

# Kelvin Transformations for Simulations on Infinite Domains

MOHAMMAD SINA NABIZADEH, University of California, San Diego

RAVI RAMAMOORTHY, University of California, San Diego

ALBERT CHERN, University of California, San Diego

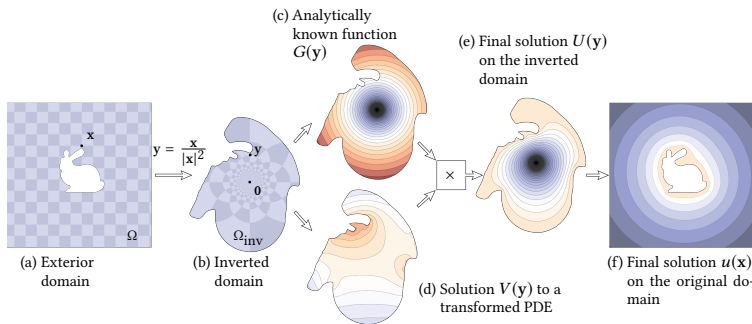


Fig. 1. Left: The proposed pipeline ((a)–(f)) for solving a PDE problem on an infinite exterior domain  $\Omega \subset \mathbb{R}^3$ . The method begins by representing the infinite domain (a) by a bounded domain (b) via the inversion map. Factoring out an analytically known function (c), we numerically solve a specifically transformed PDE (d) on the bounded domain. The solution to the original problem (e) is evaluated on the inverted domain as the product of the functions from (c) and (d). The final solution (f) is obtained by inverting (e) back to the original infinite domain. Right: The acoustic wave governed by the Helmholtz equation on the infinite 3D domain exterior to a vibrating Stanford Bunny, computed by our method. The solution is visualized as a height field over a 2D slice. (Note that the image is not of water waves despite its resemblance.) For the first time, an infinite domain Helmholtz equation is computed on a grid without the aid of artificial absorbing boundary conditions.

Solving partial differential equations (PDEs) on infinite domains has been a challenging task in physical simulations and geometry processing. We introduce a general technique to transform a PDE problem on an unbounded domain to a PDE problem on a bounded domain. Our method uses the Kelvin Transform, which essentially inverts the distance from the origin. However, naive application of this coordinate mapping can still result in a singularity at the origin in the transformed domain. We show that by factoring the desired solution into the product of an analytically known (asymptotic) component and another function to solve for, the problem can be made continuous and compact, with solutions significantly more efficient and well-conditioned than traditional finite element and Monte Carlo numerical PDE methods on stretched coordinates. Specifically, we show that every Poisson or Laplace equation on an infinite domain is transformed to another Poisson (Laplace) equation on a compact region. In other words, any existing Poisson solver on a bounded domain is readily an infinite domain Poisson solver after being wrapped by our transformation. We demonstrate the integration of our method with finite difference and Monte Carlo PDE solvers, with applications in the fluid pressure solve and simulating electromagnetism, including visualizations of the solar magnetic field. Our transformation technique also applies to the Helmholtz equation whose solutions oscillate out to infinity. After the transformation, the Helmholtz equation becomes a tractable equation on a bounded domain without infinite oscillation. To our

Authors' addresses: Mohammad Sina Nabizadeh, University of California, San Diego; Ravi Ramamoorthi, University of California, San Diego; Albert Chern, University of California, San Diego.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2021 Copyright held by the owner/author(s).

0730-0301/2021/8-ART97

<https://doi.org/10.1145/3450626.3459809>

knowledge, this is the first time that the Helmholtz equation on an infinite domain is solved on a bounded grid without requiring an artificial absorbing boundary condition.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Mathematics of computing** → **Partial differential equations**; **Numerical analysis**.

Additional Key Words and Phrases: Kelvin transform, Möbius transform, Poisson equation, Helmholtz equation, infinite domain, Monte Carlo method

## ACM Reference Format:

Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin Transformations for Simulations on Infinite Domains. *ACM Trans. Graph.* 40, 4, Article 97 (August 2021), 15 pages. <https://doi.org/10.1145/3450626.3459809>

## 1 INTRODUCTION

Solving partial differential equations (PDEs) on infinite domains is a common problem in physical simulations and geometry processing. For example, simulating acoustic wave propagation often demands solving a Helmholtz equation on an open space (Figure 1). Designing an incompressible flow around an obstacle immersed in a vast ocean would involve a pressure solve—a Poisson equation—on the entire unbounded domain (Figure 2). Inpainting or extrapolating a function to the surrounding environment from a geometry usually corresponds to a PDE problem on the infinite domain (Figure 15).

Problems on unbounded domains pose a fundamentally different set of numerical challenges compared to those on compact domains. This contrast is manifested by the fact that many numerical methods do not directly apply to unbounded domains without special treatments or compromises. For instance, solvers relying on discretizing

the space require the domain to be truncated into a finite size, introducing artificial boundaries [Tsynkov 1998]. These boundaries are imposed with conditions that either approximate the behavior of the solution at infinity, or match an outer regime simulation [Stomakhin and Selle 2017] or analytic model (including absorbing layers) [Bojsen-Hansen and Wojtan 2016]. Mesh-free methods may also require special treatments in infinite domains. For example, Monte-Carlo-based PDE solvers [Sawhney and Crane 2020] for the direct evaluation of the solution usually involve a random walk in the domain, which converges much more slowly as the walker likely wanders to infinity.

In this paper, we introduce a technique to rewrite a PDE on an infinite domain as a PDE on a compact domain. After such a transformation, standard numerical methods on compact domains apply. We refer to this transformation as the *Kelvin transform*, since it agrees with the historical Kelvin transform in the case of the Laplace and Poisson equations [Kellogg 1953].

We demonstrate the Kelvin transform for the Laplace and Poisson equations on infinite domains, as these problems underpin the fundamental building blocks in 3D data processing and many physical simulations such as fluids, electromagnetism, thermostatics and gravitation. We also include examples of a generalized Kelvin transform applied to the Helmholtz equation (Figure 1), which is the basis for acoustic simulation. This demonstrates the possibility to compute such oscillatory fields on an outer domain without requiring any artificial wave-absorbing boundary.

This paper contributes the following novel concepts and applications:

**Preconditioning PDEs.** As a classical trick, a PDE problem for a function  $u(\mathbf{x})$  on an exterior domain can be written as a bounded domain problem by a change of coordinates [Grosch and Orszag 1977]. One such example, as will be taken throughout this paper, is the inversion map  $\mathbf{x} \mapsto \mathbf{y} = \mathbf{x}/|\mathbf{x}|^2$  (Figure 1 (b)). Let  $U(\mathbf{y}) = u(\mathbf{x}) = u(\mathbf{y}/|\mathbf{y}|^2)$  represent the value of  $u$  as a function on the bounded inverted coordinates. (We will consistently use a small letter for a function depending on the original coordinates  $\mathbf{x}$ , and use its capital counterpart for the same function but dependent on the inverted coordinates  $\mathbf{y}$ . See Table 1.) However, such a coordinate stretching comes with a cost that the new PDE to solve for  $U(\mathbf{y})$  is often singular around the point mapped to infinity. The difficulty from an infinite domain persists in the form of singularities.

The key idea of this paper, detailed in Section 4, is to further factorize  $U(\mathbf{y})$  (Figure 1(e)) into

$$U(\mathbf{y}) = G(\mathbf{y})V(\mathbf{y}), \quad (1)$$

where

- (1)  $G(\mathbf{y})$  is analytically given and captures the asymptotic behavior of the solution near the singularity (Figure 1(c)), while
- (2)  $V(\mathbf{y})$  is the part to be solved numerically (Figure 1(d)).

It turns out that by factoring out  $G(\mathbf{y})$ , the new PDE satisfied by  $V(\mathbf{y})$  has the singularity removed. We have therefore transformed the problem into a regular PDE for  $V(\mathbf{y})$  on a genuinely compact domain where standard numerical methods apply. Finally, we can reconstruct the original function of interest  $U(\mathbf{y})$  by multiplying  $V(\mathbf{y})$  with  $G(\mathbf{y})$ .

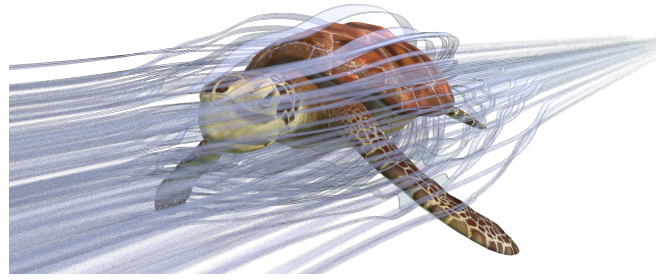


Fig. 2. A laminar flow around a sea turtle,<sup>1</sup> computed by a pressure projection of a constant vector field on the infinite domain. Such a task is made possible by the Kelvin transformation.

Note that for many existing PDE problems, the function  $G(\mathbf{y})$  is known in advance from the mathematical physics literature for the particular PDE problem. In this paper, we focus on Poisson and Helmholtz equations, where the respective asymptotic form and hence their  $G$ 's are known explicitly. If the reader has another specific problem, with a better physical understanding, an alternative form of  $G$  may apply, leading to new compactified PDEs for  $V(\mathbf{y})$ .

While it is true that the Kelvin transform for the Poisson equation has long been known [Thomson 1845; Kellogg 1953], and that the factorization (1) is a standard analytical method in both areas of ordinary differential equations (ODEs) [Bender and Orszag 2013, Chapter 3.3] and high frequency perturbation theory [Bender and Orszag 2013, Chapter 10], we have found no previous discussions on these techniques applied to solving the exterior domain problem in computer graphics or scientific computing.

**Poisson Problems on Exterior Domains.** Using the Kelvin transform, every exterior domain Poisson problem for  $u(\mathbf{x})$  becomes a standard Poisson problem for  $V(\mathbf{y})$  on a bounded domain. This means that one can apply any favorite PDE solver for an interior domain problem, such as a finite difference method on a regular grid (Section 5.1). We provide explicit formulae for how the two boundary value problems correspond to each other.

**Monte Carlo Method on Infinite Domains.** As a discretization-free method for solving the Laplace equation, the Monte Carlo method relies on simulating random walks in the domain until exiting at some boundary point [Muller 1956; Ermakov et al. 1989; Sawhney and Crane 2020]. Such random walks are efficient for an interior domain problem. However, in an infinite domain, most random walkers wander off to infinity [Weisstein 2000] and thus one must reseed the walker [Ermakov et al. 1989, §5.2] or apply a Russian Roulette procedure [Sawhney and Crane 2020] to let most particles self-decay, resulting in high variance and long sampling time. As demonstrated in Section 5.2, the Kelvin transform turns the exterior problem into an interior one, making the Monte Carlo method on an exterior domain as efficient as that on an interior domain.

**Fluids and Sounds in Open Spaces.** We provide two proof-of-concept applications to physical simulations. One such application is the pressure solve for fluids on unbounded domains. Since the pressure projection boils down to a Poisson equation, our method solving infinite domain Poisson problems directly applies to the pressure solve (Section 5.3.2, Figures 2 and 14). Another example we

<sup>1</sup>3D model by chung\_the\_artist on Sketchfab platform.

discuss (Section 6) is the wave-based acoustic simulation [James 2016], where one often needs to solve a Helmholtz equation in an infinite space. Note that since the solutions to the Helmholtz equation oscillate out to infinity, it has been generally believed that such a system is not suitable for coordinate mapping [Grosch and Orszag 1977; Givoli 1992]. By factoring out the correct factor  $G(\mathbf{y})$  in (1), we transform the Helmholtz equation into a tractable equation for  $V(\mathbf{y})$  on a bounded domain to be solved on a finite resolution grid (Figures 1 and 16). To our knowledge, this is the first time a Helmholtz equation is solved on the *entire infinite domain* using only a finite difference method without relying on absorbing boundary conditions (perfectly matched layers).

## 2 RELATED WORK

The problem of numerically solving PDEs on an infinite domain has long been recognized in scientific computing and computer graphics. Here, we mention a few general methods that tackle this problem. For a comprehensive review, we also refer the readers to [Givoli 1992] and [Tsynkov 1998].

*Boundary Integrals.* An integral method may be employed to avoid discretizing the infinite domain. Assuming the fundamental solution (Green's function) to the PDE on the free space is known, one may directly construct the solution on an infinite domain by integrating the Green's function over the source. Integral methods have been applied to surface reconstructions [Barill et al. 2018], water wave animations [Schreck et al. 2019], and evaluation of fluid velocities from vortex filaments [Weißmann and Pinkall 2010], vortex sheets [Brochu et al. 2012; Da et al. 2015, 2016], vortex particles [Golas et al. 2012; Zhang and Bridson 2014], or a vorticity field [Zhang et al. 2015], just to name a few. These integrations are straightforward in a free space. When there is a domain boundary, one then uses the integral kernel to reduce the volumetric PDE to a *boundary integral equation* (BIE) [McLean 2000]. The *boundary element method* (BEM) [Sauter and Schwab 2010; Kythe 2020] is the numerical method that discretizes and solves the boundary value problem in a BIE form. In particular, the differential equation in the domain (sparse 3D system) is replaced by an integral equation over the boundary (dense 2D system). This is attractive when discretizing the volumetric domain is difficult, making BEM one of the actively researched numerical methods for infinite domain problems. Applications of BEM include acoustic transfers [James et al. 2006; Kirkup 2019], electromagnetic scattering [Spindler 2016], diffusion processes [Claeys et al. 2017], inhomogeneous PDEs [Partridge et al. 2012], etc. In fluid animations, determining the vortex sheets on the surface in a vortex simulation [Weißmann and Pinkall 2010; Golas et al. 2012] can also be classified as a BEM.

One caveat of the integral methods is that the large summation alone is computationally expensive, besides the cost of BEM. A direct evaluation of the solution on  $N$  nodes over a source of  $M$  elements takes  $O(NM)$  flops [James et al. 2006; Zhang et al. 2015], which can be more expensive than solving the volumetric sparse linear system with a cost of  $O(N)$  [Harari and Hughes 1992]. Therefore, practical integral methods require more sophisticated summation programs such as the fast multipole method [Greengard and Rokhlin

1987; Barill et al. 2018] and a mesh-assisted acceleration [Zhang and Bridson 2014].

