

Efficient Open-Boundary Poisson Solves

XINGYU NI, Inria, France

JIONG CHEN[✉], Inria, France

SIYUAN CHEN, University of British Columbia, Canada

CAOLIWEN WANG, University of British Columbia, Canada

MATHIEU DESBRUN, Inria / École polytechnique, France

TAKU KOMURA, University of Hong Kong, Hong Kong SAR

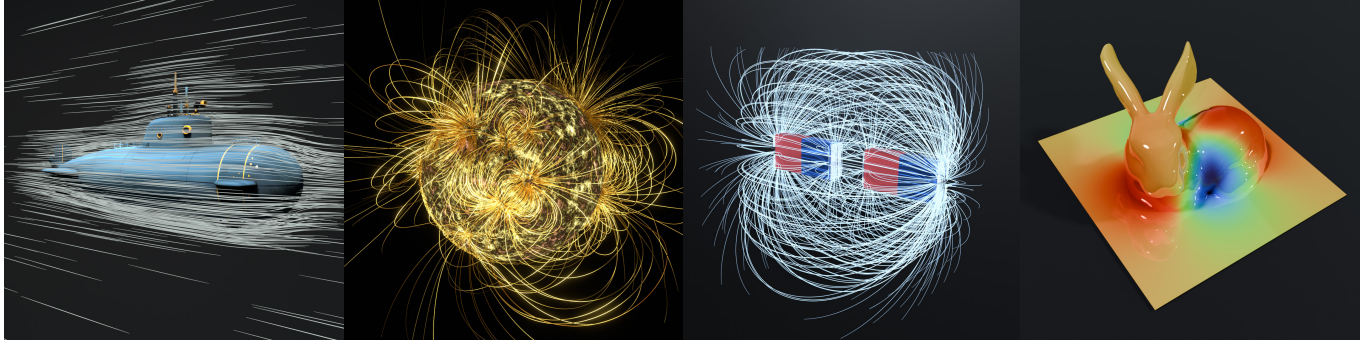


Fig. 1. **Open-boundary problems.** Many problems in graphics involve elliptic equations defined on unbounded domains with far-field conditions. Our proposed partitioned, iterative solver enables the efficient solution of such open-boundary problems and supports a range of visually compelling applications, which include simulating the flow around a submarine moving through deep water, reconstructing the Sun’s coronal magnetic field from observational data, computing the magnetic field of two interacting magnets magnetized by an external field, and recovering a harmonic function outside a bunny model from prescribed surface constraints, as illustrated through streamlines or false colors in these examples.

Poisson equations arise pervasively in computer graphics, yet efficiently solving them on unbounded domains remains a long-standing challenge. Existing approaches attempting to model boundary conditions at infinity either incur prohibitive computational costs by expanding the simulation domain, sacrifice accuracy through artificial boundary conditions, or lack the flexibility to handle varying coefficients. In this paper, we introduce a simple and practical method for solving generalized Poisson equations on finite domains with open boundaries, enabling exact enforcement of asymptotic conditions at infinity. Our method couples a finite-volume interior solver with a boundary-element formulation of the exterior problem, enforcing continuity of both the solution and its normal derivative across an artificial boundary. Inspired by Johnson–Nédélec coupling and interface relaxation, we develop a partitioned, iterative scheme based on damped fixed-point iterations, which leverages fast, inexact boundary-element solves for the exterior problem and efficient sparse solves for the interior. Theoretical analysis and empirical tests confirm that our method provides superior accuracy and performance compared to existing ad-hoc solutions across a variety of graphics applications.

Authors’ addresses: Xingyu Ni (xingyu.ni@inria.fr), Jiong Chen (jiong.chen@inria.fr), Mathieu Desbrun (mathieu.desbrun@inria.fr); Inria Saclay and LIX/DIX (IP Paris), Palaiseau, France; Siyuan Chen (chensiyuan030105@gmail.com), Caoliwen Wang (wclw1021@gmail.com); University of British Columbia, Vancouver, Canada; Taku Komura (taku@cs.hku.hk); University of Hong Kong, Hong Kong, Hong Kong SAR.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 1557-7368/2026/12-ART180
<https://doi.org/10.1145/3842558>

CCS Concepts: • **Mathematics of computing** → **Solvers**; • **Computing methodologies** → **Computer graphics**.

Additional Key Words and Phrases: generalized Poisson equations, open boundaries, exact artificial boundary conditions, boundary element method.

ACM Reference Format:

Xingyu Ni, Jiong Chen, Siyuan Chen, Caoliwen Wang, Mathieu Desbrun, and Taku Komura. 2026. Efficient Open-Boundary Poisson Solves. *ACM Trans. Graph.* 45, 6, Article 180 (December 2026), 23 pages. <https://doi.org/10.1145/3842558>

1 Introduction

Efficiently solving the Poisson equation—and its generalizations involving spatially-varying or nonlinear coefficients—is essential across a variety of computer graphics applications, ranging from geometry processing to physical simulation, including image fusion [Pérez et al. 2003; Agarwala 2007], shape-from-shading [Agrawal et al. 2006; Harker and O’Leary 2008], barycentric coordinate interpolation [Joshi et al. 2007; de Goes and Desbrun 2024], surface reconstruction [Kazhdan et al. 2006; Kazhdan and Hoppe 2013], mesh parameterization [Vaxman et al. 2016; Floater 1997], fluid animation [Bridson 2015; Stam 1999], and electromagnetic modeling [Ni et al. 2024, 2020; Huang et al. 2019; Huang and Michels 2020]. For boundary value problems (BVPs) on bounded domains with local boundary conditions, traditional discretization schemes—namely finite element (FE), finite difference (FD), or finite volume (FV) methods—transform the continuous partial differential equation and its associated boundary conditions into a structured linear

system for which a wide array of established, high-performance numerical solvers already exist.

However, many practical applications such as fluid flow in open environments, far-field electromagnetic potentials, and free-space surface reconstruction are supposed to involve unbounded domains, and thus require the use of *open boundaries* on a finite computational domain, implicitly involving boundary conditions at infinity. While graphics practitioners typically employ domain truncation with Dirichlet, Neumann, or other ad-hoc boundary conditions, the resulting approximation error decays slowly relative to the computational and memory costs incurred by larger domains. A case in point is provided by virtual wind tunnels, where the vehicle for which the aerodynamic coefficient is computed must be very small relative to the computational domain to ensure that boundary effects do not contaminate the results [Lyu et al. 2023]. Specialized techniques like infinite elements [Bettess 1977] or exact artificial boundary conditions (ABCs) [Han and Wu 2013] can approximate far-field behavior; yet, they often introduce numerical ill-conditioning or require the Green’s function of the truncated domain to be analytically tractable, limiting their general utility.

In contrast to FEM, FDM, and FVM, the boundary element method (BEM) discretizes boundary integral equations (BIEs) rather than the governing partial differential equations. By integrating the fundamental solution (or free-space Green’s function) of the differential operator to assemble the linear system, BEM can inherently deal with unbounded domains. However, BEM requires closed-form fundamental solutions, which are generally unavailable for nonlinear or variable-coefficient operators. While hybrid strategies can address general cases by coupling finite-domain methods for convoluted domains with BEM for the domain exterior, the resulting linear system matrices are dense and structurally heterogeneous. This technical complexity and its induced computational overhead have hindered the widespread adoption of such hybrid frameworks within the computer graphics community.

Overview. Inspired by FEM–BEM coupling methods [Johnson and Nédélec 1980; Elleithy and Tanaka 2003], we propose an iterative scheme that enables finite-domain Poisson solvers to handle open boundaries without modifying the underlying solver or governing equations. In this framework, the interior solver functions as a “black box” subject to Dirichlet conditions on the domain boundary. By extracting normal derivatives from the intermediate solution, we drive a Boundary Integral Equation (BIE) to enforce compatibility with the infinite domain. This BIE step provides the updated Dirichlet conditions for the subsequent interior solve. One of the key contributions of our work is the observation that an exact solve of the exterior BIE is unnecessary—inexact solutions are faster to compute and sufficient for rapid convergence. We provide formal convergence proofs for the standard Poisson equation and demonstrate the method’s versatility across canonical geometries and staggered-grid discretizations. Our results across 2D and 3D benchmarks exhibit high accuracy, often converging in just a few iterations. This approach obviates the need for large computational “padding” domains and generalizes readily and effectively to variable-coefficient and nonlinear Poisson-type problems.

2 Related Work

We briefly discuss relevant methods and their underlying methods for solving PDEs with open boundaries, including boundary integral formulations, domain truncation via artificial boundary conditions, and geometric inversion using the Kelvin transform.

Boundary Integral Equations (BIEs). For linear and homogeneous PDEs without source terms, the solution is uniquely determined by the boundary conditions constrained by the BIEs. Once these conditions are resolved, the solution can be evaluated at an arbitrary point in space. This formulation naturally accommodates infinite domains with conditions at infinity based on the far-field properties of typical Green’s functions. This “surface-only” approach has benefited a wide range of applications in computer graphics, including vector graphics [van de Gronde 2010; Sun et al. 2012], geometry processing [Solomon et al. 2017], deformation [James and Pai 1999], simulation [Sugimoto et al. 2022; Schreck et al. 2019] and rendering [Xia et al. 2020]. However, a historical bottleneck of boundary-driven methods is their poor scalability, primarily due to the dense linear systems they generate. Recent advancements have mitigated this issue through massively parallel sparse inverse preconditioners [Chen et al. 2024, 2025], which scale up and accelerate the BIE solves significantly, while hierarchical summation techniques [Greengard and Rokhlin 1987; Hackbusch 1999; Barnes and Hut 1986] enable efficient pointwise evaluation in unbounded spaces. In parallel, the stochastic formulation of boundary integrals has contributed to the rise in popularity of Monte Carlo methods [Sawhney and Crane 2020; Sugimoto et al. 2023; Sawhney et al. 2023], bringing useful techniques from the rendering community into the realm of PDE solvers. These methods facilitate massive parallelism via random walks and can further leverage boundary caching to suppress noise [Miller et al. 2023; Zhou et al. 2025]; however, their performance degrades in unbounded domains where most samples would never hit the boundary, and for the remaining ones, the path can become long, resulting in high variance [Nabizadeh et al. 2021]. Despite these advancements, a critical limitation persists: boundary-driven methods falter when the PDE includes source terms, spatially varying coefficients, or nonlinearities, which preclude a strictly boundary-based formulation, and necessitates expensive volume integration and domain sampling. Consequently, both deterministic and stochastic methods would suffer significant penalties in computational efficiency, accuracy, and numerical stability.

Artificial Boundary Conditions (ABCs). When dealing with infinite domains, only a finite portion of space is covered by the computational domain, the border of which effectively creates an *artificial boundary* requiring specific conditions. These conditions are introduced to either approximate far-field behavior, or analytically resolve the exterior problem, thereby minimizing artifacts caused by the domain truncation. For wave equations, ABCs aim to make this boundary “transparent” or “absorbing” to prevent spurious reflective waves. To this end, Engquist and Majda [1977] developed a hierarchy of *local* absorbing conditions by applying rational approximations to the non-local operators in Fourier space. Taking a different approach, the Perfectly Matched Layer (PML) method [Berenger 1994; Bojsen-Hansen and Wojtan 2016; Chern 2019] constructs a fictitious

boundary layer to absorb incoming waves exponentially fast so that they do not bounce back and pollute the interior simulation. For elliptic PDEs, the exterior problem can be solved analytically using Green's identities, which express the normal derivative of the solution as an explicit function of the boundary value. The resulting exact, non-local Dirichlet-to-Neumann (DtN) map [Givoli and Keller 1989] is further used to constrain the interior solve in a Galerkin framework. However, deriving such DtN maps typically requires canonical interfaces (e.g., sphere in 3D), where an interface-constrained fundamental solution can be obtained through Fourier expansion. In contrast, Johnson and Nédélec [1980] implicitly defined the DtN map through a BIE for non-canonical boundaries, and coupled the BIE with the interior PDE to form a global “monolithic” system. While it offers a far more general solution for arbitrary interior/exterior interfaces, it incurs a high computational cost like other DtN methods: the resulting system combines dense (boundary) and sparse (interior) blocks, a hybrid matrix structure that precludes the use of high-performance solvers developed for either domain individually. Addressing this bottleneck is our primary focus.

Iterative Coupling (IC). To bypass the prohibitive cost of monolithic FEM/BEM solvers, prior work has turned to iterative coupling, in which subdomains are solved separately and Dirichlet/Neumann data are exchanged across their common interfaces. This strategy has been applied to both linear [Kamiya et al. 1996; Feng and Owen 1996; Elleithy and Tanaka 2003] and nonlinear [Soares et al. 2004] problems. As noted in Elleithy and Tanaka [2003], however, standard Dirichlet-Neumann and Neumann-Neumann schemes degenerate when Neumann conditions cover the entire boundary of a subdomain; for the Helmholtz equation, for instance, the standard remedy following Després’ approach [Després 1991; Bendali et al. 2007] is to impose Robin-type conditions on the interface to avoid such breakdown. Our method is conceptually similar in spirit, but the Poisson setting admits a simpler fix: imposing a vanishing boundary condition at infinity ensures the well-posedness of the exterior Neumann problem, whose resulting NtD map then guarantees well-posedness of the interior problem. Moreover, while previous strategies require a full BEM solve at each iteration, we show that *faster convergence can be achieved using inexact solves*. Finally, convergence of FEM-BEM alternations depends critically on the choice of a relaxation parameter whose admissible range can be determined via eigenanalysis [El-Gebeily et al. 2002]; however, such an approach is often intractable for large-scale applications, motivating on-the-fly adjustment strategies instead. We thus recommend using Aitken acceleration [Irons and Tuck 1969], which has already been adopted for iterative coupling in the graphics community and beyond [Coulier et al. 2014; Akbay et al. 2018].

Kelvin Transform (KT). Turning unbounded domains inside out through coordinate stretching [Grosch and Orszag 1977] is another approach to solving exterior problems using standard finite-domain solvers. However, this geometric inversion often results in singular, ill-conditioned systems and may incur severe discretization errors. Originally introduced by Thomson [1845], the Kelvin transform addresses these issues by decomposing the solution of an exterior Poisson equation into a scaling factor—which captures the asymptotic behavior at infinity—and a well-behaved component that can

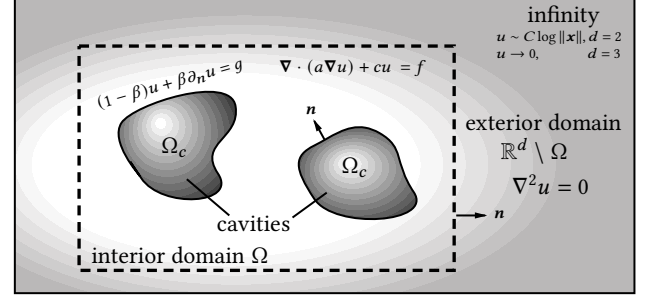


Fig. 2. **Dealing with open-boundaries.** For generalized Poisson problems in unbounded domains, we introduce an artificial boundary to truncate the computational domain and apply exact artificial boundary conditions to accurately account for the exterior to far-field behavior of the solution. Our approach alternates between the interior and exterior solvers, yielding a simple and practical method for handling open boundaries.

be numerically solved via a standard Poisson equation on a compact interior domain. Recently, Nabizadeh et al. [2021] demonstrated that KT can also be applied to Helmholtz equation and is flexible enough to accommodate different interior solvers, including mesh-based and Monte-Carlo methods. While KT excels at solving exterior homogeneous equations if the solution’s asymptotic behavior at infinity is known in advance, it cannot handle generalized Poisson equations or cases lacking a bounding surface. In contrast, our method handles general, and possibly inhomogeneous, interior problems by design while assuming the exterior is simply governed by Laplace’s equation. These two approaches are thus complementary, and one could even use KT as a viable alternative to the BEM employed by our approach for the exterior domain.

3 Method Overview

We now outline our methodology, starting with the mathematical formulation of open-boundary Poisson problems. We then introduce an artificial boundary method to handle the inside and outside of the computational domain boundary, before detailing an alternating interior–exterior solver for the resulting coupled problem.

3.1 Problem Formulation

We address the following class of boundary value problems:

$$\nabla \cdot (a(\mathbf{x}, u, \nabla u) \nabla u(\mathbf{x})) + c(\mathbf{x}, u) u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \setminus \Omega_c \quad (1a)$$

$$\text{s.t.} \quad (1 - \beta(\mathbf{x})) u(\mathbf{x}) + \beta(\mathbf{x}) \frac{\partial u}{\partial n}(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_c \quad (1b)$$

$$\text{and} \quad \begin{cases} u \sim C \log \|\mathbf{x}\|, & d = 2 \\ u \rightarrow 0, & d = 3 \end{cases}, \quad \|\mathbf{x}\| \rightarrow \infty \quad (1c)$$

in d dimensions ($d = 2, 3$), where $\Omega_c \subset \mathbb{R}^d$ denotes one or more cavities (closed subdomains, with the empty set as a trivial case). The solution $u(\mathbf{x})$ is sought in the unbounded domain outside of the cavities $\mathbb{R}^d \setminus \Omega_c$, subject to a mix of Neumann and Dirichlet conditions on the cavity boundary, together with a far-field condition governing the asymptotic behavior of $u(\mathbf{x})$ as $\|\mathbf{x}\| \rightarrow \infty$ —see Fig. 2 for an illustration. Many graphics applications boil down to solving

this type of problems, each characterized by specific PDE coefficients, source terms, and boundary data. We discuss the various components of this general problem next.

Partial Differential Equation. Eq. (1a) is a generalized Poisson equation, where a denotes a diffusion coefficient, c is a reaction coefficient, and f represents a prescribed source field. In many graphics applications, these quantities are assumed to have compact spatial support, meaning that c , f are nonzero and a varies only over a localized region—though the induced solution may propagate far beyond it. Accordingly, these coefficients are typically discretized over the elements of a mesh in a piecewise constant or linear fashion, and set to zero (or constant for a) outside of it. In general, Eq. (1a) is nonlinear since a and/or c may depend on the solution u or its gradient; however, if these coefficients reduce to functions of spatial coordinate $\mathbf{x}$ only, the equation becomes linear. Notably, setting $a \equiv 1$ and $c \equiv 0$ turns Eq. (1a) into the classical Poisson equation.

Boundary Conditions. Very often, the domain contains cavities representing physical objects, such as conductors in an electrostatic field, obstacles in a fluid flow, or scattering bodies in wave propagation. On the cavity boundary $\partial\Omega_c$, specific conditions Eq. (1b) on the solution u are imposed, involving a linear combination of the solution u and its normal derivatives $\partial u / \partial n$ (where the unit normal vector $\mathbf{n}$ points outwards from Ω_c). The weight $\beta : \partial\Omega_c \rightarrow [0, 1]$ determines the boundary type and allows for a smooth transition between Dirichlet conditions ($\beta = 0$, specifying potential), Neumann conditions ($\beta = 1$, specifying flux), or Robin conditions ($0 < \beta < 1$, specifying a mix), while the function $g : \partial\Omega_c \rightarrow \mathbb{R}$ further prescribes the associated boundary data.

