

Conformal First Passage for Epsilon-free Walk-on-Spheres

PAUL HIMMLER and TOBIAS GÜNTHER, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

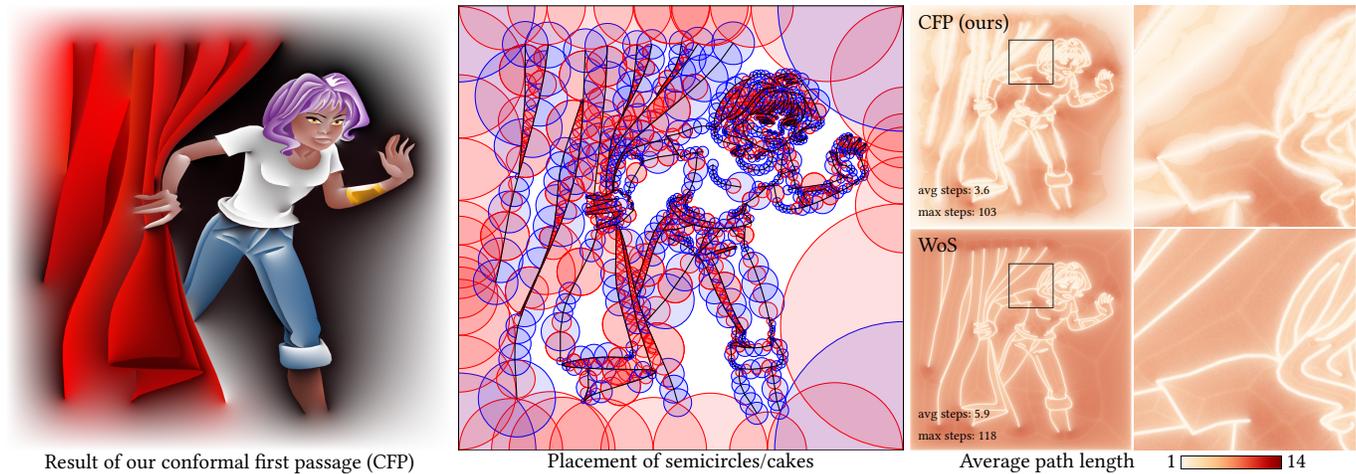


Fig. 1. We propose a recursive first passage algorithm for solving the boundary integral equation (BIE) of the Laplace equations, here applied to diffusion curves. Instead of performing a biased Walk-on-Spheres (WoS) into epsilon shells near boundaries, our method instead solves the PDE on semicircles and circle sectors (cakes) by using conformal mappings to the unit circle. From left to right, we show the resulting image, the precomputed arrangement of semicircles and cakes, and the average path length of our method compared to Walk-on-Spheres. Our method is faster and more accurate than WoS.

In recent years, grid-free Monte Carlo methods have gained increasing popularity for solving fundamental partial differential equations. For a given point in the domain, the *Walk-on-Spheres* method solves a boundary integral equation by integrating recursively over the largest possible sphere. When the walks approach boundaries with Dirichlet conditions, the number of path vertices increases considerably, since the step size becomes smaller with decreasing distance to the boundary. In practice, the walks are terminated once they reach an epsilon-shell around the boundary. This, however, introduces bias, leading to a trade-off between accuracy and performance. Instead of using spheres, we propose to utilize geometric primitives that share more than one point with the boundary to increase the likelihood of immediately terminating. Along the boundary of those new geometric primitives a sampling probability is needed, which corresponds to the exit probability of a Brownian motion. This is known as a first passage problem. Utilizing that Laplace equations are invariant under conformal maps, we transform exit points from unit circles to the exit points of our geometric primitives, for which we describe a suitable placement strategy. With this, we obtain a novel approach to solve the Laplace equation in two dimensions, which does not require an epsilon-shell, significantly reduces the number of path vertices, and reduces inaccuracies near Dirichlet boundaries.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: Monte Carlo, Partial Differential Equations, Walk-on-Spheres, First Passage, Conformal Maps

Authors' Contact Information: Paul Himmler, paul.himmler@fau.de; Tobias Günther, tobias.guenther@fau.de, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Bavaria, Germany.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2025 Copyright held by the owner/author(s).
ACM 1557-7368/2025/8-ART
<https://doi.org/10.1145/3730942>

ACM Reference Format:

Paul Himmler and Tobias Günther. 2025. Conformal First Passage for Epsilon-free Walk-on-Spheres. *ACM Trans. Graph.* 44, 4 (August 2025), 11 pages. <https://doi.org/10.1145/3730942>

1 Introduction

In the world around us, many phenomena can be modeled through linear elliptic partial differential equations. Examples are heat transfer in thermodynamics [Bergheau and Fortunier 2013], Newtonian gravity in classical mechanics [Poisson and Will 2014], electric charge densities in electromagnetism [Jones 1964], or pressure corrections in fluid simulation [Bridson 2015]. Further, linear elliptic PDEs are encountered in many branches of computer graphics, such as geometric modeling [De Lambilly et al. 2023], rendering [Ketunen et al. 2015], fluid simulation [Wang et al. 2024], vector graphics [Hou et al. 2018], and visualization [Esturo et al. 2013]. Recently, grid-free Monte Carlo methods entered the stage in graphics [Sawhney and Crane 2020; Sugimoto et al. 2023], allowing to solve PDEs on complicated domains without the need for grid discretizations. In this class of methods, the Walk-on-Spheres algorithm [Muller 1956] solves boundary integral equations stochastically on largest possible spheres. Its recursive formulation results in random walks that terminate once a boundary with Dirichlet condition is reached. The step size of the Walk-on-Spheres is determined by the distance to the nearest Dirichlet boundary. Thus, the closer the walk gets to the boundary, the smaller the steps become. In theory, an infinite number of steps would be needed to reach the boundary, although in practice the walks are terminated when they enter an epsilon shell around the boundary. The subsequent snapping to the closest boundary introduces some bias [Binder and Braverman 2012], but