*Domain Truncation.* Instead of integrating Green's functions, a more general approach is to take the original PDE problem and to truncate the infinite domain into a finite size. The newly introduced truncation boundary requires a carefully designed boundary condition such that the solution seamlessly continues beyond the truncation. The boundary artifacts are less pronounced for the *natural boundary conditions* in the case of higher order elliptic PDEs [Stein et al. 2018; Sato et al. 2018]. For a lower order PDE, such as the pressure solve in fluid simulations, one approximates the boundary condition by a Neumann boundary condition [Stomakhin and Selle 2017], a Dirichlet boundary condition [Nielsen and Bridson 2011] or a value given by the pressure inferred from simulating a reduced fluid simulation on the larger domain [Thürey et al. 2006; Stomakhin and Selle 2017]. However, these boundary conditions do not correspond to the true value of the pressure should the domain be truly infinite. The exact boundary condition for an infinite domain Poisson equation is a Dirichlet-to-Neumann (DtN) condition [Givoli and Keller 1989]. However, imposing a DtN condition leads to a more expensive and highly connected non-local system.

Oscillatory systems such as the Helmholtz equation and wave equations are more sensitive to the choice of the boundary condition. *Absorbing boundary conditions* (ABCs) [Engquist and Majda 1977] and *perfectly matched layers* (PMLs) [Berenger 1994; Chern 2019] are the suitable boundary conditions that reduce the reflection waves. These non-reflecting boundary conditions have been applied to fluid surface simulations [Söderström et al. 2010; Bojsen-Hansen and Wojtan 2016] and wave-based acoustic synthesis [James 2016; Wang et al. 2018] in an open space.

*Coordinate Mappings.* In the method of coordinate mapping or stretching, one finds a parameterization of the infinite domain by a curvilinear coordinate system with bounded coordinate values, and re-expresses the PDE on the finite coordinates. After this coordinate transformation one discretizes the PDE on the bounded domain. As investigated by Grosch and Orszag [1977], the method is useful only if the solution vanishes rapidly at infinity. They remark that the solutions oscillating out to infinity are not amenable to these techniques. Our method transforms not only the domain but also the variable function. The latter is the key to overcome the limitations of coordinate mappings.

*Kelvin Transform.* The classical *Kelvin transform* was discovered by William Thomson (Lord Kelvin) in 1845 [Thomson 1872]. It is also known as the *method of image*. The Kelvin transform is usually described as a symmetric property of the harmonic potential in electrostatics under the spherical reflection, *i.e.* the inversion map. Kelvin transform has been a tool to generate harmonic functions on domains with spherical boundaries [Kellogg 1953]. However, the Kelvin transformation has only been applied numerically to domains composed of spheres and circles [Reali et al. 1984; Gold 2020] and has drawn little attention beyond that. The current paper shows that Kelvin transformation works on arbitrary domains. Moreover, a modified Kelvin transform is able to turn an oscillatory solution on an infinite domain to a regular function on the inverted domain.

### 3 BACKGROUND

The Kelvin transformation technique in this paper involves a change of coordinates via an inversion. We use this section to review how a PDE transforms under such a coordinate change as well as to establish the notations (Table 1). Note that a mere change of coordinates to reformulate a PDE problem is called *the method of coordinate stretching* [Grosch and Orszag 1977]. In this section we also remind the readers of some caveats in a plain coordinate stretching method, in order to demonstrate the importance of the additional key steps in the Kelvin transforms explained in Section 4.

#### 3.1 Change of Coordinates under the Inversion

Let  $\Omega \subset \mathbb{R}^3$  denote the infinite domain on which our PDE problem is defined. We assume that  $\Omega$  is a domain exterior to some objects of finite sizes; that is,  $\Omega$  is a proper subset of  $\mathbb{R}^3$  (i.e.  $\Omega \subsetneq \mathbb{R}^3$ ) and  $\mathbb{R}^3 \setminus \Omega$  is bounded (Figure 1(a)). Without loss of generality, we assume  $\mathbf{0} \notin \Omega$  (which can be achieved by a translation).

To be concrete, consider the Poisson equation

$$\Delta u(\mathbf{x}) = \nabla \cdot \nabla u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, x_3) \in \Omega. \quad (2)$$

accompanied by the vanishing condition at infinity

$$u(\mathbf{x}) \rightarrow 0, \quad \text{for } |\mathbf{x}| \rightarrow \infty. \quad (3)$$

Now, map the infinite domain  $\Omega \subset \mathbb{R}^3$  to a finite one  $\Omega_{\text{inv}} \subset \mathbb{R}^3$  via a map  $\Phi: \Omega_{\text{inv}} \rightarrow \Omega$ . (A discrete analog is that the space is discretized by an adaptive mesh, which is coarser near the infinity.) Such a parameterization of the unbounded domain  $\Omega$  by a bounded one  $\Omega_{\text{inv}} = \Phi^{-1}(\Omega)$  can be accomplished by the inversion map

$$\Phi(\mathbf{y}) = \frac{\mathbf{y}}{|\mathbf{y}|^2}, \quad \mathbf{y} = (y_1, y_2, y_3) \in \Omega_{\text{inv}} \subset \mathbb{R}_*^3, \quad (4a)$$

$$\Phi^{-1}(\mathbf{x}) = \frac{\mathbf{x}}{|\mathbf{x}|^2}, \quad \mathbf{x} = (x_1, x_2, x_3) \in \Omega \subset \mathbb{R}_*^3. \quad (4b)$$

Here,  $\mathbb{R}_*^3 = \mathbb{R}^3 \setminus \{\mathbf{0}\}$ . The coordinate origin, i.e. the center of the inversion, can be chosen arbitrarily. In this paper, we pick the origin manually as far away as possible from the boundary. Through the inversion map, we pull the values of  $u$  and  $f$  back to the inverted domain. For clarity, we use capital letters to denote the functions dependent on the inverted coordinate  $\mathbf{y} \in \Omega_{\text{inv}}$ . In particular,  $U(\mathbf{y}) = u(\Phi(\mathbf{y}))$  and  $F(\mathbf{y}) = f(\Phi(\mathbf{y}))$ .

The corresponding equation for  $U(\mathbf{y})$  on the finite inverted domain is given by

$$\nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y}) \right) = \frac{1}{|\mathbf{y}|^6} F(\mathbf{y}), \quad \mathbf{y} = (y_1, y_2, y_3) \in \Omega_{\text{inv}}. \quad (5)$$

When  $\nabla$  is applied on functions on the  $(y_1, y_2, y_3)$  coordinates (i.e. functions with capital letters), we parse  $\nabla = (\frac{\partial}{\partial y_1}, \frac{\partial}{\partial y_2}, \frac{\partial}{\partial y_3})$ , which is the vector differential operator under the Euclidean metric on the inverted space. See Appendix A for the derivation of (5).

#### 3.2 Direct Approaches on Stretched Coordinates

Now (5) appears to be a PDE problem on a bounded domain. However, difficulties persist when directly solving (5).

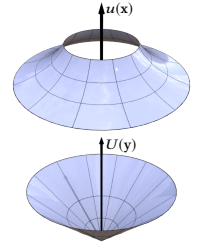
- After the change of coordinates, Eq. (5) is singular as  $\mathbf{y} \rightarrow \mathbf{0}$  (which corresponds to infinity in the original domain). Even though  $\Omega_{\text{inv}}$  appears to be a bounded domain, it is

Table 1. Notations

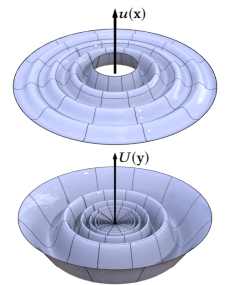
| Symbol   | Definition/ assumption  | Meaning   |
|--|---|---|
| $\mathbb{R}_*^3$   | $\mathbb{R}_*^3 = \mathbb{R}^3 \setminus \{\mathbf{0}\}$  | Space excluding the origin.   |
| $ \mathbf{x}  =  (x_1, x_2, x_3) $                         | $ \mathbf{x}  = \sqrt{x_1^2 + x_2^2 + x_3^2}$   | Euclidean distance to the origin.   |
| $\Phi: \mathbb{R}_*^3 \rightarrow \mathbb{R}_*^3$          | $\Phi(\mathbf{y}) = \mathbf{y}/ \mathbf{y} ^2$  | Inversion map.  |
| $\Omega \subset \mathbb{R}_*^3$                            | $\mathbb{R}_*^3 \setminus \Omega$ is bounded.   | Given exterior domain.  |
| $\Omega_{\text{inv}} \subset \mathbb{R}_*^3$               | $\Omega_{\text{inv}} = \Phi^{-1}(\Omega)$   | Inverted domain.  |
| $\Sigma \subset \mathbb{R}_*^3$                            | $\Sigma = \partial\Omega$   | Boundary surfaces of the exterior and inverted domains.   |
| $\Sigma_{\text{inv}} \subset \mathbb{R}_*^3$               | $\Sigma_{\text{inv}} = \Phi^{-1}(\Sigma)$   |   |
| $\mathbf{x} = (x_1, x_2, x_3)$                             | $\mathbf{x} = \Phi(\mathbf{y}) \in \Omega$  | Position vectors in the exterior and inverted domains.  |
| $\mathbf{y} = (y_1, y_2, y_3)$                             |   |   |
| $f: \Omega \rightarrow \mathbb{R}$                         | $F(\mathbf{y}) = f(\Phi(\mathbf{y}))$   | Functions on the exterior and the inverted domains.   |
| $F: \Omega_{\text{inv}} \rightarrow \mathbb{R}$            |   |   |
| $\mathbf{n}: \Sigma \rightarrow \mathbb{R}^3$              | $ \mathbf{n}  =  \mathbf{N}  = 1$ .   | Unit outward normals for the respective domains.  |
| $\mathbf{N}: \Sigma_{\text{inv}} \rightarrow \mathbb{R}^3$ |   |   |
| $\nabla f$   | $(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3})$             | The gradients of functions on the respective domains.   |
| $\nabla F$   | $(\frac{\partial F}{\partial y_1}, \frac{\partial F}{\partial y_2}, \frac{\partial F}{\partial y_3})$             |   |
| $\nabla \cdot (f_1, f_2, f_3)$                             | $\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} + \frac{\partial f_3}{\partial x_3}$       | The divergences of vector fields on the respective domains.   |
| $\nabla \cdot (F_1, F_2, F_3)$                             | $\frac{\partial F_1}{\partial y_1} + \frac{\partial F_2}{\partial y_2} + \frac{\partial F_3}{\partial y_3}$       |   |
| $\Delta f$   | $\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \frac{\partial^2 f}{\partial x_3^2}$ | The Laplacians of functions on the respective domains.  |
| $\Delta F$   | $\frac{\partial^2 F}{\partial y_1^2} + \frac{\partial^2 F}{\partial y_2^2} + \frac{\partial^2 F}{\partial y_3^2}$ |   |
| $u: \Omega \rightarrow \mathbb{R}$                         | $U(\mathbf{y}) = u(\Phi(\mathbf{y}))$   | The solution to the PDE problem.  |
| $U: \Omega_{\text{inv}} \rightarrow \mathbb{R}$            |   |   |
| $g: \Omega \rightarrow \mathbb{R}$                         | $G(\mathbf{y}) = g(\Phi(\mathbf{y}))$   | A given function sharing the same asymptotics with the PDE solution near $ \mathbf{x}  \rightarrow \infty$ ( $\mathbf{y} \rightarrow \mathbf{0}$ ). |
| $G: \Omega_{\text{inv}} \rightarrow \mathbb{R}$            |   |   |
| $v: \Omega \rightarrow \mathbb{R}$                         | $u(\mathbf{x}) = g(\mathbf{x})v(\mathbf{x})$  | The to-be-solved function satisfying the Kelvin transformed PDE.  |
| $V: \Omega_{\text{inv}} \rightarrow \mathbb{R}$            | $U(\mathbf{y}) = G(\mathbf{y})V(\mathbf{y})$  |   |

not compactified since the point representing infinity should still be excluded from the domain. For (5), a point-boundary condition  $U|_{\mathbf{y}=\mathbf{0}} = 0$  is required.

- The solution to (5) loses smoothness near the origin. For example, after the inversion, the harmonic function  $u(\mathbf{x}) = 1/|\mathbf{x}|$  becomes  $U(\mathbf{y}) = |\mathbf{y}|$ , which has a cone singularity at the origin (see insets). This may lead to a drop in the order of convergence if the discretization method assumes a uniformly bounded Taylor expansion for the solution (e.g. in a finite difference method).



- Such a coordinate stretching works only if the expected solution  $U$  is sufficiently well-behaved at the origin [Grosch and Orszag 1977]. For equations that yield oscillatory solutions, such as the Helmholtz equation governing acoustic waves, the inversion yields an infinite spatial frequency around the origin, requiring an infinite numerical resolution to resolve each wavefront. The insets show a radiating wave  $u(\mathbf{x}) = \frac{1}{|\mathbf{x}|} \cos(|\mathbf{x}|)$  and its inversion  $U(\mathbf{y}) = |\mathbf{y}| \cos(\frac{1}{|\mathbf{y}|})$ .



All of these limitations will be overcome by the strategy described in the next section.

## 4 OUR APPROACH

In the previous section, we see that expressing the PDE problem on a stretched coordinate system (or likewise deriving a finite difference or finite element scheme on an adaptive computational grid) yields a singular PDE near the point of infinity. We tackle this problem here.

We proceed with a “preconditioning” step to reformulate the equation. We decompose the solution  $u(\mathbf{x})$  (or  $U(\mathbf{y})$ ) into

$$u(\mathbf{x}) = g(\mathbf{x})v(\mathbf{x}), \quad \text{or} \quad U(\mathbf{y}) = G(\mathbf{y})V(\mathbf{y}), \quad (6)$$

where

- (i)  $g(\mathbf{x})$ , or its inverted counterpart  $G(\mathbf{y}) = g(\Phi(\mathbf{y}))$ , is analytically derived (or guessed) and captures the asymptotic behavior of the solution  $u$  near infinity; and
- (ii) the remaining quotient  $v(\mathbf{x})$ , or  $V(\mathbf{y}) = v(\Phi(\mathbf{y}))$ , is to be solved numerically.

By performing this change of variables from  $U$  to  $V$ , we effectively remove the singularity of the equation at the point representing infinity. The resulting equation for  $V(\mathbf{y})$  on the inverted domain  $\Omega_{\text{inv}}$  extends to a genuinely compactified domain  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$ .

