Parallelized Solution Method of the Three-dimensional Gravitational Potential on the Yin–Yang Grid

Marius Almanstötter1,2, Tobias Melson2, Hans-Thomas Janka2, and Ewald Müller2

1 Institut für Mathematik, Universität Augsburg, D-86135 Augsburg, Germany
2 Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str. 1, D-85748 Garching, Germany

Received 2018 June 12; revised 2018 July 10; accepted 2018 July 10; published 2018 August 20

Abstract

We present a new method for solving the 3D gravitational potential of a density field on the Yin–Yang grid. Our algorithm is based on a multipole decomposition and is completely symmetric with respect to the two Yin–Yang grid patches. It is particularly efficient on distributed-memory machines with a large number of compute tasks, because the amount of data being explicitly communicated is minimized. All operations are performed on the original grid without the need for interpolating data onto an auxiliary spherical mesh.

Key words: gravitation – methods: numerical – stars: general – supernovae: general

1. Introduction

Solving for Newtonian gravity in 3D is relevant in a large number of astrophysical and geophysical problems. For instance, the propagation of seismic waves on Earth (Komatitsch & Tromp 2002), the formation of planetesimals in protoplanetary disks (Simon et al. 2016), shock propagation in protostellar clouds (Falle et al. 2017), and the formation of the Earth’s core (Mondal & Korenaga 2018) are 3D situations that require self-gravity to be taken into account. One example, which is in the focus of our interest, are core-collapse supernovae.

The explosion mechanism of core-collapse supernovae is one of the long-standing riddles in stellar astrophysics. Thanks to growing supercomputing power, 3D neutrino-hydrodynamics simulations can now be performed to study the physical processes responsible for the onset of the explosion with highly optimized codes on distributed-memory architectures (Takiwaki et al. 2012, 2014; Lentz et al. 2015; Melson et al. 2015a, 2015b; Roberts et al. 2016; Müller et al. 2017; Ott et al. 2018; Summa et al. 2018). The Garching group uses the PROMETHEUS-VERTEX package (Rampp & Janka 2002), which extends the finite-volume hydrodynamics module PROMETHEUS (Fryxell et al. 1989) with a state-of-the-art neutrino transport and interaction treatment. Its latest code version applies the Yin–Yang grid (Kageyama & Sato 2004)—a composite spherical mesh—to discretize the spatial domain.

Until now, self-gravity of the stellar plasma in 3D simulations has been treated in spherical symmetry on an averaged density profile in PROMETHEUS, because the gravitational potential is dominated by the spherical protoneutron star in the center. As stellar collapse to neutron stars is only a mildly relativistic problem, the Garching group applies Newtonian hydrodynamics but uses a correction of the monopole of the gravitational potential (Marek et al. 2006), which has been turned out to yield results that are well compatible with fully relativistic calculations (Liebendörfer et al. 2005; Müller et al. 2010). However, as 3D core-collapse supernova simulations become more elaborate by taking more and more physical aspects into account, it is highly desirable to include a realistic 3D gravitational potential to treat large-scale asymmetries correctly. This holds true, in particular, in cases where the collapsing star develops a global deformation, e.g., due to centrifugal effects in the case of rapid rotation.

Müller & Chan (2018) recently presented a method for solving Poisson’s equation on spherical polar grids using 3D Fast Fourier transforms. Although their algorithm yields an accurate solution of the gravitational potential even for highly aspherical density configurations, it is currently not available for the Yin–Yang grid, and its parallel efficiency needs to be improved to serve our purposes.

In this work, we present a method for efficiently computing the gravitational potential on the Yin–Yang grid based on the gravity solver of Müller & Steinmetz (1995). Also, Wong-wathanarat et al. (2010) applied a 3D gravity solver on Yin–Yang data; however, their code was not parallelized for distributed-memory systems. It relied on mapping the data to an auxiliary spherical polar grid. With our approach presented here, we compute the gravitational potential on the Yin–Yang grid directly.

In Section 2, we will briefly summarize the algorithm developed by Müller & Steinmetz (1995) for 3D spherical grids. A method for its efficient parallelization will be discussed in Section 3, followed by a detailed explanation of the modifications for the Yin–Yang grid in Section 4. Test calculations will be presented in Section 5.