more importantly the large number of walking steps induces high computational cost. The cause of this problem is that the sphere to sample from typically shares only 1 point with the boundary, making it highly unlikely that a random sample is placed close to this point. Instead of placing the largest possible sphere around the current sample point, we propose to place largest-possible semicircles along piecewise linear boundaries, and largest-possible circle sectors at convex and concave corners. For simplicity, we refer to circle sectors as *cakes* throughout the remainder of this paper. Further, for regions away from the boundary, we place empty circles in order to cover all pixels of the image. This placement of primitives can be done once in a preprocess. During the Monte Carlo random walks, we look up the next primitive to use and sample the exit point on its boundary by means of a conformal map to the unit circle, for which the Green's function and the Poisson kernel are known. We show that our approach significantly lowers the number of steps in the random walk. Fig. 1 gives an example. In this paper, we concentrate on Laplace equations in two dimensions, with piecewise-linear boundary geometry, and with Dirichlet boundary conditions, which can be found for example in diffusion curves [Orzan et al. 2008]. We later discuss the generalization to 2D Poisson equations, as well as Neumann and Robin boundary conditions [Miller et al. 2024b; Sawhney et al. 2023]. In summary, we contribute:

- the derivation of exit probabilities using conformal maps for 2D primitives that are nestled against boundary edges,
- a placement strategy for semicircles, circle sectors, and full circles that cover the domain,
- a recursive walk that has a higher chance of jumping onto boundaries without requiring an epsilon shell.

2 Related Work

2.1 Partial Differential Equations

In the following section, we recap the foundations of diffusion curves and discuss Monte Carlo methods related to our approach.

Diffusion Curves. Diffusion curves have been introduced by Orzan et al. [2008] as vector graphics primitives for modeling smoothly shaded images. Formally, these curves are cubic Bézier splines that diffuse colors from their left and right hand side into the domain, which can be modeled as boundary value problem [Jeschke et al. 2009]. For boundary value problems, the Laplace equation describes the smoothest unknown field $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^n$ residing in the domain Ω , here denoted for an n -variate function $\mathbf{u}(\mathbf{x})$, that fulfills:

$$\Delta \mathbf{u}(\mathbf{x}) = \mathbf{0} \quad \text{if } \mathbf{x} \in \Omega \quad (1)$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{if } \mathbf{x} \in \partial\Omega \quad (2)$$

In the interior of the domain Ω , the Laplacian $\Delta \mathbf{u}(\mathbf{x})$ is requested to vanish, leading to a smooth solution. Along the domain boundary, Dirichlet boundary conditions define a desired value $\mathbf{g}(\mathbf{x}) : \partial\Omega \rightarrow \mathbb{R}^n$ that is given by the colors on the left and right side of the diffusion curve, respectively. More generally, Neumann and Robin boundary conditions could be modeled similarly [Miller et al. 2024b]. For the remainder of the paper, we concentrate on pure Dirichlet boundary conditions, since we mainly focus on improving the behavior of the Walk-on-Spheres near Dirichlet boundaries.

Boundary Integral Equation (BIE). As discussed in [Sawhney et al. 2023], a *general* solution to the Laplace equation in Eqs. (1)–(2) is given by the direct boundary integral equation (BIE) [Costabel 1987] for any two sets $\mathcal{A}, C \subset \Omega$:

$$a(\mathbf{x}) \mathbf{u}(\mathbf{x}) = \int_{\partial\mathcal{A}} \mathcal{P}^C(\mathbf{x}, \mathbf{z}) \mathbf{u}(\mathbf{z}) - \mathcal{G}^C(\mathbf{x}, \mathbf{z}) \frac{\partial \mathbf{u}(\mathbf{z})}{\partial \mathbf{n}_z} dz \quad (3)$$

for the inward pointing normal \mathbf{n}_z . Here, \mathcal{G} is the *Green's function* and $\mathcal{P} = \partial \mathcal{G} / \partial \mathbf{n}_z$ is the *Poisson kernel*, which are unfortunately only known for simple domains. For smooth boundary curves, $a(\mathbf{x})$ is:

$$a(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega \\ 1/2 & \mathbf{x} \in \partial\Omega \\ 0 & \mathbf{x} \notin \Omega \end{cases} \quad (4)$$

For the treatment of non-smooth boundary curves, we refer to Sawhney et al. [2023]. Solvers of the BIE differ in their choices of \mathcal{A} and C . A variety of boundary methods exist to solve the BIE [Bang et al. 2023; Chen et al. 2024; Van de Gronde 2011]. For a more comprehensive introduction, we refer to Sabelfeld and Simonov [1994], Sawhney et al. [2023] and Sugimoto et al. [2023].

