution. Moreover, our code conserves the total energy up to 1%, for the adiabatic collapse problem. We have tracked the collapse of a uniform nonrotating sphere and observed that the central density increases by \sim five orders of magnitudes and the corresponding timescale matches very well with the analytical prediction. In the collapse of a rotating uniform sphere, which encompasses a density increase of six decades, we have achieved angular momentum conservation up to 1%. However, at the later stage, the conservation degrades ($\sim 10\%$) due to the effect from the surrounding medium. Lastly, we have reported the fragmentation and subsequent formation of binary structures of an initially perturbed rotating sphere where the density increases by more than six orders of magnitude, and the results favorably compare to those obtained by previous investigators. Although the test problems considered here can benefit even more from the employment of AMR, we have demonstrated that our gravity module can reproduce the major features of the results using a uniform static grid already at moderate resolution.

Future extensions to AMR require changes to the cycling strategy in the multigrid algorithm and to the treatment of the boundary layers for each multigrid level, which are well developed by many authors (Ricker 2008; Guillet & Teysier 2011; Wang & Yen 2020). Smoothing, restriction, and prolongation operators as well as the coarsest grid solver should not be affected by the introduction of AMR.

Acknowledgments

We like to thank Christoph Federrath for the useful discussion about boundary conditions. A.M. thanks Moun Meenakshi for her valuable input and constant support throughout the project. We gratefully acknowledge the use of the Pegasus high-performance computing facilities of IUCAA, Pune (<http://hpc.iucaa.in>). Parts of the results in this work make use of the color maps in the CMasher package.

Software: PLUTO⁶ (Mignone et al. 2007), NumPy (Harris et al. 2020), Matplotlib (Hunter 2007), and CMasher (van der Velden 2020).

Appendix A

Convergence Property of the RKL-based Poisson Solver

The RKL method is a class of STS methods that has been developed to deal with the restrictive time step for an explicit scheme for parabolic equations (Meyer et al. 2012, 2014). This allows us to consider a much longer time step (τ) than the maximum permissible explicit parabolic time step (Δt_{par}) without affecting the solution. The parabolic diffusion equation can be written in the following form,

$$\frac{du}{dt} = \mathbf{M}u(t), \quad (\text{A1})$$

where \mathbf{M} is a symmetric, constant coefficient matrix resulting from the space discretization of the parabolic operator. The STS methods are a class of stabilized Runge–Kutta methods where one time step (τ) consists of additional substeps (s) to ensure stability. The solution after one τ step can be expressed in terms of the amplification factor R_s as

$$u(t + \tau) = R_s(\tau\mathbf{M})u(t). \quad (\text{A2})$$

The stability of the solution is guaranteed if $|R_s(\tau\lambda)| \leq 1$ for all λ , where λ is the eigenvalues of \mathbf{M} . Now the amplification factor $R_s(\tau\lambda)$, also known as the stability polynomial for an s -stage STS scheme, can be defined as

$$R_s(\tau\lambda) = \prod_{i=1}^s (1 + \lambda\Delta t_i), \quad (\text{A3})$$

with $\sum_{i=1}^s \Delta t_i = \tau$. Matching the coefficients in Equation (A3) with the analytic expansion of Equation (A1), as given by

$$u(t + \tau) = e^{\tau\mathbf{M}}u(t) \approx \left[1 + (\tau\mathbf{M}) + \frac{1}{2}(\tau\mathbf{M})^2 + \dots \right] u(t), \quad (\text{A4})$$

gives the temporal accuracy of the solution.

In RKL methods, shifted Legendre polynomials are used as stability polynomials. For a general s -stage RKL scheme, the stability polynomials are chosen as

$$R_s(\tau\lambda) = a_s + b_s P_s(w_0 + w_1(\tau\lambda)), \quad (\text{A5})$$

where $w_0 = 1$ for all RKL methods. For a first-order RKL scheme, $a_s = 0$, $b_s = 1$, and $w_1 = 2/(s^2 + s)$. In the rest of the derivation, we consider only the first-order RKL technique. Thus, for simplicity, whenever ‘‘RKL’’ is mentioned it implies the first-order RKL method. Therefore, the stability polynomial

of an s -stage RKL scheme takes the following form,

$$R_s(\tau\lambda) = P_s \left(1 + \frac{2}{s^2 + s} \tau\lambda \right). \quad (\text{A6})$$

Thus, the j th stage of an s -stage RKL scheme can be expressed in terms of the corresponding stability polynomial as

$$u_j = P_j \left(1 + \frac{2}{s^2 + s} \tau\lambda \right) u^{(n)},$$

$$u^{(n+1)} = u_s = P_s \left(1 + \frac{2}{s^2 + s} \tau\lambda \right) u^{(n)}, \quad (\text{A7})$$

where $u^{(n)}$ represents the solution at the beginning of the τ -step, u_j denotes the solution at the j th intermediate step, and $u^{(n+1)}$ corresponds to the solution after advancing one τ -step. Throughout the text, we use subscripts to represent the intermediate steps within one τ -step and the superscripts refer to each τ -step.

