

PDE-Based Multidimensional Extrapolation of Scalar Fields over Interfaces with Kinks and High Curvatures

Daniil Bochkov, Frederic Gibou

May 4, 2020

Abstract

We present a PDE-based approach for the multidimensional extrapolation of smooth scalar quantities across interfaces with kinks and regions of high curvature. Unlike the commonly used method of [2], in which normal derivatives are extrapolated, the proposed approach is based on the extrapolation and weighting of Cartesian derivatives. As a result, second- and third-order accurate extensions in the L^∞ norm are obtained with linear and quadratic extrapolations, respectively, even in the presence of sharp geometric features. The accuracy of the method is demonstrated on a number of examples in two and three spatial dimensions and compared to the approach of [2]. The importance of accurate extrapolation near sharp geometric features is highlighted on an example of solving the diffusion equation on evolving domains.

1 Introduction

Extrapolation procedures are ubiquitous in scientific computing and generally allow one to estimate a valid value of a quantity at points where data is not given; either in space or in time. In the context of level-set methods [30], extrapolation procedures in space have been frequently used since the advent of the ghost-fluid method [11], where constant extrapolations were originally used. Generalized ghost-fluid methods were then designed, in part based on higher-order extrapolations for which Aslam introduced a partial differential equation (PDE) approach to perform linear and quadratic extrapolation [2] and Gibou and Fedkiw introduced a cubic extrapolation in the same PDE framework [13]. It is natural in the level-set context to perform such extrapolations using PDE formulations for their solutions are based on Hamilton-Jacobi solvers that have been designed for other standard level-set equations, see e.g. [39]. A typical situation that needs extrapolation is that of an implicit treatment of a field in a free boundary problem. In this case, a valid value of the field at time t^n needs to be known when assembling the right-hand side of the linear system of equations at time t^{n+1} . Since the interface at the new time step has swept grid points that are outside the domain at the previous time step, valid values of the field at time t^n are needed in the domain at time t^{n+1} , which requires an extrapolation procedure.

Typical use of extrapolation methods can be found in a multitude of level-set applications including multiphase flow simulations [7, 12, 17, 20, 21, 28, 34], in the solution of Poisson-Boltzmann [19, 25] and Poisson-Nernst-Planck equations [23] for studying transport in ionic solutions, in heat and diffusion flow problems [4, 13, 15, 32], in the study of epitaxial growth and diblock-copolymer self-assembly used in the semi-conductor industry [31, 33], in shape optimization [1, 40], surface reconstruction of biomolecules [10, 25] and in Stefan-type problems [6, 14, 26, 35]. PDE-based extrapolation procedures have also been extended to adaptive Quad-/Oc-tree grids and parallel architectures [18, 22, 24]. In addition, fast methods have been introduced for computationally efficient extrapolation procedures using the Fast Marching method, including parallel implementations [5, 37, 38] or the Fast Sweeping method [3, 42],

including efficient parallel algorithms on adaptive grids [8, 9]. We also refer the interested reader to [27] for another implicit approach to extrapolation based on solving the biharmonic equation.

However, those methods behave poorly in the case where the free boundary presents high-curvature features or kinks. Typical examples of such situations are multimaterial flows with triple junction points, motion of sharp-edged bodies in fluids, contact line dynamics in wetting phenomena, phase-change front propagation in the presence of confining walls, etc. We introduce a method that solves that problem. We present the method in section 2 and numerical examples in sections 3 and 4 that illustrate its benefits and comment on its efficiency. Section 5 considers an example of solving a diffusion equation on evolving domains that demonstrates the importance of accurate extrapolation near sharp geometric features. Section 6 draws some conclusions.

2 Numerical Method

2.1 Level-set Representation

The level set representation [30] defines the interface of a domain by $\{\mathbf{x} : \phi(\mathbf{x}) = 0\}$, its interior and exterior by $\phi(\mathbf{x}) < 0$ and $\phi(\mathbf{x}) > 0$, respectively, where $\phi(\mathbf{x})$ is a Lipschitz continuous function called *the level-set function*. In this paper, the only geometrical quantity that is needed is the outward normal to the interface, \mathbf{n} , which can be computed as:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \quad (1)$$

using central differencing for ϕ_x and ϕ_y . In typical level-set simulations, the level-set function is reinitialized as a signed distance function [39]. We refer the interested reader to [29, 36] for a thorough presentation of the level-set method and [16] for a recent review.

2.2 Normal-derivative based multidimensional PDE extrapolation of [2]

High order extrapolations in the normal direction are traditionally performed in a series of steps, as proposed by Aslam in [2] and referred to in the present manuscript as the *normal-derivative based partial differential equation (ND-PDE) extrapolation*. For example, suppose that we seek to extrapolate a scalar field q from the region where $\phi \leq 0$ to the region where $\phi > 0$. In the case of a quadratic extrapolation, we first compute $q_{nn} = \nabla(\nabla q \cdot \mathbf{n}) \cdot \mathbf{n}$ in the region $\phi \leq 0$ and extrapolate it across the interface in a constant fashion, that is, such that its normal derivative is zero in the region $\phi > 0$, by solving the following partial differential equation:

$$\frac{\partial q_{nn}}{\partial \tau} + H(\phi) (\mathbf{n} \cdot \nabla q_{nn}) = 0, \quad (2)$$

where H is the Heaviside function. Then, the value of q across the interface is found by solving the following two partial differential equations:

$$\frac{\partial q_n}{\partial \tau} + H(\phi) (\mathbf{n} \cdot \nabla q_n - q_{nn}) = 0, \quad (3)$$

$$\frac{\partial q}{\partial \tau} + H(\phi) (\mathbf{n} \cdot \nabla q - q_n) = 0, \quad (4)$$

defining q_n in such a way that its normal derivative is equal to the previously extrapolated q_{nn} and then defining q in such a way that its normal derivative is equal to the previously extrapolated q_n . These PDEs are solved in fictitious time τ for a few iterations (typically 15) since we only seek to extrapolate the values of q in a narrow band of a few grid cells around the interface.

This extrapolation procedure produces accurate results in the case where the interface is smooth, but generates large error in the case where sharp geometric features occur, e.g. thin elongated shapes or interfaces with kinks as illustrated in sections 3 and 4.

2.3 Weighted-Cartesian-derivative based multidimensional PDE extrapolation

Instead of calculating the normal derivatives in the negative region before extrapolating them, we instead compute the derivatives in the Cartesian directions, extrapolate them and then construct the normal derivatives. Specifically, consider the following quantities, that are computed in the negative level-set region:

$$\mathbf{q}_\nabla = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \text{ and the symmetric matrix } \mathbf{Q}_{\nabla\nabla} = \begin{pmatrix} q_{xx} & q_{xy} & q_{xz} \\ q_{xy} & q_{yy} & q_{yz} \\ q_{xz} & q_{zy} & q_{zz} \end{pmatrix}.$$

Similar to the method described in the previous section, we extrapolate the elements of $\mathbf{Q}_{\nabla\nabla}$ in a constant fashion:

$$\frac{\partial \mathbf{Q}_{\nabla\nabla}}{\partial \tau} + H(\phi) (\mathbf{n} \cdot \nabla \mathbf{Q}_{\nabla\nabla}) = 0, \quad (5)$$

before successively solving the following equations:

$$\frac{\partial \mathbf{q}_\nabla}{\partial \tau} + H(\phi) (\mathbf{n} \cdot (\nabla \mathbf{q}_\nabla - \mathbf{Q}_{\nabla\nabla})) = 0, \quad (6)$$

$$\frac{\partial q}{\partial \tau} + H(\phi) (\mathbf{n} \cdot (\nabla q - \mathbf{q}_\nabla)) = 0. \quad (7)$$

Note that now, the normal vector field \mathbf{n} enters the equations merely as some sort of weighting factor. Thus, as long as field q is sufficiently smooth this approach to multidimensional extrapolation is expected to produce accurate results even when the normal vector field \mathbf{n} is not smooth (as is the case of domains with sharp features). To distinguish the proposed approach from the one in [2], we refer to it as the *weighted-Cartesian-derivative based partial differential equation (WCD-PDE) extrapolation*.

