Walk on Decomposed Subdomains: A Hybrid Monte Carlo–Deterministic Solver for Elliptic PDEs

CLÉMENT JAMBON, Massachusetts Institute of Technology (MIT), USA
MOHAMMAD SINA NABIZADEH, Massachusetts Institute of Technology (MIT), USA
MINA KONAKOVIĆ LUKOVIĆ, Massachusetts Institute of Technology (MIT), USA

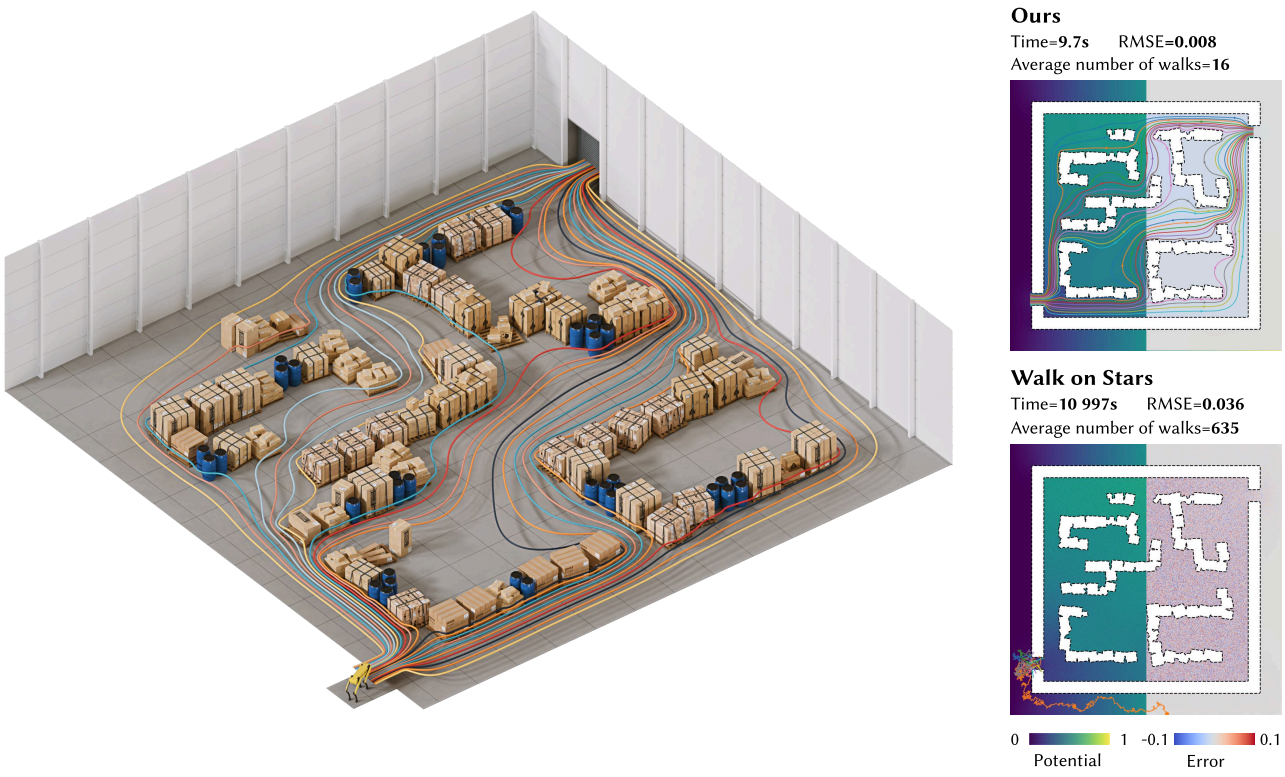
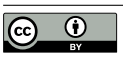


Fig. 1. We use our hybrid solver to solve a heat-flow problem and extract smooth potential paths in the complex geometry of a cluttered warehouse (left), where objects and walls are modeled as Neumann boundaries (shown with dashed black lines on the right). By combining the geometric flexibility of Monte Carlo methods (e.g., Walk on Stars [Sawhney et al. 2023]) with the global coupling of grid-based solvers, our approach produces potentials with low variance and controlled bias (top-right) that enable reliable streamline extraction. In contrast, Walk on Stars is dominated by noise, making streamline extraction impractical. The efficiency of our method arises from two factors: shorter, localized walks within decomposed subdomains (16 steps on average versus 635 for Walk on Stars) that reduces Monte Carlo variance, and a deterministic global solve that exactly solves an induced discrete random-walk system with controlled bias and no additional variance. As a result, our method computes the full solution in 9.7 seconds at equal sample count—compared to over 3 hours for Walk on Stars—while achieving a 4.5× lower error. This setup also naturally allows for efficient re-solves given local scene geometry edits, see Figure 3.

Authors' Contact Information: Clément Jambon, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA, cjambon@mit.edu; Mohammad Sina Nabizadeh, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA, sinabiz@mit.edu; Mina Konaković Luković, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA, minakl@mit.edu.



ACM 1557-7368/2026/7-ART132
https://doi.org/10.1145/3811340

Elliptic partial differential equations are ubiquitous in graphics and engineering, but remain challenging to solve on complex or evolving geometries. Traditional discretization schemes (e.g., FEM/FDM) provide stable, globally coupled solutions but require heavy meshing or extreme refinement to accurately resolve geometric detail. In contrast, grid-free Monte Carlo methods (e.g., Walk on Spheres/Stars) adapt naturally to arbitrary geometry and offer massive parallelism, but rely on long random walks whose variance grows rapidly, particularly in the presence of Neumann boundaries, leading to slow convergence. We introduce a hybrid approach that combines the geometric flexibility of Monte Carlo estimation with deterministic global

solves that do not introduce additional stochastic error. Our method decomposes the domain into simple, regular subdomains and uses Monte Carlo to estimate local first-passage solution operators (Poisson kernels), where walk lengths and variance are inherently controlled by the reduced spatial scale. These local operators are assembled into a sparse global system whose solution is obtained via a deterministic linear solve that exactly replaces simulating discrete random walks throughout the domain. This global solve trades stochastic variance for a fixed, resolution-dependent discretization bias, yielding stable and reusable solution operators. As a result, our method attains accurate, geometry-aware solutions even on coarse discretizations, and enables efficient solves and re-solves by computing and updating only the local operators affected by the geometry and its changes. We evaluate the approach on complex two-dimensional domains, benchmarking accuracy and convergence against standard grid-free and grid-based baselines, and demonstrate applications to microstructure simulation and flow-based path planning and streamline visualization.

CCS Concepts: • **Mathematics of computing → Partial differential equations**; **Integral equations**; **Probabilistic algorithms**.

ACM Reference Format:
Clément Jambon, Mohammad Sina Nabizadeh, and Mina Konaković Luković. 2026. Walk on Decomposed Subdomains: A Hybrid Monte Carlo–Deterministic Solver for Elliptic PDEs. *ACM Trans. Graph.* 45, 4, Article 132 (July 2026), 22 pages. https://doi.org/10.1145/3811340

1 Introduction

Many real-world systems and environments involve complex geometries, such as cities, datacenters, electronic circuits, and warehouses. In such settings, the physical phenomena of interest are often well modeled by elliptic boundary value problems (BVPs) [Evans 2022], e.g., heat conduction, electrostatics, path planning (Figure 1), and steady-state potential flows (Figure 4). However, analyzing or designing such systems through their governing PDEs is particularly challenging due to their highly complex geometry, and becomes even more cumbersome when the geometry is repeatedly modified during iterative design or time evolution.

Grid-based PDE solvers, such as the finite difference method (FDM) and finite element method (FEM), do not scale gracefully with geometric complexity. FDM struggles to represent intricate boundaries without excessive or adaptive grid refinement, while FEM requires expensive volumetric meshing to faithfully capture fine geometric detail. In practice, this creates a stark trade-off: achieving low error requires meshes with a very large number of degrees of freedom, whereas coarser discretizations exhibit substantial bias (Figure 2, right). On the other hand, grid-free Monte Carlo methods [Sawhney and Crane 2020; Sawhney et al. 2023; Miller et al. 2024] have recently gained popularity in the graphics community. The *Walk on Spheres* algorithm (WoS) [Muller 1956; Sawhney and Crane 2020] and its extensions such as the *Walk on Stars* algorithm (WoSt) [Sawhney et al. 2023; Miller et al. 2024] avoid the need for meshing and explicit basis functions, and instead rely on point-wise geometric queries, leading to unbiased estimators that offer scalability, parallelism and geometric flexibility. Unfortunately, Monte Carlo methods trade bias-free estimation for variance. Even in the presence of simple geometry, they are plagued by notoriously slow convergence, in particular when Neumann boundaries dominate (see Figure 2, left). A key observation is that this slow convergence is inherent to point-wise, iterative, uncorrelated estimators rather than

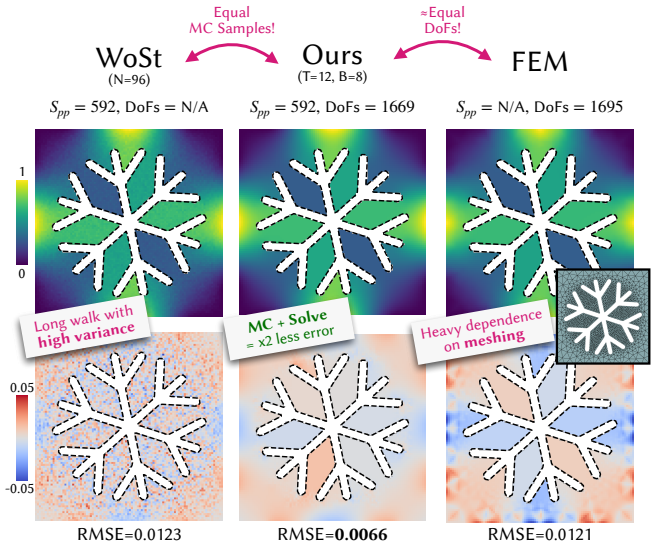


Fig. 2. Pure Monte Carlo (MC) methods, here Walk on Stars (WoSt) [Sawhney et al. 2023], suffer from excessive variance in Neumann-dominated scenarios (left). FEM solutions, in contrast, depend strongly on the underlying mesh (inset), resulting in significant bias at coarse resolutions (right). Our method (center) bridges these two regimes using a simple grid-like decomposition—without explicit meshing—achieving in this example roughly half the error at equal Monte Carlo sample budget S_{pp} and comparable numbers of degrees of freedom (DoFs). Dashed lines indicate Neumann boundaries, while Dirichlet conditions are prescribed on the square domain. Please refer to Section 6 for a definition and discussion of N, T and B.

the coupled structure of a global solve. This observation leads us to ask: *can we bridge continuous Monte Carlo estimators and discrete approximations to leverage the best of both worlds?*

Our main intuition is that Monte Carlo (MC) estimators are particularly effective at estimating solutions of BVPs on local domains with simple exit statistics but tend to struggle on larger and more intricate domains, where the underlying random walks become excessively long and high-variance. By contrast, once a graph or mesh coupling spatial locations has been constructed, global coupling enables discrete elliptic systems to be solved efficiently and reliably [LeVeque 2007; Hughes 2003]. We therefore propose to decompose complex domains into simple, regular subdomains—typically a regular tiling—for which local exit statistics can be efficiently estimated using Monte Carlo methods, such as WoS/WoSt [Sawhney and Crane 2020; Sawhney et al. 2023]. The variance in these local operator estimates is significantly reduced due to the smaller spatial scale and simpler boundary geometry encountered locally. Once tabulated, these statistics define *local solution operators* that map boundary values of each subdomain to its interior solution, providing immediate access to interior values given boundary data. However, the values at interfaces between subdomains are themselves unknowns of the global BVP. To address this, we observe that the global solution can be interpreted as a *random walk between subdomain interfaces*, rather than within the full continuous domain. We estimate the corresponding transition probabilities using the same Monte Carlo machinery as above. When discretized,

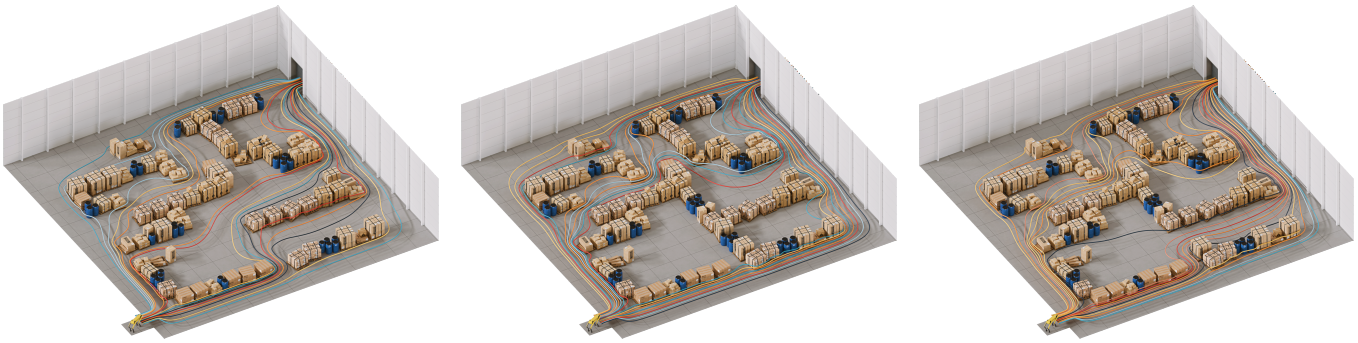


Fig. 3. In the warehouse setting of Figure 1, which exemplifies a scene dominated by Neumann boundaries, local geometry edits—such as adding or removing obstacles—can lead to large changes in potential paths. In our method, these edits translate to local modifications of the discrete random walk induced on the decomposed subdomains, enabling efficient, localized recomputation.

this construction leads to a discrete Dirichlet problem that can be formulated as an absorbing Markov chain [Doyle and Snell 1984; Levin and Peres 2017] and solved via a sparse linear system. Crucially, once the local operators have been estimated, the global solve exactly replaces simulating the discrete random walks. This step is fully *deterministic*, introduces no additional variance, and absorbs geometric complexity into the precomputed transition statistics.

We empirically validate each component of our method, study its convergence, and analyze how its components interact. Remarkably, our method can be viewed as a bridge between grid-based approaches and a pure Monte Carlo caching of the solution operator, exposing a wide range of practical trade-offs. We provide a detailed analysis of these trade-offs and identify *when* and *how* our hybrid approach is most effective, depending on the target resolution, the Monte Carlo sampling budget, and the computational regime, namely *parallel* operator estimation versus *serial* global solve. In particular, our method offers a principled way to amortize the variance inherent to Monte Carlo estimators, which typically arises from long random walks. Conversely, approximating larger solution operators in parallel with Monte Carlo amortizes the global discrete solve by reducing system size. Furthermore, by relying on Monte Carlo–estimated local statistics, the resulting discrete system becomes geometry-aware and subgrid-accurate, while retaining the locality and structure of grid-based methods. This locality enables efficient re-solves when the geometry evolves over time, as only a small subset of local operators needs to be recomputed.

We demonstrate the capabilities of our solver on challenging scenarios involving complex and changing geometry, including microstructures and a range of geometric problems such as flow-based path planning or streamline visualization. While our experiments focus on the two-dimensional setting, extending the method to three dimensions entails the same algorithmic challenges as standard grid-based solvers, while continuing to benefit from the geometric flexibility of Monte Carlo methods. Finally, although we restrict our treatment to zero-Neumann conditions, we view its extension to more general boundary conditions as a promising direction for future work.

2 Related Work

Monte Carlo Methods for Solving PDEs. Monte Carlo methods for elliptic PDEs build on probabilistic interpretations (e.g., Feynman–Kac and first-passage formulations) to solve boundary value problems (BVPs). The *Walk on Spheres* (WoS), originally introduced by Muller [1956], has become a practical tool for solving Poisson-type problems in complex geometry [Sawhney and Crane 2020; Sawhney 2024]. Subsequent works extended WoS to spatially varying coefficients [Sawhney et al. 2022], various boundary conditions, including Neumann [Sawhney et al. 2023] and Robin [Miller et al. 2024], and infinite domains [Nabizadeh et al. 2021]. For a self-contained and comprehensive exposition, we refer the reader to the thesis of Sawhney [2024] and the recent course by Sawhney et al. [2025].

A growing body of work studies variance reduction and caching mechanisms [Miller et al. 2023; Bakbouk and Peers 2023; Zhou et al. 2025], relying on neural networks [Li et al. 2023] or mixtures of Gaussians [Lu et al. 2025]. These methods mostly rely on "reuse" [Miller et al. 2023; Bakbouk and Peers 2023] or correlations across walks [Bao et al. 2025] to reduce variance for a single solution given specific boundary data and geometry, requiring full recomputation when either changes. In contrast, we use Monte Carlo primarily as a way to estimate *local solution operators* on subdomains. Once estimated, one can use them to solve for any boundary values via a fully *deterministic* global solve. Furthermore, these existing works rely on Monte Carlo walks in the entire domain to "enrich" their caches or statistics, which suffer from exceedingly long walks in Neumann-heavy geometry. Our decomposition restricts walks to small subdomains with significantly shorter walks (e.g., 43× in Figure 15).