Far-Field Condition. To ensure the well-posedness of the problem in this unbounded setting, a standard far-field condition is imposed through Eq. (1c) to guarantee the uniqueness of the solution. In three dimensions, the condition simply requires $u(\mathbf{x})$ to vanish at infinity. Such a condition arises naturally across a number of physical contexts: in electrostatics, the electrical potential due to a localized charge distribution decays to zero at infinity; in fluid dynamics, the perturbation velocity induced by a stationary obstacle vanishes far from the body; in heat conduction, the temperature rise above the ambient value decays to zero as the thermal influence diminishes with distance; and in quantum mechanics, the wavefunction of a bound-state particle must decay to zero as a result of spatial localization and probability normalization; etc. In two dimensions, the situation is slightly different: the solution admits at most logarithmic growth at infinity, and vanishes in the far field only when the constant C (representing the net flux through the artificial boundary introduced below) is zero. Far-field conditions are essential for characterizing isolated systems, as they define how a system's influence asymptotically vanishes or stabilizes at infinite distance.

3.2 Artificial Boundary

Since tessellating the entire space is computationally infeasible, we handle unbounded domains using truncation and artificial boundary conditions. Geometrically, this means that we introduce a *finite* computational domain Ω , which truncates the entire space for computational convenience. Its boundary $\partial\Omega$ can be understood as an

Algorithm 1: Open-boundary Poisson solver

Input: Generalized Poisson equation on an unbounded domain;
Grid domain Ω with its boundary faces centered at $\{\mathbf{x}_i\}$.
Output: Unbounded-domain solution $u(\mathbf{x})$, evaluated within Ω .

```

1 forall  $i$  do  $u^{(0)}(\mathbf{x}_i) \leftarrow 0$ ; // Zeroing Dirichlet data
2  $k \leftarrow 0$ ;
3 repeat
4   // Interior finite-difference solve
5   Call the bounded-domain Poisson solver to compute  $u^{(k)}$  on  $\Omega$ 
   using Dirichlet data prescribed on  $\partial\Omega$  as conditions; ▶ §4
6   Calculate  $\partial u^{(k)} / \partial n$  on  $\partial\Omega$  through finite differences;
7   // Exterior boundary-element solve
8   forall  $i$  do Evaluate  $\mathcal{B}[u^{(k)}](\mathbf{x}_i)$  via Eq. (15); ▶ §5.2
9   forall  $i$  do  $r^{(k)}(\mathbf{x}_i) \leftarrow \mathcal{B}[u^{(k)}](\mathbf{x}_i) - u^{(k)}(\mathbf{x}_i)$ ;
10   $\omega^{(k)} \leftarrow \text{CalcRelaxationFactor}(k, r^{(k)}, r^{(k-1)}, \omega^{(k-1)})$ ; ▶ §6.2
11  forall  $i$  do  $u^{(k+1)}(\mathbf{x}_i) \leftarrow u^{(k)}(\mathbf{x}_i) + \omega^{(k)} r^{(k)}(\mathbf{x}_i)$ ;
12   $k \leftarrow k + 1$ ;
13 until  $\sum_i (r^{(k)}(\mathbf{x}_i))^2 < \eta \sum_i (r^{(0)}(\mathbf{x}_i))^2$ ; ▶ §7.1
14  $u \leftarrow u^{(k)}$ ;
15 Function CalcRelaxationFactor( $k, r^{(k)}, r^{(k-1)}, \omega^{(k-1)}$ ):
16   if  $k = 0$  then
17     return  $1/2$ 
18   else
19     forall  $i$  do  $\delta r(\mathbf{x}_i) \leftarrow r^{(k)}(\mathbf{x}_i) - r^{(k-1)}(\mathbf{x}_i)$ ;
20     return  $-\omega^{(k-1)} \sum_i r^{(k-1)}(\mathbf{x}_i) \delta r(\mathbf{x}_i) / \sum_i \delta r^2(\mathbf{x}_i)$ ;

```

interface, or *artificial boundary*, that partitions the domain into two regions: the bounded interior domain (denoted by the superscript “−”) and the unbounded exterior domain (denoted by “+”).

The Interior Problem. Within the computational domain $\Omega \setminus \Omega_c$, the solution u^- satisfies the original generalized Poisson equation (Eq. (1a)) subject to boundary conditions defined on the cavities (Eq. (1b)). The PDE coefficients and source terms are typically defined either analytically over a finite volume, or through discrete representations, such as piecewise-linear or piecewise-constant functions on structured grids. Both sources and heterogeneous coefficients are typically confined to a finite region, and we assume here that they are encompassed within the computational domain.

The Exterior Problem. We assume that the artificial boundary is reasonably large enough to enclose all geometrical features (such as cavities) and physical heterogeneities (e.g., regions where the diffusion coefficients are spatially varying or where reaction and source terms are non-zero). By confining all such complexities to the interior, the exterior solution $u^+(\mathbf{x})$ (for $\mathbf{x} \in \mathbb{R}^d \setminus \Omega$) can now be assumed to satisfy a Laplace equation of the type

$$\nabla^2 u^+(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathbb{R}^d \setminus \Omega \quad (2)$$

subject to the far-field condition Eq. (1c). According to Green's identities, u^+ is uniquely determined by the boundary data—such as Dirichlet and/or Neumann conditions—on the artificial interface $\partial\Omega$. Once the complete boundary data is obtained through the boundary integral equation, u^+ can be evaluated anywhere in the exterior

domain, while the free-space Green's function naturally ensures that the solution satisfies the far-field condition.

Coupling Conditions. Although the domain is partitioned into two parts, the interior (u^-) and exterior (u^+) solutions are obviously not independent. Because the interface $\partial\Omega$ is virtual and purely for computational purposes, we must enforce continuity of the solution and its gradient across this boundary, yielding

$$u^- = u^+, \quad \mathbf{x} \in \partial\Omega, \quad (3a)$$

$$\frac{\partial u^+}{\partial n} = \frac{\partial u^-}{\partial n}, \quad \mathbf{x} \in \partial\Omega. \quad (3b)$$

Note that these natural conditions are also present in the approach of Johnson and Nédélec [1980] for coupling FEM with BEM.

3.3 Alternating Solves

Solving the coupled system (Eqs. (1a) to (3b)) can be achieved by tessellating Ω with a finite mesh and discretizing using Galerkin methods both the Poisson equation on the volumetric mesh and the BIE on the surface mesh, while enforcing the continuity conditions via Lagrange multipliers. This yields a large monolithic system coupling the interior potential and the boundary flux [Johnson and Nédélec 1980], whose direct factorization is often intractable as its hybrid structure (mixing sparse and dense blocks) precludes the use of direct solvers and of high-performance iterative solvers optimized specifically for *either* sparse FE/FD/FV systems *or* dense BE systems.

To address this challenge, we adopt an iterative domain coupling strategy inspired by [Kamiya et al. 1996; Elleithy and Tanaka 2003] that decomposes the monolithic problem into alternating volumetric and BEM solves. Specifically, we solve a Dirichlet problem on the interior and a Neumann problem on the exterior, a pairing that avoids the ill-posedness that can arise in the work of Elleithy and Tanaka [2003] when, for instance, coupling is enforced over the entire subdomain interface. Our approach is summarized in Alg. 1 and proceeds as follows: we solve the interior problem (Eqs. (1a), (1b) and (3a)) based on Dirichlet boundary conditions defined by the *current* boundary value u^+ , and compute the resulting boundary flux $\partial u^-/\partial n$; this flux is then used to solve the exterior problem (Eqs. (2) and (3b)) through a boundary integral equation, whose solution yields updated boundary value u^+ to be used as Dirichlet boundary data in the *next* iteration. We note that the interior solver actually builds a DtN map while the exterior builds a NtD map. While reversing the two is technically possible, it can leave the interior problem defined only up to an additive constant in the absence of any cavity or associated Dirichlet conditions, thereby requiring additional constraints and likely hindering performance.

A further distinction from prior coupling works is that our method solves the BIE for the exterior problem (which recovers boundary values from normal derivatives) *only inexactly* via a damped fixed-point iteration (Line 9, Alg. 1). Solving this step to high accuracy would be computationally expensive compared to the sparse interior system; yet, this is totally *unnecessary* for the convergence of the overall interior-exterior alternation. We support this strategy with a theoretical convergence analysis on canonical geometries (accounting for both artificial and cavity boundaries) and experiments (see Table 1), before further deriving an adaptive relaxation scheme

for general cases. As our results will demonstrate, this iterative approach proves highly effective (see Figs. 17 and 18).

In the remainder of this paper, we explain the details of the interior (Sec. 4) and exterior solves (Sec. 5), analyze and validate the convergence of our iterative algorithm (Sec. 6), and demonstrate its wide applicability to graphics applications (Sec. 7).

4 Interior Solves

In this section, we now describe how we solve the interior Poisson problem defined by Eqs. (1a), (1b) and (3a). As an illustrative numerical scheme that can be typically used to solve the interior problem, assume that we discretize the interior domain Ω using a Marker-and-Cell (MAC) grid [Harlow and Welch 1965], where each grid cell is a d -dimensional hypercube of edge length Δx . This staggered discretization offers high numerical accuracy and is well-suited for the generalized Poisson equation Eq. (1a). As illustrated in the inset, the scalar unknown u , the reaction coefficient c , and the source function f are sampled at cell centers, while the diffusion coefficient a is sampled on face centers. Accordingly, Eq. (1a) is discretized as:

$$a_{i+\frac{1}{2},j} u_{i+1,j} + a_{i-\frac{1}{2},j} u_{i-1,j} + a_{i,j+\frac{1}{2}} u_{i,j+1} + a_{i,j-\frac{1}{2}} u_{i,j-1} - (a_{i+\frac{1}{2},j} + a_{i-\frac{1}{2},j} + a_{i,j+\frac{1}{2}} + a_{i,j-\frac{1}{2}} - c_{i,j} \Delta x^2) u_{i,j} = f_{i,j} \Delta x^2, \quad (4)$$

where integer indices (e.g., (i, j)) denote cell centers while half-integer indices (e.g., $(i \pm 1/2, j)$) refer to the corresponding face centers, following standard conventions. The 3D discretization follows analogously. Eq. (4) assumes that all cells adjacent to (i, j) lie within the computational domain. However, if a neighboring cell (i', j') falls outside the domain (e.g., in the cavity or the exterior domain), its corresponding unknown $u_{i',j'}$ must be eliminated using the prescribed boundary conditions. For instance, if the boundary coincides exactly with the face between (i, j) and (i', j') , considering the boundary condition $(1 - \beta)u + \beta(\partial u/\partial n) = g$ evaluated at the face center, $u_{i',j'}$ in Eq. (4) is replaced according to the rule:

$$u_{i',j'} \rightarrow \frac{2\beta - (1 - \beta) \Delta x}{2\beta + (1 - \beta) \Delta x} u_{i,j} + \frac{2g \Delta x}{2\beta + (1 - \beta) \Delta x}. \quad (5)$$

For more complex geometries, we adopt a cut-cell approach [Gundelman et al. 2005; Ng et al. 2009; Batty et al. 2007] to impose the boundary conditions on cavity boundaries not aligned with grid faces, as in, for example, Fig. 11.

Linear Case. For now, assume that both a and c depend solely on position for simplicity. The interior problem is then assembled into a sparse symmetric linear system by applying Eq. (4) across all interior cells, incorporating boundary modifications via Eq. (5) where necessary. When the reaction coefficients satisfy $c_{i,j} \leq 0$, the system matrix is an M -matrix—and is thus positive definite. The resulting linear system enables the use of highly efficient iterative solvers, such as the preconditioned conjugate gradient (PCG) method with algebraic multigrid (AMG) preconditioning [Xu and Zikatanov 2017], as we will use in our examples. In cases where positive definiteness is not guaranteed, PCG can be replaced by general Krylov subspace methods such as BiCGSTAB or GMRES.

Handling Nonlinear Problems. If the coefficients a and c depend on the solution or its gradient, then Eq. (1a) becomes nonlinear. In this case, the interior solution is typically solved with the Newton–Raphson method. We define the state vector $\mathbf{u} \in \mathbb{R}^N$ containing the cell-centered unknowns with N being the number of cells. Similarly, $\mathbf{c}, \mathbf{f} \in \mathbb{R}^N$ represent the cell-centered reaction and source terms, while $\mathbf{a} \in \mathbb{R}^M$ contains the diffusion coefficients defined on the M faces. With $\mathbf{G} \in \mathbb{R}^{M \times N}$ denoting the discrete gradient operator, the following vector is assembled as:

$$\mathbf{F}(\mathbf{u}) = -\mathbf{G}^\top \text{diag}(\mathbf{a}(\mathbf{u})) \mathbf{G} \mathbf{u} + \text{diag}(\mathbf{c}(\mathbf{u})) \mathbf{u} - \mathbf{f}. \quad (6)$$

At the k -th Newton iteration to solve $\mathbf{F}(\mathbf{u}) = \mathbf{0}$, we update $\mathbf{u}^{(k)}$ to $\mathbf{u}^{(k+1)}$ by solving the following linear system:

$$\mathbf{J}(\mathbf{u}^{(k)}) (\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}) = -\mathbf{F}(\mathbf{u}^{(k)}), \quad (7)$$

where $\mathbf{J}(\mathbf{u}) = \partial \mathbf{F} / \partial \mathbf{u}$ denotes the Jacobian of $\mathbf{F}$. As described previously, variables from the exterior domain can be eliminated via boundary conditions similar to the linear case. Note that the action of the Jacobian on an arbitrary vector $\delta \mathbf{u}$ is given by:

$$\mathbf{J}(\mathbf{u}) \delta \mathbf{u} = \mathbf{F}(\delta \mathbf{u}) - \mathbf{G}^\top \text{diag}(\mathbf{G} \mathbf{u}) \frac{\partial \mathbf{a}}{\partial \mathbf{u}} \delta \mathbf{u} + \text{diag}(\mathbf{u}) \frac{\partial \mathbf{c}}{\partial \mathbf{u}} \delta \mathbf{u}. \quad (8)$$

Starting from $\mathbf{u}^{(0)} = \mathbf{0}$, we proceed with the Newton iterations and line search until convergence, i.e., until $\|\mathbf{F}(\mathbf{u}^{(k)})\| / \|\mathbf{F}(\mathbf{u}^{(0)})\|$ is less than a given accuracy threshold, to obtain the solution vector $\mathbf{u}$.

5 BEM Exterior Solves

In this section, we describe how to solve the exterior Laplace equation Eq. (2) subject to the Neumann boundary condition from Eq. (3b) imposed on the computational domain boundary. Leveraging the boundary element method for this case reduces the problem to finding the unknown function values on the boundary based on the known normal derivatives provided by the interior solver. The solution in the exterior domain is then fully determined via the boundary integral, which naturally satisfies the far-field condition (1c).

5.1 Representation and BIE

Since the exterior solution is assumed to be harmonic, Green’s third identity applies under a standard far-field condition (see App. B for proof). This yields a particular representation, with the convention that the unit normal vector $\mathbf{n}$ points *outward* from Ω , that reads:

$$u(\mathbf{x}) = \mathcal{V} \left[\frac{\partial u}{\partial \mathbf{n}}; \partial \Omega \right](\mathbf{x}) - \mathcal{K}[u; \partial \Omega](\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \setminus \bar{\Omega}, \quad (9)$$

where $\bar{\Omega} = \Omega \cup \partial \Omega$ is the closure of the domain, and the operators $\mathcal{V}$ and $\mathcal{K}$ denote the single- and double-layer potentials, respectively. These functionals act on a density by integrating it against Green’s function or its normal derivative along the boundary Γ , i.e.,

$$\mathcal{V}[\phi; \Gamma](\mathbf{x}) = \int_{\Gamma} \phi(\mathbf{x}') G(\mathbf{x}', \mathbf{x}) dS', \quad (10)$$

$$\mathcal{K}[\phi; \Gamma](\mathbf{x}) = \int_{\Gamma} \phi(\mathbf{x}') \frac{\partial G}{\partial \mathbf{n}'}(\mathbf{x}', \mathbf{x}) dS', \quad (11)$$

where $G(\mathbf{x}', \mathbf{x})$ is the fundamental solution of the Laplace operator satisfying $\nabla^2 G(\mathbf{x}', \mathbf{x}) = \delta(\mathbf{x}' - \mathbf{x})$ (see §A.1). Note that due to the jump properties of the double-layer potential, we have $u \equiv 0$ for $\mathbf{x} \in \Omega$ [Costabel 1987]; this implies that the exterior representation vanishes inside the domain, ensuring it does not interfere with the interior solution.

Assume that the normal derivatives on the boundary, $\partial u / \partial \mathbf{n}|_{\partial \Omega}$, are already known from the interior solver as we will detail in the next section. To fully reconstruct the solution, we must determine the unknown boundary values $u|_{\partial \Omega}$. As the evaluation point $\mathbf{x}$ approaches the boundary, the flux and potential values are known to be constrained by the following integral equation:

$$\frac{1}{2} u(\mathbf{x}) + \mathcal{K}[u; \partial \Omega](\mathbf{x}) = \mathcal{V} \left[\frac{\partial u}{\partial \mathbf{n}}; \partial \Omega \right](\mathbf{x}), \quad \mathbf{x} \in \partial \Omega, \quad (12)$$

where the coefficient $1/2$ is valid provided the boundary is Lipschitz continuous. Because the normal derivative of Green’s function constitutes a *strongly singular* kernel, its boundary integral must be interpreted in the *Cauchy principal value* sense [Kanwal 2013].

5.2 Fast Updates Through Fixed-Point Iteration

Next, we explain how to discretize and solve Eq. (12) for fast updates of boundary values.

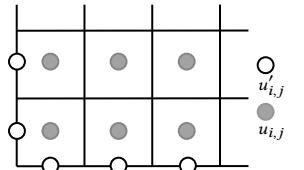
Discretization. Since the interior solver employs a MAC grid, the artificial boundary $\partial \Omega$ is naturally discretized by the outermost layer of faces of the computational grid. We discretize the boundary integral equation Eq. (12) using a piecewise constant collocation method, similar to the work of Chen et al. [2025]. With this approximation, we assume both u and $\partial u / \partial \mathbf{n}$ constant over each boundary facet Γ_j and their values are sampled at face centers. For each target facet index i centered at $\mathbf{x}_i$, the discrete boundary integral equation is thus enforced as:

$$\frac{1}{2} u(\mathbf{x}_i) + \sum_j \mathcal{K}[u(\mathbf{x}_j); \Gamma_j](\mathbf{x}_i) \approx \sum_j \mathcal{V} \left[\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}_j); \Gamma_j \right](\mathbf{x}_i). \quad (13)$$

A challenge arises because the MAC grid stores potential values at cell centers, whereas the boundary integral method requires values on the boundary faces. To bridge this gap, we introduce an additional set of degrees of freedom (DoFs) for u exactly on the boundary.

As illustrated in the inset, $\{u_{i,j}\}$ denotes the original cell-centered unknowns of the (interior) Poisson solver, and $\{u'_{i,j}\}$ denotes the new DoFs for the BIE solver, stored at face centers. Once these values are updated, the normal derivative on a boundary face can then be approximated using a one-sided finite difference between the face and the neighboring cell center, yielding $2(u'_{i,j} - u_{i,j}) / \Delta x$. Note that a corner cell is adjacent to multiple boundary facets.