Remark: It is possible to construct cubic and even higher-order extrapolations following this approach as well, however one needs to keep in mind the rapidly growing computational cost, because an m -th order method requires solving advection equations for tensor variables of order up to m ($3 \times 3 \times 3$ for cubic, $3 \times 3 \times 3 \times 3$ for quartic, etc).

2.4 Implementation details

In this work we demonstrate the proposed method on uniform Cartesian grids and our implementation follows very closely the one from [2] with just few differences. Consider a two dimensional computational grid with nodes defined as:

$$\mathbf{r}_{i,j} = \begin{pmatrix} x_{\min} + (i-1)\Delta x \\ y_{\min} + (j-1)\Delta y \end{pmatrix}, \quad i \in [1; N_x], j \in [1; N_y], \quad \Delta x = \frac{x_{\max} - x_{\min}}{N_x - 1}, \quad \Delta y = \frac{y_{\max} - y_{\min}}{N_y - 1},$$

where $[x_{\min}; x_{\max}] \times [y_{\min}; y_{\max}]$ denotes the computational domain, N_x and N_y are number of grid nodes in the Cartesian directions. Standard second-order accurate central difference formulas are used for calculating the normal vector field $\mathbf{n}(\mathbf{r})$ (in the entire domain) and derivatives (first and second) of q in the negative region. Normal derivatives of q are computed as:

$$q_n = \nabla q \cdot \mathbf{n} \quad \text{and} \quad q_{nn} = \mathbf{n} \cdot \nabla \nabla q \cdot \mathbf{n} + \mathbf{n} \cdot \nabla \mathbf{n} \cdot \nabla q.$$

Since the first and second order derivatives of q are not well-defined at all grid points where $\phi < 0$ we replace the Heaviside function $\mathcal{H}(\phi)$ in equations (3), (6) and in equations (2), (5) with discrete fields $\mathcal{H}^{\phi, \nabla}$ and $\mathcal{H}^{\phi, \nabla \nabla}$, respectively, where:

$$\begin{aligned}\mathcal{H}_{i,j}^{\phi, \nabla} &= \begin{cases} 0, & \text{if } \phi \leq 0 \text{ at } \mathbf{r}_{i\pm 1,j}, \mathbf{r}_{i,j\pm 1}, \\ 1, & \text{otherwise,} \end{cases} \\ \mathcal{H}_{i,j}^{\phi, \nabla \nabla} &= \begin{cases} 0, & \text{if } \phi \leq 0 \text{ at } \mathbf{r}_{i\pm 1,j}, \mathbf{r}_{i,j\pm 1}, \mathbf{r}_{i\pm 1,j\pm 1}, \mathbf{r}_{i\pm 1,j\mp 1} \\ 1, & \text{otherwise.} \end{cases}\end{aligned}$$

Applying an explicit first-order accurate in time discretization to equations (2)-(7) one obtains the following updating formulas:

$$\begin{aligned}[qnn]_{i,j}^{k+1} &= [qnn]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi, \nabla \nabla} \left([\mathbf{n} \cdot \nabla qnn]_{i,j}^k \right), \\ [q\mathbf{n}]_{i,j}^{k+1} &= [q\mathbf{n}]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi, \nabla} \left([\mathbf{n} \cdot \nabla q\mathbf{n}]_{i,j}^k - [qnn]_{i,j}^k \right), \\ [q]_{i,j}^{k+1} &= [q]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi} \left([\mathbf{n} \cdot \nabla q]_{i,j}^k - [q\mathbf{n}]_{i,j}^k \right),\end{aligned}\tag{8}$$

and

$$\begin{aligned}[\mathbf{Q}_{\nabla \nabla}]_{i,j}^{k+1} &= [\mathbf{Q}_{\nabla \nabla}]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi, \nabla \nabla} \left([\mathbf{n} \cdot \nabla \mathbf{Q}_{\nabla \nabla}]_{i,j}^k \right), \\ [q_{\nabla}]_{i,j}^{k+1} &= [q_{\nabla}]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi, \nabla} \left([\mathbf{n} \cdot \nabla q_{\nabla}]_{i,j}^k - [\mathbf{n} \cdot \mathbf{Q}_{\nabla \nabla}]_{i,j}^k \right), \\ [q]_{i,j}^{k+1} &= [q]_{i,j}^k - \Delta\tau \mathcal{H}_{i,j}^{\phi} \left([\mathbf{n} \cdot \nabla q]_{i,j}^k - [\mathbf{n} \cdot q_{\nabla}]_{i,j}^k \right),\end{aligned}\tag{9}$$

When extrapolating first- and second-order derivatives (i.e. $q\mathbf{n}$, qnn , q_{∇} and $\mathbf{Q}_{\nabla \nabla}$), first-order spatial derivatives in the equations above are computed using first-order accurate upwind discretizations. For example, derivatives in the x -direction are approximated as:

$$[n_x \partial_x f]_{i,j}^k = \begin{cases} [n_x]_{i,j} \frac{[f]_{i,j}^k - [f]_{i-1,j}^k}{\Delta x} + \mathcal{O}(\Delta x), & \text{if } [n_x]_{i,j} > 0, \\ [n_x]_{i,j} \frac{[f]_{i+1,j}^k - [f]_{i,j}^k}{\Delta x} + \mathcal{O}(\Delta x), & \text{if } [n_x]_{i,j} < 0, \end{cases}\tag{10}$$

where f is $q\mathbf{n}$, qnn , q_{∇} or $\mathbf{Q}_{\nabla \nabla}$. This is sufficient to achieve second-order accuracy in the extended fields $q\mathbf{n}$ and q_{∇} . For extrapolation of the field q itself (last equations in (8) and (9)), however, second-order accurate upwind discretizations are used. For example, derivatives in the x -direction are approximated as:

$$[n_x \partial_x q]_{i,j} = \begin{cases} [n_x]_{i,j} \left(\frac{[q]_{i,j} - [q]_{i-1,j}}{\Delta x} + \frac{\Delta x}{2} \text{minmod}([q_{xx}]_{i,j}, [q_{xx}]_{i-1,j}) \right) + \mathcal{O}(\Delta x^2), & \text{if } [n_x]_{i,j} > 0, \\ [n_x]_{i,j} \left(\frac{[q]_{i+1,j} - [q]_{i,j}}{\Delta x} - \frac{\Delta x}{2} \text{minmod}([q_{xx}]_{i,j}, [q_{xx}]_{i+1,j}) \right) + \mathcal{O}(\Delta x^2), & \text{if } [n_x]_{i,j} < 0, \end{cases}\tag{11}$$

where

$$\text{minmod}(a, b) = \begin{cases} 0, & \text{if } ab \leq 0, \\ a, & \text{if } |a| \leq |b|, \\ b, & \text{if } |b| \leq |a|. \end{cases}$$

Derivatives in the y -directions are approximated in a similar fashion. We note that approximation of derivatives is done in the same way for both, ND-PDE and WCD-PDE, extrapolation methods. The difference between the approaches lies in which quantities are extended over interfaces.

Since in the new method the approximation of second-order derivatives in all Cartesian directions are already available during solving the PDE for q , the minmod corrections in (11) can be computed only once during first iteration and reused in subsequent iterations, reducing the cost of each iteration by approximately 2 times. Specifically, the total count of arithmetic operations to compute $[q]_{i,j}^{k+1}$ using the ND-PDE method is approximately 22 in two spatial dimensions and 32 in three spatial dimensions, while for the WCD-PDE method the total count is 10 and 14, correspondingly. Thus, if we denote as T the cost of solving a single advection equation using first-order accurate approximations of derivatives, then the total cost of performing quadratic extrapolation using the ND-PDE method is approximately $(1 + 1 + 2)T = 4T$ in two and three spatial dimensions, while the total cost of performing quadratic extrapolation using the WCD-PDE method is $(3 + 2 + 1)T = 5T$ in two spatial dimensions and $(6 + 3 + 1)T = 10T$ in three spatial dimensions.