In order to calculate the stability polynomials, we can use the three-point recursion relation of Legendre polynomials,

$$P_j(x) = \frac{2j-1}{j} x P_{j-1}(x) - \frac{j-1}{j} P_{j-2}(x). \quad (\text{A8})$$

Replacing $x = 1 + 2\tau\lambda/(s^2 + s)$, we can easily show that the stability polynomials of the RKL methods obey the following recursion relation,

$$P_j \left(1 + \frac{2}{s^2 + s} \tau\lambda \right) = \frac{2j-1}{j} \left(1 + \frac{2}{s^2 + s} \tau\lambda \right) P_{j-1} \left(1 + \frac{2}{s^2 + s} \tau\lambda \right) - \frac{j-1}{j} P_{j-2} \left(1 + \frac{2}{s^2 + s} \tau\lambda \right). \quad (\text{A9})$$

Using Equation (A9), one can write the s -stage RKL scheme for \mathbf{M} as follows,

$$Y_0 = u^{(n)},$$

$$Y_1 = Y_0 + \tilde{\mu}_1 \tau \mathbf{M} Y_0,$$

$$Y_j = \mu_j Y_{j-1} + \nu_j Y_{j-2} + \tilde{\mu}_j \tau \mathbf{M} Y_{j-1}, \quad 2 \leq j \leq s,$$

$$u^{(n+1)} = Y_s, \quad (\text{A10})$$

where the coefficients μ_j , ν_j , and $\tilde{\mu}_j$ can be obtained from the recursion relation (Equation (A9)) and given by

$$\mu_j = \frac{2j-1}{j},$$

$$\nu_j = \frac{1-j}{j},$$

$$\tilde{\mu}_j = \mu_j w_1 = \frac{2j-1}{j} \frac{2}{s^2 + s}. \quad (\text{A11})$$

The stability condition of Legendre polynomials provides the maximum permissible time step τ for an s -stage RKL method. Legendre polynomials are bounded within the range $[-1, 1]$, and therefore, to ensure solution stability, the condition $(1 - w_1 \tau |\lambda_{\text{max}}|) \geq -1$ must hold true. Here, λ_{max} represents the maximum (negative) eigenvalue of the operator \mathbf{M} . This condition determines the maximum value of τ applicable to an

⁶ <http://plutocode.ph.unito.it>

s -stage RKL scheme, yielding,

$$\tau_{\max} = \frac{2}{w_1 |\lambda_{\max}|} = \Delta t_{\text{par}} \frac{s^2 + s}{2}, \quad (\text{A12})$$

where $\Delta t_{\text{par}} = 2/|\lambda_{\max}|$ is the explicit parabolic time step.

A.1. Poisson Equation as a Limiting Case of the Diffusion Equation

A simple extension of the diffusion equation (Equation (A1)) can be made to accommodate a time-independent source term (f) on the right-hand side,

$$\frac{du}{dt} = \mathbf{M}u(t) - f. \quad (\text{A13})$$

The initial solution u reaches equilibrium as $t \rightarrow \infty$ and $du/dt \rightarrow 0$, which essentially becomes the solution of Poisson's equation, i.e.,

$$\mathbf{M}u - f = 0. \quad (\text{A14})$$

Thus, successively applying one τ -step to the previous step of Equation (A13) until $t \rightarrow \infty$ gives the solution of Poisson's equation for a given source term f .

To fit the source term (f) in the intermediate stages of an s -stage RKL, we replace $\mathbf{M}Y_j \rightarrow \mathbf{M}Y_j - f$ in Equation (A10), which takes the following form,

$$\begin{aligned} Y_0 &= u^{(n)}, \\ Y_1 &= Y_0 + \tilde{\mu}_1 \tau \mathbf{M}Y_0 - \tilde{\mu}_1 \tau f, \\ Y_j &= \mu_j Y_{j-1} + \nu_j Y_{j-2} + \tilde{\mu}_j \tau \mathbf{M}Y_{j-1} - \tilde{\mu}_j \tau f, \quad 2 \leq j \leq s, \\ u^{(n+1)} &= Y_s. \end{aligned} \quad (\text{A15})$$

Using Equation (A15), it can be shown that the solution of Equation (A13) after one τ step in an s -stage RKL scheme can be written as

$$u^{(n+1)} = P_s(1 + w_1 \tau \mathbf{M})u^{(n)} - F_{(s-1)}(w_1 \tau \mathbf{M})w_1 \tau f, \quad (\text{A16})$$

where F is a polynomial of degree $(s-1)$. Therefore, Equation (A16) can be thought of as one iteration from $u^{(n)}$ to $u^{(n+1)}$ with the RKL iteration matrix

$$\mathbf{R}_{s,\text{RKL}} = P_s(1 + w_1 \tau \mathbf{M}). \quad (\text{A17})$$

Thus, starting from a guess solution $u^{(0)}$, we can successively apply Equation (A16) until m steps, for which $|u^{(m)} - u^{(m-1)}| < 10^{-d}$, where 10^{-d} is some specified tolerance level.

A.2. Convergence Criteria

Consider the initial guess solution of Equation (A16) is $u^{(0)}$, and we apply one iteration step. Then, the solution after one τ step is

$$u^{(1)} = \mathbf{R}_{s,\text{RKL}}u^{(0)} - F_{(s-1)}(w_1 \tau \mathbf{M})w_1 \tau f. \quad (\text{A18})$$

Now, if v is the exact solution of Equation (A14), then the iteration steps do not change the solution, i.e.,

$$v = \mathbf{R}_{s,\text{RKL}}v - F_{(s-1)}(w_1 \tau \mathbf{M})w_1 \tau f. \quad (\text{A19})$$