Efficient Updates of u' . To simplify the notation hereafter, we drop the prime superscript ($'$) previously introduced to distinguish the boundary DoFs from the interior ones. Eq. (13) is an asymmetric and dense linear system, for which direct solvers (such as LU decomposition) become computationally prohibitive as the number of boundary DoFs grows. Even with efficient iterative solvers such as preconditioned GMRES [Chen et al. 2025] or Neumann series expansion [Ni et al. 2024], the computational cost remains higher than that of the interior, sparse Poisson solver: this is primarily due to repeated dense matrix-vector products, which are time consuming even when optimized with hierarchical summation techniques. Crucially, however, solving this system to *high accuracy* is *not* necessary for global convergence—akin to inexact Newton methods. Therefore, we opt for a cost-effective adapted fixed-point iteration,



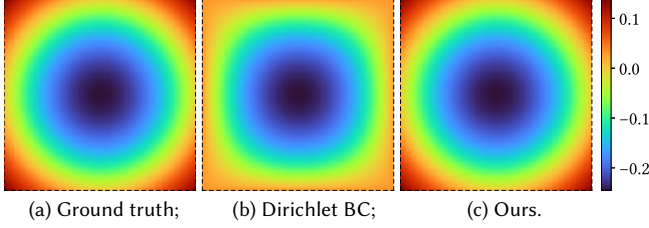


Fig. 3. **One-step convergence.** For a uniform circular (2D) or spherical (3D) source centered at the origin with radius 0.9, we solve a *classical Poisson problem without cavities* on the domain $[-1, 1]^d$ using a 256^d grid. For this class of problems, our method converges in a single iteration and, compared with homogeneous Dirichlet boundary conditions, reduces the solution error by three to four orders of magnitude: in 2D, the L^2 error decreases from 2.23×10^{-2} to 3.76×10^{-6} , and the L^∞ error from 1.08×10^{-1} to 1.19×10^{-5} ; in 3D, the L^2 error decreases from 1.74×10^{-2} to 5.76×10^{-6} , and the L^∞ error from 7.15×10^{-2} to 1.52×10^{-5} . The figure visualizes the solutions in the 2D case through false colors.

which strikes a good balance between BEM computational costs and convergence speed. The method proceeds as follows: during the k^{th} interior-exterior alternation, we move the double-layer term to the right-hand side of Eq. (13) and update u via a single step:

$$u^{(k+1)}(x_i) = \mathcal{B}[u^{(k)}](x_i), \quad (14)$$

where $\mathcal{B}$ is the boundary operator defined as

$$\begin{aligned} \mathcal{B}[\phi](x) &= 2 \left(\mathcal{V} \left[\frac{\partial \phi}{\partial n}; \partial \Omega \right] - \mathcal{K}[\phi; \partial \Omega] \right)(x) \\ &\approx 2 \sum_i \left(\mathcal{V} \left[\frac{\partial \phi}{\partial n}(x_i); \Gamma_i \right] - \mathcal{K}[\phi(x_i); \Gamma_i] \right)(x), \end{aligned} \quad (15)$$

for all boundary point $x \in \partial \Omega$. To control global convergence, we introduce a *relaxation factor*, a standard technique in iterative schemes; that is, we determine the final update rule for the exterior solve by interpolating between the current iterate $u^{(k)}$ and the intermediate update from Eq. (14), i.e.,

$$u^{(k+1)}(x_i) = (1 - \omega^{(k)}) u^{(k)}(x_i) + \omega^{(k)} \mathcal{B}[u^{(k)}](x_i), \quad (16)$$

where $\omega^{(k)} > 0$ denotes the relaxation factor chosen at the k -th iteration—Sec. 6.2 provides an adaptive strategy for choosing it. This updated value serves as the boundary constraint for the subsequent interior solve, which is assembled from Eq. (5). The process then repeats until the difference between $\mathcal{B}[u^{(k)}]$ and $u^{(k)}$ drops below a prescribed tolerance, signaling that the interior and exterior solutions are now compatible and the iterations have converged to the correct solution.

6 Convergence Analysis and Adaptive Relaxation

While conceptually simple, our interior-exterior alternating approach is robust and efficient in practice as we now demonstrate. In particular, we prove convergence in one single step for a special case in Sec. 6.1, and discuss the behavior of the dynamic relaxation scheme for general cases in Sec. 6.2.

6.1 One-Step Convergence for Poisson BVP and Beyond

We analyze the convergence behavior of our iterative solver from a boundary perspective. For now, we assume the interior PDE problem

to be linear. Let u denote the exact solution of the original problem (Eqs. (1a) to (1c)) on the artificial interface $\partial \Omega$. For the k^{th} iteration, we define the error as $\varepsilon^{(k)} = u^{(k)} - u$. Since u is a fixed point of $\mathcal{B}$ (see Eq. (14)) and by linearity, we have

$$\mathcal{B}[u^{(k)}] = \mathcal{B}[u] + \mathcal{B}[\varepsilon^{(k)}] = u + \mathcal{B}[\varepsilon^{(k)}]. \quad (17)$$

Neglecting numerical discretization errors and substituting this identity into the update rule (Eq. (16)), we obtain an equation for the evolution of the error as:

$$\varepsilon^{(k+1)} = \left((1 - \omega^{(k)}) \mathcal{I} + \omega^{(k)} \mathcal{B} \right) [\varepsilon^{(k)}], \quad (18)$$

where $\mathcal{I}$ is the identity operator. Consequently, the convergence of the iteration depends on the spectrum of the operator $\mathcal{B}$.

For the *classical Poisson equation* with no cavities, the third Green's identity in Ω yields the following principal value formulation for the boundary solution:

$$u(x) = -\mathcal{B}[u](x) + 2 \int_{\Omega} f(x') G(x', x) dV', \quad x \in \partial \Omega, \quad (19)$$

where the negative sign preceding $\mathcal{B}$ arises because of our choice of outwards normal vectors on $\partial \Omega$, opposite in the interior formulation compared to the exterior formulation used in Eq. (12). Since $u^{(k)}$ is also a solution to the interior Poisson problem (solved with high precision), it must satisfy this interior Green's identity. Subtracting Eq. (19) for the exact solution u from the corresponding equation for the iterate $u^{(k)}$ yields:

$$\varepsilon^{(k)} = -\mathcal{B}[\varepsilon^{(k)}], \quad (20)$$

which implies that -1 is an eigenvalue of $\mathcal{B}$. Substituting this result into Eq. (18) leads to

$$\varepsilon^{k+1} = (1 - 2\omega^{(k)}) \varepsilon^{(k)}. \quad (21)$$

As a consequence, if we choose $\omega^{(0)} = 1/2$, the error for this simple Poisson equation vanishes after a single iteration (see Fig. 3), regardless of the initial guess $u^{(0)}$. In App. C, we present spectral analyses of the operator $\mathcal{B}$ for several non-classical problems defined on canonical geometries. These results provide deeper insights into the convergence behavior of the proposed iterative method within more general settings.

Discussion. It is important to note that the boundary operator $\mathcal{B}$ relies on the boundary normal derivative, which is intrinsically linked to the interior solution. Formally, this relationship is:

$$\mathcal{B}[\phi] = 2 (\mathcal{V} \circ \mathcal{T} - \mathcal{K})[\phi; \partial \Omega], \quad (22)$$

where $\mathcal{T}$ maps prescribed Dirichlet values on $\partial \Omega$ to their corresponding Neumann values (normal derivatives), subject to the interior governing equations. In our implementation, $\mathcal{T}$ is realized via the MAC-grid Poisson solver. For general cases such as linear problems with non-canonical cavities or even nonlinear Poisson equations, analyzing the spectrum of $\mathcal{B}$ becomes significantly more complex. Nevertheless, from a qualitative perspective, these generalized problems can be viewed as perturbations of the classical Poisson BVP. For most reasonable cases, we therefore expect the spectral properties of $\mathcal{B}$, and consequently the convergence behavior of our approach, to remain comparable to the classical case. This observation enables us to expect convergence through the use of an adaptive relaxation strategy, as detailed below.

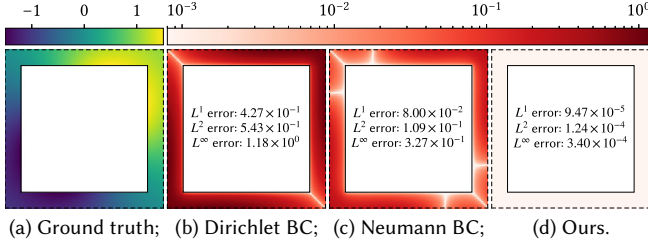


Fig. 4. **Large cavities.** For this exterior Dirichlet problem of the Laplace equation in 2D, a square of side length 0.8 within a computational domain $[-1, 1]^2$ is given Dirichlet conditions on its boundary as $u = (x+y)/(x^2+y^2)$, which is therefore itself the analytical expression of its solution owing to its zero Laplacian. Even in this extreme setting where the computational domain is quite degenerate, our method for a 256^2 grid converges after four boundary-condition updates. In contrast, both homogeneous Dirichlet and Neumann ABCs lead to errors that are 10^3 to 10^4 times larger. Here, (a) visualizes the solution, while (b)–(d) show the absolute error.

6.2 Adaptive Relaxation

As explained in Sec. 6.1 and App. C, in most practical cases of linear problems, the eigenvalues of the operators $\mathcal{B}$ cluster around -1 . However, this behavior may change drastically in more challenging scenarios. For instance, when a cavity becomes so large that it comes very close to the boundary of the computational domain (thus making the *effective* computational domain $\Omega \setminus \Omega_c$ very small), the spectrum of $\mathcal{B}$ can shift significantly. In such cases, the fixed relaxation $\omega^{(k)} = 1/2$ is no longer optimal and may even lead to divergence of the iterative method. To address this, we employ an adaptive, dynamic update of the relaxation factor based on *Aitken's delta-squared process* [Aitken 1927].

Consider the iteration operator from step k , denoted as $\mathcal{L}^{(k)} := (1 - \omega^{(k)})\mathcal{I} + \omega^{(k)}\mathcal{B}$, which acts on functions defined on $\partial\Omega$. The error evolution from Eq. (18) can be rewritten as

$$u^{(k+1)} - u = \mathcal{L}^{(k)}[u^{(k)} - u]. \quad (23)$$

If we tentatively perform one additional iteration while keeping the relaxation factor unchanged, we obtain

$$\tilde{u}^{(k+2)} - u = \mathcal{L}^{(k)}[u^{(k+1)} - u]. \quad (24)$$

Assuming that the error term $u^{(k)} - u$ is an eigenfunction of the current operator $\mathcal{L}^{(k)}$ (an assumption that becomes progressively more accurate as the iterations near convergence), it follows that for each point on the boundary, the update of $u(x)$ at step $k+1$ satisfies

$$\frac{u^{(k+1)}(x) - u(x)}{u^{(k)}(x) - u(x)} = \frac{\tilde{u}^{(k+2)}(x) - u(x)}{u^{(k+1)}(x) - u(x)}, \quad x \in \partial\Omega. \quad (25)$$

From this equation, we solve for $u(x)$ from $u^{(k)}$, $u^{(k+1)}$, and $\tilde{u}^{(k+2)}$ and use the result as the new $u^{(k+2)}$ to improve on $\tilde{u}^{(k+2)}$, i.e.,

$$u^{(k+2)} = u^{(k+1)} - \frac{(\tilde{u}^{(k+2)} - u^{(k+1)})(u^{(k+1)} - u^{(k)})}{(\tilde{u}^{(k+2)} - u^{(k+1)}) - (u^{(k+1)} - u^{(k)})}. \quad (26)$$

Noting that $\mathcal{L}^{(k)} - \mathcal{I} = \omega^{(k)}(\mathcal{B} - \mathcal{I})$ and defining the residual to be expressed as $r^{(k)} := \mathcal{B}[u^{(k)}] - u^{(k)}$, Eq. (26) can be rewritten as

$$u^{(k+2)} = u^{(k+1)} - \frac{\omega^{(k)} r^{(k)}}{r^{(k+1)} - r^{(k)}} r^{(k+1)}, \quad (27)$$

which effectively corresponds to a different relaxation factor than the original fixed $\omega^{(k)}$.

In general, the assumption that $u^{(k)} - u$ is an eigenfunction does not strictly hold. Nevertheless, motivated by the expression of

Eq. (27), we can derive an update formula for the next relaxation factor in a least-squares sense through

$$\begin{aligned} \omega^{(k+1)} &= \arg \min_{\alpha \in \mathbb{R}} \left\| r^{(k)} - \alpha (r^{(k+1)} - r^{(k)}) \right\|^2 \omega^{(k)} \\ &= - \frac{\langle r^{(k)}, r^{(k+1)} - r^{(k)} \rangle}{\|r^{(k+1)} - r^{(k)}\|^2} \omega^{(k)}, \end{aligned} \quad (28)$$

where the inner product of residual functions is evaluated via a summation over the boundary faces, i.e., $\langle \phi, \psi \rangle \approx \sum_i \phi(x_i) \psi(x_i) (\Delta x)^{d-1}$. In light of the convergence analysis in Sec. 6.1, we keep $\omega^{(0)} = 1/2$ as a reasonable initial guess. Finally, we note that the resulting acceleration strategy is equivalent to the vector-form Aitken accelerator for fixed-point iteration from Irons and Tuck [1969], as reformulated more recently by Birken et al. [2015].

7 Results

In this section, we discuss implementation details, before demonstrating the broad applicability of our solver to computer graphics tasks, and providing further evaluations of our method.

7.1 Implementation Details

Our numerical approach to open boundaries was implemented in C++, and all our reported timings were obtained on a desktop equipped with an Intel® Core™ i9-14900K processor (3.2 GHz) and 64 GB of RAM. Our implementation follows the overall algorithmic procedure summarized in Alg. 1, with just a few key implementation details that we elaborate upon next. The code is available at <https://gitlab.inria.fr/geomerix/public/OpenPoisson>.

The Interior Solver. At each iteration, the interior Poisson problem is solved using AMG-preconditioned Krylov subspace solvers from AMGCL [Demidov 2019] and Eigen [Guennebaud et al. 2010], with conjugate gradient for symmetric positive-definite systems and GMRES for non-symmetric or indefinite cases. Our framework is largely agnostic to the choice of linear solver or spatial discretization (FEM, FDM, or FVM), enabling easy substitution of alternative schemes to suit specific applications or requirements.

The Exterior Solver. Evaluating single-layer ($\mathcal{V}$) and double-layer ($\mathcal{K}$) potentials for boundary updates involves integrating Green's functions and their normal derivatives over constant elements (line segments in 2D, square facets in 3D). Although closed-form expressions exist (and given in App. A.2), direct summation is computationally prohibitive in 3D due to quadratic scaling. We therefore accelerate computations using the fast multipole method (FMM) implemented in the FMM3D library [Askham et al. 2024], approximating single- and double-layer potentials with monopole and dipole sources, respectively. Accuracy is maintained via 4 Gauss-Legendre quadrature points per face, and singular self-contributions are corrected by replacing the quadrature term with its analytical value.

Convergence Criterion. As discussed in Sec. 6.1, convergence of the algorithm is reached when the fixed-point condition (that is, $u^{(k)}(x) \approx \mathcal{B}[u^{(k)}](x)$) is fulfilled for all $x \in \partial\Omega$, meaning that the norm of the residual $r^{(k)}$ must be sufficiently small throughout the boundary of the computational domain. In practice, the magnitude of the solution $u(x)$ depends strongly on the specific problem setup,

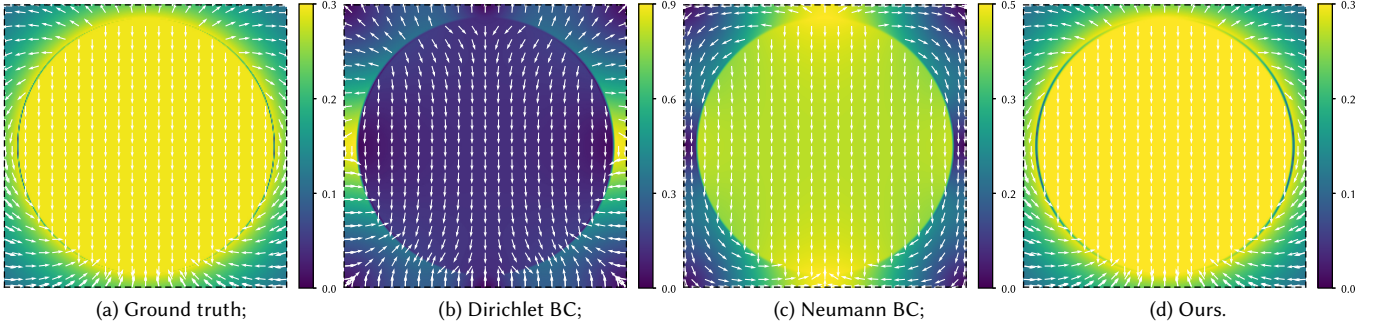


Fig. 5. **Spherical magnetic inclusion.** For an inclusion (a disk in 2D or a ball in 3D) of radius $R = 0.9$ centered at the origin with an homogeneous magnetic susceptibility $\chi=1$ inside the inclusion and $\chi=0$ in the surrounding medium, embedded in the domain $[-1, 1]^d$, we solve a heterogeneous Laplace problem on a 256^d grid, with solution values aligned at the origin. A uniform background magnetic field is applied vertically upward. Compared with homogeneous Dirichlet and Neumann BCs, our method converges in 3 iterations and reduces the L^2 error from 1.65×10^{-1} (Dirichlet) and 1.07×10^{-1} (Neumann) to 1.04×10^{-2} in 2D, and from 6.99×10^{-2} (Dirichlet) and 8.93×10^{-1} (Neumann) to 7.21×10^{-3} in 3D. The figure visualizes the 2D case, showing the magnitude of the gradient field $\|\nabla u\|$ in false colors and overlaying the vector field $-\nabla u$ as arrows.

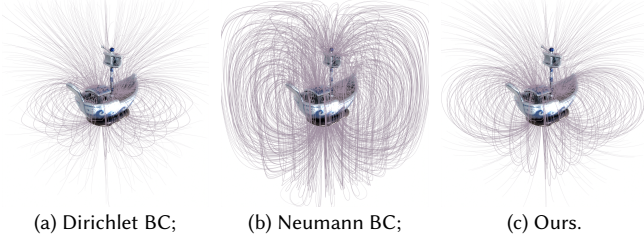


Fig. 6. **Magnetizing a boat.** We visualize the magnetic field lines induced by a metallic boat ($\chi=1$) when a uniform vertical magnetic field is imposed. Computations are performed on a 256^3 grid, with our method converging after two iterations. Note that both Dirichlet and Neumann artificial boundary conditions introduce pronounced artifacts; only our result correctly captures the natural decay of the magnetic field at infinity.

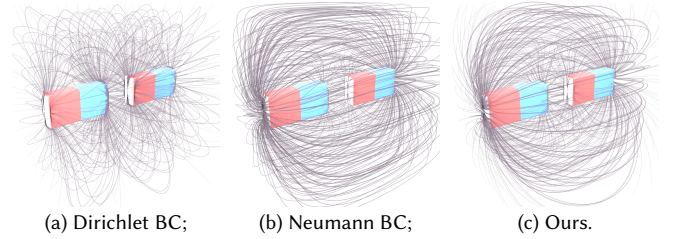


Fig. 7. **Attracting magnets.** We visualize the magnetic field of two attracting magnets ($\chi=1$) in a uniform horizontal magnetic field. Computation are performed on a 256^3 grid, with our method converging after two iterations. Unlike the spurious artifacts induced by Dirichlet (a) or Neumann (b) boundary conditions, our result correctly captures the decay behavior of the magnetic field at infinity.

so relying on an absolute error threshold for checking convergence lacks flexibility and robustness. We use the initial residual $r^{(0)}$ as a reference and establish a *relative* convergence criterion via:

$$\|r^{(k)}\|^2 < \eta \|r^{(0)}\|^2, \quad (29)$$

where η is a user-specified tolerance. In our experiments, we have found $\eta=10^{-6}$ provides a reasonable balance between accuracy and computational cost.

