PAPER

# Source term method for binary neutron stars initial data



IOP Publishing Classical and Quantum Gravity
Class. Quantum Grav. **38** (2021) 135008 (23pp) https://doi.org/10.1088/1361-6382/abfc29

# Source term method for binary neutron stars initial data

**Bing-Jyun Tsao[1,*], Roland Haas[1] and Antonios Tsokaros[2]**

[1] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States of America
[2] Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States of America

E-mail: btsao2@illinois.edu

Received 12 February 2021, revised 8 April 2021
Accepted for publication 27 April 2021
Published 1 June 2021

**Abstract**

The initial condition problem for a binary neutron star system requires a Poisson equation solver for the velocity potential with a Neumann-like boundary condition on the surface of the star. Difficulties that arise in this boundary value problem are: (a) the boundary is not known *a priori*, but constitutes part of the solution of the problem; (b) various terms become singular at the boundary. In this work, we present a new method to solve the fluid Poisson equation for irrotational/spinning binary neutron stars. The advantage of the new method is that it does not require complex fluid surface fitted coordinates and it can be implemented in a Cartesian grid, which is a standard choice in numerical relativity calculations. This is accomplished by employing the source term method proposed by Towers, where the boundary condition is treated as a jump condition and is incorporated as additional source terms in the Poisson equation, which is then solved iteratively. The issue of singular terms caused by vanishing density on the surface is resolved with an additional separation that shifts the computation boundary to the interior of the star. We present two-dimensional tests to show the convergence of the source term method, and we further apply this solver to a realistic three-dimensional binary neutron star problem. By comparing our solution with the one coming from the initial data solver cocal, we demonstrate agreement to approximately 1%. Our method can be used in other problems with non-smooth solutions like in magnetized neutron stars.

Keywords: source term method, binary neutron stars, initial data, Poisson solvers

(Some figures may appear in colour only in the online journal)

*Author to whom any correspondence should be addressed.

  

## 1. Introduction

A system of two neutron stars (NSs) in a quasicircular orbit constitutes one of the finest laboratories in modern astrophysics since it is the place were a trinity of important phenomena intersect: (a) densities of matter beyond nuclear density, (b) the nucleosynthesis of heavy elements, (c) the formation of powerful gamma-ray bursts. Behind the scenes, it is the synergy of all the forces in nature with maximal strength that is responsible for the creation of the extreme conditions which lead to this intersection. Event GW170817 revealed in the most dramatic way the above crossroads and initiated the era of gravitational wave multimessenger astronomy by Advanced LIGO/Virgo [1], the Fermi Gamma-ray burst monitor [2, 3] and INTEGRAL [4, 5]. This event was the first time both the gravitational waves and the electromagnetic signal emitted during a quasicircular binary neutron star (BNS) collision were detected [2, 4, 6–9]. Although the possibility of exotic compact objects or even a neutron star-black hole system could not be ruled out, the characteristics of GW170817 suggest that a BNS is the most probable scenario [6].

In order to predict the outcome of a BNS merger, one needs to solve at least the Einstein and Euler equations (for more realistic scenarios additional complexity is present through Maxwell equations and many others) for two NSs in circular orbits. Initial data for such configurations were first presented by Baumgarte *et al.* [10, 11] and Marronetti *et al.* [12], which described two NSs in synchronous rotation (an example of synchronous rotation is the Moon in the Earth-Moon system). These solutions where both objects are in synchronous rotation are called corotating solutions, and they gave valuable insights of the two body problem in general relativity. Since the works of Bildsten and Cutler [13] and Kochanek [14], such NS configurations were considered unrealistic as viscosity is considered to be too small in order for synchronization to manifest. Despite this, corotating solutions still play an important role since they represent the simplest spinning binary NSs. According to our current understanding BNSs are considered to be mostly irrotational, i.e. having zero vorticity. Data describing such systems are more complicated to be computed since the conservation of rest mass is not identically satisfied as in the corotating case but results in an additional potential equation. The first numerical solutions of irrotational BNSs were presented by Bonazzola *et al.* [15], Gourgoulhon *et al.* [16], Marronetti *et al.* [17, 18], and Uryū *et al.* [19, 20]. Notwithstanding the fact that irrotational BNSs are expected to be the most frequent in nature, one cannot exclude the possibility of BNSs that exhibit spin. After all, even event GW170817 [1] was unable to exclude a highly spinning binary. Solving the Euler equations in the presence of spin is a challenging problem since no trivial integral exists. There were many attempts to address that problem, for example, by Marronetti and Shapiro [21], Baumgarte and Shapiro [22, 23], and Tsatsin and Marronetti [24], but the first self-consistent formulation was presented by Tichy [25–27].

One common characteristic of both irrotational and spinning solutions is the potential equation that results from the conservation of rest mass and the symmetry of the quasicircular motion [28]. In particular, the fluid velocity can be decomposed as a gradient of a potential plus a spinning part which is zero for the irrotational case. Then the divergence of the fluid velocity leads to an elliptic equation for the fluid potential with boundary conditions on the surface of the star. There are two problems that arise in this boundary value problem (BVP): (a) the surface of the star is not known and constitutes part of the solution of the problem, (b) terms that contain the density of the star become singular as we approach the surface. In order to address these issues, various techniques have been employed over the past decades [15, 19, 28] that, despite exhibiting slow convergence, proved robust in providing physically acceptable solutions.

In this paper, we present a new method to solve the nonlinear Poisson equation for the irrotational/spinning flow in a BNS system. One of the main advantages of our method is its simplicity since it does not require complex surface fitted coordinates and it can be implemented in a Cartesian grid which is a standard choice in numerical relativity calculations. This is accomplished by employing the source term method (STM) proposed by Towers [29] where the boundary condition is treated as a jump condition and is incorporated as additional source terms in the Poisson equation, which is then solved iteratively. As a result, the fluid sector of a BNS can be treated in similar manner as the gravitational elliptic equations with no need of extra steps (see [28]) that make the iteration process complex. The achieved generality will be very useful in the treatment of more complicated physical systems that have non-smooth solutions across the surface, such as magnetized rotating NSs.

This article is presented in four parts. In section 2, the fluid equation for the BNS initial value problem is presented; section 3 gives a detailed introduction of the STM that will be used for solving the Poisson problem; section 4 describes the discretization scheme; numerical tests and comparisons with realistic initial data from the cocal solver are presented in section 5. Throughout the article we use units of $G = c = M_\odot = 1$, and a space-time signature $(-+++)$. Spacetime indices will be indicated with Greek letters, and spatial indices with Latin letters.

## 2. Fluid equations in binary neutron stars

In this section we introduce the fluid equation describing irrotational and spinning BNS [15, 25, 30, 31]. Our discussion will follow closely the works of [19, 28, 32] where details are provided in full. The numerical implementation of this BVP in the context of the cocal code is described in [28, 33] and will not be repeated here. Other successful initial data solver implementations include the lorene code [34], the sgrid code [35], and the spells code [36].

### *2.1. 3 + 1 decomposition*

We assume that the spacetime $\mathcal{M}$ is foliated by a family of spacelike hypersurfaces $\Sigma_t$, i.e. $\mathcal{M} = \mathbb{R} \times \Sigma_t$ with $t \in \mathbb{R}$. The spacetime metric can then be written in $3 + 1$ form as

$$ds^2 = -\alpha^2\, dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j\, dt), \tag{1}$$

where $\alpha$, $\beta^i$, $\gamma_{ij}$ are, respectively, the lapse, the shift vector, and the spatial three-metric on $\Sigma_t$. For the three-geometry we assume it is conformally flat

$$\gamma_{ij} = \psi^4 \delta_{ij}, \tag{2}$$

where $\psi$ is the conformal factor.

If we neglect the loss of energy due to gravitational radiation and assume closed orbits for the BNS system, this implies the existence of a helical Killing vector

$$k^\mu \equiv t^\mu + \Omega\phi^\mu, \tag{3}$$

such that $\mathcal{L}_{\boldsymbol{k}} g_{\alpha\beta} = 0$, where $\mathcal{L}_{\boldsymbol{k}}$ is the Lie derivative along $\boldsymbol{k}$. In equation (3) $t^\mu$ is the vector that represents the flow of time, $\Omega$ is the orbital angular velocity, and $\phi^\mu$ is the generator of rotational symmetry. In a Cartesian coordinate system, without loss of generality, we can assume $\phi^\mu = (0, \phi^i) = (0, -y, x, 0)$, i.e. that the orbital angular momentum is along the $z$-axis. By introducing

the unit timelike normal $n_\mu \equiv -\alpha \nabla_\mu t$ (where $\nabla$ denotes the covariant derivative associated with the metric $g_{\alpha\beta}$) to the hypersurfaces $\Sigma_t$, one can rewrite the Killing vector $k^\alpha$ as

$$k^\alpha \equiv \alpha n^\mu + \omega^\mu \tag{4}$$

where $\omega^\mu = \beta^\mu + \Omega\phi^\mu$ is the corotating shift, and $t^\mu \equiv \alpha n^\mu + \beta^\mu$.

The four-velocity of the fluid is $u^\alpha = u^t(1, v^i)$ where $v^i$ is the Newtonian velocity (velocity with respect to the inertial observers) that can be split into two parts: one that follows the rotation around the center of mass, $\Omega\phi^i$, and the velocity with respect to the corotating observer $V^i$, i.e.

$$u^\alpha \equiv u^t(t^\alpha + v^\alpha) = u^t(k^\alpha + V^\alpha). \tag{5}$$

where $V^\alpha = (0, V^i)$ and $v^\alpha = (0, v^i) \equiv \Omega\phi^\alpha + V^\alpha$. For corotating BNSs, $V^i = 0$.

### 2.2. The potential fluid equation

For an irrotational flow, the relativistic vorticity tensor

$$\omega_{\alpha\beta} \equiv \nabla_\alpha(hu_\beta) - \nabla_\beta(hu_\alpha) \tag{6}$$

is zero. Here $h$ is the relativistic specific enthalpy. This implies that in such a case $hu_\alpha = \nabla_\alpha \Phi$ i.e. the four-velocity can be derived from a fluid potential $\Phi$. In the general case where the stars are arbitrarily spinning, one can write

$$hu_\alpha = \nabla_\alpha \Phi + s_\alpha, \tag{7}$$

where $s_\alpha$ is their spin vector. In the decomposition above, we implicitly follow the discussion in [25, 28] which is by no means unique. In a similar way, one can follow [32] where another decomposition based on the circulation has been proposed. In such case the potential equation has a different form although it still preserves the characteristics that motivate this work.