2. Solving Poisson’s Equation on a Spherical Grid

In this section, we briefly summarize the procedure for computing the 3D gravitational potential on a spherical polar grid as shown by Müller & Steinmetz (1995). Given the density distribution, \( \rho(\mathbf{r}) \), the gravitational potential at the location \( \mathbf{r} \) is determined by solving Poisson’s equation, which can be expressed in its integral form as

\[
\Phi(\mathbf{r}) = -G \int_V \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}',
\]

where \( G \) is the gravitational constant, and \( V \) denotes the whole computational domain. In the following, we employ spherical coordinates and express spatial vectors as \( \mathbf{r} = (r, \theta, \phi) \). A decomposition of \(|\mathbf{r} - \mathbf{r}'|^{-1} \) into spherical harmonics \( Y_{lm} \) yields

\[
|\mathbf{r} - \mathbf{r}'|^{-1} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} Y_{lm}(\theta, \phi) Y_{lm}(\theta', \phi')^* \]

with the spherical harmonics of degree \( l \) and order \( m \) defined as

\[
Y_{lm}(\theta, \phi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-l}^{l} (-1)^m \sqrt{(2l+1)(l-m)! \frac{(l+m)!}{(l-m)!}} \times (\cos \theta)^l \times (\exp(i m \phi))^m
\]

and their complex conjugates.

The gravitational potential at a point \( \mathbf{r} \) can then be expressed as

\[
\Phi(\mathbf{r}) = -G \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{(2l+1)(l-m)! \frac{(l+m)!}{(l-m)!}} \int_V \rho(\mathbf{r}') Y_{lm}(\theta', \phi')^* (\cos \theta)^l \exp(-i m \phi') d^3\mathbf{r}'
\]

Thus, the gravitational potential is determined by solving Poisson’s equation on the Yin–Yang grid. Our approach is an efficient generalization of the method of Müller & Steinmetz (1995) to 3D Yin–Yang grids.
(Müller & Steinmetz 1995, Equation (8))

\[ \Phi(r) = -G \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4 \pi}{2 \ell + 1} Y_{\ell m}(\vartheta, \varphi) \times \left( \frac{1}{r_{\ell+1}} C_{\ell m}(r) + r^\ell D_{\ell m}(r) \right), \]

(2)

In the latter equation, \(C_{\ell m}(r)\) and \(D_{\ell m}(r)\) are defined as

\[ C_{\ell m}(r) := \int_0^{2\pi} \int_0^{\pi} d\varphi' d\vartheta' \sin \vartheta' Y_{\ell m}(\vartheta', \varphi') \times \int_0^{r_{\ell+1}} dr' (r')^{\ell+2} \varphi(r'), \]

(3)

\[ D_{\ell m}(r) := \int_0^{2\pi} \int_0^{\pi} d\varphi' d\vartheta' \sin \vartheta' Y_{\ell m}(\vartheta', \varphi') \times \int_r^{\infty} dr' (r')^{\ell+1} \varphi(r'), \]

(4)

where \(Y_{\ell m}(\vartheta', \varphi')\) are the complex conjugates of the spherical harmonics. After inserting their definition and rearranging the terms, Müller & Steinmetz (1995) wrote the gravitational potential as a sum of two contributions: one for the gravitational potential inside a sphere of radius \(r\), \(\Phi_{\text{in}}(r)\), and a second for the potential outside, \(\Phi_{\text{out}}(r)\),

\[ \Phi(r) = -G \sum_{\ell=0}^{\infty} \sum_{m=0}^{\ell} N^{(\ell m)} P_{\ell m}(\cos \vartheta) [\Phi_{\text{in}}^{(\ell m)}(r) + \Phi_{\text{out}}^{(\ell m)}(r)], \]

(5)

where \(P_{\ell m}\) are the associated Legendre polynomials. The integrals for these inner and outer contributions can be expressed as

\[ \Phi_{\text{in}}^{(\ell m)}(r) = \frac{1}{r_{\ell+1}} \int_0^{2\pi} \int_0^{\pi} d\varphi' d\vartheta' \sin \vartheta' P_{\ell m}(\cos \vartheta') \times \cos(m(\varphi - \varphi')) \int_0^{r_{\ell+1}} dr' (r')^{\ell+2} \varphi(r'), \]

(6)

and

\[ \Phi_{\text{out}}^{(\ell m)}(r) = r^\ell \int_0^{2\pi} \int_0^{\pi} d\varphi' d\vartheta' \sin \vartheta' P_{\ell m}(\cos \vartheta') \times \cos(m(\varphi - \varphi')) \int_r^{\infty} dr' (r')^{\ell+1} \varphi(r'). \]

