A Practical Walk-on-Boundary Method for Boundary Value Problems

RYUSUKE SUGIMOTO, University of Waterloo, Canada
TERRY CHEN, University of Waterloo, Canada
YITI JIANG, University of Waterloo, Canada
CHRISTOPHER BATTY, University of Waterloo, Canada
TOSHIYA HACHISUKA, University of Waterloo, Canada

We introduce the *walk-on-boundary* (WoB) method for solving boundary value problems to computer graphics. WoB is a grid-free Monte Carlo solver for certain classes of second order partial differential equations. A similar Monte Carlo solver, the walk-on-spheres (WoS) method, has been recently popularized in computer graphics due to its advantages over traditional spatial discretization-based alternatives. We show that WoB's intrinsic properties yield further advantages beyond those of WoS. Unlike WoS, WoB naturally supports various boundary conditions (Dirichlet, Neumann, Robin, and mixed) for both interior and exterior domains. WoB builds upon boundary integral formulations, and it is mathematically more similar to light transport simulation in rendering than the random walk formulation of WoS. This similarity between WoB and rendering allows us to implement WoB on top of Monte Carlo ray tracing, and to incorporate advanced rendering techniques (e.g., bidirectional estimators with multiple importance sampling, the virtual point lights method, and Markov chain Monte Carlo) into WoB. WoB does not suffer from the intrinsic bias of WoS near the boundary and can estimate solutions precisely on the boundary. Our numerical results highlight the advantages of WoB over WoS as an attractive alternative to solve boundary value problems based on Monte Carlo.

CCS Concepts: • **Mathematics of computing** → **Integral equations**; *Partial differential equations*; • **Computing methodologies** → *Ray tracing*.

Additional Key Words and Phrases: Monte Carlo, Walk on Boundary

ACM Reference Format:
Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4 (August 2023), 16 pages. https://doi.org/10.1145/3592109

1 INTRODUCTION

Boundary value problems are relevant to diverse applications in computer graphics and beyond. Since the analytical solution is generally unavailable, one must resort to a numerical method to estimate the solution for practical problems. Such numerical methods are conventionally based on discretizing and then solving matrix equations. Sawhney and Crane [2020] recently introduced to computer graphics an alternative Monte Carlo (MC) solver called the *walk-on-spheres* (WoS) method [Muller 1956]. They showed that WoS possesses various advantages over the conventional methods, including flexibility of geometric representation, robustness, parallelism, applicability to many geometric problems, and pointwise solution estimations. The connection between WoS and MC ray tracing [Pharr et al. 2018] was also suggested by these authors and others [Rioux-Lavoie et al. 2022; Sawhney et al. 2022; Yılmazer et al. 2022], which enables various techniques from MC ray tracing to be modified and adapted to WoS, such as in recent work by Qi et al. [2022] on bidirectional WoS.

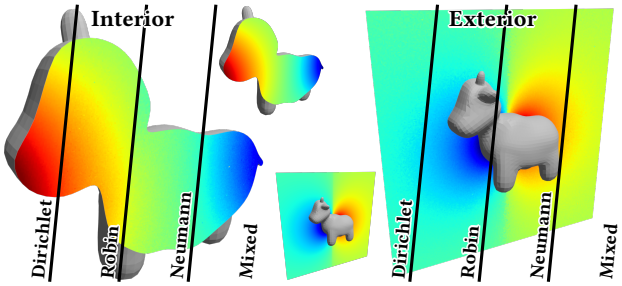


Fig. 1. The walk-on-boundary (WoB) method can handle various boundary value problems including Dirichlet, Robin, Neumann, and mixed for both interior and exterior problems under the same framework based on boundary integral equations. In this experiment, each problem is configured to have the same ground-truth solution (middle), and this figure shows that our WoB estimators all converge to that same solution (left and right).

We introduce another MC-based solver for boundary value problems: the *walk-on-boundary* (WoB) method [Sabelfeld 1982, 1991]. Unlike WoS, which is based on random walks, WoB is based on *potential theory*, which is the study of harmonic functions in mathematical physics. Potential theory allows us to convert the partial differential equations of boundary value problems into various forms of *boundary integral equations* (BIEs). Just like the integral equation in rendering [Kajiya 1986], the resulting equations can be solved by tracing rays, rather than using random walks as in WoS. WoB is arguably less widely known compared to WoS and we are the first to introduce WoB to computer graphics. WoB and WoS share many advantages over spatial discretization-based methods, like finite element or finite difference methods, but WoB possesses further advantages that fundamentally differentiate it from WoS. We explore and demonstrate those advantages, specifically:

- **Generality:** WoB can handle problems for which WoS is either inapplicable or inefficient. Handling Neumann or Robin boundaries with WoS [Simonov 2008, 2017] is known to be inefficient for non-convex domains. Only concurrent work by Sawhney

Authors' addresses: Ryusuke Sugimoto, University of Waterloo, Canada, rsugimot@uwaterloo.ca; Terry Chen, University of Waterloo, Canada, ty6chen@uwaterloo.ca; Yiti Jiang, University of Waterloo, Canada, yt2jiang@uwaterloo.ca; Christopher Batty, University of Waterloo, Canada, christopher.batty@uwaterloo.ca; Toshiya Hachisuka, University of Waterloo, Canada, toshiya.hachisuka@uwaterloo.ca.

0730-0301/2023/8-ART $15.00
https://doi.org/10.1145/3592109

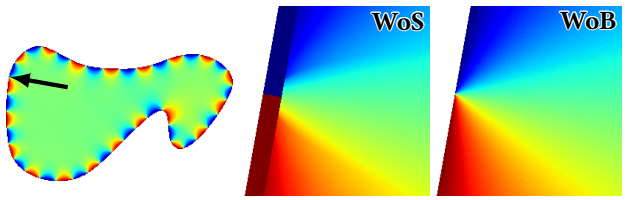


Fig. 2. Unlike WoS, WoB does not introduce any errors associated with the ϵ-shell path termination. The indicated region (left) displays visible banding artifacts with WoS (middle), where WoB presents no such error (right).

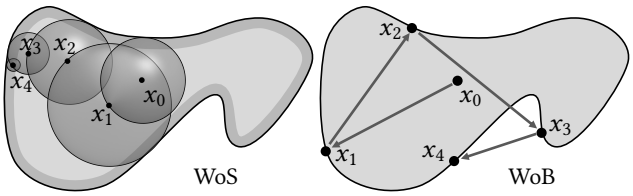


Fig. 3. While WoS takes random walks on spheres, WoB takes random walks on the boundary to compute a sample contribution.

et al. [2023] offers an extension to WoS for efficiently handling Neumann boundaries. Likewise, efficient handling of exterior problems in WoS requires Kelvin transformations into interior problems [Nabizadeh et al. 2021]. By formulating the appropriate integral equation, our WoB solver handles these cases effortlessly, without changing the core algorithm. Fig. 1 showcases this generality with results for Dirichlet, Neumann, Robin, and mixed boundary conditions for both interior and exterior domains.

- **Accuracy:** WoB can accurately estimate the solution close to and even directly *on* the boundary itself, where WoS becomes inefficient, inaccurate, or inapplicable. Fig. 2 compares the accuracy of WoS and WoB near the boundary. Solutions on the boundary are relevant for Neumann and Robin problems, since the boundary values are unknown, and they are the main quantity of interest for certain cases [Da et al. 2016; Sugimoto et al. 2022].
- **Similarity to MC ray tracing:** WoB performs calculations by sampling points *on* the boundary as in MC ray tracing; by contrast, WoS is based on random walks *inside* the boundary shape using closest point queries, which is not entirely equivalent to MC ray tracing. Fig. 3 illustrates this difference. WoB is both mathematically and algorithmically more similar to MC rendering than WoS. This similarity allows WoB to leverage existing ray tracing frameworks and straightforwardly transform advanced rendering methods into solvers for boundary value problems.

2 OVERVIEW

Let us examine instances of WoS and WoB estimators to highlight the differences. Consider a boundary value problem in which a function $\overline{u}_D(\mathbf{x})$ is specified along the boundary of a convex domain D, for simplicity. We will relax this assumption of convexity in Section 4. We seek a function $u(\mathbf{x})$ that conforms to $\overline{u}_D(\mathbf{x})$ on the boundary while satisfying Laplace's equation, $\Delta u(\mathbf{x}) = 0$, inside the domain.

The WoS estimator in this case is defined as the mean of sample contributions $\hat{u}(\mathbf{x}_0)$ defined recursively as

$$\hat{u}(\mathbf{x}_i) = \begin{cases} \overline{u}_D(\mathbf{x}_i) & \mathbf{x}_i \in \partial D_\epsilon, \\ \hat{u}(\mathbf{x}_{i+1}) & \text{otherwise.} \end{cases} \tag{1}$$

Starting from an evaluation point $\mathbf{x}_0$ within the domain, each subsequent $\mathbf{x}_{i+1}$ is generated by sampling a point on the largest ball centered at $\mathbf{x}_i$ that fits within the domain. The set ∂D_ϵ is a shell-like region that lies within a small distance of the boundary. WoS thus returns $\overline{u}_D(\mathbf{x}_i)$ when $\mathbf{x}_i$ is only *approximately* on the boundary; indeed, none of the generated points are exactly on the boundary.

The WoB estimator with path length M we introduce is defined as

$$\hat{u}(\mathbf{x}_i) = \begin{cases} \overline{u}_D(\mathbf{x}_{i+1}) & i = M-1, \\ 2\overline{u}_D(\mathbf{x}_{i+1}) - \hat{u}(\mathbf{x}_{i+1}) & \text{otherwise.} \end{cases} \tag{2}$$

Like WoS, WoB forms a sequence of $\mathbf{x}_i$ starting from the evaluation point $\mathbf{x}_0$. Each $\mathbf{x}_{i+1}$, however, is generated by tracing a random ray from $\mathbf{x}_k$ and finding the intersection with the boundary. Unlike WoS, all the points (after $\mathbf{x}_0$) are *exactly* on the boundary, and thus $\overline{u}_D(\mathbf{x}_{i+1})$ is well-defined, without approximation via the ϵ-shell ∂D_ϵ. As in MC ray tracing, the recursion terminates after some number of steps M and does not depend at all on the ϵ-shell ∂D_ϵ. Algorithm 1 summarizes this method and Fig. 3 illustrates how sequences of $\mathbf{x}_i$ differ between WoS and WoB in a non-convex domain.

Just like MC ray tracing, in both WoS and WoB, one would run the estimator multiple times and take its average as a final estimation of $u(\mathbf{x}_0)$. However, because WoB is also based on sampling and tracing rays recursively, the analogy with MC ray tracing is much stronger, in sharp contrast to the random walk on spheres process used by WoS. One of our contributions is to demonstrate WoB's strong mathematical and algorithmic similarity to MC ray tracing.

Algorithm 1: Interior, convex-domain, and Dirichlet WoB

Input: domain Ω, evaluation point $\mathbf{x}_0 \in \Omega$, path length M, sample count N

Function `EstimateSolution`(Ω, $\mathbf{x}_0$, M, N)**:**
- $\hat{u}_{\text{sum}} \leftarrow 0$
- **for** $n \leftarrow 1$ *to* N **do**
  - $\hat{u} \leftarrow$ `RecursiveEstimate` (Ω, $\mathbf{x}_0$, M, 0)
  - $\hat{u}_{\text{sum}} \leftarrow \hat{u}_{\text{sum}} + \hat{u}$
- **end**
- **return** $\hat{u}_{\text{sum}}/N$

Function `RecursiveEstimate`(Ω, $\mathbf{x}_i$, M, depth)**:**
- $\mathbf{x}_{i+1} \leftarrow$ `RayIntersectionSampling`(Ω, $\mathbf{x}_i$)
- **if** depth $= M-1$ **then**
  - **return** $\overline{u}_D(\mathbf{x}_{i+1})$
- **else**
  - $\hat{u}_{i+1} \leftarrow$ `RecursiveEstimate`(Ω, $\mathbf{x}_i$, M, depth + 1)
  - **return** $2\overline{u}_D(\mathbf{x}_{i+1}) - \hat{u}_{i+1}$
- **end**

Function `RayIntersectionSampling`(Ω, $\mathbf{x}$)**:**
- $\mathbf{d} \leftarrow$ `UniformRayDirectionSampling`(Ω, $\mathbf{x}$)
- **return** `GetIntersectionPoint`(Ω, `Ray`($\mathbf{x}$, $\mathbf{d}$))

The ray tracing approach to diffusion curves [Bowers et al. 2011] also solves a boundary integral and estimates the solution by sampling points on the boundary using ray intersections, similarly to WoB. While this approach provides a *visually* faithful approximation for diffusion curves [Orzan et al. 2008], it does not actually solve Laplace's equation. By contrast, the WoB formulation does solve the specified PDE (i.e., Laplace's equation) and can handle more general problems, such as Neumann problems.