In $3 + 1$ form, the components of interest of the velocity are the ones projected into the hypersurface $\Sigma_t$ through the projection tensor $\gamma_{\mu\nu} = g_{\mu\nu} + n_\mu n_\nu$. Its pullback to $\Sigma_t$ is the tensor $\gamma_{ij}$. Defining

$$\hat{u}^i \equiv \gamma^i_\mu hu^\mu = hu^t(\omega^i + V^i). \tag{8}$$

Equations (7) and (5) imply

$$V^i = \frac{D^i\Phi + s^i}{hu^t} - \omega^i = \frac{\delta^{ij}\partial_j\Phi/\psi^4 + s^i}{hu^t} - (\beta^i + \Omega\phi^i) \tag{9}$$

where $D_i$ is the covariant derivative associated with $\gamma_{ij}$, and $\delta^{ij}$ is the Kronecker delta. Conservation of rest mass together with equation (9) yields an elliptic equation for $\Phi$

$$\begin{aligned}\partial_i\partial^i\Phi = &-\frac{2}{\psi}\partial_i\psi\partial^i\Phi + \psi^4\omega^i\partial_i(hu^t) - \psi^4\left(\partial_i s^i + s^i\partial_i \ln \psi^6\right)\\ &- \frac{h}{\alpha\rho}(\partial^i\Phi + \psi^4 s^i - \psi^4 hu^t\omega^i)\partial_i\frac{\alpha\rho}{h},\end{aligned} \tag{10}$$

where $\rho$ is the rest-mass density inside the star. We call equation (10) the potential fluid equation. On the surface of the star we have $\rho = 0$, therefore evaluating equation (10) there, we obtain a Neumann-type boundary condition [28]

$$\left[(\partial^i\Phi + \psi^4 s^i - \psi^4 h u^t \omega^i)\partial_i\rho\right]_{\text{surface}} = 0. \tag{11}$$

In the cocal solver, a numerical method to find the solution of the boundary problem above is implemented and described in detail in [19, 28]. The method used in cocal inverts the Laplacian through the Green's function $G(x, x') = 1/|x - x'|$ while adding a harmonic function such that the sum of the two will satisfy the boundary condition equation (11). In addition, separate to the normal coordinate system, fluid surface fitted coordinates are implemented. In section 5.3 of this article, we present a comparison between the solution from the new approach using the STM and the solution from cocal. Notice that the STM does not need to implement fluid surface fitted coordinates.

It should be noted that in real BNS initial data, the variables in equations (10) and (11) besides $\Phi$ (i.e. $h$, $u^t$, $\alpha$, etc) are not known *a priori*, and they also depend on $\Phi$. Specifically, five more elliptic equations [28] for $\psi$, $\alpha$, and $\beta^i$ would have to be solved simultaneously with equation (10). The elliptic equation for $\Phi$ is the primary focus in this article particularly because the boundary condition equation (11) is imposed on the surface of the star. In contrast, the other five elliptic equations impose boundary conditions at spatial infinity where asymptotic flatness of spacetime is assumed [28]. For a complete BNS initial data solver, an iterative scheme for equation (10) as well as the other five equations is implemented such that each variable is solved independently while the others are fixed.

For simplicity, we will assume irrotational flows, i.e. $s^i = 0$ in this work. All terms in the right-hand side of equation (10) are easily computed except the last one that includes a division by $\rho$, which tends to zero at the surface of the NS. This 'problematic' term exists even when $s^i = 0$ and in the following sections we describe in detail how to treat it. The qualitative behavior of equation (10) is not affected by the spin since in effect it only modifies the also regular shift terms (proportional to $\omega^i$) both in equation (10) as well as in the boundary condition equation (11). Term $\partial_i s^i$ for NSs that are uniformly rotating is approximately constant, thus it does not influence the solution method presented here. Therefore, including spin does not complicate the problem and can be trivially accounted for in the methods presented below.

## 3. The source term method

In this section, we will discuss a method by Towers [29], called the STM, that solves the fluid potential BVP (equations (10) and (11)) in a novel way that encapsulates the boundary condition on the surface of the star as jump conditions. Notably, by embedding the domain of the star in a *larger computational domain*, this method does not require a surface fitted coordinate system, thus providing an easy and robust alternative method to the BNS initial value problem.

The fluid potential BVP equations (10) and (11) can be written in general as

$$\begin{cases} \rho > 0\,(\text{interior}), & \nabla^2\Phi = S^+(\vec{x}), \\ \rho = 0\,(\text{surface}), & \Phi_n = a(\vec{x}), \end{cases} \tag{12}$$

where

$$\Phi_n \equiv \frac{\partial}{\partial n}\Phi \equiv \hat{n}\cdot\nabla\Phi$$

is the derivative of $\Phi$ along the unit normal of the surface of the star $\hat{n}$, and $S^+(\vec{x})$ is the right-hand side of equation (10). Note that from this section onwards, we will denote by $\hat{n}$ the unit normal of the boundary (not to be confused with the unit timelike normal $n^\mu$ introduced in equation (4)) and $\nabla$ the spatial derivative operator in Cartesian coordinates (not to be confused with the covariant derivative operator $\nabla_\mu$ in section 2). In the real BNS initial value problem, the surface of the star is not known *a priori* and constitutes part of the solution. Here, we consider the problem where the density $\rho$ is given, which indicates the surface where $\rho = 0$ is known as well. The unit normal to the surface can be written as $\hat{n} = -\nabla\rho/|\nabla\rho|$, and thus the boundary condition equation (11) yields

$$\Phi_n \equiv -\frac{\nabla\Phi\cdot\nabla\rho}{|\nabla\rho|} = -\frac{\psi^4 h u^t \vec{\omega}\cdot\nabla\rho}{|\nabla\rho|}. \tag{13}$$

The main idea of the STM [29] is, instead of solving the Neumann BVP with an unknown boundary (the surface of the star), to embed these boundary conditions as jump conditions in the source terms, $S^+$, and solve a generalized Poisson equation that contains extra terms in addition to $S^+$ due to the boundary conditions on a *larger computational grid*. In effect, this procedure moves the unknowns from the boundary to the source terms of the Poisson equations, and hence its name. In the next section we describe how this procedure can be accomplished and the meaning of the larger computational domain.

### 3.1. Domain embedding and the source term method equation

Consider the domain $\Omega = \Omega^+ \cup \Omega^-$ in figure 1 and the following BVP,

$$\begin{cases} \nabla^2\Phi = S^+(\vec{x}), & \vec{x}\in\Omega^+, \\ \nabla^2\Phi = S^-(\vec{x}), & \vec{x}\in\Omega^-, \\ \Phi = 0, & \vec{x}\in\partial\Omega, \\ [\Phi_n] = a(\vec{x}), & \vec{x}\in\partial\Omega^+, \\ [\Phi] = b(\vec{x}), & \vec{x}\in\partial\Omega^+, \end{cases} \tag{14}$$

where we denote by $[\Phi] \equiv \Phi^+ - \Phi^- = b(\vec{x})$ and $[\Phi_n] \equiv \Phi_n^+ - \Phi_n^- = a(\vec{x})$. Here $\Phi^+, \Phi_n^+$ are the solution $\Phi$ and its normal derivative $\Phi_n$ evaluated at the interior side of the boundary $\partial\Omega^+$, and $\Phi^-, \Phi_n^-$ are those evaluated at the exterior side of $\partial\Omega^+$. The interior of the star is represented by $\Omega^+$ and its boundary $\partial\Omega^+$ is defined through the level set function $\varphi(\vec{x})$ which typically can be taken to be the rest-mass density $\rho(\vec{x})$ of the star (as in figure 1) and is used to characterize the various regions of interest.

$$\begin{cases} \varphi > 0, & \vec{x}\in\Omega^+, \\ \varphi < 0, & \vec{x}\in\Omega^-, \\ \varphi = 0, & \vec{x}\in\partial\Omega^+. \end{cases} \tag{15}$$

Notice that although $\rho < 0$ does not exist in a BNS problem, we can still make such an assumption in order to establish the level set function equation (15). The unit normal vector corresponding to the level set function is thus

$$\hat{n} \equiv -\frac{\nabla\varphi}{|\nabla\varphi|}. \tag{16}$$

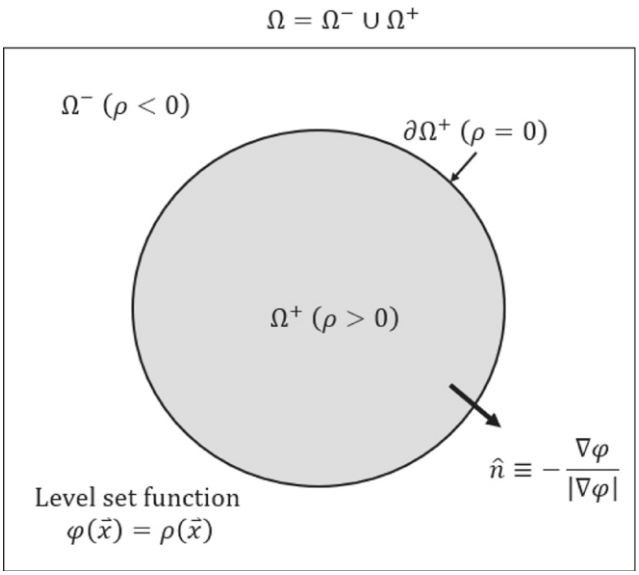


**Figure 1.** The diagram shows the domain of interest $\Omega^+$ in gray with the boundary condition specified on $\partial\Omega^+$. Instead of solving the BVP on $\partial\Omega^+$, the boundary condition is embedded as jump conditions and the Poisson equation is solved over the larger domain $\Omega$.

The STM treats the boundary condition at $\partial\Omega^+$ as a source term and solves the equation in the larger rectangular grid $\Omega = \Omega^+ \cup \Omega^-$. This allows the boundary conditions on $\partial\Omega^+$ to become jump conditions that can be absorbed into the sources. The generalized Poisson equation (see appendix A.1 for a derivation) that will be solved in the whole domain $\Omega$ becomes

$$\nabla^2\Phi = \nabla^2(bH) - H\nabla^2 b - \left(a - \frac{\partial b}{\partial n}\right)|\nabla\varphi|\delta(\varphi) + S, \tag{17}$$

where $S(\vec{x}) = S^+H(\varphi(\vec{x})) + S^-(1 - H(\varphi(\vec{x})))$, $H(\varphi)$ is the Heaviside step function, and $\delta(\varphi)$ is the Dirac delta function.