Subtracting Equation (A18) from Equation (A19), we get

$$v - u^{(1)} = \mathbf{R}_{s,\text{RKL}}(v - u^{(0)}) \Rightarrow e^{(1)} = \mathbf{R}_{s,\text{RKL}}e^{(0)}, \quad (\text{A20})$$

where $e^{(j)} = v - u^{(j)}$ is the error in the approximated solution at the j th step. Thus, it is possible to reduce the error from the previous step, if and only if

$$\|\mathbf{R}_{s,\text{RKL}}\|_p < 1, \quad \forall p \in (1, \infty), \quad (\text{A21})$$

where $\|\cdot\|_p$ refers to the p -norm of the matrix. Now, if we consider $p=2$ (the Euclidean norm), then for the symmetric matrix $\mathbf{R}_{s,\text{RKL}}$, we have

$$\begin{aligned} \|\mathbf{R}_{s,\text{RKL}}\|_2 &= \sqrt{\rho((\mathbf{R}_{s,\text{RKL}})^T \mathbf{R}_{s,\text{RKL}})} \\ &= \sqrt{\rho(\mathbf{R}_{s,\text{RKL}}^2)} = \rho(\mathbf{R}_{s,\text{RKL}}), \end{aligned} \quad (\text{A22})$$

where $\rho(\mathbf{R}_{s,\text{RKL}})$ is the spectral radius of matrix $\mathbf{R}_{s,\text{RKL}}$ which is defined as

$$\rho(\mathbf{R}_{s,\text{RKL}}) = \max|\lambda(\mathbf{R}_{s,\text{RKL}})|, \quad (\text{A23})$$

where $\lambda(\mathbf{R}_{s,\text{RKL}})$ are the eigenvalues of $\mathbf{R}_{s,\text{RKL}}$. Thus, from Equations (A21) and (A22), we find that the convergence to the exact solution for any initial guess solution is possible through successive iterations, if and only if

$$\rho(\mathbf{R}_{s,\text{RKL}}) < 1. \quad (\text{A24})$$

So, after m iterations, the error in the m th approximation is given by

$$e^{(m)} = \mathbf{R}_{s,\text{RKL}}^m e^{(0)}. \quad (\text{A25})$$

Therefore, if we want to reduce the initial error by an order of d , we must need

$$\frac{\|e^{(m)}\|_2}{\|e^{(0)}\|_2} = \|\mathbf{R}_{s,\text{RKL}}\|_2^m \leq 10^{-d} \Rightarrow [\rho(\mathbf{R}_{s,\text{RKL}})]^m \leq 10^{-d}. \quad (\text{A26})$$

The minimum number of iterations (m) needed to reduce the initial error by d orders of magnitude in an s -stage RKL method is given by

$$m \geq -\frac{d}{\log_{10}[\rho(\mathbf{R}_{s,\text{RKL}})]} = \frac{d}{\log_{10}[\rho(P_s(1 + w_1 \tau \mathbf{M}))]}. \quad (\text{A27})$$

A.3. Spectral Analysis

In order to find the spectral radius of the RKL iteration matrix ($\mathbf{R}_{s,\text{RKL}}$), we have to find how the eigenvalues ($\lambda_{k,\text{RKL}}(s)$) behave. From Equation (A17), we note that $\mathbf{R}_{s,\text{RKL}}$ is a polynomial of \mathbf{M} . Thus, if $\lambda_{k,\text{M}}$ is the k th eigenvalue of \mathbf{M} , then the eigenvalues of $\mathbf{R}_{s,\text{RKL}}$ is given by

$$\lambda_{k,\text{RKL}}(s) = P_s(1 + w_1 \tau \lambda_{k,\text{M}}). \quad (\text{A28})$$

We first consider a 1D problem with grid points at $j=0, 1, \dots, N$ with a grid spacing of Δx . The \mathbf{M} matrix for this problem is given by

$$\mathbf{M} = \frac{1}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}_{(N-1) \times (N-1)}. \quad (\text{A29})$$

The eigenvalues of \mathbf{M} are

$$\lambda_{k,M} = -\frac{4}{(\Delta x)^2} \sin^2\left(\frac{k\pi}{2N}\right), \quad k \in \mathbb{Z}^+ \text{ and } 1 \leq k \leq N-1. \quad (\text{A30})$$

Also, it can be easily shown that the eigenvectors of \mathbf{M} are given by

$$\mathbf{\Lambda}_{k,j,M} = \sin\left(\frac{jk\pi}{N}\right), \quad 1 \leq k \leq N-1, \quad 0 \leq j \leq N, \quad (\text{A31})$$

where $\mathbf{\Lambda}_{k,j,M}$ is the j th component of the k th eigenvector of the \mathbf{M} operator. Interestingly, the eigenvectors of \mathbf{M} are simply the Fourier modes in 1D.

Now, assuming $\tau = \tau_{\max} = 2/(w_1 \max\{\lambda_{k,M}\}) = (\Delta x)^2/2w_1$ (where $\max\{\lambda_{k,M}\} = 4/(\Delta x)^2$), the eigenvalues of $\mathbf{R}_{s,\text{RKL}}$ are given by

$$\lambda_{k,\text{RKL}}(s) = P_s\left(1 - 2\sin^2\left[\frac{k\pi}{2N}\right]\right) = P_s\left(\cos\left[\frac{k\pi}{N}\right]\right). \quad (\text{A32})$$

From the properties of the Legendre polynomials, we know that $P_s(x)$ is a symmetric and antisymmetric function when s is even or odd, respectively. Thus, $\lambda_{k,\text{RKL}}(s)$ has the following property:

$$\lambda_{k,\text{RKL}}(s) = (-1)^s \lambda_{N-k,\text{RKL}} \text{ or } |\lambda_{k,\text{RKL}}(s)| = |\lambda_{N-k,\text{RKL}}(s)|. \quad (\text{A33})$$

