Walk-on-Interfaces: A Monte Carlo Estimator for Elliptic Interface Problem

Xinwen Ding, Adam R Stinchcombe

University of Toronto

Preprint Here

Motivation

Elliptic interface problems arise from elliptic PDEs with discontinuous coefficients. They appear in physics, biology, and engineering to model heterogeneous media where material properties change abruptly across internal interfaces. Traditional numerical solvers struggle with such problems: they require high-quality meshes and carefully-designed quadratures for singular kernels. As a result, computations become memory-intensive and expensive, especially for domains containing multiple irregular interfaces.

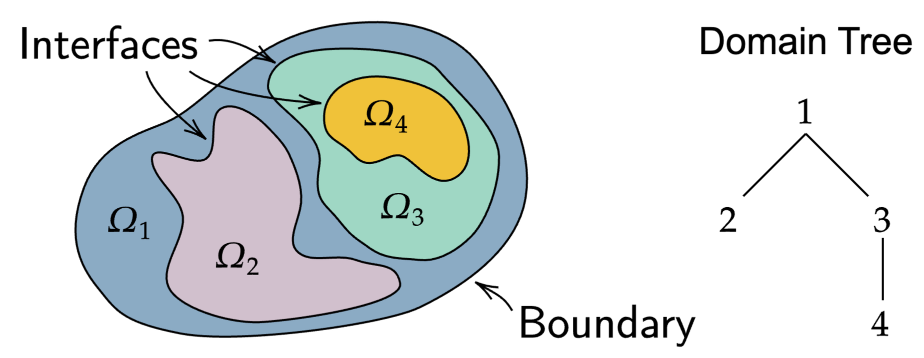

Figure 1. (Left) A domain of mixed media. (Right) Tree structure definition of the domain.

Contributions

We present **Walk-on-Interfaces (WoI)**, a **grid-free Monte Carlo estimator** for Neumann elliptic interface problems with nonhomogeneous flux jump conditions.

- **Pointwise Estimation:** Estimates the solution queried anywhere in the domain.
- **Uniform accuracy:** Preserves accuracy near boundary and interfaces.
- **Scalable:** Highly parallelizable and naturally extends to higher dimensions.
- **Neural Representation:** Combines our Monte Carlo method with a deep neural network to produce smooth and continuous representation of solution.

Mathematical Formulation & Key Notation

Problem setup. We solve a Neumann elliptic interface problem defined by

$$\begin{cases} \Delta u(\boldsymbol{x}) = 0 & \text{in } \Omega_1 \setminus \bigcup_{i\geq 2} \partial\Omega_i \\ \sigma_1 \partial_{\boldsymbol{n}} u(\boldsymbol{x}) = b_1(\boldsymbol{x}) & \text{on } \partial\Omega_1 \\ [u](\boldsymbol{x}) = 0 & \text{on } \partial\Omega_i,\ \forall i \geq 2 \\ [\sigma \partial_{\boldsymbol{n}} u](\boldsymbol{x}) = b_i(\boldsymbol{x}) & \text{on } \partial\Omega_i,\ \forall i \geq 2 \end{cases}$$

in which $b_i(\boldsymbol{x}) \in \mathcal{C}^{0,1}$, for all $i = 1, \dots, N$.

Solution Representation. The solution can be represented by a sum of N single-layer potentials:

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} \int_{\partial\Omega_j} \Phi(\boldsymbol{x}, \boldsymbol{y}) \gamma_j^*(\boldsymbol{y})\, dA_{\boldsymbol{y}}. \quad (1)$$

Table 1. In this table $f : \partial\Omega_j \to \mathbb{R}$.

Notation	Definition
$\partial_{\boldsymbol{n}}, \frac{\partial}{\partial \boldsymbol{n}}$	normal derivative
σ_i	material property, constant in Ω_i
$u_+(\boldsymbol{x}_0)$	$\lim_{\boldsymbol{x}\to\boldsymbol{x}_0} u(\boldsymbol{x})$ from outside of $\partial\Omega_i$
$u_-(\boldsymbol{x}_0)$	$\lim_{\boldsymbol{x}\to\boldsymbol{x}_0} u(\boldsymbol{x})$ from inside of $\partial\Omega_i$
$[u](\boldsymbol{x}_0)$	$u_+(\boldsymbol{x}_0) - u_-(\boldsymbol{x}_0)$
$\Phi(\boldsymbol{x}, \boldsymbol{y})$	free-space Green's function
γ_j^*	charge density function
$K_{i,j}^*(\boldsymbol{x}, \boldsymbol{y})$	$-\frac{\partial \Phi(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{n}(\boldsymbol{x})}$, $\boldsymbol{x} \in \partial\Omega_i$ and $\boldsymbol{y} \in \partial\Omega_j$
$\mathcal{K}_{i,j}^*$	$\mathcal{K}_{i,j}^* f = \int_{\partial\Omega_j} K_{i,j}^*(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y})\, dA_{\boldsymbol{y}}$
p_i	parent index of i in the domain tree
w_i	$1 - \frac{1}{2}\delta_{i,M}$
α_i	2, if $i = 1$; otherwise $\frac{2(\sigma_i - \sigma_{p_i})}{\sigma_{p_i} + \sigma_i}$
$\beta_i(\boldsymbol{x})$	$\frac{2}{\sigma_1} b_1(\boldsymbol{x})$, if $i = 1$; otherwise $-\frac{2}{\sigma_{p_i}+\sigma_i} b_i(\boldsymbol{x})$