3 BOUNDARY INTEGRAL FORMULATIONS

WoB builds upon boundary integral equation formulations. Such formulations are applicable to many second-order linear elliptic PDEs [Clements 2004] and some other PDEs [Liu et al. 2012], but we focus on boundary value problems based on Poisson's and Laplace's equations. The book by Sabelfeld and Simonov [1994] and notes by Pechstein [2013] discuss the details of BIEs summarized here.

3.1 Poisson's Equation and Boundary Conditions

Poisson's equation is commonly given as

$$\Delta v(\mathbf{x}) = \overline{b}(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega \tag{3}$$

where Ω is a closed domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, $v(\mathbf{x})$ is the unknown field, and $\overline{b}(\mathbf{x})$ is a known source function defined inside the domain. Poisson's equation with $\overline{b}(\mathbf{x}) = 0$ is also called Laplace's equation. We will later consider exterior problems where the domain is $\mathbb{R}^2 \backslash \Omega$ or $\mathbb{R}^3 \backslash \Omega$. Let us denote the boundary of Ω by $\Gamma = \partial\Omega$; note that Γ is not included in Ω. The *boundary value problem* in this paper concerns solving for $v(\mathbf{x})$ in Eq. (3) with boundary conditions

$$\begin{aligned} v(\mathbf{x}) &= \overline{u}_D(\mathbf{x}) && \mathbf{x} \in \Gamma_D \subseteq \Gamma, \\ \frac{\partial v}{\partial \mathbf{n}}(\mathbf{x}) &= \overline{q}_N(\mathbf{x}) && \mathbf{x} \in \Gamma_N \subseteq \Gamma, \\ \frac{\partial v}{\partial \mathbf{n}}(\mathbf{x}) + \overline{\alpha}(\mathbf{x}) v(\mathbf{x}) &= \overline{g}_R(\mathbf{x}) && \mathbf{x} \in \Gamma_R \subseteq \Gamma, \end{aligned} \tag{4}$$

where the normal derivative $\frac{\partial v}{\partial \mathbf{n}}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \nabla v(\mathbf{x})$, the weight $\overline{\alpha}(\mathbf{x}) \neq 0$, and a bar indicates that the function is given. Any point $\mathbf{x} \in \Gamma$ belongs to strictly one of Γ_D, Γ_N, or Γ_R. When $\Gamma_D = \Gamma$, $\Gamma_N = \Gamma$, or $\Gamma_R = \Gamma$, we call the problem a Dirichlet, Neumann, or Robin problem, respectively. Otherwise, we call it a mixed boundary problem.

One can always convert Poisson's equation for $v(\mathbf{x})$ into Laplace's equation for $u(\mathbf{x})$ using a relation $u(\mathbf{x}) = v(\mathbf{x}) + V_0(\mathbf{x})$ as follows. Let u be a function satisfying Laplace's equation

$$\begin{aligned} \Delta u(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega \\ u(\mathbf{x}) &= \overline{u}_D(\mathbf{x}) + V_0(\mathbf{x}) && \mathbf{x} \in \Gamma_D, \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) &= \overline{q}_N(\mathbf{x}) + \frac{\partial V_0}{\partial \mathbf{n}}(\mathbf{x}) && \mathbf{x} \in \Gamma_N, \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) + \overline{\alpha}(\mathbf{x}) u(\mathbf{x}) &= \overline{g}_R(\mathbf{x}) + \frac{\partial V_0}{\partial \mathbf{n}}(\mathbf{x}) + \overline{\alpha}(\mathbf{x}) V_0(\mathbf{x}) && \mathbf{x} \in \Gamma_R, \end{aligned} \tag{5}$$

where $V_0(\mathbf{x}) = \int_\Omega G(\mathbf{x}, \mathbf{y}) \overline{b}(\mathbf{y}) \, dV_\mathbf{y}$ and $G(\mathbf{x}, \mathbf{y})$ is the fundamental solution (i.e., the solution to Poisson's equation in an infinite domain with a negative Dirac delta source at each point). We list the fundamental solution and their derivatives used in this paper in Appendix A for readers' convenience. One can confirm that $v(\mathbf{x}) = u(\mathbf{x}) - V_0(\mathbf{x})$ satisfies Poisson's equation by noting that $\Delta V_0(\mathbf{x}) = -\overline{b}(\mathbf{x})$ for $\mathbf{x} \in \Omega$. We focus on solutions to Laplace's equation since it is easy to handle $V_0(\mathbf{x})$, as we discuss later.

3.2 Direct Boundary Integral Equations

The basic idea of boundary integral formulations is to define the solution based on integrals *only of boundary values*. They can actually take several different forms. We first explain one such formulation called a *direct* BIE formulation. As the name suggests, direct BIEs describe the relationship between the solution values in the interior or on the boundary *directly*, and are derived based on Green's third identity. One common such direct BIE is given as

$$c(\mathbf{x}) u(\mathbf{x}) = -\int_\Gamma \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, dA_\mathbf{y} + \int_\Gamma G(\mathbf{x}, \mathbf{y}) \frac{\partial u}{\partial \mathbf{n}}(\mathbf{y}) \, dA_\mathbf{y} \tag{6}$$

for $\mathbf{x} \in \Omega \cup \Gamma$, where $c(\mathbf{x})$ is an integral free term that evaluates to 1 if $\mathbf{x} \in \Omega$, and to $\frac{1}{2}$ if $\mathbf{x} \in \Gamma$ when the boundary is smooth in the sense of Lyapunov. We consider only smooth surfaces in the following, but the extension to non-smooth surfaces is straightforward. Note that polygonal boundaries do not violate the smoothness assumption, unless $\mathbf{x}$ is evaluated *exactly* on a vertex of the polygon.

Eq. (6) says that the solution $u(\mathbf{x})$ satisfies this integral equation involving only boundary integrals. The boundary conditions alone do not provide $u(\mathbf{y})$ and $\frac{\partial u}{\partial \mathbf{n}}(\mathbf{y})$ everywhere along Γ, so one must solve for such unknown boundary values first to evaluate the solution $u(\mathbf{x})$ inside of the domain. Concurrent work by Miller et al. [2023] uses this equation to cache WoS (and Walk-on-Stars [Sawhney et al. 2023]) estimates along the boundary to accelerate its computation; our WoB readily supports an analogous caching scheme.

One can also derive a BIE for the directional derivative of the solution by taking the directional derivative of Eq. (6):

$$\frac{\partial u}{\partial \mathbf{x}_k}(\mathbf{x}) = -\int_\Gamma \frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}}(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, dA_\mathbf{y} + \int_\Gamma \frac{\partial G}{\partial \mathbf{x}_k}(\mathbf{x}, \mathbf{y}) \frac{\partial u}{\partial \mathbf{n}}(\mathbf{y}) \, dA_\mathbf{y}, \tag{7}$$

for $\mathbf{x} \in \Omega$. We denote the first order derivative with respect to the k-th direction by $\partial u / \partial \mathbf{x}_k$. This equation is valid only in the interior of the domain, so we omitted the integral free term $c(\mathbf{x}) = 1$.

3.3 Indirect Boundary Integral Equations

In contrast to direct BIEs, an *indirect* BIE describes the relationship between an unknown *source density function* on the boundary and the known boundary values, and expresess the solution only *indirectly* based on the source density function. There are two types of indirect formulations, derived from potential theory.

Single layer potential. The solution $u(\mathbf{x})$ to Laplace's equation can be expressed in the form of a *single layer potential* given by

$$u(\mathbf{x}) = \int_\Gamma G(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y}) \, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Omega \cup \Gamma, \tag{8}$$

where the unknown source density function μ corresponds to the jump of the normal derivative of u across the boundary. In short, Eq. (8) expresses the solution inside the domain (and on the boundary) in terms of monopole sources, distributed over the boundary, which decay according to the fundamental solution $G(\mathbf{x}, \mathbf{y})$. Taking the directional derivative of Eq. (8) and taking the limit to the

boundary gives an integral equation for $\mu(\mathbf{x})$:

$$\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) = c(\mathbf{x})\mu(\mathbf{x}) + \int_\Gamma \frac{\partial G}{\partial \mathbf{n}_\mathbf{x}}(\mathbf{x},\mathbf{y})\mu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Gamma. \tag{9}$$

Similarly, the directional derivative at any interior point $\mathbf{x}$ is

$$\frac{\partial u}{\partial \mathbf{x}_k}(\mathbf{x}) = \int_\Gamma \frac{\partial G}{\partial \mathbf{x}_k}(\mathbf{x},\mathbf{y})\mu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Omega. \tag{10}$$

This equation is invalid exactly on the boundary because of jump discontinuities across Γ.

Double layer potential. An alternative is a *double layer potential* which uses dipole source on the boundary:

$$u(\mathbf{x}) = -\int_\Gamma \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})\nu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Omega \tag{11}$$

where ν is an unknown source density function, corresponding to the jump of the solution across the boundary. In the limit as $\mathbf{x} \to \Gamma$, one finds an integral equation for $\nu(\mathbf{x})$:

$$u(\mathbf{x}) = [1 - c(\mathbf{x})]\nu(\mathbf{x}) - \int_\Gamma \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})\nu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Gamma. \tag{12}$$

The normal derivative can be computed with

$$\frac{\partial u}{\partial \mathbf{n}_\mathbf{x}}(\mathbf{x}) = -\frac{\partial}{\partial \mathbf{n}_\mathbf{x}} \int_\Gamma \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})\nu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Gamma, \tag{13}$$

and the directional derivative can be computed with

$$\frac{\partial u}{\partial \mathbf{x}_k}(\mathbf{x}) = -\int_\Gamma \frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})\nu(\mathbf{y})\, dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Omega. \tag{14}$$

Having outlined BIE formulations for Laplace's equation, we proceed to solve them via WoB. To develop practical numerical WoB methods and also generalize WoB to mixed boundary problems, we will adopt different formulations for different problems.

4 THE WALK-ON-BOUNDARY METHOD

WoB is a stochastic estimator for certain classes of second-order PDEs based on BIEs. It uses a sequence of stochastically chosen sample points on the boundary, hence the name "walk-on-boundary". Sabelfeld [1982] first proposed WoB for the Lamé equation for linear elasticity. Subsequent books by Sabelfeld [1991] and Sabelfeld and Simonov [1994] generalized the method and established theoretical foundations for Dirichlet, Neumann, and Robin problems for Poisson's equation using the indirect BIE formulation, along with extensions to a few other equations. Karaivanova et al. [2004] considered the applicability of quasi-Monte Carlo methods and Sabelfeld [2012] studied how WoB can be combined with the method of fundamental solutions to improve its efficiency.

None of this prior work has considered the application of WoB ideas to direct BIE formulations or mixed boundary problems; these extensions are our contributions. We also have identified that using double layer potential indirect BIEs for Dirichlet boundary problems, single layer potential indirect BIEs or direct BIEs for Neumann boundary problems, and single layer potential indirect BIEs for Robin and mixed boundary problems will result in practical solvers. We summarize the equations for formulations for all of our estimators discussed in this paper in Table 1.

4.1 Dirichlet Problems with Double Layer BIE

For Dirichlet problems, reordering terms in Eq. (12) and substituting in the boundary condition $u = \overline{u}_D$ gives

$$\nu(\mathbf{x}) = \int_\Gamma 2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})\nu(\mathbf{y})\, dA_\mathbf{y} + 2\overline{u}_D(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Gamma, \tag{15}$$

assuming that $\mathbf{x}$ lies on a smooth boundary ($c = 1/2$). Since $\overline{u}_D(\mathbf{x})$ is a known quantity, Eq. (15) is a Fredholm equation of the second kind for ν, as the rendering equation is also commonly understood to be. Appendix B elaborates on this point. We can thus apply a recursive estimate for $\nu(\mathbf{x})$ similarly to what is done for the rendering equation in light transport simulation [Pharr et al. 2018].