7.2 Applications

We now review a number of graphics applications for which our open-boundary solver is particularly suitable.

7.2.1 Magnetostatics Problems. A magnetostatic field is a magnetic field generated by steady electric currents or permanent magnets. It is governed by Maxwell's equations under the magnetostatic assumption, whereby time-dependent effects and the electric field are neglected. When the magnetostatic field is expressed as the negative gradient of a magnetic scalar potential, its governing equations can typically be reduced to a variable-coefficient Poisson equation. This formulation has been widely adopted in computer graphics for applications such as ferrofluid simulation and magnetic field-line visualization [Ni et al. 2020, 2024; Huang et al. 2019; Huang and Michels 2020; Shao et al. 2023].

Magnetization. Placing a magnetizable object within an external magnetic field induces its *magnetization*, which in turn generates an auxiliary field. Calculating this magnetization—and the resulting total field—is critical for analyzing the interactions between magnetic bodies. Let the induced magnetic field be denoted by $H_a = -\nabla\phi$, where ϕ is the magnetic scalar potential. The corresponding governing equation is given by

$$\nabla \cdot (1 + \chi) \nabla \phi = \nabla \cdot \chi H_e, \quad (30)$$

where H_e denotes the known external magnetic field, and χ is the magnetic susceptibility, which is nonzero within the magnetizable media and vanishes outside. In the absence of other surrounding media, the magnetic field is unbounded and propagates to infinity, naturally requiring a far-field boundary condition.

We validate our method on a classical benchmark problem: a homogeneous medium sphere (or a disk in 2D) placed in a uniform external magnetic field, as shown in Fig. 5. The disk or sphere is concentric with a square or cubic computational domain discretized on a grid of resolution 256^d , and the ratio between the diameter of the disk/sphere and the side length of the domain is 0.9. The magnetic susceptibility is set to 1 inside the medium and 0 outside. To regularize the discontinuity across the interface, we smooth the jump using a smooth Heaviside step function [Ni et al. 2020], with an interface bandwidth of $4\Delta x$. Under this setting, the additional magnetic field

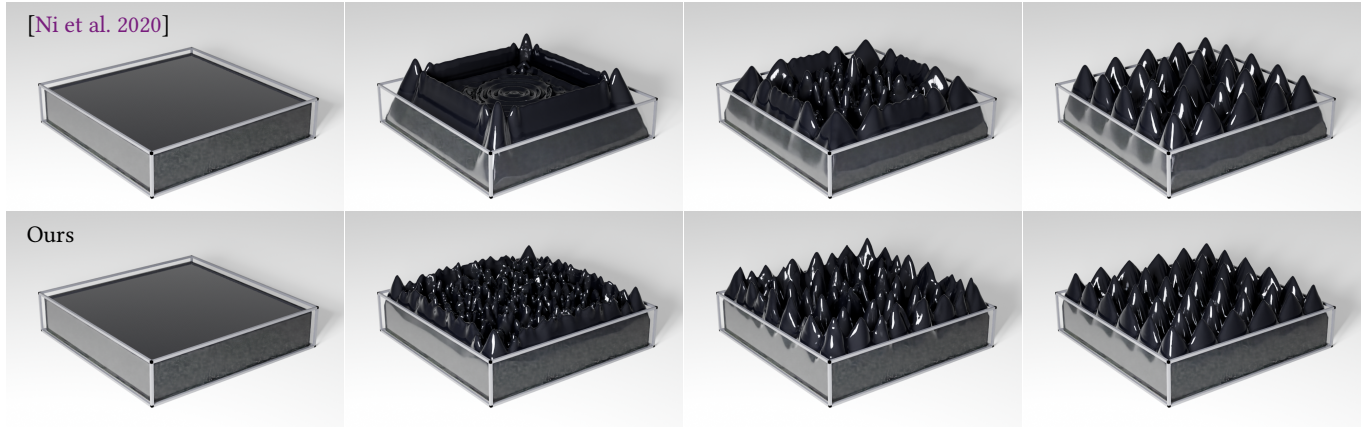


Fig. 8. **Ferrofluid simulation.** Comparison of the Rosensweig instability in a ferrofluid ($\chi=0.33$) simulated on a $192 \times 144 \times 192$ grid. Top row: a grid-based magnetic field solver with ad-hoc magnetic shielding [Ni et al. 2020]; bottom row: the same solver enhanced with our iterative coupling scheme. From left to right, the results are rendered at $t = 0$ s, 0.1 s, 0.2 s, and 0.8 s. Due to incorrectly specified boundary conditions, the top-row simulation fails to keep the fluid attached to the solid walls, resulting in only 5×5 spikes at equilibrium, whereas the bottom row produces 6×6 spikes.

H_a obtained using ad-hoc Dirichlet or Neumann boundary conditions (as used in ferrofluid simulation by Ni et al. [2020]) deviates significantly from the analytical solution: the Dirichlet condition produces severely overestimated field strengths near the equator (with the external field direction taken as the polar axis), while the Neumann condition leads to extreme overestimation near the poles. In contrast, our method converges within only three iterations in both two and three dimensions, and the results agree closely with the ground truth. Over the entire domain $[-1, 1]^d$, our approach achieves root-mean-square errors (RMSE) in the potential gradient that are one to two orders of magnitude smaller than those obtained with Dirichlet or Neumann ABCs.

Magnetized Objects. Fig. 6 visualizes the magnetic field lines of the internal magnetic field induced in a metallic boat by a vertically oriented external field in three dimensions. This example more intuitively illustrates the non-physical artifacts introduced by ad-hoc boundary conditions: although all boundary treatments qualitatively capture the concentration of magnetic field lines near sharp tips of the object, noticeable discrepancies arise away from the object and close to the artificial boundary. With Dirichlet boundary conditions, the field lines exhibit nonsmooth, large-angle deflections, while under Neumann boundary conditions the field lines are unable to leave the computational domain (known as *magnetic shielding*). In contrast, our results correspond to a faithful restriction of the infinite-domain solution onto the finite computational domain. Similar behavior can also be clearly observed in the case of two magnets magnetized by a horizontal external field and attracting each other (Fig. 7), where the Dirichlet BC produces highly disordered field lines while the Neumann BC introduces pronounced spurious boundary effects changing the overall solution drastically.

Ferrofluid Simulation. The use of artificial Neumann boundary conditions inducing magnetic shielding is a major problem in the ferrofluid simulation framework of Ni et al. [2020]. We easily overcome this limitation in dynamic simulations by wrapping our iterative

method around their grid-based magnetostatic solver. As illustrated in Fig. 8, when a quiescent ferrofluid surface develops spike-like structures under an externally applied uniform vertical magnetic field (a phenomenon known as the *Rosensweig instability* [Cowley and Rosensweig 1967]), simulations with spurious magnetic shielding fail to generate spikes tangent to the solid boundary (i.e., the boundary of the computational domain) because of the imposed Neumann conditions. In contrast, our approach removes these artifacts within only *three* iterations, producing an additional outer ring of spikes. Each step of magnetic field computation uses the magnetic scalar potential ϕ from the last step as the initial guess, while the boundary values are initialized to zero. The enforcement of far-field conditions at each time step of this dynamic example results in an artifact-free simulation that is consistent with the one obtained using a pure BEM for the same scenario [Ni et al. 2024, Fig. 5]. For the animations, please refer to the supplementary video.

Solar Coronal Field. In cases where the magnetic susceptibility is nonlinear or unobservable, such as in astrophysical objects, it is generally infeasible to directly solve Eq. (30). Nevertheless, since Eq. (30) always degenerates to the Laplace equation $\nabla^2 \phi = 0$ in vacuum (owing to Gauss’s law for magnetism that implies $\nabla \cdot \mathbf{H}_e = 0$), the magnetic field computation can be reformulated if the magnetic field values on the surfaces of all magnetic bodies (forming cavities in the domain) are known. Specifically, given the normal derivative of ϕ on these surfaces, the problem reduces to a Neumann exterior BVP for the Laplace equation, known as the potential field source surface (PFSS) model [Altschuler 1969; Priest 2014, Sec. 3.3]. As shown in Fig. 9, we compute the magnetic field distribution around the Sun using the zero-point-corrected magnetogram synoptic map provided by the U.S. National Solar Observatory [2026], and visualize the resulting magnetic field lines. The figure demonstrates that our method faithfully reconstructs the local structure of the solar coronal magnetic field, whereas Dirichlet and Neumann artificial boundary conditions introduce non-physical magnetic leakage or spurious magnetic shielding.

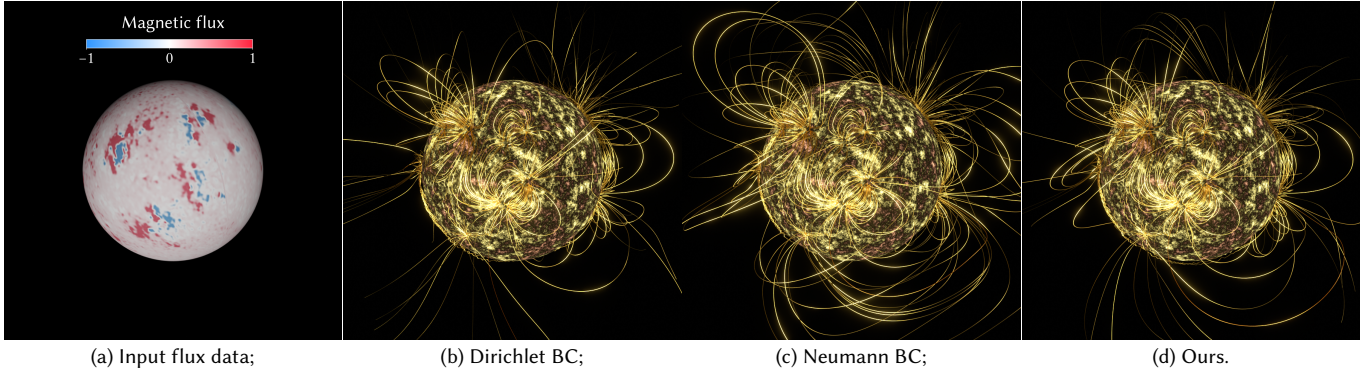


Fig. 9. **Solar magnetic field.** Using magnetic flux data on the solar surface obtained from real observations (a) as input—where red indicates outward flux and blue indicates inward flux—we compute the magnetic field distribution in the vicinity of the solar corona based on the PFSS model on a 256^3 grid. In the ideal case, all magnetic field lines emerging from the surface eventually return to it, and their strength naturally decays to zero at infinity. Our method (d) converges to this desired behavior after two iterations, whereas the Dirichlet (b) and Neumann (c) ABCs suffer from different forms of boundary effects. In all results, the magnetic field lines originate from the same starting points to ease visual assessment.

7.2.2 Fluid Dynamics Problems. In computational fluid dynamics (CFD) and fluid animation for computer graphics, Poisson-like BVPs play a central role. The widespread appearance of the Laplace operator in fluid models primarily stems from two sources: the divergence-free constraint imposed on the velocity field by incompressibility, and the isotropy of pressure, which gives rise to gradient forces that resist volumetric deformation.

Leray Projection. Fluid simulation methods based on operator splitting decompose the velocity update within a single time step into two stages: advection and velocity projection [Stam 1999]. After advection, the intermediate velocity field $\mathbf{u}^*$ is generally not divergence-free. Enforcing incompressibility, therefore, requires solving the following Poisson problem

$$\begin{cases} \nabla^2 p^* = \nabla \cdot \mathbf{u}^*, & \text{in the fluid,} \\ \mathbf{u}^* - \nabla p^* = \mathbf{u}_s, & \text{on the fluid–solid interface,} \end{cases} \quad (31a)$$

followed by the update $\mathbf{u} \leftarrow \mathbf{u}^* - \nabla p$, which extracts the divergence-free component of $\mathbf{u}^*$. This operation is commonly referred to as a *Leray projection*. Here, p^* denotes a scaled pressure that incorporates the time-step size and fluid density, and $\mathbf{u}_s$ is the solid velocity, which vanishes for stationary boundaries. In many practical scenarios—such as a plume of smoke rising in the atmosphere or a submarine moving in the deep ocean—the simulated domain has no explicit physical boundary. In such cases, an accurate projection is only well-defined under far-field boundary conditions at infinity, whereas imposing homogeneous Dirichlet or Neumann conditions for the pressure on an artificial computational boundary constitutes merely an ad-hoc approximation used purely for numerical convenience.

Fig. 10 visualizes the streamlines of a flow field around a dolphin. The input field is generated from a uniform flow whose incompressibility has been violated by perturbations near the dolphin. We compare the effects of performing the Leray projection using the conventional approach with Neumann boundary conditions imposed on the computational domain boundary and using our method. Artificial boundary conditions cause the streamlines near the outer boundary of the domain to bend erroneously towards the boundary, leading the entire flow field to deviate from the behavior

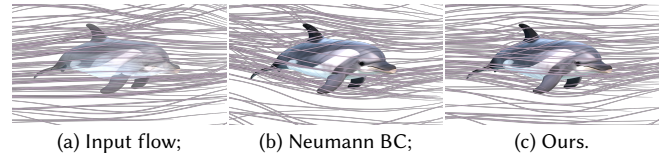


Fig. 10. **Leray projection.** We randomly generate a velocity field with nonzero divergence on a $384 \times 216 \times 216$ grid (see streamlines in (a)) and place a dolphin as a solid boundary. By solving a Poisson equation for pressure with interior Neumann boundary conditions, we enforce incompressibility of the flow. Imposing $\partial p / \partial n = 0$ on the computational domain boundary (b) as often done in graphics causes the streamlines to be incorrectly “attracted” to the boundary, whereas our method, converging in only two iterations, produces streamlines in an effectively unbounded ocean (c).

expected in unbounded space. Note that the streamline starting points are identical in all images to help with visual comparison.

Velocity from Vorticity. In contrast to methods that directly evolve the velocity field $\mathbf{u}$, vortex methods operate on the vorticity field $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, thereby better preserving the angular momentum of the flow. However, for advection and measurement, the velocity field must be reconstructed from the vorticity at the end of each time step. Two reconstruction strategies are commonly used in practice: one based on the Biot–Savart law [Angelidis and Neyret 2005], and the other based on solving a Poisson equation [Bridson 2015, Sec. 14.1]. The former evaluates the contribution of Green’s functions induced by vorticity “charges” at all sample points, typically with a mollified kernel to remove the singularity. The latter introduces a vector potential $\boldsymbol{\psi}$ (or a scalar stream function in two dimensions) satisfying the gauge condition $\nabla \cdot \boldsymbol{\psi} = 0$ and the relation $\nabla \times \boldsymbol{\psi} = \mathbf{u}$. Using standard vector identities, this leads to the Poisson equation

$$\nabla^2 \boldsymbol{\psi} = -\nabla \times \nabla \times \boldsymbol{\psi} = -\boldsymbol{\omega}, \quad (32)$$

which is then solved numerically. Both reconstruction strategies, however, exhibit certain limitations due to the way boundary conditions are handled, which our method effectively addresses.

The Biot–Savart formulation is derived in free space and therefore naturally induces a far-field condition at infinity. However, it does not account for the influence of solid boundaries. Among all velocity

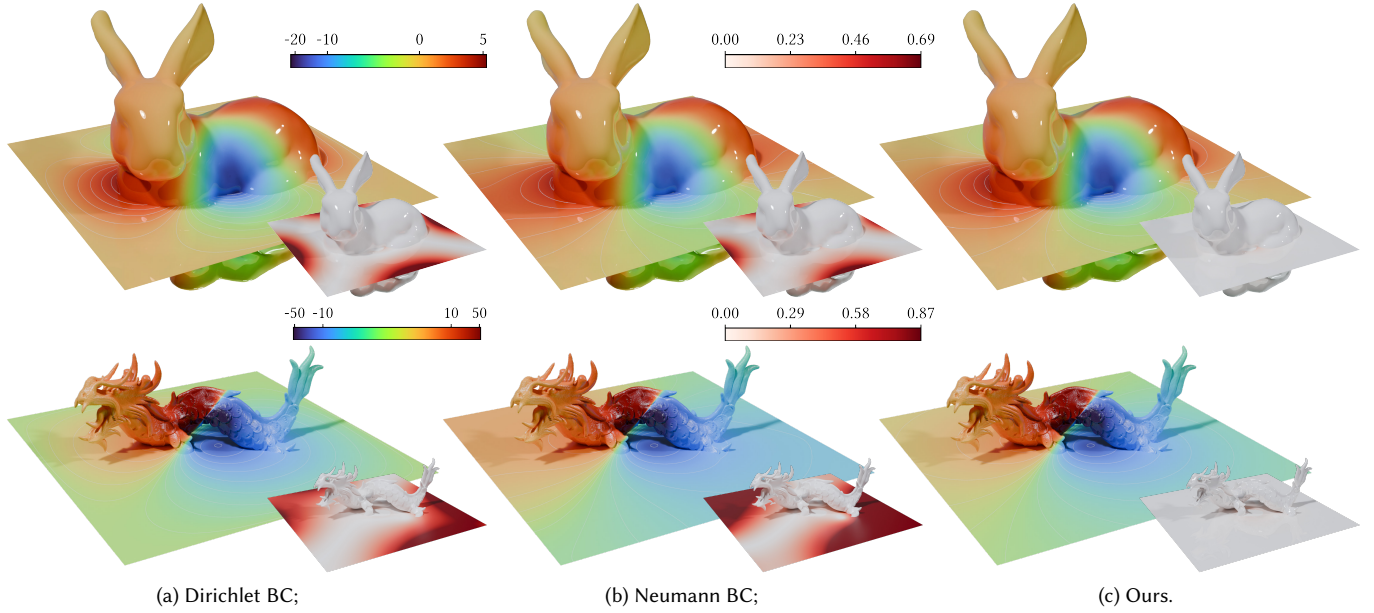


Fig. 11. **Exterior Dirichlet problem.** For an exterior Dirichlet problem of the Laplace equation, we show two examples, one prescribing a function $u = (x^2 - z^2)/(x^2 + y^2 + z^2)^{2.5}$ on a bunny-shaped model and one prescribing $u = (x + y + z)/(x^2 + y^2 + z^2)^{1.5}$ on a dragon-shaped domain as respective Dirichlet boundary conditions. We compare the performance of our method with that of Dirichlet and Neumann ABCs on a 256^3 MAC grid. We visualize the resulting field values with false colors and overlay an inset showing the absolute error between the numerical solution and the ground truth for a middle slice of the computational domain. Both Dirichlet and Neumann ABCs exhibit significant boundary-induced errors. For the bunny, the volume-averaged L^2 error over the entire computational domain reduces from 2.11×10^{-1} (Dirichlet) and 1.94×10^{-1} (Neumann) to 1.60×10^{-2} (ours). For the dragon, the volume-averaged L^2 error reduces from 5.14×10^{-1} (Dirichlet) and 9.27×10^{-1} (Neumann) to 3.27×10^{-2} (ours).

fields $\mathbf{u}$ satisfying $\nabla \times \mathbf{u} = \boldsymbol{\omega}$, selecting the one that is consistent with the boundary conditions of the simulated scene requires an additional projection step [Bridson 2015, Sec. 14.3.7]. This projection is analogous to the one used in Stable Fluids (cf. Eq. (31)), except that the velocity field is already divergence-free in the fluid interior. To remain consistent with the far-field assumption of the Biot–Savart law, one must solve a Neumann exterior BVP for the Laplace equation, which is challenging for conventional numerical methods. As shown in Fig. 12, a uniform flow in an unbounded domain is deflected by the presence of a solid boundary, namely the whale. Based on the undisturbed flow, we impose Neumann boundary conditions on the whale surface for the pressure and solve an exterior Laplace problem, after which the flow field is corrected using the negative gradient of the pressure. The resulting streamlines are free from any artificial computational domain influence.