First-Passage Methods and Tabulated Kernels. First-passage methods appear under many names including first-passage distributions, Green's-function first-passage (GFFP), and Poisson kernels, and have been tabulated or approximated in several communities [Given et al. 1997; Hwang et al. 2001; Hwang and Mascagni 2003]. Closed-form kernels derived via the method of images are accurate but typically restricted to simple primitives [Given et al. 1997]. Recent work has also explored connections with conformal maps in 2D to derive closed-form formulas for simple primitives [Himmler and Günther

2025; Han et al. 2025], such as semicircles and "cake" domains. Our method complements these efforts by avoiding analytic derivations: we estimate first-passage statistics numerically on arbitrary subdomains using existing Monte Carlo estimators, typically *Walk on Stars* (WoSt). Qi et al. [2022] previously proposed to approximate Poisson kernels of arbitrary domains, but their approach relies on a biased finite-difference approximation arising from the use of reverse walks. We instead rely on "forward" exit statistics which provide unbiased point-to-point estimates at discretization points.

Approximating Kernels and Operators. Approximating scattering kernels and transport distributions is a recurring theme in graphics (e.g., tabulated BRDFs and BSSRDFs, microflake models), and similar ideas have recently been applied to PDE kernels and operators. Lu et al. [2025] approximate integral operators such as Poisson kernels with expressive analytic mixtures, while neural operator approaches aim to learn mappings between function spaces [Azizzadenesheli et al. 2024]. These approaches are most compelling when many queries share a fixed domain or a training distribution. Our focus is different: we target *composition across many subdomains* by discretizing local operators on collocation points and coupling them through a sparse Markov chain, enabling reuse while preserving geometric flexibility. This approach bears resemblance to works in rendering that precompute and compress transport operators. Blumer et al. [2016] build reduced per-asset aggregate scattering operators (ASO) using reduced bases. Zhao et al. [2013] precompute flux transfer matrices for repeated exemplar blocks and stitch them via a process called "Monte Carlo matrix inversion," which is analogous to our absorbing Markov chains. These methods, however, target radiative transfer, not elliptic PDEs, which allows several simplifications (e.g., isotropy and reciprocity). In particular, the Markov chain of Zhao et al. [2013] is only used to model long scattering paths, for which approximations are perceptually tolerable.

Discrete Markov Chains. In graphics, random-walk formulations have been used for image and mesh segmentation [Grady 2006; Lai et al. 2008], feature-preserving mesh smoothing and denoising [Sun et al. 2008], and conformal parameterization [Ben-Chen et al. 2008]. We leverage random walks in a different way: we construct a Markov chain whose states are collocation points on subdomain interfaces and estimate transition probabilities using continuous WoS/WoSt trajectories. We then build on the classical connection between Markov chains and potential theory—namely, that absorption probabilities solve discrete Dirichlet problems on graphs [Doyle and Snell 1984; Levin and Peres 2017]—to recover the unknown interface values of the global BVP.

Domain Decomposition. Domain decomposition (DD) [Toselli and Widlund 2004] methods accelerate or enable large-scale elliptic solves by partitioning the domain into subproblems, which are coupled either by matching solution values along shared boundaries (*non-overlapping* DD) [Liu et al. 2016] or by enforcing agreement of solutions within overlapping regions (*overlapping* DD) [Sellán et al. 2019]. These methods can handle very complex geometries by relying on background grids fine enough to capture geometric details [Liu et al. 2016], or on CSG-like combinations of primitives that are simpler to mesh individually or have been pre-meshed [Sellán et al. 2019], provided that a suitable decomposition is available. Our method is conceptually closest to non-overlapping DD in that (a) we solve local subproblems on tiles and (b) we enforce global consistency by solving a linear system on interface degrees of freedom. The key difference is *how* local interface couplings are obtained: rather than assembling them from a grid-based stencil or FEM basis, we estimate the relevant solution operators *stochastically* (Section 4.2) and then couple them deterministically through a discrete Markov chain on collocation points. This approach allows us to handle sub-tile geometry accurately without meshing the interior of subdomains. Furthermore, unlike Schur-complement methods, which often yield dense interface operators, our coupling is done only on subdomain

Fig. 4. By decomposing a complex urban scene into subdomains and encoding its intricate geometry into an induced random-walk system, our method characterizes the influence of buildings on wind patterns, visualized here as steady-state potential streamlines.

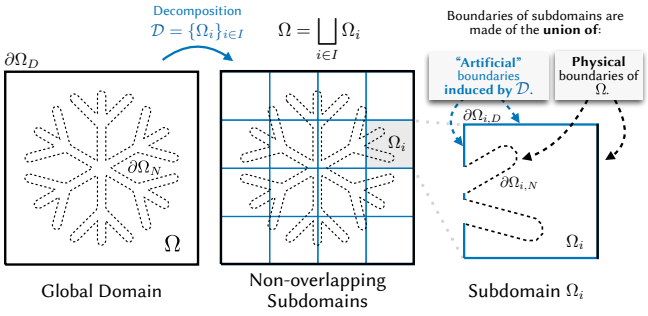


Fig. 5. Given a global domain Ω with Dirichlet boundary $\partial\Omega_D$ and Neumann boundary $\partial\Omega_N$ (left), we decompose Ω into *non-overlapping* subdomains $\mathcal{D} = \{\Omega_i\}_{i\in I}$ (middle). Each subdomain Ω_i inherits its boundary from the restriction of the original *physical* boundaries of Ω, together with additional "*artificial*" Dirichlet *interfaces* introduced by the decomposition.

interfaces—interior values are reconstructed via independent, per-domain matrix multiplications. This leads to sparser systems whose connectivity closely resembles that of finite-difference matrices.

Boundary Integral Methods. Boundary integral methods (e.g., BEM) have a rich history in graphics [Sugimoto et al. 2022, 2023; Hahn and Wojtan 2016]. Some of the challenges are dense linear systems, singular kernels, and the need for careful treatment of corners, non-smooth boundaries, and mixed boundary conditions. Recent work has significantly improved the practicality of BEM-style solvers [Chen et al. 2025]. Our method is related in spirit—we also exploit boundary-driven representations—but differs in two crucial ways. First, we do *not* assemble or solve a dense boundary integral equation: we estimate local operators via Monte Carlo sampling on subdomains, avoiding singular quadrature. Second, our global coupling is *sparse*: the interface unknowns live only on collocation points and are connected by a Markov chain whose transition weights are estimated locally.

Homogenization. Multiscale methods often replace fine-scale heterogeneity with an effective (homogenized) differential operator estimated from a representative volume element (RVE) under limiting assumptions (e.g., periodicity, scale separation) [Allaire 1992]. This philosophy underlies a number of pipelines for fabrication-oriented design of structured materials and microstructures [Schumacher et al. 2015; Panetta et al. 2015]. While efficient, homogenization can break down near boundary layers, sharp corners and non-smooth or non-periodic geometry—precisely the settings that arise when inclusions, voids, or complex boundaries drive the solution. Motivated by these limitations, our work aligns with a complementary viewpoint: instead of approximating the *differential operator* (by finding homogenized diffusion coefficients), we seek to approximate the *solution/integral operator* that maps boundary data to interior values. In our setting, this operator is approximated locally through first-passage statistics (Poisson kernels) and composed globally by coupling interfaces.

3 Overview

Problem setting. We consider elliptic boundary value problems (BVPs) with mixed Dirichlet–Neumann boundary conditions,

$$\begin{aligned} \Delta u(x) &= f(x) && \text{in } \Omega, \\ u(x) &= g(x) && \text{on } \partial\Omega_D, \\ \frac{\partial u(x)}{\partial n_x} &= h(x) && \text{on } \partial\Omega_N, \end{aligned} \tag{1}$$

where the boundary decomposes as $\partial\Omega = \partial\Omega_D \sqcup \partial\Omega_N$, and $\partial\Omega_D$ denotes the outer boundary of the domain (or a subset thereof), while $\partial\Omega_N$ represents internal inclusions or obstacles (Figure 5, left). While our formulation focuses on the homogeneous case $f = 0$ and $h = 0$, it already supports a wide range of applications (see Section 7). Extending the method to more general settings is left for future work.

Domain decomposition. To solve Problem (1), we adopt a domain decomposition approach and partition Ω into a collection of *non-overlapping* subdomains:

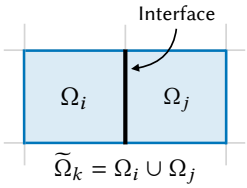
$$\mathcal{D} = \{\Omega_i\}_{i\in I}, \quad \Omega = \bigsqcup_{i\in I} \Omega_i,$$

where I indexes the subdomains (see Figure 5). Each subdomain Ω_i inherits part of its boundary from the original *physical* boundaries of Ω, carrying the corresponding Dirichlet or Neumann conditions. The remaining portions of $\partial\Omega_i$ consist of "*artificial*" Dirichlet boundaries introduced by the decomposition (see Figure 5, right). Concretely, the boundary of each subdomain Ω_i decomposes as $\partial\Omega_i = \partial\Omega_{i,D} \sqcup \partial\Omega_{i,N}$. Here, $\partial\Omega_{i,N}$ denotes the restriction of the physical Neumann boundary to Ω_i, while $\partial\Omega_{i,D}$ consists of the union of the physical Dirichlet boundary restricted to Ω_i and the artificial Dirichlet boundaries induced by the decomposition. These artificial Dirichlet boundaries define *interfaces* between subdomains. Throughout this paper, the term interface refers exclusively to these induced boundaries and not the physical Neumann boundary $\partial\Omega_N$.

Local BVP and solution operator. Our first key observation is that, for each subdomain Ω_i, we can define a *local* BVP. In other words, knowing the solution of (1) on its Dirichlet boundary $\partial\Omega_{i,D}$ suffices to locally reconstruct the solution inside Ω_i. This can be formalized as the local *solution operator* $\mathcal{H}_i$ that directly maps values on the boundary $\partial\Omega_{i,D}$ to interior values in Ω_i. This operator may be interpreted as the harmonic extension on the subdomain Ω_i, or, equivalently, as an integral operator with the Poisson kernel of Ω_i. The latter viewpoint is particularly interesting, as it gives $\mathcal{H}_i$ a *probabilistic interpretation*: it encodes the first-passage distribution of Brownian trajectories exiting Ω_i through $\partial\Omega_{i,D}$. Instead of repeatedly simulating long WoS/WoSt trajectories to evaluate interior solution values, $\mathcal{H}_i$ directly encodes the resulting boundary influence in *a single step*! Concretely, we construct a discrete matrix representation of $\mathcal{H}_i$ by tabulating its action between internal and boundary collocation points (see Section 4.2). Evaluating $\mathcal{H}_i$ then reduces to a simple matrix–vector product.

Unknown interface values and discrete Markov chain. In practice, interface values on $\partial\Omega_{i,D}$ are not known *a priori* and are themselves

part of the global solution of Problem (1). Our second key observation is that the coupling between interface values can itself be characterized *probabilistically*, in close analogy with Monte Carlo methods such as WoS/WoSt. The main intuition is that, instead of walking on spheres or star-shaped domains, one can directly *walk on interfaces* of neighboring subdomains. The corresponding "walk" probabilities can be derived from the same probabilistic interpretation of the local solution operators. However, this requires treating interfaces as *interior* regions of subdomains. With the non-overlapping decomposition $\mathcal{D}$, this is not possible, since interfaces are boundaries by construction. We therefore introduce another *overlapping* decomposition:



$$\mathcal{C} = \{\widetilde{\Omega}_k\}_{k\in K}, \quad \Omega = \bigcup_{k\in K} \widetilde{\Omega}_k,$$

called the *co-edge cover* (see Section 5.2), where each subdomain $\widetilde{\Omega}_k$ contains a single interface segment of the non-overlapping decomposition $\mathcal{D}$ in its interior, and is bounded by neighboring interface segments (see inset above).

For each subdomain $\widetilde{\Omega}_k$, we apply the same Monte Carlo strategy as above to estimate the corresponding solution operator, yielding transition probabilities between interface segments. After *discretization*, these probabilities define a discrete "absorbing" Markov chain (Section 5.1). Rather than explicitly simulating random walks in this Markov chain, we observe that we can directly solve for interface values via a linear system, producing deterministic estimates of the solution! After this global solve on the interfaces, we recover the interior solution for each subdomain by applying the precomputed interior operators $\mathcal{H}_i$.

Summary. The steps above can be summarized as follows: local solution operators on decomposed subdomains and overlapping co-edge domains are estimated independently, in parallel, using Monte Carlo methods, then coupled globally through a deterministic sparse linear system defined on interface collocation points. After solving for unknown values on the interfaces, the solution to problem (1) is recovered by locally applying tabulated interior solution operators. This separation between stochastic operator estimation and deterministic global solve allows us to combine the geometric flexibility of Monte Carlo methods with the robustness and efficiency of a discrete solve.

Generality. In practice, we use a regular tile-based decomposition for simplicity and its natural compatibility with parallelism (Sections 6 and 7). However, any non-overlapping decomposition equipped with a suitable parameterization of subdomain boundaries and an adjacency structure can be used. Throughout the paper, we focus on a two-dimensional construction for simplicity, but the formulation extends directly to three dimensions in a structurally identical manner.

4 Monte Carlo Estimation of the Poisson Kernel

In this section, we develop Monte Carlo approximations of the boundary-to-interior *solution operator* associated with the mixed Dirichlet–Neumann problem (1), or equivalently the Poisson kernel. Our approach relies on Monte Carlo sampling of *Walk on Spheres* (WoS) and *Walk on Stars* (WoSt) trajectories, which we first review from a boundary integral equation (BIE) perspective (Section 4.1). Next, we leverage these estimators to discretize solution operators on arbitrary subdomains (Section 4.2).

4.1 Background: Boundary Integral Equation and Monte Carlo Estimators

Grid-free Monte Carlo methods rely on two complementary interpretations. The first, rooted in stochastic calculus, represents the solution of a BVP as an expectation over random processes—typically Brownian motion—via the Feynman–Kac formula [Øksendal 2003]. The second approach, from potential theory, formulates the solution as a *recursive* integral equation that can be evaluated by sampling, analogous to the rendering equation [Kajiya 1986]. We will rely extensively on the latter interpretation.

Boundary integral equation. Considering Problem (1), one can show that for arbitrary sets $A \subseteq \Omega$ and $C \subseteq \mathbb{R}^n$ [Sawhney 2024; Hunter and Pullan 2017]:

$$\alpha(x)\,u(x) = \int_{\partial A} P^C(x,z)\,u(z) - G^C(x,z)\,\frac{\partial u(z)}{\partial n_z}\,dz + \int_A G^C(x,y)\,f(y)\,dy,$$

where $\alpha(x) = 1$ for $x \in A$, $\frac{1}{2}$ for $x \in \partial A$, and 0 otherwise. Here, $G^C(x,y)$ denotes the Green's function, i.e., the solution of $\Delta u(y) = -\delta_x(y)$ for $x \in C$ and $y \in A$, and represents the potential induced at y by a unit point source located at x. The Poisson kernel $P^C(x,z) := -\partial_{n_z} G^C(x,z)$ corresponds to the boundary flux generated by this point source and therefore quantifies how the boundary values contribute to the solution at interior points.

With no source term and zero Neumann conditions (Section 3), this reduces to:

$$\alpha(x)u(x) = \int_{\partial A} P^C(x,z)\,u(z)\,dz. \tag{2}$$

Observe that we can make a *completely arbitrary choice* of $A \subseteq \Omega$ and $C \subseteq \mathbb{R}^n$ in (2). This will become crucial in the next sections.

Walk on Spheres algorithm. For example, if $x \in \Omega$ and $A = C = B(x,R)$ is a sphere of radius R, then Equation (2) turns into the well-known mean value property [Axler et al. 2013]:

$$u(x) = \frac{1}{|\partial B(x,R)|}\int_{\partial B(x,R)} u(z)\,dz. \tag{3}$$

This directly gives the WoS algorithm. Given a point $x \in \Omega$, choose a sphere $B(x,R)$ of maximal radius R centered at x. The value $u(x)$ at x can be estimated by sampling the integral in Equation (3), namely by drawing a point z on $\partial B(x,R)$ and evaluating $u(z)$. Since $u(z)$ is also unknown, WoS repeats the strategy until reaching an ε-neighborhood of $\partial\Omega$ where $u(z) \approx g(z)$, which we know from the boundary conditions in Problem (1).