4.1.1 MC estimation. Let us consider estimating $\nu(\mathbf{x}_1)$ where $\mathbf{x}_1 \in \Gamma$ based on MC integration. We can estimate the integral in Eq. (15) via MC integration by first sampling a point $\mathbf{x}_2 \in \Gamma$ with a probability density function (PDF) $p(\mathbf{x}_2|\mathbf{x}_1)$ (e.g., tracing a random ray from $\mathbf{x}_1$ to $\mathbf{x}_2$). A sample $\hat{\nu}(\mathbf{x}_1)$ to estimate $\nu(\mathbf{x}_1)$ can be written as

$$\hat{\nu}(\mathbf{x}_1) \coloneqq \frac{2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_1,\mathbf{x}_2)}{p(\mathbf{x}_2|\mathbf{x}_1)}\nu(\mathbf{x}_2) + 2\overline{u}_D(\mathbf{x}_1) \quad \text{for} \quad \mathbf{x}_1 \in \Gamma. \tag{16}$$

Because $\nu(\mathbf{x}_2)$ is unknown, we again use an MC estimate $\hat{\nu}(\mathbf{x}_2)$ in the equation above. Thus a recursive definition for the i-th step is:

$$\hat{\nu}(\mathbf{x}_i) \coloneqq \frac{2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_i,\mathbf{x}_{i+1})}{p(\mathbf{x}_{i+1}|\mathbf{x}_i)}\hat{\nu}(\mathbf{x}_{i+1}) + 2\overline{u}_D(\mathbf{x}_i) \quad \text{for} \quad \mathbf{x}_i \in \Gamma. \tag{17}$$

Just like MC ray tracing, we perform the recursive estimate of $\nu(\mathbf{x})$ up to a certain recursion depth M, forming a path of vertices on the boundary with length M. We can use $\hat{\nu}(\mathbf{x}_1)$ to construct an MC estimate for the solution u at an interior point $\mathbf{x}_0$. Applying another MC integration with a PDF $p(\mathbf{x}_1|\mathbf{x}_0)$ and $\hat{\nu}(\mathbf{x}_1)$ to Eq. (11) gives

$$\hat{u}(\mathbf{x}_0) \coloneqq -\frac{\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_0,\mathbf{x}_1)}{p(\mathbf{x}_1|\mathbf{x}_0)}\hat{\nu}(\mathbf{x}_1) \quad \text{for} \quad \mathbf{x}_0 \in \Omega. \tag{18}$$

Therefore, the MC estimate for $u(\mathbf{x}_0)$ in the domain interior is $u(\mathbf{x}_0) \approx \frac{1}{N}\sum_{n=1}^{N} \hat{u}(\mathbf{x}_0)$. One can think of the PDF $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ as the PDF of sampling a ray from $\mathbf{x}_i$ toward $\mathbf{x}_{i+1}$, the term $2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_i,\mathbf{x}_{i+1})$ as the geometry term times BRDF term (i.e., the integrand) of the rendering equation, and the term $2\overline{u}_D(\mathbf{x}_i)$ as the emission term in the rendering equation. Implementation of WoB on top of ray tracing systems is thus straightforward.

4.1.2 Path truncation. One difference between the rendering equation and the above BIE is that, for the rendering equation, as the path length increases, the contribution coming from each recursion becomes smaller and smaller due to the nature of light transport. However, the integral kernel $2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}$ above will not "attenuate" its contribution per recursion, but will rather maintain it. It thus appears that it never converges, just like having reflectance equal to one everywhere does not converge in light transport. This intuition contradicts with the fact that solutions usually uniquely exist for Dirichlet problems and it is incorrect.

A solution to this issue is surprisingly simple. We just need to multiply the contribution $2\overline{u}_D(\mathbf{x}_i)$ coming from the last recursion step by a factor of $1/2$ before it terminates: $\hat{\nu}(\mathbf{x}_M) \coloneqq \overline{u}_D(\mathbf{x}_M)$. While this strategy deceivingly looks the same as just truncating

Table 1. List of equations for WoB estimators. The highlighted equations are the second kind Fredholm equations (or the modified first-kind equation for mixed boundary problems at Dirichlet boundaries) we use to get the unknown direct or indirect quantities on the boundary. It can be combined with other equations to find the unknown quantities of interest in the interior (or exterior) or on the boundary. It is assumed that the left hand side unknowns are functions of $\mathbf{x}$, and the integrals are taken over boundary points $\mathbf{y}$. The explicit dependencies on variables are omitted for brevity when it is not confusing. For interior problems, $\phi = 1$, and for exterior problems, $\phi = -1$.

Problem	Formulation	Quantity to Estimate	Evaluation Point $\mathbf{x}$	Equation
Dirichlet	indirect BIE double layer potential	solution	interior/exterior	$u = -\int \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}} v \, dA$
			boundary	$u = \overline{u}_D$ (given)
		normal derivative	boundary	Sect. 3 in the book by Sabelfeld and Simonov [1994].
		gradient	interior/exterior	$\frac{\partial u}{\partial \mathbf{x}_k} = -\int \frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}} v \, dA$
		source density	**boundary**	$v = 2\phi \int \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}} v \, dA + 2\phi \overline{u}_D$
Neumann	direct BIE	**solution**	interior/exterior	$u = -\phi \int \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}} u \, dA + \phi \int G \overline{q}_N \, dA$
			boundary	$u = -2\phi \int \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}} u \, dA + 2\phi \int G \overline{q}_N \, dA$
		normal derivative	boundary	$\frac{\partial u}{\partial \mathbf{n}} = \overline{q}_N$ (given)
		gradient	interior/exterior	$\frac{\partial u}{\partial \mathbf{x}_k} = -\phi \int \frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}} u \, dA + \phi \int \frac{\partial G}{\partial \mathbf{x}_k} \overline{q}_N \, dA$
Mixed or degenerate problem	indirect BIE single layer potential	solution	interior/exterior/boundary	$u = \int G\mu \, dA$ (or given)
		normal derivative	boundary	$\frac{\partial u}{\partial \mathbf{n}} = \frac{1}{2}\phi\mu + \int \frac{\partial G}{\partial \mathbf{n}_\mathbf{x}} \mu \, dA$ (or given)
		gradient	interior/exterior	$\frac{\partial u}{\partial \mathbf{x}_k} = \int \frac{\partial G}{\partial \mathbf{x}_k} \mu \, dA$
		source density	**Dirichlet boundary**	$\mu = \mu - k \int G\mu \, dA + k\overline{u}_D$
			Neumann boundary	$\mu = -2\phi \int \frac{\partial G}{\partial \mathbf{n}_\mathbf{x}} \mu \, dA + 2\phi \overline{q}_N$
			Robin boundary	$\mu = -2\phi \int \left(\frac{\partial G}{\partial \mathbf{n}_\mathbf{x}} + \overline{\alpha}(\mathbf{x}) G \right) \mu \, dA + 2\phi \overline{g}_R$

a path while reducing the contribution from the last "bounce", its derivation is more involved than that. Below, we briefly summarize the rough reasoning behind this strategy.

Let us first define an integral operator H when applied to a function f defined over the boundary as

$$(\mathrm{H} f)(\mathbf{x}) = \int_\Gamma 2 \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, dA_\mathbf{y}. \tag{19}$$

Then, Eq. (15) can be rewritten as

$$v = \mathrm{H} v + 2\overline{u}_D, \tag{20}$$

where we dropped the variable dependence for brevity. Using the identity operator I, we can write the expression above as

$$(\mathrm{I} - \mathrm{H}) v = 2\overline{u}_D. \tag{21}$$

One can use Neumann series expansion to solve for v as

$$v = (\mathrm{I} - \mathrm{H})^{-1} \, 2\overline{u}_D = (\mathrm{I} + \mathrm{H} + \mathrm{H}^2 + \cdots) \, 2\overline{u}_D, \tag{22}$$

where the operator H^i for any positive integer i is defined by

$$\mathrm{H}^i f = \mathrm{H}^{i-1}(\mathrm{H} f). \tag{23}$$

The same approach is used for building a recursive MC estimator for the rendering equation where the operator is defined by BRDF and the geometry term instead [Pharr et al. 2018].

As noted earlier, the key difference from the rendering equation is that $(\mathrm{H}^i) 2\overline{u}_D(\mathbf{x})$ does not approach zero as i increases. Simply truncating this series at M thus introduces non-negligible truncation error. We instead transform the series as

$$\begin{aligned} &\left(\frac{1}{2}(\mathrm{I} + \cdots + \mathrm{H}^{i-1} + \cdots) + \frac{1}{2}(\mathrm{I} + \cdots + \mathrm{H}^i + \cdots) \right) 2\overline{u}_D(\mathbf{x}) \\ &\quad = \frac{1}{2}(\mathrm{I} + (\mathrm{I} + \mathrm{H}) + \cdots + (\mathrm{H}^{i-1} + \mathrm{H}^i) + \cdots) 2\overline{u}_D(\mathbf{x}). \end{aligned} \tag{24}$$

Just like multiple bounces in light transport, the average of the integrals $(1/2)(\mathrm{H}^i + \mathrm{H}^{i+1}) 2\overline{u}_D(\mathbf{x})$ now converges to zero as $i \to \infty$ due to the alternating sign in the series because of the negative factor included in the operator H, so it is safe to truncate this modified series at $i = M$:

$$\begin{aligned} f(\mathbf{x}) &\approx \frac{1}{2} \left(\mathrm{I} + (\mathrm{I} + \mathrm{H}) + \cdots + (\mathrm{H}^{M-1} + \mathrm{H}^M) \right) 2\overline{u}_D(\mathbf{x}) \\ &= \left(\mathrm{I} + \cdots + \mathrm{H}^{M-1} + \frac{1}{2} \mathrm{H}^M \right) 2\overline{u}_D(\mathbf{x}) \end{aligned} \tag{25}$$

Therefore, the term for the last point $\mathbf{x}_M$ should now be multiplied by $1/2$, when compared to just truncating the original series at the M-th term. For such fixed-length truncation, the resulting estimator is biased, just like MC rendering with a finite path length. One could potentially apply the Russian roulette technique to truncate a path without the bias, though we leave it as future work.

Formally, this transformation of a Neumann series can be mathematically interpreted as an analytic continuation of the series, and many other transformations are possible [Sabelfeld 1991; Sabelfeld and Simonov 1994]. Sabelfeld [1991] numerically compares some transformations, and our initial experiments also suggest that series acceleration, such as the van Wijngaarden transformation [van Wijngaarden 1953], can reduce the error of the estimator based on Neumann series. All the results we show use the modified Neumann series with multiplication of $1/2$ on the last term.

4.1.3 Derivative and boundary value estimators. With WoB, in addition to the solution within the domain, we can easily estimate the gradient inside the domain, solution on the boundary, and normal derivative on the boundary, just by replacing the first step of our recursive MC solution estimators. For example, we can apply an MC estimator based on the equation for the interior gradient, $\frac{\partial u}{\partial \mathbf{x}_k} = -\int \frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}} v \, dA$, instead of Eq. (18) to get a gradient estimate in the case of the Dirichlet problem estimator with double layer potential formulation. Similar to WoS [Sawhney and Crane 2020], we can reuse the same paths to get samples for the solution and the gradient in the interior by changing the initial weight at almost no additional cost. We however observed increased noise near the boundary in the gradient estimates for this Dirichlet problem estimator. This additional noise is likely due to the presence of the hypersingular kernel $\frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}}$ in the computation, which has a very large variance when the interior point $\mathbf{x}$ is placed very close to a sampled boundary point $\mathbf{y}$. To overcome this issue, we could separate the gradient into the normal and tangential components defined with respect to the nearest boundary point and carefully evaluate each term as described by Sabelfeld and Simonov [1994].

The estimates of the normal derivatives on the boundary with WoB use the equations we get by taking the limit of the integral equations for gradient estimation to the boundary. For this Dirichlet estimator, however, the normal derivative estimator derived this way involves a hypersingular integral, which has an infinite variance and cannot be used directly; we will need to transform it to another form for evaluation [Sabelfeld and Simonov 1994].

4.2 Neumann Problems with Direct BIE

The strength of WoB is that we can apply essentially the same approach of building a recursive MC estimator to address other boundary problems than Dirichlet problems. While Sabelfeld and Simonov [1994] proposed using single layer potentials to solve Neumann problems, we propose another formulation based on direct BIEs since this formulation allows us to utilize a different estimator than the one with single layer potentials as we will explain later. Our formulation uses Eq. (6) in combination with the Neumann boundary conditions $\overline{q}_N$ as

$$u(\mathbf{x}) = -\int_\Gamma 2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y})u(\mathbf{y})\,dA_\mathbf{y} + \int_\Gamma 2G(\mathbf{x},\mathbf{y})\overline{q}_N(\mathbf{y})\,dA_\mathbf{y}, \quad (26)$$

at a point $\mathbf{x}$ on the boundary; the second term can be estimated with another MC estimator without recursion because $\overline{q}_N$ is a known boundary value. We have found that it usually suffices to sample one boundary point to estimate the integral of the second term per recursion, though it is possible to have more samples.