2.2 Monte Carlo Methods for PDEs

Walk-on-Spheres. The WoS [Muller 1956] solves Eq. (3) for Dirichlet boundary conditions by placing at \mathbf{x} the largest possible ball $\mathcal{A} = C = \mathcal{B}(\mathbf{x})$ centered at \mathbf{x} :

$$\mathbf{u}(\mathbf{x}) = \int_{\partial\mathcal{B}(\mathbf{x})} \mathcal{P}^{\mathcal{B}}(\mathbf{x}, \mathbf{z}) \mathbf{u}(\mathbf{z}) dz \quad (5)$$

For a ball in 2D, the Green's function and Poisson kernel are:

$$\mathcal{G}^{\mathcal{B}}(\mathbf{c}, \mathbf{x}) = \frac{1}{2\pi} \log\left(\frac{R}{r}\right), \quad \mathcal{P}^{\mathcal{B}}(\mathbf{c}, \mathbf{x}) = \frac{\partial \mathcal{G}^{\mathcal{B}}(\mathbf{c}, \mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = \frac{1}{2\pi r} \quad (6)$$

for $r = \|\mathbf{c} - \mathbf{x}\|$, and a ball with radius R centered at \mathbf{c} . In Eq. (5), the second term of Eq. (3) vanished, since the Green's function of a ball is zero on the boundary, i.e., when $R = r$. Monte Carlo integration of Eq. (5) gives:

$$\mathbf{u}(\mathbf{x}_k) \approx \frac{1}{N} \sum_{n=1}^N \frac{\mathcal{P}^{\mathcal{B}}(\mathbf{x}_k, \mathbf{x}_{k+1}) \mathbf{u}(\mathbf{x}_{k+1})}{p(\mathbf{x}_k, \mathbf{x}_{k+1})} = \frac{1}{N} \sum_{n=1}^N \mathbf{u}(\mathbf{x}_{k+1}) \quad (7)$$

when the Poisson kernel is used as probability distribution for placing the next sample at \mathbf{x}_{k+1} , which corresponds to a uniform sampling of the boundary of the ball. This recursive walk terminates once \mathbf{x}_{k+1} is within the epsilon shell of a boundary with Dirichlet condition, where the value $\mathbf{u}(\mathbf{x}_{k+1}) \approx \mathbf{g}(\mathbf{x}'_{k+1})$ is chosen and where $\mathbf{x}'_{k+1} \in \partial\mathcal{B}$ is the closest point to \mathbf{x}_{k+1} on the boundary $\partial\mathcal{B}$. This introduces bias that reduces at the rate $O(1/\log \epsilon)$ [Binder and Braverman 2012] with ϵ being the thickness of the epsilon shell.

Recent Extensions and Applications. The Walk-on-Spheres (WoS) [Muller 1956] was popularized in graphics by Sawhney and Crane [2020]. Analogous to bi-directional path tracing, Qi et al. [2022] proposed constructing random walks in forward and reverse directions. Hwang et al. [2015] introduced off-centered walks to increase the exit probability within the epsilon-shell. Instead, we utilize other primitives, increase the exit probability in the whole domain, and avoid epsilon shells entirely. Based on indirect BIEs, which utilize single-layer and double-layer potentials, Sabelfeld [1982]

introduced the Walk-on-Boundaries. The method was generalized for Poisson equations with Dirichlet, Neumann, or Robin boundary conditions [Sabelfeld and Simonov 1994]. Sugimoto et al. [2023] generalized it to direct BIEs and mixed boundary problems, and studied combinations to develop a practical solver. Within WoS, reflections have been used at Neumann boundaries [Mascagni and Simonov 2004]. Since the radius of the ball becomes very small near a Neumann boundary, the walks may require a large number of steps to terminate. Thus, Sawhney et al. [2023] introduced the Walk-on-Stars, which determines the radius of the sphere for Neumann boundaries by the distance to the closest silhouette point. Near Dirichlet boundaries, however, the step size is still very small. This is the problem we address in this paper. Rather than Monte Carlo integrating the BIE for each pixel independently, Miller et al. [2023] proposed caching partial results on the boundary (for the boundary integral) and in the interior (for the interior integral of the source term) and used those as correlated samples across pixels. Li et al. [2023] used a neural network to represent caches in WoS and sampled those caches after a pre-determined walk depth. Recently, Czekanski et al. [2024] proposed a caching strategy based on the continuity of paths of Brownian motion to reduce variance. The Walk-on-Stars has also been extended to Robin boundary conditions [Miller et al. 2024b]. Further, inverse problems [Miller et al. 2024a; Yilmazer et al. 2024; Yu et al. 2024] have been treated through a differentiable WoS where source terms, boundary conditions, and the boundary positions have been optimized. The WoS was used to derive a vorticity-based Monte Carlo fluid solver [Rioux-Lavoie et al. 2022] and the Walk-on-Boundaries was used to form a velocity-based Monte Carlo fluid solver [Sugimoto et al. 2024a]. Recent work by Sugimoto et al. [2024b] adapted WoS to a volumetric neighborhood, developing a method for surface PDE's with Dirichlet boundaries. To handle exterior problems more effectively, Nabizadeh et al. [2021] proposed an inversion map to transform them into a bounded interior domain. Although they solve a different problem, their mapping in 2D is also conformal and thus related to our approach.