In the context of the fluid potential equation (12), the terms $S^-$ and $\Phi^-$ are freely specifiable and we choose $S^- = 0$ and $\Phi^- = 0$ to simplify the jump conditions $a(\vec{x})$, $b(\vec{x})$. To summarize, the coefficients in equation (17) are:

$$\begin{cases} S^+(\vec{x}) = -\left(\dfrac{2}{\psi}\nabla\psi + \nabla\left(\ln\dfrac{\alpha\rho}{h}\right)\right)\cdot\nabla\Phi^+ + \psi^4\vec{\omega}\cdot\nabla(hu^t) + \psi^4 hu^t\vec{\omega}\cdot\nabla\left(\ln\dfrac{\alpha\rho}{h}\right), \\ S^-(\vec{x}) = 0, \\ a(\vec{x}) \;\; = \Phi_n^+ - \Phi_n^- = \Phi_n^+ = -\dfrac{\psi^4 hu^t\vec{\omega}\cdot\nabla\rho}{|\nabla\rho|}, \\ b(\vec{x}) \;\; = \Phi^+ - \Phi^- = \Phi^+, \\ \varphi(\vec{x}) \;\; = \rho. \end{cases} \tag{18}$$

For a Neumann BVP, the values for $b(\vec{x})$ and $S^+(\vec{x})$ remain unknown as they depend on $\Phi$. This problem is addressed in [29], and an iterative scheme is proposed. The iterative scheme starts with an initial $\Phi = 0$. During each iteration, the STM solves for the Poisson equation with equation (17), and the coefficients in equation (18) are then updated based on the current $\Phi$ from the solver. The process continues until the solution $\Phi$ converges.

In order to simplify our code, equation (17) can be rewritten as a variable coefficient Poisson equation with the same boundary condition equation (13):

$$\begin{cases} \nabla\left(\zeta(\vec{x})\nabla\Phi\right) = S_\zeta^+(\vec{x}), & \vec{x} \in \Omega^+, \\ \nabla\left(\zeta(\vec{x})\nabla\Phi\right) = S_\zeta^-(\vec{x}), & \vec{x} \in \Omega^-, \\ \Phi_n = a(\vec{x}), & \vec{x} \in \partial\Omega^+, \end{cases} \tag{19}$$

where

$$\begin{cases} \zeta(\vec{x}) &= \dfrac{\psi^2\alpha\rho}{h}, \\ S_\zeta^+(\vec{x}) = \psi^6\vec{\omega}\cdot\nabla(\alpha\rho u^t), \\ S_\zeta^-(\vec{x}) = 0. \end{cases}$$

The source terms are being related through $S^+ = (S_\zeta^+ - \nabla\zeta\cdot\nabla\Phi)/\zeta$. This formulation is used for two reasons: (a) $S_\zeta^+$ does not depend on $\Phi$, thus the iterative STM can be simplified by taking only $\zeta, S_\zeta^+, a, \varphi$ as input variables; (b) in the Newtonian limit, $\zeta(\vec{x}) = \rho(\vec{x})$, so the algorithm is further simplified by taking $\zeta, S_\zeta^+, a$ as the only input variables.

## 4. Numerical discretization

In this work, we follow the algorithm and numerical discretization for the STM as described in [29]. For simplicity, we use the notation $Z_{i,j,k} \equiv Z(x_i, y_j, z_k)$ for any field $Z$ defined on the grid. The grid has spacing $\Delta x = \Delta y = \Delta z$ with $(N_{\text{grid}} + 1)$ points in every direction. We use second order finite differences for the partial derivatives and the Laplacian in 3D Cartesian coordinates

$$\begin{aligned} \frac{\partial Z_{i,j,k}}{\partial x} &= \frac{1}{2\Delta x}\left(Z_{i+1,j,k} - Z_{i-1,j,k}\right), \\ \nabla^2 Z_{i,j,k} &= \frac{1}{\Delta x^2}\left(Z_{i+1,j,k} + Z_{i-1,j,k} + Z_{i,j+1,k} + Z_{i,j-1,k}\right. \\ &\quad \left. + Z_{i,j,k+1} + Z_{i,j,k-1} - 6Z_{i,j,k}\right). \end{aligned}$$

For the source term in equation (17), a discretization of the Heaviside and delta function is required. Here we use the discretization provided in [37] and repeat the salient points for completeness of presentation. Since the boundary $\partial\Omega^+$ is defined through the level function $\varphi$, we first define the set $N_1$ that contains the grid points on or adjacent to $\partial\Omega^+$ (see figure 2) as

$$\begin{aligned} N_1 = \{(x_i, y_j, z_k) \mid\ & \varphi(x_{i\pm1}, y_j, z_k)\varphi(x_i, y_j, z_k) \leqslant 0 \quad \text{or} \\ & \varphi(x_i, y_{j\pm1}, z_k)\varphi(x_i, y_j, z_k) \leqslant 0 \quad \text{or} \\ & \varphi(x_i, y_j, z_{k\pm1})\varphi(x_i, y_j, z_k) \leqslant 0 \ \}. \end{aligned}$$

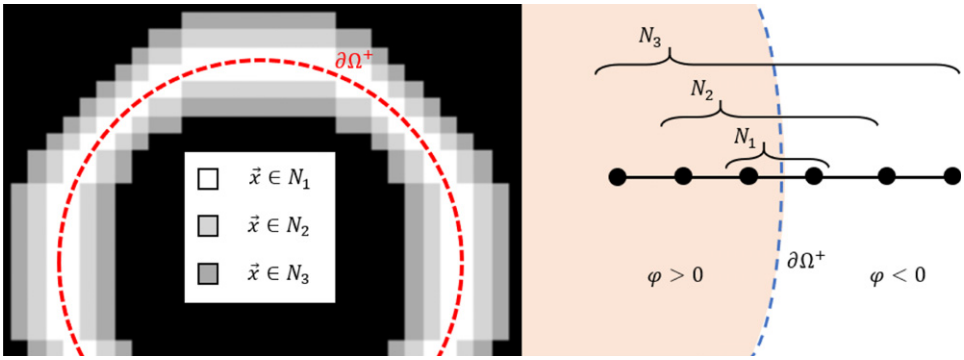


**Figure 2.** Diagrams showing the neighboring points for the discretization of the STM. The image on the left is a two-dimensional diagram showing $N_1$, $N_2$, $N_3$ based on the boundary $\partial\Omega^+$. The image on the right demonstrates a slice of the two-dimensional plot where $N_1 \subset N_2 \subset N_3$.

The characteristic function $\chi(\vec{x})$ of the set $N_1$ is defined to be

$$\chi(\vec{x}) = \begin{cases} 1, & \vec{x} \in N_1, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Further we define the neighboring sets $N_2, N_3, \ldots$ as

$$N_2 = \Xi(N_1), \qquad N_3 = \Xi(N_2), \ldots \tag{21}$$

where $\Xi(P)$ is a function that returns the set of grid points on or adjacent to the set of points $P$

$$\Xi(P) \equiv P \cup \{(x_i, y_j, z_k) \mid (x_{i\pm1}, y_j, z_k) \text{ or } (x_i, y_{j\pm1}, z_k) \text{ or } (x_i, y_j, z_{k\pm1}) \in P\}.$$

The neighboring sets $N_1, N_2, N_3, \ldots$ are visualized in figure 2, and these sets are used in this section and beyond to specify the domain used in the implementation of the STM.

For simplicity of the discretization formulas, we define functions $I(\varphi(\vec{x}))$, $J(\varphi(\vec{x}))$, and $K(\varphi(\vec{x}))$ as

$$I(\varphi(\vec{x})) = \max(0, \varphi(\vec{x})), \quad J(\varphi(\vec{x})) = \frac{1}{2} I(\varphi(\vec{x}))^2, \quad K(\varphi(\vec{x})) = \frac{1}{6} I(\varphi(\vec{x}))^3, \tag{22}$$

whose gradients can be written as $\nabla I(\varphi) = H(\varphi)\nabla\varphi$, $\nabla J(\varphi) = I(\varphi)\nabla\varphi$, and $\nabla K(\varphi) = J(\varphi)\nabla\varphi$. The discretization of the Heaviside function $H^h_{i,j,k}$ for $H(\varphi)$ and the delta function $\delta^h_{i,j,k}$ for $\delta(\varphi)$ are then expressed [38] as (see appendix A.2 for a derivation)

$$\begin{aligned} H^h_{i,j,k} &= \chi_{i,j,k} \left( \frac{\nabla^2 J_{i,j,k}}{|\nabla\varphi_{i,j,k}|^2} - \frac{(\nabla^2 K_{i,j,k} - J_{i,j,k}\nabla^2\varphi_{i,j,k})\nabla^2\varphi_{i,j,k}}{|\nabla\varphi_{i,j,k}|^4} \right) \\ &\quad + (1 - \chi_{i,j,k}) H(\varphi_{i,j,k}), \end{aligned} \tag{23}$$

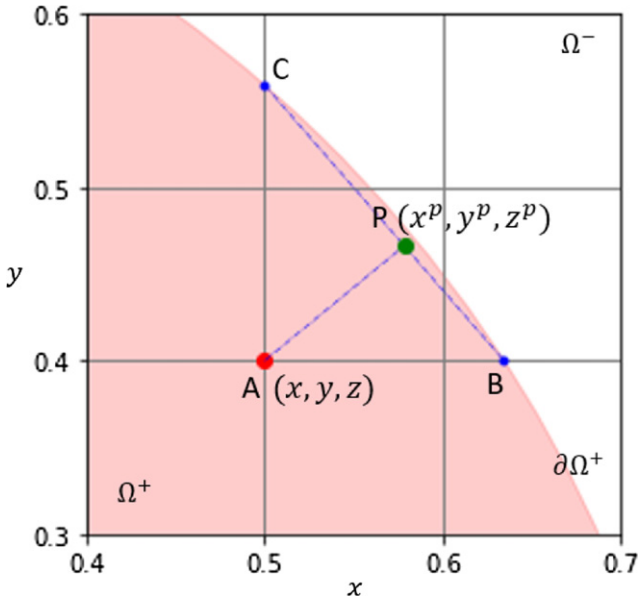


**Figure 3.** A diagram of the boundary extension of the grid points near the boundary. Assuming the level set $\varphi$ is decreasing trilinearly near the boundary, the projected point $P$ is the projection of the original grid point $A$ onto the plane determined by the projected points of $A$ on the boundary $\partial\Omega^+$ in the $x, y, z$ direction (as indicated by point $B$ and $C$ in the 2D diagram).