Solve for γ_j^*: 1.Impose boundary and interface conditions

2.Express γ_j^* by Neumann series that involves operators $\mathcal{K}_{i,j}^*$.

Approximate Solution: Truncating the series for γ_j^* in a convergent manner, and substituting it into Eq. (1), we approximate the solution $u(\boldsymbol{x})$ as

$$u(\boldsymbol{x}) \approx \sum_{i=0}^{M} w_i \sum_{h_i,\dots,h_0=1}^{N} \int_{\partial\Omega_{h_i}} \Phi(\boldsymbol{x}, \boldsymbol{y}) \left((\alpha_{h_i} \mathcal{K}_{h_i,h_{i-1}}^*) \cdots (\alpha_{h_2} \mathcal{K}_{h_2,h_1}^*)(\alpha_{h_1} \mathcal{K}_{h_1,h_0}^*) \beta_{h_0} \right)(\boldsymbol{y})\, dA_{\boldsymbol{y}}\ . \quad (2)$$

A Theorem Towards Walk-on-Interfaces

Given a query point $\boldsymbol{x}$ and a fixed set of $h_0, h_1, \dots, h_i \in \{1, 2, \dots, N\}$. Let $\{Y_0, Y_1, \dots, Y_i, \dots\}$ be a Markov chain starting from $\boldsymbol{x}$ such that $Y_i \in \partial\Omega_{h_i}$. Let $p_0(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x}, \boldsymbol{y})$ be any probability density functions (PDFs) denoting the transitional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$. Define a family of functions

$$Q_{(h_i,\dots,h_1,h_0)}^* = \begin{cases} \frac{\beta_{h_0}(Y_0)}{p_0(\boldsymbol{x},Y_0)} & \text{if } i = 0 \\ Q_{(h_{i-1},\dots,h_1,h_0)}^* \frac{K_{h_i,h_{i-1}}^*(Y_i,Y_{i-1})}{p(Y_{i-1},Y_i)} & \text{if } i \geq 1 \end{cases}. \quad (3)$$

Then, $\displaystyle\int_{\partial\Omega_{h_i}} \Phi(\boldsymbol{x}, \boldsymbol{y}) (\mathcal{K}_{h_i,h_{i-1}}^* \cdots \mathcal{K}_{h_2,h_1}^* \mathcal{K}_{h_1,h_0}^* \beta_{h_0})\, dA_{\boldsymbol{y}} = \mathbb{E}\left[Q_{(h_i,\dots,h_1,h_0)}^* \Phi(\boldsymbol{x}, Y_i) \right]$.