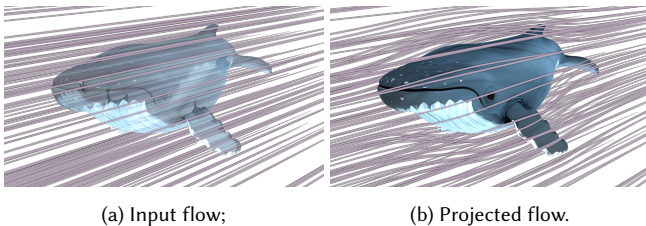


Fig. 12. **Uniform flow around a whale.** In four iterations, our method successfully makes the incompressible flow aware of solid boundaries upon a $384 \times 216 \times 216$ computational grid, without introducing any artificial domain truncation. The whale covers about 80% of the domain size.

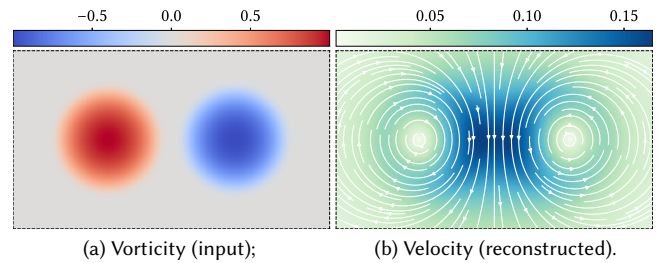


Fig. 13. **From vorticity to velocity.** From an opposite pair of vortices (a), our method correctly recovers the stream function corresponding to the flow in an unbounded domain after a single iteration on a 480×270 grid, visualized through its velocity field (b).

The Poisson-based method, on the other hand, benefits from the locality of differential operators and handles solid boundaries more easily. However, due to the bounded-domain discretization, it cannot account for, and accurately reproduce, the far-field behavior at infinity. Our method provides a practical remedy to this issue: with only a few iterations, we can convert a stream function (or vector potential) defined on a bounded domain into one that is consistent with open-space conditions. In the absence of solid boundaries, even a single iteration suffices. See Fig. 13 for an example.

Vortex-In-Cell Simulation. We also evaluate the performance of our method in dynamic fluid simulations within the vortex-in-cell framework of Selle et al. [2005], where vortex particles are advected and then transferred to a grid to reconstruct the velocity field. As

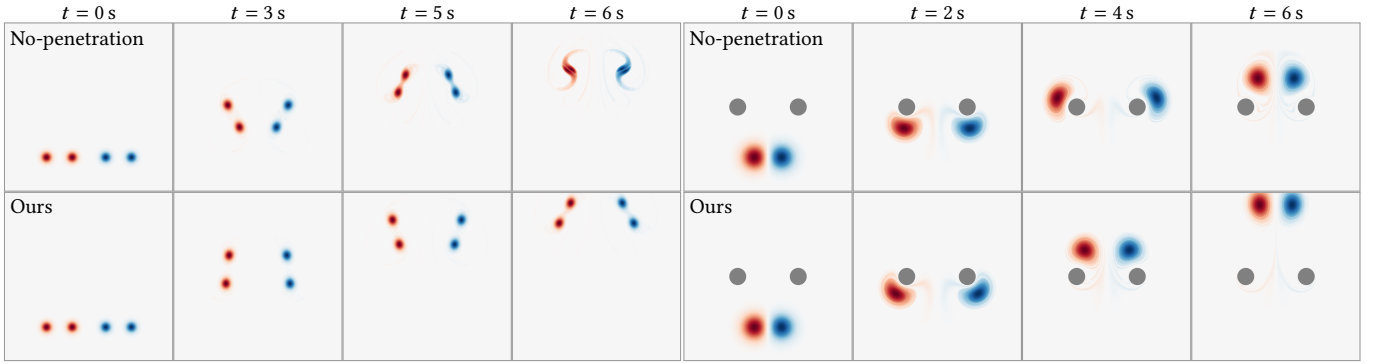


Fig. 14. **Vortex-In-Cell simulation.** We conduct two simulation scenarios on a 512×512 grid. The top row shows results with no-penetration BCs imposed at the domain boundaries, while the bottom row demonstrate our method that enforces open-boundary conditions. All the other conditions, in particular, the systematic discretization error introduced by the finite-difference scheme, are identical. Using our approach, the leapfrogging vortex pairs remain separated, rotating around each other and rising upward as expected (left). And the oppositely-signed vortex pairs continuously rise and pass over the obstacle (right). In contrast, ad-hoc boundary treatments lead to visually obvious artificial deceleration and even vortex merging (top). Such differences are observed even when the convergence tolerance of the interior solves is reduced to machine precision. Red and blue indicate counterclockwise and clockwise vortices respectively, while the gray disk represent solid obstacles.

shown in Fig. 14 (left), in the leapfrogging vortex experiment, the presence of artificial boundary conditions causes the velocity field to deviate from the far-field solution, resulting in reduced flow speeds and even the merging of vortex structures near the boundary. In contrast, our method reproduces the correct behavior with just *a single iteration*: the discrete vortices remain well separated throughout the simulation, unaltered by the presence of a computational domain. In Fig. 14 (right), for the oppositely-signed vortex pair experiment in the presence of obstacles, our method reconstructs a flow field that satisfies both the no-penetration condition on solid boundaries and the far-field condition within *two* iterations. In contrast, the standard Vortex-In-Cell method underestimates the vortex convection speed due to spurious boundary effects. Note that the interior streamfunction ψ from the last step serves as the initial guess in each solve, and that the frame-by-frame treatment of the far-field condition does not result in dynamic artifacts in the resulting simulation—see the supplementary video for animations.

7.2.3 Dirichlet Problems. Thus far, we have presented variable-coefficient Poisson equations as well as Neumann exterior BVPs for Laplace and Poisson equations. In addition to these, Dirichlet exterior problems also arise in a wide range of graphics applications, and they can likewise be solved efficiently using our proposed iterative method. We now provide several representative examples.

Harmonic Extension. Solutions to the Laplace equation—commonly referred to as harmonic functions—possess a number of important properties and are widely used in geometry processing tasks such as surface parameterization and generalized barycentric coordinates. In many applications, we know a field on the surface of an object, and need to smoothly interpolate it throughout space, corresponding to Dirichlet boundary conditions imposed on a polygonal mesh. With our iterative method, we efficiently solve exterior Laplace problems from any given Dirichlet data. This further implies that, for any harmonic function defined over the entire space $\mathbb{R}^d$, if one selects an arbitrary closed surface and stores the function values

on that surface, then the harmonic function can be uniquely and efficiently reconstructed within a prescribed neighborhood of the surface. This property follows from the uniqueness of harmonic extension in exterior domains.

Fig. 11 shows the numerical solutions of exterior problems on two complex geometries where Dirichlet boundary conditions are prescribed using the harmonic functions $u = (x^2 - z^2) / (x^2 + y^2 + z^2)^{2.5}$ and $u = (x + y + z) / (x^2 + y^2 + z^2)^{1.5}$. Naturally, these functions also serve as ground-truth solutions over the entire domain $\mathbb{R}^d \setminus \Omega_c$. When Dirichlet ABCs are applied, the resulting equipotential surfaces contract relative to the ground truth; conversely, Neumann ABCs expand them. Both treatments introduce visible spurious boundary effects. In contrast, our method converges in only 4 (dragon) or 6 (bunny) iterations, achieving smaller RMSE by one to two orders of magnitude—in our visualizations, the error is nearly indistinguishable from zero, demonstrating the high fidelity of our method.

Path Planning. The artificial potential field (APF) method [Warren 1989; Barraquand et al. 1992] is a classical approach in robot path planning. By treating the goal as an attractor and obstacles as repulsors, one can construct a virtual potential field that guides the robot’s motion. Compared to potential fields constructed from inverse-square laws or other ad-hoc kernel functions, potentials obtained by solving the Laplace equation are free of spurious local minima and exhibit superior smoothness properties [Sato 1992]. In a typical formulation, negative Dirichlet values are imposed on the surface of the target object, while positive Dirichlet values are imposed on the surfaces of obstacles. Solving the Laplace equation under these boundary conditions yields a potential ϕ whose negative gradient $-\nabla\phi$ defines the robot’s desired direction of motion. Evidently, an ideal potential field should guide a robot from any initial position in free space toward the target, which can only be achieved by solving the Laplace equation under *open* (unbounded) boundary conditions, rather than within a truncated domain.

In Fig. 15, we construct a robot path-planning test environment containing many obstacles. Under the conventional setup—assigning



(a) Neumann BC;

(b) Ours (Dirichlet on obstacles);

(c) Ours (Neumann on obstacles).

Fig. 15. **Laplace-based path planning.** Path-planning of a robot towards a target (red flag) is performed on a 640×360 grid, visualized as field lines on the ground (negated gradient of the potential field) to indicate the motion direction. Neumann BC on the computational domain boundary (a) leads to the trajectory being deflected by the computational domain, while an open-boundary Laplace equation with Dirichlet BC on the obstacles (b) results in overly smooth paths that become quite long, even exiting the domain; solving an open-boundary path-planning problem with Neumann BC on obstacles, instead, produces paths toward the target that are both short and smooth (c).

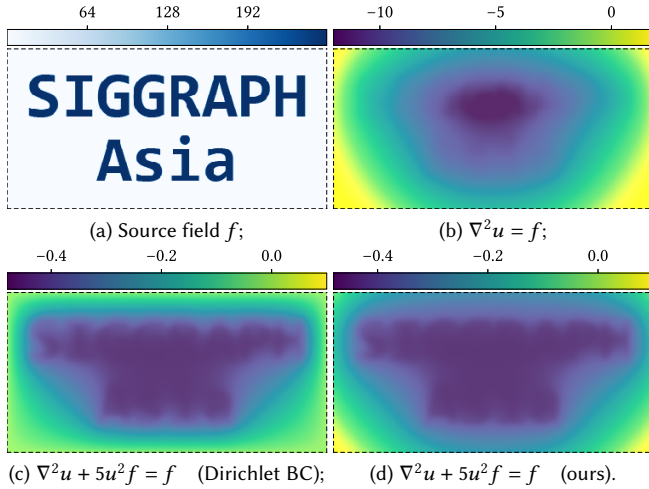


Fig. 16. **Nonlinear Poisson problem.** Our method can also address nonlinear Poisson problems. For instance, our method (d) successfully solves Eq. (33) for a piecewise-constant source field shown in (a). It provides a solution that correctly takes the far-field BC into account and is markedly different from a naive linearization (b) or Dirichlet ABC (c). Here we use AMG-preconditioned BiCGSTAB instead of AMG-PCG because of the lack of positive definiteness of the resulting system.

a positive potential on obstacle surfaces (here $\phi = 1$), a negative potential on the target surface (here $\phi = -5$), and imposing the artificial boundary condition $\partial_n \phi = 0$ —the resulting trajectory is confined within the computational boundary. Moreover, the path bends unnaturally near the boundary due to the Neumann boundary condition (Fig. 15a). If the actual problem does not involve a hard spatial limit (e.g., when the robot is an aerial vehicle or a ship), such a treatment inevitably restricts the agent’s freedom of navigation. By removing the Neumann ABC, our iterative method converges within six iterations and yields a solution with much smoother turning behavior. The paths exiting the boundary indicate that the computational domain should be enlarged to better guide the robot’s motion (Fig. 15b). In addition, far-field BCs enable another Laplace-based path-planning strategy: imposing $\partial_n \phi = 0$ on obstacle surfaces instead of Dirichlet conditions. In this case, the resulting paths preserve harmonic smoothness while favoring shorter routes, as illustrated in Fig. 15c. This latter approach is not possible for Neumann ABCs, as it yields the trivial constant solution.

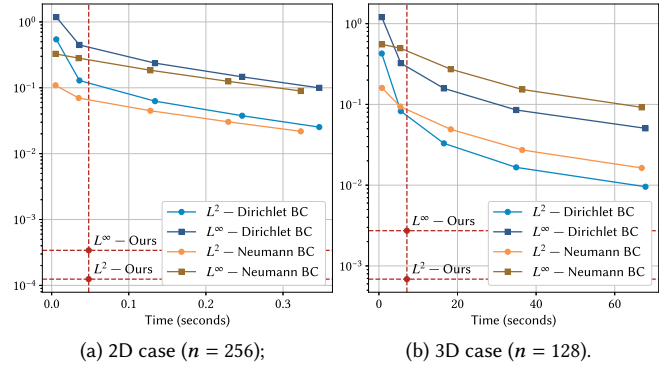
(a) 2D case ($n = 256$);(b) 3D case ($n = 128$).

Fig. 17. **Performance comparisons.** For the exterior Dirichlet problem of the Laplace equation from Fig. 4 and its 3D generalization (using a cube of side length 0.8, with $u = (x+y+z)/(x^2+y^2+z^2)^{1.5}$), we evaluate the performance of Dirichlet and Neumann ABCs as the size of the computational domain increases. Grid resolution is set to $(sn)^d$ and the computational domain is $[-s, s]^d$, with $s \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$; we plot the error as a function of time measured in the region $[-1, 1]^d$ for $n=256$ in 2D and $n=128$ in 3D. Artificial boundary methods converge very slowly, whereas our method achieves accurate results within only a few iterations. The coordinates of the intersection of the two dashed lines represent the computational time and error of our method when it converges at $s=1$, for comparison.

7.3 Further Evaluations

Nonlinear Case. Our method is not limited to linear Poisson BVPs; it is also effective for nonlinear problems. We demonstrate this case using the nonlinear equation

$$\nabla^2 u(\mathbf{x}) + 5f(\mathbf{x})u^2(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2, \quad (33)$$

in which the reaction term $5fu^2$ depends quadratically on the solution u . We let $f(\mathbf{x})$ follow the pattern shown in Fig. 16a and discretize the problem on a 400×200 MAC grid. If the nonlinear reaction term is ignored, the resulting field is completely diffuse, as shown in Fig. 16b. When the nonlinear term is included but Dirichlet ABCs are applied, the visualization of the solution u in Fig. 16c roughly recovers the source pattern but still suffers from noticeable boundary effects. In contrast, our method successfully produces a solution that decays at infinity using only four boundary updates, as shown in Fig. 16d.

Computational Complexity. We note that Alg. 1 requires the same memory footprint as the original bounded-domain formulation, namely $O(N)$. The total computational complexity is $O(KTN)$, where T denotes the number of iterations used by the interior linear

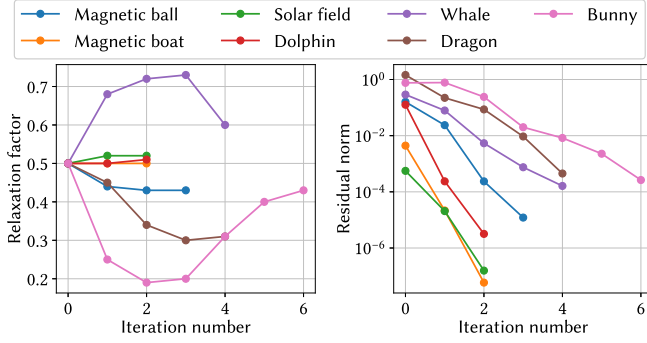


Fig. 18. **Adaptive relaxation and convergence.** The adaptive relaxation strategy performs robustly across a wide range of experimental scenarios, as our method always converges within only a few iterations. Left: the evolutions of the relaxation factor $\omega^{(k)}$. Right: the corresponding residual norm $\|r^{(k)}\|$. The results reveal several interesting observations; for example, cavities with Dirichlet BCs lead to an optimal ω smaller than $1/2$, whereas cavities with Neumann BCs yield an optimal ω larger than $1/2$.

solver, and K denotes the number of boundary-condition iterations required for convergence. The cost of evaluating the boundary integrals has been omitted from the asymptotic complexity estimate due to their negligible nature: when performed on a 256^d grid (with the setup shown in Fig. 3, where all cells are active), the time spent on boundary integrals accounts for only about 7.6% of the total runtime in 2D (without FMM) and 11.3% in 3D (with FMM), which is indeed a small overhead. The overall performance of the algorithm is therefore dominated primarily by the number of calls to the interior solver, i.e., by K . One also benefits from restarting the interior solves with the previous solution (warm start), as it speeds up the total algorithm runtime.

To quantitatively evaluate the performance of the proposed iterative method, we construct the experimental setup shown in Fig. 4. Even in the presence of such an extreme interior cavity, our method converges within only four iterations and achieves substantially higher accuracy than ABCs. In this scenario and its 3D generalization, we compare our approach against homogeneous ABCs combined with an enlargement of the computational domain; the results are shown in Fig. 17. These experiments highlight that the latter strategy is not only less accurate, but also significantly more expensive than our proposed iterative scheme.

Relaxation Factor. Regarding the optimal choice of the relaxation factor ω in our method, its adaptive update is essential for efficiency across different scenarios. As shown in the fifth column of Table 1, fixing the relaxation factor (e.g., keeping it equal to $1/2$) can significantly increase the number of iterations required for non-standard Poisson problems, and may even lead to divergence. This sensitivity arises because the admissible range and optimal value of ω depend on factors such as problem type and size of the interior cavity.

To better understand this behavior, we recorded the evolution of the optimal relaxation factors during the execution of the algorithm over various experiments; the results are presented in Fig. 18. Qualitatively, cavities with Dirichlet boundary conditions tend to yield an optimal relaxation factor smaller than $1/2$, with smaller values for larger cavity sizes. In contrast, cavities with Neumann

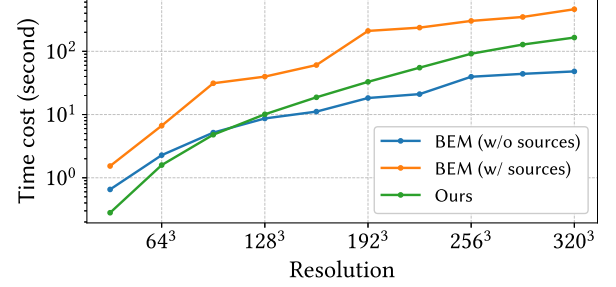


Fig. 19. **Comparison with BEM in 3D.** We compare the computational cost of our method with that of collocation BEM for solving the magnetization problem of a sphere (see Fig. 5) at comparable accuracy. In the absence of source terms, BEM has a lower complexity, which is precisely where its primary advantage lies. However, when a uniform source field is distributed, the volume integrals needed in BIE have the same asymptotic complexity as FDM while incurring a relatively large constant factor. In contrast, the performance of our method is essentially unaffected by the presence of source terms. Both methods use four quadrature points per face and eight quadrature points per cell. Note that memory allocation in the FMM3D library is heuristic, so the runtime growth of BEM is less smooth than ours.

boundary conditions lead to an optimal relaxation factor larger than $1/2$, with larger values for larger cavity sizes. Spatially-varying diffusion coefficients also cause deviations from $1/2$, although without a clear general trend. Notably, the presence or absence of source terms does not affect the optimal relaxation factor. Even in extreme cases, there always exists a choice of ω ensuring convergence; this is partially explained by our analysis in App. C, which also provides insight into the mechanisms underlying these observations.