Equations (8) and (9) are iterated in the fictitious time τ until steady-state. Time step $\Delta\tau$ is chosen based on consideration of satisfying the CFL condition as:

$$\Delta\tau = \frac{\min(\Delta x, \Delta y)}{2} \quad \text{and} \quad \Delta\tau = \frac{\min(\Delta x, \Delta y, \Delta z)}{3}$$

in two and three spatial dimensions, respectively. Iterations are terminated when the maximum difference between two successive steps $\max_{i,j} |[f]_{i,j}^{k+1} - [f]_{i,j}^k|$ within the band of interest, that is, among all grid nodes within the distance of $2\sqrt{\Delta x^2 + \Delta y^2}$ (or $2\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ in three spatial dimensions) around the domain boundary, is less than a specified tolerance $\epsilon_{\text{tol}} = 10^{-12}$.

Remark. Since in the proposed approach there is no need to recalculate second derivatives and apply the nonlinear minmod operator at every iteration, it is possible to obtain the steady-state solution of the advection equations in an implicit fashion. This could be very beneficial in cases when a good guess for the extended field is available (for example, solutions from preceding time instants in time-dependent problems). Such an approach will be explored in future works.

Remark. In case of linear extrapolations, the first equations in (8) and (9) are not solved, in second equations $[q_{nn}]_{i,j}$ and $[\mathbf{Q}_{\nabla\nabla}]_{i,j}$ are set to zero and first-order accurate formulas (10) are used during the extrapolation of both field q and its derivatives for more efficient computations.

2.4.1 Extension to adaptive Quad-/Oc-tree grids

The methodology introduced in this paper can be trivially extended to Quad-/Oc-tree data structures. Specifically, we sample data fields at nodes of Quad-/Oc-tree grids and use the second-order accurate discretizations of [22] for regions where grids are non-uniform. A band of uniform grids are usually imposed near the interface in practical free boundary applications (see Fig. 1). In this case the extrapolation within some neighborhood around the interface (where it is primarily required) is as accurate as for uniform grids, however, the extrapolation procedure is much faster on adaptive grids for their significant reduction in the total number of grid points.

3 Numerical Results in Two Spatial Dimensions

We consider four physical domains: a disk, a star shape, a union of two disks and an intersection of two disks (see figure 2). The disk is a smooth interface for which the approach of [2] performs well. The star-shape domain is an example where regions of high curvature are present (crest and trough of the wavy shape). The union/intersection of two disks are examples where kinks occur and illustrate

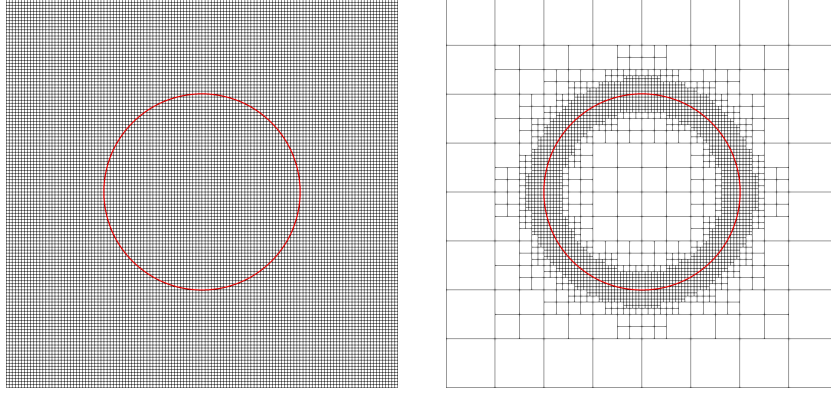


Figure 1: Examples of uniform (left) and adaptive (right) Cartesian grids for a circular interface.

the case of typical free boundary simulations where changes in topology occur. The definition of those domains are given by the level-set functions:

$$\begin{aligned}\phi_0(x, y) &= \sqrt{x^2 + y^2} - 0.501, \\ \phi_1(x, y) &= \sqrt{x^2 + y^2} - 0.501 - 0.25 \frac{y^5 + 5x^4y - 10x^2y^3}{(x^2 + y^2)^{\frac{5}{2}}}, \\ \phi_2(x, y) &= \min \left(\sqrt{(x + .1)^2 + (y + .3)^2} - 0.501, \sqrt{(x - .2)^2 + (y - .2)^2} - 0.401 \right), \\ \phi_3(x, y) &= \max \left(\sqrt{x^2 + y^2} - 0.501, \sqrt{(x - .4)^2 + (y - .3)^2} - 0.401 \right),\end{aligned}$$

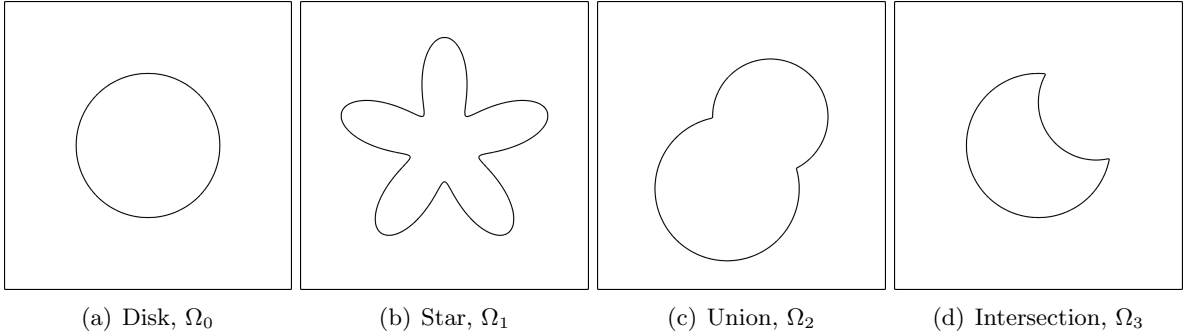


Figure 2: Computational domains considered in section 3.

In each case we consider a computational domain $\Omega = (-1, 1) \times (-1, 1)$. We define the function $q = \sin(\pi x) \cos(\pi y)$ inside every domain and extrapolate it in the outside region. Then the maximum difference between the exact values of q and extrapolated ones, that is, the L^∞ norm of the error, is computed within a band of thickness $2\sqrt{\Delta x^2 + \Delta y^2}$ in the outside regions. Figures 3 and 4 summarize the convergence behavior of the ND-PDE approach of [2] and the proposed WCD-PDE approach. Figure 5 demonstrates the error distribution for both methods in the case of the quadratic extrapolation on a 128^2 grid.

In case of the smooth domain Ω_0 (disk), both approaches produce almost indistinguishable results attaining second- and third-order rates of convergence for the linear and quadratic extrapolation, respectively.