2.3 Brownian Motion

Instead of only using circles as primitives in our walk, we solve 2D BIEs with Monte Carlo sampling on geometric primitives such as semicircles and circle sectors. In order to confirm the correctness of the derived probability distributions, we compare them later with Monte Carlo simulations of Brownian motion, which approach the steady state of the heat equation, and thereby solve Laplace problems. For a coordinate $b(t) : \mathbb{R} \rightarrow \mathbb{R}$, we speak of Brownian motion if $b(t+h) - b(t) \sim \mathcal{N}(0, h)$ holds independently $\forall t \in \mathbb{R}$ where $\mathcal{N}(0, h)$ is a mean-free normal distribution with variance h . The Brownian path unfolds as [Glasserman 2004, p. 81]:

$$b(t_{i+1}) = b(t_i) + \sqrt{t_{i+1} - t_i} \cdot Z_i \quad (8)$$

where Z_i are independent random samples from a uniform normal distribution. For a seed point $\mathbf{x}(0) = \mathbf{c}$ within a certain domain, the first passage problem asks where a particle exits the domain when it performs a Brownian motion. If a particle is seeded at the center \mathbf{c} of a ball \mathcal{B} with radius R , then the first exit probability $p(\mathbf{c}, \mathbf{x})$ at

boundary point $\mathbf{x} \in \partial\mathcal{B}$ is the same for all directions:

$$p(\mathbf{c}, \mathbf{x}) = \frac{1}{2\pi r}, \quad r = \|\mathbf{c} - \mathbf{x}\| \quad (9)$$

Since the exit position $\mathbf{x} \in \partial\mathcal{B}$ is on the boundary of the ball, we have $r = R$. Note that the exit probability of the Brownian motion on a ball in Eq. (9) is identical to the Poisson kernel $\mathcal{P}^B(\mathbf{c}, \mathbf{x})$ of a ball in Eq. (6). This equivalence holds similarly for other geometric primitives [Given et al. 1997], for which we later compare the derived exit probabilities with first passage walks using Eq. (8).

2.4 Method of Images

In electrostatics, the method of images is a well-known approach to derive Green's functions by introducing exterior artificial point charges [Jackson 1998]. Unfortunately, the method of images is applicable to a limited set of geometric configurations only. Motivated by its application to molecular dynamics, Given et al. [1997] used the method of images to derive Green's functions for spheres that contain or intersect one boundary sphere, or one boundary edge passes through it, which is referred to as a *first passage sphere*. They proposed a greedy algorithm, which chooses the largest possible first passage sphere, for which the Green's function is known. Conceptually, there are similarities to our approach. Likewise, we generate first passage circles that intersect with the boundary geometry. In order to support convex and concave corners, we compute Green's functions for circle segments, as well, which share two edges with the boundary geometry. Further, instead of generating the circles on-the-fly during the walk around the current sample point, we precompute all circles and let the random walk select the next primitive from the precomputed set. This means that our sample point is not at the center of the first passage circles. When selecting the next primitive, the primitive boundary that is not shared with the domain should be far away from the start point of the walk, since the likelihood of first passage decreases with distance, resulting in the desired higher probability of exiting on the domain boundary. Johnston et al. [2005] derived Green's functions for equilateral triangles. Placing equilateral triangles that share edges with the domain, while the remaining non-shared edges are far away from the start of the walk, is an optimization problem in itself. The placement of our primitives is simpler since the radius is found with closest-point queries. Sawhney and Crane [2020] already mentioned in their seminal work that first passage approaches are an option to avoid the epsilon shell of the Walk-on-Spheres. As far as we know, this has not been attempted in the graphics community.

2.5 Conformal Maps

Conformal maps $F(x) : \mathcal{X} \rightarrow \mathcal{Y}$ between the spaces \mathcal{X} and \mathcal{Y} are well-known in geometry processing for their property of being angle-preserving [Choi et al. 2020; Sawhney and Crane 2017; Weber and Gotsman 2010]. By representation in the complex number plane $\mathcal{X} \subset \mathbb{C}, \mathcal{Y} \subset \mathbb{C}$, such maps are biholomorphic, i.e., they are bijective and complex differentiable [Kythe 2019]. By the Riemann mapping theorem a conformal map exists between non-empty simply connected open subsets such as circles, semicircles, or circle sectors. An important observation is that Laplace equations are invariant under

conformal maps [Kythe 2019, Theorem 17.6]. Thus, we later use conformal maps to solve Laplace equations on different primitives.

3 Conformal First Passage

3.1 Overview

In the Walk-on-Spheres [Sawhney and Crane 2020], the number of steps in the random walk depends on the thickness of the epsilon shell. The smaller the epsilon shell, the less bias is introduced but the longer the walks become. Fig. 2a illustrates random walks near straight edges and at corners. The epsilon is necessary since circles touch the boundary only in one point, making it highly unlikely to sample. Instead of using largest-possible empty circles, we take inspiration from the first passage algorithm of Given et al. [1997], and use circles that intersect with Dirichlet boundary geometry. For the intersection with one edge, we use semicircles, and to support corners, we use circle segments (cakes), which are shown in Fig. 2b. While our semicircle primitive is conceptually similar to one of the Green's functions proposed by Given et al. [1997], the formulation and derivation differ significantly, since we use conformal maps instead of the method of images, cf. Section 2.4. To the best of our knowledge, circle segments have not been used before. We calculate exit probabilities for walks that start not at the center of circles, but that start at arbitrary positions. We derive these probabilities from conformal maps in the following sections for semicircles and cakes, which is based on Green's functions. Fig. 3 visualizes Green's functions on the three primitives for a given start point using isocontours. Along the boundary of the primitives, the normal derivative of the Green's function, corresponding to the density of the isocontours, is equal to the exit probability according to which we want to generate sample points efficiently. We achieve this by generating sample points on the circle, and conformally mapping them to the semicircle or cake, respectively. After introducing the conformal maps, for which an overview is given in Fig. 4, we elaborate on the placement algorithm that fills the scene with overlapping primitives. Lastly, we describe our recursive conformal first passage algorithm.