7.4 Comparisons with Existing Methods

Finally, we compare our approach with existing methods for solving Poisson problems on unbounded domains. Overall, our simpler method exhibits strong generality, achieving a favorable balance between efficiency and accuracy across a wide range of problems.

With Pure BEM. By exploiting the properties of BIEs over the entire solution domain, BEM is particularly well-suited for solving both interior and exterior problems of the Laplace equation [Da et al. 2016; Solomon et al. 2017]. For variable-coefficient problems in which the diffusion coefficient is piecewise constant, BIEs can be formulated on the interfaces where coefficient discontinuities occur, enabling BEM to effectively address such cases as well [Huang and Michels 2020]. Compared with volumetric methods such as the FDM, FVM, and FEM, a major advantage of the BEM is that the resulting discretized system involves far fewer unknowns, which generally leads to substantially higher computational efficiency.

In Fig. 19, we compare the computational efficiency of our method with a collocation BEM solver based on matrix-free GMRES for the 3D version of the scenario shown in Fig. 5, under different spatial resolutions. To ensure comparable accuracy, on an n^3 grid we set the number of BEM DoFs (i.e., the number of vertices on the spherical boundary mesh) to $4n^2$. In terms of iteration count, the BEM solver requires approximately three times as many iterations as our method. Each iteration of our method consists of one PCG solve and one boundary-to-boundary integral evaluation, whereas each

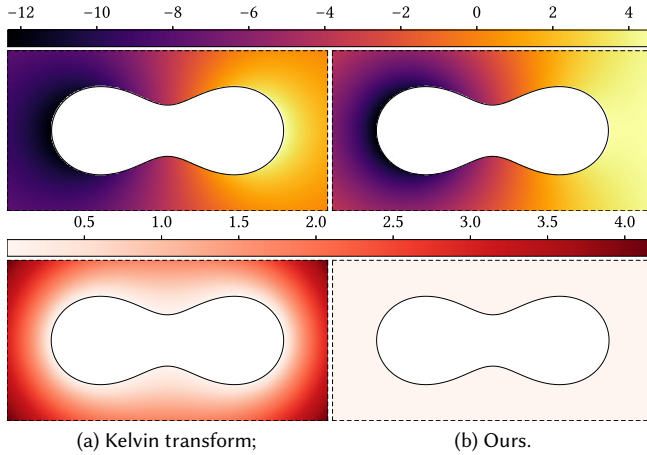


Fig. 20. **Comparison with KT in 2D.** For a 2D exterior Dirichlet problem for the Laplace equation with nonzero net flux, the Kelvin transform uses 779 800 valid cells, whereas our method uses 89 060. Over $[-1, 1] \times [-0.5, 0.5]$, KT yields L^2 and L^∞ errors of 1.91 and 4.18 respectively, reflecting a systematic bias; after just four iterations, our method achieves L^2 and L^∞ errors of 1.78×10^{-3} and 5.98×10^{-3} instead. The upper row visualizes the result fields, while the lower row shows the error fields relative to the analytical ground truth.

BEM iteration requires only one boundary-to-boundary integral evaluation and, after convergence, an additional boundary-to-bulk integral to propagate the solution onto the grid. Across all tested resolutions—particularly at higher resolutions—the lower asymptotic time complexity of BEM becomes increasingly advantageous.

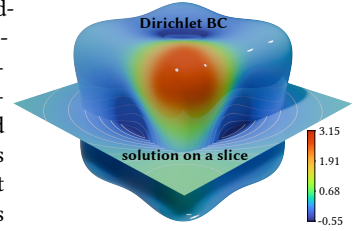
However, as soon as the system contains non-removable source terms, the BIE must take additional volume integrals into account. The dual reciprocity BEM [Partridge et al. 1991] deals with the integrals by fitting source terms with Radial Basis Functions (RBF) before solving BIE, converting volume integrals back into surface ones and thus bypassing volumetric discretization. Yet, solving the dense RBF system obviously lacks scalability. Instead, we evaluate the volume integrals directly over internal grids, a method that proved more efficient in our experiments. This entails one bulk-to-boundary integral as a preprocessing step and one bulk-to-bulk integral for solution reconstruction. Since the complexity of FMM-based integration grows linearly with the number of points while carrying a relatively large constant factor, our method becomes more favorable in this setting—the presence of source terms does not affect the convergence of either the interior FVM solves or the exterior iterations, the proof following analogously to that in Sec. 6.1. If the grid resolution continues to increase, BEM will even quickly run out of the memory in 3D, because its spatial complexity is $O(p^d N)$, where p is the order of expansion. Finally, for more general variable-coefficient or nonlinear equations, the applicability of BEM is severely limited.

With Kelvin Transform. KT factorizes the decay term $\|\mathbf{x}\|^{2-d}$ from the solution $u(\mathbf{x})$, thus mapping an exterior Laplace problem to an interior one without altering the form of the equation. We compared our approach vs. KT for solving the exterior Dirichlet problem for the Laplace equation in the unbounded domain outside the region enclosed by the Goursat surface $(x^4 + y^4 + z^4) - 2(x^2 + y^2 + z^2) = 1$, with

Table 1. **Ablation study.** Our method is compared against two variants: (1) using a fixed relaxation factor of 0.5 (no Aitken’s acceleration); and (2) solving the full BEM in the exterior domain instead of performing only a single BIE evaluation. The BEM solver is implemented using a Neumann-series formulation [Sugimoto et al. 2023; Ni et al. 2024], where repeated BIE evaluations are accelerated by FMM. The results demonstrate that our proposed scheme achieves a significant advantage in convergence: across all tested scenarios, it consistently requires the fewest iterations.

Fig.	Dim.	Name	Ours	Fixed ω	FVM-BEM
3	2	Circular source	1	1	6
3	3	Spherical source	1	1	5
4	2	Large cavity	4	∞	5
5	2	Magnetic disk	3	7	6
5	3	Magnetic ball	3	4	7
9	3	Solar field	2	3	7
10	3	Dolphin	2	2	7
12	3	Whale	4	6	7
11	3	Bunny	6	∞	7
11	3	Dragon	4	13	8

boundary condition and ground-truth solution given, respectively, as $u(\mathbf{x}) = \mathbf{r}_1 \cdot \mathbf{d} / \|\mathbf{r}_1\|^3 - \mathbf{r}_2 \cdot \mathbf{d} / \|\mathbf{r}_2\|^3$, where $\mathbf{r}_1 = \mathbf{x} - (1, 1, 1)$, $\mathbf{r}_2 = \mathbf{x} + (1, 1, 1)$, and $\mathbf{d} = (1, 1, 1)$ (see inset). In this simple 3D constant-coefficient Laplace case, KT outperforms our approach: by solving a single linear system, the Kelvin transform is 4× faster than our approach, which takes four iterations to converge to a similar accuracy.



Nevertheless, the situation changes in two dimensions. When the net flux of the system is nonzero, the natural far-field behavior in 2D diverges logarithmically, which cannot be factorized using $\|\mathbf{x}\|^{2-d}$. In Fig. 20, we solve the exterior Dirichlet problem for the Laplace equation outside the closed region enclosed by the Cassini curve $(x^2 + y^2)^2 - 2c^2(x^2 - y^2) = a^4 - c^4$ with $a = 0.525$ and $c = 0.5$. The boundary condition and ground-truth solution are both generated from $u(\mathbf{x}) = 4 \log \|\mathbf{x} - \mathbf{x}_1\| - \log \|\mathbf{x} - \mathbf{x}_2\|$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the left and right foci of the curve, respectively. Since the KT factor does not match the asymptotic behavior of the solution anymore, it yields significant error in this case. In contrast, our approach provides greater accuracy by handling the far-field conditions via Green’s functions and BIE, and thus works consistently in both 2D and 3D. While we could use $\log(r)$ as the KT factor in 2D, the resulting interior equation is no longer a simple Laplace equation as it now has an additional singular term. Similarly, for variable-coefficient or nonlinear problems lacking spherical symmetry, KT significantly complicates the governing equations and is generally not a recommended choice in practice.

With Coupled FVM-BEM. As discussed earlier, our method does not exactly solve the exterior BEM problem. Instead, it applies a damped update of Dirichlet data from the derived Neumann data of the interior FVM—that is, an inexact solve of the BIE. Alternatively, one could solve the exact BIE for the exterior Neumann problem

of the Laplace equation, and then feed the resulting solution into subsequent iterations. However, as shown in the last column of Table 1, this approach is almost always inferior. Not only does solving the full BEM significantly increase the cost per iteration, but it also violates the properties established in Sec. 6.1, thereby increasing the total number of iterations. Together, these issues would result in a substantial increase in overall computational cost.

8 Conclusion and Future Work

We presented a simple and practical Poisson solver that effectively handles open boundaries, and showcased its efficacy across a variety of problems in computer graphics. Our method eliminates the need to discretize regions far from the main area of interest, ensuring that all discrete cells maintain comparable physical scales—a feature that benefits numerical conditioning and accuracy. Our method is also highly versatile: it consistently processes domains with or without explicit boundaries and accommodates governing equations ranging from the simple Laplace equation to the more general variable-coefficient and nonlinear Poisson equations, with convergence automatically managed via an adaptive relaxation scheme.

We emphasize that although this work focuses on solving the Poisson equation, our method can be naturally extended to unbounded-domain problems for other equations, such as the screened Poisson and Helmholtz equations by simply replacing the integral kernel in the exterior BEM solve with the Yukawa or Helmholtz potential, respectively. This represents a straightforward next step that we wish to evaluate. Beyond that, applying our method to the simulation of nonlinear magnetizable materials in motion is an interesting direction for future exploration, as it allows our method to be directly wrapped around the solver of Sun et al. [2021]. For fluid simulation applications, the vortex-in-cell comparison in Fig. 14 currently must be terminated once any vortex particle reaches the boundary. This is purely a limitation of the present fixed-grid experimental setup. Since our method is fully compatible with adaptive grids, this issue should be addressed in practical deployments. Looking further ahead, developing a boundary-free advection solver for particle-free stable fluids can also naturally complement the unbounded-domain velocity projection enabled by our method. Finally, more formal proofs of efficient convergence of our approach for highly non-linear equations would also be desirable.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their constructive feedback. This work was partially funded by the Research Grants Council of Hong Kong (Ref: 17216826), and by the Innovation and Technology Commission of the HKSAR Government under the ITSP-Platform grant (Ref: ITS/335/23FP, ITS/469/24FP). MD also acknowledges the support of the MediTwin consortium (funded by the French government as part of France 2030), an Inria Chair, and of Synopsys and Adobe Research.

References

- Aseem Agarwala. 2007. Efficient gradient-domain compositing using quadrees. *ACM Trans. Graph.* 26, 3 (July 2007), 94–es. doi:10.1145/1276377.1276495
- Amit Agrawal, Ramesh Raskar, and Rama Chellappa. 2006. What Is the Range of Surface Reconstructions from a Gradient Field?. In *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 578–591. doi:10.1007/11744023_45
- Alexander C. Aitken. 1927. On Bernoulli's Numerical Solution of Algebraic Equations. *Proceedings of the Royal Society of Edinburgh* 46 (1927), 289–305. doi:10.1017/S0370164600022070
- Muzaffer Akbay, Nicholas Nobles, Victor Zordan, and Tamar Shinar. 2018. An extended partitioned method for conservative solid-fluid coupling. *ACM Trans. Graph.* 37, 4, Article 86 (July 2018), 12 pages. doi:10.1145/3197517.3201345
- Martin D. Altschuler. 1969. Magnetic fields and the structure of the solar corona. *Solar Physics* 9, 1 (1969), 131–149. doi:10.1007/BF00145734
- Alexis Angelidis and Fabrice Neyret. 2005. Simulation of smoke based on vortex filament primitives. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) (SCA '05). Association for Computing Machinery, New York, NY, USA, 87–96. doi:10.1145/1073368.1073380
- Travis Askham, Zydrunas Gimbutas, Leslie Greengard, Libin Lu, Jeremy Magland, Dhairya Malhotra, Michael O'Neil, Manas Rachh, Vladimir Rokhlin, and Felipe Vico. 2024. Fast multipole methods in three dimensions (FMM3D). <https://fmm3d.readthedocs.io>
- Josh Barnes and Piet Hut. 1986. A hierarchical O(N log N) force-calculation algorithm. *Nature* 324, 6096 (1986), 446–449. doi:10.1038/324446a0
- J. Barraquand, B. Langlois, and J.-C. Latombe. 1992. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics* 22, 2 (1992), 224–241. doi:10.1109/21.148426
- Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.* 26, 3 (July 2007), 100–es. doi:10.1145/1276377.1276502
- A. Bendali, Y. Boubendir, and M. Fares. 2007. A FETI-like domain decomposition method for coupling finite elements and boundary elements in large-size problems of acoustic scattering. *Computers & Structures* 85, 9 (2007), 526–535. doi:10.1016/j.compstruc.2006.08.029
- Jean-Pierre Berenger. 1994. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.* 114, 2 (1994), 185–200. doi:10.1006/jcph.1994.1159
- Peter Bettess. 1977. Infinite elements. *Internat. J. Numer. Methods Engrg.* 11, 1 (1977), 53–64. doi:10.1002/nme.1620110107
- P. Birken, T. Gleim, D. Kuhl, and A. Meister. 2015. Fast solvers for unsteady thermal fluid structure interaction. *International Journal for Numerical Methods in Fluids* 79, 1 (2015), 16–29. doi:10.1002/flid.4040
- Morten Bojsen-Hansen and Chris Wojtan. 2016. Generalized non-reflecting boundaries for fluid re-simulation. *ACM Trans. Graph.* 35, 4, Article 96 (July 2016), 7 pages. doi:10.1145/2897824.2925963
- Robert Bridson. 2015. *Fluid simulation for computer graphics* (2nd ed.). A K Peters/CRC Press, Boca Raton, FL, USA. doi:10.1201/9781315266008
- Jiong Chen, Florian Schaefer, and Mathieu Desbrun. 2024. Lightning-fast Method of Fundamental Solutions. *ACM Trans. Graph.* 43, 4, Article 77 (July 2024), 16 pages. doi:10.1145/3658199
- Jiong Chen, Florian Schäfer, and Mathieu Desbrun. 2025. Lightning-fast Boundary Element Method. *ACM Trans. Graph.* 44, 4, Article 38 (July 2025), 14 pages. doi:10.1145/3731196
- Albert Chern. 2019. A reflectionless discrete perfectly matched layer. *J. Comput. Phys.* 381 (2019), 91–109. doi:10.1016/j.jcp.2018.12.026
- Martin Costabel. 1987. Principles of boundary element methods. *Computer Physics Reports* 6, 1 (1987), 243–274. doi:10.1016/0167-7977(87)90014-1
- P. Coulier, S. François, G. Lombaert, and G. Degrande. 2014. Coupled finite element – hierarchical boundary element methods for dynamic soil–structure interaction in the frequency domain. *Internat. J. Numer. Methods Engrg.* 97, 7 (2014), 505–530. doi:10.1002/nme.4597
- M. D. Cowley and Ronald E. Rosensweig. 1967. The interfacial stability of a ferro-magnetic fluid. *Journal of Fluid mechanics* 30, 4 (1967), 671–688. doi:10.1017/S0022112067001697
- Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Trans. Graph.* 35, 4, Article 78 (July 2016), 12 pages. doi:10.1145/2897824.2925899
- Fernando de Goes and Mathieu Desbrun. 2024. Stochastic Computation of Barycentric Coordinates. *ACM Trans. Graph.* 43, 4, Article 42 (July 2024), 13 pages. doi:10.1145/3658131
- Denis Demidov. 2019. AMGCL: An Efficient, Flexible, and Extensible Algebraic Multigrid Implementation. *Lobachevskii Journal of Mathematics* 40, 5 (2019), 535–546. doi:10.1134/S1995080219050056
- Bruno Després. 1991. *Méthodes de décomposition de domaine pour la propagation d'ondes en régime harmonique. Le théorème de Borg pour l'équation de Hill vectorielle*. Ph.D. Dissertation. Thèse de doctorat dirigée par Joly, Patrick Physique Paris 9 1991. <http://www.theses.fr/1991PA090032>
- M. El-Gebeily, Wael M. Elleithy, and Husain J. Al-Gahtani. 2002. Convergence of the domain decomposition finite element–boundary element coupling methods. *Computer Methods in Applied Mechanics and Engineering* 191, 43 (Sept. 2002), 4851–4867. doi:10.1016/s0045-7825(02)00405-x