(7)

The normalization factor in Equation (5) is given by

\[ N^{(\ell m)} := \frac{(\ell - m)!}{(\ell + m)!} 2^m \delta^{(m)}, \]

(8)

with

\[ \delta^{(m)} := \begin{cases} 2, & \text{if } m = 0, \\ 1, & \text{if } m > 0. \end{cases} \]

(9)

To discretize these equations, we divide the spatial domain into \(n_R \times n_\vartheta \times n_\varphi\) grid cells, each spanning from \((r^-_i, \vartheta^-_j, \varphi^-_k)\) to \((r^+_i, \vartheta^+_j, \varphi^+_k)\) with \(i = 1, \ldots, n_R\), \(j = 1, \ldots, n_\vartheta\), and \(k = 1, \ldots, n_\varphi\). The finite-volume method as being used in the PROMETHEUS code assumes that in each cell, the density is given as a cell average, i.e.,

\[ \varrho(r) = \varrho_{ijk}, \]

(10)

for \(r^- \leq r \leq r^+_i\), \(\vartheta^-_j \leq \vartheta \leq \vartheta^+_j\), and \(\varphi^-_k \leq \varphi \leq \varphi^+_k\). This assumption allows for simplifying Equations (6) and (7), which can then be written as

\[ \Phi_{\text{in}}^{(\ell m)}(r) = \frac{1}{r_{\ell+1}} \sum_{i=1}^{n_R} \sum_{j=1}^{n_\vartheta} \sum_{k=1}^{n_\varphi} \varrho_{ijk} R_{\ell m,i}^{(\ell)} T_{\ell m,i}^{(\ell)} F_k^{(m)}(\varphi), \]

(11)

\[ \Phi_{\text{out}}^{(\ell m)}(r) = r^\ell \sum_{i=n_R+1}^{n_R} \sum_{j=1}^{n_\vartheta} \sum_{k=1}^{n_\varphi} \varrho_{ijk} R_{\ell m,i}^{(\ell)} T_{\ell m,i}^{(\ell)} F_k^{(m)}(\varphi), \]

(12)

with

\[ F_k^{(m)}(\varphi) := \cos(m\varphi)C_k^{(m)} + \sin(m\varphi)S_k^{(m)}. \]

(13)

In our chosen coordinates, the gravitational potential is computed at the cell interfaces. The radial index, \(n_R\), introduced above is equal to the cell index, \(i\), if \(r = r^-_i\). The two integrals \(C_k^{(m)}\) and \(S_k^{(m)}\) can be evaluated analytically:

\[ C_k^{(m)} := \int_{r^-_i}^{r^+_i} d\varrho' \cos(m\varphi') \]

\[ = \begin{cases} \frac{r^+_i - r^-_i}{m}[\sin(m\varphi^+_i) - \sin(m\varphi^-_i)], & \text{if } m > 0, \\ \frac{r^+_i - r^-_i}{m}[\sin(m\varphi^-_i) - \sin(m\varphi^+_i)], & \text{if } m < 0, \\ \frac{1}{m}[\sin(m\varphi^-_i) - \sin(m\varphi^+_i)], & \text{if } m = 0, \end{cases} \]

(14)

and

\[ S_k^{(m)} := \int_{r^-_i}^{r^+_i} d\varrho' \sin(m\varphi') \]

\[ = \begin{cases} 0, & \text{if } m = 0, \\ \frac{1}{m}[\cos(m\varphi^-_i) - \cos(m\varphi^+_i)], & \text{if } m > 0, \\ \frac{1}{m}[\cos(m\varphi^-_i) - \cos(m\varphi^+_i)], & \text{if } m < 0. \end{cases} \]

(15)

The radial integrals in Equations (11) and (12) can also be computed directly:

\[ R_{\text{in},\ell,i}^{(\ell)} := \int_{r^-_i}^{r^+_i} dr' (r')^{\ell+2} = \frac{1}{\ell + 3}((r^+_i)^{\ell+3} - (r^-_i)^{\ell+3}), \]

(16)

\[ R_{\text{out},\ell}^{(\ell)} := \int_{r^-_i}^{r^+_i} dr' (r')^{\ell+1} \]

\[ = \begin{cases} \ln(r^-_i) - \ln(r^-_i), & \text{if } \ell = 2, \\ \frac{1}{2-\ell}((r^+_i)^{2-\ell} - (r^-_i)^{2-\ell}), & \text{if } \ell \neq 2. \end{cases} \]

(17)

The remaining integrals in Equations (11) and (12),

\[ T_{\ell m,i}^{(\ell)} := \int_{\vartheta^-_j}^{\vartheta^+_j} d\vartheta' \sin(m\varphi')P_{\ell m}(\cos \vartheta'), \]

(18)

can be evaluated efficiently and analytically, i.e., without numerical integration errors, using recurrence relations (see the Appendix).