### 4.1 Poisson Equation and the Classical Kelvin Transform

In the example of a Poisson equation (5), the asymptotic behavior of  $U(\mathbf{y})$  is known to be  $U(\mathbf{y}) \sim O(|\mathbf{y}|)$  as  $\mathbf{y} \rightarrow \mathbf{0}$ . In other words,  $u(\mathbf{x}) \sim O(1/|\mathbf{x}|)$  as  $|\mathbf{x}| \rightarrow \infty$  [Finn and Gilbarg 1957; Meyers 1963]. Therefore, for Poisson equations, we let  $G(\mathbf{y}) = |\mathbf{y}|$  in (6):

$$V(\mathbf{y}) = \frac{1}{|\mathbf{y}|}U(\mathbf{y}). \quad (7)$$

**Theorem 1.** *Written in terms of  $V(\mathbf{y})$ , the exterior domain Poisson problem (2) (or (5)) with condition (3) is equivalent to a Poisson equation with a standard Laplacian on a compact interior domain*

$$\Delta V = \nabla \cdot \nabla V = \frac{1}{|\mathbf{y}|^5}F(\mathbf{y}), \quad \mathbf{y} \in \Omega_{\text{inv}} \cup \{\mathbf{0}\}. \quad (8)$$

PROOF. See Appendix B. □

In many scenarios the right-hand side  $F(\mathbf{y})$  vanishes in a neighborhood of the origin, *i.e.*  $f(\mathbf{x})$  has bounded support. Hence,  $F(\mathbf{y})/|\mathbf{y}|^5$  is usually a regular right-hand side throughout the domain.

We remark that (8) is a *standard Poisson problem on a compact domain* for which any numerical method for the interior domain problem applies. In practice, solving the Poisson equation or Laplace equation ( $f = 0$ ) on an exterior domain only requires a wrapper Algorithm 1 around a Poisson solver on interior domains.

---

#### Algorithm 1 Poisson equation solver on an exterior domain

---

**Input:** Poisson problem (2), (3) on an infinite domain, given a boundary condition.

- 1:  $V \leftarrow$  Solve the interior problem (8) with appropriate boundary conditions for  $V$  (Section 4.1.1).
- 2:  $U(\mathbf{y}) = |\mathbf{y}|V(\mathbf{y})$ .
- 3:  $u(\mathbf{x}) = U(\frac{\mathbf{x}}{|\mathbf{x}|^2})$ .

**Output:**  $u(\mathbf{x})$ .

---

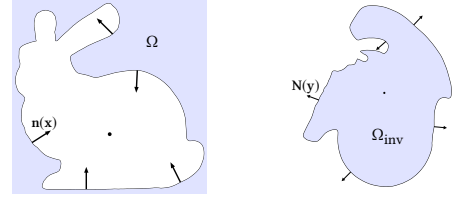


Fig. 3. The convention of the boundary normal vectors. Normals  $\mathbf{n}$  to the boundary  $\Sigma$  of the exterior domain  $\Omega$  point towards the origin (left). Normals  $\mathbf{N}$  to the boundary  $\Sigma_{\text{inv}}$  of the inverted domain  $\Omega_{\text{inv}}$  point away from the origin (right).

**4.1.1 Boundary Conditions.** The Poisson problem (2) must be accompanied by a boundary condition on the surface  $\Sigma = \partial\Omega$ . The associated boundary condition for (8) is derived from the relation (7). Let  $\Sigma_{\text{inv}} = \Phi^{-1}(\Sigma)$  denote the boundary of  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$ . A Dirichlet boundary condition for a given data  $p: \Sigma \rightarrow \mathbb{R}$

$$u(\mathbf{x}) = p(\mathbf{x}), \quad \mathbf{x} \in \Sigma, \quad (9)$$

corresponds to a Dirichlet boundary condition for  $V$ :

$$V(\mathbf{y}) = \frac{1}{|\mathbf{y}|}p(\Phi(\mathbf{y})), \quad \mathbf{y} \in \Sigma_{\text{inv}}. \quad (10)$$

A Neumann boundary condition for a given Neumann data  $q: \Sigma \rightarrow \mathbb{R}$

$$\mathbf{n}(\mathbf{x}) \cdot \nabla u(\mathbf{x}) = q(\mathbf{x}), \quad \mathbf{x} \in \Sigma, \quad (11)$$

corresponds to a Robin boundary condition for  $V$  (see Appendix B.1)

$$\mathbf{N}(\mathbf{y}) \cdot \nabla V(\mathbf{y}) + \frac{\mathbf{N}(\mathbf{y}) \cdot \mathbf{y}}{|\mathbf{y}|^2}V(\mathbf{y}) = \frac{1}{|\mathbf{y}|^3}q(\Phi(\mathbf{y})), \quad \mathbf{y} \in \Sigma_{\text{inv}}. \quad (12)$$

Here  $\mathbf{n}$  is the unit normal vector of  $\Sigma$  pointing away from  $\Omega$  (pointing towards the origin), and  $\mathbf{N}$  is the unit outward normal vector of  $\Sigma_{\text{inv}}$  (pointing away from the origin), both normalized by the Euclidean metric in the respective domains (Figure 3). Note that Robin boundary conditions are straightforward to implement (Appendix C).

## 5 RESULTS AND APPLICATIONS

In this section, we show the advantages of applying Algorithm 1 when we solve the Poisson and Laplace equations on infinite domains in practice. Afterwards, in Section 6, we apply our method to the Helmholtz equation which admits oscillatory solutions.

### 5.1 Efficient Grid- and Mesh-based Solver

In Algorithm 1, all that is required is to solve a Poisson or Laplace equation (8) in the interior domain, which can be solved by a standard finite difference or finite element method.

**5.1.1 Finite Difference Method.** Here we demonstrate a grid-based finite difference solution. The Laplacian is discretized into a 7-stencil central difference

$$\Delta V(\mathbf{y}) \approx \sum_{i=1}^3 \frac{-2V(\mathbf{y}) + V(\mathbf{y} + h\mathbf{e}_i) + V(\mathbf{y} - h\mathbf{e}_i)}{h^2} \quad (13)$$

on each grid node  $\mathbf{y}$  within  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$ . Here,  $h$  is the grid size, and  $\mathbf{e}_i$  is the unit coordinate vector (for the inverted domain).

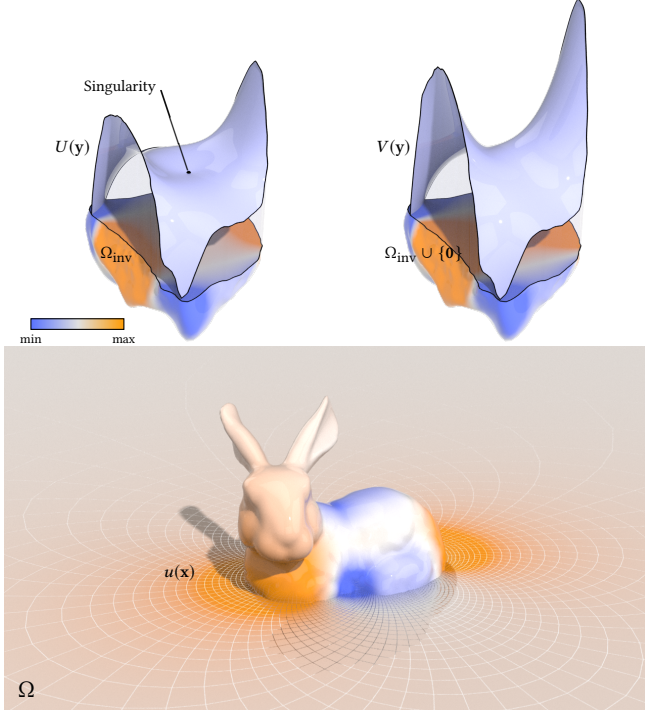


Fig. 4. Computing a harmonic function  $u(x)$  on the exterior 3D space  $\Omega$  of the bunny (bottom) on a stretched coordinate via inversion is equivalent to solving  $U(y)$  (top-left) on the inverted domain  $\Omega_{\text{inv}}$ , which exhibits a singularity at the origin. After Kelvin transform the singular factor is removed, leaving us a regular Laplace problem for  $V(y)$  on a compact domain  $\Omega_{\text{inv}} \cup \{0\}$  (top-right). Both original and exterior domain functions are visualized on a 2D slice at the same height.

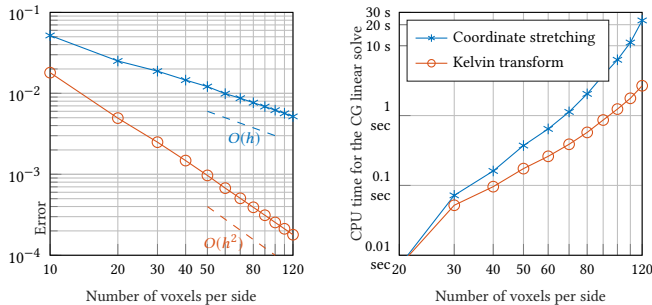


Fig. 5. A performance comparison for the example in Figure 4, between a plain coordinate stretching method and our Kelvin transform method, both discretized by the finite difference scheme. The computation was performed on a 2.5 GHz Quad-Core Intel Core i7 processor. In a coordinate stretching method, one solves the infinitely stiff system (5) for  $U(y)$ . In the Kelvin transform method, one only needs to solve a regular Laplace equation (8).

For the grid points near a Dirichlet boundary, one may apply a ghost point method [Gibou et al. 2002] while maintaining symmetry of the system matrix. For simplicity, in many graphics applications, one may also just rasterize  $\Omega_{\text{inv}}$  so that all stencils lie either in the interior or on the boundary. This grid rounding can give rise to an

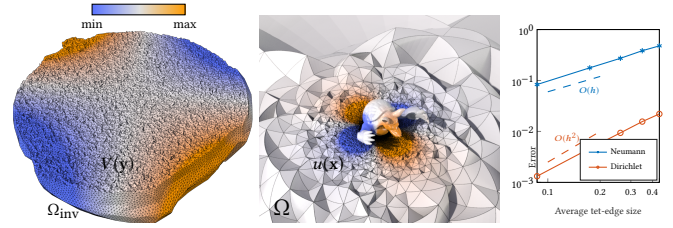


Fig. 6. An exterior domain Laplace equation solved using the Kelvin transform and the finite element method. The transformed function  $V(y)$  is solved on the tetrahedralized inverted domain (left), from which we obtain the solution  $u(x)$  on the original domain (middle). A comparison with the exact solution shows the expected convergence rate under mesh refinement for both the Dirichlet and Neumann problem with rough boundary (right).

additional  $O(h)$  error. Still, this method has a lower error compared to truncating the infinite domain which would cause  $O(1)$  errors.

The Robin boundary condition (12) amounts to adding an additional diagonal matrix corresponding to the entries occupied by the boundary surface (Appendix C).

Figure 4 shows a Dirichlet boundary value problem for the Laplace equation on the infinite domain exterior to the Stanford Bunny.<sup>2</sup> We solve for  $V(y)$  (top-right) which satisfies the Laplace equation (8) ( $F(y) = 0$ ), and we evaluate the solution  $U(y)$  (top-left) and  $u(x)$  (bottom) respectively on the inverted and original domain.

We compare our method against a more traditional approach of transforming only the coordinates, both discretized on the same grid. In the traditional method, one solves (5) for  $U(y)$  on  $\Omega_{\text{inv}}$  with a point constraint  $U(0) = 0$ . We note that solving (5) on a grid is equivalent to solving the original equation discretized on a curvilinear coordinate, which stretches to infinity on the original domain (see the visible grid in Figure 4, bottom). On the inverted domain, we see more clearly that the equation (5) is infinitely stiff near  $y = 0$ . Namely, the equation involves a general elliptic differential operator  $\nabla \cdot (\sigma(y)\nabla U(y))$  with a highly non-uniform value of  $\sigma(y)$  (in fact  $\sigma(y) \rightarrow \infty$  around  $y = 0$ ). This causes the system matrix to be ill-conditioned and leads to slower performance using an iterative linear solver (e.g. conjugate gradient solver). Figure 5 (right) shows that solving (8) in our method is much more efficient than solving the stiff system (5). Another issue our method overcomes is that the solution  $U(y)$  to (5) is singular (not differentiable) at  $y = 0$  (Figure 4, top-left), causing a lower order of convergence (Figure 5, left).

**5.1.2 Finite element method.** The transformed Poisson or Laplace equation (8) can also be solved with a Finite Element Method (FEM). The inverted domain is tessellated into a tetrahedral mesh that conforms to the boundary, and the function  $V(y)$  is approximated by a piecewise linear function. The Laplace operator is discretized into a stiffness matrix [Crane 2019]. In Figure 6, using an exact solution  $u(x) = (x_1^2 - x_3^2)/|x|^5$  (i.e.  $V(y) = (y_1^2 - y_3^2)$ ) to the Laplace equation as a reference, we set up a Dirichlet problem and a Neumann problem

<sup>2</sup>In this example,  $u(x) = \frac{1}{|x|} \cosh\left(\frac{3\sqrt{2}x_1}{|x|^2}\right) \cos\left(\frac{3x_2}{|x|^2}\right) \cos\left(\frac{3x_3}{|x|^2}\right)$  is taken as the reference solution.

(i.e. Robin problem for  $V$ ). The method converges in  $O(h^2)$  for the Dirichlet problem and in  $O(h)$  for the Neumann problem, where  $h$  is the average edge length. Note that the Neumann problem shows a drop in the convergence rate due to the non-smooth boundary.

## 5.2 Efficient Monte Carlo Method on Exterior Domains

The paradigm of using the Monte Carlo method for solving PDEs was recently introduced to geometry processing and simulations in computer graphics [Sawhney and Crane 2020]. The main advantage of the Monte Carlo method over the finite difference or finite element methods is that there is no need to discretize the domain. In the Monte Carlo method, the solution is evaluated upon query. In particular, if we are interested in visualizing the solution on a lower dimensional plane, then we only need to query the solution on the plane. That is, we compute only as much as we see. The resolution of the visualized solution can be as high as the screen to which we wish to render. Such a resolution is much higher than what a discrete 3D computation grid can afford.

Though no discretization is involved, PDE problems on an infinite domain still pose difficulties for Monte Carlo methods. Our method fundamentally removes those difficulties by replacing the infinite domain PDE with a compact domain PDE.

**5.2.1 The Walk-on-Spheres (WoS) Process.** Let us look at a specific Monte Carlo method, the Walk-on-Spheres (WoS) algorithm [Muller 1956; Sawhney and Crane 2020] for solving the Laplace equation with a Dirichlet boundary condition

$$\begin{cases} \Delta u(\mathbf{x}) = 0, & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = p(\mathbf{x}), & \mathbf{x} \in \Sigma. \end{cases} \quad (14)$$

The solution  $u(\mathbf{x}_0)$  at any  $\mathbf{x}_0 \in \Omega$  is evaluated as the expected value of  $p(\mathbf{x})$  observed by a random walk starting at  $\mathbf{x}_0$  when it first reaches the boundary. Concretely, given  $\mathbf{x}_0 \in \Omega$ , consider the WoS stochastic process  $\gamma = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{m_\gamma}) \subset \Omega$  characterized by

- (i) the step vectors  $\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_1, \dots$  are mutually independent;
- (ii) given  $\mathbf{x}_i$ , the next random point  $\mathbf{x}_{i+1}$  is uniformly distributed over the spherical surface  $\partial B_{r_i}(\mathbf{x}_i)$  of the largest ball centered at  $\mathbf{x}_i$  that is contained in  $\Omega$ , i.e.  $r_i = \text{dist}(\mathbf{x}_i, \Sigma)$ ;
- (iii)  $\mathbf{x}_{m_\gamma}$  is the first time  $\text{dist}(\mathbf{x}_{m_\gamma}, \Sigma) < \varepsilon$ , for a small tolerance  $\varepsilon > 0$ .