For the high-curvature domain Ω_1 (star), both methods still reach optimal orders of convergence; however the ND-PDE approach demonstrates the optimal order of convergence only at relatively high

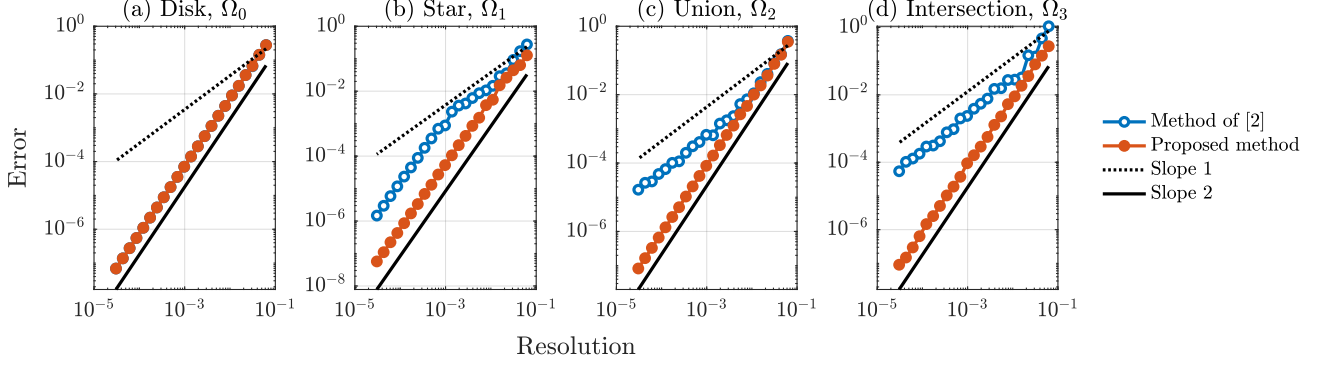


Figure 3: Accuracy of the linear extrapolation (in the L^∞ norm) in two spatial dimensions measured in a narrow band of thickness $2\sqrt{\Delta x^2 + \Delta y^2}$ around an interface using the approach of [2] and the proposed approach.

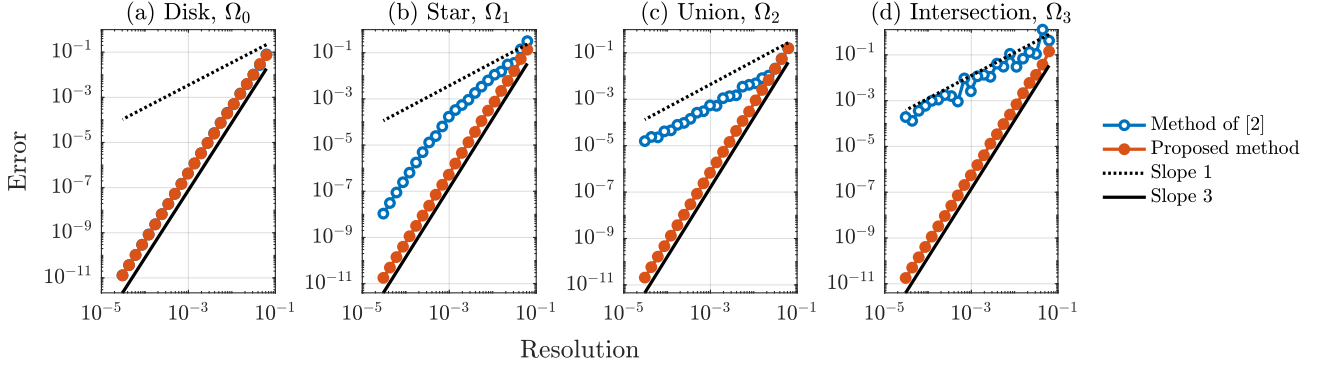


Figure 4: Accuracy of the quadratic extrapolation (in the L^∞ norm) in two spatial dimensions measured in a narrow band of thickness $2\sqrt{\Delta x^2 + \Delta y^2}$ around an interface using the approach of [2] and the proposed approach.

grid resolutions when all geometric features are well-resolved. Moreover, for a given grid resolution the WCD-PDE approach produces results that are more than one order of magnitude more accurate in the case of the linear extrapolation and almost three orders of magnitude more accurate in the case of the quadratic extrapolation compared to the ND-PDE approach. Figure 5b shows that the ND-PDE approach produces very large errors near regions with the highest curvature, while the error in the case of the WCD-PDE approach is much smaller and exhibits very little variation throughout all regions around the interface.

The results are even more significantly improved with the proposed approach in the case of interfaces with kinks Ω_2 (union) and Ω_3 (intersection). Figures 5c and 5d show that the ND-PDE method of [2] produces large errors near kinks; those errors are significantly reduced with the WCD-PDE approach. In particular, Figures 3c-d and 4c-d demonstrate that the second-order (third-order) accuracy of the linear (quadratic) extrapolations are recovered with the proposed approach; the rates of convergence for the approach of [2] are close to first order, which corresponds to the constant extrapolation, due to the fact that errors near kinks do not decrease despite grid refinement and the apparent first order of convergence is only because the neighborhood in which errors are computed is shrinking closer to the domain.

4 Numerical Results in Three Spatial Dimensions

We consider three different domains, $\tilde{\Omega}_1$, $\tilde{\Omega}_2$ and $\tilde{\Omega}_3$, that present high-curvature features or kinks in three spatial dimensions. In addition, we consider a smooth spherical domain $\tilde{\Omega}_0$ with center $(0, 0, 0)$

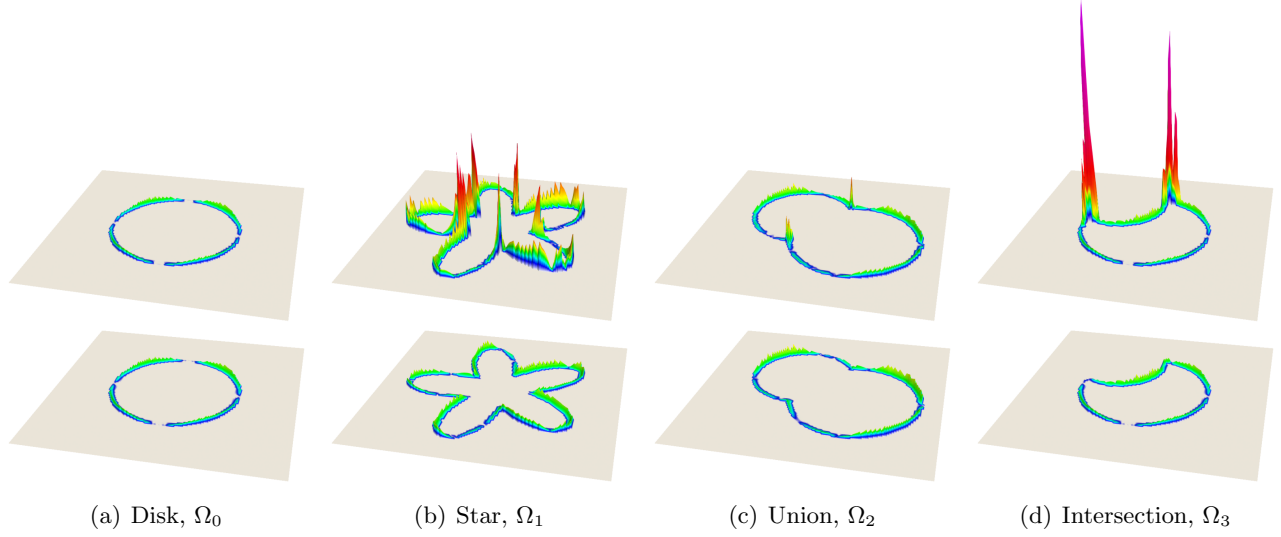


Figure 5: Comparison of error distributions in the case of the quadratic extrapolation on a 128^2 grid. Top row: the approach of [2]. Bottom row: the present approach. In each case the error is multiplied by a factor of 30 for visualization purpose.

and radius 0.501. The definition of those domains are given by the level-set functions:

$$\begin{aligned}\tilde{\phi}_0(x, y, z) &= \sqrt{x^2 + y^2 + z^2} - 0.501, \\ \tilde{\phi}_1(x, y, z) &= \sqrt{x^2 + y^2 + z^2} - 0.501 - 0.15 \frac{y^5 + 5x^4y - 10x^2y^3}{(x^2 + y^2 + z^2)^{\frac{5}{2}}} \cos\left(\frac{\pi}{2} \frac{z}{0.501}\right), \\ \tilde{\phi}_2(x, y, z) &= \min\left(\sqrt{(x + .1)^2 + (y + .3)^2 + (z + .2)^2} - 0.501, \sqrt{(x - .2)^2 + (y - .2)^2 + (z - .1)^2} - 0.401\right), \\ \tilde{\phi}_3(x, y, z) &= \max\left(\sqrt{x^2 + y^2 + z^2} - 0.501, \sqrt{(x - .4)^2 + (y - .3)^2 + (z - .2)^2} - 0.401\right),\end{aligned}$$

Figure 6 depicts those domains along with the octree grid refined near their boundaries.