Therefore, for calculating the spectral radius for $\mathbf{R}_{s,\text{RKL}}$, we can only consider $1 \leq k \leq N/2$ eigenmodes and the spectral radius of $\mathbf{R}_{s,\text{RKL}}$ can be expressed as

$$\rho(\mathbf{R}_{s,\text{RKL}}) = \max_{1 \leq k \leq N/2} \left\{ \left| P_s\left(\cos\left[\frac{k\pi}{N}\right]\right) \right| \right\}. \quad (\text{A34})$$

In Figure 15, the absolute values of the eigenvalues of $\mathbf{R}_{s,\text{RKL}}$, denoted by $|\lambda_{k,\text{RKL}}(s)|$, are plotted as a function of k for various values of s . It can be observed that for small values of s , the maximum eigenvalue (which is essentially the spectral radius) of $\mathbf{R}_{s,\text{RKL}}$ is associated with the longest-wavelength mode or the $k=1$ mode. However, as s increases beyond a certain threshold, the maximum eigenvalue shifts to the first few k -modes, and this phenomenon repeats itself due to the highly oscillatory nature of the Legendre polynomials at large s . Therefore, it can be inferred that the first few k -modes determine the maximum eigenvalue of the $\mathbf{R}_{s,\text{RKL}}$ operator for different values of s .

In the top-left panel of Figure 16, we plot the variation of $|\lambda_{k,\text{RKL}}(s)|$ as a function of s for the first seven eigenmodes of $\mathbf{R}_{s,\text{RKL}}$. It is apparent that the maximum eigenvalue is not solely determined by a single eigenmode. Rather, different eigenmodes dominate at different values of s , and this behavior oscillates. Additionally, we find that the first six modes, i.e., $1 \leq k \leq 6$, are sufficient for calculating the maximum eigenvalue for 1D problems. Therefore, the spectral radius does not exhibit a monotonically decreasing trend with increasing s , as evidenced by the bottom-left panel of Figure 16. Nevertheless, $\rho(\mathbf{R}_{s,\text{RKL}})$ approaches zero asymptotically as $s \rightarrow \infty$.

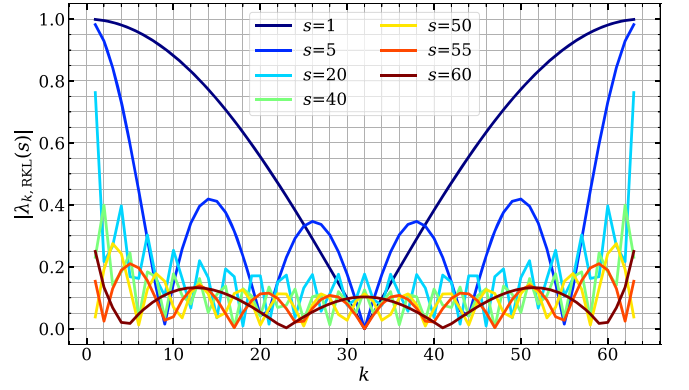


Figure 15. The absolute values of the eigenvalues ($\lambda_{k,\text{RKL}}(s)$) of the RKL iteration matrix ($\mathbf{R}_{s,\text{RKL}}$) as a function of k for different values of s .

In order to reduce the initial error to any value, a single s -stage RKL iteration is required by selecting a suitably high value that results in a spectral radius close to zero. The top-right panel of Figure 16 demonstrates this, displaying the minimum number of τ steps, m_{RKL} (Equation (A26)), needed to decrease the initial error by a factor of two. It is evident that m_{RKL} decreases with an increase in s and exhibits an oscillatory behavior similar to the spectral radius, which approaches 1 as $s \rightarrow \infty$. While it may seem that increasing s indefinitely would result in faster convergence, it is important to note that each τ step in an s -stage RKL iteration consists of s substages. As a result, if a problem achieves a certain tolerance level in m_{RKL} τ -steps, it actually requires $m_{\text{eff}} = s \times m_{\text{RKL}}$ single-loop iterations. Therefore, if the rate of decrease in m_{RKL} is slower than the rate of increase in s , then after a certain value of s , m_{eff} begins to increase as s increases. As a result, there exists an optimal value of s (s_{opt}) at which the RKL iteration converges the fastest. This is depicted in the bottom-right panel of Figure 16, where it is clear that m_{eff} initially decreases with increasing s , but begins to oscillate after $s = 35$. The value of m_{eff} reaches its global minimum value at $s = 56$.

Ideally, the theoretical value of s_{opt} can be found by the following equation:

$$\begin{aligned} \left. \frac{dm_{\text{eff}}}{ds} \right|_{s=s_{\text{opt}}} &= 0, \\ \left. \frac{d}{ds} \left(\frac{s \times d}{\log_{10}[\rho(\mathbf{R}_{s,\text{RKL}})]} \right) \right|_{s=s_{\text{opt}}} &= 0. \end{aligned} \quad (\text{A35})$$

Finding an analytical expression for s_{opt} from Equation (A35) is highly challenging in the absence of a closed-form expression for $\rho(\mathbf{R}_{s,\text{RKL}})$. Nevertheless, s_{opt} can be computed numerically for a given problem prior to commencing the iterative process. As can be seen from the bottom panels of Figure 16, the optimal value of s is where the value of $\rho(\mathbf{R}_{s,\text{RKL}})$ is at its minimum within the range $1 \leq s \leq N$. Additionally, the top-left panel of Figure 16 demonstrates that the first six k -modes determine the spectral radius of the $\mathbf{R}_{s,\text{RKL}}$ operator. Thus, we find the value of s_{opt} in the following two steps:

1. First we compute the spectral radius $\rho(\mathbf{R}_{s,\text{RKL}})$ as a function of s in the range $1 \leq s \leq N$ using Equation (A34) where k is restricted to $1 \leq k \leq 6$.