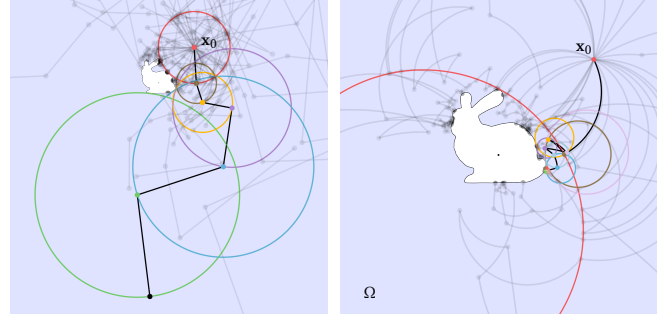
Up to the tolerance, we regard the final point  $\mathbf{x}_{m_\gamma}$  as a point on  $\Sigma$  (e.g. via a closest point projection). Let  $p(\gamma) := p(\mathbf{x}_{m_\gamma})$  denote the boundary value  $p$  evaluated at the point as the path  $\gamma$  reaches the boundary. Then, the solution to (14) at  $\mathbf{x}_0$  is given by an expectation

$$u(\mathbf{x}_0) = \mathbb{E}[p(\gamma)]. \quad (15)$$

In practice, (15) is numerically computed by a sample average.

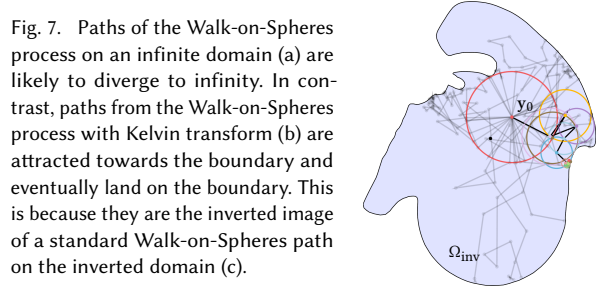
Each WoS path is efficient to sample on a bounded domain [Sawhney and Crane 2020] (empirically  $m_\gamma \approx 15$ ). However, on an infinite domain, there is a nontrivial probability that the random walk  $\gamma = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots)$  will not terminate due to Pólya's recurrence and transience theorem in 3D [Weisstein 2000; Novak 2014].

**5.2.2 Walk-on-Spheres with Russian Roulette (WoS-RR).** For the WoS method on an infinite domain, Sawhney and Crane [2020] adopted a Russian Roulette (RR) procedure [Pharr et al. 2016, Section 13.7] to



(a) WoS-RR on an infinite domain

(b) WoS-KT on the original domain



(c) WoS-KT on the inverted domain

Fig. 7. Paths of the Walk-on-Spheres process on an infinite domain (a) are likely to diverge to infinity. In contrast, paths from the Walk-on-Spheres process with Kelvin transform (b) are attracted towards the boundary and eventually land on the boundary. This is because they are the inverted image of a standard Walk-on-Spheres path on the inverted domain (c).

eliminate paths that wander too long. In the RR procedure, one fixes a termination probability  $0 < \lambda < 1$ . At every step of the random walk, with probability  $\lambda$ , the walk is terminated. With this additional decay process, the path  $\gamma$  will almost surely have a finite number of steps  $m_\gamma$ . If  $\gamma$  is terminated before reaching the boundary, one sets the evaluation  $p(\gamma) = 0$ . Under this WoS-RR process, we have

$$u(\mathbf{x}_0) = \mathbb{E} \left[ \frac{p(\gamma)}{(1-\lambda)^{m_\gamma}} \right] \quad (16)$$

which is numerically computed by a sample average.

The drawback of the WoS-RR process is that the factor  $1/(1-\lambda)^{m_\gamma}$  amplifies the variance, leading to a slower convergence. Also, the original issue that it takes more steps for a path to reach  $\Sigma$  in an open space persists. Every effective sample costs more computation time, compared to the WoS in the interior domain.

**5.2.3 Walk-on-Spheres with Kelvin Transform (WoS-KT).** The challenge in WoS on an infinite domain is resolved by the Kelvin transform. Using Algorithm 1, we only need to solve a Laplace equation in the interior domain, which is computed by the WoS process without the need for Russian Roulette.

Explicitly, for each point of query  $\mathbf{x}_0 \in \Omega$ , we compute the inverted point  $\mathbf{y}_0 = \mathbf{x}_0/|\mathbf{x}_0|^2 \in \Omega_{\text{inv}}$ . We then evaluate  $V(\mathbf{y}_0)$  which solves

$$\begin{cases} \Delta V(\mathbf{y}) = 0, & \mathbf{y} \in \Omega_{\text{inv}} \cup \{\mathbf{0}\}, \\ V(\mathbf{y}) = \frac{p(\mathbf{y})}{|\mathbf{y}|}, & \mathbf{y} \in \Sigma_{\text{inv}}, \end{cases} \quad (17)$$

by the classical WoS. Note that sampling this WoS process is as efficient as any WoS in a bounded domain. Finally, we evaluate

$$u(\mathbf{x}_0) = V(\mathbf{y}_0)|\mathbf{y}_0|. \quad (18)$$

Figure 7 shows that the WoS-KT effectively uses a random walk process different from the standard WoS on the original domain. For

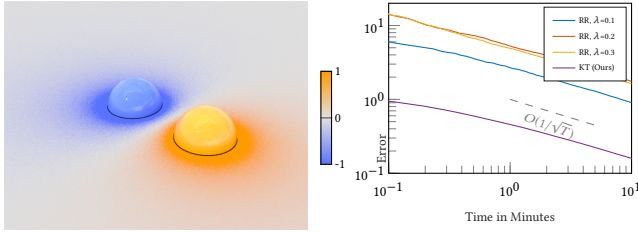


Fig. 8. Laplace equation solved with a Monte Carlo method together with Kelvin transform (left). The error plot (right) shows that the error decreases in the expected rate  $O(1/\sqrt{T})$  under an increasing number of samples measured in the total computation time  $T$ .

the classical WoS (top-left),  $\mathbf{x}_{i+1}$  is sampled on a sphere centered at  $\mathbf{x}_i$ . In WoS-KT, once again the standard WoS process is used, but now on the inverted domain (bottom). Mapping this WoS path to the original exterior domain (top-right), we see that  $\mathbf{x}_{i+1}$  is sampled (non-uniformly) on a sphere no longer centered at  $\mathbf{x}_i$ . The sphere also does not necessarily enclose  $\mathbf{x}_i$ . Effectively, the path gravitates towards the boundary.

**5.2.4 Numerical Examples and Comparisons.** In Figures 8–11, we compare our method (WoS-KT) against the WoS-RR with various choices of termination probability  $\lambda$ . The computation was performed on a 2.3 GHz 8-Core Intel Core i9 processor. In each example,  $u(\mathbf{x})$  is interpreted as an electric potential in an electrostatic setup. The electric potential satisfies either the Laplace equation (14) in the absence of electric charge in the domain (Figures 8–10), or otherwise a Poisson equation (Figure 11). Every example consistently shows that at an equal computational time, WoS-KT achieves about an order of magnitude lower error than WoS-RR.

In Figure 8 we validate the WoS-RR and WoS-KT algorithms by comparing the results against an exact solution  $u(\mathbf{x}) = 1/|\mathbf{x}| - 1/|\mathbf{x} - (2, 0, 0)|$ , the potential of two opposite point charges. The Dirichlet boundary is set on the  $\{u = 1\}$  and  $\{u = -1\}$  isosurfaces. The numerical solution to the Laplace equation evaluated on a plane (Figure 8, left) converges to the exact solution for both the WoS-RR and WoS-KT algorithms as the sample size increases (Figure 8, right).

In Figure 9, we prescribe a patterned electric potential on the dragon model as the Dirichlet boundary condition. The potential extends into a 3D harmonic function visualized on a 2D slice computed by both methods.

With a similar setup, Figure 10 is another example. One computational task in Biochemistry is to find the electric potential around a given molecule. We take the protein molecule 1CRN from the protein data bank (PDB).<sup>3</sup> We first convert the molecule data into the so-called electrostatic potential map (EPM), which is the electric potential value defined over the van der Waals surface of the molecule. This step is computed using the conventional software *Avogadro 1.2* [Hanwell et al. 2012]. We then extend the potential to the surrounding 3D space by solving the Laplace equation. The solution is evaluated on a 2D slice using the Monte Carlo method.

<sup>3</sup><http://doi.org/10.2210/pdb1CRN/pdb>

In Figure 11, we solve an electrostatic potential around a point charge located at  $\mathbf{s} \in \Omega$  with charge  $c \in \mathbb{R}$  next to a fiddler crab assigned with a prescribed potential. In this case  $u$  satisfies a Poisson equation with Dirac- $\delta$  right-hand side:

$$\begin{cases} \Delta u(\mathbf{x}) = c\delta_{\mathbf{s}}(\mathbf{x}) & \mathbf{x} \in \Omega \\ u(\mathbf{x}) = p(\mathbf{x}) & \mathbf{x} \in \Sigma, \end{cases} \quad (19)$$

which can be solved by WoS with a slight modification [Sawhney and Crane 2020, Section 4.2.2]. Note that the Kelvin transform of (19) is once again a Poisson equation with a point source on a bounded domain

$$\begin{cases} \Delta V(\mathbf{y}) = \frac{c}{|\mathbf{s}|} \delta_{\mathbf{s}/|\mathbf{s}|^2}(\mathbf{y}), & \mathbf{y} \in \Omega_{\text{inv}} \cup \{\mathbf{0}\} \\ V(\mathbf{y}) = \frac{p(\mathbf{y})}{|\mathbf{y}|}, & \mathbf{y} \in \Sigma_{\text{inv}}, \end{cases} \quad (20)$$

which we solve by WoS.

### 5.3 Physical and Geometric Systems on Infinite Domains

Here, we demonstrate three additional examples of solving Poisson and Laplace problems on infinite domains using the Kelvin transformation. These proof-of-concept examples may be integrated into larger practical applications.

**5.3.1 Magnetic Fields around a Star.** A standard problem in computational solar physics is to compute the magnetic field in the space around the sun [Priest 2014]. The magnetic field lines compose the visually outstanding fibrous texture in the solar corona, and hence such a magnetic field is demanded not only in astronomy but also in scientific [NASA Scientific Visualization Studio 2018] and cinematic [CADENS 2015; Borkiewicz et al. 2019] visualizations.

The magnetic flux data on the solar surface is called the *magnetogram* (Figure 12, left), which is available from NASA Solar Dynamics Observatory [2020]. Hence, the problem is more specifically a boundary value problem, where one extends the magnetogram to a magnetic field over the 3D domain  $\Omega$  exterior to the sun. One simple model for this task is the Potential Field Source Surface (PFSS) model [Altschuler and Newkirk 1969; Priest 2014, Sec. 3.3], where one assumes that the magnetic field is divergence-free and curl-free. Under this assumption, the magnetic vector field is the gradient of a potential satisfying the Laplace equation (2) ( $f = 0$ ). The magnetogram is the Neumann data for the boundary condition (11) of this harmonic potential.

Figure 12 (right) shows the resulting magnetic field in the entire exterior domain computed by our Kelvin transformation method. The computation involves a Laplace problem with a Robin boundary condition (12) on the interior of a sphere, solved on a  $128^3$  grid.

Note that the PFSS field computed in conventional softwares [Freeland and Bentley 2000] often relies on truncating the domain into a finite region. This causes a distortion in the field lines near the artificial boundary (Figure 13, left). In contrast, the field generated by our method smoothly extends to the entire  $\Omega$  (Figure 13, right).

**5.3.2 Pressure Solves for Incompressible Flows.** The *pressure projection* is a common routine in incompressible fluid simulations [Bridson 2015]. Taking an arbitrary vector field  $\mathbf{b}: \Omega \rightarrow \mathbb{R}^3$  as the input (Figure 14, top-left), this method returns a unique velocity



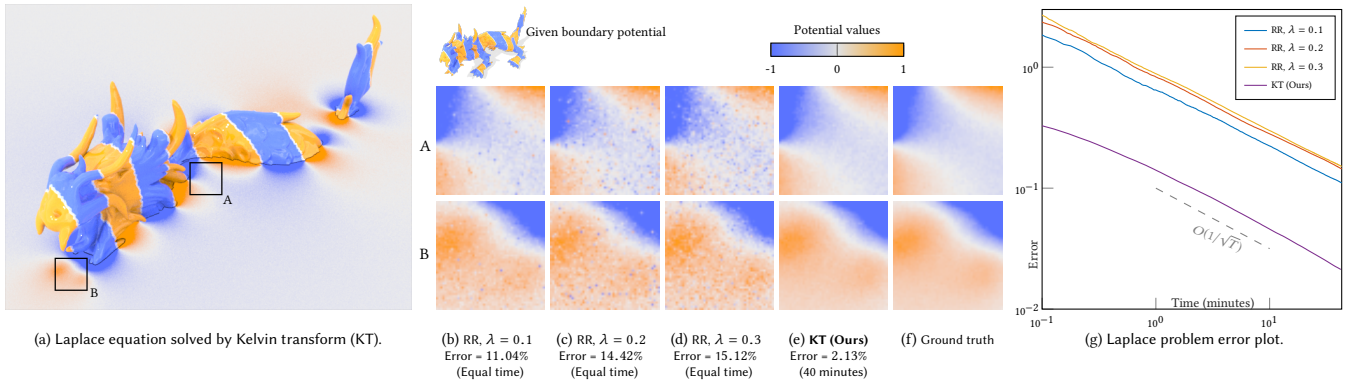


Fig. 9. A Laplace equation on the solution plane given the boundary potential data on the surface of the dragon is solved using the Monte Carlo method. First, we run the Kelvin transform experiment to achieve 2% error, which takes about 40 minutes. Next, we run Russian Roulette experiments with different termination probabilities in equal time. By extrapolation from the error plot, to achieve equal quality results from the Russian Roulette experiments, they must run for about 20 to 40 hours depending on the termination probability.