$$\delta^h_{i,j,k} = \chi_{i,j,k}\left(\frac{\nabla^2 I_{i,j,k}}{|\nabla\varphi_{i,j,k}|^2} - \frac{(\nabla^2 J_{i,j,k} - I_{i,j,k}\nabla^2\varphi_{i,j,k})\nabla^2\varphi_{i,j,k}}{|\nabla\varphi_{i,j,k}|^4}\right). \tag{24}$$

Given the discretization for $H^h_{i,j,k}$ equation (23) and $\delta^h_{i,j,k}$ equation (24), one can proceed with the discretization of equation (17) (note that $H^h_{i,j,k}$ is different from $H_{i,j,k} = H(\varphi_{i,j,k})$):

$$\begin{aligned}\nabla^2\Phi_{i,j,k} = \; & \nabla^2(b_{i,j,k}H(\varphi_{i,j,k})) - H^h_{i,j,k}\nabla^2 b_{i,j,k} \\ & - \left(a_{i,j,k} - \frac{\partial b_{i,j,k}}{\partial n}\right)|\nabla\varphi_{i,j,k}|\delta^h_{i,j,k} + H^h_{i,j,k}S^+_{i,j,k}.\end{aligned} \tag{25}$$

### *4.1. Boundary treatment*

When solving for the Poisson equation using equation (17), the boundary condition $a(\vec{x}) = \Phi^+_n(\vec{x})$ (equation (13)) needs to be imposed. There are two difficulties here. First, the boundary does not coincide with grid points in a typical case. Second, in the realistic case of BNSs, the boundary is not known *a priori*, but only when a solution is found. In other words, the boundary is changing through the iteration process. Both these problems imply that it is not trivial to apply boundary condition equation (13) for the Poisson equation. In order to address these problems, we compute the so-called 'projected points' $\{x^p, y^p, z^p\}$ [29] which in effect represent the numerical boundary where the boundary condition equation (13) is evaluated. These locations do not belong to the grid and are close to the true boundary $\partial\Omega^+$.

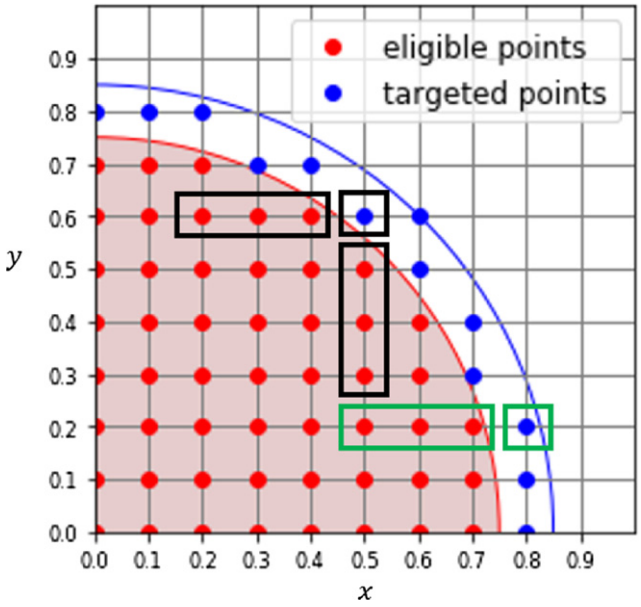


**Figure 4.** A diagram of the extrapolation method. For the target point at $(0.8, 0.2)$, the returned value is extrapolated from the three stencil points left to itself; for the target point at $(0.5, 0.6)$, the returned value is the average of the extrapolated value from the three stencil points to its left and the extrapolated value from the three stencil points underneath.

Using a first order Taylor approximation, the projected points can be written as

$$x^p = x + \frac{\varphi}{|\nabla\varphi|}\hat{n}_x, \qquad y^p = y + \frac{\varphi}{|\nabla\varphi|}\hat{n}_y, \qquad z^p = z + \frac{\varphi}{|\nabla\varphi|}\hat{n}_z,$$

$$\text{where } \hat{n}_x = \frac{-\partial_x\varphi}{|\nabla\varphi|}, \quad \hat{n}_y = \frac{-\partial_y\varphi}{|\nabla\varphi|}, \quad \hat{n}_z = \frac{-\partial_z\varphi}{|\nabla\varphi|}.$$

Figure 3 shows how the projected points are calculated from the grid points.

The extended boundary condition at each grid points $a^p(\vec{x})$ that will be used as the boundary condition in our code takes the value of $a(\vec{x})$ evaluated at the projected points $a^p(x, y, z) = a(x^p, y^p, z^p)$. For test cases where the boundary function $a(\vec{x})$ is given, the extended boundary condition $a^p$ can be directly calculated; for BNSs where the boundary condition is given as an array of values, an interpolation must be performed first to obtain the boundary function. A more detailed description of the effect of the extended boundary treatment on various types of BVP can be found in [29].

### *4.2. Extrapolation method*

For the finite difference discretization of the STM, the values of $\Phi$ and $\Phi_n$ are required to be defined throughout $N_2$ to avoid discrete jumps in the derivatives. Thus, an extrapolation method is needed to get the values outside the boundary in $N_2/\Omega^+$. Specifically, a quadratic extrapolation method used in [29] is implemented, as shown in figure 4.

The quadratic extrapolation algorithm is implemented for any given field $v(\vec{x})$ to find its extrapolated value $v_{\text{ext}}$, where in the STM $v$ can be $\Phi$ or $\Phi_n$. The algorithm takes in three

inputs: (a) an initial array of values $v_0$; (b) targeted points $\mathcal{T}_0$ that represent the initial set of points where the values are not defined and need to be extrapolated; (c) eligible points $\mathcal{E}_0$ that represent the initial set of grid points where the values are defined and available to use for extrapolation. The extrapolation algorithm uses an iterative method to extrapolate the points adjacent to the eligible points layer by layer until all targeted points have an extrapolated value. First, the algorithm initializes the variables $\mathcal{T} = \mathcal{T}_0$, $\mathcal{E} = \mathcal{E}_0$ and, $v = v_0$. In each iteration, for a target point $\vec{x} \in \mathcal{T}$, the algorithm checks, in the $\pm x, \pm y, \pm z$ direction, if all three stencil points are eligible points (as shown in figure 4), also expressed as an eligible direction $d$, and the extrapolated value $v_{\text{ext}}^{(d)}$ can be evaluated with quadratic extrapolation. For example, let $x_1$, $x_2$, $x_3$ be the three stencil points in the $+x$ direction, the extrapolated value $v_{\text{ext}}$ is given with quadratic extrapolation

$$v_{\text{ext}} = 3v(x_1) - 3v(x_2) + v(x_3). \tag{26}$$

To be specific, for a grid point $(x_i, y_j, z_k)$, the three stencil points in the $+x$ direction are $x_1 = (x_{i+1}, y_j, z_k)$, $x_2 = (x_{i+2}, y_j, z_k)$, $x_3 = (x_{i+3}, y_j, z_k)$. Similarly, in the $-x$ direction, the stencil points are $x_1 = (x_{i-1}, y_j, z_k)$, $x_2 = (x_{i-2}, y_j, z_k)$, $x_3 = (x_{i-3}, y_j, z_k)$.

If there is at least one eligible direction, then the grid point is eliminate from target points $\mathcal{T}$ and added to eligible points $\mathcal{E}$, and the value of the grid point $v$ is updated to the average of all the extrapolated values from all eligible directions. The process continues until the set of target points becomes empty.

### *4.3. Coefficient singularity on the surface*

It is an intuitive choice to have the density $\rho$ to be the level set function since the surface is defined where $\rho = 0$. However, this raises a singularity issue in the coefficient $\zeta$. As $\rho \to 0$ near the surface, $\zeta = \psi^2 \alpha \rho / h \to 0$. This becomes troublesome as the second order derivative in the equation vanishes, and a dividing by zero occurs when evaluating $S^+ = (S_\zeta^+ - \nabla\zeta \cdot \nabla\Phi)/\zeta$. Our solution is to shift the level set with a constant separation $\delta\varphi$:

$$\varphi(\vec{x}) = \rho(\vec{x}) - \delta\varphi. \tag{27}$$

The additional separation introduces a domain $W^+ \equiv \{\vec{x} | \varphi(\vec{x}) > 0\}$ that is slightly smaller than the original domain $\Omega^+ \equiv \{\vec{x} | \rho(\vec{x}) > 0\}$ (as shown in figure 5).

The iterative STM will solve for $\Phi$ in the smaller domain $W^+$, and then the solution is extrapolated from $W^+$ to the entire domain $\Omega^+$ to get the final solution $\Phi_{\text{ext}}$. Nevertheless, this additional separation in the domain also changes the boundary condition $a(\vec{x})$ from $\partial\Omega^+$ to $\partial W^+$, which will introduce increasing error as $\delta\varphi$ increases. Thus, the value of $\delta\varphi$ should be carefully chosen such that it is not so small that the singularity in $\Omega^+$ causes the iterative scheme to diverge, and it is not so large that the change in boundary condition $a(\vec{x})$ is no longer consistent. Typically, we choose the separation $\delta\varphi$ to be a portion of the central density $\rho_0$, such as $\delta\varphi = 0.05\rho_0$. The choice for the value of the separation will be discussed in more detail in section 5.2.

A detailed description of the pseudocode for the STM algorithm is presented in [29]. For the numerical experiments in the next section, the iterative STM terminates when the maximum relative change of the solution between successive iterations is less than $0.01\%$. For each iteration that does not terminate, the new solution is updated with a relaxation ratio of 0.1. For the fast Poisson solver for the entire domain $\Omega = \Omega^+ \cup \Omega^-$, we use the Jacobi iterative method that terminates when the relative change of the method in the two-norm of the solution between successive iterations is less than $0.001\%$.