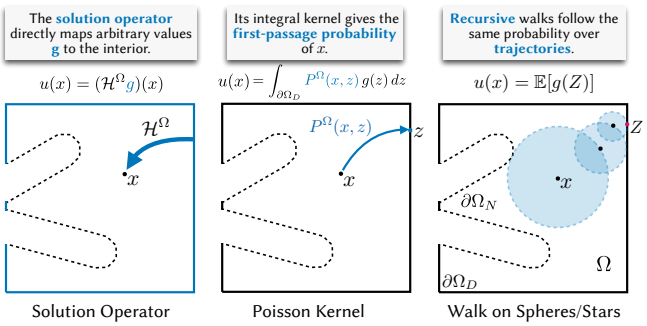


Fig. 6. The solution of Problem (1) admits several equivalent interpretations. The *solution operator* $\mathcal{H}^\Omega$ maps prescribed boundary values g directly to interior solution values (left). Its integral kernel, the *Poisson kernel* P^Ω, encodes first-passage probabilities to the Dirichlet boundary $\partial\Omega_D$, and thus describes a one-step walk to the boundary (middle). These quantities are generally not available in closed form. *Recursive* Monte Carlo estimators such as Walk on Spheres and Walk on Stars implicitly sample the Poisson kernel by drawing *multiple successive steps* until reaching the boundary, providing a practical way to estimate $\mathcal{H}^\Omega$ and equivalently P^Ω (right).

Walk on Stars algorithm. If $A = \mathrm{St}(x, R)$ is a star-shaped domain centered at x, and $C = B(x, R)$, Equation (2) becomes

$$\alpha(x)\, u(x) = \int_{\partial \mathrm{St}(x,R)} P^{B(x,R)}(x, z)\, u(z)\, dz, \tag{4}$$

which yields the Walk on Stars (WoSt) algorithm [Sawhney et al. 2023]. Since x is the center of $B(x, R)$, the kernel $P^{B(x,R)}(x, z)\, dz$ coincides with the *solid-angle* measure seen from x (see Section 4.4 of [Sawhney et al. 2023]). Consequently, the integral in Equation (4) can be estimated by Monte Carlo sampling: draw a random direction $\omega \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, where $\mathbb{S}^{d-1} = \{\, \omega \in \mathbb{R}^d : \|\omega\| = 1 \,\}$ denotes the unit sphere in $\mathbb{R}^d$, march along the ray defined by ω until its first intersection $z(\omega) \in \partial\mathrm{St}(x, R)$, and use $u\big(z(\omega)\big)$ as a Monte Carlo sample contributing to the estimate of $u(x)$. Since u is generally unknown at $z(\omega)$, WoSt applies this procedure recursively, generating a random walk that terminates when it reaches an ε-neighborhood of the Dirichlet boundary $\partial\Omega_D$, where $u(z) \approx g(z)$. We refer the reader to [Sawhney et al. 2023] for a complete description of the algorithm and its implementation.

4.2 Estimating Solution Operators with Monte Carlo

Our goal in this section is to construct a discrete approximation of the solution operator $\mathcal{H}^\Omega$ associated with the mixed Dirichlet–Neumann problem (1). To this end, we first recall a probabilistic interpretation of the boundary integral representation, which will motivate a Monte Carlo strategy for tabulating the operator.

Solution operator and boundary integral representation. With no source term and zero Neumann conditions (Section 3), the solution of Problem (1) admits the representation

$$u(x) = \int_{\partial\Omega_D} P^\Omega(x, z)\, g(z)\, dz, \tag{5}$$

where $\partial\Omega_D \subset \partial\Omega$ denotes the Dirichlet boundary. This formula defines a linear *solution operator* (harmonic extension) $\mathcal{H}^\Omega$ mapping

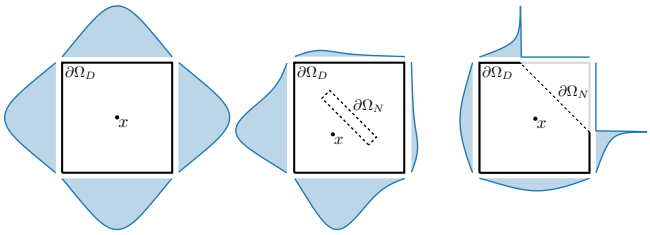


Fig. 7. Depicted in blue shade, different geometries and boundary conditions induce different Poisson kernels $P^\Omega(x, \cdot)$. For simple primitives such as a square with a fully Dirichlet boundary (left), the Poisson kernel exhibits a regular, smoothly varying structure dictated primarily by the symmetries of the domain, whereas introducing zero-Neumann boundaries, either enclosed (middle) or intersecting the Dirichlet boundary (right), breaks this structure and leads to significant, geometry-dependent variations.

Dirichlet boundary data to interior values via

$$(\mathcal{H}^\Omega g)(x) = \int_{\partial\Omega_D} P^\Omega(x, z)\, g(z)\, dz,$$

that directly gives interior values from the prescribed boundary data (see Figure 6, left). While P^Ω is available in closed form only for simple geometries, Equation (5) admits a probabilistic reinterpretation that enables stochastic estimation.

Probabilistic interpretation. For a fixed interior point x, the Poisson kernel $P^\Omega(x, \cdot)$ defines a probability density on $\partial\Omega_D$, which corresponds to the first-passage probability on $\partial\Omega_D$ (see Figure 6, middle). Figure 7 visualizes several Poisson kernels and illustrates the influence of Neumann boundaries on their structure.

Remarkably, *recursive* Monte Carlo estimators, such as WoS/WoSt, implicitly sample boundary locations according to the Poisson kernel $P^\Omega(x, z)\, dz$ (see Figure 6, right). Concretely, let $Z \in \partial\Omega_D$ denote the first point at which a WoS/WoSt trajectory, initiated at x, reaches the Dirichlet boundary (up to an ε-shell). Then the boundary integral representation (5) can be written equivalently as

$$u(x) = \mathbb{E}[g(Z)]\,, \tag{6}$$

where the expectation is taken over all WoS/WoSt trajectories. This provides us with a recipe to approximate the Poisson kernel for an arbitrary domain Ω: sample trajectories using WoS/WoSt and match their exit statistics. We provide a justification of this equivalence in Appendix A.

Kernel estimation by Monte Carlo sampling. We now exploit the probabilistic interpretation (6) to construct a discrete approximation of $\mathcal{H}^\Omega$. Let $\{z_j\}_{j=1}^m \subset \partial\Omega_D$ be collocation points with associated boundary panels $\{\Gamma_j\}_{j=1}^m$, such that $\partial\Omega_D \approx \bigsqcup_j \Gamma_j$ (see Figure 8, left). Launching S independent WoS/WoSt sample trajectories from x and recording their terminal Dirichlet hits $Z^{(k)}$, we estimate the probability mass of each panel as

$$\widehat{p}_j(x) := \frac{1}{S} \sum_{k=1}^{S} \mathbf{1}_{\Gamma_j}\big(Z^{(k)}\big), \qquad j = 1, \ldots, m,$$

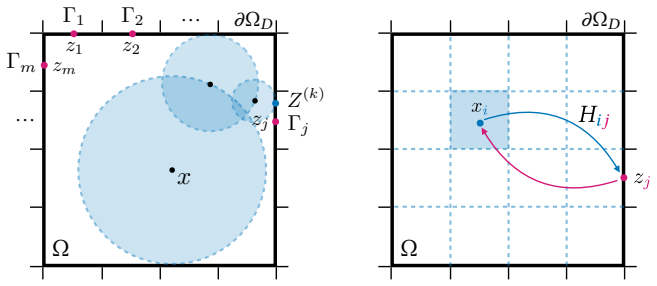


Fig. 8. For a fixed point $x \in \Omega$, we estimate the Poisson kernel $P^\Omega(x, \cdot)$ by discretizing the Dirichlet boundary $\partial\Omega_D$ into panels $\{\Gamma_j\}$ with collocation points $\{z_j\}$ (left). Each recursive walk that hits the boundary at $Z^{(k)}$ after k successive walks contributes probability mass to the corresponding panel. Discretizing the interior with points $\{x_i\} \subset \Omega$ leads to a matrix $\mathbf{H}$ that approximates the solution operator (right). Row-wise, H_{ij} represents first-passage probabilities from x_i to the boundary; column-wise, H_{ij} gives the interior response to a unit source at z_j.

where $\mathbf{1}_{\Gamma_j}$ denotes the indicator function of the boundary panel Γ_j, i.e., $\mathbf{1}_{\Gamma_j}(z) = 1$ if $z \in \Gamma_j$ and 0 otherwise. This approximates

$$\widehat{p}_j(x) \approx \int_{\Gamma_j} P^\Omega(x, z)\, dz.$$

A point-wise estimate of the Poisson kernel is then obtained via

$$P^\Omega(x, z_j) \approx \frac{\widehat{p}_j(x)}{|\Gamma_j|}.$$

Discrete solution operator. We now discretize the solution operator by a finite set of collocation points $\{x_i\}_{i=1}^n \subset \Omega$, and represent it with a matrix $\mathbf{H} \in \mathbb{R}^{n\times m}$ with entries

$$H_{ij} \approx \int_{\Gamma_j} P^\Omega(x_i, z)\, dz,$$

as shown in Figure 8 (right). For piecewise-constant boundary data $g(z) = \sum_j g(z_j)\mathbf{1}_{\Gamma_j}(z)$, the discrete solution operator is given by

$$(\widehat{\mathcal{H}}g)(x_i) = \sum_{j=1}^m H_{ij}\, g(z_j), \qquad i = 1, \ldots, n.$$

Thus, evaluating the harmonic extension for arbitrary boundary data reduces to a *single matrix–vector product.*

Note that although the figures in this section suggest a square, tile-based description, this choice is made solely to anticipate the practical implementation introduced in Section 6. The method itself is fully general and is not restricted to this representation.

Dual interpretation of the solution operator. As in the continuous setting, the matrix $\mathbf{H}$ admits two equivalent representations. Viewed column-wise, $\mathbf{H}$ acts as a discrete solution operator: for boundary data supported on a single panel Γ_j,

$$(\widehat{\mathcal{H}}\mathbf{1}_{\Gamma_j})(x_i) = H_{ij}, \qquad i = 1, \ldots, n,$$

so the j-th column corresponds to the harmonic extension of unit data on Γ_j. Viewed row-wise, $\mathbf{H}$ defines a discrete first-exit kernel on the boundary: for each interior point x_i, the entries satisfy

$$H_{ij} \geq 0, \qquad \sum_{j=1}^m H_{ij} \approx 1,$$

and H_{ij} represents the probability that a trajectory started at x_i first exits on panel Γ_j. Thus, $\mathbf{H}$ can be regarded either as a boundary-to-interior operator or as a row-stochastic first-exit probability distribution. We exploit this latter viewpoint in Section 5 to formulate a deterministic global solve over interface unknowns using absorbing Markov chains.

5 Discrete Solve over the Entire Domain

The previous section provides a recipe for approximating local solution operators $\mathcal{H}_i$ on each subdomain Ω_i of the global domain Ω, enabling the solution of boundary value problems restricted to individual subdomains. However, in practice, boundary values on $\partial\Omega_{i,D}$, including unknown interface values between subdomains, must be specified.

This section addresses the recovery of these interface values. We first recast the global interface problem in the language of absorbing Markov chains (Section 5.1), where simulating discrete random walks is equivalent to solving a linear system. We then show how to recover the weights on the edges of this Markov chain through a second set of discrete solution operators as before, this time defined on an overlapping *co-edge cover* of the domain (Section 5.2).

5.1 Background: Absorbing Markov Chains

Absorbing Markov chains have their roots in the study of random walks [Lawler and Limic 2010]. They provide a convenient finite-dimensional language for discrete harmonicity and Dirichlet boundary value problems, closely mirroring the first-exit interpretation used in WoS/WoSt and the Poisson-kernel viewpoint of Section 4.2.

Discrete-time Markov chains and the memoryless property. Let $(X_t)_{t\geq 0}$ be a discrete-time stochastic process on a finite state space $\mathcal{S}$. It is a (time-homogeneous) Markov chain if the conditional distribution of the next state depends only on the current state and not on the past history, i.e., it satisfies the *Markov (memoryless) property*

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, X_{t-1}, \ldots, X_0) = \mathbb{P}(X_{t+1} = j \mid X_t = i), \quad (7)$$

for all states $i, j \in \mathcal{S}$. We define the transition probabilities $P_{ij} := \mathbb{P}(X_{t+1} = j \mid X_t = i)$, and collect them into the transition matrix $\mathbf{P} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$. Equivalently, each row of $\mathbf{P}$ is a probability distribution: $P_{ij} \geq 0$ and $\sum_j P_{ij} = 1$. When $\mathcal{S}$ is equipped with an adjacency structure, the transition matrix $\mathbf{P}$ is typically sparse, with P_{ij} nonzero only for neighboring states. In this case, the Markov chain corresponds to a local random walk on a *directed* graph.

Absorbing chains. The state space $\mathcal{S}$ of a Markov chain can be decomposed as $\mathcal{S} = \mathcal{T} \sqcup \mathcal{A}$ into *transient* states $\mathcal{T}$ and *absorbing* states $\mathcal{A}$, where a state $a \in \mathcal{S}$ is *absorbing* if $P_{aa} = 1$, and *transient* otherwise (see Figure 9, top row, for an example). After reordering states, the transition matrix takes the block form [Kemeny and Snell 1976; Doyle and Snell 1984]

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (8)$$

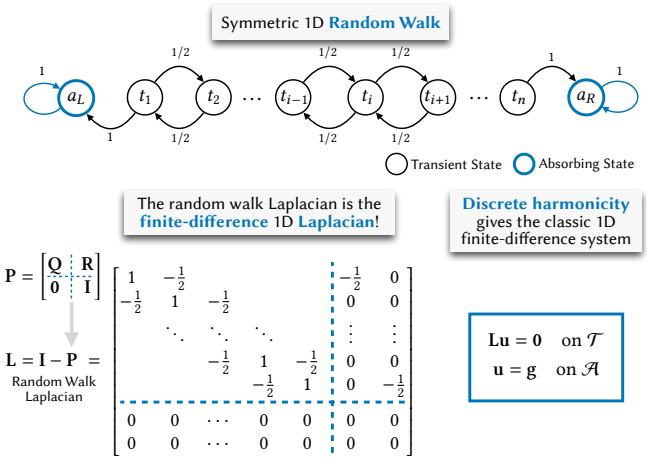


Fig. 9. The symmetric 1D random walk with absorbing boundary states $a_L, a_R \in \mathcal{A}$ is a canonical example of an absorbing Markov chain (top). Given the transition matrix $\mathbf{P}$ of this Markov chain, the corresponding random-walk Laplacian $\mathbf{L}$ coincides with the normalized 1D finite-difference Laplacian for a three-point, second-order stencil (bottom-left). Indeed, $\mathbf{P}$-harmonicity of the Markov chain (10) and the discrete random-walk Laplacian solve are equivalent (bottom-right).

where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{T}|}$ encodes transient-to-transient transitions, $\mathbf{R} \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{A}|}$ transient-to-absorbing transitions.

Given the prescribed Dirichlet boundary data $\mathbf{g} : \mathcal{A} \to \mathbb{R}$, define the first-hit stopping time

$$T := \inf\{t \geq 0 : X_t \in \mathcal{A}\},$$

and the value function

$$u(i) \;:=\; \mathbb{E}[g(X_T) \mid X_0 = i], \qquad i \in \mathcal{S}.$$

Thus, $u(i)$ is the expected boundary value observed when a walk started at i first hits the absorbing set. This is the discrete analog of the first-exit representation used by WoS/WoSt in Equation (6)! In the continuous setting, the boundary hit distribution is governed by the Poisson kernel, whereas here it is encoded by the hitting distribution of the chain on $\mathcal{A}$.

Averaging (harmonicity) and random-walk Laplacian. Conditioning on the first step and using the memoryless property (7) yields, for any transient state $i \in \mathcal{T}$, the Markov averaging property (see Appendix B for details):

$$u(i) = \sum_{j\in\mathcal{S}} \mathbb{P}(X_1 = j \mid X_0 = i)\, u(j) = \sum_{j\in\mathcal{S}} P_{ij}\, u(j), \qquad i \in \mathcal{T}, \quad (9)$$

together with the boundary condition $u(a) = g(a)$ for $a \in \mathcal{A}$. In operator form, $\mathbf{u}$ thus satisfies the fixed-point system

$$\mathbf{u} = \mathbf{P}\mathbf{u} \quad \text{on } \mathcal{T}, \qquad \mathbf{u} = \mathbf{g} \quad \text{on } \mathcal{A}, \quad (10)$$

i.e., $\mathbf{u}$ is $\mathbf{P}$-*harmonic* on the transient states. This formulation also admits a complementary *algebraic interpretation*. The transition matrix $\mathbf{P}$ naturally induces a *random walk Laplacian*

$$\mathbf{L} := \mathbf{I} - \mathbf{P},$$

such that u satisfies:

$$\mathbf{Lu} = \mathbf{0} \quad \text{on } \mathcal{T}, \qquad \mathbf{u} = \mathbf{g} \quad \text{on } \mathcal{A}.$$

As an example, Figure 9 illustrates this construction for a symmetric 1D random walk, highlighting its correspondence with the finite-difference Laplacian. Although purely algebraic, this equation directly mirrors the boundary value problem in Equation 1 and motivates the use of Markov chains as a model for information propagation across interfaces. Note that unlike standard combinatorial Laplacians (e.g., cotangent and finite-difference Laplacians), the random-walk Laplacian is generally *non-symmetric*, reflecting the directed and potentially non-reversible nature of the underlying Markov chain. We discuss this in more detail in Section 8.

