The Astrophysical Journal, 863:142 (7pp), 2018 August 20 https://doi.org/10.3847/1538-4357/aad33a

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

Marius Almanstötter[1,2], Tobias Melson[2], Hans-Thomas Janka[2], and Ewald Müller[2]
[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 proto-neutron 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, Wongwathanarat 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, $\varrho(\boldsymbol{r})$, the gravitational potential at the location $\boldsymbol{r}$ is determined by solving Poisson's equation, which can be expressed in its integral form as

$$\Phi(\boldsymbol{r}) = -G \int_V \frac{\varrho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d^3\boldsymbol{r}', \tag{1}$$

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 $\boldsymbol{r} = (r, \vartheta, \varphi)$. A decomposition of $|\boldsymbol{r} - \boldsymbol{r}'|^{-1}$ into spherical harmonics $Y_{\ell m}$ yields

(Müller & Steinmetz 1995, Equation (8))

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

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

$$C_{\ell m}(r) := \int_0^{2\pi}\int_0^{\pi} d\varphi' d\vartheta' \sin\vartheta' \, \overline{Y_{\ell m}}(\vartheta', \varphi') \times \int_0^r dr'(r')^{\ell+2}\varrho(\boldsymbol{r}'), \tag{3}$$

$$D_{\ell m}(r) := \int_0^{2\pi}\int_0^{\pi} d\varphi' d\vartheta' \sin\vartheta' \, \overline{Y_{\ell m}}(\vartheta', \varphi') \times \int_r^{\infty} dr'(r')^{1-\ell}\varrho(\boldsymbol{r}'), \tag{4}$$

where $\overline{Y_{\ell m}}$ 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, $\Phi_{\text{in}}^{(\ell m)}(\boldsymbol{r})$, and a second for the potential outside, $\Phi_{\text{out}}^{(\ell m)}(\boldsymbol{r})$,

$$\Phi(\boldsymbol{r}) = -G\sum_{\ell=0}^{\infty}\sum_{m=0}^{\ell} N^{(\ell m)}P_\ell^m(\cos\vartheta)[\Phi_{\text{in}}^{(\ell m)}(\boldsymbol{r}) + \Phi_{\text{out}}^{(\ell m)}(\boldsymbol{r})], \tag{5}$$

where P_ℓ^m are the associated Legendre polynomials. The integrals for these inner and outer contributions can be expressed as

$$\Phi_{\text{in}}^{(\ell m)}(\boldsymbol{r}) = \frac{1}{r^{\ell+1}}\int_0^{2\pi}\int_0^{\pi} d\varphi' d\vartheta' \sin\vartheta' P_\ell^m(\cos\vartheta') \times \cos(m(\varphi - \varphi'))\int_0^r dr'(r')^{\ell+2}\varrho(\boldsymbol{r}') \tag{6}$$

and

$$\Phi_{\text{out}}^{(\ell m)}(\boldsymbol{r}) = r^{\ell}\int_0^{2\pi}\int_0^{\pi} d\varphi' d\vartheta' \sin\vartheta' P_\ell^m(\cos\vartheta') \times \cos(m(\varphi - \varphi'))\int_r^{\infty} dr'(r')^{1-\ell}\varrho(\boldsymbol{r}'). \tag{7}$$

The normalization factor in Equation (5) is given by

$$N^{(\ell m)} := \frac{(\ell - m)!}{(\ell + m)!}\frac{2}{\delta^{(m)}}, \tag{8}$$

with

$$\delta^{(m)} := \begin{cases} 2, & \text{if} \quad m = 0, \\ 1, & \text{if} \quad m > 0. \end{cases} \tag{9}$$

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.,

$$\varrho(\boldsymbol{r}) = \varrho_{ijk}, \tag{10}$$

for $r_i^- \leqslant r \leqslant r_i^+$, $\vartheta_j^- \leqslant \vartheta \leqslant \vartheta_j^+$, and $\varphi_k^- \leqslant \varphi \leqslant \varphi_k^+$. This assumption allows for simplifying Equations (6) and (7), which can then be written as

$$\Phi_{\text{in}}^{(\ell m)}(\boldsymbol{r}) = \frac{1}{r^{\ell+1}}\sum_{i=1}^{n_r}\sum_{j=1}^{n_\vartheta}\sum_{k=1}^{n_\varphi}\varrho_{ijk}\,\mathcal{R}_{\text{in,i}}^{(\ell)}\,\mathcal{T}_j^{(\ell m)}\,\mathcal{F}_k^{(m)}(\varphi), \tag{11}$$

$$\Phi_{\text{out}}^{(\ell m)}(\boldsymbol{r}) = r^{\ell}\sum_{i=n_r+1}^{n_R}\sum_{j=1}^{n_\vartheta}\sum_{k=1}^{n_\varphi}\varrho_{ijk}\,\mathcal{R}_{\text{out,i}}^{(\ell)}\,\mathcal{T}_j^{(\ell m)}\,\mathcal{F}_k^{(m)}(\varphi), \tag{12}$$

with

$$\mathcal{F}_k^{(m)}(\varphi) := \cos(m\varphi)\mathcal{C}_k^{(m)} + \sin(m\varphi)\mathcal{S}_k^{(m)}. \tag{13}$$

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 $\mathcal{C}_k^{(m)}$ and $\mathcal{S}_k^{(m)}$ can be evaluated analytically:

$$\mathcal{C}_k^{(m)} := \int_{\varphi_k^-}^{\varphi_k^+} d\varphi' \cos(m\varphi') = \begin{cases} \varphi_k^+ - \varphi_k^- = \Delta\varphi_k, & \text{if } m = 0, \\ \dfrac{1}{m}[\sin(m\varphi_k^+) - \sin(m\varphi_k^-)], & \text{if } m > 0, \end{cases} \tag{14}$$

and

$$\mathcal{S}_k^{(m)} := \int_{\varphi_k^-}^{\varphi_k^+} d\varphi' \sin(m\varphi') = \begin{cases} 0, & \text{if } m = 0, \\ \frac{1}{m}[\cos(m\varphi_k^-) - \cos(m\varphi_k^+)], & \text{if } m > 0. \end{cases} \tag{15}$$

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

$$\mathcal{R}_{\text{in},i}^{(\ell)} := \int_{r_i^-}^{r_i^+} dr'(r')^{\ell+2} = \frac{1}{\ell+3}((r_i^+)^{\ell+3} - (r_i^-)^{\ell+3}), \tag{16}$$

$$\mathcal{R}_{\text{out},i}^{(\ell)} := \int_{r_i^-}^{r_i^+} dr'(r')^{1-\ell} = \begin{cases} \ln(r_i^+) - \ln(r_i^-), & \text{if } \ell = 2, \\ \frac{1}{2-\ell}[(r_i^+)^{2-\ell} - (r_i^-)^{2-\ell}], & \text{if } \ell \neq 2. \end{cases} \tag{17}$$

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

$$\mathcal{T}_j^{(\ell m)} := \int_{\vartheta_j^-}^{\vartheta_j^+} d\vartheta' \sin\vartheta' P_\ell^m(\cos\vartheta'), \tag{18}$$

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 ℓ, which we denote as $\ell_{\max}$. We will discuss the choice of $\ell_{\max}$ 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