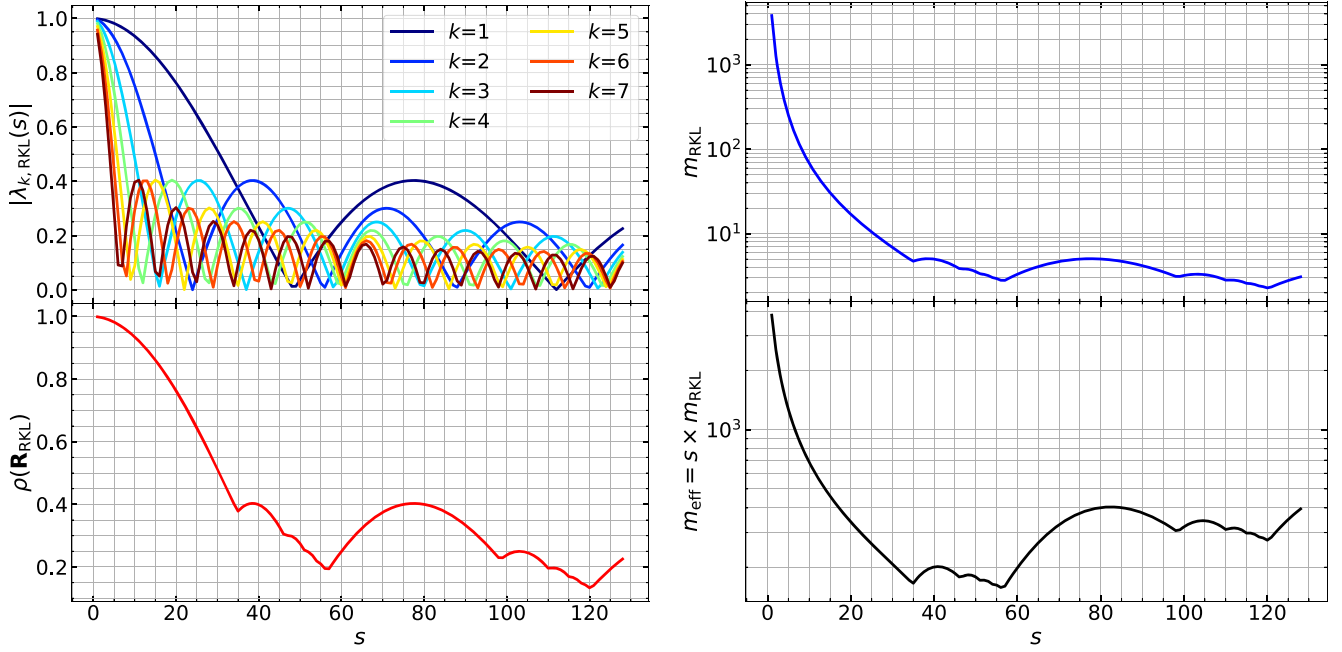


Figure 16. Top left: $|\lambda_{k,\text{RKL}}(s)|$ as a function of s for the first seven eigenmodes. Bottom left: the spectral radius ($\rho(\mathbf{R}_{s,\text{RKL}})$) of the $\mathbf{R}_{s,\text{RKL}}$ operator. Top right: minimum number of τ steps required to reduce the initial error by an order of two in an s -stage RKL scheme as a function of s . Bottom right: total number of single-loop iterations ($m_{\text{eff}} = s \times m_{\text{RKL}}$) required for the same convergence as a function of s .

- Then we find for what value s the above computed spectral radius has the minimum value. This will give s_{opt} .

The process described above is a general approach for determining the value of s_{opt} for a 1D problem. However, it can be further optimized by limiting the range of s used in the computation of the spectral radius in step 1. As can be observed from the top-left panel of Figure 16, the first few values of s are redundant. We know that s_{opt} lies between the first zero of $\lambda_{1,\text{RKL}}(s)$ and N . Thus, restricting the s -range to $[\text{first zero of } \lambda_{1,\text{RKL}}(s) \leq s \leq N]$ will reduce the number of operations required to calculate the value of s_{opt} . The procedure for identifying the first zero of $\lambda_{k,\text{RKL}}(s)$ will be outlined in Appendix A.4.

A.4. An Accelerated RKL Scheme

From the top-left panel of Figure 16, we notice that each Fourier mode or k -mode of the error has a unique optimal value of s denoted by $s_{k,\text{opt}}$. At $s_{k,\text{opt}}$, the k th eigenvalue or $\lambda_{k,\text{RKL}}(s)$ has a value near zero, and this occurs at the first zero of $\lambda_{k,\text{RKL}}(s)$. Therefore, instead of selecting a universal s for the entire spectrum, we can choose $s_{k,\text{opt}}$ for each k -mode and decrease the corresponding error individually. The approach is to commence the τ iteration with $s = s_{k,\text{opt}}$, where we choose $k = N/2$ at the initial iteration. Subsequently, after each τ -step, decrease k by one, i.e., $k = k_{\text{prev}} - 1$, and use the corresponding $s_{k,\text{opt}}$ value. This progression continues until we reach $k = 1$. If we still have not achieved the specified tolerance, we use $s = s_{1,\text{opt}}$ for the remaining iterations until the desired tolerance level is reached. This is because, during these accelerated iterations, starting from $k = N/2$ to $k = 1$, all the short-wavelength modes of the error are reduced below the tolerance level. Consequently, the remaining residual corresponds to the longest wavelength ($k = 1$) of the error. Therefore, we use $s_{1,\text{opt}}$ for the rest of the iterations, which yields the fastest

convergence. We show the simplified algorithm in the following:

```

=====
k=N/2
while (res>tol)
  s=s_opt(k)
  Iterate(s)
  k=k-1
  if k<1
    k=1
=====