Random Walks as Linear Systems. Writing (10) in block form using (8) results in the following

$$\mathbf{u}_{\mathcal{T}} = \mathbf{Q}\,\mathbf{u}_{\mathcal{T}} + \mathbf{R}\,\mathbf{g} \quad \Longleftrightarrow \quad (\mathbf{I} - \mathbf{Q})\,\mathbf{u}_{\mathcal{T}} = \mathbf{R}\,\mathbf{g}, \quad (11)$$

where $\mathbf{u}_{\mathcal{T}} \in \mathbb{R}^{|\mathcal{T}|}$ are the unknown transient values and $\mathbf{u}_{\mathcal{A}} = \mathbf{g} \in \mathbb{R}^{|\mathcal{A}|}$ are prescribed absorbing values. When absorption occurs with probability 1 from every transient state, $(\mathbf{I} - \mathbf{Q})$ is invertible and the solution is

$$\mathbf{u}_{\mathcal{T}} = \mathbf{N}\mathbf{R}\,\mathbf{g}, \quad (12)$$

where matrix $\mathbf{N} := (\mathbf{I} - \mathbf{Q})^{-1}$ is known as the *fundamental matrix* of the absorbing Markov chain [Kemeny and Snell 1976]. Its (i, j)-entry equals the expected number of visits to the transient state j before absorption, for a walk started at i.

Remarkably, (12) shows that the first-exit representation for the discrete Markov chain can be interpreted *purely algebraically*. The matrix

$$\mathbf{H}_{\text{global}} := \mathbf{N}\mathbf{R},$$

defines a *global* discrete solution operator mapping prescribed absorbing boundary values $\mathbf{g}$ to transient values $\mathbf{u}_{\mathcal{T}}$. In this sense, the expected first-exit values are not obtained by explicitly simulating random walks, but by directly applying the linear operator $\mathbf{H}_{\text{global}}$.

5.2 An Absorbing Markov Chain over Interfaces

We showed that discretized Dirichlet boundary value problems can be interpreted as absorbing Markov chains, leading to the linear system (12). Accordingly, the problem reduces to determining the transition matrix $\mathbf{P}$, which is the focus of this section.

The matrix $\mathbf{P}$ encodes both the adjacency structure of the graph and the associated transition probabilities between states. To define these consistently within the Markov-chain interpretation, we introduce an overlapping decomposition of the domain that allows interface degrees of freedom to be handled using the same local constructions developed in Section 4.2.

Overlapping cover. Let Γ denote the union of all interior interfaces induced by the decomposition $\mathcal{D}$. We construct an *overlapping cover*

$$\mathcal{C} = \{\widetilde{\Omega}_k\}_{k\in K}, \qquad \Omega \subset \bigcup_{k\in K} \widetilde{\Omega}_k,$$

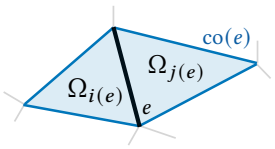
with the property that each interface segment $e \subset \Gamma$ lies in the *interior* of at least one element $\widetilde{\Omega}_k$, and that the boundary of each $\widetilde{\Omega}_k$ is composed entirely of interface segments. This construction ensures that, for any state associated with an interface, the corresponding

local stochastic process admits a well-defined first-exit distribution supported on the boundary of $\widetilde{\Omega}_k$, so that the resulting transition probabilities are *nonnegative* and *sum to one*.

The choice of the overlapping cover is *not unique*. To guide this choice, we impose two design principles: (i) the resulting transition matrix should be *sparse*, so that the global solve remains efficient, and (ii) transition probabilities should be *local*, ensuring that influence spreads through successive local transitions rather than artificial long-range jumps. Together with the stochastic consistency guaranteed by the cover construction above, these principles lead us to our desired transition matrix $\mathbf{P}$.

Co-edge subdomains. These requirements lead naturally to an overlapping cover based on *co-edges*. Let $\mathcal{E}$ denote the set of interface edges induced by the non-overlapping decomposition $\mathcal{D}$. For any interface edge $e \in \mathcal{E}$ shared by two adjacent subdomains $\Omega_{i(e)}$ and $\Omega_{j(e)}$, we define the associated *co-edge subdomain* as the union of these two subdomains,

$$\widetilde{\Omega}_e := \Omega_{i(e)} \cup \Omega_{j(e)}.$$

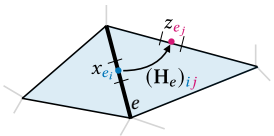


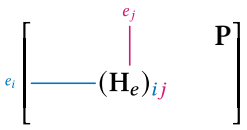
By construction, the interface edge e lies strictly in the interior of $\widetilde{\Omega}_e$. Moreover, the boundary of $\widetilde{\Omega}_e$ consists entirely of interface edges of the original decomposition. We denote by $\mathrm{co}(e)$ the union of all such boundary edges *except* e itself, and refer to them as the *co-edges* of e (shown in blue in the inset).

The collection $\{\widetilde{\Omega}_e\}_{e\in\mathcal{E}}$ forms an overlapping cover of Ω, which we call the *co-edge cover*. Each interface degree of freedom is surrounded by a minimal local region whose boundary consists only of neighboring interfaces, ensuring locality and sparsity while admitting a well-defined boundary value problem.

Defining the matrix $\mathbf{P}$. The co-edge cover enables a direct reuse of the solution-operator construction of Section 4.2 at the level of interfaces. We retain the same discretization of interface geometry and the same collocation points used in the parametrization of the internal solution operator.

For each interface edge e and each collocation point $x \in e$, we restrict WoS/WoSt to the co-edge subdomain $\widetilde{\Omega}_e$ and estimate the corresponding boundary hitting distribution on $\mathrm{co}(e)$. These first-exit probabilities define the transition weights from x to collocation points on the neighboring interface edges in $\mathrm{co}(e)$ and can be stored as a matrix $\mathbf{H}_e$. Neumann boundary conditions are incorporated in the same manner as described in Section 4.2 (see Figure 7).





As a result, the adjacency structure of the matrix $\mathbf{P}$ is determined by the co-edge relations, and the associated transition probabilities are obtained from the estimated local solution operators. The inset illustrates how the coefficients of a local operator $\mathbf{H}_e$ are scattered into the global matrix $\mathbf{P}$. Here, e_i and e_j denote the *global* indices of collocation points associated with edge e, while i and j refer to their *local* indices within $\mathbf{H}_e$.

Algorithm 1 Walk on Decomposed Subdomains (summary).

Input: Domain Ω with boundary partition $(\partial\Omega_D, \partial\Omega_N)$; non-overlapping decomposition $\mathcal{D} = \{\Omega_i\}$; associated co-edge cover $C = \{\widetilde{\Omega}_k\}$ (Section 5.2).
Output: Solution $\mathbf{u}$ of Problem (1), evaluated at interior collocation points.

1: **Discretize:** construct panels and collocation points on all subdomains Ω_i and interfaces $\partial\Omega_{i,D}$.
2: **Estimate subdomain solution operators:** for each $\Omega_i \in \mathcal{D}$, apply the Monte Carlo procedure of Section 4.2 to tabulate its interior solution operator $\mathbf{H}_i$.
3: **Estimate co-edge solution operators:** for each $\widetilde{\Omega}_k \in C$, apply the same Monte Carlo procedure to approximate the co-edge solution operator between the corresponding interfaces (Section 5.2).
4: **Assemble an interface Markov chain:** construct the absorbing Markov chain matrix $\mathbf{P}$ by the estimated co-edge solution operators $\mathbf{H}_e$ for each co-edge cover $\tilde{\Omega}_k$ (Section 5.2).
5: **Solve for interface values:** recover all unknown interface values by applying the global solution operator $\mathbf{H}_{\text{global}}$, obtained by solving the associated deterministic linear system (Section 5.1, Eq. (11)).
6: **Recover interior values:** for each $\Omega_i \in \mathcal{D}$, apply its tabulated operator $\mathbf{H}_i$ to obtain $\mathbf{u}$ at interior collocation points (Section 4.2).
7: **Return** $\mathbf{u}$ (optionally caching the tabulated operators for reuse across repeated solves and/or evolving geometries).

Although $\mathbf{H}_e$ is indexed locally, its entries contribute to $\mathbf{P}$ at the corresponding global locations. Together, these fully specify the matrix $\mathbf{P}$ governing the interface solve.

Once the interface values have been recovered by solving the linear system (11), all remaining interior degrees of freedom are reconstructed independently using the interior solution operators of Section 4.2. We summarize our procedure in Algorithm 1.

6 Regular Tile-based Decomposition

So far, the construction has not depended on the geometry of the domain decomposition. We now restrict our attention to regular tilings, where the domain Ω is partitioned into congruent subdomains. In particular, we consider square tilings in two dimensions, whose regular structure is amenable to parallelism, efficient bookkeeping, and scalable implementation. In Section 6.1, we describe this decomposition and the underlying choices of discretization. In Section 6.2, we provide a description of the Monte Carlo strategy to tabulate solution operators. Finally, Section 6.3 discusses how balancing Monte Carlo estimation of solution operators and decomposition provides a trade-off between parallelizable precomputation and serial deterministic solve.

6.1 Tile-based Decomposition and Uniform Collocation Points

Our decomposition $\mathcal{D}$ is a $T \times T$ set of square tiles (Figure 10, left), where the corresponding co-edge cover C can be easily constructed

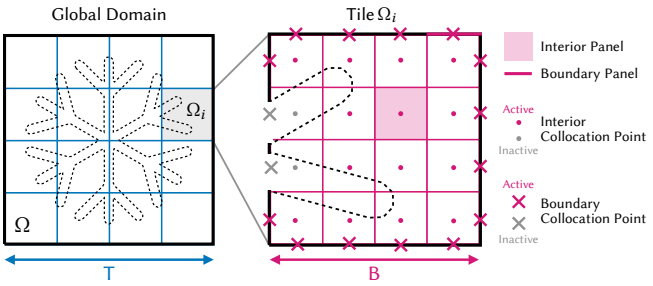


Fig. 10. The global domain Ω is decomposed into a $T \times T$ grid of tiles (left). Each tile Ω_i is discretized with B collocation points per axis: $4B$ boundary points and $B \times B$ interior points (right). Each collocation point is associated with a corresponding boundary or interior panel. Collocation points that fall inside the physical Neumann boundary are marked as invalid and excluded from the discretization: they are ignored both when launching Monte Carlo walks and when accumulating their contributions.

as described in Section 5.2. For each tile, we discretize the interior using B panels along each spatial dimension, and associate a collocation point with the center of each interior panel (Figure 10, right). The same parameter B is used to discretize the edges surrounding the tile into boundary panels, each equipped with a collocation point at its center. Consequently, each discretized local solution operator defines a linear map between boundary data and interior values. In particular, the interior operator

$$\widehat{\mathcal{H}}_{\text{inside}} : \mathbb{R}^{4B} \to \mathbb{R}^{B \times B}$$

maps discretized boundary values on the four edges of a tile to values on a $B \times B$ interior grid. Similarly, the co-edge operator

$$\widehat{\mathcal{H}}_{\text{co-edge}} : \mathbb{R}^{6B} \to \mathbb{R}^{B}$$

maps discretized boundary data on six boundary segments to B interior values along the corresponding interface. In practice, these operators are stored as matrices of size $B^2 \times 4B$ and $B \times 6B$, respectively. Put together, this means that for a given T and B, our solver's effective resolution along each dimension is $N = T \times B$. Readers should note that we will always compare against WoSt and finite differences with matching effective resolution N.

6.2 Monte Carlo Approximation of Local Solution Operators

For a given decomposition, we compute all interior and co-edge solution operators entirely *in parallel*. We refer to this process as *tabulation*. We use stratified sampling [Pharr et al. 2023] within each panel to sample the starting points of Monte Carlo walks. Upon termination, the exit locations are mapped to their corresponding panels by binning the boundary hit points. To account for the size of the tiles, the ε-shell in the Walk on Stars algorithm is rescaled locally. In particular, for a $T \times T$ tiling we use an *effective* $\overline{\varepsilon} := \varepsilon / T$ within each tile, which is essential for subtile precision, as tiles smaller than the ε-shell would otherwise trigger immediate termination.

In practice, local operators are computed only for subdomains that intersect Neumann boundaries (see Figure 16). Subdomains with purely Dirichlet boundaries can be precomputed and reused, while tiles that do not intersect the computational domain (e.g., those lying inside holes) are omitted. We also ignore *invalid* panels that lie *outside* of the computational domain (see Figure 10). Concretely, a collocation point and its corresponding panel are considered *valid* if its corresponding point lies at least a distance $\overline{\varepsilon}$ away from the Neumann boundary. We mask all invalid panels during tabulation: only *valid* collocation panels are considered to *launch* and *receive* Monte Carlo walks during tabulation. If a sample falls into an invalid collocation panel, we remap it to its nearest valid panel. This guarantees that the tabulated probability distributions sum to one. Within each valid interior panel (for interior operators) or along each valid boundary panel (for coedge operators), we use rejection sampling to ensure that samples lie inside the corresponding subdomain Ω_i (resp. $\widetilde{\Omega}_k$). This masking strategy is particularly simple to implement and works well for most cases. However, for very complex geometries at very high tiling resolutions, it may filter panels close to fine geometric features and introduce bias that propagates to the global solution. In Appendix E, we describe an alternative strategy that retains panels whose segment *partially* lies inside the domain, with negligible overhead. Figure 4 uses this strategy.

When assembling the Markov chain matrix $\mathbf{P}$ in Equation (8), we enforce stochasticity by renormalizing each row to sum to 1, compensating for numerical precision.

In the following, we denote by S_{pp} the *effective* number of Monte Carlo samples drawn per output point and by $\overline{W}$ the average length of Walk-on-Stars (WoSt) trajectories, measured as the number of steps taken before reaching the ε-shell. Since our total sample count also depends on interface collocation points, and not only on interior ones, we compare against Walk on Stars using an equal *effective* sample count, defined as the total number of samples used for interior and co-edge operators over the full domain, normalized by the number of valid output points.

6.3 Balancing Stochastic Estimation and Deterministic Solves

As illustrated in Figure 11, our method exposes an adjustable tradeoff between stochastic Monte Carlo estimation of solution operators (within each tile) and deterministic global solves by decomposing the output resolution as

$$N = T \times B,$$

where T denotes the number of tiles in the decomposition and B the number of Monte Carlo panels used to tabulate local solution operators along a given axis.