- Wael M. Elleithy and Masataka Tanaka. 2003. Interface relaxation algorithms for BEM-BEM coupling and FEM-BEM coupling. *Computer Methods in Applied Mechanics and Engineering* 192, 26–27 (July 2003), 2977–2992. doi:10.1016/S0045-7825(03)00312-8
- Björn Engquist and Andrew Majda. 1977. Absorbing boundary conditions for the numerical simulation of waves. *Proceedings of the National Academy of Sciences* 74, 5 (1977), 1765–1766. doi:10.1073/pnas.74.5.1765
- Yuntian Feng and D. Roger J. Owen. 1996. Iterative Solution of Coupled Fe/Be Discretizations for Plate-Foundation Interaction Problems. *Internat. J. Numer. Methods Engrg.* 39, 11 (1996), 1889–1901. doi:10.1002/(SICI)1097-0207(19960615)39:11<1889::AID-NME934>3.0.CO;2-Z
- Michael S. Floater. 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design* 14, 3 (1997), 231–250. doi:10.1016/S0167-8396(96)00031-3
- Dan Givoli and Joseph B. Keller. 1989. A finite element method for large domains. *Computer Methods in Applied Mechanics and Engineering* 76, 1 (1989), 41–66. doi:10.1016/0045-7825(89)90140-0
- Leslie Greengard and Vladimir Rokhlin. 1987. A fast algorithm for particle simulations. *Journal of computational physics* 73, 2 (1987), 325–348. doi:10.1016/0021-9991(87)90140-9
- Chester E. Grosch and Steven A. Orszag. 1977. Numerical solution of problems in unbounded regions: Coordinate transforms. *J. Comput. Phys.* 25, 3 (November 1977), 273–295. [http://dx.doi.org/10.1016/0021-9991\(77\)90102-4](http://dx.doi.org/10.1016/0021-9991(77)90102-4)
- Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph.* 24, 3 (July 2005), 973–981. doi:10.1145/1073204.1073299
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen. <https://libeigen.gitlab.io>
- Wolfgang Hackbusch. 1999. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to-matrices. *Computing* 62, 2 (1999), 89–108. doi:10.1007/s006070050015
- Houde Han and Xiaonan Wu. 2013. *Artificial Boundary Method*. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-35464-9
- Matthew Harker and Paul O’Leary. 2008. Least squares surface reconstruction from measured gradient fields. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (Anchorage, AK, USA). IEEE, New York, NY, USA, 1–7. doi:10.1109/CVPR.2008.4587414
- Francis H. Harlow and J. Eddie Welch. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids* 8, 12 (1965), 2182–2189. doi:10.1063/1.1761178
- Libo Huang, Torsten Hädrich, and Dominik L. Michels. 2019. On the accurate large-scale simulation of ferrofluids. *ACM Trans. Graph.* 38, 4, Article 93 (July 2019), 15 pages. doi:10.1145/3306346.3322973
- Libo Huang and Dominik L. Michels. 2020. Surface-only ferrofluids. *ACM Trans. Graph.* 39, 6, Article 174 (Nov. 2020), 17 pages. doi:10.1145/3414685.3417799
- Bruce M. Irons and Robert C. Tuck. 1969. A version of the Aitken accelerator for computer iteration. *Internat. J. Numer. Methods Engrg.* 1, 3 (1969), 275–277. doi:10.1002/nme.1620010306
- Doug L. James and Dinesh K. Pai. 1999. ArtDefo: accurate real time deformable objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’99)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–72. doi:10.1145/311535.311542
- Claes Johnson and J. Claude Nédélec. 1980. On the Coupling of Boundary Integral and Finite Element Methods. *Math. Comp.* 35, 152 (1980), 1063–1079. doi:10.2307/2006375
- Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (July 2007), 71–es. doi:10.1145/1276377.1276466
- Norio Kamiya, Hidehito Iwase, and Eisuke Kita. 1996. Parallel computing for the combination method of BEM and FEM. *Engineering Analysis with Boundary Elements* 18, 3 (Oct. 1996), 223–229. doi:10.1016/S0955-7997(96)00052-5
- Ram P. Kanwal. 2013. *Linear Integral Equations: Theory & Technique* (2nd ed.). Birkhäuser, New York, NY, USA. doi:10.1007/978-1-4614-6012-1
- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy) (SGP ’06). Eurographics Association, Goslar, DEU, 61–70. doi:10.5555/1281957.1281965
- Michael Kazhdan and Hugues Hoppe. 2013. Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, 3, Article 29 (July 2013), 13 pages. doi:10.1145/2487228.2487237
- Chaoyang Lyu, Kai Bai, Yiheng Wu, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. 2023. Building a Virtual Weakly-Compressible Wind Tunnel Testing Facility. *ACM Trans. Graph.* 42, 4, Article 125 (July 2023), 20 pages. doi:10.1145/3592394
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4, Article 82 (July 2023), 11 pages. doi:10.1145/3592400
- Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin transformations for simulations on infinite domains. *ACM Trans. Graph.* 40, 4, Article 97 (July 2021), 15 pages. doi:10.1145/3450626.3459809
- National Solar Observatory. 2026. Zero Point Corrected Magnetogram Synoptic Map Images. <https://magmap.nso.edu/>
- Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.* 228, 23 (2009), 8807–8829. doi:10.1016/j.jcp.2009.08.032
- Xingyu Ni, Ruicheng Wang, Bin Wang, and Baoquan Chen. 2024. An Induce-on-Boundary Magnetostatic Solver for Grid-Based Ferrofluids. *ACM Trans. Graph.* 43, 4, Article 56 (July 2024), 14 pages. doi:10.1145/3658124
- Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2020. A level-set method for magnetic substance simulation. *ACM Trans. Graph.* 39, 4, Article 29 (Aug. 2020), 15 pages. doi:10.1145/3386569.3392445
- Paul William Partridge, Carlos Alberto Brebbia, and Luiz C. Wrobel. 1991. *Dual Reciprocity Boundary Element Method*. Springer, Dordrecht, Netherlands. doi:10.1007/978-94-011-3690-7
- Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Trans. Graph.* 22, 3 (July 2003), 313–318. doi:10.1145/882262.882269
- Eric Priest. 2014. *Magnetohydrodynamics of the Sun*. Cambridge University Press, Cambridgeshire, England, UK. doi:10.1017/CBO9781139020732
- Keisuke Sato. 1992. Deadlock-free motion planning using the Laplace potential field. *Advanced Robotics* 7, 5 (1992), 449–461. doi:10.1163/156855393X00285
- Rohan Sawhney and Keenan Crane. 2020. Monte Carlo geometry processing: a grid-free approach to PDE-based methods on volumetric domains. *ACM Trans. Graph.* 39, 4, Article 123 (Aug. 2020), 18 pages. doi:10.1145/3386569.3392374
- Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4, Article 80 (July 2023), 20 pages. doi:10.1145/3592398
- Camille Schreck, Christian Hafner, and Chris Wojtan. 2019. Fundamental solutions for water wave animation. *ACM Trans. Graph.* 38, 4, Article 130 (July 2019), 14 pages. doi:10.1145/3306346.3323002
- Andrew Selle, Nick Rasmussen, and Ronald Fedkiw. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.* 24, 3 (July 2005), 910–914. doi:10.1145/1073204.1073282
- Han Shao, Libo Huang, and Dominik L. Michels. 2023. A Current Loop Model for the Fast Simulation of Ferrofluids. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2023), 5394–5405. doi:10.1109/TVCG.2022.3211414
- Delfim Soares, Jr, Otto von Estorff, and Webe Jo ao Mansur. 2004. Iterative coupling of BEM and FEM for nonlinear dynamic analyses. *Computational Mechanics* 34, 1 (March 2004), 67–73. doi:10.1007/s00466-004-0554-4
- Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary Element Octahedral Fields in Volumes. *ACM Trans. Graph.* 36, 4, Article 114b (July 2017), 16 pages. doi:10.1145/3072959.3065254
- Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’99)*. ACM Press/Addison-Wesley Publishing Co., USA, 121–128. doi:10.1145/311535.311548
- Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2022. Surface-Only Dynamic Deformables using a Boundary Element Method. *Computer Graphics Forum* 41, 8 (2022), 75–86. doi:10.1111/cgf.14625
- Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4, Article 81 (July 2023), 16 pages. doi:10.1145/3592109
- Xin Sun, Guofu Xie, Yue Dong, Stephen Lin, Weiwei Xu, Wencheng Wang, Xin Tong, and Baining Guo. 2012. Diffusion curve textures for resolution independent texture mapping. *ACM Trans. Graph.* 31, 4, Article 74 (July 2012), 9 pages. doi:10.1145/2185520.2185570
- Yuchen Sun, Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2021. A material point method for nonlinearly magnetized materials. *ACM Trans. Graph.* 40, 6, Article 205 (Dec. 2021), 13 pages. doi:10.1145/3478513.3480541
- William Thomson. 1845. Extrait d’une lettre de M. William Thomson à M. Liouville. *Journal de mathématiques pures et appliquées* 10 (1845), 364–367.
- Jasper van de Gronde. 2010. *A High Quality Solver for Diffusion Curves*. Master’s thesis. University of Groningen.
- Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional Field Synthesis, Design, and Processing. *Computer Graphics Forum* 35, 2 (2016), 545–572. doi:10.1111/cgf.12864
- C. W. Warren. 1989. Global path planning using artificial potential fields. In *1989 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, Los Alamitos, CA, USA, 316–321. doi:10.1109/ROBOT.1989.100007
- Mengqi (Mandy) Xia, Bruce Walter, Eric Michielssen, David Bindel, and Steve Marschner. 2020. A wave optics based fiber scattering model. *ACM Trans. Graph.* 39, 6, Article 252 (Nov. 2020), 16 pages. doi:10.1145/3414685.3417841
- Jinchao Xu and Ludmil Zikatanov. 2017. Algebraic multigrid methods. *Acta Numerica* 26 (2017), 591–721. doi:10.1017/S0962492917000083
- Zihong Zhou, Eugene d’Eon, Rohan Sawhney, and Wojciech Jarosz. 2025. Harmonic Caching for Walk on Spheres. *ACM Trans. Graph.* 44, 6, Article 253 (Dec. 2025), 15 pages. doi:10.1145/3763322

A Analytic Expressions

In this appendix, we provide a number of analytical expressions of Green's functions and details about layer potentials when piecewise constant elements are used to ease implementation.

A.1 Fundamental Solutions of the Laplace Operator

Throughout this paper, the fundamental solution of the Laplace operator refers to the Green's function of the Poisson equation in free space, namely the solution

$$\nabla^2 G(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}'), \quad \mathbf{x} \in \mathbb{R}^d, \quad (34)$$

where δ is the d -dimensional Dirac delta function. In 2D and 3D, the fundamental solution admits the following closed-form expressions:

$$G^{2D}(\mathbf{x}, \mathbf{x}') = \frac{1}{2\pi} \log \|\mathbf{x} - \mathbf{x}'\|, \quad (35a)$$

$$G^{3D}(\mathbf{x}, \mathbf{x}') = -\frac{1}{4\pi} \frac{1}{\|\mathbf{x} - \mathbf{x}'\|}. \quad (35b)$$

Their gradients are given by

$$\nabla G^{2D}(\mathbf{x}, \mathbf{x}') = \frac{1}{2\pi} \frac{\mathbf{x} - \mathbf{x}'}{\|\mathbf{x} - \mathbf{x}'\|^2}, \quad (36a)$$

$$\nabla G^{3D}(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi} \frac{\mathbf{x} - \mathbf{x}'}{\|\mathbf{x} - \mathbf{x}'\|^3}. \quad (36b)$$

These expressions directly yield the normal derivatives required for the evaluation of the layer potentials.

A.2 Layer Potentials for Constant Elements

For each element, the source density is assumed constant over the element, while the boundary integral kernel still varies with spatial location. Owing to the simple rectangular geometry of boundary faces in grid-based discretizations, the boundary integrals can be evaluated in closed form and exploited efficiently in practice.

We first consider the two-dimensional case, where a boundary face is a line segment of length $\Delta x = 2d$, as illustrated in the right inset. For a field point located at a tangential offset a from the segment center and a normal distance h , the single-layer potential induced by a constant element is proportional to

$$\frac{1}{2\pi} \int_{-d}^{+d} \log \sqrt{(x-a)^2 + h^2} dx = \frac{1}{4\pi} \Delta_d^x F(a), \quad (37)$$

$$F(x) := x\sqrt{x^2 + h^2} + 2h \arctan \frac{x}{h} - 2x,$$

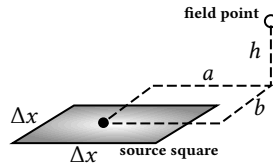
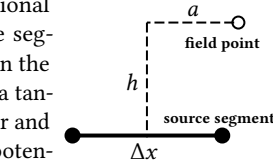
while the corresponding double-layer potential is proportional to

$$\frac{1}{2\pi} \int_{-d}^{+d} \frac{h}{(x-a)^2 + h^2} dx = \frac{1}{2\pi} \Delta_d^x F(a), \quad (38)$$

$$F(x) := \arctan \frac{x}{h},$$

Here the first-order centered finite difference operator Δ_d^x is defined by $\Delta_d^x F(a) = F(a+d) - F(a-d)$.

Analogously, in three dimensions a boundary face is a square of side length $\Delta x = 2d$, as shown in the right inset. For a field point located at tangential offsets a and b from the square



center and a normal distance h , the single-layer potential induced by a constant element is proportional to

$$-\frac{1}{4\pi} \int_{-d}^{+d} \int_{-d}^{+d} \frac{1}{\sqrt{(x-a)^2 + h^2}} dx dy = \frac{1}{4\pi} \Delta_d^x \Delta_d^y F(a, b), \quad (39)$$

$$F(x, y) := h \arctan \frac{xy}{hr} - x \log(r+y) - y \log(r+x),$$

where $r := \sqrt{x^2 + y^2 + h^2}$. The corresponding double-layer potential is proportional to

$$\frac{1}{4\pi} \int_{-d}^{+d} \int_{-d}^{+d} \frac{h}{((x-a)^2 + h^2)^{3/2}} dx dy = \frac{1}{4\pi} \Delta_d^x \Delta_d^y F(a, b), \quad (40)$$

$$F(x, y) := \arctan \frac{xy}{hr}.$$

Here the second-order centered finite difference operator is now $\Delta_d^x \Delta_d^y F(a, b) := F(a+d, b+d) - F(a+d, b-d) - F(a-d, b+d) + F(a-d, b-d)$.

Gauss-Legendre Quadrature. Gauss-Legendre quadrature approximates a continuous integral by a weighted sum of the integrand evaluated at a set of discrete quadrature points. For rectangular constant elements, a second-order Gauss-Legendre rule admits a particularly simple form. In 2D, two quadrature points are placed at distances $\pm \Delta x / 2\sqrt{3}$ from the segment center, each with weight $\Delta x / 2$. In 3D, four quadrature points are placed at tangential offsets $(\pm \Delta x / 2\sqrt{3}, \pm \Delta x / 2\sqrt{3})$ from the square center, each assigned a weight of $\Delta x^2 / 4$.

Correction of Self-Contribution. From Eqs. (36a)–(36b), we observe that when the normal vector $\mathbf{n}$ is orthogonal to $\mathbf{x} - \mathbf{x}'$, the normal derivative $\mathbf{n} \cdot \nabla G(\mathbf{x}, \mathbf{x}')$ vanishes. Consequently, the double-layer potential induced by a boundary face at its own center is identically zero. This property does not hold for the single-layer potential. Assuming a unit source density, a constant element produces a nonzero self-contribution at its own center, given by

$$w_0^{2D} = \frac{\Delta x}{2\pi} \left(\log \frac{\Delta x}{2} - 1 \right), \quad (41a)$$

$$w_0^{3D} = -\frac{\Delta x}{\pi} \log(\sqrt{2} + 1). \quad (41b)$$

Because the underlying integral is weakly singular, special care must be taken when evaluating the self-contribution numerically. In the three-dimensional case, the self-contribution predicted by the aforementioned Gauss-Legendre quadrature rule is

$$\tilde{w}_0 = -\frac{\sqrt{6}\Delta x}{4\pi}, \quad (42)$$

which differs from w_0 by an amount on the order of $0.1\Delta x$.

B Green's Identities in an Exterior Domain

Let $U \subset \mathbb{R}^d$ be a bounded domain with a smooth boundary. For functions $u, v \in C^2(U) \cap C^1(\bar{U})$, Green's second identity states that

$$\int_U (u \nabla^2 v - v \nabla^2 u) dV = \oint_{\partial U} \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) dS, \quad \mathbf{x} \in U, \quad (43)$$

where $\partial/\partial n$ denotes the outward normal derivative with respect to the domain U . Taking $v(\mathbf{x}) = G(\mathbf{x}, \mathbf{x}')$, we obtain the well-known

Green's third identity in a bounded domain:

$$u(\mathbf{x}) = \oint_{\partial U} \left(u(\mathbf{x}') \frac{\partial G(\mathbf{x}', \mathbf{x})}{\partial n} - G(\mathbf{x}', \mathbf{x}) \frac{\partial u(\mathbf{x}')}{\partial n} \right) dS', \quad \mathbf{x} \in U. \quad (44)$$

Here, $\nabla^2 u = 0$ is assumed without loss of generality.

Then, let B_R denote the ball (or disk) centered at the origin with radius R , chosen such that $\Omega \subset B_R$. Given $U = B_R \setminus \overline{\Omega}$, the boundary ∂U consists of two closed surfaces: the inner boundary $\partial\Omega$ and the outer boundary ∂B_R . Accordingly, Eq. (44) can be rewritten as

$$u = \left(\mathcal{K}[u; \partial B_R] - \mathcal{V} \left[\frac{\partial u}{\partial n}; \partial B_R \right] \right) - \left(\mathcal{K}[u; \partial\Omega] - \mathcal{V} \left[\frac{\partial u}{\partial n}; \partial\Omega \right] \right), \quad (45)$$

where the definitions of the boundary integral operators can be found in Eqs. (10) and (11). Naturally, the Green's third identity in the exterior domain $\mathbb{R}^d \setminus \overline{\Omega}$ should be defined as the limit of the above expression as $R \rightarrow \infty$.

Consider the following far-field condition:

$$u(\mathbf{x}) \sim C G(\|\mathbf{x}\|), \quad \|\mathbf{x}\| \rightarrow \infty, \quad (46)$$

$$G(r) = \begin{cases} \frac{1}{2\pi} \log r, & d = 2, \\ -\frac{1}{4\pi} \frac{1}{r}, & d = 3, \end{cases} \quad (47)$$

which is equivalent to Eq. (1c) with C a constant. Given $\mathbf{x} \in \partial B_R$, differentiating yields

$$\frac{\partial u}{\partial n}(\mathbf{x}) \sim \frac{C}{S(R)}, \quad \|\mathbf{x}\| \rightarrow \infty, \quad (48)$$

in which $S(R)$ denotes the surface area of a d -dimensional sphere of radius R . Substituting these asymptotics into the boundary integral, we obtain

$$\begin{aligned} \lim_{R \rightarrow \infty} \mathcal{K}[u; \partial B_R](\mathbf{x}) &= C \lim_{R \rightarrow \infty} G(R) \int_{\partial B_R} \frac{\partial G(\mathbf{x}', \mathbf{x})}{\partial n} dS' \\ &= C \lim_{R \rightarrow \infty} G(R), \end{aligned} \quad (49)$$

where the last equality follows from Gauss's divergence theorem. On the other hand,

$$\begin{aligned} \lim_{R \rightarrow \infty} \mathcal{V} \left[\frac{\partial u}{\partial n}; \partial B_R \right](\mathbf{x}) &= C \lim_{R \rightarrow \infty} \frac{1}{S(R)} \int_{B_R} G(\mathbf{x}', \mathbf{x}) dS' \\ &= C \lim_{R \rightarrow \infty} G(R), \end{aligned} \quad (50)$$

where the final step follows from Newton's shell theorem.

In conclusion, under the standard far-field condition that shares the same order with the fundamental solution, the terms Eq. (49) and Eq. (50) cancel each other. As a result, the Green's third identity in the exterior domain reduces to the form given in Eq. (9).

C Additional Convergence Analysis

We have discussed the convergence of our iterative method for unbounded-domain extension in Sec. 6; however, a rigorous proof was provided only for the classical Poisson equation without cavities. In this appendix, we investigate several generalized cases quantitatively. For convenience in the eigenvalue analysis, we specialize the problem by assuming symmetric geometries—for example, the rectangular computational domain Ω commonly used in grid-based solvers is treated as spherical in the analysis. Nevertheless, one can show via bounding methods that the conclusions below remain qualitatively valid for domains with more complex geometries.