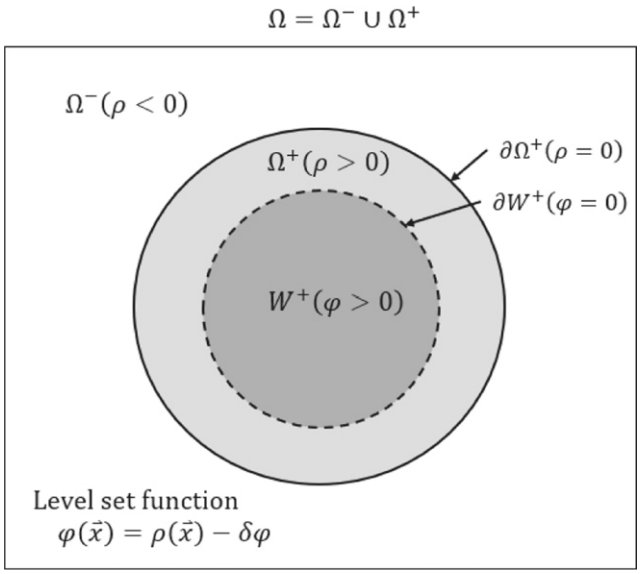


**Figure 5.** The diagram of the interior $\Omega^+$ and the domain $W^+$ that the solver actually solves for. An additional separation $\delta\varphi$ is added to the level set function to shift the boundary inwards in order to avoid the singularity of the coefficient $\zeta$.

## 5. Numerical results

In this section, numerical experiments are conducted to test the performance of the STM. We use the following rule for calculating the $L_2$ error of the result defined in the domain $\Omega^+$:

$$E_2 \equiv \sqrt{\frac{1}{N_{\Omega^+}} \sum_{\vec{x}\in\Omega^+} (\Phi - \Phi_{\text{theory}})^2}, \tag{28}$$

where $\Phi$ is the result of the solver, $\Phi_{\text{theory}}$ is the theoretical solution, and $N_{\Omega^+}$ is the number of grid points in the domain $\Omega^+$. It should be noted that since $\Phi$ is a solution of a Neumann BVP, the result is calibrated by subtracting the value at the center grid point to ensure a unique solution.

Section 5.1 presents a test case in 2D with a regular coefficient, while section 5.2 shows 2D test cases with singular coefficients. Table 1 shows our choices for the velocity potential $\Phi$, the coefficient function $\zeta$, and the density $\rho$, which defines the level set function $\varphi$. The choice of the coefficient $\zeta$ was motivated by the need to probe different singularity behaviors as $\sqrt{r}$ and $r$. For synthetic cases in sections 5.1 and 5.2, the resolution $N_{\text{grid}}$ is varied in order to observe the convergence rate of the STM, and the set of values for the resolution are selected to be even numbers ranging from $32 \times 32$ to $246 \times 246$. The grid is then set up with $(x, y) \in [-1, 1] \times [-1, 1]$, which has spacing $\Delta x = \Delta y = 2/N_{\text{grid}}$. Finally, in section 5.3, the solution of the BVP equations (10) and (11) of a realistic 3D BNS is computed and compared with the corresponding solution of the cocal code.

**Table 1.** Functions and parameters for 2D test cases. For test case 1, the modified radius $\bar{r} \equiv \sqrt{(\epsilon x)^2 + y^2}$ is used to form an elliptic domain with eccentricity $\epsilon = 1.2$, and $r_0 = \sqrt{e/4} \approx 0.824$. Test cases 2.1 and 2.2 have singular coefficient near the boundary $(\zeta, \rho \to 0)$.

| No. | $\Phi(x, y)$ | $\rho(x, y)$ | $\zeta(x, y)$ |
|---|---|---|---|
| 1 | $1 + \sin(x)\cos(y)$ | $\bar{r}^2 - r_0^2$ | $e^{r^2}$ |
| 2.1 | $y + \sin(x)$ | $r - r_0$ | $\rho$ |
| 2.2 | $y + \sin(x)$ | $\sqrt{r} - \sqrt{r_0}$ | $\rho$ |

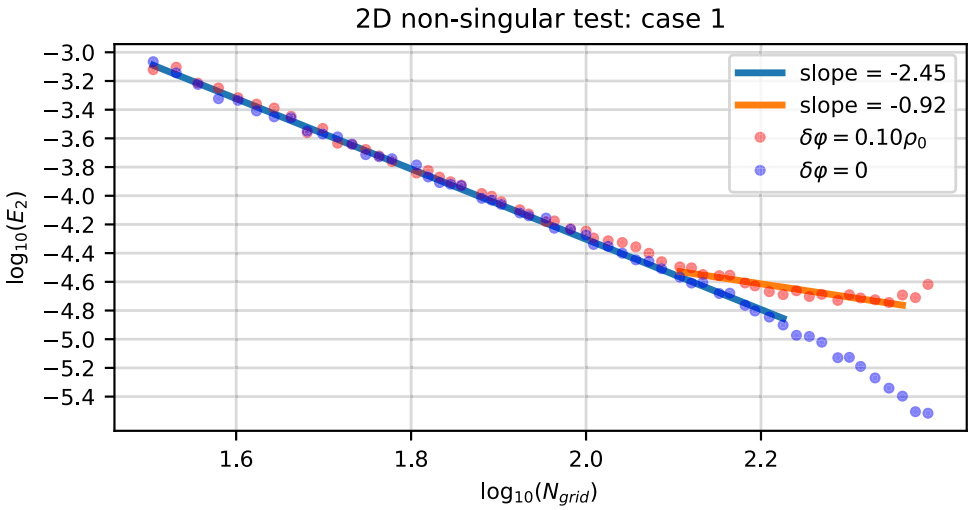


**Figure 6.** Convergence plot for 2D non-singular test case (test case 1). The plots shows the result: (a) with no separation (data points in blue), which has a convergence rate of 2.45; (b) with a separation $\delta\varphi = 0.10\rho_0$ (data points in orange).

### *5.1. Two-dimensional test with non-singular coefficient*

The result of the two-dimensional test case with non-singular coefficient (case 1 in table 1) shows an overall convergence of order $\sim 2.45$ as shown in figure 6. Since in this example we have a non-singular coefficient, the treatment of additional separation is not required here. Notwithstanding that and in order to assess its effect in convergence we made a similar experiment with separation $\delta\varphi = 0.10\rho_0$. Figure 6 shows the difference in convergence between the solver with separation $\delta\varphi = 0.10\rho_0$ and without any separation $\delta\varphi = 0$. The effect of a finite separation becomes apparent when the $N_{\text{grid}} = 128$ where the rate drops to $\sim 0.92$ until $N_{\text{grid}} = 238$. Beyond $N_{\text{grid}} = 238$ or $\Delta x \sim 0.0084$ convergence is lost. As described in section 4.3, the separation introduces an error in the boundary condition $a(\vec{x})$, thus resulting in the drop of the convergence rate.

### *5.2. Two dimensional tests with singular coefficients*

As mentioned in section 4.3, zeroes in the coefficient $\zeta$ near the boundary $\partial\Omega^+$ result in an instability. Therefore, an additional separation in the level set $\varphi = \rho - \delta\varphi$ is proposed at the cost of accuracy in the final result. In this section, such cases with singular coefficients are

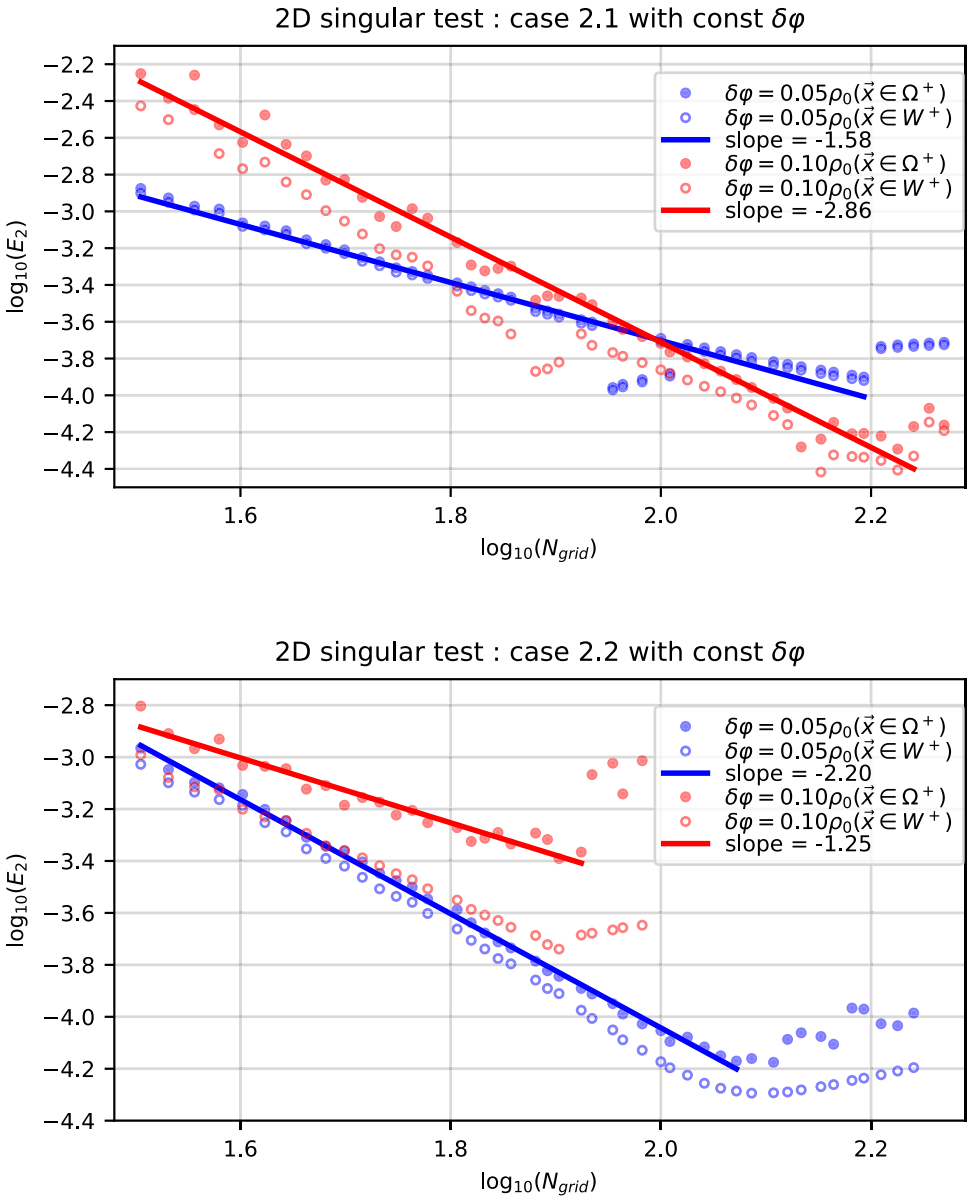


**Figure 7.** Convergence plot for 2D singular test cases with constant separation $\delta\varphi$. The top plot shows the convergence of test case 2.1; the bottom plot shows the convergence of test case 2.2. The unfilled points are the $L_2$ error of the result directly calculated from the source term method, which is only defined in $W^+$; the color-filled points are the error of the extrapolated result to the full domain $\Omega^+$. In both test cases, the solver shows convergence up to some resolution, and after that the convergence is lost and the error starts to grow for higher resolutions.