At one extreme, setting B=1 and T=N (or, more generally, B small compared to T) yields a method that behaves more like a grid-like solver and resembles a standard finite difference discretization. The global system contains $O(N^2)$=$O(T^2)$ degrees of freedom and, as we show in Section 7.2.2, exhibits the same asymptotic convergence rate as finite differences. Importantly, this similarity is only structural: the entries of the global system are not derived from local Taylor expansions, but from Monte Carlo–estimated solution operators computed on the true geometry. As a result, even in this regime, our method benefits from geometry-aware local operators

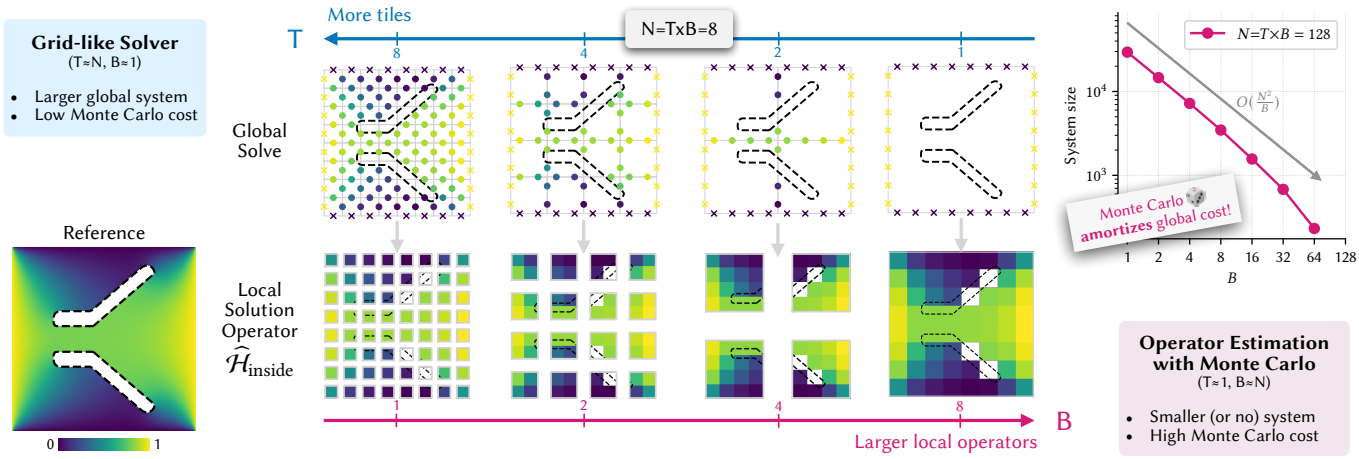


Fig. 11. At fixed output resolution $N = T \times B$, our method "interpolates" between the behavior of a grid-like solver and a pure solution operator approach that aggregates boundary-to-interior mappings through Monte Carlo estimation. This exposes a trade-off between the two stages of the solve: (1) the global solve over the interface Markov chain induced by co-edge transitions (top row), and (2) the local reconstruction via interior solution operators *independently for each tile* (bottom row). For large T, the global system is large with small tiles and therefore requires very short Monte Carlo walks to estimate its underlying operators. In this regime, the interior solution operators are relatively simple, as they effectively "average" nearby interface values. As B increases, the cost shifts toward estimating richer interior operators: global coupling is amortized over more distant interfaces, effectively reducing the system DoFs to $O(N^2/B)$ (top row and right plot), while Monte Carlo effort increases and concentrates on the local reconstruction step (bottom row).

that improve asymptotic constants relative to classical finite differences (see Figure 13). At the opposite extreme, setting T=1 and B=N recovers a purely Monte Carlo regime akin to Walk on Stars, where the interior solution operator $\widehat{\mathcal{H}}_{\text{inside}}$ is represented by a dense (and storage-heavy) mapping from boundary values to interior points, and classically requires $O(S_{\text{pp}}^{-1/2})$ samples per collocation point to converge. Intermediate choices of B and T span a spectrum of methods that all converge at the same spatial order when B remains small (see Figure 13), while differing in how computational effort is distributed between *parallelizable* stochastic estimation and *serial* deterministic solve.

This decomposition provides a principled way to balance accuracy, memory, and solver complexity. As shown in Figure 11 (right), increasing B shifts degrees of freedom from the interior of the domain to subdomain interfaces, resulting in a smaller global linear system whose size scales as $O(N^2/B)$ instead of $O(N^2)$. See Appendix C for a more thorough derivation and discussion on effective complexity and memory. This *amortizes* the cost of the serial global solve, but comes at the price of increased Monte Carlo effort, since each local operator requires more Monte Carlo samples to be estimated more accurately (see Section 7.2.1 for a discussion on the convergence cost as a function of B). Conversely, smaller values of B favor a more classical grid-based regime with a larger global system but minimal stochastic overhead and variance.

Crucially, the Monte Carlo work required to increase B is embarrassingly parallel: all local subdomain problems can be solved independently, and, for sufficiently smooth geometries, their number scales with the boundary complexity rather than the domain area (see Figure 16). This allows the method to trade serial computation in the global solve for parallel computation in the operator estimation stage. However, pushing B too far reintroduces the limitations of purely stochastic methods—most notably in Neumann-dominated problems, where variance grows prohibitively—indicating the existence of a problem-dependent sweet spot. *Our method makes this trade-off explicit and tunable!*

7 Results

We start by providing implementation details and the baselines used throughout our evaluation (Section 7.1). We then validate the core components and convergence of our approach in the two limiting regimes introduced earlier: a purely grid-free Monte Carlo setting and a grid-like dominated regime with minimal Monte Carlo overhead (Section 7.2). We then show in Section 7.3 when and how our method strikes a balance between grid-based and grid-free Monte Carlo solvers, and in which conditions it benefits from each. Finally, in Section 7.4, we present applications of our method to flow-based path planning and streamline visualization, as well as microstructure simulation, where it enables efficient large-scale solves while retaining sensitivity to fine geometric detail.

7.1 Implementation

We implement our Monte Carlo solver using `Dr.Jit` [Jakob et al. 2022], which allows us to execute the Walk on Stars (WoSt) algorithm in parallel on the GPU, across all subdomains. For geometric queries (e.g., ray intersection, closest-point queries, silhouette queries), we use the spatialized normal cone hierarchy (SNCH) from the `fcpw` library [Sawhney 2021]. More specifically, we rely on the implementation of Yu et al. [2025], which builds the hierarchy on the CPU while performing traversal on the GPU. All experiments and timings reported in this paper were conducted on a workstation equipped with an AMD Ryzen 9 7950X (16 cores, 32 GB RAM) and an NVIDIA RTX 4090 GPU (24 GB VRAM). For reference solutions, we

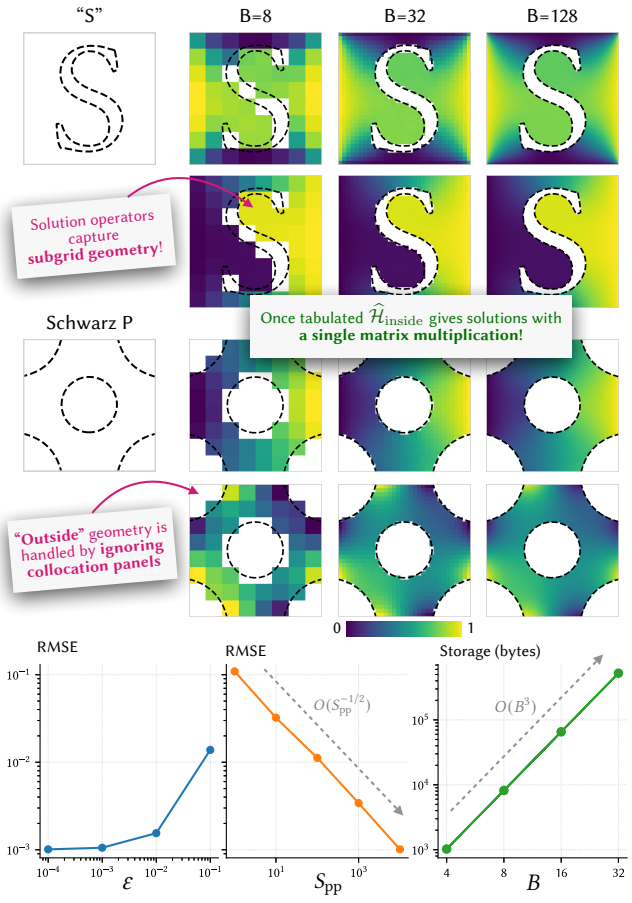


Fig. 12. Once tabulated, $\widehat{\mathcal{H}}_{\text{inside}}$ operators reconstruct solutions in the interior of each subdomain domain with a simple matrix-vector multiplication. As they are tabulated using "unbiased" WoSt (in the limit of the ε-shell), they accurately reflect the underlying geometry even at subgrid resolutions (e.g., "S" with B=8). "Outside" geometry is simply handled by ignoring these invalid collocation panels. Estimating solution operators with Monte Carlo naturally mirrors the properties of the underlying stochastic estimator (bottom), here WoSt: the error decreases as a function of the size of the ε-shell (left), and converges as $O(S_{\text{pp}}^{-1/2})$ (middle). With our point-wise collocation strategy, storage for the interior solution operator scales as $O(B^3)$ (right).

use the finite element method with a very fine discretization (see Figure 18 for representative discretization sizes). Finite element and homogenization baselines are implemented using `DOLFINx`, the recent backend of the `FEniCS` project [Alnæs et al. 2015; Baratta et al. 2024]. For FEM, we use piecewise linear triangular elements with a direct LU solver (`PETSc`, via `FEniCS`). Meshing is done with `gmsh` [Geuzaine and Remacle 2009] via the `MeshAdapt` [Remacle and Geuzaine 2022] algorithm. For homogenization, we use bilinear quadrilateral elements on a structured unit-cell mesh, following the two-scale strategy described in Appendix D. Finite differences are implemented from scratch and employ a first-order discretization of the Neumann boundary conditions in 2D, which is reflected in the convergence behavior shown in this section. Conceptually, this treats Neumann geometry as blocks depending on whether the center of each pixel lies in the domain or not. To solve both the FD system and our global interface system, we use `scipy.sparse.linalg.spsolve` (direct solver).

7.2 Validation

7.2.1 Monte Carlo estimation of solution operators (B=N). We start by validating our strategy to estimate solution operators (i.e., Poisson kernels) using Monte Carlo, as described in Section 4.2. Conceptually, this corresponds to the extreme case where the domain is made of *only one tile*, namely B=N and T=1. In this case, we only need $\widehat{\mathcal{H}}_{\text{inside}}$ to reconstruct the solution in Ω given the boundary values.

Since tabulation amounts to launching WoSt trajectories from collocation points and recording their exit locations, it remains fully consistent with the underlying Monte Carlo estimator. First, it is geometry-aware: even when the resolution is small, the solution is locally accurate and faithful to the actual *subgrid* geometry (e.g., the "trapped" solution in the "S" of Figure 12). Second, it converges similarly to WoSt: decreasing ε reduces bias and thus error (Figure 12, bottom), while convergence is naturally $O(S_{\text{pp}}^{-1/2})$.

As with any caching strategy, both storage and sampling costs grow with the discretization parameter B, but in a predictable manner. First, $\widehat{\mathcal{H}}_{\text{inside}}$ is stored as a $B^2 \times 4B$ matrix, leading to an $O(B^3)$ memory footprint (compared to $O(B^2)$ for $\widehat{\mathcal{H}}_{\text{co-edge}}$). This higher cost reflects the larger number of interior collocation points relative to interface points. Second, the Monte Carlo effort required to estimate these operators scales with the number of collocation points from which walks need to be started: for a per-panel sampling budget S_{pp}, the effective sampling cost grows as $O(B^2/S_{\text{pp}})$ for $\widehat{\mathcal{H}}_{\text{inside}}$ and $O(B/S_{\text{pp}})$ for $\widehat{\mathcal{H}}_{\text{co-edge}}$. Together, these two points motivate the decomposition strategy introduced in this paper: rather than tabulating a single large operator, we construct smaller local solution operators and embed them into a larger, but sparse, global system.

Note that although we have used the word "caching", this differs from existing grid-free Monte Carlo caching techniques [Li et al. 2023; Miller et al. 2023; Zhou et al. 2025] in the sense that we *cache the solution operator* and not the solution u itself. In our setting, given any new values for the boundary, we can solve directly for the interior using a single matrix multiplication and *without running any additional walks* or updating the cache!

7.2.2 Convergence of the discrete Markov chain ($T \approx N$). Following the strategy introduced in Section 5.2, we now investigate the discrete random-walk system solve that couples interfaces, for small resolution B relative to subdomain resolution T. This places us in a *regime analogous to a grid-like solver*, i.e., $T \approx N$, as illustrated in Figure 11 (left).

Figure 13 (left) shows that, for a fixed low value of B used to tabulate both interior and co-edge local operators, the method converges with the same rate as first-order finite difference methods, namely $O(1/T)$. The observed linear convergence is consistent with known results on random-walk/graph Laplacians [Singer 2006], which are typically first-order accurate unless the transition probabilities are constructed to satisfy additional symmetry (e.g., in the

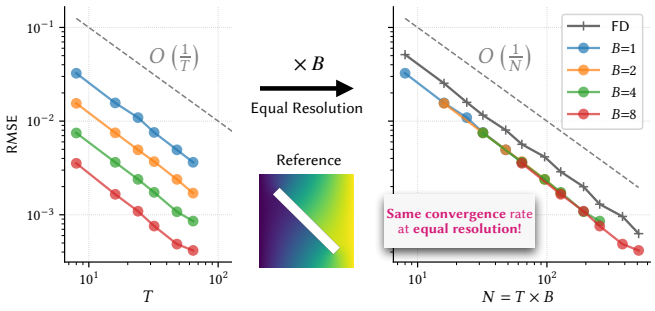


Fig. 13. For small (relative to T) and fixed resolution B used in the approximation of local operators, our method converges as $O(1/T)$, consistently with first-order discretizations of finite differences with Neumann boundaries. When matching resolutions (remember that the effective resolution of our method is $N{=}T{\times}B$), we see that all methods converge with the same constant regardless of the choice of B.

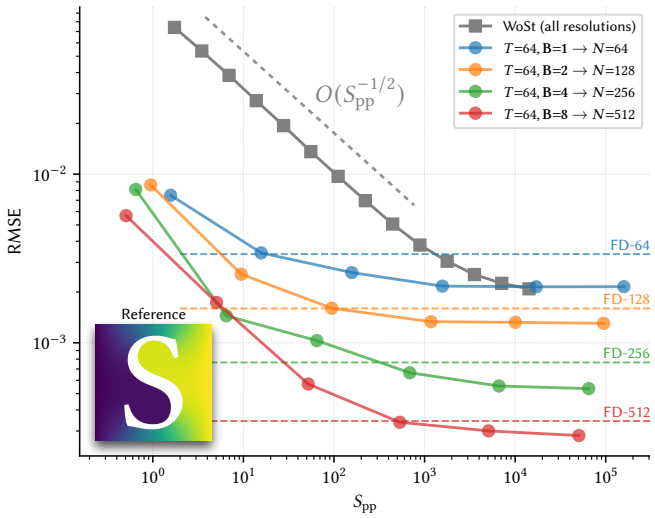


Fig. 14. By increasing the Monte Carlo budget (measured as the number of samples per point S_{pp}), our method rapidly matches and then improves upon the accuracy of finite differences at equal "effective" resolution $N{=}T{\times}B$ (dashed lines with matching colors). Notably, spatial decomposition allows us to mitigate the slow $O(S_{pp}^{-1/2})$ convergence of Walk on Stars (gray), which eventually plateaus due to the bias introduced by the ε-shell. At high sample counts, our method also exhibits a plateau, arising from the deterministic bias of discretizing local solution operators. Increasing B lowers this plateau but requires more samples due to the higher dimensionality of the discretized operators (see Section 7.2.1).

absence of Neumann boundary conditions) or higher-order consistency conditions [Gibou et al. 2002]. Since the effective output resolution of our method after applying interior solution operators is $N{=}T{\times}B$, we replot the curves as a function of this effective resolution in Figure 13 (right). Remarkably, the convergence rate becomes $O(1/N)$ regardless of the choice of B. This confirms that the underlying Markov chain remains consistent across different interface discretizations. Additionally, note that our method benefits from the geometric awareness of the Monte Carlo–estimated operators, which manifests as a favorable vertical shift relative to finite differences. Although these experiments demonstrate empirical convergence, providing formal convergence guarantees is beyond the scope of this paper.

7.3 Comparisons and Advantages

In this section, we analyze the trade-offs between our method, pure grid-based discretizations, and grid-free Monte Carlo methods, and identify the regimes in which our approach is most advantageous, depending on the chosen resolution, sampling budget, and computational paradigm.

7.3.1 When does Monte Carlo become beneficial over a standard grid-based approach? Figure 14 reports the solution error as a function of the stochastic budget allocated to our method, measured by the number of Monte Carlo samples *per point*, S_{pp}. In this plot, we fix the tiling to T=64 and vary only B, which controls the fidelity of the approximation of local solution operators. For our method, we count *all the samples* used in the tabulation of both the interior and the co-edge solution operators.

As a point-wise estimator, WoSt requires the same number of Monte Carlo samples *per point* regardless of the output resolution, which is why all resolutions collapse onto a single curve (in gray). Note first how WoSt converges at a rate of $O(S_{pp}^{-1/2})$ before plateauing due to the bias of the ε-shell and the minimum radius for star-shaped regions. Note that we use the same ε value between our method (within each subdomain) and WoSt. See Section 7.3.2 for a discussion of sample efficiency. We additionally show horizontal dashed lines indicating the error of a deterministic grid-based finite difference (FD) solve at *matched* resolution $T{\times}B$ that appears as a constant line because FD does not involve Monte Carlo sampling.

Reading the plot vertically (fixed samples-per-point budget) highlights that methods *with spatial discretization* (i.e., our method and FD) achieve substantially lower error than point-wise Monte Carlo at low sample counts, since sampling effort is amortized by (a) the coupling of the spatial discretization, and (b) the reuse of precomputed local operators for empty regions of the domain. Recall that only $\approx O(T)$ operators need to be precomputed (Section 7.3.3, Figure 16). As the sample count increases, our method reaches a plateau: once Monte Carlo variance in the estimation of local operators becomes negligible, the error becomes dominated by *deterministic bias from the discretization* of these local operators. Increasing B lowers this *asymptotic plateau* (i.e., improves the operator/discretization accuracy), but reaching the corresponding FD baseline requires more samples as B grows, consistent with the fact that discretized operators with a higher B represent a higher-dimensional quantity and are harder to estimate accurately with Monte Carlo samples as discussed previously in Section 7.2.1.