Solving Equation (5) numerically requires setting an upper bound for the summation over \(\ell\), which we denote as \(\ell_{\text{max}}\). We will discuss the choice of \(\ell_{\text{max}}\) below in Section 6.

### 3. Parallelization of the Method

For efficiently computing the gravitational potential on distributed-memory systems, it is important to minimize the amount of data being explicitly exchanged between compute tasks. The easiest parallel solution of the aforementioned equations would be to collect the entire density field from all computing units, calculate the gravitational potential in serial, and send the result back to all tasks. This, however, would require a large amount of data being communicated very
inefficiently. More precisely, roughly $n_R \times n_\phi \times n_\psi \times 2$ floating-point numbers would be sent serially, where the factor of two accounts for the collection of the data and their subsequent redistribution. The serial calculation of the gravitational potential would cause an additional load imbalance between the tasks. Especially if the number of grid cells is very large, this strategy is un.rewarding.

Here, we present an approach for solving the gravitational potential accurately on a spherical grid, whose angular domain is decomposed among a large number of tasks, while the full potential accurately on a spherical grid, whose angular domain

$$
\Phi^{(\ell m)}(r) = \frac{1}{r^{\ell+1}} \sum_{i=1}^{n_i} R^{(\ell)}_{\text{in},i}(\cos(m\phi)A^{(\ell m)}_{C,i} + \sin(m\phi)A^{(\ell m)}_{S,i}),
$$

(19)

$$
\Phi^{(\ell m)}(r) = r^\ell \sum_{i=1}^{n_i} R^{(\ell)}_{\text{out},i}(\cos(m\phi)A^{(\ell m)}_{C,i} + \sin(m\phi)A^{(\ell m)}_{S,i}),
$$

(20)

we immediately see that the summation over the angular domain is separated from the remaining calculation. After each compute task has solved Equations (21) and (22) locally, a global summation of $A^{(\ell m)}_{C,i}$ and $A^{(\ell m)}_{S,i}$ is performed. This is the only step, where explicit data communication is necessary. Parallel computing libraries provide efficient built-in methods to compute the sum across $n_{\text{tasks}}$ compute tasks. In case of the commonly used Message Passing Interface (MPI), such a function is $\text{MPI}$.\text{Allreduce}$

Figure 1. Three-dimensional visualization of the Yin–Yang grid cells on a sphere of constant radius.

for data communication. To tackle this issue, the “double-double summation” algorithm described by He & Ding (2001) is applied, which treats each number as a pair of the value itself and its error being accumulated during floating-point operations. With this method, we achieve reproducibility of the results up to machine precision.

Generally, only the terms that explicitly depend on the density have to be calculated in each step of a time-dependent simulation. All other terms remain constant and therefore need to be computed only once, as long as the grid coordinates do not change with time.

### 4. Algorithm and its Parallelization for the Yin–Yang Grid

In its latest version, the PROMETHEUS-VERTEX code solves the hydrodynamics on the Yin–Yang grid (Kageyama & Sato 2004; Wongwathanarat et al. 2010), which is a combined mesh consisting of two low-latitude parts of a spherical polar grid. Its main advantage over a spherical polar grid covering the full $4\pi$ sphere is the exclusion of the grid singularities at the poles with their numerical difficulties. We show the geometry of the Yin–Yang grid in Figure 1.