3.2 Semicircle Primitive

Our semicircle has a certain radius, position, and orientation such that its straight edge aligns with a part of the domain boundary $\partial\Omega$. We translate, rotate, and scale the semicircle to become a *unit* semicircle, namely the right half of a unit circle. When expressing the domain of the unit semicircle $\mathcal{Z} \subset \mathbb{C}$ in the complex domain, where the real part is the x coordinate and the imaginary part is the y coordinate, then all points inside the semicircle have positive real parts. That is, $\forall z \in \mathcal{Z}$ it holds that $|z| \leq 1$ and $-\frac{\pi}{2} < \arg(z) < \frac{\pi}{2}$. As intermediate domains we also need the upper half plane $\mathcal{Y} \subset \mathbb{C}$ and the unit circle $\mathcal{X} \subset \mathbb{C}$. Next, we describe the conformal maps and their inverses, which map between those domains.

Unit Semicircle To Upper Half Plane. The mapping from a unit semicircle to the upper half plane is given by $F_0(z) : \mathcal{Z} \rightarrow \mathcal{Y}$, which is a rotated version of Map 3.40 in Kythe [2019]:

$$F_0(z) = \left(\frac{z-i}{z+i} \right)^2, \quad F_0^{-1}(y) = \frac{2\sqrt{-y} + (1+y)i}{1-y} \quad (10)$$

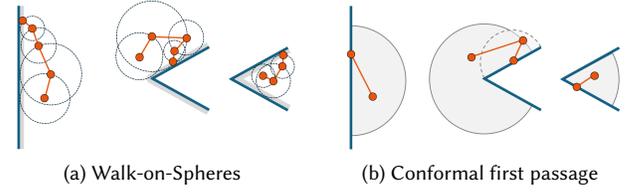


Fig. 2. Side-by-side comparison of Walk-on-Spheres [Sawhney and Crane 2020] and our conformal first passage. Three cases are shown from left to right: a straight boundary curve, a convex corner, and a concave corner.

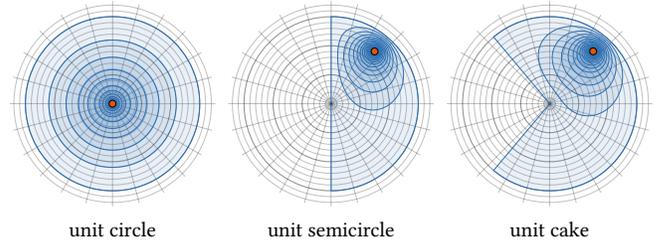


Fig. 3. Visualization of Green's functions using isocontours. The normal derivative of the Green's function (i.e., the density of isocontours) is equal to the exit probability of a random walker starting at the orange dot.

Upper Half Plane To Unit Circle. $F_1(y) : \mathcal{Y} \rightarrow \mathcal{X}$ maps from the upper half plane to the unit circle, c.f. Map 3.23 in Kythe [2019]:

$$F_1(y) = \frac{y-i}{y+i}, \quad F_1^{-1}(x) = \frac{(1+x)i}{1-x} \quad (11)$$

Unit Semicircle To Unit Circle. With this, the mapping from the unit semicircle to the unit circle can be expressed by concatenating the upper two mappings, which we denote by $F_2(z) : \mathcal{Z} \rightarrow \mathcal{X}$:

$$F_2(z) = F_1(F_0(z)), \quad F_2^{-1}(x) = F_0^{-1}(F_1^{-1}(x)) \quad (12)$$

Offset Unit Circle To Unit Circle. The closed-form expression of the Poisson kernel of the circle in Eq. (6) is only valid if the start point of the first passage walk is at the circle center. In practice, our Monte Carlo walks may arrive at any point in a semicircle. Thus, mapping the point to the unit circle usually results in a place c that is not at the circle center. To map point c conformally such that it is at the center of the unit circle, we use $F_3(c, x) : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$, cf. Map 3.33 in Kythe [2019], with \bar{c} being the complex conjugate of c :

$$F_3(c, x) = \frac{x-c}{1-x\bar{c}}, \quad F_3^{-1}(c, x) = \frac{x+c}{1+x\bar{c}} \quad (13)$$

Offset Unit Semicircle To Unit Circle. Combining all together, the conformal map $G(c, z) : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathcal{X}$ takes the unit semicircle to the unit circle and makes sure that point c in the unit semicircle is mapped to the center of the unit circle:

$$G(c, z) = F_3(F_2(c), F_2(z)), \quad G^{-1}(c, x) = F_2^{-1}(F_3^{-1}(F_2(c), x)) \quad (14)$$

The inverse $G^{-1}(c, x) : \mathcal{Z} \times \mathcal{X} \rightarrow \mathcal{Z}$ takes a coordinate x that was sampled on the unit circle into the unit semicircle, whereas c is the previous sample point in the unit semicircle.