The rest is similar to the Dirichlet case described above — we expand u recursively. We can estimate the unknown $u(\mathbf{x}_i)$ as

$$\hat{u}(\mathbf{x}_i) := -\frac{2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_i,\mathbf{x}_{i+1})}{p_1(\mathbf{x}_{i+1}|\mathbf{x}_i)}\hat{u}(\mathbf{x}_{i+1}) + \frac{2G(\mathbf{x}_i,\mathbf{x}'_{i+1})}{p_2(\mathbf{x}'_{i+1}|\mathbf{x}_i)}\overline{q}_N(\mathbf{x}'_{i+1}). \quad (27)$$

In general, the two PDFs p_1 and p_2 can differ, sampling two distinct points $\mathbf{x}_{i+1}$ and $\mathbf{x}'_{i+1}$ based on the current point $\mathbf{x}_i$. This estimator can be used to estimate the interior value based on Eq. (6) with

$$\hat{u}(\mathbf{x}_0) := -\frac{\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}_0,\mathbf{x}_1)}{p_1(\mathbf{x}_1|\mathbf{x}_0)}\hat{u}(\mathbf{x}_1) + \frac{G(\mathbf{x}_0,\mathbf{x}'_1)}{p_2(\mathbf{x}'_1|\mathbf{x}_0)}\overline{q}_N(\mathbf{x}'_1), \quad (28)$$

where $\mathbf{x}_0$ is an interior point and $\mathbf{x}_1$ and $\mathbf{x}'_1$ are boundary points. Estimating the gradient is also possible but with a high variance similar to the case of the Dirichlet problem estimator.

4.3 Mixed Boundary Problems with Single Layer BIE

For mixed boundary and Robin problems, we adopt a single layer potential formulation (Eq. (8)) where the boundary unknown we need to estimate is μ. This formulation was used for pure Dirichlet, Neumann, and Robin problems by Sabelfeld and Simonov [1994], but not for mixed boundary problems. This formulation leads to a Fredholm equation of the second kind for parts of the boundary where Neumann or Robin conditions are specified (see Table 1 for the equations). However, for parts of the boundary where Dirichlet boundary conditions are specified, this formulation would result in an equation in the form of a Fredholm equation of the *first* kind:

$$\overline{u}_D(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{y})\mu(\mathbf{y})\,dA_\mathbf{y} \quad \text{for} \quad \mathbf{x} \in \Gamma_D. \quad (29)$$

Unlike the second kind equation, the unknown quantity μ appears only inside the integral. One cannot simply apply recursive MC estimation for equations in the form of the first kind equation since there is no recursion. A common approach is to discretize and solve the corresponding matrix equation, which ruins the advantages of WoB over the conventional alternatives.

Inspired by a similar technique by Sabelfeld and Simonov [1994], we propose to transform Eq. (29) by multiplying by a nonzero constant k on both sides and adding $\mu(\mathbf{x})$ to both sides:

$$\mu(\mathbf{x}) = \mu(\mathbf{x}) - k\int_\Gamma G(\mathbf{x},\mathbf{y})\mu(\mathbf{y})\,dA_\mathbf{y} + k\overline{u}_D(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Gamma. \quad (30)$$

This equation now has a structure similar to the second kind equation, with the additional $\mu(\mathbf{x})$ on the right-hand side. This equation can be estimated recursively, just like the second kind equation. We estimate the unknown quantity on the left-hand side by sampling the next point, and estimate the contribution by

$$\hat{\mu}(\mathbf{x}_i) := \begin{cases} -\frac{1}{p_k}\cdot k\frac{G(\mathbf{x}_i,\mathbf{x}_{i+1})}{p(\mathbf{x}_{i+1}|\mathbf{x}_i)}\hat{\mu}(\mathbf{x}_{i+1}) + k\overline{u}_D(\mathbf{x}_i) & \text{with prob. } p_k \\ \frac{1}{1-p_k}\hat{\mu}(\mathbf{x}_{i+1}) + k\overline{u}_D(\mathbf{x}_i) & \text{with prob. } 1-p_k. \end{cases} \quad (31)$$

We sample one of the first two terms in Eq. (30) based on the given probability p_k. In the second case when we sample the term $\mu(\mathbf{x})$, we remain at the same point, i.e., $\mathbf{x}_{i+1} := \mathbf{x}_i$, in the next recursion step. With this method of handling the first kind equation, we can construct a recursive MC estimator for μ for mixed boundary problems to recover the solution by another MC integration of Eq. (8).

The choice of the multiplication constant k is critical in this estimator, and we picked a value by trial and error. If it is too small, the bias is higher given the same path length. If it is too large, the Neumann series diverges and the estimator fails as justified by [Sabelfeld and Simonov 1994] in the case of pure Dirichlet problems. Sabelfeld and Simonov [1994] also discuss the restriction on the mixture weight $\overline{\alpha}$ of Robin boundary problems. We found that deriving theoretical bounds in the case of mixed boundaries are extremely involved and are left for future work.

The gradient estimator with this formulation does not exhibit the additional noise as with the other formulations because this formulation uses an integral kernel with a lower order of singularity. For the normal derivative estimator, we get μ in two terms, in the integral and outside of the integral. We need to sample either one of the terms or run two paths for the estimates of μ for the two terms.

One could alternatively employ direct BIEs to get a similar estimator for mixed boundary problems. However, we would still encounter a first kind equation, and in practice the resulting schemes are not much different from the indirect BIE-based method above.

4.4 Sampling Strategies

WoB can use various strategies to sample paths as in MC ray tracing, and a well-designed strategy can sharply reduce variance of the estimator. Both WoB and MC ray tracing have a vast design space of sampling strategies for different problems, and our proposed strategies in this paper are by no means exhaustive. We leave further exploration of different strategies as future work and briefly explain the strategies we implemented in the following.

4.4.1 Ray sampling. The integral kernel $2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}$ is in fact proportional to the differential solid angle of $\mathbf{y}$ from another point $\mathbf{x}$. Similar to the fact that sampling a ray will cancel out the geometry term in rendering, we can use ray tracing from $\mathbf{x}$ to perfectly importance sample $2\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}$ at any $\mathbf{y}$. The main difference from rendering is that we do not have the visibility term between $\mathbf{y}$ and $\mathbf{x}$. We thus have to sample a ray from a sphere, not hemisphere, and sample all the intersection points (not just the first hit) along the ray from $\mathbf{x}_i$ via "all-hits" ray intersection queries in general. Such all-hits queries are available and used in ray tracing [Gribble et al. 2014].

Due to our recursive formulation, using all hits would cause exponential branching of paths which may not be ideal for GPU ray tracing [Parker et al. 2010]. We can instead pick one intersection out of the m such intersections at random, which results in multiplying the PDF by $1/m$, and the sample contribution is thus multiplied by m but with no branching in recursion. We used this approach in all of our results in this paper. Since this approach leads to an exponential increase in variance instead of exponential computation cost per sample, we do not claim that this strategy is always better than using all hits and branching exponentially.

For the special case when the domain is convex, we revert to the original hemispherical sampling strategy with closest-hit query and the PDF in this strategy simplifies to $p(\mathbf{y}|\mathbf{x}) = -\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x}, \mathbf{y})$ because there is only one hit for a direction within the hemisphere. The solution estimator for the Dirichlet problem with path length M then becomes $\hat{u}(\mathbf{x}_0) = 2[u_D(\mathbf{x}_1) - u_D(\mathbf{x}_2) + \cdots] + (-1)^M u_D(\mathbf{x}_M)$. This estimator is the one introduced in Section 2 (and Algorithm 1).

Another strategy is to use only the first hit point, but combine it with direct sampling of a point on the boundary (like sampling a point on light sources in rendering) via multiple importance sampling (MIS) [Veach and Guibas 1995]. While the first-hit-only strategy would not cover the entire integration domain, MIS ensures an unbiased estimator since direct sampling of a point will cover the entire domain. We did not employ this strategy since it often had higher variance in our experiments. However, similar to different strategies in MC rendering, there might be certain scenarios where this particular strategy works better than the others. We suggest that readers explore different options to determine a suitable strategy for a given problem.

4.4.2 Backward estimator. We have so far considered tracing a path by sampling a series of points starting from the evaluation point $\mathbf{x}_0$. We call this estimator a "backward" estimator, consistent with "backward tracing" in rendering which traces a path of light in a backward manner from the sensor (pixel) all the way to the light source. One can think of our evaluation point as a pixel and a boundary as a light source. Note that Qi et al. [2022] adopted the opposite definitions of backward and forward from those in rendering, so their forward estimator corresponds to our backward estimator.

4.4.3 Forward estimator. Sabelfeld and Simonov [1994] proposed an *adjoint estimator* in WoB, which forms paths starting from a point on the boundary. We have found that it is analogous to light tracing in rendering, which traces a path starting from a point on the light source and can also be explained via the adjoint of the rendering equation [Christensen 2003]. We call it a "forward estimator" as in "forward tracing" in rendering. In the forward estimator, we will need to make an explicit connection between each point along the path and the evaluation point to compute the contribution of a path to the evaluation point. The forward estimator has less variance than the backward estimator in some of the formulations where the integral kernel is proportional to $\frac{\partial G}{\partial \mathbf{n}_\mathbf{x}}$ as opposed to $\frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}$. In this case, we want to generate a ray from $\mathbf{y}$ to sample a point $\mathbf{x}$ proportional to its differential solid angle. For Neumann problems, the formulation with single layer potentials [Sabelfeld and Simonov 1994] is easier to importance sample by a forward estimator, while our formulation with the direct BIE matches better with a backward estimator.

4.4.4 Other strategies. Importance sampling only the integral kernel does not perfectly importance sample all the terms, which is also true in rendering. For example, the Dirichlet boundary value $\overline{u}_D$ will not be importance sampled that way. Because our WoB is based on a Fredholm equation of the second kind and we can use ray tracing, it is easy to apply more advanced sampling techniques used in rendering, such as MIS, resampled importance sampling (RIS) [Talbot et al. 2005], Markov chain MC (MCMC) [Kelemen et al. 2002; Veach and Guibas 1997], or the zero variance theory [Křivánek and d'Eon 2014] to implement WoB. It contrasts with WoS where applications of these techniques are not necessarily straightforward (e.g., an MIS bidirectional estimator is not available in WoS [Qi et al. 2022]). We show preliminary results with bidirectional estimators with MIS,

resampling via RIS, MCMC WoB, and path reuse as in the virtual point lights method [Keller 1997], but many other techniques can be made available to WoB. One interesting aspect of WoB that might lead to further development of sampling strategies is that samples can have positive or negative contributions, unlike rendering where all contributions are non-negative.

4.5 Generalization

4.5.1 Exterior problems. WoB efficiently handles exterior problems in its basic form. Instead of the domain Ω, we can solve Laplace's equation in $\mathbb{R}^2\backslash\Omega$ or $\mathbb{R}^3\backslash\Omega$ for exterior problems. We define the normals to remain oriented outward from the interior domain Ω and replace the definitions of the boundary values with those obtained by taking the limit from the exterior domain. Moreover, in addition to the boundary conditions, we require that the solution $u(\mathbf{x})$ approaches zero at infinity for exterior problems. The solvers for the exterior domain largely remain the same as for the interior domain, except for a few sign changes in the terms of the BIEs (see Table 1). Since WoB relies on neither ϵ-shell approximation nor closest point queries in WoS, its accuracy and performance for exterior domains remain the same as for interior problems. For example, WoB does not need Kelvin transformations [Nabizadeh et al. 2021].

Of the two Dirichlet problem estimators, the double layer potential formulation estimator requires additional attention when used for exterior problems. This formulation allows us to find solutions that decay according to $O(|\mathbf{x}|^{1-d})$ as $|\mathbf{x}| \to \infty$, where d is the dimension of the problem. It thus cannot handle more general cases where solutions decay as $O(|\mathbf{x}|^{2-d})$. Sabelfeld and Simonov [1994] explain how to generalize the double layer potential formulation for such cases. The basic idea is to reduce the problem to one without the slowly decaying component with some precomputation before applying WoB. We enabled this extension for the scenes in Figures 7 and 11 where we expect the decay rate $O(|\mathbf{x}|^{2-d})$, while we did not do so for the other examples for which we use analytical solutions with decay rate $O(|\mathbf{x}|^{1-d})$. The single layer potential formulation, on the other hand, supports solutions with a decay rate $O(|\mathbf{x}|^{2-d})$ without modification. The exterior problem estimators for Neumann, Robin, and mixed boundary problems based on the single layer potential formulation are also general enough to handle the general decay rate of harmonic functions.

4.5.2 Multiply-connected domain problems. We mainly focus on simply connected domains, e.g., domains without holes inside, for simplicity, with the exception of the interior Neumann problem in Fig. 9. Multiply connected domains require some additional considerations [Sabelfeld and Simonov 1994]. For Neumann problems with single layer potential, we can apply the same WoB estimators without modifications as long as the standard compatibility condition that the integral of the normal derivative evaluates to zero for each connected domain is satisfied. For Dirichlet (with double layer potential) and Robin problems (with single layer potential), the situation is more complicated. The applicability of WoB in its original form is guaranteed by assuming more artificial compatibility conditions in these cases, and we need to perform some precomputation to modify the problem similarly to the exterior Dirichlet problem estimator. We are yet to confirm the applicability and efficiency of these estimators by Sabelfeld and Simonov [1994] or to derive solvability conditions for Neumann problem estimators based on direct BIE formulation. For Dirichlet problems, we observed some successful applications of the single layer potential formulation with multiply-connected domains, but the restriction on the multiplicative constant k seems more strict, and it too requires further investigation.