Let the coordinates on Yin and Yang be denoted as $(r_{\text{Yin}}, \theta_{\text{Yin}}, \phi_{\text{Yin}})$ and $(r_{\text{Yang}}, \theta_{\text{Yang}}, \phi_{\text{Yang}})$, respectively, and defined in the intervals

$$0 \leq r \leq r_{\text{max}},$$

$$\frac{\pi}{4} - \delta_\theta \leq \theta \leq \frac{3\pi}{4} + \delta_\theta,$$

$$-\frac{3\pi}{4} - \delta_\phi \leq \phi \leq \frac{3\pi}{4} + \delta_\phi,$$

where $r_{\text{max}}$ is the outer radius of the grid. $\delta_\theta$ and $\delta_\phi$ are small values that control the width of the buffer zones in the Yin–Yang overlap region. Both grid patches consist of $n_R \times n_\theta \times n_\phi$ cells, and have equal coordinate axes. The angular coordinates of a certain point on Yin are transformed
into the Yang system according to

\[
\vartheta_{\text{Yang}} = \arccos(\sin \vartheta_{\text{Yin}} \sin \varphi_{\text{Yin}}),
\]

\[
\varphi_{\text{Yang}} = \arctan\left(-\frac{\cos \vartheta_{\text{Yin}}}{\sin \vartheta_{\text{Yin}} \cos \varphi_{\text{Yin}}} \right),
\]

where the inverse transformation is obtained by exchanging the subscripts.

The computation of the gravitational potential on the Yin–Yang grid is not trivial, because both grid patches are rotated against each other and are partially overlapping. Wongwathanarat et al. (2010) also used a version of PROMETHEUS with the Yin–Yang grid. They computed the gravitational potential by interpolating the density from the individual Yin–Yang grid parts onto a new “auxiliary” spherical polar grid that covers the entire domain. The algorithm by Müller & Steinmetz (1995) could then be applied without any modifications. After that, the gravitational potential was mapped back to the Yin–Yang grid patches. Although the orientation and grid resolution of the gravitational potential was mapped back to the Yin–Yang grid, we developed a new method based on computing the potential for a density distribution on the Yin grid. The strong scaling deviates from the theoretically expected scaling behavior. Note the logarithmic scales of both coordinate axes.

After evaluating the latter equations, the global sums are computed separately for both grid parts resulting in different values for \(A_{C,j,Yin/Yang}^{(l,m)}\), \(A_{S,j,Yin/Yang}^{(l,m)}\), \(A_{C,j,Yang/Yin}^{(l,m)}\), and \(A_{S,j,Yang/Yin}^{(l,m)}\). Four different potentials are calculated using Equations (19), (20), and (5), namely \(\Phi_{\text{Yin}}(r_{\text{Yin}})\), \(\Phi_{\text{Yin}}(r_{\text{Yang}})\), \(\Phi_{\text{Yang}}(r_{\text{Yang}})\), and \(\Phi_{\text{Yang}}(r_{\text{Yin}})\). Here, the subscript of \(\Phi\) denotes the grid part on which \(A_{C,j}^{(l,m)}\) and \(A_{S,j}^{(l,m)}\) are computed, while the subscript of \(r\) determines in which coordinate system the angles \(\vartheta\) and \(\varphi\) in Equations (5), (19), and (20) are expressed.

The final gravitational potential for all compute tasks on Yin and Yang is, respectively,

\[
\Phi(r_{\text{Yin}}) = \Phi_{\text{Yin}}(r_{\text{Yin}}) + \Phi_{\text{Yang}}(r_{\text{Yin}}),
\]

\[
\Phi(r_{\text{Yang}}) = \Phi_{\text{Yin}}(r_{\text{Yang}}) + \Phi_{\text{Yang}}(r_{\text{Yang}}).
\]

Hence, the contributions from both grid parts are transformed into the local coordinates of the task and added. This procedure is completely symmetric with respect to both Yin–Yang grid patches and does not introduce a preferred direction. An interpolation onto an auxiliary grid is not required thus avoiding further numerical errors.

In Figure 2, we present the strong scaling behavior of our implementation of the Yin–Yang gravity solver. The gravitational potential was computed for a density distribution on the Yin–Yang grid with an angular resolution of 1°. The dashed line indicates the theoretically perfect scaling behavior. Note the logarithmic scales of both coordinate axes.
code by Wongwathanarat et al. (2010) would result in a flat
curve in Figure 2. Our approach is thus suitable for large-scale
simulations involving thousands of processors.

5. Test Calculations

To validate our approach for solving the gravitational
potential on the Yin–Yang grid, we performed various tests.
First, the gravitational potential of a spherically symmetric
density distribution was verified to be equal to the corre-
sponding analytical solution up to machine precision. Second, an
axisymmetric density distribution was mapped to the Yin–
Yang grid with 1° angular resolution, and its gravitational
potential was compared with the result gained from an
axisymmetric gravity solver yielding accurate agreement of
$10^{-4}$. Third, the potential of a homogeneous ellipsoid was
compared to the semi-analytical solution. We will elaborate on
the latter test here in detail.