7.3.2 How does spatial decomposition help reduce Monte Carlo budget? It is well known that although Monte Carlo estimators such as Walk on Stars are unbiased, they suffer from excessive variance and slow convergence when Neumann boundaries dominate, mostly due to the presence of very long reflecting walks (see Figures 2, 15 and 18). Our method amortizes this cost by introducing *spatial decomposition* and *interface coupling*.

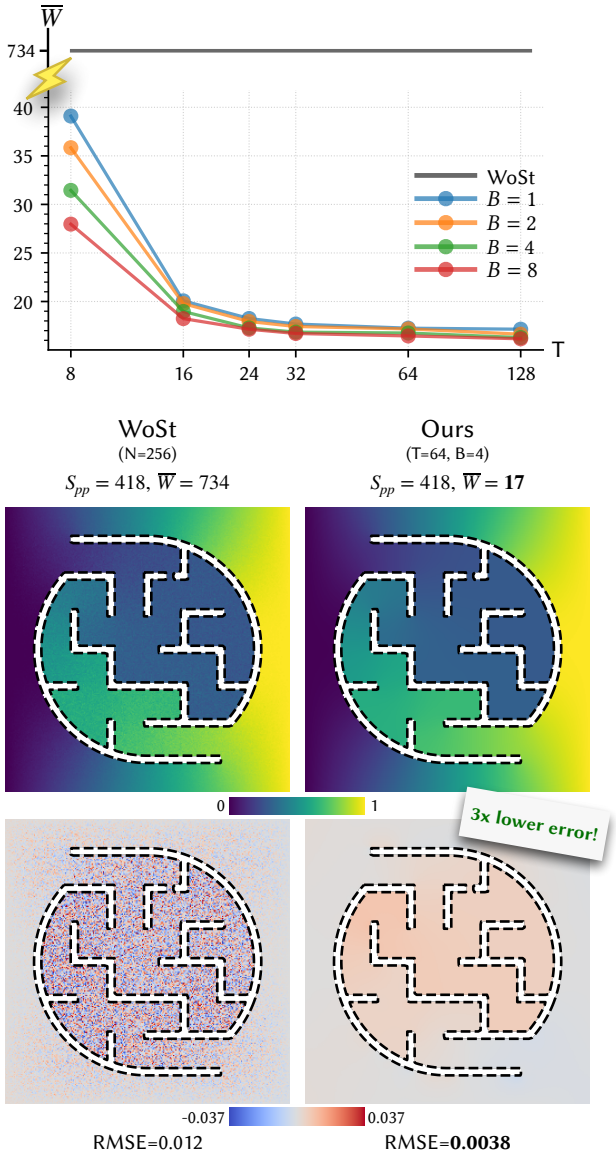


Fig. 15. By decomposing the domain into smaller tiles, our method rapidly and substantially reduces the average walk length, $\overline{W}$, in Neumann-dominated scenarios such as this maze (top). At moderately high tile counts T, this results in a 43x reduction in walk length, effectively amortizing the Monte Carlo cost and significantly reducing variance (see error map at the bottom). Finally, the global solve on interfaces further does not incur additional variance at the cost of a controlled bias (bottom).

First, decomposing the global domain into smaller subdomains results in Monte Carlo trajectories with substantially shorter paths, as quantified by the average number of walks per sample, $\overline{W}$ (Figure 15, top), yielding more than an order-of-magnitude reduction. In our setting, both the total number of samples per point, S_{pp}, and the average walk count $\overline{W}$ account for all samples used to estimate *both* the interior and the co-edge solution operators.

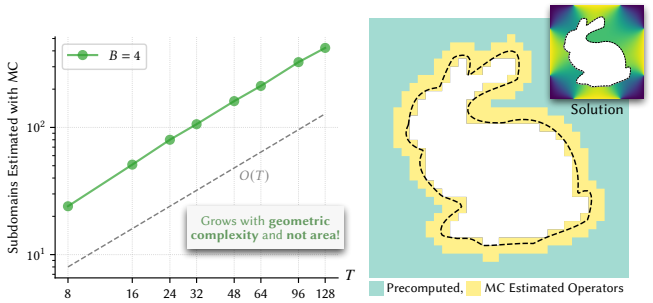


Fig. 16. Our method inherits the locality of grid-based approaches by requiring the estimation of solution operators (both interior and co-edge) only in regions near the geometry (right). Empty tiles share identical distributions and can therefore be precomputed once, independently of the target domain. As a result, the Monte Carlo cost scales with *geometric complexity* rather than domain size.

Second, introducing a global coupling via the interfaces between tiles leads to a global solve that does not incur additional variance at the cost of a controlled bias (Figure 15, bottom-right). At equal sample counts but with dramatically shorter walks (approximately 43× fewer), our method achieves a 3× reduction in error on the Neumann-dominated geometry shown in Figure 15.

7.3.3 Spatial decomposition enables locality and scales with geometric complexity. Although Monte Carlo methods excel at handling complex geometry, they typically require recomputing the solution from scratch whenever the geometry is modified—even very local changes! Existing caching approaches [Li et al. 2023; Miller et al. 2023; Zhou et al. 2025] cache *solutions* rather than solution operators, and therefore require a *full recomputation* once the underlying geometry changes. In contrast, grid-based methods benefit from an explicit spatial discretization: local geometric modifications only affect nearby elements (in FEM) or stencils (in FDM), and the resulting linear system can often be updated or re-solved efficiently by *exploiting the locality* of the discretization.

Our method inherits the same notion of locality. First, the number of solution operators that must be estimated grows with the *geometric complexity* and not the domain size (i.e., area), as shown empirically in Figure 16. Second, this structure allows solution operators to be recomputed only in regions where the geometry changes. This property is particularly advantageous in time-varying or design-oriented settings, where solutions must be re-evaluated repeatedly without recomputing them from scratch. We illustrate this in Figure 3.

7.4 Applications

Microstructure and Metamaterial Simulation. Our method and, in particular, its tile-based specialization, provide a natural way to simulate diffusion in microstructures and metamaterials. These structures are attractive because rich macroscopic behavior can often be induced by modulating only a few geometric parameters (see Figure 17 for an example with rotating panels). At the same time, faithfully simulating even seemingly simple patterns such as

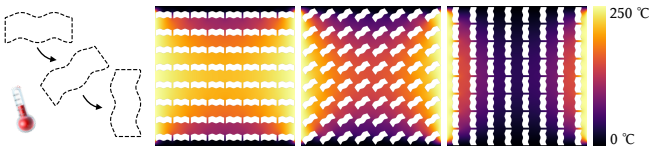


Fig. 17. Microstructures are a powerful design primitive: small geometric changes can induce large variations in the solution. In this example, rotating reflective panels controls the interior temperature of a heat exchanger with two opposing heat sources, on the left and right. Since our method tabulates solution operator statistics for a *single tile* once and reuses them across a large lattice, it enables rapid evaluation of large-scale solutions, making it well suited for inverse design and exploration.

the wave or dumbbell-shaped unit tiles in Figures 17 and 19, respectively, is challenging. Standard FEM pipelines require high-quality meshes, which become increasingly difficult to generate as features grow thin, oscillatory, or densely packed, and resolving subtile detail rapidly leads to prohibitively large discretizations (Figure 18, bottom row). Purely grid-free Monte Carlo methods (e.g., WoSt) avoid meshing, but for microstructures this tends to produce long, high-variance walks, resulting in noisy solutions and pathologically long runtimes (Figure 18, second row), which is especially problematic for downstream tasks such as streamline tracing that require smooth fields (Figure 19). Moreover, these runtimes make such methods impractical for inverse design or workflows that require iterating over multiple design variants.

A standard alternative for periodic materials is homogenization [Allaire 1992; Panetta et al. 2015; Schumacher et al. 2015] (see Appendix D for a full description), which replaces each unit cell by an effective (homogenized) diffusion tensor and enables large-scale simulation at negligible cost (Figure 18, top row). However, this efficiency comes from deliberately abstracting away fine geometry: homogenized models cannot recover the subtile, within-cell structure that is essential for applications like drawing streamlines in Figure 19, and their assumptions (periodicity and scale separation) can break down in the presence of boundary layers, defects, or non-repeated patterns. By contrast, our approach can be viewed as an *operator-based* analog of homogenization: instead of collapsing a cell into a single effective coefficient, we estimate (via first-passage statistics) how boundary information propagates through and *between* neighboring tiles, then compose these interactions through a deterministic global solve. Concretely, for a uniform tiling with a repeating unit cell, we only tabulate the interior solution operator for one cell and the co-edge transition operator on a representative two-cell patch; these cached operators can then be reused to assemble and solve systems on large lattices (Figure 18, third row), making preprocessing far more tractable than meshing pipelines whose cost grows rapidly with the number of cells. When the pattern is not perfectly repeated (e.g., varying periods, defects, or boundary-adjacent cells), we still benefit from locality and parallelism: only the affected tiles (and their co-edges) need to be (re)computed. Finally, as with WoS/WoSt, the geometry need not be meshed: our preprocessing relies only on geometric queries, allowing complex microstructures to be taken *as-is* while retaining subtile geometric influence in the resulting solution.

Flow-based path planning and streamline visualization. Our method enables rapid, low-variance solutions of boundary value problems on complex geometries, which naturally supports steady-state potential flow applications in environments that are both intricate and subject to changes.

Figure 4 shows how the geometric flexibility of Monte Carlo estimators can be leveraged to model a city-scale layout derived from real-world data and to visualize the influence of building placement and geometry on wind patterns. Figure 1 illustrates how our method can be used to obtain potential paths and help a robot navigate in a warehouse cluttered with complex geometry. When the geometry is modified, as in Figure 3, the locality induced by our domain decomposition allows the paths to be updated efficiently, without re-estimating the solution from scratch, in contrast to classical Monte Carlo approaches. Crucially, the deterministic global solve plays a central role in reducing variance and enabling reliable streamline tracing, whereas, at equal sample count ($S_{\text{pp}} = 109$ for Figure 1), pure Monte Carlo methods produce noisy solutions or would require substantially larger sample counts. Even at high resolution, our method produces *smooth* global solutions rapidly (9.6 seconds for Figure 1 and 23.2 seconds for Figure 4).

In all examples, we compute a potential field with Dirichlet values 0 on the left boundary, 1 on the right, and interpolated values along the horizontal axis, and trace streamlines using a standard midpoint ODE integrator. For Figures 1 and 3, we used $T = 128$, $B = 8$, and 1000 samples *per panel*. For Figure 4, we used $T = 96$, $B = 8$, and 2500 samples per panel.

8 Discussion and Limitations

Discretization and operator representation. Our method critically depends on how local solution operators are discretized and represented. We chose to adopt a point-wise collocation strategy, explicitly tabulating boundary-to-interior mappings at discrete panels. A natural alternative is to represent these operators in a smooth functional basis, rather than exhaustively storing their action at collocation points associated with piecewise-constant panels. While approximating *solutions* in function bases is a well-studied topic, directly approximating *solution operators* remains comparatively unexplored, yet promising. Introducing additional structure—such as assumptions on the smoothness of the mapping, as recently explored for solutions [Zhou et al. 2025] (rather than operators!)—could substantially reduce the number of Monte Carlo samples required by our current point-wise approach. Classical domain decomposition methods often rely on such representations, for instance, through spectral or hierarchical bases [Martinsson 2015, 2019]. However, transferring these ideas to our setting is non-trivial: complex geometry and mixed boundary conditions induce sharp variations and discontinuities in the associated Poisson kernels (Figure 7, right), which challenge standard smooth bases.

Adaptivity and decomposition strategies. A complementary direction is adaptivity in both the discretization and the decomposition. Adaptivity could be introduced at multiple levels, for example by using larger subdomains in regions far from geometry, akin to tile-based adaptivity strategies used in fluid simulation [Losasso et al. 2004; Goldade et al. 2019]. More general subdivision units, or even

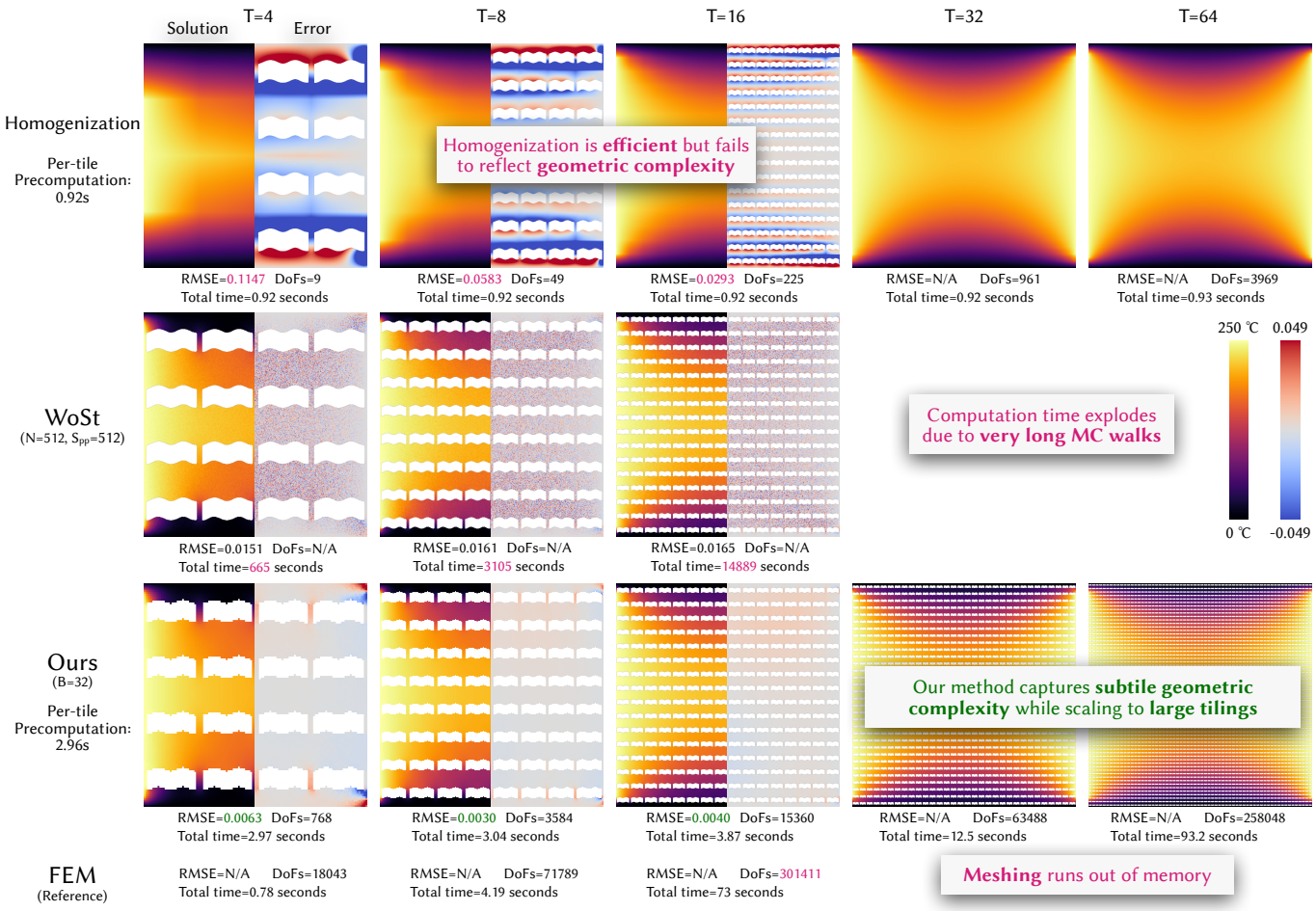


Fig. 18. Large periodic microstructures are particularly challenging to simulate. Homogenization (top row) is appealing for its efficiency: each tile is reduced to a single element, enabling near-instant solves even for large lattices (right). However, at low or intermediate resolutions it fails to capture the true geometry, leading to large errors as the solution is effectively interpolated. Walk on Stars (second row) struggles with Neumann boundaries, resulting in very long Monte Carlo walks and high variance, which makes large lattices impractical (already over 4 hours for 16 tiles). FEM (bottom row), which we use as a reference solution, is robust but requires costly meshing and fails beyond 16 tiles due to an out-of-memory error (with the setup described in Section 7.1). Our method (third row) strikes a trade-off: we precompute solution operators for a single tile (2.96 s), whose statistics are then duplicated in the global discrete Markov chain; the online cost is therefore dominated by the linear solve. As everywhere else in the paper, we do not interpolate our solution below the resolution of the collocation panels, even though interpolation would visually smooth the result. For homogenization, where each tile is modeled as a single quad abstracting away geometry, we interpolate to the reference resolution for comparison; errors are then evaluated only on the physical domain by masking with the true geometry.

hybrid approaches that locally introduce meshing, could also be considered. At the same time, a central motivation of our approach is to avoid resolving geometric complexity through meshing. Instead, we decompose the domain into simple, fixed subdomains and capture the interaction between geometry and the PDE exclusively through statistics estimated with Monte Carlo, thus leveraging the geometric flexibility of grid-free methods while mitigating their limitations via global coupling.