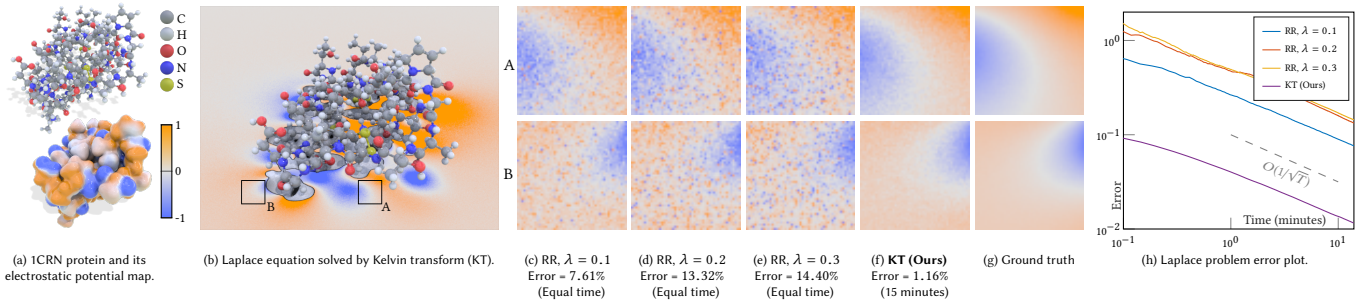


Fig. 10. The 1CRN protein molecule figure captured from protein data bank (PDB) with potential values assigned according to its electrostatic potential map (EPM). A Laplace equation on the solution plane given the boundary EPM data on the surface of the molecule is solved using the Monte Carlo method. First, we run the Kelvin transform experiment to achieve 1% error, which takes about 15 minutes. Next, we run Russian Roulette experiments with different termination probabilities in equal time. By extrapolation from the error plot, to achieve equal quality results from the Russian Roulette experiments, they must run for about 2 to 10 hours depending on the termination probability.

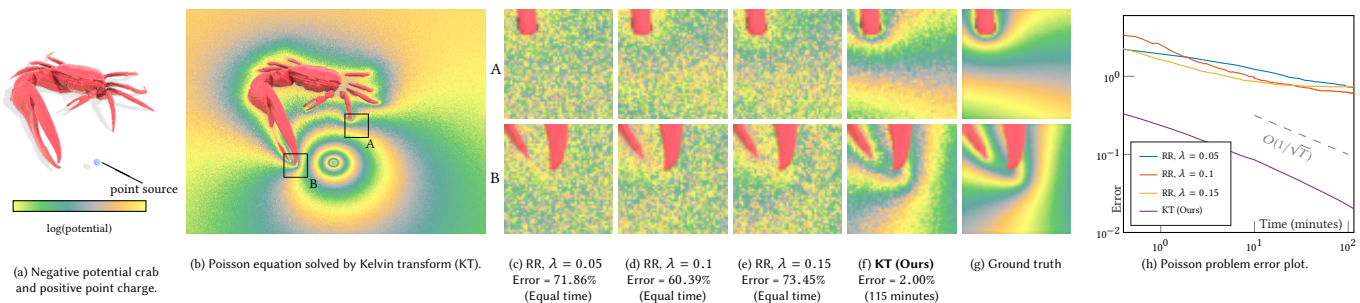


Fig. 11. A fiddler crab with negative potential equal to  $-1$  statV is in close proximity to a point charge of  $+10$  statC. Potential values are solved on the solution plane and visualized as level sets using alternating colors. Without the crab, the color rings would be concentric around the point source. With the crab, as seen here, the potential level set is forced to follow the crab's shape. First, we run the Kelvin transform experiment to achieve 2% error, which takes about two hours. Next, we run Russian Roulette experiments with different termination probabilities in equal time. For the Russian Roulette experiments, we encounter many samples where the denominators of (16) are too close to zero. Rejecting these samples can lead to biased results. This problem is not present in our method. (Model courtesy Oliver Laric.)

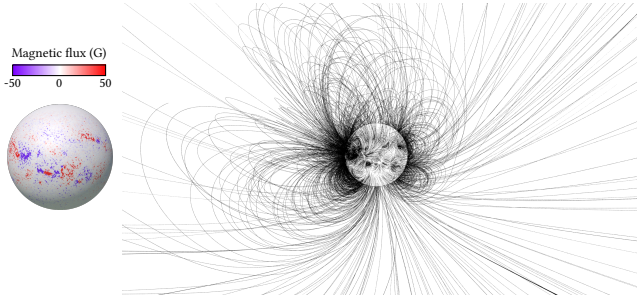


Fig. 12. Given the magnetic flux data over the surface of the Sun (left), a magnetic field is found in the unbounded exterior domain (right), computed using a Kelvin transformation. We adopt the Potential Field Source Surface (PFSS) model, where the magnetic field is the gradient of a harmonic potential.

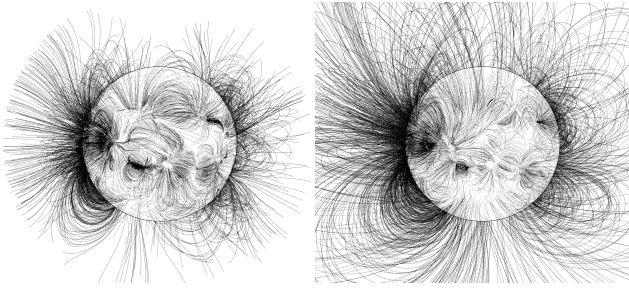


Fig. 13. A comparison between a domain truncation (left) and our Kelvin transform (right) approach to solve the PFSS magnetic extrapolation problem (cf. Figure 12). A direct numerical method often requires truncating the infinite domain into a bounded domain (left), which distorts the field lines into the normal direction of an artificial boundary. In contrast, using the Kelvin transform technique, we obtain the magnetic field smoothly extending in the infinite space without border (right).

vector field  $\mathbf{a}: \Omega \rightarrow \mathbb{R}^3$  (Figure 14, bottom-left) of the form<sup>4</sup>

$$\mathbf{a}(\mathbf{x}) = \mathbf{b}(\mathbf{x}) - \nabla u(\mathbf{x}), \quad u: \Omega \rightarrow \mathbb{R}, \quad (21)$$

that satisfies the incompressibility conditions

$$\begin{cases} \nabla \cdot \mathbf{a}(\mathbf{x}) = 0, & \mathbf{x} \in \Omega \\ \mathbf{a}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0, & \mathbf{x} \in \Sigma = \partial\Omega. \end{cases} \quad (22)$$

This amounts to solving the Poisson equation

$$\begin{cases} \Delta u(\mathbf{x}) = \nabla \cdot \mathbf{b}(\mathbf{x}), & \mathbf{x} \in \Omega \\ \mathbf{n}(\mathbf{x}) \cdot \nabla u(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}), & \mathbf{x} \in \Sigma \\ u(\mathbf{x}) \rightarrow 0 & |\mathbf{x}| \rightarrow \infty. \end{cases} \quad (23)$$

Note that the domain  $\Omega$  we consider in (23) is an infinite domain, which we solve using the Kelvin transform (Figure 2).

Figure 14 shows a pipeline for pressure projection performed on the inverted domain. Let  $\mathbf{B}(\mathbf{y}) = d\Phi^{-1}(\mathbf{b}(\mathbf{x}))$  and  $\mathbf{A}(\mathbf{y}) = d\Phi^{-1}(\mathbf{a}(\mathbf{x}))$  (top-middle and bottom-middle) respectively be the input and output vector fields represented on  $\Omega_{\text{inv}}$  via the pushforward  $d\Phi^{-1}$ . Call the

<sup>4</sup>We do not use a more conventional notation where velocities are denoted by  $\mathbf{u}$  or  $\mathbf{v}$  in order to keep these letters for the variables subject to the Poisson equation.

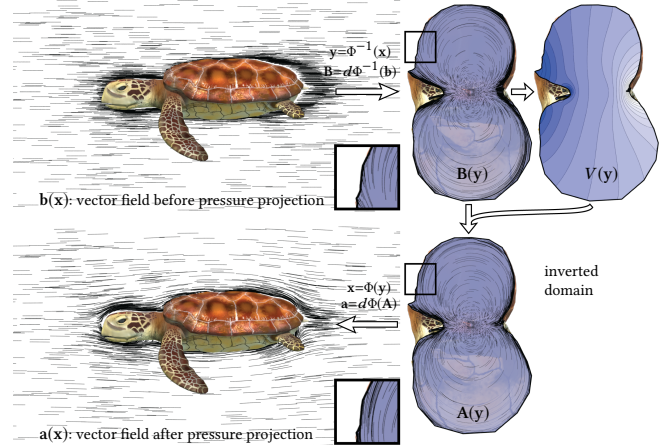


Fig. 14. Pressure projection on an infinite domain. Given an arbitrary vector field  $\mathbf{b}(\mathbf{x})$  around an obstacle such as the turtle (top-left), the process produces a vector field  $\mathbf{a}(\mathbf{x})$  with the enforced divergence free constraint and no-through boundary condition (bottom-left). First, we invert the domain as well as the vector field (top-middle). The inset shows that the inverted vector field  $\mathbf{B}$  does not satisfy the no-through boundary condition. Next, according to the Kelvin transform, we solve the Poisson equation (25) in the bounded inverted domain (top-right). Taking the solution  $V(\mathbf{y})$  and using the formula (28), we assemble the vector field  $\mathbf{A}(\mathbf{y})$  (bottom-middle), which is now tangent to the boundary (bottom inset). Finally, by another inversion the vector field  $\mathbf{a}(\mathbf{x})$  is brought back to the original exterior domain (bottom-left).

Poisson equation right-hand side  $f(\mathbf{x}) = \nabla \cdot \mathbf{b}(\mathbf{x})$ , whose inverted counterpart  $F(\mathbf{y}) = f(\Phi(\mathbf{y}))$  is given by (Appendix A.2)

$$F(\mathbf{y}) = |\mathbf{y}|^6 \nabla \cdot \left( \frac{1}{|\mathbf{y}|^6} \mathbf{B}(\mathbf{y}) \right). \quad (24)$$

Denote the Neumann data by  $q(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x})$ ,  $Q(\mathbf{y}) = q(\Phi(\mathbf{y}))$ . Now, applying Algorithm 1 with (12), we solve the Poisson equation with a Robin boundary condition for  $V(\mathbf{y})$  (Figure 14, top-right)

$$\begin{cases} \Delta V(\mathbf{y}) = \frac{1}{|\mathbf{y}|^5} F(\mathbf{y}), & \mathbf{y} \in \Omega_{\text{inv}} \cup \{\mathbf{0}\} \\ \mathbf{N}(\mathbf{y}) \cdot \nabla V(\mathbf{y}) + \frac{\mathbf{N}(\mathbf{y}) \cdot \mathbf{y}}{|\mathbf{y}|^2} V(\mathbf{y}) = \frac{Q(\mathbf{y})}{|\mathbf{y}|^3}, & \mathbf{y} \in \Sigma_{\text{inv}}. \end{cases} \quad (25)$$

The solution is expected to be smooth across  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$  and it can be approximated using a finite difference method on a moderately coarse grid. Finally, the pressure-projected velocity field is found by

$$\mathbf{A}(\mathbf{y}) = \mathbf{B}(\mathbf{y}) - |\mathbf{y}|^4 \nabla U(\mathbf{y}). \quad (26)$$

See Appendix A.1 for the gradient represented in the inverted coordinates. However, numerically computing  $\nabla U(\mathbf{y})$  on a discrete grid is not ideal since  $U(\mathbf{y})$  is not smooth at  $\mathbf{y} = \mathbf{0}$ . Alternatively, one may express  $\nabla U$  in terms of  $\nabla V$ :

$$\begin{aligned} \nabla U(\mathbf{y}) &= \nabla (|\mathbf{y}|V(\mathbf{y})) \\ &= \frac{\mathbf{y}}{|\mathbf{y}|} V(\mathbf{y}) + |\mathbf{y}| \nabla V(\mathbf{y}), \end{aligned} \quad (27)$$

which turns (26) into

$$\mathbf{A}(\mathbf{y}) = \mathbf{B}(\mathbf{y}) - V(\mathbf{y})|\mathbf{y}|^3 \mathbf{y} + |\mathbf{y}|^5 \nabla V(\mathbf{y}). \quad (28)$$

Finally, we obtain  $\mathbf{a}(\mathbf{x})$  (Figure 14, bottom-left) by  $\mathbf{a} = d\Phi(\mathbf{A})$ .

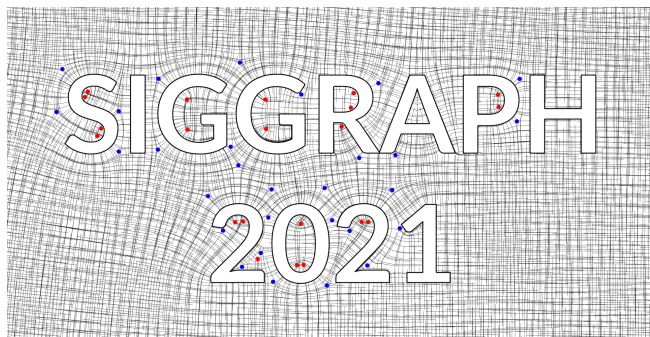


Fig. 15. Cross field visualized on the exterior of the letters using Monte Carlo together with Kelvin transform. Blue and red points represent negative and positive singularities, respectively.

---

**Algorithm 2** Pressure solve on an infinite domain  $\Omega$ 


---

**Input:** A vector field  $\mathbf{b}$  on  $\Omega$ , represented by  $\mathbf{B} = d\Phi^{-1}(\mathbf{b})$  on  $\Omega_{\text{inv}}$ .

- 1: Define  $F(\mathbf{y})$  and  $Q(\mathbf{y})$  by (24).
- 2: Solve the Robin boundary value problem (25) for  $V$  on  $\Omega_{\text{inv}}$ .
- 3: Compute  $\mathbf{A}$  by (28).

**Output:** The divergence-free vector field  $\mathbf{a} = d\Phi(\mathbf{A})$  on  $\Omega$ .

---

**5.3.3 Direction Field Designs.** Designing a smooth direction field on a general domain has led to many applications in geometry processing including quadrilateral meshing [Vaxman et al. 2016; Viertel and Osting 2019], hexagonal meshing [Solomon et al. 2017] and graphics vectorization [Bessmeltsev and Solomon 2019], just to name a few. In Figure 15, we demonstrate a smooth cross field on an infinite domain  $\Omega$  in the plane exterior to some letters. The cross field is constrained to be tangent and normal to the boundary, and is otherwise as smooth as possible. This process gives rise to emergent singularities of the cross field, which sit at geometrically interesting locations that sparsely summarize the concavity and convexity of the nearby boundary.