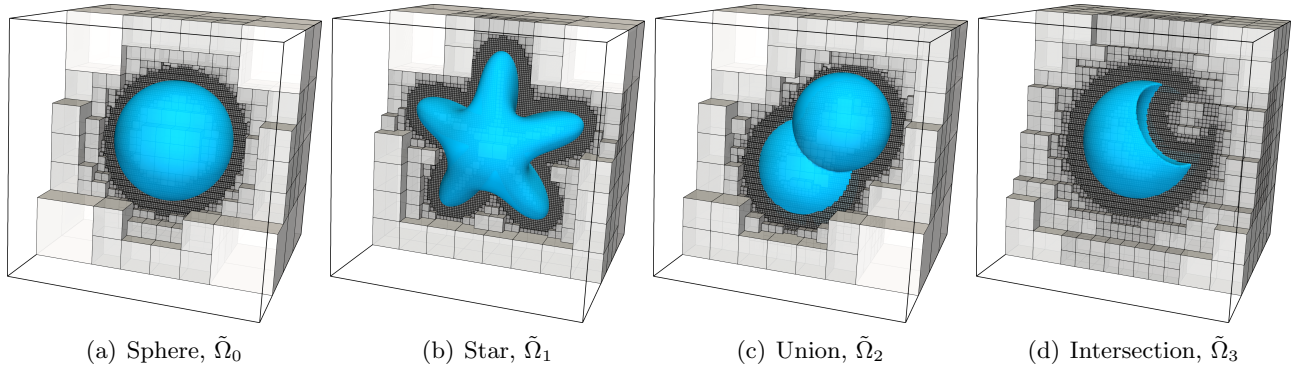


Figure 6: Irregular domains considered in section 4 along with the octree grids refined near their boundaries.

Similar to the two-dimensional examples, we consider a computational domain $\Omega = (-1, 1)^3$. We extrapolate the function $q = \sin(\pi x) \cos(\pi y) \exp(z)$ from the inside to the outside for every domain and compute the difference between the exact values of q and the extrapolated ones, that is, the L^∞ norm of the error, within a band of thickness $2\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ in the outside region.

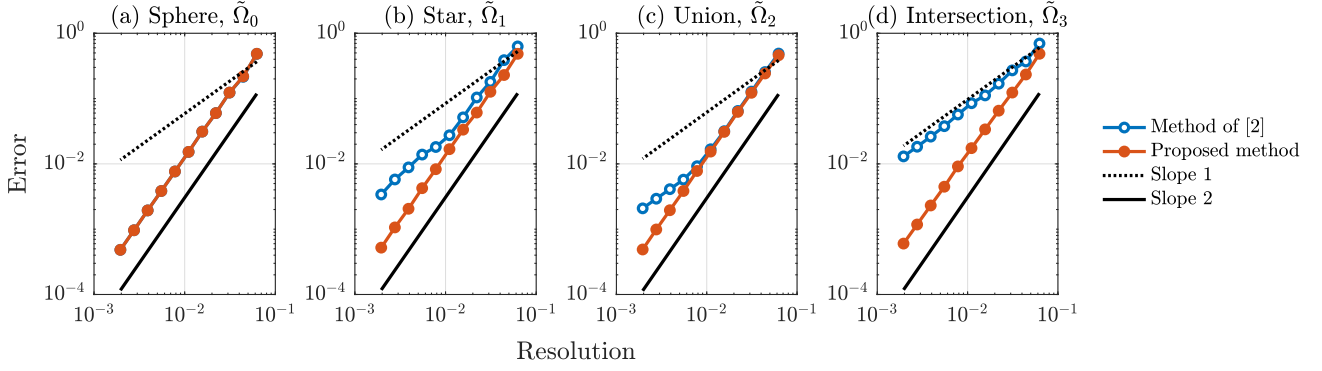


Figure 7: Accuracy of the linear extrapolation (in the L^∞ norm) in three spatial dimensions measured in a narrow band of thickness $2\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ around an interface using the approach of [2] and the proposed approach.

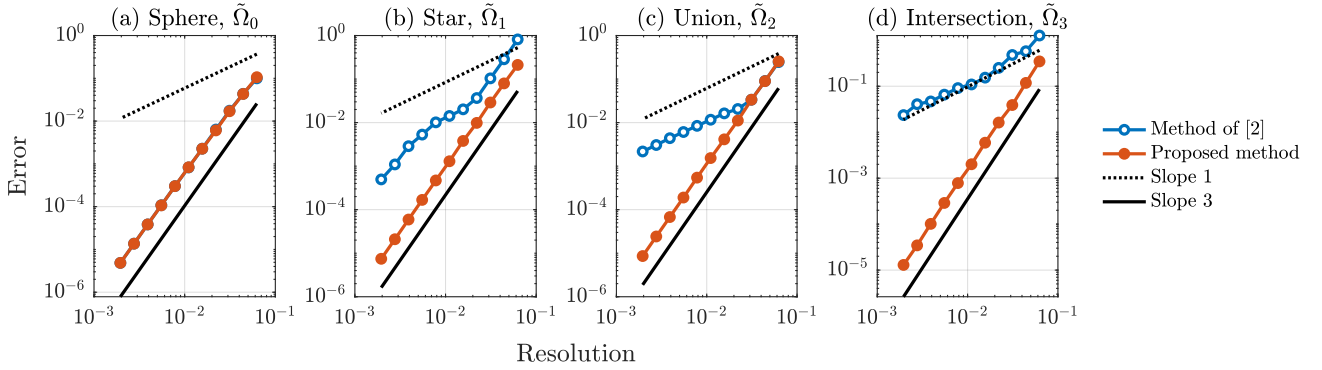


Figure 8: Accuracy of the quadratic extrapolation (in the L^∞ norm) in three spatial dimensions measured in a narrow band of thickness $2\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ around an interface using the approach of [2] and the proposed approach.

Conclusions similar to the two dimensional case can be drawn from the results in figures 7 and 8. Specifically, for a smooth and well-resolved domain $\tilde{\Omega}_0$ (sphere) both approaches produce almost indistinguishable results with optimal order of convergence (second and third for the linear and quadratic extrapolations, respectively). When the interface curvature is high ($\tilde{\Omega}_1$, star) the WCD-PDE approach produce extrapolated fields that are several orders of magnitude more accurate than for the ND-PDE approach. For geometries with sharp features $\tilde{\Omega}_2$ (union) and $\tilde{\Omega}_3$ (intersection) only the WCD-PDE approach demonstrates optimal orders of convergence, while for the ND-PDE approach the rate of convergence is stuck to 1.

5 Application to Solving the Diffusion Equation in Time-Dependent Domains

In order to illustrate the importance of accurate extrapolation near sharp corners in moving interface problems, we present a simple example of solving diffusion equation around a moving object that may have a non-smooth boundary. Specifically, we consider a two-dimensional rectangular region $[-1; 1] \times [-1; 1]$ and an object that moves diagonally from its starting position at $(x_s, y_s) = (-0.51, 0.52)$ at time $t = 0$ to the final position $(x_f, y_f) = (0.49, -0.48)$ at time $t = 1$ while making half a turn around its center as demonstrated in Figure 9. A diffusion equation subject to Neumann boundary conditions is

solved in the rectangular box excluding the region occupied by the moving object. While the problem at hand does not correspond to any specific practical application, it represents a prototypical situation arising in simulation of more relevant (and more complex) processes and at the same time allows a precise analysis of numerical errors. Note that a non-deformable shape is considered for the sake of simplicity, we expect the main conclusions to hold true in more general cases, e.g. multiphase flow with triple junction points.