Domain decomposition revisited via absorbing Markov chains. Although we define a discrete Markov chain only on the interfaces of the decomposition, absorbing Markov chains provide a unifying interpretation of our two-stage strategy. One could indeed construct a single augmented Markov chain that also includes interior collocation points as transient states, with transitions to the surrounding interfaces defined by the interior solution operator $\widetilde{\mathcal{H}}_{\text{inside}}$. In principle, this formulation would allow all unknowns (interface and interior) to be solved simultaneously as part of one global system.

However, the interior states in this augmented chain have a very particular structure: once a walk leaves an interior state and reaches an interface, it never returns to the interior of the same subdomain. In the Markov chain literature, such states are sometimes referred to as *ephemeral* (illustrated in pink in the inset). From the perspective of the global interface problem, these states

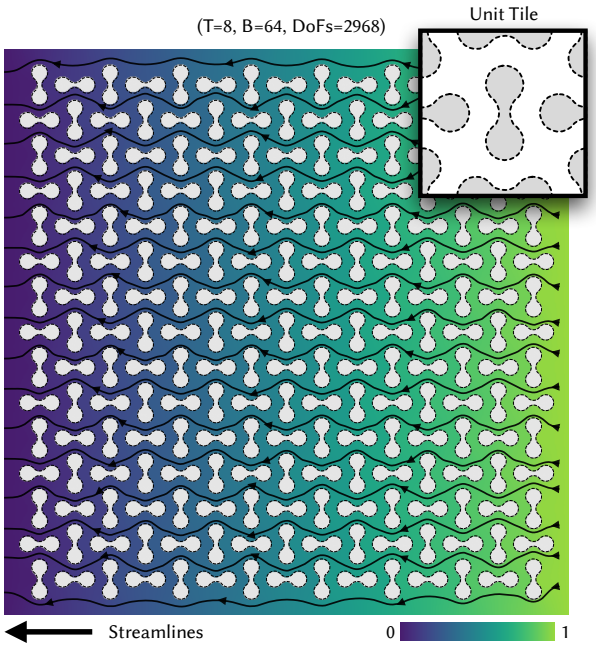
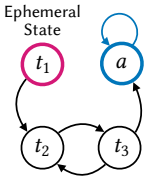


Fig. 19. Our method captures fine-grained subtile complexity to simulate streamlines in complex geometry. In this example, we use a high B to capture subtile geometric details. The *one-time* precomputation of solution operators for the reference pattern (top right, ≈ 8 min) amortizes the global solve, as these operators are reused to construct a large lattice. As discussed in Section 6.3, increasing B reduces the number of degrees of freedom for a fixed output resolution ($O(N^2/B)$), at the cost of higher Monte Carlo cost, which explains the longer precomputation time. At matched resolution ($T{\times}B = 512$), finite differences would require 178 444 DoFs, while meshing for FEM would produce 57 558 vertices. Note that to account for boundary repeatability effects, we precompute operators on a 3×3 lattice and dilate them to form the 8×8 lattice shown.

therefore introduce no additional coupling: they only result in one-way transitions to interfaces and can be eliminated without affecting the solution on interface states. This observation leads to a principled reduction of the system to interface states only, reducing the number of degrees of freedom from $O(T^2B^2)$ to $O(T^2B)$.



The reduced system, as shown in Section 5.1, corresponds to a random-walk Laplacian solve

$$\mathbf{u}_{\mathcal{T}} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R}\,\mathbf{g} = -\mathbf{L}_{\mathcal{T}\mathcal{T}}^{-1}\mathbf{L}_{\mathcal{T}\mathcal{A}}\,\mathbf{g},$$

expressed in terms of the transient–transient and transient–absorbing blocks of the Laplacian associated with the decomposition of the reduced states $\mathcal{S} = \mathcal{T} \sqcup \mathcal{A}$. With the interface values recovered, the previously eliminated interior degrees of freedom are obtained independently on each subdomain. Consistent with the solution-operator formulation described in Section 4.2, each such recovery actually amounts to a *degenerate* absorbing Markov chain in which interior states transition directly to absorbing interface states as

$$\mathbf{u}_{\mathcal{T}} = \mathbf{R}\mathbf{g} = -\mathbf{L}_{\mathcal{T}\mathcal{A}}\,\mathbf{g}.$$

Symmetry, reversibility, and link to geometry and discretization. We begin by examining the structure of the random-walk solve (8). As in the criteria discussed by [Wardetzky et al. 2007] for a discretized Laplacian, this system exhibits (i) locality, (ii) positive weights, and (iii) empirically observed convergence (see Figure 13). Since $\mathbf{Q}$ is a substochastic matrix and absorption occurs almost surely, the block $\mathbf{L}_{\mathcal{T}\mathcal{T}}$ is nonsingular and the associated Dirichlet problem admits a unique solution. A natural question is whether the system also enjoys (iv) symmetry and (v) positive (semi-)definiteness.

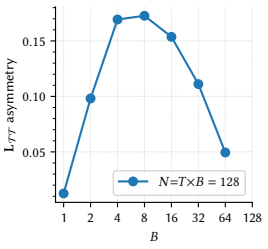
These additional structures arise when the chain is *reversible*. A Markov chain with transition matrix $\mathbf{P}$ on a finite state space $\mathcal{S}$ is reversible with respect to a positive measure $\pi : \mathcal{S} \to \mathbb{R}_{>0}$ if it satisfies the *detailed balance* condition

$$\pi_i P_{ij} = \pi_j P_{ji} \qquad \forall i, j \in \mathcal{S}.$$

Restricting this to $i, j \in \mathcal{T}$ implies that $\mathbf{L}_{\mathcal{T}\mathcal{T}}$ is self-adjoint with respect to the π-weighted inner product $\langle u, v\rangle_\pi := \sum_{i\in\mathcal{T}} \pi_i u_i v_i$. Moreover, since absorption occurs almost surely, $\mathbf{L}_{\mathcal{T}\mathcal{T}}$ is positive definite in this inner product. An example of this situation is random walks on *undirected* weighted graphs (e.g., finite-difference or cotangent Laplacians).

When the domain contains no obstacles, our method can exploit this additional structure; however, in the general setting the resulting linear system is asymmetric. This asymmetry arises because co-edge transitions are estimated on overlapping covers whose interior geometry varies across interfaces. As a result, the forward transition from an interface panel—estimated on a cover $\widetilde{\Omega}_e$—and its reverse counterpart—estimated on distinct covers leading to e—are not, in general, reciprocal.

We quantify this effect for matrix $\mathbf{L}_{\mathcal{T}\mathcal{T}}$ using relative asymmetry $\eta_{\text{asym}}(\mathbf{A}) := \|\mathbf{A}-\mathbf{A}^\top\|_F \,/\, \|\mathbf{A}\|_F$ where $\|\cdot\|_F$ designates the Frobenius norm. As shown in the inset, increasing B for a fixed N results in a gradual loss of symmetry in the global linear system, which is then recovered as B approaches N. In practice, the degree of asymmetry is governed by the block size B. For small B and large T, the system exhibits only mild asymmetry. In this regime, tiles closely conform to the underlying geometry, resulting in nearly reciprocal transitions in the induced Markov chain. As B increases and T decreases, the transient coupling between interface panels diminishes, so the matrix $\mathbf{L}_{\mathcal{T}\mathcal{T}}$ becomes increasingly symmetric. Although this asymmetry does not affect correctness—the system remains invertible—it prevents the direct use of robust sparse symmetric positive definite solvers. Improving the performance of this asymmetric solve is left for future work.



9 Conclusion and Future Work

In this paper, we take a first step toward reconciling grid-free Monte Carlo methods with grid-based solvers for elliptic boundary value problems. Our approach builds on the observation that grid-free

Monte Carlo methods excel at producing unbiased estimates of solutions—or, as we show in this work, of solution operators—but often suffer from slow convergence rates, even in relatively simple scenarios. Deterministic grid-based solvers, in contrast, offer robust, well-understood convergence behavior, yet their computational cost grows prohibitively with geometric complexity.

The main contribution of our work is to make explicit a direct analogy between continuous boundary value problems and their discrete counterparts. In the continuous setting, solutions of elliptic PDEs admit boundary integral representations whose Poisson kernels encode first-exit distributions of stochastic processes. In the discrete setting, absorbing Markov chains provide an algebraic analogue: harmonic functions arise as fixed points of a transition operator and can be recovered by solving a sparse linear system. Our key idea is to bring the geometry-awareness of first-passage statistics into this discrete framework by constructing transition probabilities from Monte Carlo–estimated Poisson kernels.

To this end, the states of the Markov chain must correspond to geometrically meaningful regions on which first-passage statistics can be estimated reliably. This requirement naturally leads to a decomposition of the domain into subdomains, with random walks proceeding by hopping between them through their interfaces, leading to what we call *walks on decomposed subdomains*! In practice, local solution operators are estimated stochastically on simple subdomains, where recursive Monte Carlo estimators like Walk on Spheres/Stars are most effective, while global consistency is enforced deterministically through a sparse system that couples interfaces and whose structure mirrors that of classical grid-based discretizations. Sparsity, locality, and efficient global coupling—hallmarks of grid-based solvers—emerge here not from an explicit mesh or stencil, but from the adjacency structure of the transition probability matrix induced by the (overlapping) domain decomposition, while its weights encode geometric information through first-passage probabilities. While we focus on a particular, well-motivated construction based on co-edge subdomains, the proposed framework is more general and suggests a family of decompositions and couplings yet to be explored.

In its concrete tile-based realization, our method exposes a continuous spectrum of computational regimes. At one extreme, it resembles the caching of solution operators, closely mirroring traditional estimation of solutions using recursive Monte Carlo estimators such as Walk on Stars, which are highly parallelizable but lack information sharing due to the absence of global coupling. At the other extreme, it approaches a classical deterministic solve with a fixed discretization bias. Between these endpoints lies a rich space of trade-offs, balancing variance, bias, parallelism, and global solve cost depending on the target resolution, sampling budget, and application. Crucially, no volumetric meshing is required: geometric complexity is fully absorbed into local operators, enabling subgrid accuracy and efficient re-solves when geometry evolves.

Our work opens several directions for future research. Extending the framework to more general boundary conditions (e.g., non-zero Neumann or Robin conditions), source terms, and broader classes of elliptic operators (e.g., spatially varying coefficients) is a natural next step. Introducing additional structure into the approximation of solution operators may further reduce Monte Carlo effort and storage requirements. More broadly, the close connection to domain decomposition suggests multilevel and hierarchical constructions, in which solution operators are nested across scales.

Acknowledgments

We thank the reviewers for their insightful feedback. We are grateful to Pavle Konaković for his help with the design and rendering of 3D scenes. We thank Rohan Sawhney, Bailey Miller and Hamid Kamkari for valuable discussions. This work was supported by the Wistron Corporation, the Siebel Scholars program and the MIT Generative AI Impact Consortium.

References

Grégoire Allaire. 1992. Homogenization and two-scale convergence. *SIAM Journal on Mathematical Analysis* 23, 6 (1992), 1482–1518.

Martin S. Alnæs, Jan Blechta, Johan Hake, Andreas Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. 2015. The FEniCS Project Version 1.5. *Archive of Numerical Software* 3, 100 (2015). doi:10.11588/ans.2015.100.20553

Sheldon Axler, Paul Bourdon, and Ramey Wade. 2013. *Harmonic function theory*. Vol. 137. Springer Science & Business Media.

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. 2024. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics* 6, 5 (2024), 320–328.

Ghada Bakbouk and Pieter Peers. 2023. Mean Value Caching for Walk on Spheres. (2023).

Anchang Bao, Jie Xu, Enya Shen, and Jianmin Wang. 2025. Off-Centered WoS-Type Solvers with Statistical Weighting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. 2024. DOLFINx: Next generation FEniCS problem solving environment. https://github.com/FEniCS/dolfinx. Version used: 0.10.0.

Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. 2008. Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 449–458.

Ilia Binder and Mark Braverman. 2012. The rate of convergence of the walk on spheres algorithm. *Geometric and Functional Analysis* 22, 3 (2012), 558–587.

Adrian Blumer, Jan Novák, Ralf Habel, Derek Nowrouzezahrai, and Wojciech Jarosz. 2016. Reduced aggregate scattering operators for path tracing. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 461–473.

Jiong Chen, Florian T Schäfer, and Mathieu Desbrun. 2025. Lightning-fast Boundary Element Method. *ACM Transactions on Graphics* 44, 4 (2025).

Peter G Doyle and J Laurie Snell. 1984. *Random walks and electric networks*. Vol. 22. American Mathematical Soc.

Lawrence C Evans. 2022. *Partial differential equations*. Vol. 19. American mathematical society.

Christophe Geuzaine and Jean-François Remacle. 2009. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering* 79, 11 (2009), 1309–1331.

Frederic Gibou, Ronald P Fedkiw, Li-Tien Cheng, and Myungjoo Kang. 2002. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *Journal of computational physics* 176, 1 (2002), 205–227.

James A Given, Joseph B Hubbard, and Jack F Douglas. 1997. A first-passage algorithm for the hydrodynamic friction and diffusion-limited reaction rate of macromolecules. *The Journal of chemical physics* 106, 9 (1997), 3761–3771.

Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. 2019. An adaptive variational finite difference framework for efficient symmetric octree viscosity. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Leo Grady. 2006. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 28, 11 (2006), 1768–1783.

David Hahn and Chris Wojtan. 2016. Fast approximations for boundary element based brittle fracture simulation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Qiansheng Han, Antti Rasila, and Tommi Sottinen. 2025. Efficient simulation of mixed boundary value problems and conformal mappings. *Appl. Math. Comput.* 488 (2025), 129119.

Paul Himmler and Tobias Günther. 2025. Conformal First Passage for Epsilon-free Walk-on-Spheres. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–11.

Thomas JR Hughes. 2003. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.

Peter Hunter and Andrew Pullan. 2017. Fem/bem notes. (2017).

Chi-Ok Hwang, James A Given, and Michael Mascagni. 2001. The simulation–tabulation method for classical diffusion Monte Carlo. *J. Comput. Phys.* 174, 2 (2001), 925–946.
Chi-Ok Hwang and Michael Mascagni. 2003. Analysis and comparison of Green's function first-passage algorithms with "Walk on Spheres" algorithms. *Mathematics and computers in simulation* 63, 6 (2003), 605–613.
Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. Dr. jit: A just-in-time compiler for differentiable rendering. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–19.
James T Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.
John G. Kemeny and J. Laurie Snell. 1976. *Finite Markov Chains*. Springer.
Yu-Kun Lai, Shi-Min Hu, Ralph R Martin, and Paul L Rosin. 2008. Fast mesh segmentation using random walks. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*. 183–191.
Gregory F Lawler and Vlada Limic. 2010. *Random walk: a modern introduction*. Vol. 123. Cambridge University Press.
Randall J LeVeque. 2007. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM.
David A Levin and Yuval Peres. 2017. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc.
Zilu Li, Guandao Yang, Xi Deng, Christopher De Sa, Bharath Hariharan, and Steve Marschner. 2023. Neural caches for monte carlo partial differential equation solvers. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.
Haixiang Liu, Nathan Mitchell, Mridul Aanjaneya, and Eftychios Sifakis. 2016. A scalable schur-complement fluids solver for heterogeneous compute platforms. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–12.
Frank Losasso, Frédéric Gibou, and Ron Fedkiw. 2004. Simulating water and smoke with an octree data structure. In *Acm siggraph 2004 papers*. 457–462.
Haolin Lu, Yash Belhe, Gurprit Singh, Tzu-Mao Li, and Toshiya Hachisuka. 2025. Gaussian Integral Linear Operators for Precomputed Graphics. *ACM Trans. Graph.* 44, 6 (2025). doi:10.1145/3763321
PG Martinsson. 2015. The hierarchical Poincaré-Steklov (HPS) solver for elliptic PDEs: A tutorial. *arXiv preprint arXiv:1506.01308* (2015).
Per-Gunnar Martinsson. 2019. *Fast direct solvers for elliptic PDEs*. SIAM.
Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Transactions on Graphics* 42, 4 (July 2023), 1–11. doi:10.1145/3592400
Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024. Walkin'robin: Walk on stars with robin boundary conditions. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–18.
Mervin E Muller. 1956. Some continuous Monte Carlo methods for the Dirichlet problem. *The Annals of Mathematical Statistics* (1956), 569–589.
Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin transformations for simulations on infinite domains. *ACM Trans. Graph.* 40, 4 (2021), 97–1.
Bernt Øksendal. 2003. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications*. Springer, 38–50.
Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph.* 34, 4 (2015), 135–1.
Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2023. *Physically Based Rendering: From Theory to Implementation* (4th ed.). MIT Press, Cambridge, MA, USA. https://pbrt.org/ Fourth edition — full text available at https://pbr-book.org/4ed.
Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 51–62.
Jean-François Remacle and Christophe Geuzaine. 2022. Gmsh's approach to robust mesh generation of surfaces with irregular parametrizations. In *Mesh Generation and Adaptation: Cutting-Edge Techniques*. Springer, 95–112.
Rohan Sawhney. 2021. *FCPW: Fastest Closest Points in the West*.
Rohan Sawhney. 2024. *Monte Carlo Geometry Processing: A Grid-Free Approach to Solving Partial Differential Equations on Volumetric Domains*. Ph. D. Dissertation. Carnegie Mellon University, Pittsburgh, PA, USA. https://www.cs.cmu.edu/afs/cs.cmu.edu/Web/Posters/CSThesis-RohanSawhney24.pdf Ph.D. thesis; defended May 1, 2024.
Rohan Sawhney and Keenan Crane. 2020. Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains. *ACM Transactions on Graphics* 39, 4 (2020).
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. *arXiv preprint arXiv:2302.11815* (2023).
Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2025. State of the Art in Grid-Free Monte Carlo Methods for Partial Differential Equations. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses (SIGGRAPH Courses '25)*. Association for Computing Machinery, New York, NY, USA, Article 3, 8 pages. doi:10.1145/3721241.3734001
Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-free Monte Carlo for PDEs with spatially varying coefficients. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17.
Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics (Tog)* 34, 4 (2015), 1–13.
Silvia Sellán, Herng Yi Cheng, Yuming Ma, Mitchell Dembowski, and Alec Jacobson. 2019. Solid geometry processing on deconstructed domains. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 564–579.
Amit Singer. 2006. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis* 21, 1 (2006), 128–134.
Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2022. Surface-Only Dynamic Deformables using a Boundary Element Method. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 75–86.
Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A practical walk-on-boundary method for boundary value problems. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–16.
Xianfang Sun, Paul L Rosin, Ralph R Martin, and Frank C Langbein. 2008. Random walks for feature-preserving mesh denoising. *Computer Aided Geometric Design* 25, 7 (2008), 437–456.
Andrea Toselli and Olof Widlund. 2004. *Domain decomposition methods-algorithms and theory*. Vol. 34. Springer Science & Business Media.
Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. 2007. Discrete Laplace operators: no free lunch. In *Symposium on Geometry processing*, Vol. 33. Aire-la-Ville, Switzerland, 37.
Z. Yu, R. Sawhney, B. Miller, L. Wu, and S. Zhao. 2025. Robust Derivative Estimation with Walk on Stars. *ACM Trans. Graph.* 44, 6 (2025), 253:1–253:16.
Shuang Zhao, Miloš Hašan, Ravi Ramamoorthi, and Kavita Bala. 2013. Modular flux transfer: efficient rendering of high-resolution volumes with repeated structures. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.
Zihong Zhou, Eugene d'Eon, Rohan Sawhney, and Wojciech Jarosz. 2025. Harmonic Caching for Walk on Spheres. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 44, 6 (Dec. 2025). doi:10.1145/3763322