Obtaining Figure 15 amounts to solving a Laplace equation. A cross consists of four orthogonal directions  $\psi, i\psi, -\psi, -i\psi$  where  $\psi = e^{i\theta} \in \mathbb{C}$  describes a direction in terms of its complex phase. This quadruple of directions is uniquely described by a single complex number  $u = \psi^4$ . Under this representation, a cross field is a complex-valued function  $u: \Omega \rightarrow \mathbb{C}$ . To find a smooth cross field that conforms to the boundary, we solve the Laplace equation with a Dirichlet boundary condition for  $u$ . The infinite domain is handled by the Kelvin transformation. The Kelvin transformed Laplace equation is numerically solved using the Monte Carlo method.

Note that the Kelvin transform we have discussed so far (Theorem 1) applies to the 3D Poisson and Laplace equations. However, the domain in Figure 15 is two dimensional, which takes a different substitution rule from (7). In fact, in 2D, we have a simpler Kelvin transform:  $\Delta u(\mathbf{x}) = 0$  if and only if  $\Delta U(\mathbf{y}) = 0$ , where  $U(\mathbf{y}) = u(\mathbf{y}/|\mathbf{y}|^2)$ . This is known as the conformal invariance of the Laplace operator on 2D. In Figure 15, we solve for a harmonic complex-valued function  $U(\mathbf{y})$  on the bounded domain  $\Omega_{\text{inv}} \cup \{0\}$ , and, after inversion, visualize the cross field on the original infinite domain.

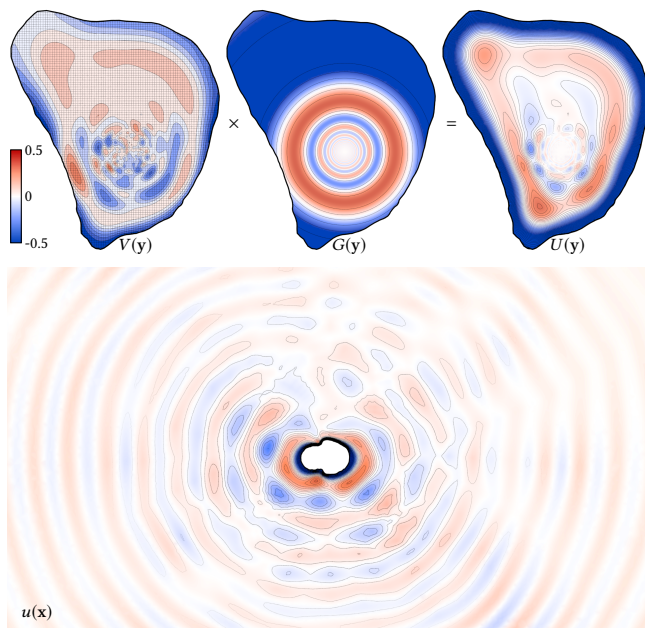


Fig. 16. The Helmholtz equation solved on an infinite domain using our method. The top-row shows the inverted domain. For this problem, once the domain is inverted, the solution  $U(\mathbf{y})$  exhibits an infinite oscillation at the origin. We factor out a known oscillatory portion  $G(\mathbf{y})$ , and focus on solving for the quotient  $V(\mathbf{y})$  governed by (35) (top-left). Next, we evaluate  $U(\mathbf{y})$  (top-right) by multiplying  $G(\mathbf{y})$  (top-middle) and  $V(\mathbf{y})$  (top-left) together. Finally, by inverting the domain back to the exterior, the solution to the Helmholtz equation is found over the entire infinite domain (bottom).

---

**Algorithm 3** Direction field extrapolation on unbounded domain
 

---

**Input:** Normal vectors evaluated on the boundary as  $\psi = e^{i\theta}$ .

- 1: Construct  $u = \psi^4$ .
- 2: Solve the interior problem  $\Delta U(\mathbf{y}) = 0$  where  $U(\mathbf{y}) = u(\mathbf{y}/|\mathbf{y}|^2)$ .

**Output:**  $u(\mathbf{x}) = U(\frac{\mathbf{x}}{|\mathbf{x}|^2})$ .

---

## 6 HELMHOLTZ EQUATION ON EXTERIOR DOMAINS

So far, we have been solving the Poisson or the Laplace equation. In this section, we consider an equation whose solution is oscillatory all the way to infinity as a challenging case for a coordinate mapping approach. Take the Helmholtz equation

$$\Delta u(\mathbf{x}) + \omega^2 u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega \quad (29)$$

on the exterior domain  $\Omega$ . This equation is the fundamental equation in the physically-based acoustic simulation in computer graphics [James 2016]. The equation governs the acoustic vibration mode in an environment  $\Omega$  at a constant frequency  $\omega \in \mathbb{R}_{>0}$ . The function  $u: \Omega \rightarrow \mathbb{C}$  encodes the amplitude and phase in its complex values. A Dirichlet boundary condition

$$u(\mathbf{x}) = p(\mathbf{x}), \quad \mathbf{x} \in \Sigma = \partial\Omega \quad (30)$$

describes wave sources with amplitudes and phases, as well as obstacles for which  $p = 0$ . For the exterior domain problem, it is also

necessary to impose the *Sommerfeld radiating condition* at infinity:

$$\frac{\mathbf{x}}{|\mathbf{x}|} \cdot \nabla u(\mathbf{x}) - \mathbb{i}\omega u(\mathbf{x}) = o\left(\frac{1}{|\mathbf{x}|}\right) \quad \text{as } |\mathbf{x}| \rightarrow \infty. \quad (31)$$

The Sommerfeld radiating condition describes the asymptotic behavior of the solution as an outward traveling circular wave. This condition eliminates background plane waves or inward traveling circular waves, establishing the uniqueness of the PDE problem.

To capture the oscillatory behavior of  $u(\mathbf{x})$  near infinity, we take

$$g(\mathbf{x}) = \frac{1}{|\mathbf{x}|} e^{\mathbb{i}\omega|\mathbf{x}|}, \quad G(\mathbf{y}) = |\mathbf{y}| e^{\mathbb{i}\frac{\omega}{|\mathbf{y}|}}, \quad (32)$$

for our preconditioning strategy (6). In fact, (32) is the radially symmetric solution to (29) that satisfies (31). This gives a generalized Kelvin transform adaptive to the Helmholtz equation:

$$v(\mathbf{x}) = |\mathbf{x}| e^{-\mathbb{i}\omega|\mathbf{x}|} u(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (33)$$

or on the inverted domain,

$$V(\mathbf{y}) = \frac{1}{|\mathbf{y}|} e^{-\mathbb{i}\omega\frac{1}{|\mathbf{y}|}} U(\mathbf{y}), \quad \mathbf{y} \in \Omega_{\text{inv}}. \quad (34)$$

**Theorem 2.** *Under the change of variable (34),  $u: \Omega \rightarrow \mathbb{C}$  satisfies the Helmholtz equation (29) with boundary conditions (30) and (31) if and only if  $V: \Omega_{\text{inv}} \cup \{\mathbf{0}\} \rightarrow \mathbb{C}$  satisfies*

$$\Delta V(\mathbf{y}) - 2\mathbb{i}\omega \frac{\mathbf{y}}{|\mathbf{y}|^3} \cdot \nabla V(\mathbf{y}) = 0, \quad \mathbf{y} \in \Omega_{\text{inv}} \cup \{\mathbf{0}\}. \quad (35)$$

with boundary condition

$$V(\mathbf{y}) = \frac{1}{|\mathbf{y}|} e^{-\mathbb{i}\omega\frac{1}{|\mathbf{y}|}} P(\mathbf{y}), \quad \mathbf{y} \in \Sigma_{\text{inv}}, \quad (36)$$

where  $P(\mathbf{y}) = p(\mathbf{y}/|\mathbf{y}|^2)$ .

PROOF. See Appendix D for the derivation.  $\square$

---

#### Algorithm 4 Helmholtz equation solver on an exterior domain

---

**Input:** Helmholtz problem (29), (31) on an infinite domain, given boundary condition.

- 1:  $V \leftarrow$  Solve the interior problem (35) with appropriate boundary conditions (36).
- 2:  $U(\mathbf{y}) = G(\mathbf{x})V(\mathbf{y})$ , where  $G(\mathbf{x})$  is given by (32).
- 3:  $u(\mathbf{x}) = U(\frac{\mathbf{x}}{|\mathbf{x}|^2})$ .

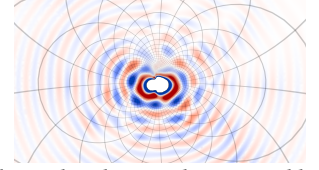
**Output:**  $u(\mathbf{x})$ .

---

One thing to note is that the second term in (35) is a directional derivative in a radial vector with an unbounded magnitude near the origin. This is consistent with what we expect for the Helmholtz equation. To see this, let us ignore for a moment the Sommerfeld radiating condition. Then, the equation admits many oscillating solutions, such as the plane waves, that do not radiate outward. After transforming those waves to  $V(\mathbf{y})$  via (34), we obtain infinite oscillation frequency at the origin. The unboundedness of the second coefficient in (35) balances with the higher order derivative from the first term. Among all possible solutions, the Sommerfeld-condition-satisfying solution is the only one that is not infinitely oscillatory at the origin. Therefore, by including the origin as part of the domain in (35) we have effectively imposed the Sommerfeld radiating condition.

Figure 1 (right) shows a solution  $u(\mathbf{x})$  to the Helmholtz equation on the exterior of the Stanford Bunny. The boundary condition is set as  $u(\mathbf{x}) = -1, \mathbf{x} \in \Sigma$ . We visualize only the imaginary part of the solution on a 2D slice as a height field. We use the finite difference method to discretize (35) on a  $128^3$  grid, whose result is shown on Figure 16 (top-left). By multiplying  $G(\mathbf{y})$  given by (32) (top-middle) and our numerical solution  $V(\mathbf{y})$ , we obtain the solution  $U(\mathbf{y})$  on the inverted domain (top-right). Finally, we visualize the solution  $u(\mathbf{x}) = U(\mathbf{x}/|\mathbf{x}|^2)$  on the original infinite domain (bottom).

Note that, on the original domain, even though the computation grid is effectively much coarser further away from the Stanford Bunny, the high frequencies in the solution  $u(\mathbf{x})$  persist nevertheless. See the inset where the grid is shown. This is possible since the subgrid frequencies are carried by  $g(\mathbf{x}) = G(\mathbf{y})$ .



## 7 CONCLUSION

We introduce the method of Kelvin transformation which turns a PDE problem on an infinite domain to another PDE problem on a bounded domain. Specifically, the Kelvin transformation turns every Poisson problem on an exterior domain to another Poisson problem on an interior domain. In other words, using the Kelvin transform, every compact domain Poisson solver is readily an infinite domain Poisson solver. The integration of the Kelvin transformation with a grid-based scheme and a Monte Carlo method are demonstrated. They are shown to perform much more efficiently than the previous attempts without incorporating the Kelvin transform. The transformation technique also applies beyond the Poisson problems. We demonstrate solving the Helmholtz equation on an infinite domain on a grid using the generalized Kelvin transformation. This has previously been regarded intractable without the aid of artificial wave-absorbing boundary conditions. These equations are very common in computer graphics and scientific computing.

Though the Kelvin transform technique is general, we have only demonstrated it for the aforementioned equations. The change of variable requires analyzing the asymptotic behavior at the singularity for the inverted solution, which is usually different for different problems. We do not have a general theory that guarantees the well-posedness of the equation after the suitable Kelvin transformation. These are interesting mathematical questions to study.

Our Kelvin transform method relies on factoring out a scalar function  $G(\mathbf{y})$  which captures the behavior of the solution near the origin of the inverted domain. It is therefore not obvious how to generalize the Kelvin transform to PDEs of vector-valued functions or tensor valued-functions such as in continuum mechanics. The possible extensions to the theory are yet to be explored.

Nevertheless, the Kelvin transformation in the demonstrated cases shows a wide range of applications. In many cases, there is no longer a clear distinction between boundedness and unboundedness of the problem. These notions have permeated the way we phrase a PDE problem in numerical simulations as well as in their analysis. We expect many new and exciting discoveries along the line of the Kelvin transformations in the future.

## ACKNOWLEDGMENTS

The authors would like to thank Marcel Padilla for the magnetogram data processing. This work was funded in part by the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing. Additional support was provided by SideFX software.