C.1 Dirichlet Boundary Cavity

We consider the following exterior Dirichlet problem for the classical Poisson equation:

$$\begin{cases} \nabla^2 u(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \notin \overline{\Omega_c}, \\ u(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \partial\Omega_c, \end{cases} \quad (51a)$$

$$\quad (51b)$$

where $\overline{\Omega_c}$ denotes the closure of the cavity domain. The far-field condition (1c) is implicitly imposed. During the iterative process, for all k , both the intermediate solution $u^{(k)}$ and the exact solution u satisfy the above problem. Therefore, the error function $\varepsilon^{(k)} = u^{(k)} - u$ satisfies a homogeneous version

$$\begin{cases} \nabla^2 \varepsilon^{(k)}(\mathbf{x}) = 0, & \mathbf{x} \notin \overline{\Omega_c}, \\ \varepsilon^{(k)}(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega_c. \end{cases} \quad (52a)$$

$$\quad (52b)$$

For $\varepsilon^{(k)}$, applying Green's third identity in $\Omega \setminus \Omega_c$ and taking the boundary limit yields

$$\mathcal{B}[\varepsilon^{(k)}](\mathbf{x}) = -\varepsilon^{(k)}(\mathbf{x}) + 2\mathcal{V} \left[\frac{\partial \varepsilon^{(k)}}{\partial n}; \partial\Omega_c \right](\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \quad (53)$$

This effectively relates the operator $\mathcal{B}$ to the error within Ω , but the operator acts only on the boundary trace of $\varepsilon^{(k)}$ on $\partial\Omega$. Consequently, in order to analyze the properties of $\mathcal{B}$, it is necessary to express the interior values of $\varepsilon^{(k)}$ in terms of boundary data, subject to Eqs. (52a)–(52b),

Concentric Balls (3D). Suppose that Ω and Ω_c are concentric balls with radii R and R_c , respectively, where $R > R_c$, and both are centered at the origin. In three-dimensional spherical coordinates, any solution of the Laplace equation (e.g., the error function ε , where the superscript (k) is omitted for brevity) can always be expanded in a separated-variable basis as

$$\varepsilon(r, \theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(A_{lm} r^l + \frac{B_{lm}}{r^{l+1}} \right) Y_l^m(\theta, \varphi), \quad (54)$$

where r , θ , and φ denote the radial distance, polar angle, and azimuthal angle, respectively. Here Y_l^m are the complex spherical harmonics of degree l and order m , and A_{lm} and B_{lm} are coefficients to be determined. As a result, the radial derivative of ε admits the following expansion:

$$\frac{\partial \varepsilon}{\partial r}(r, \theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(l A_{lm} r^{l-1} - \frac{(l+1) B_{lm}}{r^{l+2}} \right) Y_l^m(\theta, \varphi). \quad (55)$$

By setting $r = R_c$, we obtain the normal derivative on $\partial\Omega_c$.

Since the function values at $r = R$ are the objects acted upon by the operator, we define

$$C_{lm} = A_{lm} R^l + B_{lm} R^{-l-1}, \quad (56)$$

which are assumed to be known. On $\partial\Omega_c$, the function ε satisfies the homogeneous Dirichlet boundary condition $\varepsilon(R_c, \theta, \varphi) = 0$. Because spherical harmonics form an orthogonal and complete basis, this condition is equivalent to requiring that the coefficient of each spherical harmonic vanish, namely, for all l and m ,

$$A_{lm} R_c^l + B_{lm} R_c^{-l-1} = 0. \quad (57)$$

Solving Eqs. (56) and (57) simultaneously yields

$$\begin{cases} A_{lm} = \frac{R^{l+1}}{R^{2l+1} - R_c^{2l+1}} C_{lm}, \\ B_{lm} = -R_c^{2l+1} A_{lm}. \end{cases} \quad (58a)$$

$$(58b)$$

Substituting these expressions into Eq. (55), $\partial\epsilon/\partial n$ on Ω_c can be expressed in terms of the expansion coefficients of ϵ on $\partial\Omega$ as

$$\frac{\partial\epsilon}{\partial r}(R_c, \theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{(2l+1) R^{l+1} R_c^{l-1}}{R^{2l+1} - R_c^{2l+1}} C_{lm} Y_l^m(\theta, \varphi), \quad (59)$$

This relation is completely diagonal w.r.t. the coefficients C_{lm} .

By expanding the fundamental solution $G(\mathbf{x}, \mathbf{x}')$ in the same basis, one can further show that a source distributed as Y_{lm} on $r=R_c$ produces the following single-layer potential on $r=R$:

$$\mathcal{V}[Y_l^m; \partial\Omega_c](R, \theta, \varphi) = -\frac{R_c^{l+2}}{(2l+1) R^{l+1}} Y_l^m(\theta, \varphi). \quad (60)$$

Therefore, combining Eqs. (53), (59), and (60), we obtain

$$\mathcal{B}[\epsilon] = -\sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{R^{2l+1} + R_c^{2l+1}}{R^{2l+1} - R_c^{2l+1}} C_{lm} Y_l^m. \quad (61)$$

This shows that all eigenvalues of $\mathcal{B}$ are negative and no greater than $\lambda_{\infty} = -1$, with the smallest (most negative) eigenvalue $\lambda_0 = -(R + R_c)/(R - R_c)$ attained at $l=0$.

For relaxed iterations, the spectral radius $\max_l |1 + \omega(\lambda_l - 1)|$ must be strictly less than 1 and ideally as small as possible. This condition ensures convergence provided that $-1 < 1 + \omega(\lambda_0 - 1) < 1 + \omega(\lambda_{\infty} - 1) < 1$, which yields $0 < \omega < (R - R_c)/R$. The optimal choice is obtained by balancing the two extrema, i.e., $1 + \omega(\lambda_{\infty} - 1) = -(1 + \omega(\lambda_0 - 1))$, leading to $\omega = (R - R_c)/(2R - R_c)$. In summary, a larger ratio R_c/R requires a smaller relaxation parameter ω , which agrees with our experimental observations.

Concentric Disks (2D). In two dimensions, the analysis proceeds in an analogous manner. Let Ω and Ω_c be concentric disks with radii R and R_c , respectively. In the annular region between $\partial\Omega$ and $\partial\Omega_c$, ϵ admits the expansion

$$\epsilon(\rho, \theta) = \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \left(A_m \rho^{|m|} + \frac{B_m}{\rho^{|m|}} \right) e^{im\theta} + A_0 + B_0 \log \rho, \quad (62)$$

where ρ denotes the radial coordinate. As before, we define the following coordinates that are assumed known:

$$C_m = \begin{cases} A_m R^{|m|} + B_m R^{-|m|}, & m \neq 0, \\ A_0 + B_0 \log R, & m = 0. \end{cases} \quad (63a)$$

$$(63b)$$

Imposing the Dirichlet boundary condition on $\partial\Omega_c$, one can derive

$$\frac{\partial\epsilon}{\partial\rho}(R_c, \theta) = \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \frac{2|m|R^{|m|}R_c^{|m|-1}}{R^{2|m|} - R_c^{2|m|}} C_m e^{im\theta} + \frac{C_0}{R_c \log \frac{R}{R_c}}. \quad (64)$$

Note that in 2D the zero mode must always be treated separately.

By expanding the fundamental solution $G(\mathbf{x}, \mathbf{x}')$ in the same Fourier basis, one can further show that a source distributed as $e^{im\theta}$

on $\rho=R_c$ generates the following single-layer potential on $\rho=R$:

$$\mathcal{V}[e^{im\theta}; \partial\Omega_c](R, \theta) = \begin{cases} -\frac{R_c}{2|m|} \left(\frac{R_c}{R} \right)^{|m|} e^{im\theta}, & m \neq 0, \\ R_c \log R, & m = 0. \end{cases} \quad (65a)$$

$$(65b)$$

Combining Eqs. (53), (64), and (65), we obtain

$$\mathcal{B}[\epsilon] = -\sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \frac{R^{2|m|} + R_c^{2|m|}}{R^{2|m|} - R_c^{2|m|}} C_m e^{im\theta} + \frac{\log R + \log R_c}{\log R - \log R_c} C_0. \quad (66)$$

If the eigenvalue associated with the zero mode is ignored (its magnitude depends on the choice of zero point), all remaining eigenvalues of $\mathcal{B}$ are no greater than $\lambda_{\infty} = -1$. The smallest eigenvalue $\lambda_1 = -(R^2 + R_c^2)/(R^2 - R_c^2)$ is attained at $|m|=1$. It follows that the convergence condition for the relaxed iteration is $0 < \omega < (R^2 - R_c^2)/R^2$, with the optimal choice given by $\omega = (R^2 - R_c^2)/(2R^2 - R_c^2)$. The qualitative conclusion is therefore identical to the one obtained in the three-dimensional case.

C.2 Neumann Boundary Cavity

Similar to the previous subsection, we consider the following exterior Dirichlet problem for the classical Poisson equation:

$$\begin{cases} \nabla^2 u(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \notin \overline{\Omega_c}, \end{cases} \quad (67a)$$

$$\begin{cases} \frac{\partial u}{\partial n}(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \partial\Omega_c, \end{cases} \quad (67b)$$

where the far-field condition (1c) is implicitly imposed. During the iterative process, for all k , both the intermediate solution $u^{(k)}$ and the exact solution u satisfy the above problem. Therefore, the error function $\epsilon^{(k)} = u^{(k)} - u$ satisfies a homogeneous version, namely,

$$\begin{cases} \nabla^2 \epsilon^{(k)}(\mathbf{x}) = 0, & \mathbf{x} \notin \overline{\Omega_c}, \end{cases} \quad (68a)$$

$$\begin{cases} \frac{\partial \epsilon^{(k)}}{\partial n}(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega_c. \end{cases} \quad (68b)$$

For $\epsilon^{(k)}$, applying Green's third identity in $\Omega \setminus \Omega_c$ and taking the boundary limit yields

$$\mathcal{B}[\epsilon^{(k)}](\mathbf{x}) = -\epsilon^{(k)}(\mathbf{x}) - 2\mathcal{K}[\epsilon^{(k)}; \partial\Omega_c](\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \quad (69)$$

The remaining analysis largely parallels the approach in §C.1.

Concentric Balls (3D). We adopt exactly the same geometric assumptions and solution expansions as in §C.1. The only difference is that we now require, for all l and m ,

$$lA_{lm}R_c^{l-1} - (l+1)B_{lm}R_c^{-l-2} = 0. \quad (70)$$

As a result, the trace of ϵ on $\partial\Omega_c$ can be written as

$$\epsilon(R_c, \theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{(2l+1) R^{l+1} R_c^l}{(l+1) R^{2l+1} + l R_c^{2l+1}} C_{lm} Y_l^m(\theta, \varphi). \quad (71)$$

Taking into account that

$$\mathcal{K}[Y_l^m; \partial\Omega_c](R, \theta, \varphi) = -\frac{l R_c^{l+1}}{(2l+1) R^{l+1}} Y_l^m(\theta, \varphi), \quad (72)$$

we deduce that

$$\mathcal{B}[\epsilon] = -\sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{(l+1) R^{2l+1} - l R_c^{2l+1}}{(l+1) R^{2l+1} + l R_c^{2l+1}} C_{lm} Y_l^m(\theta, \varphi). \quad (73)$$

It follows that all eigenvalues of $\mathcal{B}$ are negative and lie in the interval $[-1, 0)$. Consequently, the iteration converges for any relaxation factor $\omega \in (0, 1)$. As the ratio R_c/R increases, the largest eigenvalue

approaches zero, implying that the optimal choice of ω becomes larger accordingly, which we observed during experiments indeed.

Concentric Disks (2D). We adopt exactly the same geometric assumptions and solution expansions as in §C.1, and impose a homogeneous Neumann boundary condition on $\partial\Omega_c$. This yields

$$\varepsilon(R_c, \theta) = \sum_{m=-\infty}^{\infty} \frac{2R_c^{|m|} R_c^{|m|}}{R_c^{2|m|} + R_c^{2|m|}} C_m e^{im\theta}. \quad (74)$$

Taking into account that

$$\mathcal{K}[e^{im\theta}; \partial\Omega_c](R, \theta) = \begin{cases} -\frac{R_c^m}{2R^m} e^{im\theta}, & m \neq 0, \\ 0, & m = 0, \end{cases} \quad (75a)$$

$$(75b)$$

we obtain

$$\mathcal{B}[\varepsilon] = - \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \frac{R^{2|m|} - R_c^{2|m|}}{R^{2|m|} + R_c^{2|m|}} C_m e^{im\theta}. \quad (76)$$

It follows that all eigenvalues of $\mathcal{B}$ are negative and lie in the interval $[-1, 0)$. The largest eigenvalue, $\lambda_1 = -(R^2 - R_c^2)/(R^2 + R_c^2)$, increases monotonically as the ratio R_c/R increases. The qualitative conclusions regarding the choice of the relaxation factor are therefore the same as in the three-dimensional case.

C.3 Varying Diffusion Coefficient

We consider the following Poisson problem with a spatially-varying diffusion coefficient:

$$\nabla \cdot a(x) \nabla u(x) = f(x). \quad (77)$$

The far-field condition (1c) is implicitly assumed throughout. During the iterative process, for all k , since both the intermediate solution $u^{(k)}$ and the exact solution u satisfy the above problem, the error function $\varepsilon^{(k)} = u^{(k)} - u$ satisfies its homogeneous version

$$\nabla \cdot a(x) \nabla \varepsilon^{(k)}(x) = 0. \quad (78)$$

Applying Green's third identity to $\varepsilon^{(k)}$ over the domain Ω and taking the boundary limit yields

$$\mathcal{B}[\varepsilon^{(k)}](x) = -\varepsilon^{(k)}(x) + \mathcal{U}[\varepsilon^{(k)}; \Omega](x), \quad x \in \partial\Omega, \quad (79)$$

$$\mathcal{U}[\phi; \Omega](x) = 2 \int_{\Omega} \nabla^2 \phi(x') G(x', x) dV'. \quad (80)$$

It is straightforward to verify that, for $\varepsilon^{(k)}$, the Laplacian satisfies $\nabla^2 \varepsilon^{(k)} = -\nabla a \cdot \nabla \varepsilon^{(k)} / a$.

We further specialize the diffusion coefficient a by assuming it is piecewise-constant, i.e., there exists a homogeneous inclusion occupying a subdomain $\Omega_i \subset \Omega^*$, within which $a = a_i$ is a constant, while $a = 1$ in the exterior region. Under this assumption, Eq. (78) reduces to

$$\begin{cases} \nabla^2 \varepsilon^{(k)}(x) = 0, & x \notin \partial\Omega_i, \\ \left[\left[a(x') \frac{\partial \varepsilon^{(k)}}{\partial n}(x') \right] \right] = 0, & x \in \partial\Omega_i, \end{cases} \quad (81)$$

$$(82)$$

where the unit normal $\mathbf{n}$ is taken to point outward from the interior of Ω_i , and $[\![\cdot]\!]$ denotes the jump across the interface (defined as the exterior limit minus the interior limit). In this case, the volume

integral appearing in the operator $\mathcal{U}$ can be equivalently rewritten as a surface integral over the material interface:

$$\mathcal{U}[\varepsilon^{(k)}; \Omega](x) = 2 \int_{\partial\Omega_i} \left[\frac{\partial \varepsilon^{(k)}}{\partial n}(x') \right] G(x', x) dS'. \quad (83)$$

In the remainder of this subsection, our quantitative analysis focuses on this representative configuration.

Concentric Balls (3D). We adopt geometric assumptions and solution expansions similar to those in §C.1. Specifically, we assume that Ω and Ω_i are concentric balls with radii R and R_i , respectively, where $R > R_i$, and both are centered at the origin. Clearly, the error function ε is harmonic in both $\Omega \setminus \Omega_i$ and Ω_i . It therefore admits the following spherical harmonic expansions:

$$\varepsilon(r, \theta, \varphi) = \begin{cases} \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(A_{lm} r^l + \frac{B_{lm}}{r^{l+1}} \right) Y_l^m(\theta, \varphi), & r > R_i, \\ \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(C_{lm} r^l + \frac{D_{lm}}{r^{l+1}} \right) Y_l^m(\theta, \varphi), & r < R_i. \end{cases} \quad (84a)$$

$$(84b)$$

We define the coefficients E_{lm} , which are assumed to be known, by

$$E_{lm} = A_{lm} R^l + B_{lm} R^{-l-1}. \quad (85)$$

Moreover, the regularity of ε at the origin requires that the solution remain bounded as $r \rightarrow 0$, which immediately implies $D_{lm} = 0$. By enforcing the continuity of ε across $r = R_i$ together with the jump condition (82) on the normal derivative, one can explicitly solve for the coefficients A_{lm} , B_{lm} , and C_{lm} . This yields

$$\llbracket \varepsilon(R_i) \rrbracket = - \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{(2l+1)(l-a_i) R_i^{l+1} R_i^{l-1} E_{lm} Y_l^m}{(l+a_i l+1) R_i^{2l+1} + (l-a_i l) R_i^{2l+1}}, \quad (86)$$

where the angular variables (θ, φ) have been omitted for brevity. This expression shows that the spherical harmonic coefficients of $\llbracket \varepsilon \rrbracket$ on $\partial\Omega_i$ depend diagonally on the coefficients E_{lm} .

Combining Eqs. (79), (83) and (86), and invoking the result of Eq. (60), we obtain

$$\mathcal{B}[\varepsilon] = - \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{(l+a_i l+1) R_i^{2l+1} - (l-a_i l) R_i^{2l+1}}{(l+a_i l+1) R_i^{2l+1} + (l-a_i l) R_i^{2l+1}} E_{lm} Y_l^m. \quad (87)$$

Consequently, as long as $a_i > 0$, all eigenvalues of the operator $\mathcal{B}$ are negative, and the iterative scheme is guaranteed to converge. The optimal choice of the relaxation factor ω is thus jointly influenced by both the contrast a_i and the inclusion radius R_i .

Concentric Disks (2D). We adopt geometric assumptions and solution expansions analogous to those in §C.1. Specifically, we assume that Ω and Ω_i are concentric disks with radii R and R_i , respectively, where $R > R_i$, and that both are centered at the origin. Under these assumptions, the error function ε admits the following Fourier expansions in polar coordinates:

$$\varepsilon(\rho, \theta) = \begin{cases} \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \left(A_m \rho^{|m|} + \frac{B_m}{\rho^{|m|}} \right) e^{im\theta} + E_0, & r > R_i, \\ \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} C_m \rho^{|m|} e^{im\theta} + E_0, & r < R_i. \end{cases} \quad (88a)$$

$$(88b)$$

Here, several terms have been eliminated a priori by enforcing regularity at the origin and continuity across the interface. We assume that the constant mode E_0 and the coefficients

$$E_m = A_m R^{|m|} + B_m R^{-|m|}, \quad m \neq 0 \quad (89)$$

are known. By combining the continuity of ε at $\rho = R_i$ with the jump condition (82) on the normal derivative, we obtain

$$\llbracket \varepsilon(R_i) \rrbracket = - \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \frac{2|m| (1 - a_i) R^{|m|} R_i^{|m|-1} E_m e^{im\theta}}{(1 + a_i) R^{2|m|} + (1 - a_i) R_i^{2|m|}}, \quad (90)$$

whose Fourier coefficients depend diagonally on E_m , in complete analogy with the three-dimensional case.

Combining Eqs. (79), (83) and (90), and invoking the result of Eq. (65), we arrive at

$$\mathcal{B}[\varepsilon] = - \sum_{\substack{m=-\infty \\ m \neq 0}}^{\infty} \frac{(1 + a_i) R^{2|m|} - (1 - a_i) R_i^{2|m|}}{(1 + a_i) R^{2|m|} + (1 - a_i) R_i^{2|m|}}. \quad (91)$$

Since all eigenvalues of $\mathcal{B}$ are again strictly negative, the qualitative behavior of the optimal relaxation factor is identical to that observed in the three-dimensional setting.