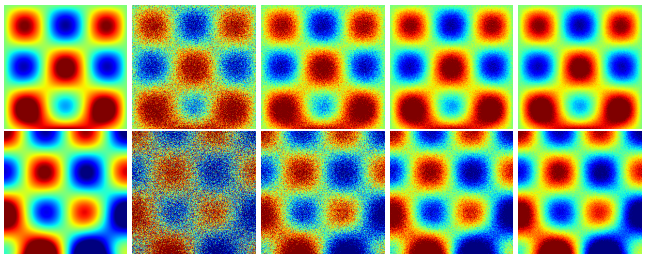


Fig. 4. Estimates for Poisson's equation for Dirichlet (top) and Neumann (bottom) problems. WoB can handle the non-zero source term. For each sample path, we used 16 volume samples to estimate all domain integrals.

4.5.3 Non-zero source term. For Poisson's equation (i.e., $\overline{b}(\mathbf{x}) \neq 0$), we need to additionally sample an interior (or exterior) point to have an MC estimate for the volume integral of V_0. We include this term when we retrieve the boundary values $v(\mathbf{x})$ or $\frac{\partial v}{\partial \mathbf{n}}(\mathbf{x})$ in Laplace's equation and when we compute the function V_0 in the relation $u(\mathbf{x}) = v(\mathbf{x}) - V_0(\mathbf{x})$. One strategy is to sample such a point uniformly within the domain, but a well-designed sampling strategy can reduce variance. For interior problems, one possible sampling strategy is to sample a point along a line that goes through the point $\mathbf{x}$. Another possibility is to sample the point depending on the distribution of the source term. Fig 4 shows examples for Dirichlet and Neumann problems by sampling interior points uniformly in terms of the area measure. We focus on Laplace's equation in the other results since the volume integral of V_0 can always be added trivially. Sawhney et al. [2022] dismissed WoB as being formulated only for Laplace's equation, but as we show, WoB is certainly capable of handling Poisson's equation.

5 RESULTS

Generality of WoB. The same WoB framework based on boundary integral equations successfully handles Dirichlet, Neumann, Robin, and mixed boundary problems, for either interior or exterior domains. Figures 1 and 5 show the results of WoB for interior and exterior Laplace problems with known analytical solutions, where we set the boundary conditions to satisfy the known analytical solution. For Robin boundaries, we use a constant value $\overline{\alpha} = 1$ for the mix weight in these examples. For the mixed boundary problems, we uniformly randomly assign one of the Dirichlet, Neumann, or Robin boundary conditions to each boundary triangle or line segment, and use $k = 4$ and $p_k = 2/3$. We show results using the sampling strategies described earlier. For Robin and mixed boundary problems, a pure ray sampling strategy without MIS would miss nonzero contributions, so we instead use RIS with candidates

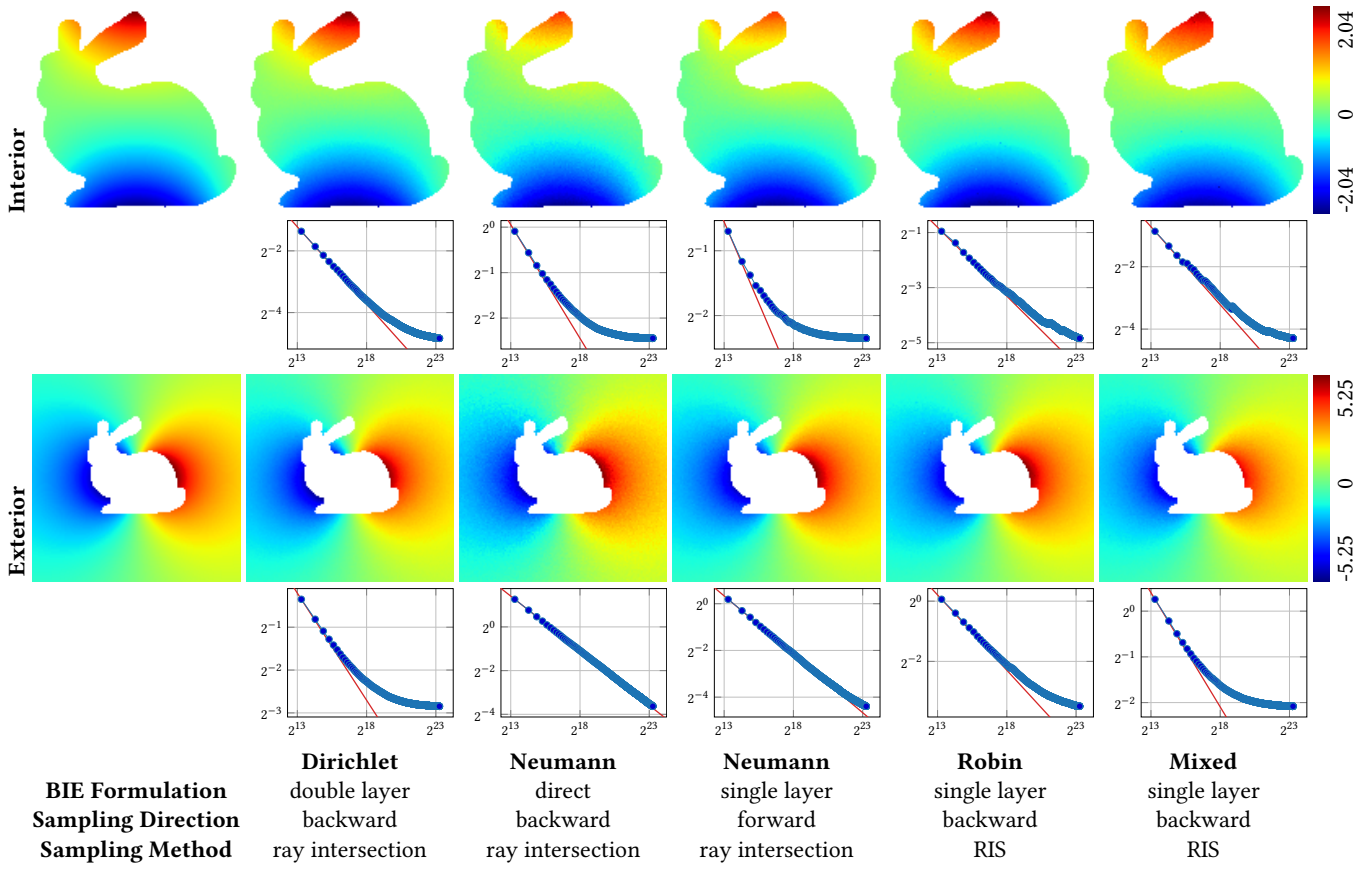
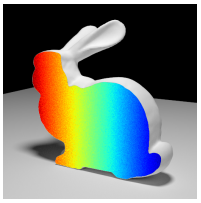


Fig. 5. WoB applied to various problems. For the interior (top) and exterior (bottom) problems, we run WoB with path length $M = 4$ and $N = 10^7$ samples per evaluation point with the formulations and sampling techniques as labeled. The solution estimates are visualized with a color map. For each problem, we show the absolute root mean square error (vertical axis, varying scales) with respect to the number of samples (horizontal axis) measured against the reference analytical solution (left). The red lines show the $O(1/\sqrt{N})$ decay rate for reference.

generated uniformly over the boundary to approximately sample the integral kernel. In Fig. 5, the number of candidates is 16. All results in Fig. 5 are generated with path length $M = 4$ and sample path count per evaluation point $N = 10^7$ for consistency, using a highly parallelized CUDA implementation. We observe that *all* of the results are consistent with the analytical solution as expected.

Convergence rate and truncation error. WoB exhibits the expected MC convergence rate of $O(1/\sqrt{N})$. The RMSE curve can eventually become flat when we take enough samples because of the error introduced by path truncation. This error decreases as we increase the path length although that introduces larger variance to the estimator. For the results in Fig. 5, we observe in particular a relatively large truncation error for the two interior Neumann problem estimators and the exterior Dirichlet and mixed boundary problem estimators. We thus show their results with different path lengths in Fig. 6. As expected, we observe that having longer paths decreases the truncation error at the cost of having a higher variance, requiring a larger sample count to converge. It might be possible to apply stochastic truncation as in MC rendering [Misso et al. 2022] to avoid this error, though a careful investigation is needed to handle subtle differences between WoB and MC rendering.

WoB within MC rendering. Fig. 7 show the results of WoB implemented within our MC ray tracing system. Our implementation of WoB utilizes the existing functionalities of MC ray tracing such as ray-object intersection, stochastic sampling, and textures (for boundary values). The code for WoB itself is roughly 100 lines and can support both interior and exterior problems seamlessly. Visualization was done by simultaneously running rendering with ambient occlusion using the MC ray tracing system. The inset figure shows a result generated with our prototype interior Dirichlet solver on top of a popular open-source renderer for research purposes, PBRT [Pharr et al. 2018], with minimal modifications. For this problem, we set the boundary values such that we expect to see a linear horizontal gradient of solution values mapped to colors ranging from red to green to blue.



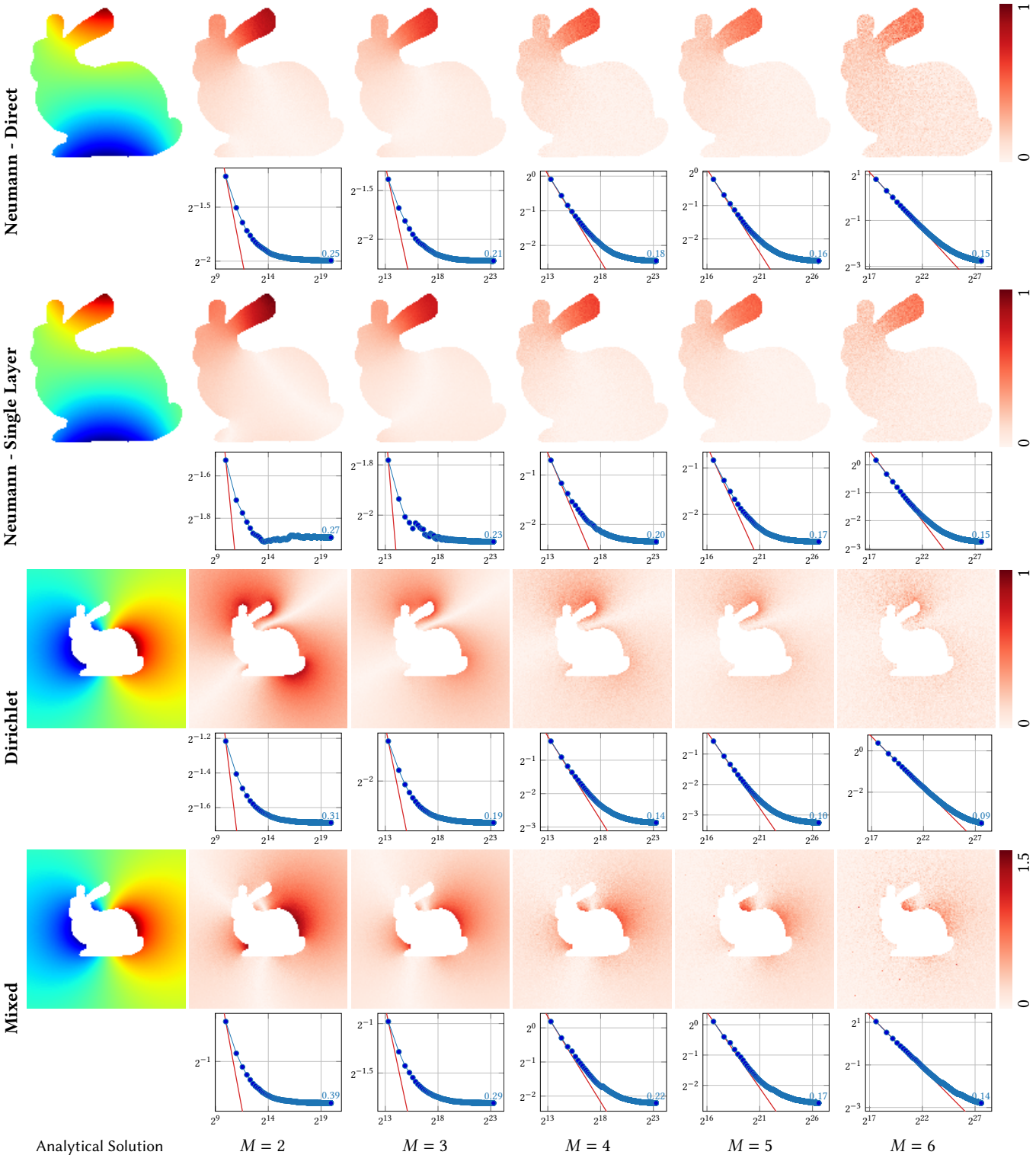


Fig. 6. Path truncation error study. For the interior Neumann problem estimators and the exterior Dirichlet and mixed boundary problem estimators in Fig. 5, we show the absolute errors with different path lengths M. We show the absolute root mean square error (vertical axis) with respect to the number of samples (horizontal axis), and also show the remaining error in blue text. As path length increases the bias decreases, but larger numbers of samples are needed to achieve convergence.