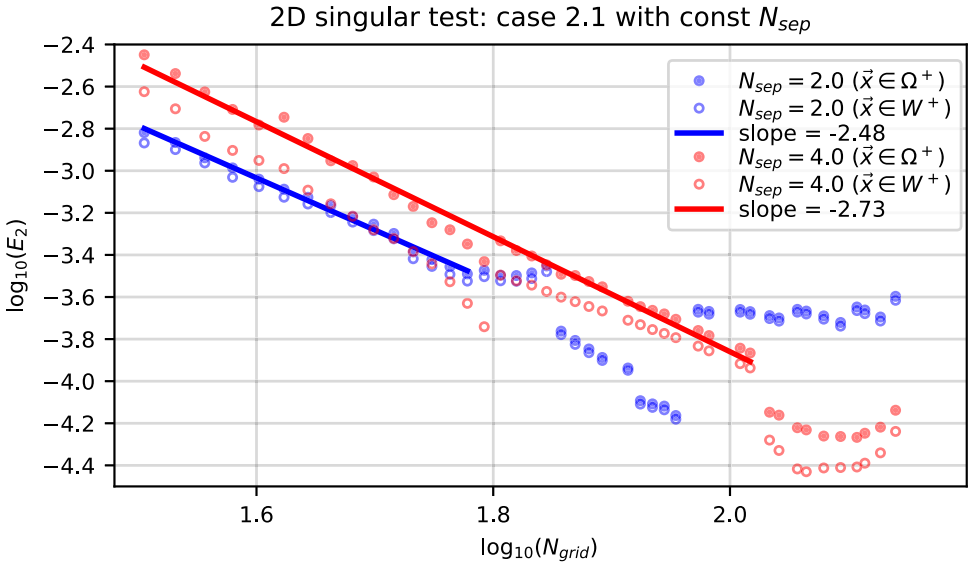


**Figure 8.** Convergence plot for 2D singular test case with constant separation number $N_{\text{sep}}$. The plot presents two sets of separation values with $\delta\varphi = 2.0\Delta x, 4.0\Delta x$. The unfilled points are the error of the result directly calculated from the source term method, which is only defined in $W^+$; the color-filled points are the error of the extrapolated result to the full domain $\Omega^+$. Similar to figure 7, the solver shows convergence up to some resolution. However, as the resolution increases from this threshold, the error stays at approximately the same value as opposed to the increasing error for the constant separation case in figure 7.

tested with various resolutions. For simpler visualization of the separation, two test cases 2.1 and 2.2 in table 1 were selected with circularly symmetric coefficients $\zeta(r) = \rho(r) = r_0^\kappa - r^\kappa$, where $\kappa = 0.5, 1.0$. Since both $\rho$ and $\varphi$ are circularly symmetric, we can define $\delta r$ to be the spatial separation between $\partial\Omega^+$ and $\partial W^+$ in any direction. Given the value for $\delta r$, one can find the separation $\delta\phi = (r_0^\kappa - (r_0 - \delta r)^\kappa)$ for test case 2.1 and 2.2. Additionally, we also define the separation number $N_{\text{sep}}$ as the number of grid points across the separation:

$$N_{\text{sep}} \equiv \frac{\delta r}{\Delta x}. \tag{29}$$

Here, two convergence tests with singular coefficients are performed: (a) constant separation $\delta\varphi$ (which implies $\delta r$ is also a constant) with both cases 2.1 and 2.2, where the convergence plots are presented in figure 7; (b) constant separation number $N_{\text{sep}}$ with case 2.1, where the convergence plot is presented in figure 8. For the singular coefficient tests with constant separation $\delta\varphi = 0.05\rho_0, 0.1\rho_0$ in figure 7, the $L_2$ error defined in $\Omega^+$ shows case 2.1 in the top panel with a convergence rate of $\sim 1.58$ up to $N_{\text{grid}} = 162$ for $\delta\varphi = 0.05\rho_0$ and a convergence rate of $\sim 2.86$ up to $N_{\text{grid}} = 180$ for $\delta\varphi = 0.1\rho_0$. In the bottom panel it shows case 2.2 with a convergence rate of $\sim 2.2$ up to $N_{\text{grid}} = 122$ for $\delta\varphi = 0.05\rho_0$ and convergence rate of $\sim 1.25$ up to $N_{\text{grid}} = 86$ for $\delta\varphi = 0.1\rho_0$.

The $L_2$ error of the singular test cases defined in the domain $\Omega^+$ (as indicated by the color-filled points in figure 7) shows similar patterns of convergence as the non-singular test. Furthermore, beyond the resolution where the error defined in $\Omega^+$ ceases to show monotonic convergence, the error increases with the resolution. Figure 7 shows that this turn-over point

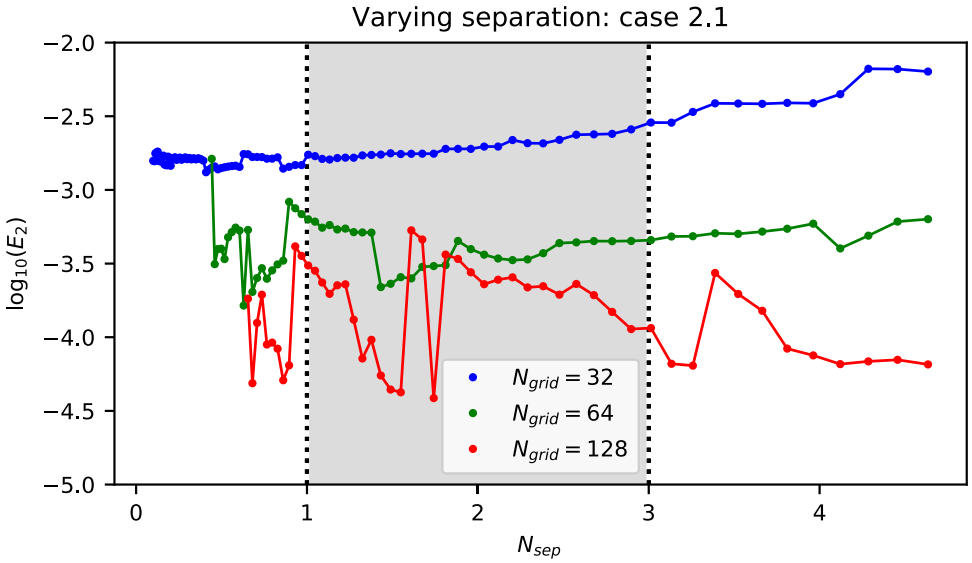


**Figure 9.** The plot shows the error for test case 2.1 at resolutions $N_{\text{grid}} = 32, 64, 128$ with various separation number $N_{\text{sep}}$ in the $x$-axis. The two vertical lines indicate the recommended upper and lower limits for $N_{\text{sep}}$: (a) a lower limit at $N_{\text{sep}} = 1.0$ where the error grows sharply for $N_{\text{sep}} < 1.0$, which is expected since $\rho \to 0$ when $N_{\text{sep}} \to 0$; (b) an upper limit at $N_{\text{sep}} = 3.0$ since the extrapolation error dominates the final error for higher separation number. Given these two issues, the recommended value for $N_{\text{sep}}$ is between 1.0 and 3.0.

occurs at smaller resolutions for larger separations. The increase in the error is incurred by the extrapolation of the result from $W^+$ to $\Omega^+$. Beyond the turn-over point, the extrapolation error dominates the total error as the resolution increases (with fixed $\delta r$, $N_{\text{sep}}$ increases when $\Delta x$ decreases, and larger $N_{\text{sep}}$ results in larger extrapolation error).

To further inspect the convergence with singular test cases, we fix the separation number $N_{\text{sep}}$ as a constant instead of the separation $\delta\varphi$ in the second set of singular tests. Here we use $N_{\text{sep}} = 2.0, 4.0$ with case 2.1, where the convergence plot is shown in figure 8. The result of the error defined in $\Omega^+$ (the color-filled points) shows a convergence rate of $\sim 2.48$ up to $N_{\text{grid}} = 62$ for $N_{\text{sep}} = 2.0$ and a convergence rate of $\sim 2.73$ up to $N_{\text{grid}} = 108$ for $N_{\text{sep}} = 4.0$. Beyond this resolution, the achieved accuracy stagnates but the solution does not show a significant increase in error, as opposed to the singular tests with constant separation.

The result in both test cases illustrates that a carefully-chosen separation $\delta\varphi$ in the level set can solve the issue of having an instability in the STM solver. To further explore the region of suitable separation values, figure 9 shows a numerical experiment with test case 2.1 at resolutions $N_{\text{grid}} = 32, 64, 128$ where the separation number is varied $N_{\text{sep}} \in [0.32, 4.82]$.

As indicated by figure 9, a suitable value for $N_{\text{sep}}$ is approximately between 1.0 and 3.0. However, for a general case where the level set is given as tabulated values $\varphi_{i,j,k}$ instead of an analytical function $\varphi(\vec{x})$, it would not be possible to analytically compute the separation $\delta\varphi$ that satisfies $1.0\Delta x < \delta r < 3.0\Delta x$. For such cases, it is recommended to take $W^+ = \Omega^+/N_2$ (note that $N_2$ here is a neighboring set of $\Omega^+$) such that two grid points are taken as the separation in all directions. As an example, table 2 provides five different test cases with singular coefficients, and the treatment $W^+ = \Omega^+/N_2$ is performed with $N_{\text{grid}} = 128$.