```

To determine $s_{k,\text{opt}}$ for a specific k -mode, we can numerically compute the first zero of $P_s(\cos[k\pi/N])$ using Equation (A32). However, we can also use the asymptotic form of the Legendre polynomials to obtain the value of $s_{k,\text{opt}}$ numerically. For $-1 + \delta < \cos \theta < 1$ (with δ a small positive fixed number), the asymptotic form of $P_s(\cos \theta)$ is given by Temme (2014),

$$P_s(\cos \theta) \sim \sqrt{\frac{\theta}{\sin \theta}} J_0\left(\left[s + \frac{1}{2}\right]\theta\right) + \mathcal{O}(s^{-1}), \quad \theta \in (0, \pi), \quad (\text{A36})$$

where $J_0(x)$ is the zeroth Bessel function of the first kind. Thus, from Equation (A32) the eigenvalues of $\mathbf{R}_{s,\text{RKL}}$ can be approximated as

$$\lambda_{k,\text{RKL}}^{\text{apprx}}(s) = \sqrt{\frac{\theta}{\sin \theta}} J_0\left(\left[s + \frac{1}{2}\right]\theta\right), \quad \text{where } \theta = \frac{k\pi}{N}. \quad (\text{A37})$$

Although the validity of Equation (A37) is limited to large values of s , we observe in Figure 17 that it agrees well with the analytical values for $1 \leq k \leq N/2$. The top panel of Figure 17 displays the eigenvalues ($\lambda_{k,\text{RKL}}(s)$) for $k = 1$ (red), 5 (cyan), and 32 (yellow) as a function of s , represented by solid lines. The dotted lines in the same panel correspond to the

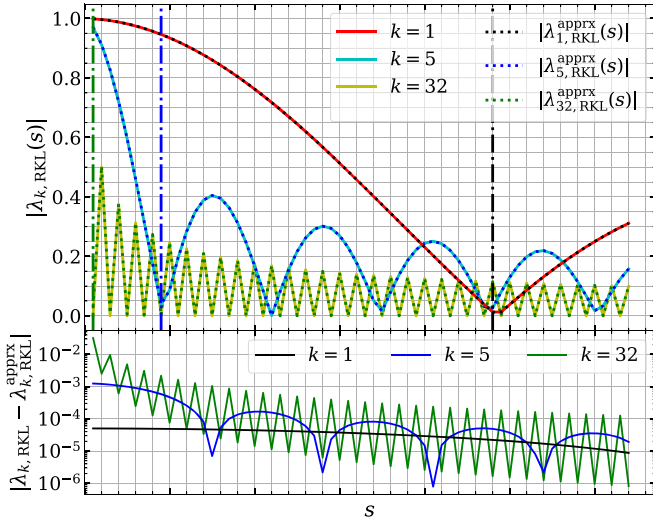


Figure 17. Top: the analytic (solid) and approximated (dotted) values of $\lambda_{k, \text{RKL}}(s)$ as a function of s for $k = 1, 5,$ and 32 . To calculate the approximation we use Equation (A37). The dashed-dotted lines represent the value of $s_{k, \text{opt}}$ (Equation (A38)) for $k = 1$ (red), 5 (blue), and 32 (green), i.e. the value of s where the first zero of the corresponding $\lambda_{k, \text{RKL}}(s)$ occurs. Bottom: the error in the approximation, i.e., $|\lambda_{k, \text{RKL}}(s) - \lambda_{k, \text{RKL}}^{\text{approx}}(s)|$ as a function of s for $k = 1$ (black), 5 (blue), and 32 (green). We notice that the error in the approximation (Equation (A37)) is $\lesssim 1\%$ even for $k = 32$.

approximation ($\lambda_{k, \text{RKL}}^{\text{approx}}(s)$) for the respective k -modes. The bottom panel shows the difference between $\lambda_{k, \text{RKL}}(s)$ and $\lambda_{k, \text{RKL}}^{\text{approx}}(s)$ for $k = 1$ (black), 5 (blue), and 32 (green).

For a given k value, θ is constant in Equation (A37). Therefore, the first zero of $\lambda_{k, \text{RKL}}^{\text{approx}}(s)$ corresponds to the first zero of $J_0(x)$. It is known that the first zero of $J_0(x)$ occurs at $x \approx 2.4048$. Hence, we can obtain the optimal value of s for the k th mode as follows:

$$\left(s_{k, \text{opt}} + \frac{1}{2}\right) \frac{k\pi}{N} = 2.4048, \quad (A38)$$

$$s_{k, \text{opt}} = \text{Round}\left(\frac{2.4048}{\pi} \left[\frac{N}{k}\right] - \frac{1}{2}\right),$$

where $\text{Round}(p)$ is defined as the nearest integer value of the real number p . In the top panel of Figure 17, we show $s_{k, \text{opt}}$ (horizontal dashed-dotted line) for $k = 1$ (black), 5 (blue), and 32 (green). We see that Equation (A38) represents the first zero of the corresponding $\lambda_{k, \text{RKL}}(s)$ very well.