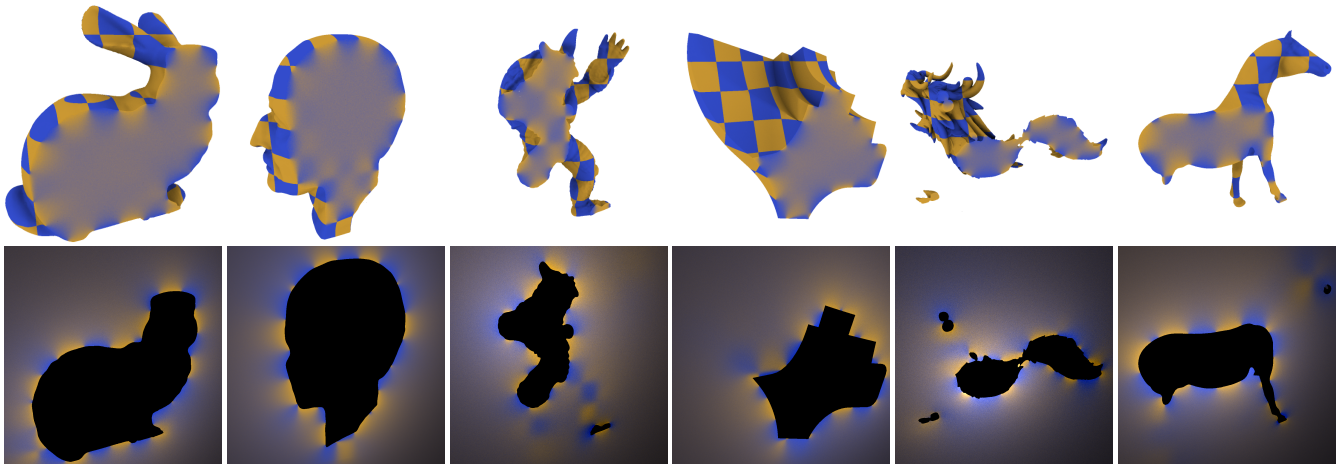
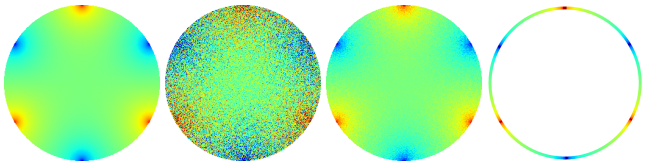


Fig. 7. Results of a WoB solver implemented on top of an MC ray tracing system. WoB's strong similarity to MC ray tracing makes such an implementation easy to carry out. The images show the estimated solution for interior (top) and exterior (bottom) Dirichlet problems on a cutting plane. WoB can solve both problems efficiently with a unified MC ray tracing solver. Ambient occlusion was computed at the same time as the solution using the same rendering system.



(a) reference (b) uniform sampling (c) boundary sampling (d) estimate on boundary

Fig. 8. Neumann solver with single layer formulation. The boundary value $\partial u/\partial \mathbf{n} = 0$ except for around the six points. By sampling the start of paths according to the distribution of the boundary value (c) in our forward estimator, we get much less noise compared to the one with uniform sampling (b). (a) shows the reference estimate we get with a high sample count. (d) shows the estimate of the solution exactly on the boundary.

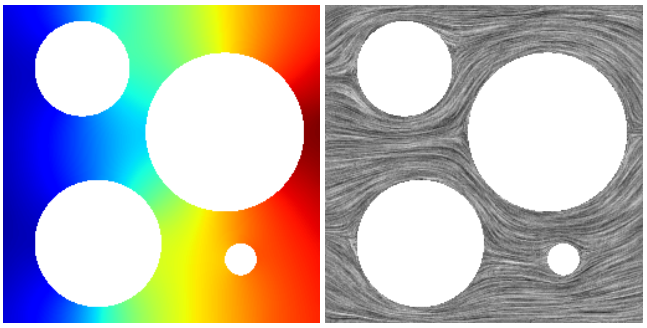


Fig. 9. Potential flow reconstruction from the velocity boundary condition. We show the estimated potential field on the left and the estimated velocity field on the right. On the outer boundary, a constant inflow boundary condition $\partial u/\partial \mathbf{n} = -1$ is given on the left edge, and a constant outflow boundary condition $\partial u/\partial \mathbf{n} = 1$ is given on the right edge. On the other outer boundary edges and the inner boundaries, $\partial u/\partial \mathbf{n} = 0$ is given.

Boundary sampling and estimation on the boundary. WoB for Neumann problems offers two distinct advantages over WoS, as demonstrated in Fig. 8. First, we can begin the walk-on-boundary process by sampling a starting point on the boundary, according to the magnitude of the boundary value, to design an efficient sampling strategy when the boundary value is specified sparsely. With WoS, such a strategy is available only for Dirichlet problems [Qi et al. 2022]. Second, we can exploit the BIE formulation to estimate the solution value exactly on the boundary, with a very minor modification to the underlying BIE. The original WoS and its basic extensions for Neumann problems that rely on epsilon shell termination criteria are incapable of estimating the solution or the normal derivative exactly on the boundary or exhibit significant bias in such cases.

Gradient estimator and a path reuse strategy. Similarly to WoS, WoB can estimate the potential and the gradient simultaneously with almost no additional cost. Fig. 9 demonstrates the use of WoB for interpolating potential flow velocities, as described by Nielsen and Bridson [2011]. By solving for the gradient of the scalar potential u, which satisfies the Laplace equation and a prescribed Neumann boundary condition, we obtain the velocity field that follows potential flow assumptions (incompressibility and irrotationality) and matches the inflow/outflow conditions at the boundaries. In this example, we use a backward estimator for the single layer formulation with RIS, in combination with a path reuse strategy analogous to the virtual point lights method [Keller 1997] in rendering; we generate sample paths from the boundaries to get samples for the unknown boundary value estimates first, and connect each of them to all the evaluation points to reuse the subpaths starting from the boundaries. This effectively increases the number of sample paths per evaluation point while introducing correlation of the estimates at different evaluation points. Though this correlation may seem undesirable, this approach guarantees that the estimated solution and gradient fields are always smooth, making it a potentially preferable alternative in some settings like fluid simulation applications. Miller et al. [2023] introduced a similar boundary value caching technique

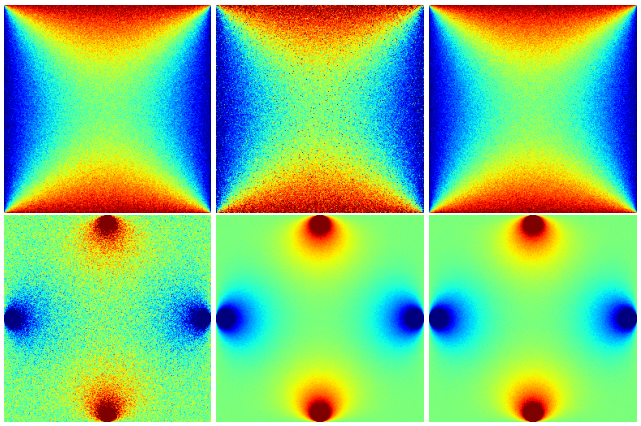


Fig. 10. Equal-time comparison of bidirectional WoB estimators. Left to right: purely backward estimator, next-event estimation, and the MIS combination of the two. Top row: non-zero boundary values on each side. Bottom row: non-zero boundary values around the center of each side. Either the backward estimator or next-event estimation is more efficient than the other in each setting. WoB allows us to trivially combine the two estimators via MIS, and the combined estimator is robust across different settings (right).

for WoS in concurrent work, and their analysis largely applies to the case of WoB as well.

MIS estimator. Due to the similarity between WoB and MC rendering, it is trivial to combine WoB estimators via MIS. Fig. 10 compares the backward estimator, the backward estimator with importance sampling of boundary values, and the MIS combination of the two estimators with the balance heuristic for a Dirichlet problem. The two estimators correspond to unidirectional path tracing and path tracing with next-event estimation in MC ray tracing. When the non-zero boundary values are not localized (top row), the backward estimator performs well, although boundary sampling (which corresponds to next-event estimation) suffers from additional noise due to its explicit connection to a boundary point. Conversely, when the non-zero boundary values are localized (bottom row), boundary sampling becomes significantly more efficient than the backward estimator. This behavior is analogous to unidirectional path tracing and next-event estimation for cases where light sources are large or small. The MIS combination of the two estimators (right column) is robust across different settings. The related work for WoS [Qi et al. 2022] left this MIS combination to future work, as the integration domain changes at each step in WoS (i.e., WoS solves a Volterra equation; see Appendix B). WoB allows us to incorporate MIS since it has the same mathematical and algorithmic structure as MC ray tracing. Notably, our bidirectional estimators for WoB also do not introduce extra bias, unlike their WoS counterparts [Qi et al. 2022].

Markov chain Monte Carlo. We implemented primary sample space MLT [Kelemen et al. 2002] (PSSMLT) on top of WoB interior/exterior Dirichlet estimators. In this implementation, each sample in WoB is generated according to a Markov chain over the sampling domain including the image space. PSSMLT formulates this sampling domain as a unit hypercube of random numbers used

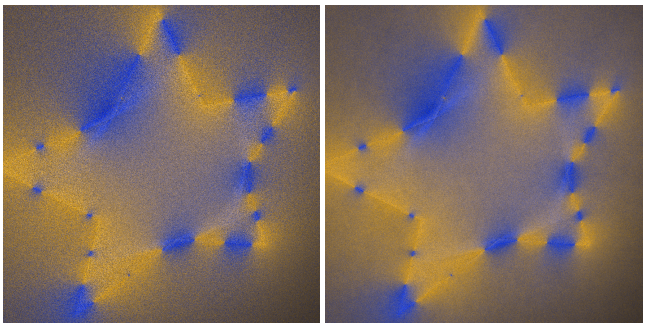


Fig. 11. Equal-time comparison of Monte Carlo and Markov chain Monte Carlo (MCMC) estimators for solving interior and exterior Dirichlet problems on a plane passing through a concave 3D star shape. Left image: Monte Carlo sampling. Right image: Markov chain Monte Carlo sampling formulated as PSSMLT [Kelemen et al. 2002]. MCMC can result in lower variance than MC as in rendering.

to generate each sample and the location in the image space. In our case, the first two dimensions are used to pick a pixel (i.e., evaluation point) and the rest of the dimensions are used for generating a sample in the corresponding WoB estimator (i.e., interior or exterior, depending on the pixel), just like PSSMLT in rendering. We set the target distribution to be the absolute value of the sample's contribution, so samples that have higher absolute contributions are likely to be generated more often. Since this target distribution includes all the terms in each sample (e.g., the boundary condition and the probability of selection in all-hits), MCMC is expected to perform better than MC with a limited form of importance sampling. Fig. 11 shows our preliminary examples that compare MC and MCMC in equal time. These examples solve interior and exterior Dirichlet problems at the same time. We observe that PSSMLT performs similarly in rendering and WoB in the sense that the image is less noisy than MC at the cost of correlation artifacts. This application of MCMC is straightforward due to the similarity between rendering and WoB.

Numerical comparisons between WoB and WoS. Fig. 12 compares the efficiency of WoB and WoS interior Dirichlet estimators in examples of convex and non-convex domains. Both WoB and WoS are implemented with similarly optimized CUDA code in their basic forms with no advanced sampling techniques. Our experiments suggest that WoS performs more efficiently than WoB for complex non-convex domains, but comparably for simple convex domains. The reason is that, while WoS typically takes more computation time per sample (i.e., per individual random walk that terminates near the boundary), WoB requires more sample paths due to its higher variance, resulting in lower overall efficiency. We expect that this difference may become smaller with additional variance reduction techniques. Moreover, this observation only provides general guidance on the choice between WoS and WoB. We do not claim that one of them is fundamentally more efficient than the other in any given problem (be it convex or non-convex) based on these numerical comparisons; the theoretical comparisons offered throughout this paper are more generally relevant (e.g., accuracy near the boundary and generality regarding supported problems).