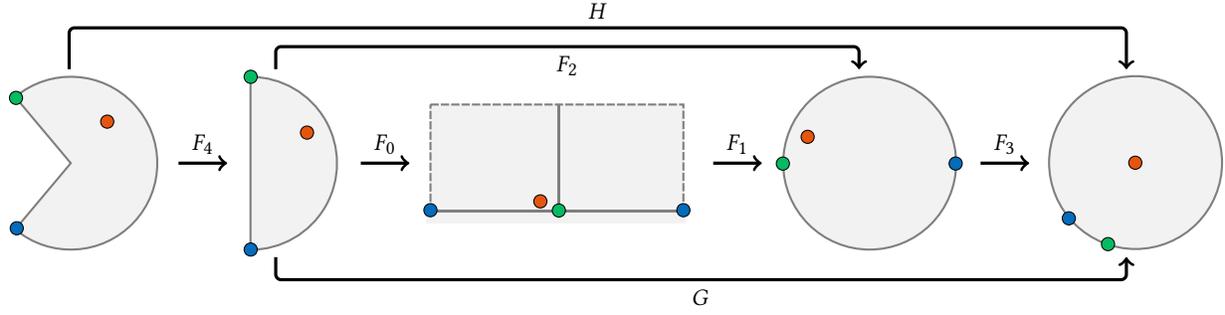


Fig. 4. Overview of the conformal maps used in this paper. From left to right: cakes, semicircles, half planes, circles, and circles for which a given point is mapped to the circle center. The Green's function $\mathcal{G}^{\mathcal{B}}$ is known for the circle. Using the inverse maps G^{-1} and H^{-1} it can be transformed to the semicircle and cake, respectively. In this figure, selected points are colored, giving an impression how they are transformed between the domains.

Sample Generation. To perform the random walk x_0, \dots, x_n in the domain Ω , we need to generate random samples on the semicircle boundaries. This is done by uniformly generating samples on the boundary of the unit circle $\partial\mathcal{X}$ and transforming them with the conformal map $G^{-1}(c, x)$ to the boundary of the unit semicircle $\partial\mathcal{Z}$:

$$z_{k+1} = G^{-1}(z_k, \cos(2\pi\eta) + i \sin(2\pi\eta)) \quad (15)$$

where η is a random number in $[0,1]$. The complex points z_k , which are still in the *unit* semicircle, are rotated, scaled, and translated back into the image domain $x_k \in \Omega$. Since the resulting point distribution follows the Poisson kernel already, it can be readily used for sampling. Next, we derive the Poisson kernel, since it can be used as ground truth probability distribution for our experimental validations [Given et al. 1997].

Poisson Kernel. For a given point c inside the unit semicircle \mathcal{Z} , we compute the Poisson kernel $\mathcal{P}^{\mathcal{Z}}(c, z)$ at a boundary point z by conformally mapping the Green's function $\mathcal{G}^{\mathcal{B}}$ of a ball from the unit circle \mathcal{X} to the unit semicircle \mathcal{Z} and take its normal derivative in the semicircle, which is done via chain rule:

$$\mathcal{P}^{\mathcal{Z}}(c, z) = \frac{d\mathcal{G}^{\mathcal{B}}(0, G(c, z))}{dz} \cdot n_z \quad (16)$$

$$= \frac{\partial \mathcal{G}^{\mathcal{B}}(0, x)}{\partial x} \Big|_{x=G(c, z)} \cdot \frac{\partial G(c, z)}{\partial z} \cdot n_z \quad (17)$$

where $\frac{\partial \mathcal{G}^{\mathcal{B}}(0, x)}{\partial x} = -\frac{x}{2\pi x\bar{x}}$, and $\frac{\partial G(c, z)}{\partial z}$ is the coordinate Jacobian from the unit semicircle \mathcal{Z} to the unit circle \mathcal{X} . Given a path vertex $z_k \in \mathcal{Z}$, the Poisson kernel $\mathcal{P}^{\mathcal{Z}}(z_k, z_{k+1})$ gives us the probability for placing the next point z_{k+1} on the semicircle boundary $\partial\mathcal{Z}$.

Boundary Integral. Phrased on the unit semicircle, the boundary integral equation in Eq. (5) can be split into a part on the straight edge \mathcal{Z}_1 and a part on the circle arc \mathcal{Z}_2 :

$$u(c) = \underbrace{\int_{\partial\mathcal{Z}_1} \mathcal{P}^{\mathcal{Z}}(c, z) g(z) dz}_{\text{direct edge integral}} + \underbrace{\int_{\partial\mathcal{Z}_2} \mathcal{P}^{\mathcal{X}}(c, z) u(z) dz}_{\text{recursive arc integral}}. \quad (18)$$

Similar as in the first passage formulation of Given et al. [1997], the direct part in Eq. (18) is a 1-dimensional integral that could be calculated numerically without the need for a recursion, where $g(z)$

is the Dirichlet boundary condition. The recursive part in Eq. (18) would be computed recursively as usual. For piecewise constant colors on the boundaries, the direct edge integral would simplify to calculating the probability of hitting the edge times the color.

Edge Probability. The probability of directly hitting the straight edge is found by transforming the corners of the edge (i and $-i$) to the unit circle and computing its signed enclosed angle, normalized to a probability by dividing by 2π [Snipes and Ward 2016]:

$$p_{\mathcal{Z}_1}(c) = \frac{\angle(G(c, i), 0, G(c, -i))}{2\pi} \quad (19)$$

3.3 Cake Primitive

To better cover corners, our second primitive is the sector of a unit circle, which we refer to as unit cake for simplicity. The unit cake is likewise defined in the complex plane as $\mathcal{W} \subset \mathcal{C}$ and it is parametrized by a half angle α . A point $w \in \mathcal{W}$ is part of the cake if $|w| \leq 1$ and $-\alpha \leq \arg(w) \leq \alpha$. The conformal map to the unit cake is based on the previously introduced conformal maps to the unit semicircle, cf. Section 3.2.