The Eulerian framework is employed, more precisely, the region $[-1; 1] \times [-1; 1]$ is discretized into a static uniform rectangular grid with N nodes in each Cartesian direction while the object is implicitly described by a time-dependent level-set function.

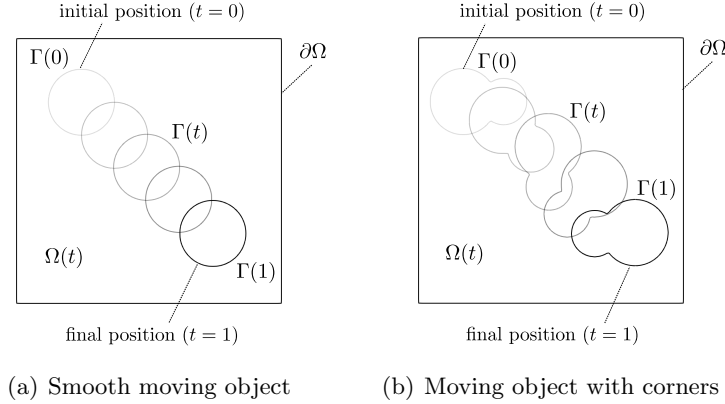


Figure 9: Problem geometry in diffusion equation example from Section 5.

Suppose the shape of the moving object in its local system of coordinate $\boldsymbol{\xi}$ is described by a level-set function $\phi_0(\boldsymbol{\xi})$ (such that $\phi_0(\boldsymbol{\xi}) > 0$ inside the object). Then the object's motion in the global system of coordinates \boldsymbol{r} can be expressed by a time-dependent level-set function $\phi(t, \boldsymbol{r}) = \phi_0(\boldsymbol{\xi}(t, \boldsymbol{r}))$, where global-to-local coordinate transformation $\boldsymbol{\xi}(t, \boldsymbol{r})$ is given by:

$$\boldsymbol{\xi}(t, \boldsymbol{r}) = \begin{pmatrix} \xi(t, \boldsymbol{r}) \\ \eta(t, \boldsymbol{r}) \end{pmatrix} = \begin{pmatrix} x - (x_s + t(x_f - x_s)) \\ y - (y_s + t(y_f - y_s)) \end{pmatrix} \begin{pmatrix} \cos(\pi t) & \sin(\pi t) \\ -\sin(\pi t) & \cos(\pi t) \end{pmatrix}.$$

The solution domain $\Omega(t)$ can then be defined as:

$$\Omega(t) = \{\boldsymbol{r} \in [-1; 1] \times [-1; 1] : \phi(t, \boldsymbol{r}) < 0\}.$$

The boundary of the computational box is denoted as $\partial\Omega$ and the boundary of the moving object is denoted as $\Gamma(t)$.

In order to investigate the influence of non-smooth interface, we consider two choices of moving object, one having a smooth boundary, a disk of radius 0.25, and another one having a non-smooth boundary, a union of two disk with radii 0.25 and 0.175, motivated by multimaterial compound bubbles. In the first case the level-set function of the object is given by (in the local system of coordinates):

$$\phi_0^{\text{smooth}}(\boldsymbol{\xi}) = \phi(\boldsymbol{\xi}) = r_0 - \sqrt{\xi^2 + \eta^2},$$

while in the latter case:

$$\phi_0^{\text{non-smooth}}(\boldsymbol{\xi}) = \max \left(r_0 - \sqrt{(\xi + \xi_0)^2 + \eta^2}, r_0 q - \sqrt{(\xi - \xi_0)^2 + \eta^2} \right),$$

where $r_0 = 0.25$, $\xi_0 = \frac{1}{2}r_0\sqrt{1+q^2}$ and $q = 0.7$.

We choose the following test solution:

$$u(t, \mathbf{r}) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j} \cos\left(\frac{i\pi}{2}(x+1)\right) \cos\left(\frac{j\pi}{2}(y+1)\right) \exp\left(-D\frac{\pi^2}{4}(i^2+j^2)t\right),$$

with

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} = \begin{pmatrix} -0.5 & -0.1 & -0.5 & 0.6 \\ -0.6 & -0.5 & -0.1 & -0.1 \\ 0.2 & -0.2 & -0.2 & 0.4 \\ 0.1 & -0.9 & 0.8 & 0.4 \end{pmatrix},$$

which satisfies a homogeneous diffusion equation:

$$\partial_t u = D\nabla^2 u, \quad \text{for } t \in [0, 1], \mathbf{r} \in \Omega(t), \quad (12)$$

with initial conditions:

$$u(0, \mathbf{r}) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j} \cos\left(\frac{i\pi}{2}(x+1)\right) \cos\left(\frac{j\pi}{2}(y+1)\right), \quad \text{for } \mathbf{r} \in \partial\Omega(0)$$

and boundary conditions:

$$\begin{aligned} D\partial_{\mathbf{n}} u &= 0, & \text{for } t \in [0, 1], \mathbf{r} \in \partial\Omega, \\ D\partial_{\mathbf{n}} u &= g(t, \mathbf{r}), & \text{for } t \in [0, 1], \mathbf{r} \in \Gamma(t), \end{aligned}$$

where function $g(t, \mathbf{r})$ is given by:

$$g(t, \mathbf{r}) = D \frac{\nabla\phi(t, \mathbf{r}) \cdot \nabla u(t, \mathbf{r})}{|\nabla\phi(t, \mathbf{r})|}.$$

The time range $[0; 1]$ is discretized into time layers t_n , $n = 0, 1, 2, \dots$, where the time step between adjacent time layers $\Delta t_{n+1} = t_{n+1} - t_n$ is determined such that the maximum displacement of the moving object boundary during the given time step is expected not to exceed a user-defined fraction f of the grid cell diagonal $\sqrt{\Delta x^2 + \Delta y^2}$, that is:

$$\Delta t_{n+1} = \frac{f\sqrt{\Delta x^2 + \Delta y^2}}{\max_{\Gamma(t_n)} v_{\mathbf{n}}(t_n, \mathbf{r})},$$

where $v_{\mathbf{n}}$ denotes the normal velocity of the object's boundary. Specifically, in this example $f = 0.8$ is taken.

We use the second-order variable-step backward differentiation formula (BDF2) for discretizing the diffusion equation (12) in time, that is:

$$(\alpha_0 - \Delta t_n \nabla^2) u^n = -\alpha_1 u^{n-1} - \alpha_2 u^{n-2}, \quad (13)$$

where

$$\alpha_0 = \frac{1+2\rho}{1+\rho}, \quad \alpha_1 = -(1+\rho), \quad \alpha_2 = \frac{\rho^2}{1+\rho} \quad \text{and} \quad \rho = \frac{\Delta t_n}{\Delta t_{n-1}},$$

and use the superconvergent second-order accurate method of [4] (which is specifically designed to handle irregular domains with non-smooth boundaries) for solving the resulting Poisson-type equation (13).

Solution of equation (13) produces values of u^n at all grid nodes that belong to the current solution domain $\Omega(t_n)$. However, as the object moves in time some of grid nodes outside of $\Omega(t_n)$ may become part of $\Omega(t_{n+1})$ or $\Omega(t_{n+2})$ and solving at subsequent time layers t_{n+1} and t_{n+2} would require valid values of u^n at such grid nodes. This is typically addressed in free boundary value problems by smoothly extrapolating u^n into some neighborhood of $\Omega(t_n)$. In this example we quadratically extrapolate solutions using both the proposed in this work WCD-PDE approach and the ND-PDE approach of [2]. Also, since solving Poisson-type equations on irregular domains produces additional errors, we generate a reference solution where instead of performing extrapolation of numerical values we fill the grid nodes outside of the current solution domain $\Omega(t_n)$ with exact values given by the analytical solution.