## REFERENCES

- Martin D Altschuler and Gordon Newkirk. 1969. Magnetic fields and the structure of the solar corona. *Solar Physics* 9, 1 (1969), 131–149.
- Gavin Barill, Neil G Dickson, Ryan Schmidt, David IW Levin, and Alec Jacobson. 2018. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 43:1–43:12.
- Carl M Bender and Steven A Orszag. 2013. *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*. Springer Science & Business Media.
- Jean-Pierre Berenger. 1994. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics* 114, 2 (1994), 185–200.
- Mikhail Bessmeltsev and Justin Solomon. 2019. Vectorization of line drawings via polyvector fields. *ACM Transactions on Graphics (TOG)* 38, 1 (2019), 9:1–9:12.
- Morten Bojsen-Hansen and Chris Wojtan. 2016. Generalized non-reflecting boundaries for fluid re-simulation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 96:1–96:7.
- Kalina Borkiewicz, AJ Christensen, Drew Berry, Christopher Fluke, Greg Shirah, and Kel Elkins. 2019. Cinematic Scientific Visualization: The Art of Communicating Science. In *SIGGRAPH Asia 2019 Courses*. ACM Press, Article 107.
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC press.
- Tyson Brochu, Todd Keeler, and Robert Bridson. 2012. Linear-time smoke animation with vortex sheet meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Citeseer, 87–95.
- CADENS. 2015. Computational Science and Data Visualization take the spotlight in new Solar Superstorms documentary. [https://www.ncsa.illinois.edu/news/story/solar\\_superstorms](https://www.ncsa.illinois.edu/news/story/solar_superstorms)
- Albert Chern. 2019. A Reflectionless discrete perfectly matched layer. *J. Comput. Phys.* 381 (2019), 91–109.
- Xavier Claeys, Ralf Hiptmair, and Elke Spindler. 2017. Second kind boundary integral equation for multi-subdomain diffusion problems. *Advances in Computational Mathematics* 43, 5 (2017), 1075–1101.
- Keenan Crane. 2019. The n-dimensional cotangent formula. *Online note* (2019). <https://www.cs.cmu.edu/~kmc Crane/Projects/Other/nDCotanFormula.pdf>
- Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 149:1–149:9.
- Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 78:1–78:12.
- Björn Engquist and Andrew Majda. 1977. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences* 74, 5 (1977), 1765–1766.
- Sergey Ermakov, Vladimir Viktorovich Nekrutkin, and Aleksandr Stepanovich Sipin. 1989. *Random processes for classical equations of mathematical physics*. Vol. 34. Springer Science & Business Media.
- Robert Finn and David Gilbarg. 1957. Three-dimensional subsonic flows, and asymptotic estimates for elliptic partial differential equations. *Acta Mathematica* 98 (1957), 265–296.
- Sam Freeland and Robert Bentley. 2000. SolarSoft. *Encyclopedia of Astronomy and Astrophysics* (2000), 3390.
- Frederic Gibou, Ronald P Fedkiw, Li-Tien Cheng, and Myungjoo Kang. 2002. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.* 176, 1 (2002), 205–227.
- Dan Givoli. 1992. *Numerical methods for problems in infinite domains*. Elsevier.
- 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.
- Abhinav Golas, Rahul Narain, Jason Sewall, Pavel Krajcevski, Pradeep Dubey, and Ming Lin. 2012. Large-scale fluid simulation using velocity-vorticity domain decomposition. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 148:1–148:9.
- Gerald Gold. 2020. On the Kelvin Transformation in Finite Difference Implementations. *Electronics* 9, 3 (2020), 442.
- Leslie Greengard and Vladimir Rokhlin. 1987. A fast algorithm for particle simulations. *Journal of computational physics* 73, 2 (1987), 325–348.
- Chester E Grosch and Steven A Orszag. 1977. Numerical solution of problems in unbounded regions: coordinate transforms. *J. Comput. Phys.* 25, 3 (1977), 273–295.
- Marcus D Hanwell, Donald E Curtis, David C Lonie, Tim Vandermeersch, Eva Zurek, and Geoffrey R Hutchison. 2012. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of cheminformatics* 4, 1 (2012), 17:1–17:17.
- Isaac Harari and Thomas JR Hughes. 1992. A cost comparison of boundary element and finite element methods for problems of time-harmonic acoustics. *Computer Methods in Applied Mechanics and Engineering* 97, 1 (1992), 77–102.
- Doug L James. 2016. Physically based sound for computer animation and virtual environments. In *ACM SIGGRAPH 2016 Courses*. 22:1–22:8.
- Doug L James, Jernej Barbič, and Dinesh K Pai. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 987–995.
- Oliver Dimon Kellogg. 1953. *Foundations of Potential Theory*. Vol. 31. Courier Corporation.
- Stephen Kirkup. 2019. The boundary element method in acoustics: A survey. *Applied Sciences* 9, 8 (2019), 1642.
- Prem K Kythe. 2020. *An introduction to boundary element methods*. CRC press.
- William McLean. 2000. *Strongly elliptic systems and boundary integral equations*. Cambridge university press.
- Norman Meyers. 1963. An expansion about infinity for solutions of linear elliptic equations. *Journal of Mathematics and Mechanics* 12, 2 (1963), 247–264.
- Mervin E Muller. 1956. Some continuous Monte Carlo methods for the Dirichlet problem. *The Annals of Mathematical Statistics* 27, 3 (1956), 569–589.
- NASA Scientific Visualization Studio. 2018. The Dynamic Solar Magnetic Field with Introduction. <https://svs.gsfc.nasa.gov/4623>
- NASA Solar Dynamics Observatory. 2020. NASA AIA/HMI Data. <https://sdo.gsfc.nasa.gov/data/aiahmi/>
- Michael B Nielsen and Robert Bridson. 2011. Guide shapes for high resolution naturalistic liquid simulation. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 83:1–83:12.
- Jonathan Novak. 2014. Pólya’s random walk theorem. *The American Mathematical Monthly* 121, 8 (2014), 711–716.
- Paul William Partridge, Carlos Alberto Brebbia, et al. 2012. *Dual reciprocity boundary element method*. Springer Science & Business Media.
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.
- Eric Priest. 2014. *Magnetohydrodynamics of the Sun*. Cambridge University Press.
- M Reali, G Dassie, and V Pennati. 1984. Solution of the two-dimensional exterior Laplace problem by a general finite difference method via inversion transformation. In *Numerical methods for transient and coupled problems: proceedings of an International Conference held in Venice, Italy on July 9th-13th, 1984*. Pine Ridge Press, 447.
- Syuhel Sato, Yoshinori Dobashi, and Tomoyuki Nishita. 2018. Editing fluid animation using flow interpolation. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 173:1–173:12.
- Stefan A Sauter and Christoph Schwab. 2010. Boundary element methods. In *Boundary Element Methods*. Springer, 183–287.
- Rohan Sawhney and Keenan Crane. 2020. Monte Carlo geometry processing: a grid-free approach to PDE-based methods on volumetric domains. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 123:1–123:18.
- Camille Schreck, Christian Hafner, and Chris Wojtan. 2019. Fundamental solutions for water wave animation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 130:1–130:14.
- Andreas Söderström, Matts Karlsson, and Ken Museth. 2010. A PML-based nonreflective boundary for free surface fluid animation. *ACM Transactions on Graphics (TOG)* 29, 5 (2010), 136:1–136:17.
- Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary element octahedral fields in volumes. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 114b:1–114b:16.
- Elke Spindler. 2016. *Second kind single-trace boundary integral formulations for scattering at composite objects*. Ph.D. Dissertation. ETH Zurich.
- Oded Stein, Eitan Grinspun, Max Wardetzky, and Alec Jacobson. 2018. Natural boundary conditions for smoothing in geometry processing. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 23:1–23:13.
- Alexey Stomakhin and Andrew Selle. 2017. Fluxed animated boundary method. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 68:1–68:8.
- William Thomson. 1845. Extrait d’une Lettre de M. William Thomson à M. Liouville: Electric Images. *Journal de mathématiques pures et appliquées* (1845), 364–367.
- William Thomson. 1872. Chapter XIV - Electric Images - Extrait d’une Lettre de M. William Thomson à M. Liouville. In *Reprint of papers on electrostatics and magnetism*. Macmillan & Company, 144 – 177.
- Nils Thürey, Ulrich Rüdte, and Marc Stamminger. 2006. Animation of open water phenomena with coupled shallow water and free surface simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 157–164.
- Semyon V Tsynkov. 1998. Numerical solution of problems on unbounded domains. A review. *Applied Numerical Mathematics* 27, 4 (1998), 465–532.
- 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.
- Ryan Viertel and Braxton Osting. 2019. An approach to quad meshing based on harmonic cross-valued maps and the Ginzburg–Landau theory. *SIAM Journal on Scientific Computing* 41, 1 (2019), A452–A479.

- Jui-Hsien Wang, Ante Qu, Timothy R Langlois, and Doug L James. 2018. Toward wave-based sound synthesis for computer animation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 109:1–109:16.
- Steffen Weißmann and Ulrich Pinkall. 2010. Filament-based smoke with vortex shedding and variational reconnection. *ACM Transactions on Graphics (TOG)* 29 (2010).
- Eric W Weisstein. 2000. Pólya's random walk constants. *From MathWorld-A Wolfram Web Resource* (2000). <https://mathworld.wolfram.com/PolyasRandomWalkConstants.html>
- Xinxin Zhang and Robert Bridson. 2014. A PPPM fast summation method for fluids and beyond. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 206:1–206:11.
- Xinxin Zhang, Robert Bridson, and Chen Greif. 2015. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 52:1–52:8.

## A VECTOR CALCULUS UNDER THE INVERSION

This appendix provides the derivation details for Section 3. Specifically, we give the change of coordinate formulas for the differential operators under the inversion map

$$\Phi: \Omega_{\text{inv}} \subset \mathbb{R}^3 \rightarrow \Omega \subset \mathbb{R}^3, \quad \Phi(\mathbf{y}) = \frac{\mathbf{y}}{|\mathbf{y}|^2}, \quad \Phi^{-1}(\mathbf{x}) = \frac{\mathbf{x}}{|\mathbf{x}|^2}. \quad (37)$$

Its deformation gradient  $d\Phi: T_{\mathbf{y}}\Omega_{\text{inv}} \cong \mathbb{R}^3 \rightarrow T_{\mathbf{x}}\Omega \cong \mathbb{R}^3$  is derived by taking a variation of (37):<sup>5</sup>

$$d\Phi(\dot{\mathbf{y}}) = \frac{\dot{\mathbf{y}}}{|\mathbf{y}|^2} - 2 \frac{\langle \mathbf{y}, \dot{\mathbf{y}} \rangle}{|\mathbf{y}|^4} \mathbf{y} = \frac{1}{|\mathbf{y}|^2} \left( I - 2 \frac{\mathbf{y}\mathbf{y}^T}{|\mathbf{y}|^2} \right) \dot{\mathbf{y}}, \quad \dot{\mathbf{y}} \in T_{\mathbf{y}}\Omega_{\text{inv}}. \quad (38)$$

Note that  $d\Phi$  is a scaled reflection. In particular  $\Phi$  is a *conformal map*. That is, the deformation gradient  $d\Phi$  scales all vectors  $\dot{\mathbf{y}}, \dot{\mathbf{y}} \in T_{\mathbf{y}}\Omega_{\text{inv}}$  based at  $\mathbf{y}$  isotropically

$$|d\Phi(\dot{\mathbf{y}})| = \sigma(\mathbf{y})|\dot{\mathbf{y}}|, \quad \langle d\Phi(\dot{\mathbf{y}}), d\Phi(\dot{\mathbf{y}}) \rangle = \sigma(\mathbf{y})^2 \langle \dot{\mathbf{y}}, \dot{\mathbf{y}} \rangle \quad (39)$$

with the scaling factor  $\sigma: \Omega_{\text{inv}} \rightarrow \mathbb{R}_{>0}$  given by

$$\sigma(\mathbf{y}) = \frac{1}{|\mathbf{y}|^2}. \quad (40)$$

With the conformality property,  $\Omega_{\text{inv}}$  can be thought of as a Riemannian manifold equipped with another metric  $\langle \cdot, \cdot \rangle_{\sigma}$  induced by  $\Phi$ , which is a conformally changed metric from the Euclidean one:

$$\langle \dot{\mathbf{y}}, \dot{\mathbf{y}} \rangle_{\sigma} = \sigma(\mathbf{y})^2 \langle \dot{\mathbf{y}}, \dot{\mathbf{y}} \rangle, \quad \dot{\mathbf{y}}, \dot{\mathbf{y}} \in T_{\mathbf{y}}\Omega_{\text{inv}}. \quad (41)$$

Writing the exterior domain PDEs in the  $\Omega_{\text{inv}}$  coordinate amounts to expressing the differential operators, such as the gradient and divergence, with respect to the induced metric  $\langle \cdot, \cdot \rangle_{\sigma}$ .

### A.1 Gradient

The gradient  $\text{grad}^{\sigma} U$  of a function  $U: \Omega_{\text{inv}} \rightarrow \mathbb{R}$  is defined to agree with the directional derivative  $\langle \text{grad}^{\sigma} U, \dot{\mathbf{y}} \rangle_{\sigma} = dU(\dot{\mathbf{y}}) = \langle \nabla U, \dot{\mathbf{y}} \rangle$  for all vectors  $\dot{\mathbf{y}}$ . Here  $\nabla$  applied on  $U$  is the gradient with respect to the Euclidean metric  $\langle \cdot, \cdot \rangle$  (see Table 1). Therefore,

$$(\text{grad}^{\sigma} U)(\mathbf{y}) = \frac{1}{\sigma(\mathbf{y})^2} \nabla U(\mathbf{y}) = |\mathbf{y}|^4 \nabla U(\mathbf{y}). \quad (42)$$

Note that the pushforward of the gradient  $\text{grad}^{\sigma} U$  in the  $\sigma$ -metric agrees with the Euclidean gradient on the exterior domain:

$$(d\Phi(\text{grad}^{\sigma} U))(\mathbf{x}) = \nabla u(\mathbf{x}), \quad u(\mathbf{x}) = U(\Phi^{-1}(\mathbf{x})). \quad (43)$$

<sup>5</sup>We use the variation  $\dot{\mathbf{y}}$  as the notation for a general tangent vector based at  $\mathbf{y}$ . We will also use the symbol  $\dot{\mathbf{y}}$  if a second tangent vector is involved.

**A.1.1 Slope.** The slope of a function  $U(\mathbf{y})$  along a certain direction  $\mathbf{N} \in T_{\mathbf{y}}\Omega_{\text{inv}}$  is the direction derivative  $\langle \mathbf{N}^{\sigma}, \text{grad}^{\sigma} U \rangle_{\sigma}$  along the normalized vector  $\mathbf{N}^{\sigma} = \mathbf{N}/|\mathbf{N}|_{\sigma}$ . If  $\mathbf{N}$  is a Euclidean unit vector  $|\mathbf{N}| = 1$ , we have  $\mathbf{N}^{\sigma} = \sigma^{-1}\mathbf{N}$ . By (41) and (42), the slope is given by

$$\langle \mathbf{N}^{\sigma}, \text{grad}^{\sigma} U \rangle_{\sigma}(\mathbf{y}) = \frac{1}{\sigma(\mathbf{y})} \langle \mathbf{N}, \nabla U(\mathbf{y}) \rangle = |\mathbf{y}|^2 \langle \mathbf{N}, \nabla U(\mathbf{y}) \rangle. \quad (44)$$

Note that this slope measured in the  $\sigma$ -metric agrees with the Euclidean slope on the exterior domain

$$\langle \mathbf{N}^{\sigma}, \text{grad}^{\sigma} U \rangle_{\sigma}(\mathbf{y}) = \langle \mathbf{n}, \nabla u \rangle(\Phi(\mathbf{y})) \quad (45)$$

where  $\mathbf{n} = d\Phi(\mathbf{N}^{\sigma})$ , which satisfies  $|\mathbf{n}| = 1$ .