**Table 2.** A table of five singular coefficient test cases with $N_{\rm grid} = 128$ using a separation that gives the domain $W^+ = \Omega^+/N_2$. The $L_2$ error $E_2$, number of iterations, and the separation value are shown for each test. For test case 3.4, the modified radius $\bar{r} \equiv \sqrt{(\epsilon x)^2 + y^2}$ is used to form an elliptic domain, where the eccentricity $\epsilon = 1.2$; the radius of the domain $r_0 = \sqrt{e/4} \approx 0.824$.

| No. | $\Phi_{\rm theory}$ | $\rho$ | $\zeta$ | $E_2$ | Iterations | $\delta\varphi$ |
|---|---|---|---|---|---|---|
| 3.1 | $y + \sin x$ | $r_0^2 - r^2$ | $\rho$ | $1.011 \times 10^{-4}$ | 80 | $0.0728\rho_0$ |
| 3.2 | $y + \sin x$ | $r_0 - r$ | $\rho$ | $2.885 \times 10^{-4}$ | 73 | $0.0371\rho_0$ |
| 3.3 | $y + \sin x$ | $\sqrt{r_0} - \sqrt{r}$ | $\rho$ | $3.086 \times 10^{-4}$ | 76 | $0.0187\rho_0$ |
| 3.4 | $1 + \cos x \sin y$ | $r_0^2 - \bar{r}^2$ | $\rho$ | $2.925 \times 10^{-5}$ | 72 | $0.0874\rho_0$ |
| 3.5 | $y + \sin x$ | $r_0 - r$ | $\rho e^r$ | $3.251 \times 10^{-4}$ | 67 | $0.0371\rho_0$ |

These two test cases show that the STM is able to solve 2D Poisson equations that converges with resolution. In general, non-singular cases would show second order convergence since a second-order finite difference Jacobi method is used as the Poisson solver. The non-singular test shown in figure 6 exhibits symmetry in $x$ and $y$ axes, resulting in the super-convergence that has a convergence rate of 2.5. For singular cases, there are multiple contributions of the error other than the Poisson solver, such as the additional separation and extrapolation, and coefficient singularities. Thus, it is difficult to predict the theoretical convergence rate for singular test cases.

### 5.3. Irrotational binary neutron stars

This section presents a realistic irrotational BNS test case. The BNS system is set up with two equal mass NSs with a polytropic equation of state $P(\rho) = K\rho^\Gamma$ where $\Gamma = 2$, $K = 123.6$, and central density $\rho_0 = 9.574 \times 10^{-4}$ ($5.905 \times 10^{14}$ g cm$^{-3}$). The two stars orbit around each other with an angular orbital velocity $\Omega = 9.075 \times 10^{-3}$ ($1.842 \times 10^3$ rad s$^{-1}$) at a distance $D_0 = 30.47$ (45 km). The parameters listed above are in geometric units.

The initial condition for the STM solver is generated using the cocal solver [28] [3] in a $301^3$ grid, in which the $60^3$ grid at the center that encapsulates the star is taken as the input for the test case. This includes conformal factor $\psi$, lapse $\alpha$, shift $\beta^i$, density $\rho$, specific enthalpy $h$, and the time component of the four-velocity $u^t$.

The treatment for the singularity of the coefficients near the surface uses $W^+ = \Omega^+/N_2$, as described in the previous section. In other words, $\delta\varphi = \min(\varphi(\vec{x} \in W^+))$. Figure 10 shows the comparison between the BNS velocity potential from the STM and from cocal. Figure 11 shows a comparison of the velocity components, which are the most relevant in the BNS equations since the potential $\Phi$ never appears in them by itself but only as a gradient. The maximum difference obtained in the velocity potential from figure 10 is $\sim 7.7 \times 10^{-4}$. The maximum relative difference obtained in the velocity potential is $\sim 1.33 \times 10^{-2}$ and appears at the center of the grid. Given the fact that the cocal code uses spherical coordinates centered at this point its computational error is larger there [39]. On the other hand, our computational grid for the STM is Cartesian with no coordinate singularity at the center of the NSs. Therefore, we expect our solution to have better accuracy at the points close to the center. Overall, this realistic BNS test case demonstrates that the STM is able to reproduce the velocity potential of the cocal with maximum relative difference no larger than 1.4%.

[3] We note here that in the cocal solver fluid surface fitted coordinates are implemented in order to solve for the fluid potential.

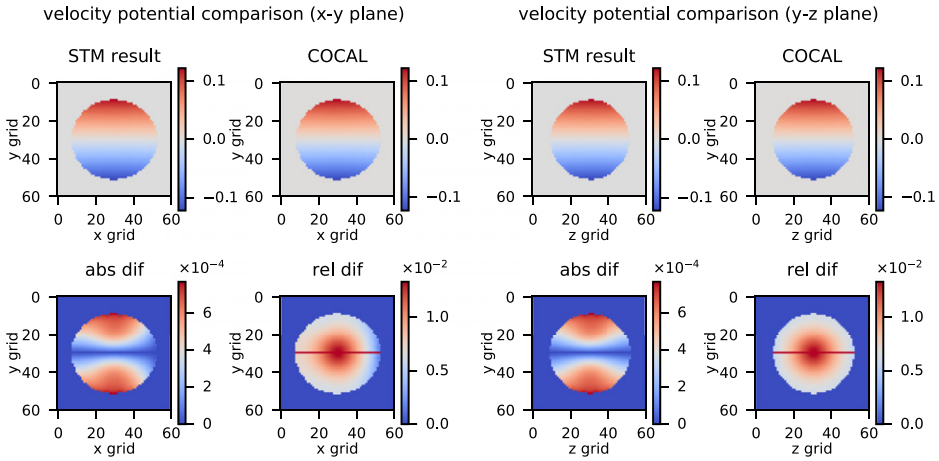


**Figure 10.** Velocity potential slices in the $x$–$y$ plane (the left panel) and the $y$–$z$ plane (the right panel) of the irrotational BNS test case. The two sets of subplots both show the velocity potential $\Phi$ of the STM (top left), the velocity potential $\Phi_{\rm COCAL}$ generated by cocal (top right), the absolute difference (bottom left) and the relative difference (bottom right) between the two solvers. Regions where the relative difference blows-up have $\Phi_{\rm COCAL} \approx 0$.

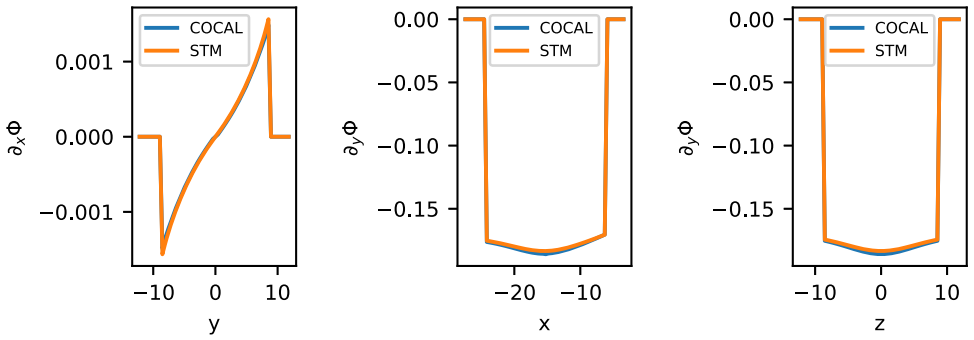


**Figure 11.** Comparison plots of the gradient of the velocity potential. The left panel shows $\partial_x\Phi$ along the $y$-axis; the middle panel shows $\partial_y\Phi$ along the $x$-axis; the right panel shows $\partial_y\Phi$ along the $z$-axis. It is evident from the plot that the derivative of the velocity potential from the STM (curves in blue) match well with that from the cocal solver (curves in orange).

## 6. Summary

To obtain initial data for an irrotational or spinning BNS system, one must solve the conservation of rest-mass and the Euler equation for a velocity potential. This process results in a Poisson equation with Neumann-type boundary condition on the surface of the stars, and this BVP calls for a robust Poisson solver on an unknown domain. In this work we apply the STM [29] to solve this type of equation in Cartesian coordinates, with the boundary conditions incorporated as jump conditions on the nonlinear sources.

A problem arises for NSs (but not with quark stars) near the surface as some of the coefficients are approaching zero, causing a divide-by-zero singularity in the source terms. We proposed a solution that introduces a separation $\delta\varphi$ that shifts the level set $\varphi = \rho - \delta\varphi$ to get a smaller domain $W^+$ within the domain of interest $\Omega^+$ such that the coefficients are non-singular inside $W^+$. The solution is then extrapolated from $W^+$ to $\Omega^+$. Given an aptly-chosen separation, the error of the extrapolated solution converges as resolution increases up to some threshold resolution.

Using the STM, we have implemented a separate elliptic solver in Cartesian coordinates and solved the BNS fluid equation for the velocity potential, while keeping the gravitational potentials and other thermodynamic variables fixed. We then compared this solution with the one coming out from the cocal binary solver and found agreement to the level of $\sim 1\%$.

One advantage of our method is its simplicity. It does not require complex surface fitted coordinates and can be implemented in a Cartesian grid which is a standard choice in numerical relativity calculations. Another advantage is that now the fluid Poisson equation can be solved similarly (for example using the Green's function approach [28]) to the gravitational equations simplifying the overall iteration scheme. Finally, we expect that our method can be used in other problems where non-smooth sources exist, as for example in magnetized rotating NSs.

## Acknowledgments

This work is funded by the Students Pushing Innovation (SPIN) program in National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign. RH gratefully acknowledges support through NSF Grants OAC-2004879, OAC-1550514, and ACI-1238993. AT is supported by NSF Grants No. PHY-1662211 and No. PHY-2006066, and NASA Grant No. 80NSSC17K0070 to the University of Illinois at Urbana-Champaign. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

## Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: 10.5281/zenodo.4515020.

## Appendix A

### *A.1. The derivation of the source term method*

Here we present the derivation, as presented in [38], of the source term equation (17) that we used to solve the fluid BVP.

Consider the domain of figure 1 and the BVP of equation (14). Integrals over $\Omega^+$ can be converted to integrals over the whole domain $\Omega$ by writing

$$\int_{\Omega^+} f \, \mathrm{d}V = \int_{\Omega} f H(\varphi) \mathrm{d}V, \tag{30}$$

where $H(z)$ is the Heaviside function

$$H(z) = \begin{cases} 1, & z \geqslant 0, \\ 0, & z < 0. \end{cases}$$

Similarly one can convert integrals over the boundary $\partial\Omega^+$ into volume integrals over $\Omega$ as follows

$$\int_{\partial\Omega^+} f \, \mathrm{d}S = \int_\Omega f \delta(\varphi) |\nabla\varphi| \, \mathrm{d}V, \tag{31}$$

where $\delta(\varphi) = H'(\varphi)$ is the Dirac delta function.