We investigate the influence of extrapolation procedures on the accuracy of the numerical solution and its gradient. In order not to measure the error of extrapolation procedure itself but rather only its influence on solving the diffusion equation, the solution error at time layer t_n is calculated only for grid nodes in $\Omega(t_n)$ and the gradient is calculated only using values from $\Omega(t_n)$ as well.

Obtained results are summarized in Figures 10, 11, 12 and 13. Figures 10 and 11 show error distributions at the final time $t = 1$ on a 128^2 grid. As one can see, for a smooth moving object (Fig. 10) using either the ND-PDE extension or the WCD-PDE one, results in errors that are very close to ones of the reference solution, while for a non-smooth moving object (Fig. 11) the WCD-PDE extension produces a much more accurate solution that is also very close to the reference one, whereas using the ND-PDE extension results in a significant accumulation of errors in the wake of the moving object. More quantitative conclusions can be drawn from the convergence studies shown in Figures 12 and 13, in which the dependence of the error in the L^∞ norm for both numerical solution and its gradient on the grid resolution is presented. In case of a smooth moving object (Fig. 12) both extension methods lead to second-order convergence both in the numerical solutions and its gradient (as what is expected from the superconvergent method of [4]) with magnitude of errors being very close to the ones of the reference solution. In case of a non-smooth moving object (Fig. 13) the accuracy of numerical solutions obtained using the ND-PDE extension degrades severely showing only first-order convergence in the solution itself and non-convergence in the gradient. At the same time the accuracy of computations based on the proposed WCD-PDE extension seems affected very slightly by the presence of sharp corners, retaining the second-order convergence in the numerical solution and its gradient with errors very close to the reference solution.

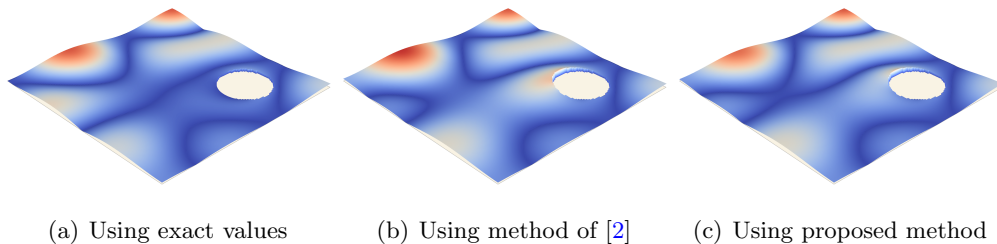


Figure 10: Error distribution at the final time moment ($t = 1$) in case of a smooth moving object using different extrapolation approaches.

Remark: Note that the presented in this work extrapolation approach is designed for extending smooth scalar fields (as in the present example). However, in general, solutions to partial differential equations in domains with sharp features may contain singularities. In such cases for best results the proposed extrapolation procedure should only be applied to the regular part of the solution, the singular

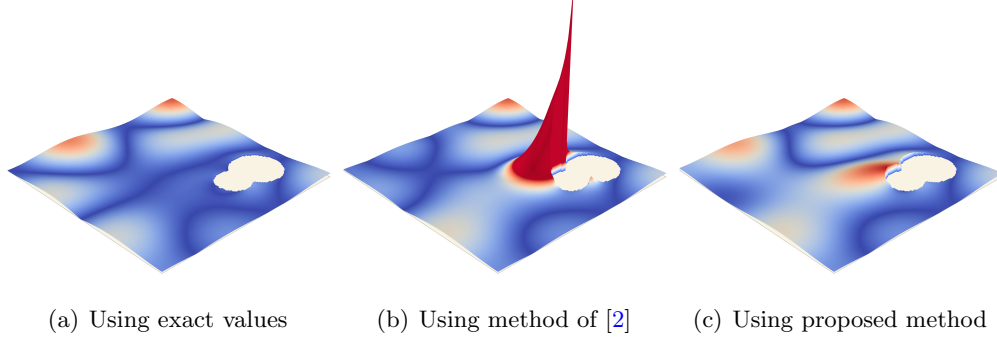


Figure 11: Error distribution at the final time moment ($t = 1$) in case of a non-smooth moving object using different extrapolation approaches.

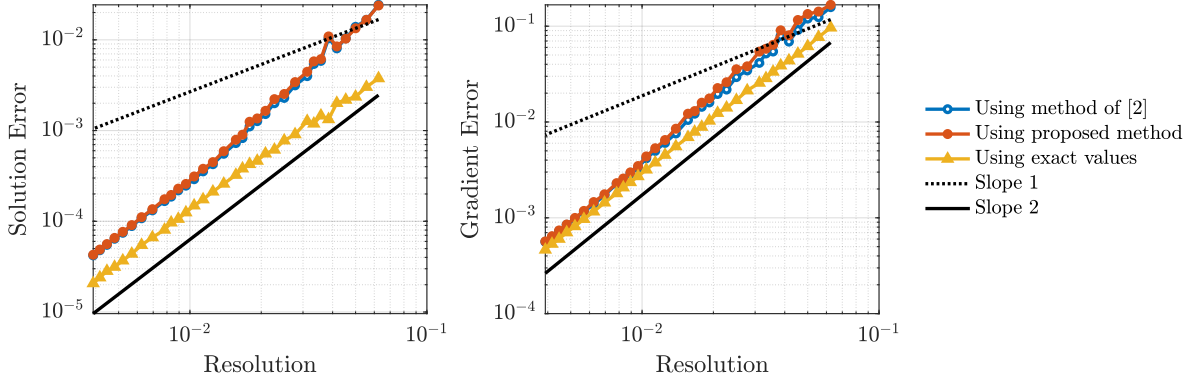


Figure 12: Accuracy of solving diffusion equation (in the L^∞ norm) in case of a smooth moving object using different extrapolation approaches.

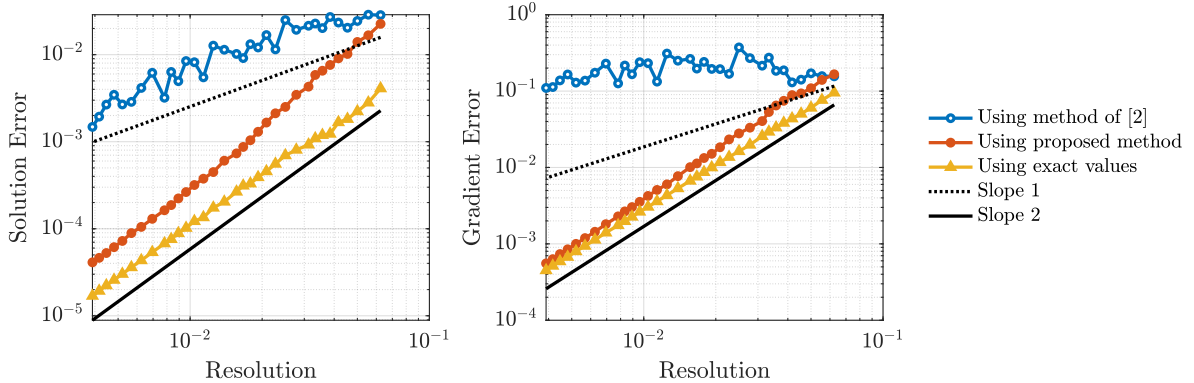


Figure 13: Accuracy of solving diffusion equation (in the L^∞ norm) in case of a non-smooth moving object using different extrapolation approaches.

part must be dealt with separately using special methods, for example, as in [41].