A Unbiased Estimation of the Poisson Kernel with WoS/WoSt

This section justifies the statement made in Section 4.2 that the terminal Dirichlet hit point produced by WoS/WoSt is distributed according to the same weights that appear in the boundary integral equation (5). In other words, WoS/WoSt enables to estimate the Poisson kernel $P^\Omega(x,z)$ in an unbiased manner.

As a reminder, we consider the mixed Dirichlet-Neumann problem (1) with $f = 0$ and $h = 0$, so that the BIE simplifies to

$$u(x) = \int_{\partial\Omega_D} P^\Omega(x,z)\, g(z)\, dz, \quad x \in \Omega. \tag{13}$$

Step 1: The Poisson kernel defines a probability measure on $\partial\Omega_D$. Let $x \in \Omega$. The Poisson kernel is nonnegative, and choosing $g \equiv 1$ in (13) yields

$$\int_{\partial\Omega_D} P^\Omega(x,z)\, dz = 1.$$

Hence the measure

$$\mu_x(dz) := P^\Omega(x,z)\, dz \tag{14}$$

is a probability measure on $\partial\Omega_D$. With this notation, the solution operator becomes an expectation:

$$u(x) = \int_{\partial\Omega_D} g(z)\, \mu_x(dz) = \mathbb{E}_{Z\sim\mu_x}[g(Z)]\,. \tag{15}$$

Thus the BIE specifies a unique set of *weights* μ_x on the Dirichlet boundary.

Step 2: The recursive BIE yields a one-step unbiasedness property. Let u be the harmonic function solving (1). For any point $y \in \Omega$ and any local domain $A \subseteq \Omega$ such that $y \in A$, applying (2) with $C = A$ gives

$$u(y) = \int_{\partial A} P^A(y,z)\, u(z)\, dz. \tag{16}$$

Now define a one-step sampling rule: given the current point y, choose a domain $A(y)$ containing y, and sample a boundary point $Y^+ \in \partial A(y)$ from the density

$$z \mapsto P^{A(y)}(y,z) \quad \text{on } \partial A(y).$$

For WoS, $A(y) = B(y,R)$ and this density is uniform on $\partial B(y,R)$; for WoSt, $A(y) = \mathrm{St}(y,R)$ and the density is the solid-angle measure seen from y. Then (16) is exactly the statement that this one-step rule is unbiased:

$$\mathbb{E}\big[u(Y^+) \mid Y = y\big] = u(y). \tag{17}$$

Step 3: Iterating the one-step rule produces a martingale. We now formalize the walk. Fix $x_0 \in \Omega$ and set $X_0 := x_0$. Given X_k, choose a local domain $A_k := A(X_k) \subseteq \Omega$ with $X_k \in A_k$, and sample

$$X_{k+1} \sim P^{A_k}(X_k, \cdot) \ \text{ on } \ \partial A_k. \tag{18}$$

Let $\mathcal{F}_k := \sigma(X_0, \dots, X_k)$ be the natural filtration. By (17), the process $M_k := u(X_k)$ satisfies

$$\mathbb{E}[M_{k+1} \mid \mathcal{F}_k] = M_k, \tag{19}$$

so $\{u(X_k)\}_{k\geq 0}$ is a martingale.

Step 4: Stopping at the Dirichlet boundary yields the global weights. Let T be the first index such that $X_T \in \partial\Omega_D$ (or $X_T \in \partial\Omega_D^\varepsilon$ for an ε-stopping rule), also known as exit time. Since $u = g$ on $\partial\Omega_D$, we have $u(X_T) = g(X_T)$. Under standard conditions ensuring the validity of optional stopping (e.g., bounded u and integrability of T, see additional discussion below), the martingale property (19) implies [Levin and Peres 2017]

$$u(x_0) = \mathbb{E}[u(X_T)] = \mathbb{E}[g(X_T)]\,. \tag{20}$$

At the same time, the BIE for Ω (15) yields

$$u(x_0) = \int_{\partial\Omega_D} g(z)\, \mu_{x_0}(dz). \tag{21}$$

Comparing (20) and (21), we conclude that the exit distribution ν_{x_0} of X_T satisfies the identity

$$\mathbb{E}_{Z\sim\nu_{x_0}}[g(Z)] = \mathbb{E}_{Z\sim\mu_{x_0}}[g(Z)] \quad \text{for all bounded measurable } g.$$

In other words, the random exit point produced by the walk reproduces *exactly the same weighted combination of boundary values* as the BIE. This justifies that WoS/WoSt trajectories directly "sample" the Poisson kernel, and thus allow to construct its weights by binning the exit points of trajectories.

Theorem A.1 (WoS/WoSt trajectories sample the Poisson kernel). *Let $x_0 \in \Omega$ and let $\mu_{x_0}(dz) = P^\Omega(x_0,z)\,dz$ be defined by* (14)*. Consider the WoS/WoSt walk $\{X_k\}$ defined by* (18) *and let T be its first hitting time at the boundary $\partial\Omega_D$ (or, in practice, the ε-shell $\partial\Omega_D^\varepsilon$). Assume optional stopping applies to the martingale $u(X_k)$. Then for any bounded measurable g on $\partial\Omega_D$,*

$$\mathbb{E}[g(X_T)] = \int_{\partial\Omega_D} g(z)\, P^\Omega(x_0,z)\, dz = \int_{\partial\Omega_D} g(z)\, \mu_{x_0}(dz). \tag{22}$$

Equivalently, the exit distribution ν_{x_0} of X_T coincides with μ_{x_0} in the weak sense (of matching expectations against all test functions g).

Proof Outline. By the local BIE (16) and the sampling rule (18), $\{u(X_k)\}$ is a martingale as in (19). By optional stopping at T, $u(x_0) = \mathbb{E}[u(X_T)]$. Since $u = g$ on $\partial\Omega_D$, we have $u(X_T) = g(X_T)$, hence $u(x_0) = \mathbb{E}[g(X_T)]$. Finally, the BIE for Ω (i.e., Equation (13)) implies $u(x_0) = \int_{\partial\Omega_D} g(z)\, P^\Omega(x_0,z)\, dz$. Combining the two identities yields (22). □

Corollary: estimating panel probability masses by counting hits. Let $\partial\Omega_D \approx \bigsqcup_{j=1}^m \Gamma_j$ be a panelization with collocation points z_j. Applying Theorem A.1 to $g = \mathbf{1}_{\Gamma_j}$ gives

$$\mathbb{P}[X_T \in \Gamma_j] = \int_{\Gamma_j} P^\Omega(x_0,z)\, dz,$$

so the empirical frequency of hits in Γ_j over many trajectories is an unbiased estimator of the panel mass of the Poisson kernel.

Additional discussion on the assumptions for the exit time and its distribution. The derivation above assumes finite exit times and exact boundary evaluations. In practice, WoS and WoSt require an ε-shell $\partial\Omega_D^\varepsilon$ to almost-surely terminate within a finite number of steps (see [Binder and Braverman 2012] for WoS). This introduces a bias since $u(X_T)$ only approximates $g(X_T)$ (typically via projection) rather than being equal. Nevertheless, consistency results for WoS [Binder and Braverman 2012] establish that the estimator converges as $\varepsilon \to 0$. Since the solution is an integral against the Poisson kernel, this consistency suggests that the exit distribution also converges to the correct Poisson kernel weights in the limit, justifying the interpretation of WoS/WoSt as sampling the Poisson kernel.

B Proof of The Markov Averaging Property

Let's fix a transient state $i \in \mathcal{T}$ and let $T := \inf\{t \geq 0 : X_t \in \mathcal{A}\}$ be the corresponding absorption time. Define $u(i) := \mathbb{E}[g(X_T) \mid X_0 = i]$, where g is prescribed on $\mathcal{A}$. Extend u to $\mathcal{A}$ by $u(a) = g(a)$. Conditioning on the first step and using the law of total expectation gives

$$\begin{aligned} u(i) &= \mathbb{E}[g(X_T) \mid X_0 = i] \\ &= \sum_{j\in\mathcal{S}} \mathbb{E}\big[g(X_T)\, \mathbf{1}_{\{X_1=j\}} \mid X_0 = i\big] \\ &= \sum_{j\in\mathcal{S}} \mathbb{E}[g(X_T) \mid X_0 = i,\, X_1 = j]\ \mathbb{P}(X_1 = j \mid X_0 = i) \\ &= \sum_{j\in\mathcal{S}} P_{ij}\, \mathbb{E}[g(X_T) \mid X_0 = i,\, X_1 = j] \end{aligned}$$

By the strong Markov property, the conditional law of the future after time 1 depends only on X_1, which gives

$$\mathbb{E}[g(X_T) \mid X_0 = i,\, X_1 = j] = \mathbb{E}[g(X_T) \mid X_0 = j] = u(j)$$

Substituting yields

$$u(i) = \sum_{j\in\mathcal{S}} P_{ij}\, u(j),$$

which is (9).

C Complexity and Memory Requirements

As a reminder, for a given discretization choice $N = T \times B$, local interior (resp. co-edge) solution operators are discretized as $\widehat{\mathcal{H}}_{\text{inside}} : \mathbb{R}^{4B} \to \mathbb{R}^{B\times B}$ (resp. $\widehat{\mathcal{H}}_{\text{co-edge}} : \mathbb{R}^{6B} \to \mathbb{R}^{B}$), and are stored respectively as $B^2 \times 4B$ and $B \times 6B$ matrices.

Storage. Each tile in $\mathcal{D}$ (resp. each co-edge subdomain in $\mathcal{C}$) therefore requires $4B^3$ (resp. $6B^2$) scalar values. In the *worst case* (i.e., in the absence of interior Neumann boundaries), $\mathcal{D}$ contains T^2 subdomains, while $\mathcal{C}$ contains $2\,T(T-1)$ subdomains. The total storage cost is thus dominated by interior solution operators and can be written as

$$4\,B^3\,T^2 + 12\,B^2\,T\,(T-1).$$

In practice, however, we only need to store subdomains intersecting Neumann boundaries, whose number grows with geometric complexity and scales as $O(T)$ (Section 7.3.3). This yields a substantially more favorable storage cost of

$$O\big(4\,B^3\,T + 2\,B^2\,T\big)\,.$$

Number of states. Our discrete Markov chain contains B states per interface, for a total of $2\,T\,(T+1)\,B$ states, or asymptotically $O(T^2B)$. Since $T = N/B$, this corresponds to $O(N^2/B)$ states, showing that increasing B effectively amortizes the size of the global system from $O(N^2)$ to $O(N^2/B)$, at the cost of increased Monte Carlo effort.

D Periodic Homogenization

For spatially varying diffusion, the boundary value problem (1) can be written as

$$\nabla \cdot \big(\mathbf{K}(x)\nabla u(x)\big) = f(x), \tag{23}$$

where $\mathbf{K} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a symmetric, positive-definite, spatially-varying diffusion matrix. When $\mathbf{K} \equiv \mathbf{I}$, Equation (23) reduces to (1), i.e., $\Delta u = f$.

Homogenization seeks an *effective* (constant) diffusion matrix $\mathbf{K}^*$ that describes the macroscopic behavior of a domain obtained by periodically tiling a representative unit cell Y:

$$\nabla \cdot \big(\mathbf{K}^*\nabla u_0(x)\big) = f(x),$$

where u_0 denotes the homogenized (macroscopic) solution. This formulation relies on three classical assumptions: (i) scale separation between the microstructure and the domain, (ii) Y-periodicity of the coefficients, and (iii) an infinite periodic tiling. Importantly, the effective diffusion matrix $\mathbf{K}^*$ is *not* given by the simple spatial average of $\mathbf{K}$ over Y.

A standard derivation of $\mathbf{K}^*$ is based on the *two-scale asymptotic expansion* [Allaire 1992]. Introducing a small parameter $\varepsilon > 0$ that characterizes the size of the microstructure, we consider

$$\nabla \cdot \left(\mathbf{K}\left(\frac{x}{\varepsilon}\right)\nabla u_\varepsilon(x)\right) = f(x), \tag{24}$$

and define the "fast" variable $y = x/\varepsilon \in Y$. The solution is assumed to admit an expansion of the form

$$u_\varepsilon(x) \approx u_0(x) + \varepsilon\, w(y), \quad \text{with} \quad \int_Y w(y)\,dy = 0, \tag{25}$$

where u_0 is the macroscopic component and w captures periodic microscopic fluctuations with zero mean.

Substituting (25) into Equation (24) gives, for each coordinate direction $\mathbf{e}_i \in \mathbb{R}^d$:

$$\nabla_y \cdot \big(\mathbf{K}(y)\big(\mathbf{e}_i + \nabla_y w_i(y)\big)\big) = 0 \quad \text{in } Y,$$

subject to Y-periodic boundary conditions and $\int_Y w_i(y)\,dy = 0$. In two dimensions, this requires solving two independent boundary value problems, one for each coordinate direction $\mathbf{e}_i$.

The effective diffusion tensor is then given by

$$K^*_{ij} = \int_Y \mathbf{K}(y)\big(\mathbf{e}_i + \nabla_y w_i(y)\big) \cdot \mathbf{e}_j\,dy.$$

We follow standard practice and symmetrize $\mathbf{K}^*$ before using it in the coarsened simulation, where each tile is represented by a *single* quad element with diffusion tensor $\mathbf{K}^*$.

E Masking Strategy for Complex Geometries and Fine Tilings

For very complex geometries, the panel masking strategy for interfaces introduced in Section 6.2 can become overly conservative as the tiling becomes finer. Collocation panels that capture fine geometric details may be discarded, introducing more bias in the final global solution.

In cases with very complex geometry and fine tilings, we propose an alternative strategy that retains all collocation panels whose segment lies *partially* inside the domain. Concretely, we test whether either endpoint of each panel lies inside the domain and cast a ray from one endpoint to the other. An intersection of that ray with the Neumann boundary indicates that part of the segment lies within the domain. During Monte Carlo estimation, we, once again, use rejection sampling to discard walks that fall outside the domain. This strategy introduces only a modest overhead since only a few additional collocation points are considered, while better preserving convergence for large tile counts in complex geometries.