For any sufficiently smooth function $\psi$ one can write

$$\int_\Omega \psi \nabla^2 \Phi \, \mathrm{d}V = \int_\Omega \Phi \nabla^2 \psi \, \mathrm{d}V + \int_{\partial\Omega} \left( \psi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \psi}{\partial n} \right) \mathrm{d}S. \tag{32}$$

We will apply the equation above separately in the domains $\Omega^+$ and $\Omega^-$ and then combine them in order to get an equation in the whole domain $\Omega$. We note that for the $\Omega^+$ case there is only one boundary $\partial\Omega^+$ while for the $\Omega^-$ case we have two boundary terms on $\partial\Omega^+$ and $\partial\Omega$. We also assume that the test function $\psi$ as well as its derivatives are continuous across the boundary $\partial\Omega^+$. Using equations (30) and (31) and denoting $S(\vec{x}) = S^+ H(\varphi(\vec{x})) + S^-(1 - H(\varphi(\vec{x})))$ we obtain

$$\begin{aligned} \int_\Omega \psi S \, \mathrm{d}V = & \int_\Omega \Phi \nabla^2 \psi \, \mathrm{d}V + \int_{\partial\Omega} \left( \psi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \psi}{\partial n} \right) \mathrm{d}S \\ & + \int_\Omega \left( \psi a - b \frac{\partial \psi}{\partial n} \right) \delta(\varphi) |\nabla\varphi| \, \mathrm{d}V, \end{aligned} \tag{33}$$

where $a = \Phi_n^+ - \Phi_n^-$ and $b = \Phi^+ - \Phi^-$. We remove the normal derivative on $\psi$ from the last term above by using integration by parts. It is

$$\begin{aligned} \int_\Omega \psi b \nabla^2 H(\varphi) \mathrm{d}V &= -\int_\Omega (b \nabla\psi + \psi \nabla b) \cdot \nabla H(\varphi) \mathrm{d}V \\ &= \int_\Omega \left( b \frac{\partial \psi}{\partial n} + \psi \frac{\partial b}{\partial n} \right) \delta(\varphi) |\nabla\varphi| \, \mathrm{d}V, \end{aligned} \tag{34}$$

since the integral over $\partial\Omega$ vanishes there. Using equation (34) we write equation (33) as

$$\begin{aligned} & \int_\Omega \Phi \nabla^2 \psi \, \mathrm{d}V + \int_{\partial\Omega} \left( \psi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \psi}{\partial n} \right) \mathrm{d}S \\ & \quad = \int_\Omega \psi \left( b \nabla^2 H - \left( a + \frac{\partial b}{\partial n} \right) |\nabla\varphi| \delta(\varphi) + S \right) \mathrm{d}V \\ & \quad = \int_\Omega \psi \left( \nabla^2 (bH) - H \nabla^2 b - \left( a - \frac{\partial b}{\partial n} \right) |\nabla\varphi| \delta(\varphi) + S \right) \mathrm{d}V. \end{aligned} \tag{35}$$

Equations (32) and (35) imply the source term equation (17) in the weak sense.

### *A.2. Finite difference discretization of the Heaviside and delta function*

This section discusses the finite difference discretization of the Heaviside and delta function in equations (23) and (24), as demonstrated in section 4 of [38]. The discretization of the Heaviside function is derived in detail in [37], and the delta function in [40]. The goal of the discretization of both functions is to better approximate the integrals

$$\mathcal{I}_{\mathcal{H}} = \int_{\Omega} H(\vec{x})\mathrm{d}V \qquad \mathcal{I}_{\delta} = \int_{\Omega} \delta(\vec{x})\mathrm{d}V. \tag{36}$$

Using $I(\varphi)$, $J(\varphi)$, $K(\varphi)$ defined in equation (22), we can write

$$\nabla^2 I(\varphi) = \nabla \cdot (H(\varphi)\nabla\varphi) = \delta(\varphi)|\nabla\varphi|^2 + H(\varphi)\nabla^2\varphi, \tag{37}$$

$$\nabla^2 J(\varphi) = \nabla \cdot (I(\varphi)\nabla\varphi) = H(\varphi)|\nabla\varphi|^2 + I(\varphi)\nabla^2\varphi, \tag{38}$$

$$\nabla^2 K(\varphi) = \nabla \cdot (J(\varphi)\nabla\varphi) = I(\varphi)|\nabla\varphi|^2 + J(\varphi)\nabla^2\varphi. \tag{39}$$

Equation (37) gives the formula for $\delta(\varphi)$ by substituting $H(\varphi)$ from equation (38)

$$\delta(\varphi) = \frac{\nabla^2 I}{|\nabla\varphi|^2} - \frac{H(\varphi)\nabla^2\varphi}{|\nabla\varphi|^2} = \frac{\nabla^2 I}{|\nabla\varphi|^2} - \frac{(\nabla^2 J - I\nabla^2\varphi)\nabla^2\varphi}{|\nabla\varphi|^4}. \tag{40}$$

Similarly, equation (38) gives the formula for $H(\varphi)$ by substituting $I(\varphi)$ from equation (39):

$$H(\varphi) = \frac{\nabla^2 J}{|\nabla\varphi|^2} - \frac{I(\varphi)\nabla^2\varphi}{|\nabla\varphi|^2} = \frac{\nabla^2 J}{|\nabla\varphi|^2} - \frac{(\nabla^2 K - J\nabla^2\varphi)\nabla^2\varphi}{|\nabla\varphi|^4}. \tag{41}$$

The expressions for $\delta(\varphi(\vec{x}))$ and $H(\varphi(\vec{x}))$ give the discretization formulas for both functions in equations (24) and (23).

## ORCID iDs

Bing-Jyun Tsao https://orcid.org/0000-0003-4614-0378
Roland Haas https://orcid.org/0000-0003-1424-6178
Antonios Tsokaros https://orcid.org/0000-0003-2242-8924

## References

[1] Abbott B P *et al* 2017 *Phys. Rev. Lett.* **119** 161101
[2] von Kienlin A, Meegan C and Goldstein A 2017 GRB 170817A: Fermi GBM detection. *GRB Coordinates Network* **21520** 1
[3] Kozlova A *et al* 2017 IPN Triangulation of GRB 170816A (short/hard) *GRB Coordinates Network* **21517** 1
[4] Savchenko V *et al* 2017 *Astrophys. J.* **848** L15 (arXiv:1710.05449)
[5] Savchenko V *et al* 2017 *LIGO/Virgo G298048: INTEGRAL detection of a prompt gamma-ray counterpart* 21507 The LIGO Scientific Collaboration and the Virgo Collaboration
[6] Abbott B P *et al* (Virgo) (LIGO Scientific) 2017 *Phys. Rev. Lett.* **119** 161101 (arXiv:1710.05832)
[7] Abbott B *et al* (LIGO Scientific 2017 *Astrophys. J. Lett.* **848** L12 (arXiv:1710.05833)
[8] Abbott B P *et al* (Virgo) (Fermi-GBM) (INTEGRAL) (LIGO Scientific) 2017 *Astrophys. J.* **848** L13 (arXiv:1710.05834)
[9] Chornock R *et al* 2017 *Astrophys. J.* **848** L19 (arXiv:1710.05454)
[10] Baumgarte T W, Cook G B, Scheel M A, Shapiro S L and Teukolsky S A 1997 *Phys. Rev. Lett.* **79** 1182–5

[11] Baumgarte T W, Cook G B, Scheel M A, Shapiro S L and Teukolsky S A 1998 *Phys. Rev.* D **57** 7299–311
[12] Marronetti P, Mathews G J and Wilson J R 1998 *Phys. Rev.* D **58** 107503
[13] Bildsten L and Cutler C 1992 *Astrophys. J.* **400** 175
[14] Kochanek C S 1992 *Astrophys. J.* **398** 234
[15] Bonazzola S, Gourgoulhon E and Marck J-A 1999 *Phys. Rev. Lett.* **82** 892–5
[16] Gourgoulhon E, Grandclément P, Taniguchi K, Marck J A and Bonazzola S 2001 *Phys. Rev.* D **63** 064029
[17] Marronetti P, Mathews G J and Wilson J R 2000 *Nucl. Phys.* B **80** 07 (arXiv:gr-qc/9903105)
[18] Marronetti P, Mathews G J and Wilson J R 1999 *Phys. Rev.* D **60** 087301
[19] Uryū K and Eriguchi Y 2000 *Phys. Rev.* D **61** 124023
[20] Uryū K, Shibata M and Eriguchi Y 2000 *Phys. Rev.* D **62** 104015
[21] Marronetti P and Shapiro S L 2003 *Phys. Rev.* D **68** 104024
[22] Baumgarte T W and Shapiro S L 2009 *Phys. Rev.* D **80** 064009
[23] Baumgarte T W and Shapiro S L 2009 *Phys. Rev.* D **80** 089901
[24] Tsatsin P and Marronetti P 2013 *Phys. Rev.* D **88** 064060
[25] Tichy W 2011 *Phys. Rev.* D **84** 024041
[26] Tichy W 2012 *Phys. Rev.* D **86** 064024
[27] Tichy W, Rashti A, Dietrich T, Dudi R and Brügmann B 2019 *Phys. Rev.* D **100** 124046
[28] Tsokaros A, Uryū K and Rezzolla L 2015 *Phys. Rev.* D **91** 104030
[29] Towers J D 2018 *J. Comput. Phys.* **361** 424–41
[30] Teukolsky S A 1998 *Astrophys. J.* **504** 442–9
[31] Shibata M 1998 *Phys. Rev.* D **58** 024012
[32] Tsokaros A, Uryu K, Ruiz M and Shapiro S L 2018 *Phys. Rev.* D **98** 124019
[33] Uryu K and Tsokaros A 2012 *Phys. Rev.* D **85** 064014
[34] Laboratoire de l'Univers et ses Théories, Observatoire de Paris 2020 LORENE Langage Objet pour la RElativité NumériquE http://lorene.obspm.fr
[35] Tichy W 2009 *Class. Quantum Grav.* **26** 175018
[36] Foucart F, Kidder L E, Pfeiffer H P and Teukolsky S A 2008 *Phys. Rev.* D **77** 124051
[37] Towers J D 2009 *J. Comput. Phys.* **228** 3478–89
[38] Towers J D 2010 *Nonlinear Partial Differential Equations and Hyperbolic Wave Phenomena* pp 360–89
[39] Tsokaros A, Mundim B C, Galeazzi F, Rezzolla L and Uryū K 2016 *Phys. Rev.* D **94** 044049
[40] Towers J D 2007 *J. Comput. Phys.* **220** 915–31
[41] Osher S and Fedkiw R 2003 *Level Set Methods and Dynamic Implicit Surfaces* (New York: Springer)