Walk-on-Interfaces

Algorithm Visualization

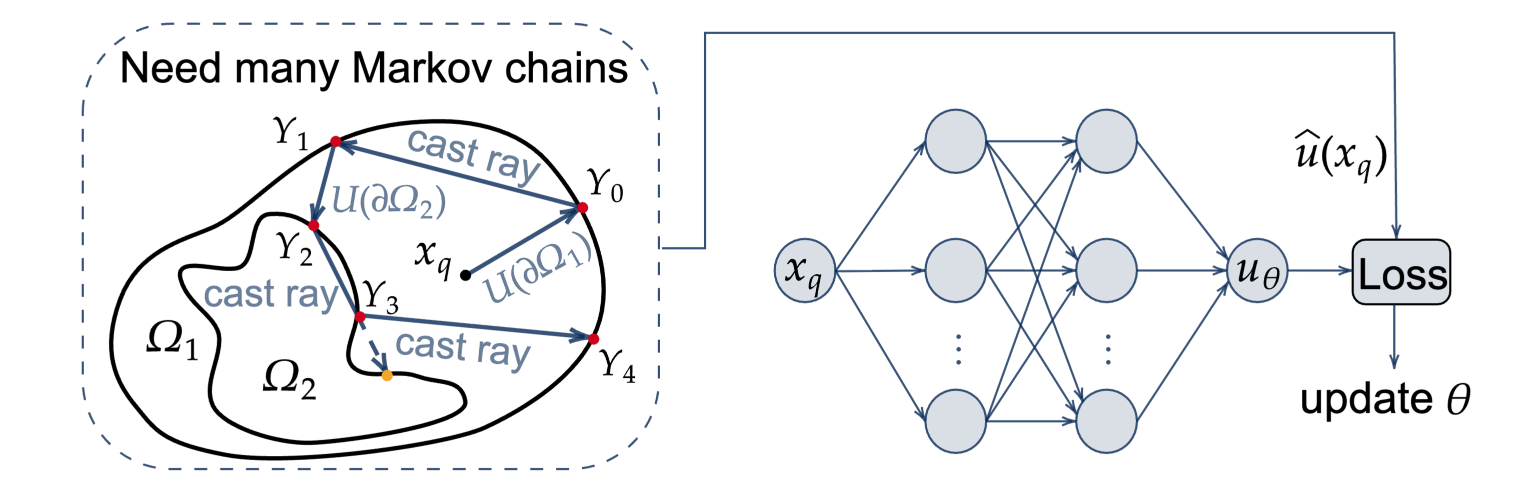

Figure 2. WoI workflow. (Left) An example Markov chain of length 5. (Right) Deep neural network.

✘ Applying the theorem directly to Eq. (2) leads to an $\mathcal{O}(N^M)$ algorithm.
✔ Rescale the coefficient α_{h_i} to interpret Eq. (2) as a discrete expected value.

Algorithm Overview

Let $\mathcal{Q} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_Q\}$ be a set of query points.

For each $\boldsymbol{x}_q \in \mathcal{Q}$ do:

1. **Schedule:** Draw $\mathcal{S}$ sets of *discrete* random variables $\{H_0, H_1, \dots, H_M\}$ under the distribution

$$\mathbb{P}(H_i = n) = \begin{cases} \frac{1}{N} & \forall n \in \{1, \dots, N\}, \text{ when } i = 0 \\ \frac{|\alpha_n|}{\|\boldsymbol{\alpha}\|_1} & \forall n \in \{1, \dots, N\}, \text{ when } i \geq 1 \end{cases}.$$

Each set of realization $\{h_0, h_1, \dots, h_M\}$ defines one *scheduled walk*.

2. **Walk:** Sample $\mathcal{W}$ Markov chains, $\{Y_0, Y_1, \dots Y_M\}$, for each scheduled walk.
   - Initialization step: sample $Y_0 \sim \mathcal{U}(\partial\Omega_{h_0})$.
   - For each subsequent step $i = 1, \dots, M$:
     - If h_{i+1} is an ancestor of h_i or $h_{i+1} = h_i$, set Y_{i+1} to be any intersection between $\partial\Omega_{h_{i+1}}$ and a random ray from Y_i.
     - Otherwise, sample $Y_{i+1} \sim \mathcal{U}(\partial\Omega_{h_{i+1}})$.

3. **Estimate:** Evaluate the following estimator

$$\hat{u}(\boldsymbol{x}_q) \approx N \sum_{i=0}^{M} w_i \|\boldsymbol{\alpha}\|_1^i \mathbb{E}_{H_i,\dots,H_0} \left[\prod_{m=1}^{i} \operatorname{sign}(\alpha_{H_m}) \mathbb{E}_{Y_i} \left[Q_{(H_i,\dots,H_1,H_0)}^* \Phi(\boldsymbol{x}_q, Y_i) \right] \right]. \quad (4)$$

Neural Representation: Use a deep neural network to smooth out the Monte Carlo error in $\{\hat{u}(\boldsymbol{x}_1), \hat{u}(\boldsymbol{x}_2), \dots, \hat{u}(\boldsymbol{x}_Q)\}$ and obtain continuous representation of solution.

Results

The Electrical Conductivity Problem.

The electrical conductivity problem models the electric potential $u(\boldsymbol{x})$ induced by injecting a current into heterogeneous media, where each subdomain Ω_i has a constant conductivity σ_i. The setup used in this example is summarized below.

- **Domain:** Ω_1 – outer sphere (conductor, $\sigma_1 = 1.2$);
  Ω_2 – inner cow (insulator, $\sigma_2 = 0$).
- **Boundary/Interface Conditions:** $b_1(r, \phi, \theta) = 10 \sin^9 \phi \cos^9 \theta$; $b_2(r, \phi, \theta) = 0$.