Once we have obtained the optimal value of s ($s_{k, \text{opt}}$) for each k using Equation (A38), we can use it to perform the accelerated RKL iterations discussed earlier in this section. Figure 18 displays the optimal value of s as a function of the wavenumber k . We observe that $s_{k, \text{opt}}$ is symmetric about $k = N/2$ as expected. Additionally, a single $s_{k, \text{opt}}$ value corresponds to multiple wavenumbers, particularly for high k -modes. Consequently, one τ -iteration with such an $s_{k, \text{opt}}$ value can simultaneously reduce the error for the corresponding k -values. Thus, there is no need to iterate the solution for each k -mode separately in the accelerated scheme discussed above. If $s_{k, \text{opt}} = s_{k+1, \text{opt}}$ for some k , we can skip the iteration for that k and proceed to the next k value. An example of the algorithm is presented below:

```

k=N/2
s_old=0

```

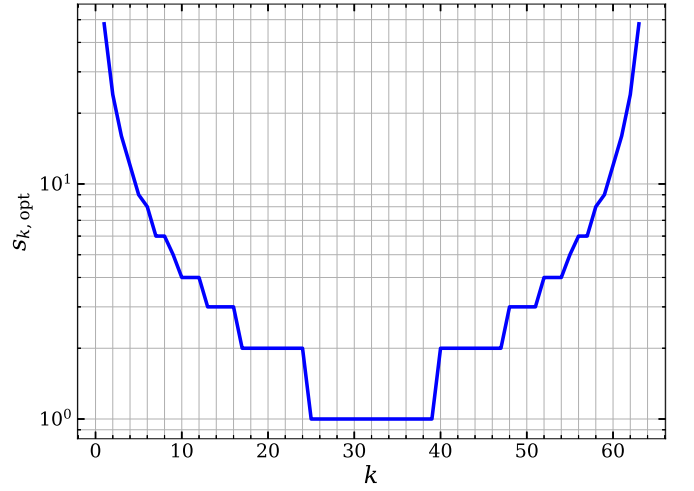


Figure 18. $s_{k, \text{opt}}$ as a function of wavenumber (k).

(Continued)

```

while (res>tol)
s=s_opt(k)
if s==s_old and k!=1
pass
else
Iterate(s)
s_old=s
k=k-1
if k<1
k=1

```

Appendix B

Performance Test for Different Iterative Solvers

Within this section, we present the comparative performance of the G-S, SOR, and RKL methods (without the application of multigrid acceleration) in solving the 3D problem discussed in Section 4.1. The optimal overrelaxation parameter (ω) in the SOR algorithm is obtained by the following expression (Press et al. 1992):

$$\omega = \frac{2}{1 + \sqrt{1 - \rho_J^2}}, \quad (B1)$$

where ρ_J is the spectral radius of the Jacobi iteration matrix. For a 3D grid of size $J \times L \times K$ with grid spacing ($\Delta_x, \Delta_y, \Delta_z$) and Dirichlet boundary conditions, ρ_J is given by

$$\rho_J = 2\Delta_t \left[\frac{1}{\Delta_x^2} \cos\left(\frac{\pi}{J}\right) + \frac{1}{\Delta_y^2} \cos\left(\frac{\pi}{L}\right) + \frac{1}{\Delta_z^2} \cos\left(\frac{\pi}{K}\right) \right], \quad (B2)$$

where

$$\Delta_t = 2 \left[\frac{1}{\Delta_x^2} + \frac{1}{\Delta_y^2} + \frac{1}{\Delta_z^2} \right]^{-1}. \quad (B3)$$

We consider a box with 128^3 zones for the calculation, which was distributed between 128 CPUs. The CPUs used were Intel(R) Xeon(R) Gold 6326, whose base clock frequency is 2.90 GHz. The parallel communications between

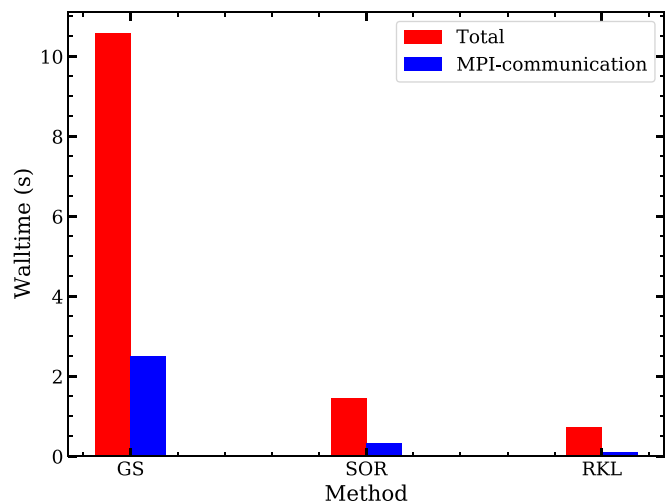


Figure 19. Performance test for different iterative solvers (excluding multigrid acceleration). The red and blue bars correspond to the time taken for the whole execution and MPI communications, respectively.

the CPUs were performed with the message passing interface (MPI) using the publicly available Open MPI v4.1.5 library with the Infiniband EDR interconnect.

Figure 19 displays the total time taken by various solvers to bring the maximum residual error below 10^{-10} , as indicated by the red bars. The corresponding time spent in MPI communications throughout the entire execution is also illustrated by the blue bars. The results clearly demonstrate that, in terms of the total execution time, the RKL method outperforms the SOR and G-S techniques by a factor of roughly 2 and 15, respectively. As previously mentioned, the communication overheads for the SOR and G-S solvers are considerably higher, comprising approximately 23% of the total execution time. On the other hand, the time spent during parallel communications in the RKL scheme amounts to only around 13% of the total time.