### A.2 Divergence

The divergence  $\text{div}^{\sigma} \mathbf{W}$  of a vector field  $\mathbf{W}$  is defined to satisfy the divergence theorem

$$\iiint_D (\text{div}^{\sigma} \mathbf{W}) d\mu^{\sigma} = \iint_{\partial D} \langle \mathbf{W}, \mathbf{N}^{\sigma} \rangle_{\sigma} dS^{\sigma} \quad (46)$$

for any volumetric region  $D \subset \Omega_{\text{inv}}$ . Here  $d\mu^{\sigma} = \sigma^3 d\mu$  is the volume element,  $dS^{\sigma} = \sigma^2 dS$  is the area element, and  $\mathbf{N}^{\sigma} = \sigma^{-1}\mathbf{N}$  is the normalized normal vector in the  $\sigma$ -metric. Comparing (46), that is  $\iiint_D (\text{div}^{\sigma} \mathbf{W}) \sigma^3 d\mu = \iint_{\partial D} \sigma^2 \langle \mathbf{W}, \sigma^{-1}\mathbf{N} \rangle \sigma^2 dS$ , with the Euclidean divergence theorem  $\iiint_D (\nabla \cdot \mathbf{W}) d\mu = \iint_{\partial D} \langle \mathbf{W}, \mathbf{N} \rangle dS$ , one arrives at

$$(\text{div}^{\sigma} \mathbf{W})(\mathbf{y}) = \frac{1}{\sigma(\mathbf{y})^3} \nabla \cdot (\sigma(\mathbf{y})^3 \mathbf{W}(\mathbf{y})) = |\mathbf{y}|^6 \nabla \cdot \left( \frac{1}{|\mathbf{y}|^6} \mathbf{W}(\mathbf{y}) \right). \quad (47)$$

This divergence in the  $\sigma$ -metric agrees with the Euclidean divergence of the exterior domain:

$$(\text{div}^{\sigma} \mathbf{W})(\mathbf{y}) = (\nabla \cdot d\Phi(\mathbf{W}))(\Phi(\mathbf{y})). \quad (48)$$

### A.3 Laplacian

Combining (42) and (47), the Laplacian  $\Delta^{\sigma} U = \text{div}^{\sigma}(\text{grad}^{\sigma} U)$  of a function  $U$  is given by

$$(\Delta^{\sigma} U)(\mathbf{y}) = \frac{1}{\sigma(\mathbf{y})^3} \nabla \cdot (\sigma(\mathbf{y}) \nabla U(\mathbf{y})) = |\mathbf{y}|^6 \nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y}) \right). \quad (49)$$

By (43) and (48), this Laplacian  $\Delta^{\sigma} U$  in the  $\sigma$ -metric agrees with the Euclidean Laplacian on the exterior domain:

$$(\Delta^{\sigma} U)(\mathbf{y}) = (\Delta u)(\Phi(\mathbf{y})), \quad u(\mathbf{x}) = U(\Phi^{-1}(\mathbf{y})). \quad (50)$$

In particular, a Poisson equation on the exterior domain

$$(\Delta u)(\mathbf{x}) = f(\mathbf{x}) \quad (51)$$

translates into

$$(\Delta^{\sigma} U)(\mathbf{y}) = F(\mathbf{y}) \quad (52)$$

for  $U(\mathbf{y}) = u(\Phi(\mathbf{y}))$  and  $F(\mathbf{y}) = f(\Phi(\mathbf{y}))$ , and therefore

$$|\mathbf{y}|^6 \nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y}) \right) = F(\mathbf{y}). \quad (53)$$

## B KELVIN TRANSFORM FOR THE POISSON EQUATION

This appendix is the proof for Theorem 1. Suppose the function  $U(\mathbf{y})$  satisfies the exterior domain Poisson equation written in the inverted domain  $\Omega_{\text{inv}}$  (Appendix A.3)

$$\nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y}) \right) = \frac{1}{|\mathbf{y}|^6} F(\mathbf{y}). \quad (54)$$

We derive the equation for  $V(\mathbf{y})$  under the classical Kelvin transformation (cf. (7))

$$U(\mathbf{y}) = |\mathbf{y}|V(\mathbf{y}). \quad (55)$$

With the substitution (55), the left-hand side of (54) becomes

$$\begin{aligned} \nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla(|\mathbf{y}|V) \right) &= \nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \left( \frac{\mathbf{y}}{|\mathbf{y}|} V + |\mathbf{y}| \nabla V \right) \right) \\ &= \nabla \cdot \left( \frac{\mathbf{y}}{|\mathbf{y}|^3} V + \frac{1}{|\mathbf{y}|} \nabla V \right) \\ (\nabla \cdot (\mathbf{y}/|\mathbf{y}|^3) = 0 \text{ on } \Omega_{\text{inv}}) &= \frac{\mathbf{y}}{|\mathbf{y}|^3} \cdot \nabla V - \frac{\mathbf{y}}{|\mathbf{y}|^3} \cdot \nabla V + \frac{1}{|\mathbf{y}|} \nabla \cdot \nabla V \\ &= \frac{1}{|\mathbf{y}|} \Delta V, \quad \mathbf{y} \in \Omega_{\text{inv}} \end{aligned} \quad (56)$$

Therefore, (54) is equivalent to

$$\Delta V(\mathbf{y}) = \frac{1}{|\mathbf{y}|^3} F(\mathbf{y}), \quad \mathbf{y} \in \Omega_{\text{inv}}. \quad (57)$$

Note that the domain  $\Omega_{\text{inv}}$  is still a punctured domain that does not contain the origin. This punctured domain can support singular solutions such as those exhibiting  $V(\mathbf{y}) \sim O(1/|\mathbf{y}|)$  around the origin, undoing the change of variables (55). By extending the domain to  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$  we purposefully select solutions to (57) that are regular across the origin. This is effectively imposing the condition (3).

Technically, we can extend the domain to  $\Omega_{\text{inv}} \cup \{\mathbf{0}\}$  by viewing (57) in the weak sense and including the measure zero set  $\{\mathbf{0}\}$ . This is possible with an assumption of the decay rate of  $F$  around  $\mathbf{0}$  on the right-hand side. For example,  $|\mathbf{y}|^{-5} F(\mathbf{y}) \in L^2(\Omega_{\text{inv}})$ . In most scenarios  $F$  is identically zero in a neighborhood of the origin.

### B.1 Transformation of Neumann Boundary Conditions

Here we derive the boundary condition for  $V$  if the original Poisson problem is imposed with a Neumann boundary condition (Section 4.1.1). Let  $\Sigma = \partial\Omega$  be the boundary on the exterior domain with unit normal  $\mathbf{n}$ . Let  $\mathbf{N} = d\Phi^{-1}(\mathbf{n})/|d\Phi^{-1}(\mathbf{n})|$  be the unit normal of the inverted boundary  $\Sigma_{\text{inv}} = \Phi^{-1}(\Sigma)$ . Here the conformality of  $\Phi$  ensures the orthogonality of  $\mathbf{N}$  to  $\Sigma_{\text{inv}}$ . By Appendix A.1.1, a Neumann boundary condition  $\mathbf{n}(\mathbf{x}) \cdot \nabla u(\mathbf{x}) = q(\mathbf{x})$ ,  $\mathbf{x} \in \Sigma$ , becomes

$$|\mathbf{y}|^2 \mathbf{N}(\mathbf{y}) \cdot \nabla U(\mathbf{y}) = q(\Phi(\mathbf{y})), \quad \mathbf{y} \in \Sigma_{\text{inv}}. \quad (58)$$

The substitution (55) yields

$$\begin{aligned} |\mathbf{y}|^2 \mathbf{N} \cdot \nabla U &= |\mathbf{y}|^2 \mathbf{N} \cdot \nabla(|\mathbf{y}|V) = |\mathbf{y}|^2 \mathbf{N} \cdot \frac{\mathbf{y}}{|\mathbf{y}|} V + |\mathbf{y}|^3 \mathbf{N} \cdot \nabla V \\ &= \mathbf{N} \cdot \mathbf{y} |\mathbf{y}| V + |\mathbf{y}|^3 \mathbf{N} \cdot \nabla V. \end{aligned} \quad (59)$$

Hence (58) is equivalent to a Robin boundary condition for  $V$ :

$$\mathbf{N} \cdot \nabla V + \frac{\mathbf{N} \cdot \mathbf{y}}{|\mathbf{y}|^2} V = \frac{1}{|\mathbf{y}|^3} q(\Phi(\mathbf{y})), \quad \mathbf{y} \in \Sigma_{\text{inv}}. \quad (60)$$

### C ROBIN BOUNDARY CONDITION

In this appendix we describe how to impose a Robin boundary condition for the Poisson equation (Section 5.1). Notice that the Poisson equation with a Robin boundary condition

$$\begin{cases} \Delta u(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \in \Omega \\ \mathbf{n} \cdot \nabla u(\mathbf{x}) + a(\mathbf{x})u(\mathbf{x}) = q(\mathbf{x}), & \mathbf{x} \in \Sigma \end{cases} \quad (61)$$

is the Euler–Lagrange equation for the critical point of the functional

$$\int_{\Omega} \frac{1}{2} |\nabla u|^2 dV + \int_{\Sigma} \frac{1}{2} a u^2 dA + \int_{\Omega} u f dV - \int_{\Sigma} q u dA. \quad (62)$$

Let  $\mathbf{u}$  denote an array representing a discretization of  $u$ . Then, the corresponding discretization of the first two quadratic forms of (62) are respectively  $\frac{1}{2} \mathbf{u}^T \mathbf{L} \mathbf{u}$  and  $\frac{1}{2} \mathbf{u}^T \mathbf{D} \mathbf{u}$ , where  $\mathbf{L}$  is the discrete Laplacian (positive semi-definite), and  $\mathbf{D}$  is essentially a diagonal matrix with nonzero entries corresponding to the location on  $\Sigma$  such that  $\mathbf{u}^T \mathbf{D} \mathbf{u} \approx \int_{\Sigma} a u^2 dA$ . Therefore, given an existing discrete Laplacian  $\mathbf{L}$ , one constructs  $\tilde{\mathbf{L}} = \mathbf{L} + \mathbf{D}$ ,  $\tilde{\mathbf{f}} = \mathbf{f} - \mathbf{q}$ , and solves the linear system  $\tilde{\mathbf{L}} \mathbf{u} = -\tilde{\mathbf{f}}$ .

### D KELVIN TRANSFORM FOR THE HELMHOLTZ EQUATION

This appendix is the proof for Theorem 2. Suppose  $u(\mathbf{x})$  satisfies the exterior domain Helmholtz equation

$$\Delta u(\mathbf{x}) + \omega^2 u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad (63)$$

which translates into (Appendix A.3)

$$\nabla \cdot \left( \frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y}) \right) + \frac{1}{|\mathbf{y}|^6} \omega^2 U(\mathbf{y}) = 0, \quad \mathbf{y} \in \Omega_{\text{inv}}. \quad (64)$$

In the following, we derive the equation satisfied by the new variable (34)

$$V(\mathbf{y}) = \frac{1}{|\mathbf{y}|} e^{-\frac{i\omega}{|\mathbf{y}|}} U(\mathbf{y}), \quad \mathbf{y} \in \Omega_{\text{inv}}. \quad (65)$$

by substituting  $U(\mathbf{y}) = e^{\frac{i\omega}{|\mathbf{y}|}} |\mathbf{y}| V(\mathbf{y})$  in (64). We first calculate  $\nabla U$ :

$$\begin{aligned} \nabla U(\mathbf{y}) &= \nabla \left( |\mathbf{y}| e^{\frac{i\omega}{|\mathbf{y}|}} V(\mathbf{y}) \right) \\ &= \frac{\mathbf{y}}{|\mathbf{y}|} e^{\frac{i\omega}{|\mathbf{y}|}} V(\mathbf{y}) - |\mathbf{y}| \frac{i\omega \mathbf{y}}{|\mathbf{y}|^3} e^{\frac{i\omega}{|\mathbf{y}|}} V(\mathbf{y}) + |\mathbf{y}| e^{\frac{i\omega}{|\mathbf{y}|}} \nabla V(\mathbf{y}) \\ &= e^{\frac{i\omega}{|\mathbf{y}|}} \left[ \left( 1 - \frac{i\omega}{|\mathbf{y}|} \right) \frac{\mathbf{y}}{|\mathbf{y}|} V(\mathbf{y}) + |\mathbf{y}| \nabla V(\mathbf{y}) \right]. \end{aligned} \quad (66)$$

Hence, the expression  $\frac{1}{|\mathbf{y}|^2} \nabla U(\mathbf{y})$  in (64) becomes

$$\frac{\nabla U(\mathbf{y})}{|\mathbf{y}|^2} = e^{\frac{i\omega}{|\mathbf{y}|}} \left[ \left( \frac{1}{|\mathbf{y}|^2} - \frac{i\omega}{|\mathbf{y}|^3} \right) \frac{\mathbf{y}}{|\mathbf{y}|} V(\mathbf{y}) + \frac{\nabla V(\mathbf{y})}{|\mathbf{y}|} \right]. \quad (67)$$

Now, we compute the first term in (64)

$$\begin{aligned} \nabla \cdot \left( \frac{\nabla U(\mathbf{y})}{|\mathbf{y}|^2} \right) &= -e^{\frac{i\omega}{|\mathbf{y}|}} \frac{i\omega \mathbf{y}}{|\mathbf{y}|^3} \cdot \left[ \left( \frac{1}{|\mathbf{y}|^2} - \frac{i\omega}{|\mathbf{y}|^3} \right) \frac{\mathbf{y}}{|\mathbf{y}|} V(\mathbf{y}) + \frac{\nabla V(\mathbf{y})}{|\mathbf{y}|} \right] \\ &\quad + e^{\frac{i\omega}{|\mathbf{y}|}} \left[ \left( -\frac{2}{|\mathbf{y}|^3} + \frac{3i\omega}{|\mathbf{y}|^4} \right) V(\mathbf{y}) \right. \\ &\quad \left. + \left( \frac{1}{|\mathbf{y}|^2} - \frac{i\omega}{|\mathbf{y}|^3} \right) \left( \frac{2}{|\mathbf{y}|} V(\mathbf{y}) + \frac{\mathbf{y}}{|\mathbf{y}|} \cdot \nabla V(\mathbf{y}) \right) \right. \\ &\quad \left. - \frac{\mathbf{y}}{|\mathbf{y}|^3} \cdot \nabla V(\mathbf{y}) + \frac{\Delta V(\mathbf{y})}{|\mathbf{y}|} \right]. \end{aligned} \quad (68)$$

Here, we have used  $\nabla \cdot \frac{\mathbf{y}}{|\mathbf{y}|} = \frac{2}{|\mathbf{y}|}$ . Expanding (68) we arrive at

$$\nabla \cdot \left( \frac{\nabla U(\mathbf{y})}{|\mathbf{y}|^2} \right) = e^{\frac{i\omega}{|\mathbf{y}|}} \left( -\frac{\omega^2}{|\mathbf{y}|^5} V(\mathbf{y}) - \frac{2i\omega \mathbf{y}}{|\mathbf{y}|^4} \cdot \nabla V(\mathbf{y}) + \frac{\Delta V(\mathbf{y})}{|\mathbf{y}|} \right). \quad (69)$$

Combining it with the second term in (64)

$$\frac{\omega^2}{|\mathbf{y}|^6} U(\mathbf{y}) = \frac{\omega^2}{|\mathbf{y}|^5} e^{\frac{i\omega}{|\mathbf{y}|}} V(\mathbf{y}), \quad (70)$$

and removing the factor  $\frac{1}{|\mathbf{y}|} e^{\frac{i\omega}{|\mathbf{y}|}}$  on every term, we obtain that Eq. (64) is equivalent to

$$\Delta V(\mathbf{y}) - 2i\omega \frac{\mathbf{y}}{|\mathbf{y}|^3} \cdot \nabla V(\mathbf{y}) = 0. \quad (71)$$