We constructed an ellipsoid of constant density, $\rho_0$, with
semi-axes $a$, $b$, and $c$ with $b = 1.5a$ and $c = 2a$. In Cartesian
coordinates, a point is inside the ellipsoid if

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 \leq 1.$$  

(30)

The gravitational potential of this configuration is given by
(Chandrasekhar 1969)

$$\Phi(r) = \pi G \rho_0 \, abc \, (A(r) \, x^2 + B(r) \, y^2 + C(r) \, z^2 - D(r)),$$

(31)

where $r = (x, y, z)$ and

$$A(r) := \int_{u_0(r)}^{\infty} du \, [(a^2 + u) \, (b^2 + u) \, (c^2 + u)]^{-1/2},$$

(32)

$$B(r) := \int_{u_0(r)}^{\infty} du \, [(a^2 + u) \, (b^2 + u) \, (c^2 + u)]^{-1/2},$$

(33)

$$C(r) := \int_{u_0(r)}^{\infty} du \, [(a^2 + u) \, (b^2 + u) \, (c^2 + u)]^{-1/2},$$

(34)

$$D(r) := \int_{u_0(r)}^{\infty} du \, [(a^2 + u) \, (b^2 + u) \, (c^2 + u)]^{-1/2}.$$  

(35)

If the point $r$ is located inside the ellipsoid, $u_0(r) = 0$. Otherwise,$u_0(r)$ is given as the positive root of the equation

$$\frac{x^2}{a^2 + u_0} + \frac{y^2}{b^2 + u_0} + \frac{z^2}{c^2 + u_0} = 1.$$  

(36)

We evaluated the root and the integrals numerically up to a
precision of $10^{-13}$.

The transformation from the Yin spherical polar coordinates
to Cartesian coordinates is given by

$$x = r \sin \vartheta \sin \varphi, \quad y = r \sin \vartheta \cos \varphi, \quad z = r \cos \vartheta,$$

(37)

(38)

(39)

where we have introduced a rotation angle $\Delta \varphi = \pi/8$ to break
the symmetry between both Yin–Yang grid patches. To compute the Cartesian coordinates for $\text{Yang}$, the transformation
given by Equations (23) and (24) has to be applied to express the $\text{Yang}$ coordinates in the $\text{Yin}$ system.

To estimate the correctness of our 3D gravity solver for the
case of the homogeneous ellipsoid, we compared the calculated
gravitational potential in the entire computational domain (i.e.,
inside and outside the ellipsoid) with the semi-analytical
solution. This was done on the Yin–Yang grid, but also on the
standard spherical polar grid covering the whole sphere. Both mesh configurations had an equidistant radial spacing and an
outer radius $r_{\text{max}} = c$. We varied both the grid resolution and
the maximum multipole order $\ell_{\text{max}}$. In the following, we denote
the models with $n_R = 400$ and an angular resolution of $2^\circ$ as
low resolution (LR). The runs with twice as many grid points in
all directions are referred to as high resolution (HR).

In Figure 3, we show the maximum relative error of the
calculated gravitational potential with respect to the semi-
analytical solution for the different mesh configurations and
grid resolutions. As expected, the relative errors of the HR
models are always smaller than in the LR cases, because the
gravitational potential is sampled more accurately in the former
runs and the surface of the ellipsoid is better resolved. While
for the LR case the results computed on the Yin–Yang grid and
on the spherical polar grid deviate from each other, both setups
yield basically identical solutions in the HR case. This
demonstrates that our adaptation of the gravity solver performs
correctly on the Yin–Yang grid.

With increasing maximum multipole order $\ell_{\text{max}}$, the relative
error decreases in all setups, as the shape of the ellipsoid is
approximated more accurately. Ideally, an infinite number of
multipole terms would be required to precisely reproduce an
ellipsoidal configuration. Including more higher-order terms
thus reduces the deviation from the semi-analytical solution.

The spatial distribution of the relative error on the Yin–Yang
grid is shown in Figure 4 for the HR case with $\ell_{\text{max}} = 80$ in the
plane $z = 0$. It can clearly be seen that the error is dominated
by the cells close to the surface of the ellipsoid. Their deviation
from the semi-analytical solution is two to three orders of
magnitude larger than for cells near the grid origin. Machine
precision is not reached in the center, because the gravitational
potential there is also dependent on the density distribution
outside (see Equation (5) and the term $\Phi_{\text{out}}$ there). As the
ellipsoidal surface is not sampled to machine precision, we do
not expect this either for the gravitational potential in the central region.