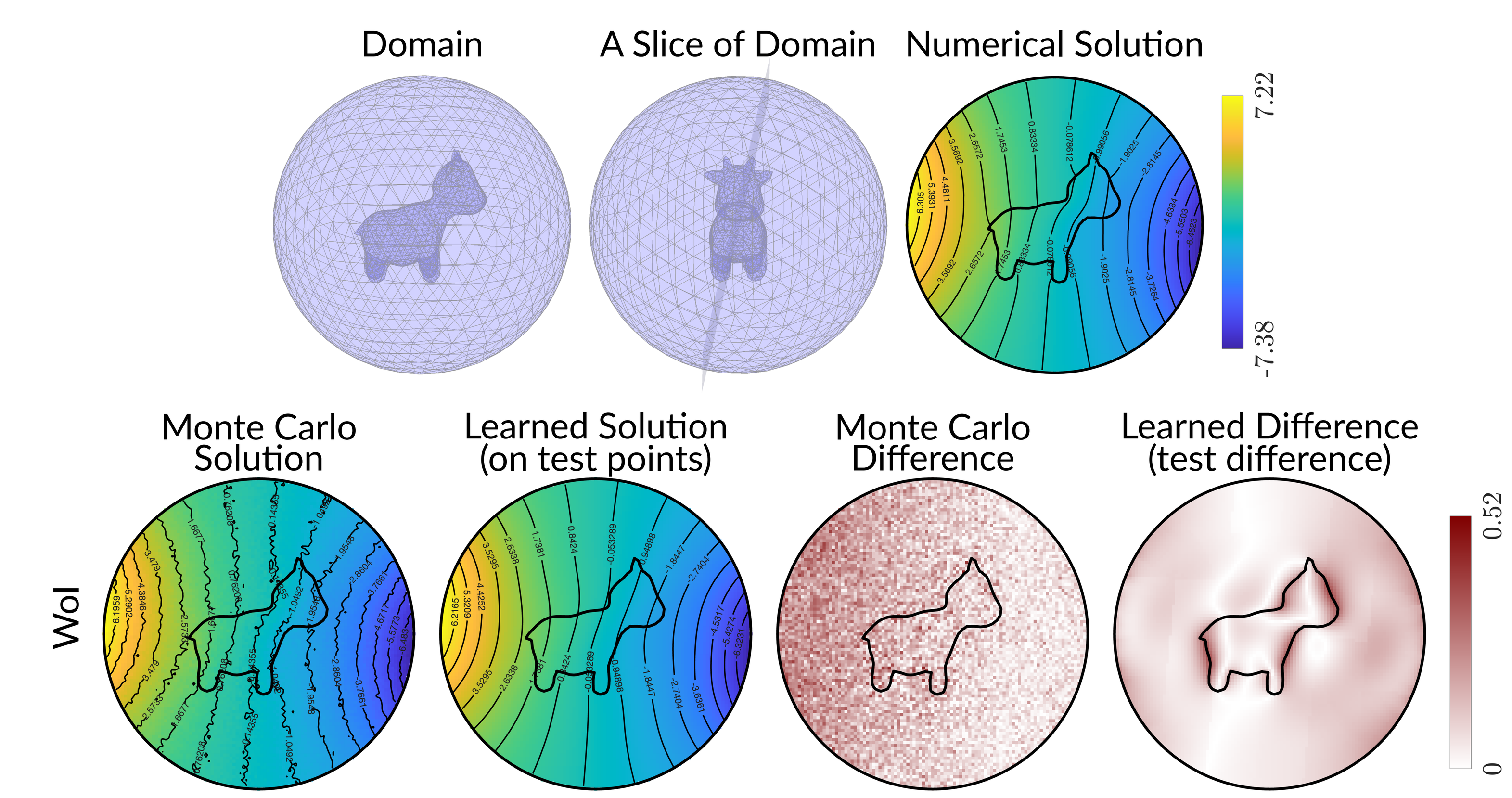

Figure 3. Monte Carlo Solution: L_2 difference is 0.15, relative L_2 difference is 5.51%. Learned Solution: L_2 difference is 0.073, relative L_2 difference is 3.39%. Set $\mathcal{W} = 10^7$ and $M = 5$.

Groundwater Flow.

Groundwater flow is governed by Darcy's law. The domain consists of subregions Ω_i, each representing a geological material (sand, soil, or rock) with distinct hydraulic conductivity σ_i. The solution $u(\boldsymbol{x})$ denotes the hydraulic head.