6 Conclusion

We have presented a numerical method for extrapolating scalar quantities across the boundaries of irregular domains that may present high-curvature features or kinks. Linear and quadratic extrapolations procedures produce second- and third-order accurate results in the L^∞ norm, respectively and do so

regardless of the irregularity of the boundaries, i.e. boundaries with kinks can readily be considered. These procedures are effective in both two and three spatial dimensions and can be implemented on quadtree and octree Cartesian grids. We have shown through numerical examples that errors associated with extrapolations can be reduced by several orders of magnitude in some cases, compared with the approach of [2] commonly used in level-set methods. We have also presented an example of solving a diffusion equation on evolving domains in order to highlight the importance of accurate extrapolation near sharp geometric features for practical applications. The numerical method we introduced is based on solving PDEs in pseudo-time, but we note that static solutions based on an implicit approach like Fast Marching or Fast Sweeping could be obtained and we expect the results to follow the same general behavior as that presented in the current manuscript.

Acknowledgment

This work was supported by ONR MURI N00014-17-1-2676.

References

- [1] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, 194(1):363–393, 2004.
- [2] Tariq Aslam. A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.*, 193(1):349–355, 2004.
- [3] Tariq Aslam, Songting Luo, and Hongkai Zhao. A static pde approach for multidimensional extrapolation using fast sweeping methods. *SIAM J. Sci. Comput.*, 36(6):A2907–A2928, 2014.
- [4] Daniil Bochkov and Frederic Gibou. Solving the poisson equation with Robin boundary conditions on piecewise smooth irregular boundaries. *J. Comput. Phys.*, 376:1156–1198, 2019.
- [5] Adam Chacon and Alexander Vladimirsky. A parallel two-scale method for Eikonal equations. *SIAM J. Sci. Comput.*, 37(1):A156–A180, 2015.
- [6] Han Chen, Chohong Min, and Frederic Gibou. A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate. *J. Comput. Phys.*, 228(16):5803–5818, 2009.
- [7] Charles Cleret de Langavant, Arthur Guittet, Maxime Theillard, Fernando Temprano-Coletto, and Frédéric Gibou. Level-set simulations of soluble surfactant driven flows. *J. Comput. Phys.*, 348:271–297, 2017.
- [8] Miles Detrixhe and Frédéric Gibou. Hybrid Massively Parallel Fast Sweeping Method for static Hamilton-Jacobi Equations. *J. Comput. Phys.*, 322:199–223, 2016.
- [9] Miles Detrixhe, Frédéric Gibou, and Chohong Min. A parallel fast sweeping method for the eikonal equation. *J. Comput. Phys.*, 237:46–55, 2013.
- [10] Raphael Egan and Frédéric Gibou. Fast and scalable algorithms for constructing solvent-excluded surfaces of large biomolecules. *J. Comput. Phys.*, 374:91–120, 2018.
- [11] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152(2):457–492, 1999.
- [12] Frederic Gibou, Liguang Chen, Duc Nguyen, and Sanjoy Banerjee. A level set based sharp interface method for the multiphase incompressible navier–stokes equations with phase change. *J. Comput. Phys.*, 222(2):536–555, March 2007.
- [13] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J. Comput. Phys.*, 202(2):577–601, 2005.
- [14] Frédéric Gibou, Ronald Fedkiw, Russel Caflisch, and Stanley Osher. A level set approach for the numerical simulation of dendritic growth. *J. Sci. Comput.*, 19(1-3):183–199, 2003.
- [15] Frederic Gibou, Ronald Fedkiw, Li-Tien Cheng, and Myungjoo Kang. A second-order accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, 176(1):205–227, 2002.

- [16] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *J. Comput. Phys.*, 353:82–109, 2018.
- [17] Frederic Gibou and Chohong Min. Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions. *J. Comput. Phys.*, 231:3246–3263, 2012.
- [18] Frederic Gibou, Chohong Min, and Ronald Fedkiw. High resolution sharp computational methods for elliptic and parabolic problems in complex geometries. *J. Sci. Comput.*, 54:369–413, 2013.
- [19] Ásdís Helgadóttir and Frederic Gibou. A Poisson-Boltzmann solver on irregular domains with Neumann or Robin boundary conditions on non-graded adaptive grid. *J. Comput. Phys.*, 230:3830–3848, 2011.
- [20] Mathieu Lepilliez, Elena Roxana Popescu, Frederic Gibou, and Sébastien Tanguy. On two-phase flow solvers in irregular domains with contact line. *J. Comput. Phys.*, 321:1217–1251, 2016.
- [21] Frank Losasso, Frederic Gibou, and Ron Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, 2004.
- [22] Chohong Min and Frederic Gibou. A second order accurate level set method on non-graded adaptive Cartesian grids. *J. Comput. Phys.*, 225(1):300–321, 2007.
- [23] Mohammad Mirzadeh and Frédéric Gibou. A conservative discretization of the Poisson–Nernst–Planck equations on adaptive cartesian grids. *J. Comput. Phys.*, 274:633–653, 2014.
- [24] Mohammad Mirzadeh, Arthur Guittet, Carsten Burstedde, and Frederic Gibou. Parallel level-set methods on adaptive tree-based grids. *J. Comput. Phys.*, 322:345–364, 2016.
- [25] Mohammad Mirzadeh, Maxime Theillard, Asdís Helgadóttir, David Boy, and Frédéric Gibou. An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations. *Commun. Comput. Phys.*, 13(1):150–173, 2012.
- [26] Pouria Mistani, Arthur Guittet, Daniil Bochkov, Joshua Schneider, Dionisios Margetis, Christian Ratsch, and Frederic Gibou. The island dynamics model on parallel quadtree grids. *J. Comput. Phys.*, 361:150–166, 2018.
- [27] Timothy J Moroney, Dylan R Lusmore, Scott W McCue, and DL Sean McElwain. Extending fields in a level set method by solving a biharmonic equation. *J. Comput. Phys.*, 343:170–185, 2017.
- [28] Duc Nguyen, Frederic Gibou, and Ronald Fedkiw. A Fully Conservative Ghost Fluid Method and Stiff Detonation Waves. In *12th Int. Detonation Symposium, San Diego, CA*, 2002.
- [29] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.
- [30] Stanley Osher and James A Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [31] Gaddiel Ouaknin, Nabil Laachi, Kris Delaney, Glenn H Fredrickson, and Frederic Gibou. Self-consistent field theory simulations of polymers on arbitrary domains. *J. Comput. Phys.*, 327:168–185, 2016.
- [32] Joseph Papac, Frederic Gibou, and Christian Ratsch. Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions. *J. Comput. Phys.*, 229:875–889, 2010.

- [33] Joseph Papac, Asdis Helgadóttir, Christian Ratsch, and Frederic Gibou. A level set approach for diffusion and Stefan-type problems with Robin boundary conditions on Quadtree/Octree adaptive Cartesian grids. *J. Comput. Phys.*, 233:241–261, 2013.
- [34] Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph.*, 27(3):1–9, 2008.
- [35] Chris H Rycroft and Frédéric Gibou. Simulations of a stretching bar using a plasticity model from the shear transformation zone theory. *J. Comput. Phys.*, 231(5):2155–2179, 2012.
- [36] James A Sethian. *Level set methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 1996. Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science.
- [37] James A Sethian and Alexander Vladimirsky. Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proc. Natl. Acad. Sci.*, 97/11:5699–5703, 2000.
- [38] James A Sethian and Alexander Vladimirsky. Ordered upwind methods for static hamilton-jacobi equations. *Proc. Natl. Acad. Sci.*, 98/20:11069–11074, 2001.
- [39] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- [40] Maxime Theillard, Landry Fokoua Djodom, Jean-Léopold Vié, and Frédéric Gibou. A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids – application to shape optimization. *J. Comput. Phys.*, 233:430–448, 2013.
- [41] Neil M Wigley. An efficient method for subtracting off singularities at corners for laplace’s equation. *J. Comput. Phys.*, 78(2):369–377, 1988.
- [42] Hongkai Zhao. A fast sweeping method for eikonal equations. *Math. Comput.*, 74:603–627, 2004.