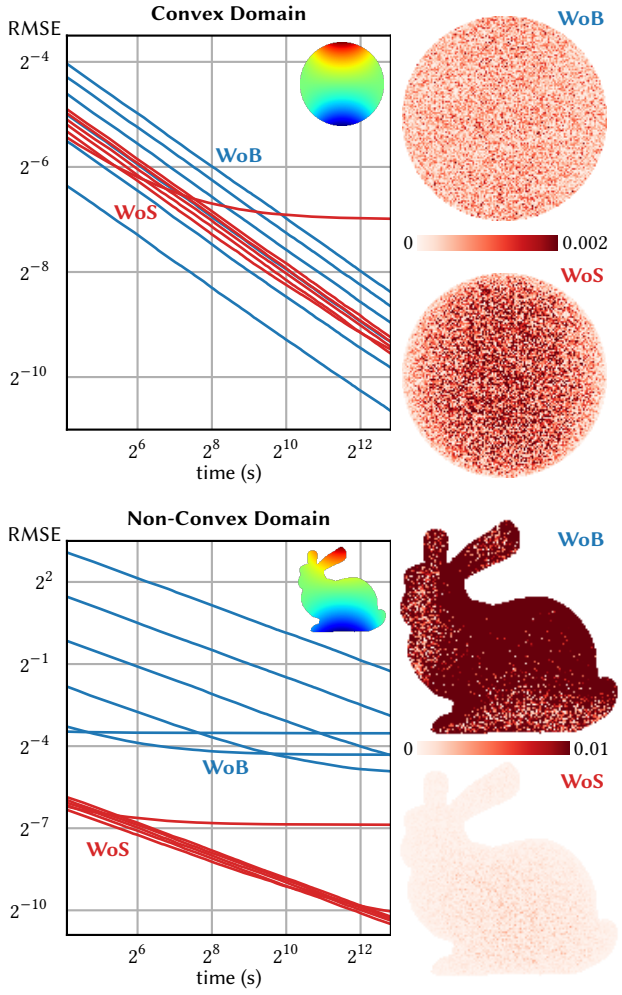


Fig. 12. Comparison of WoB and WoS with example interior Dirichlet problems in a convex domain (top) and a non-convex domain (bottom). The plots show how the root mean squared errors (vertical axis) decay with increasing time (horizontal axis). Each line corresponds to a specific parameter choice for WoB (blue) and WoS (red). We use path length $M = 2$ to 7 for WoB, and epsilon shell size 10^{-2} to 10^{-7} for WoS. For each scene and for each method, we show the error of the solution with the least error after 2 hours on the right. We observe that WoS performs more efficiently than WoB for complex non-convex domains.

6 CONCLUSION

WoB is a promising alternative to WoS as an MC estimator for boundary value problems. WoB offers a unified way to solve interior and exterior Dirichlet, Neumann, Robin, and mixed boundary problems. WoB is the only method that can estimate solutions near or on the boundary without any spatial discretization error (as in finite element and finite difference methods) or ϵ-shell error (as in WoS). The remarkable similarity of WoB to rendering allowed us to apply advanced MC rendering methods directly and to implement WoB atop existing ray tracing codebases.

Our paper has only scratched the surface of WoB's full potential. Similar to WoS, WoB can be applied to other elliptic equations with known fundamental solutions by replacing the fundamental solution. For example, the book by Sabelfeld and Simonov [1994] describes the application of WoB to other equations, including screened Poisson, linear elastostatics, and diffusion problems. We introduced WoB to graphics for the first time, but the use of BIEs has been explored in the form of conventional spatial discretization-based methods, including the boundary element method [Da et al. 2016; Hahn and Wojtan 2015; James et al. 2006; James and Pai 1999; Solomon et al. 2017; Sugimoto et al. 2022] and the method of fundamental solutions [Martin et al. 2008]. This relationship between WoB and these methods is analogous to that between rendering techniques based on Monte Carlo methods and the radiosity method [Cohen and Greenberg 1985; Goral et al. 1984; Nishita and Nakamae 1985] for the rendering equation. These traditional BIE-based techniques also suggest other potential application domains of WoB. The application to problems with spatially-varying coefficients [Sawhney et al. 2022] is another interesting extension.

The generality of WoB in terms of the types of domains and the valid ranges of parameters (e.g., first kind equation scaling factor k and Robin problem mixture weight $\overline{\alpha}$) require further investigation. We focused on simply connected domains for simplicity, with the exception of the interior Neumann problem in Fig. 9. Extensions to support other multiply-connected domain problems are explained by Sabelfeld and Simonov [1994], but we are yet to validate their formulation and evaluate their numerical performance.

The bias of the estimator is still another topic. Instead of truncating the paths with a predefined length, we expect that other path truncation techniques, such as Russian roulette, could be applied, but with a careful investigation of how this should be done considering the modification to the Neumann series.

Lastly, we are greatly interested in improving the efficiency of the method. While WoB applies to more problems than WoS, we have not yet thoroughly evaluated the performance of WoB and the variants of WoS, where available. However, we foresee that future applications of advanced rendering methods, such as the UPS/VCM methods [Georgiev et al. 2012; Hachisuka et al. 2012], together with techniques tailored for WoB, such as the use of a more efficient modified Neumann series and more efficient handling of all-hits intersections due to contributions between mutually invisible points, can make WoB an even more attractive choice.

ACKNOWLEDGMENTS

This research was partially funded by NSERC Discovery Grants (RGPIN-2021-02524 & RGPIN-2020-03918) and a grant from Autodesk. The second author was supported by the NSERC USRA program and the URF program at the University of Waterloo. This research was enabled in part by support provided by SHARCNET and the Digital Research Alliance of Canada. The 3D models in this paper are courtesy of CyberWare, Max Planck Institute, Keenan Crane, Pratt & Whitney/Hugues Hoppe, and Stanford University. We would like to thank the anonymous reviewers for their constructive evaluations and feedback.

REFERENCES

John C. Bowers, Jonathan Leahey, and Rui Wang. 2011. A Ray Tracing Approach to Diffusion Curves. *Comput. Graph. Forum* 30, 4 (2011), 1345–1352. https://doi.org/10.1111/j.1467-8659.2011.01994.x

Per H. Christensen. 2003. Adjoints and Importance in Rendering: An Overview. *IEEE Trans. Vis. Comput. Graph.* 9, 3 (jun 2003), 329–340. https://doi.org/10.1109/TVCG.2003.1207441

David L. Clements. 2004. A fundamental solution for linear second-order elliptic systems with variable coefficients. *Journal of Engineering Mathematics* 49, 3 (jul 2004), 209–216. https://doi.org/10.1023/B:ENGI.0000031193.39633.d8

Michael F. Cohen and Donald P. Greenberg. 1985. The Hemi-Cube: A Radiosity Solution for Complex Environments. *SIGGRAPH Comput. Graph.* 19, 3 (jul 1985), 31–40. https://doi.org/10.1145/325165.325171

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Trans. Graph.* 35, 4, Article 78 (jul 2016), 12 pages. https://doi.org/10.1145/2897824.2925899

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans. Graph.* 31, 6, Article 192 (nov 2012), 10 pages. https://doi.org/10.1145/2366145.2366211

Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. 1984. Modeling the Interaction of Light between Diffuse Surfaces. *SIGGRAPH Comput. Graph.* 18, 3 (jan 1984), 213–222. https://doi.org/10.1145/964965.808601

Christiaan Gribble, Alexis Naveros, and Ethan Kerzner. 2014. Multi-Hit Ray Traversal. *Journal of Computer Graphics Techniques* 3, 1 (feb 2014), 1–17. http://jcgt.org/published/0003/01/01/

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. *ACM Trans. Graph.* 31, 6, Article 191 (nov 2012), 10 pages. https://doi.org/10.1145/2366145.2366210

David Hahn and Chris Wojtan. 2015. High-Resolution Brittle Fracture Simulation with Boundary Elements. *ACM Trans. Graph.* 34, 4, Article 151 (jul 2015), 12 pages. https://doi.org/10.1145/2766896

Doug L. James, Jernej Barbič, and Dinesh K. Pai. 2006. Precomputed Acoustic Transfer: Output-Sensitive, Accurate Sound Generation for Geometrically Complex Vibration Sources. *ACM Trans. Graph.* 25, 3 (jul 2006), 987–995. https://doi.org/10.1145/1141911.1141983

Doug L. James and Dinesh K. Pai. 1999. ArtDefo: Accurate Real Time Deformable Objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–72. https://doi.org/10.1145/311535.311542

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (aug 1986), 143–150. https://doi.org/10.1145/15886.15902

Aneta Karaivanova, Michael Mascagni, and Nikolai A. Simonov. 2004. Parallel Quasirandom Walks on the Boundary. *Monte Carlo Methods and Applications* 10, 3-4 (dec 2004), 311–319. https://doi.org/10.1515/mcma.2004.10.3-4.311

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Comput. Graph. Forum* 21, 3 (may 2002), 531–540. https://doi.org/10.1111/1467-8659.t01-1-00703

Alexander Keller. 1997. Instant Radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 49–56. https://doi.org/10.1145/258734.258769

Jaroslav Křivánek and Eugene d'Eon. 2014. A Zero-variance-based Sampling Scheme for Monte Carlo Subsurface Scattering. In *ACM SIGGRAPH 2014 Talks* (Vancouver, Canada). ACM, New York, NY, USA, Article 66, 1 pages. https://doi.org/10.1145/2614106.2614138

Yijun Liu, Subrata Mukherjee, Naoshi Nishimura, Martin Schanz, Wenjing Ye, Alok Sutradhar, Ernie Pan, Ney Augusto Dumont, Attilio Frangi, and Andres Saez. 2012. Recent Advances and Emerging Applications of the Boundary Element Method. *Applied Mechanics Reviews* 64, 3 (mar 2012), 38 pages. https://doi.org/10.1115/1.4005491

Sebastian Martin, Peter Kaufmann, Mario Botsch, Martin Wicke, and Markus H. Gross. 2008. Polyhedral Finite Elements Using Harmonic Basis Functions. *Comput. Graph. Forum* 27, 5 (2008), 1521–1529. https://doi.org/10.1111/j.1467-8659.2008.01293.x

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4 (aug 2023), 11 pages. https://doi.org/10.1145/3592400

Zackary Misso, Benedikt Bitterli, Iliyan Georgiev, and Wojciech Jarosz. 2022. Unbiased and Consistent Rendering Using Biased Estimators. *ACM Trans. Graph.* 41, 4, Article 48 (jul 2022), 13 pages. https://doi.org/10.1145/3528223.3530160

Mervin E. Muller. 1956. Some Continuous Monte Carlo Methods for the Dirichlet Problem. *The Annals of Mathematical Statistics* 27, 3 (sep 1956), 569 – 589. https://doi.org/10.1214/aoms/1177728169

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

Michael B. Nielsen and Robert Bridson. 2011. Guide Shapes for High Resolution Naturalistic Liquid Simulation. *ACM Trans. Graph.* 30, 4, Article 83 (jul 2011), 8 pages. https://doi.org/10.1145/2010324.1964978

Tomoyuki Nishita and Eihachiro Nakamae. 1985. Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection. *SIGGRAPH Comput. Graph.* 19, 3 (jul 1985), 23–30. https://doi.org/10.1145/325165.325169

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-Shaded Images. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–8. https://doi.org/10.1145/1360612.1360691

Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. 2010. OptiX: A General Purpose Ray Tracing Engine. *ACM Trans. Graph.* 29, 4, Article 66 (jul 2010), 13 pages. https://doi.org/10.1145/1778765.1778803

Clemens Pechstein. 2013. Special Lecture on Boundary Element Methods. https://www.numa.uni-linz.ac.at/Teaching/LVA/2013s/SpezVO/bem13.pdf

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2018. *Physically based rendering: From theory to implementation.* Morgan Kaufmann, Burlington, Massachusetts.

Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. *Comput. Graph. Forum* 41, 4 (2022), 51–62. https://doi.org/10.1111/cgf.14586

Damien Rioux-Lavoie, Ryusuke Sugimoto, Tümay Özdemir, Naoharu H. Shimada, Christopher Batty, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2022. A Monte Carlo Method for Fluid Simulation. *ACM Trans. Graph.* 41, 6, Article 240 (nov 2022), 16 pages. https://doi.org/10.1145/3550454.3555450

Karl K. Sabelfeld. 1982. Vector algorithms in the Monte-Carlo method for solving systems of second-order elliptic equations and Lame's equation. *Doklady Akademii Nauk SSSR* 262, 5 (1982), 1076–1080. https://www.mathnet.ru/eng/dan45069 (In Russian).