6. Discussion and Conclusions

In this work, we have presented a new method for solving the 3D gravitational potential of a density field on the Yin–Yang grid based on the algorithm developed by Müller & Steinmetz (1995). Our approach is particularly efficient on distributed-memory machines with a large number of compute tasks, because the amount of data being explicitly communicated is minimized and the work load is evenly distributed across all tasks resulting in a good scaling behavior.

In contrast to the strategy by Wongwathanarat et al. (2010), who interpolated the density from the Yin–Yang grid onto an auxiliary spherical mesh, we perform all operations on the original Yin–Yang grid. Our algorithm is completely symmetric with respect to the Yin–Yang grid patches and does not introduce a preferred direction. The validity of the results have been verified by performing test calculations that have an analytical or a semi-analytical solution.

Our ellipsoidal test configuration mimics the situation near the surface of the newly formed neutron star during the later post-bounce evolution of a core-collapse supernova. The increasingly steeper density gradient between the neutron star, which may be centrifugally deformed due to rapid rotation, and its surrounding medium is represented by the boundary of the ellipsoid. Because our solver can accurately determine the gravitational potential in this considered test setup with its extremely “sharp” surface, proper results can also be expected for the more moderate situation met during the birth of neutron stars during stellar core collapse and explosion.

The maximum multipole order ℓ_max of the spherical harmonics decomposition determines the accuracy of the gravitational potential. The computational cost increases roughly as ℓ_max^2. A suitable balance between accuracy and computing time has to be found depending on the characteristics of the investigated physical problem. In core-collapse supernova models, the gravitational potential is dominated by the approximately spherically symmetric proto-neutron star. Therefore, a moderate ℓ_max of ~25 should be able to capture most of the asymmetries.

We thank Alexander Summa and Annop Wongwathanarat for fruitful discussions. This project was supported by the European Research Council through grant ERC-AdG No. 341157-COCO2CASA and by the Deutsche Forschungsgemeinschaft through the Excellence Cluster “Universe” (EXC-153). The test calculations were performed on the Hydra system of the Max Planck Computing and Data Facility (MPCDF). The code scaling was tested on SuperMUC at the Leibniz Supercomputing Center with resources granted by the Gauss Centre for Supercomputing (LRZ project ID: pr53yi).

Software: PROMETHEUS-VERTEX (Fryxell et al. 1989; Rampf & Janka 2002; Buras et al. 2006), NumPy and Scipy (Oliphant 2007), IPython (Perez & Granger 2007), Matplotlib (Hunter 2007).

Appendix

Recurrence Relations for the Integrals of the Associated Legendre Polynomials

The integrals \( T^{(ℓm)} \) defined in Equation (18) can be calculated using the following recurrence relations (Zwerger 1995), which are evaluated in the order given below.

\[
T^{(00)}_j = P^0_0(\cos \vartheta_j^+) - P^0_0(\cos \vartheta_j^-),
\]

\[
T^{(ℓ0)}_j = \frac{1}{2ℓ + 1} \left[ P^{0}_{ℓ-1}(\cos \vartheta_j^+) - P^{0}_{ℓ-1}(\cos \vartheta_j^-) + P^{0}_{ℓ+1}(\cos \vartheta_j^+) - P^{0}_{ℓ+1}(\cos \vartheta_j^-) \right]
\]

(only for \( ℓ > 0 \)),

\[
T^{(11)}_j = \frac{1}{2} \left[ \vartheta_j^+ - \vartheta_j^- + \frac{1}{2} ( \sin(2\vartheta_j^+ - \sin(2\vartheta_j^-)) \right],
\]

\[
T^{(22)}_j = \frac{1}{4} (\cos(3\vartheta_j^+ - \cos(3\vartheta_j^-)) - 9(\cos \vartheta_j^+ - \cos \vartheta_j^-),
\]

\[
T^{(ℓℓ)}_j = \frac{1}{ℓ + 1} \left[ \cos \vartheta_j^+ P^ℓ_ℓ(\cos \vartheta_j^+) - \cos \vartheta_j^- P^ℓ_ℓ(\cos \vartheta_j^-) + ℓ(2ℓ - 3)(2ℓ - 1)T^{(ℓ-2,ℓ-2)}_j \right],
\]