Unit Cake To Unit Semicircle. The mapping from a unit cake to the right unit semicircle is given by $F_4(\alpha, w) : [0, \pi] \times \mathcal{W} \rightarrow \mathcal{Z}$, which is similar to Map 3.42 in Kythe [2019]:

$$F_4(\alpha, w) = w^{\frac{\pi}{2\alpha}}, \quad F_4^{-1}(\alpha, z) = z^{\frac{2\alpha}{\pi}} \quad (20)$$

Offset Unit Cake To Unit Circle. Combining the maps defined in the previous section with Eq. (20), we obtain a conformal map from the unit cake to the unit circle with the constraint that a given point c in the unit cake is mapped to the center of the unit circle. The map is given by $H(\alpha, c, w) : [0, \pi] \times \mathcal{W} \times \mathcal{W} \rightarrow \mathcal{X}$:

$$H(\alpha, c, w) = G(F_4(\alpha, c), F_4(\alpha, w)), \quad (21)$$

$$H^{-1}(\alpha, c, x) = F_4^{-1}(\alpha, G^{-1}(F_4(\alpha, c), x)) \quad (22)$$

Again, to map a sample point x from the unit circle to the unit cake, the inverse map $H^{-1}(\alpha, c, x) : \mathbb{R} \times \mathcal{W} \times \mathcal{X} \rightarrow \mathcal{W}$ is used.

Sample Generation. To generate random samples on the unit cake that follow the distribution of the Poisson kernel, we again rely on the following sampling strategy: First, we uniformly sample

points on the unit circle \mathcal{X} . Second, we transform the points with the conformal map $H^{-1}(\alpha, c, x)$ to the unit cake \mathcal{W} :

$$w_{k+1} = H^{-1}(\alpha, w_k, \cos(2\pi\eta) + i \sin(2\pi\eta)) \quad (23)$$

where η is a random variable, uniformly distributed in $[0, 1]$. When generating the point w_{k+1} , the previous point w_k is conformally mapped to be at the center of the unit circle, which is the configuration in which the Green function is known, cf. Eq. (6).

Poisson Kernel. To compute the Poisson kernel $\mathcal{P}^{\mathcal{W}}(\alpha, c, w)$ at a boundary point $w \in \partial\mathcal{W}$ for a given seed point $c \in \mathcal{W}$ inside the unit cake, we map the Green's function conformally from the unit circle and take its normal derivative in the unit cake:

$$\mathcal{P}^{\mathcal{W}}(\alpha, c, w) = \frac{d\mathcal{G}^B(0, H(\alpha, c, w))}{d_w} \cdot n_w \quad (24)$$

$$= \frac{\partial\mathcal{G}^B(0, x)}{\partial x} \Big|_{x=H(\alpha, c, w)} \cdot \frac{\partial H(\alpha, c, w)}{\partial w} \cdot n_w \quad (25)$$

with $\frac{\partial H(\alpha, c, w)}{\partial w}$ being the Jacobian from unit cake to unit circle. Analogous to Eq. (18), the boundary integral of the BIE can be split into a part for the two edges $\mathcal{W}_1, \mathcal{W}_2$ and a circular arc \mathcal{W}_3 .

Edge Probability. Similar to the semicircle primitive, the probability of having a first passage through an edge is computed by mapping the end points of the edges from the unit cake \mathcal{W} to the unit circle \mathcal{X} and measuring the signed enclosed angle, divided by the circumference of the unit circle. For edge \mathcal{W}_1 , the points e^{ai} and 0 are transformed, and for edge \mathcal{W}_2 the points 0 and e^{-ai} are transformed to the unit circle, giving the probabilities:

$$p_{\mathcal{W}_1}(c) = \frac{\angle(H(\alpha, c, e^{ai}), 0, H(\alpha, c, 0))}{2\pi} \quad (26)$$

$$p_{\mathcal{W}_2}(c) = \frac{\angle(H(\alpha, c, 0), 0, H(\alpha, c, e^{-ai}))}{2\pi} \quad (27)$$

3.4 Geometry Placement

While the WoS needs distance queries to determine the next largest-possible sphere for a given path vertex, we instead precompute an arrangement of semicircles, cakes, and circles that covers the domain. Our placement of primitives in a domain with piecewise linear boundaries consists of three steps. First, we generate a largest-possible cake (circle sector) for every corner (both convex and concave), as shown in Fig. 5a. The cake may not intersect with geometry other than the two edges belonging to its corner. Second, for all boundary edges we place one or multiple semicircles, following a greedy strategy, see Fig. 5b. Starting from both end points of the edge, a semicircle is tentatively generated with a radius half the edge length. If the tentative semicircle intersects with geometry, then its radius is reduced until it no longer intersects with geometry. If the semicircle happens to contain another primitive entirely, then the other primitive can safely be removed as it is redundant. The next semicircle is centered at the end point of the previous semicircle. In order to further increase coverage, the creation of additional semicircles is enforced at locations where two of the already instantiated semicircles meet. Third, we determine for every pixel the overlapping primitive that has the highest probability to lead from the pixel center onto a straight edge. For this, the probabilities in Eqs. (19),

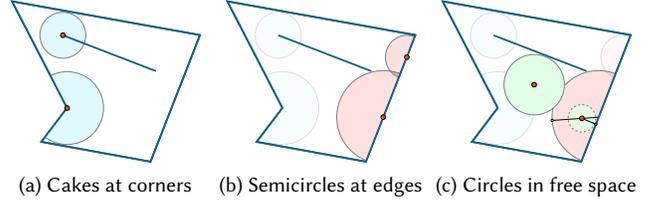


Fig. 5. Exemplary placements of cakes, semicircles, and circles.