ORCID iDs

Ankush Mandal <https://orcid.org/0000-0002-0912-8081>

Dipanjan Mukherjee <https://orcid.org/0000-0003-0632-1000>

Andrea Mignone <https://orcid.org/0000-0002-8352-6635>

References

- Almgren, A. S., Beckner, V. E., Bell, J. B., et al. 2010, *ApJ*, 715, 1221
- Bate, M. R., & Burkert, A. 1997, *MNRAS*, 288, 1060
- Bertschinger, E. 1998, *ARA&A*, 36, 599
- Binney, J., & Tremaine, S. 1987, *Galactic Dynamics* (Princeton, NJ: Princeton Univ. Press)
- Boss, A. P. 1997, *Sci*, 276, 1836
- Boss, A. P., & Bodenheimer, P. 1979, *ApJ*, 234, 289
- Boss, A. P., & Myhill, E. A. 1992, *ApJS*, 83, 311
- Brandt, A. 1977, *MaCom*, 31, 333
- Briggs, W. L., Henson, V. E., & McCormick, S. F. 2000, *A Multigrid Tutorial* (2nd ed.; Philadelphia, PA: SIAM)
- Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, *CoPhC*, 89, 149
- Burkert, A., & Bodenheimer, P. 1993, *MNRAS*, 264, 798
- Couch, S. M., Graziani, C., & Flocke, N. 2013, *ApJ*, 778, 181
- Evrard, A. E. 1988, *MNRAS*, 235, 911
- Falco, M., Hansen, S. H., Wojtak, R., & Mamon, G. A. 2013, *MNRAS*, 431, L6
- Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, 131, 273
- Grudić, M. Y. 2021, *MNRAS*, 507, 1064
- Guillet, T., & Teyssier, R. 2011, *JCoPh*, 230, 4756
- Hanawa, T. 2019, *JPhCS*, 1225, 012015
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, 585, 357
- Hernquist, L., & Katz, N. 1989, *ApJS*, 70, 419
- Hopkins, P. F. 2015, *MNRAS*, 450, 53
- Hubber, D. A., Goodwin, S. P., & Whitworth, A. P. 2006, *A&A*, 450, 881
- Hubber, D. A., Rosotti, G. P., & Booth, R. A. 2018, *MNRAS*, 473, 1603
- Hunter, J. D. 2007, *CSE*, 9, 90
- Jiang, Y.-F., Belyaev, M., Goodman, J., & Stone, J. M. 2013, *NewA*, 19, 48
- Katz, M. P., Zingale, M., Calder, A. C., et al. 2016, *ApJ*, 819, 94
- Kiessling, M. K. H. 1999, arXiv:astro-ph/9910247
- Klein, R. I. 1999, *JCoAM*, 109, 123
- Mandal, A., Mukherjee, D., Federrath, C., et al. 2021, *MNRAS*, 508, 4738
- McKee, C. F., & Ostriker, E. C. 2007, *ARA&A*, 45, 565
- Meyer, C. D., Balsara, D. S., & Aslam, T. D. 2012, *MNRAS*, 422, 2102
- Meyer, C. D., Balsara, D. S., & Aslam, T. D. 2014, *JCoPh*, 257, 594
- Mignone, A., Bodo, G., Massaglia, S., et al. 2007, *ApJS*, 170, 228
- Mignone, A., Zanni, C., Tzeferacos, P., et al. 2012, *ApJS*, 198, 7
- Mo, H., van den Bosch, F. C., & White, S. 2010, *Galaxy Formation and Evolution* (1st ed.; Cambridge: Cambridge University Press)
- Mullen, P. D., Hanawa, T., & Gammie, C. F. 2021, *ApJS*, 252, 30
- Nordhaus, J., Burrows, A., Almgren, A., & Bell, J. 2010, *ApJ*, 720, 694
- Norman, M. L., Wilson, J. R., & Barton, R. T. 1980, *ApJ*, 239, 968
- Ostriker, E. C., Stone, J. M., & Gammie, C. F. 2001, *ApJ*, 546, 980
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical Recipes in C: The Art of Scientific Computing* (2nd ed.; Cambridge: Cambridge Univ. Press)
- Rice, W. K. M., Lodato, G., & Armitage, P. J. 2005, *MNRAS*, 364, L56
- Ricker, P. M. 2008, *ApJS*, 176, 293
- Springel, V. 2005, *MNRAS*, 364, 1105
- Springel, V. 2010, *MNRAS*, 401, 791
- Springel, V., Yoshida, N., & White, S. D. M. 2001, *NewA*, 6, 79
- Steinmetz, M., & Mueller, E. 1993, *A&A*, 268, 391
- Temme, N. M. 2014, *Asymptotic Methods for Integrals* (Singapore: World Scientific)
- Trottenberg, U., Oosterlee, C., & Schuller, A. 2000, *Multigrid* (1st ed.; New York: Academic)
- Truelove, J. K., Klein, R. I., McKee, C. F., et al. 1997, *ApJL*, 489, L179
- Truelove, J. K., Klein, R. I., McKee, C. F., et al. 1998, *ApJ*, 495, 821
- Vaidya, B., Prasad, D., Mignone, A., Sharma, P., & Rickler, L. 2017, *MNRAS*, 472, 3147
- van der Velden, E. 2020, *JOSS*, 5, 2004
- Wadsley, J. W., Stadel, J., & Quinn, T. 2004, *NewA*, 9, 137
- Wang, H.-H., & Yen, C.-C. 2020, *ApJS*, 247, 2
- Ziegler, U. 2005, *A&A*, 435, 385