k

\[
T^{(ℓℓ-1)}_j = \frac{1}{ℓ + 1} \left[ \sin \vartheta_j^+ P^ℓ_ℓ(\cos \vartheta_j^+) - \sin \vartheta_j^- P^ℓ_ℓ(\cos \vartheta_j^-) \right],
\]

\[
T^{(ℓm)}_j = \frac{1}{(ℓ + 1)(ℓ - m)} \left[ (2ℓ - 1)(\sin^2 \vartheta_j^+ P^m_ℓ(\cos \vartheta_j^+) - \sin^2 \vartheta_j^- P^m_ℓ(\cos \vartheta_j^-)) + (ℓ - 2)(ℓ + m - 1)T^{(ℓ-2,m)}_j \right].
\]

ORCID iDs

Hans-Thomas Janka @ https://orcid.org/0000-0002-0831-3330
References

Buras, R., Rampp, M., Janka, H.-T., & Kifonidis, K. 2006, A&A, 447, 1049

Chandrasekhar, S. 1969, Ellipsoidal Figures of Equilibrium (New Haven, CT: Yale Univ. Press)

Falle, S. A. E. G., Vaidya, B., & Hartquist, T. W. 2017, MNRAS, 465, 260

Fryxell, B., Müller, E., & Arnett, D. 1989, in Proc. 5th Workshop on Nuclear Astrophysics, ed. W. Hillbrandt & E. Müller (Garching: Max-Planck-Institut für Physik und Astrophysik), 100

He, Y., & Ding, C. H. Q. 2001, J. Supercomput., 18, 259

Hunter, J. D. 2007, CSE, 9, 10

Kageyama, A., & Sato, T. 2004, GGG, 5, 9005

Komatitsch, D., & Tromp, J. 2002, Geol, 150, 303

Lentz, E. J., Brunner, S. W., Hix, W. R., et al. 2015, ApJL, 807, L31

Liebendörfer, M., Rampp, M., Janka, H.-T., & Mezzacappa, A. 2005, ApJ, 620, 840

Marek, A., Dimmelmeier, H., Janka, H.-T., Müller, E., & Buras, R. 2006, A&A, 445, 273

Marek, A., & Janka, H.-T. 2009, ApJ, 694, 664

Melson, T., Janka, H.-T., Bollig, R., et al. 2015a, ApJL, 808, L42

Melson, T., Janka, H.-T., & Marek, A. 2015b, ApJL, 801, L24

Mondal, P., & Korenaga, J. 2018, Geol, 212, 1859

Müller, B., & Chan, C. 2018, arXiv:1806.06623

Müller, B., Janka, H.-T., & Dimmelmeier, H. 2010, ApJS, 189, 104

Müller, B., Melson, T., Heger, A., & Janka, H.-T. 2017, MNRAS, 472, 491

Müller, E., & Steinmetz, M. 1995, CoPhC, 89, 45

Oliphant, T. E. 2007, CSE, 9, 10

Ott, C. D., Roberts, L. F., da Silva Schneider, A., et al. 2018, ApJL, 855, L3

Peng, X., Xiao, F., & Takahashi, K. 2006, QJRMS, 132, 979

Perez, F., & Granger, B. E. 2007, CSE, 9, 21

Rampp, M., & Janka, H.-T. 2002, A&A, 396, 361

Roberts, L. F., Ott, C. D., Haas, R., et al. 2016, ApJ, 831, 98

Simon, J. B., Armitage, P. J., Li, R., & Youdin, A. N. 2016, ApJ, 822, 55

Summa, A., Hanke, F., Janka, H.-T., et al. 2016, ApJ, 825, 6

Summa, A., Janka, H.-T., Melson, T., & Marek, A. 2018, ApJ, 852, 28

Takiwaki, T., Kotake, K., & Suwa, Y. 2012, ApJ, 749, 98

Takiwaki, T., Kotake, K., & Suwa, Y. 2014, ApJ, 786, 83

Thakur, R., Rabenseifner, R., & Gropp, W. 2005, Int. J. High Performance Computing Applications, 19, 49

Wongwathanarat, A., Hammer, N. J., & Müller, E. 2010, A&A, 514, A48

Zwerger, T. 1995, PhD thesis, Technische Universität München

The Astrophysical Journal, 863:142 (7pp), 2018 August 20