Karl K. Sabelfeld. 1991. *Monte Carlo Methods: in Boundary Value Problems.* Springer, Berlin, Heidelberg.

Karl K. Sabelfeld. 2012. Stochastic Boundary Methods of Fundamental Solutions for solving PDEs. *Engineering Analysis with Boundary Elements* 36, 7 (jul 2012), 1092–1103. https://doi.org/10.1016/j.enganabound.2012.02.003

Karl K. Sabelfeld and Nikolai A. Simonov. 1994. *Random Walks on Boundary for Solving PDEs.* De Gruyter, Berlin. https://doi.org/10.1515/9783110942026

Rohan Sawhney and Keenan Crane. 2020. Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-Based Methods on Volumetric Domains. *ACM Trans. Graph.* 39, 4, Article 123 (aug 2020), 18 pages. https://doi.org/10.1145/3386569.3392374

Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4 (aug 2023), 22 pages. https://doi.org/10.1145/3592398

Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-Free Monte Carlo for PDEs with Spatially Varying Coefficients. *ACM Trans. Graph.* 41, 4, Article 53 (jul 2022), 17 pages. https://doi.org/10.1145/3528223.3530134

Nikolai A Simonov. 2008. Walk-on-Spheres Algorithm for Solving Boundary-Value Problems with Continuity Flux Conditions. In *Monte Carlo and Quasi-Monte Carlo Methods 2006.* Springer, Berlin, Heidelberg, 633–643. https://doi.org/10.1007/978-3-540-74496-2_38

Nikolai A Simonov. 2017. Walk-on-spheres algorithm for solving third boundary value problem. *Applied Mathematics Letters* 64 (feb 2017), 156–161. https://doi.org/10.1016/j.aml.2016.09.008

Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary Element Octahedral Fields in Volumes. *ACM Trans. Graph.* 36, 4, Article 114b (jul 2017), 16 pages. https://doi.org/10.1145/3072959.3065254

Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2022. Surface-Only Dynamic Deformables using a Boundary Element Method. *Comput. Graph. Forum* 41, 8 (2022), 75–86. https://doi.org/10.1111/cgf.14625

Justin Talbot, David Cline, and Parris K. Egbert. 2005. Importance Resampling for Global Illumination. In *Proceedings of the Eurographics Symposium on Rendering Techniques, Konstanz, Germany, June 29 - July 1, 2005.* Eurographics Association, 139–146. https://doi.org/10.2312/EGWR/EGSR05/139-146

Adriaan van Wijngaarden. 1953. A transformation of formal series. *Indagationes mathematicae* 15 (jan 1953), 522–543.

Eric Veach. 1998. *Robust Monte Carlo Methods for Light Transport Simulation.* Ph. D. Dissertation. Stanford University, Stanford, CA, USA. Advisor(s) Leonidas J. Guibas.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 419–428. https://doi.org/10.1145/218380.218498

Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–76. https://doi.org/10.1145/258734.258775

Ekrem Fatih Yılmazer, Delio Vicini, and Wenzel Jakob. 2022. Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators. https://doi.org/10.48550/arXiv.2208.02114 arXiv:2208.02114 [cs.GR]

A FUNDAMENTAL SOLUTION

The fundamental solution $G(\mathbf{x},\mathbf{y})$ for Laplace operator is the solution to the equation in an infinite domain: $\Delta G(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y}) = 0$. In 3D, we have $G(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi r}$ and its derivatives are

$$\frac{\partial G}{\partial \mathbf{x}_k}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{r}\cdot\mathbf{e}_k}{4\pi r^3}, \quad \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y}) = -\frac{\mathbf{r}\cdot\mathbf{n}_\mathbf{y}}{4\pi r^3}, \quad \frac{\partial G}{\partial \mathbf{n}_\mathbf{x}}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{r}\cdot\mathbf{n}_\mathbf{x}}{4\pi r^3},$$
$$\frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi}\left[\frac{\mathbf{n}_\mathbf{y}\cdot\mathbf{e}_k}{r^3} - 3\frac{(\mathbf{r}\cdot\mathbf{n}_\mathbf{y})(\mathbf{r}\cdot\mathbf{e}_k)}{r^5}\right].$$

In 2D, we have $G(\mathbf{x},\mathbf{y}) = -\frac{1}{2\pi}\log r$ and its derivatives are

$$\frac{\partial G}{\partial \mathbf{x}_k}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{r}\cdot\mathbf{e}_k}{2\pi r^2}, \quad \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y}) = -\frac{\mathbf{r}\cdot\mathbf{n}_\mathbf{y}}{2\pi r^2}, \quad \frac{\partial G}{\partial \mathbf{n}_\mathbf{x}}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{r}\cdot\mathbf{n}_\mathbf{x}}{2\pi r^2},$$
$$\frac{\partial^2 G}{\partial \mathbf{x}_k \partial \mathbf{n}_\mathbf{y}}(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi}\left[\frac{\mathbf{n}_\mathbf{y}\cdot\mathbf{e}_k}{r^2} - 2\frac{(\mathbf{r}\cdot\mathbf{n}_\mathbf{y})(\mathbf{r}\cdot\mathbf{e}_k)}{r^4}\right].$$

where $\mathbf{r} = \mathbf{y}-\mathbf{x}$, $r = \|\mathbf{r}\|$, and $\mathbf{e}_k$ is the k-th basis vector.

B CLASSES OF INTEGRAL EQUATIONS

B.1 The rendering equation, WoS, and WoB

Both the rendering equation and the formulation of WoS are understood to be Fredholm equations of the second kind [Pharr et al. 2018; Qi et al. 2022]. This type of integral equation takes the form $f(\mathbf{x}) = g(\mathbf{x}) + \int_D K(\mathbf{x},\mathbf{y}) f(\mathbf{y})\, d\mathbf{y}$ where $K(\mathbf{x},\mathbf{y})$ is a given integral kernel, $g(\mathbf{x})$ is a known function, $f(\mathbf{y})$ is an unknown function we want to solve for, and D is a fixed integration domain. When the integration domain changes depending on $\mathbf{x}$, it is called a Volterra equation (of the second kind).

In rendering, for surface light transport, $\mathbf{x}$ is a tuple of a location on the surface and a direction from there (i.e., to measure radiance coming from that particular location toward the particular direction). The function $g(\mathbf{x})$ is the emission term, and $K(\mathbf{x},\mathbf{y})$ is defined as $K(\mathbf{x},\mathbf{y}) = f_r(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y})G_{eo}(\mathbf{x},\mathbf{y})$ where f_r is the BSDF, V is the visibility term, and G_{eo} is the geometry term [Pharr et al. 2018]. The integration domain is fixed as the surfaces of the scene, so it fits the definition of Fredholm equations.

In WoS, $g(\mathbf{x})$ is defined as $g(\mathbf{x}) = \int_{B_\mathbf{x}} \overline{b}(\mathbf{z}) G_{B_\mathbf{x}}(\mathbf{x},\mathbf{z}) d\mathbf{z}$ where $\overline{b}(\mathbf{x})$ is the source function, $G_{B_\mathbf{x}}(\mathbf{x},\mathbf{y})$ is Green's function for the largest ball $B_\mathbf{x}$ contained within the domain centered at $\mathbf{x}$. Note that both functions are given so $g(\mathbf{x})$ is also still given. The kernel $K(\mathbf{x},\mathbf{y})$ for WoS is defined as $K(\mathbf{x},\mathbf{y}) = \frac{\partial G_{B_\mathbf{x}}(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}}$ and the integration domain is $D = \partial B_\mathbf{x}$, the surface of the ball $B_\mathbf{x}$. While Qi et al. [2022] claimed that the formulation of WoS is a Fredholm equation of the second kind, it is a Volterra equation of the second kind since the integration domain $D = \partial B_\mathbf{x}$ changes according to $\mathbf{x}$. A connection between WoS and the rendering equation was imperfectly made in this sense. On the other hand, in WoB, the integration domain is fixed as the boundary, so it is precisely a Fredholm equation.

The rendering equation, commonly referred to as a Fredholm equation [Pharr et al. 2018], appears to contradict the definition of a fixed integration domain when it is solved by MC ray tracing. Ray tracing from a point $\mathbf{x}$ results in a varying set of visible points $\mathbf{y}$ depending on $\mathbf{x}$, and thus it appears to be a Volterra equation. However, the integration domain is fixed as the surfaces of the scene, so it still fits the definition of a Fredholm equation. While this mismatch is paradoxical, we have identified that classifying the rendering equation as *solely* a Fredholm equation is inaccurate.

In addition to the aforementioned area form, the rendering equation can take a solid angle form, in which the integration domain D becomes the hemispherical angular domain around $\mathbf{x}$, and the kernel $K(\mathbf{x},\mathbf{y})$ is defined as $f_r(\mathbf{x}\to\mathbf{y})\cos\theta$, where $\mathbf{y}$ is the first visible point from $\mathbf{x}$ along the direction towards $\mathbf{y}$ [Pharr et al. 2018]. This solid-angle form is the form used in MC ray tracing, and in this context, the rendering equation is a Volterra equation of the second kind, due to the changing angular integration domain based on $\mathbf{x}$. However, the area form of the rendering equation, which integrates over all surface points, is a Fredholm integral equation of the second kind. The assertions made by Qi et al. [2022] still hold if one accepts that the solid-angle form of the rendering equation is a Volterra equation, just like the formulation of WoS.

This distinction is subtle since a Volterra equation can be converted into a Fredholm equation by properly expanding the kernel $K(\mathbf{x},\mathbf{y})$ with zeros in a common fixed integration domain. This conversion is precisely what actually occurs in the area form of the rendering equation, where the visibility term $V(\mathbf{x},\mathbf{y})$ returns zero for points $\mathbf{y}$ that are not visible from $\mathbf{x}$. The area form of the rendering equation is still a Fredholm equation, but it can be transformed into a Volterra equation by redefining the integration domain to include only visible points from $\mathbf{x}$. The formulation of WoB cannot be reduced to a Volterra equation, as its kernel is nonzero everywhere.

B.2 Singularity and reciprocity of the kernel

Boundary integral equations involve singular kernels, where the kernel becomes unbounded as the distance between points $r = |\mathbf{x}-\mathbf{y}|$ approaches zero. The order of singularity in the kernel, K, can be classified as weakly singular, strongly singular, and hypersingular, for $K = O(1/r)$, $K = O(1/r^2)$, and $K = O(1/r^3)$ respectively. The orders of singularity here pertain to three-dimensional scenarios, although the underlying concepts remain unchanged in two-dimensional scenarios. As the order of singularity increases, robust estimation becomes challenging. Weak and strong singularities in the kernel are prevalent in the rendering equation, and a plethora of techniques to circumvent numerical difficulties have been studied [Pharr et al. 2018]. For instance, tracing a ray, as opposed to directly sampling a point on surfaces, can avoid singularities arising from the geometry term. Similar methods are applied to WoB. However, hypersingular integrals necessitate special attention during computation and are not typically encountered in rendering. It is advisable to avoid high-order singularities whenever possible.

The kernel K is called symmetric if $K(\mathbf{x},\mathbf{y}) = K(\mathbf{y},\mathbf{x})$ and asymmetric otherwise. In rendering, the terms reciprocal and non-reciprocal are used instead of symmetric and asymmetric, respectively; the kernel of the rendering equation is usually symmetric due to the physics of light, but can be asymmetric in certain cases [Veach 1998]. Both WoB and WoS deal with asymmetric kernels but they

do not pose a problem for the solvers as long as they are treated properly, similarly to handling of asymmetric kernels in rendering.

B.3 First-kind equations and their MC estimation

Both Fredholm and Volterra integral equations have first and second kind forms. A Fredholm equation of the first kind takes the form $g(\mathbf{x}) = \int_D K(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, d\mathbf{y}$, where both g and K are known and f is the unknown function to be solved for. Note we used the term first kind equation abusively in the main text: strictly, we should call it a first kind equation if the equation holds for the entire integral domain, but in our mixed boundary problem estimator, we have different integral equations defined conditionally on the type of boundary at each point. There is also a third kind of Fredholm equation, but it is not relevant to WoB in our paper.

While second-kind equations can be estimated using MC integration via Neumann series expansion, this technique is not applicable to first-kind equations. For instance, one might consider estimating the integral using MC integration as: $g(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{K(\mathbf{x},\mathbf{y}_i) f(\mathbf{y}_i)}{p(\mathbf{y}_i)}$. This equation cannot be used, as f is unknown and cannot be solved for using MC integration alone. We therefore focused on having second-kind equations that are compatible with MC integration, which requires us to choose a specific BIE to achieve this goal.