- **Domain:** Ω_1 – hill ($\sigma_1 = 0.2$ cm/s); Ω_2 – tilted rock at the top ($\sigma_2 = 0.05$ cm/s); $\Omega_{3\text{–}6}$ – four mid-layer rocks ($\sigma_{3\text{–}6} = 0.03$ cm/s); Ω_7 – bedrock ($\sigma_7 = 0.005$ cm/s).
- **Boundary Condition:** Rainfall on the hilltop and water extraction at downhill.
- **Interface Conditions:** $b_i(\boldsymbol{x}) = 0$ for all $i \in \{2, 3, \dots 7\}$.

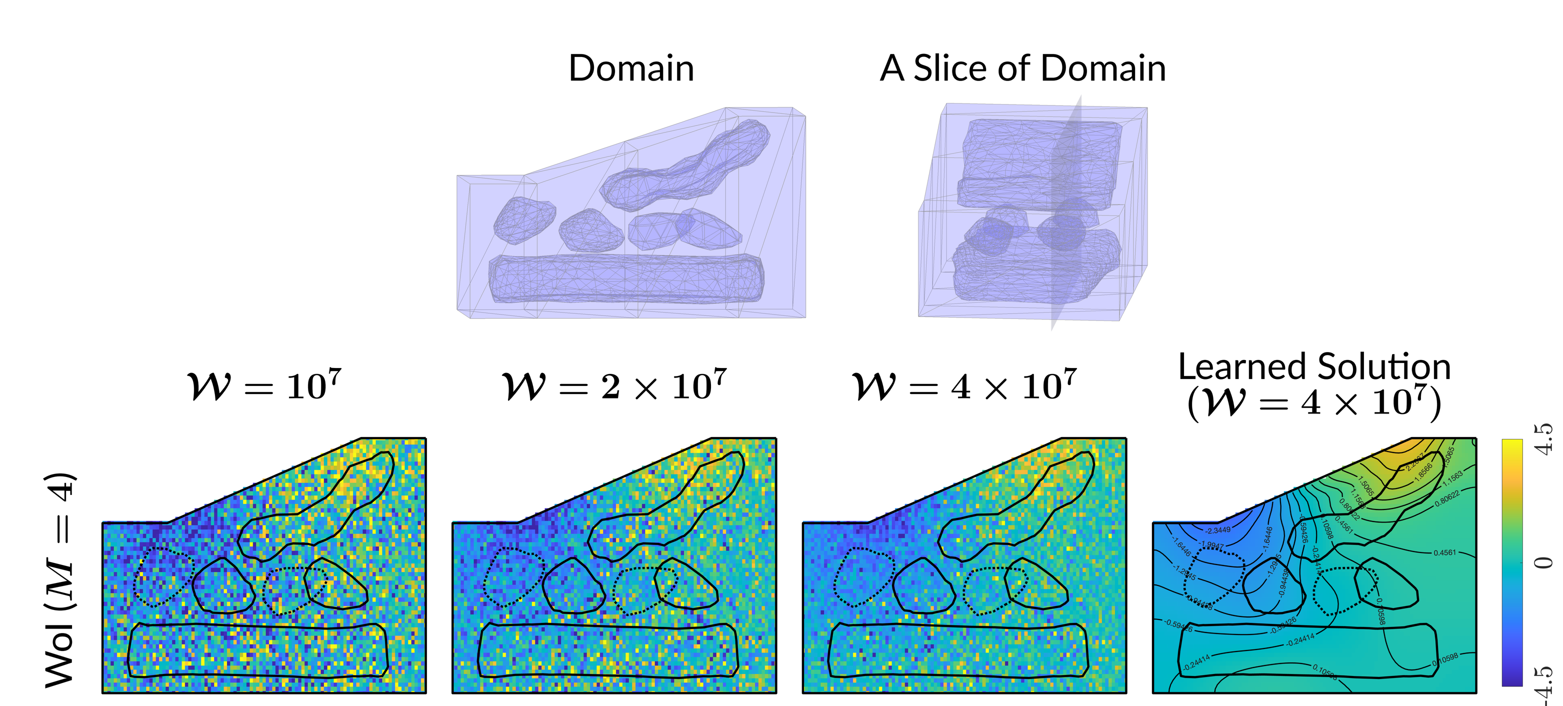

Figure 4. Monte Carlo solutions with different number of Markov chains and learned solution. The rocks with dotted outline do not intersect with the cutting plane. This problem is challenging for grid-based numerical solvers due to the number of interfaces in the domain.

xinwen.ding@mail.utoronto.ca