(26), and (27) can be used. As simple approximation, we choose the primitive whose radius is farthest away from the pixel center. The ideal primitive is stored for each pixel to enable fast access during the random walk. In case no good primitive was found, we select the largest-possible circle containing the point, see Fig. 5c. Note that we generate primitives only once in a pre-process, while WoS generates a comparable number of circles in every iteration.

3.5 Recursive Algorithm

For each pixel in the image, we first generate a random subsample, which is transformed from image space to the domain of the scene Ω , giving the coordinate x_0 . For the given pixel, we look up the ideal primitive to sample. For this explanation, we assume that a semicircle is chosen. The process is analogous for cakes and circles. Next, the sample point x_0 is transformed from the scene domain Ω into the *unit* semicircle domain, giving $z_0 \in \mathcal{Z}$. By applying Eq. (15), we obtain a sample $z_1 \in \partial\mathcal{Z}$ on the boundary of the semicircle. The coordinate is transformed back into the scene domain, resulting in $x_1 \in \Omega$. If the point x_1 is on the domain boundary $\partial\Omega$, then the Dirichlet boundary condition $g(x_1)$ is evaluated and added to the Monte Carlo sum of the pixel, cf. Eq. (7). Otherwise, the process repeats recursively, generating samples x_2, x_3 , etc. We provide a C++ reference implementation in the supplemental material.

4 Results

In the following sections, we confirm the correctness of our obtained first passage probabilities, we evaluate parameter choices, discuss the scaling behavior of the performance, and we compare the approach with the Walk-on-Spheres regarding path length and runtime. The test scenes were kindly provided by Orzan et al. [2008].

4.1 Brownian Motion Comparison

To confirm the correctness of our approach, we compute first passage samples using Brownian motion, the Walk-on-Spheres, and using our conformal maps, as shown in Fig. 6. For each method, we calculate a histogram of the exit points, normalize the histogram, and compare this with the analytically calculated probability distributions. The first set of experiments in (a)-(c) is conducted on the semicircle. We begin with the Brownian motion, which has the user parameter $h = t_{i+1} - t_i$. We can see that as h approaches 0, the Brownian motion distribution converges to the ground truth. Similarly, the Walk-on-Spheres has a parameter, namely the thickness of the ϵ -shell. The ϵ parameter influences the shape of the distribution when chosen too large. Both Brownian motion and the Walk-on-Spheres

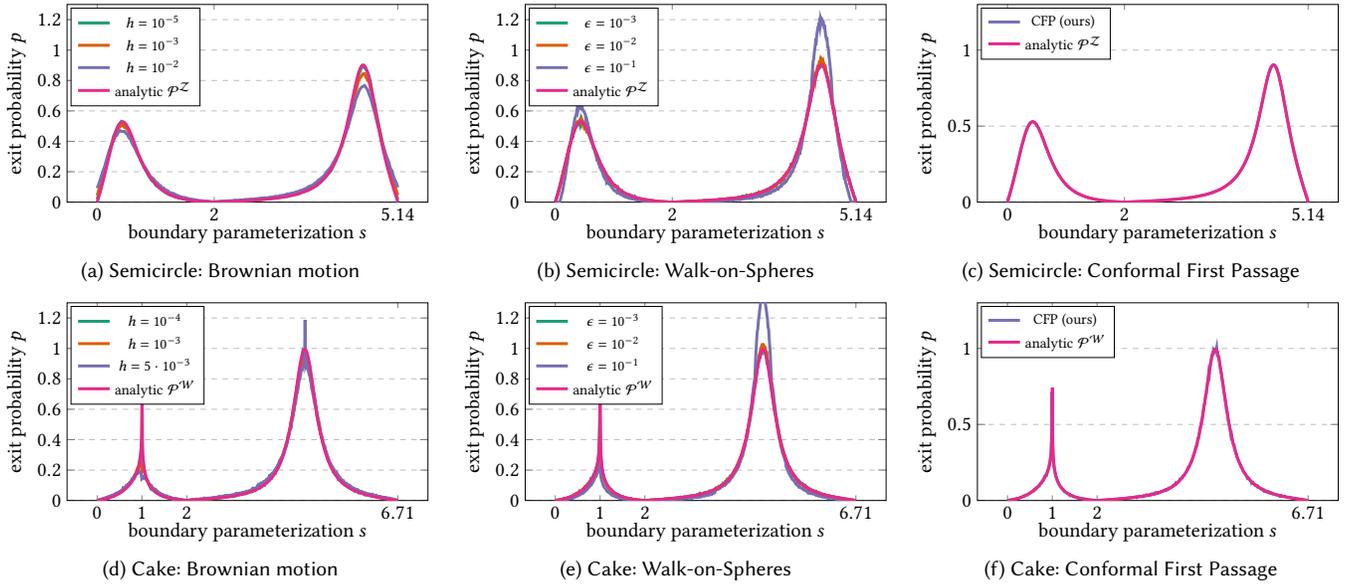


Fig. 6. Visualization of the probability distribution estimated along the boundary. Top: semicircle with $c = 0.4 + 0.6i$, the boundary is parameterized from $[0, 2]$ for the vertical part of the semicircle and from $[2, 2 + \pi]$ for the arc. Bottom: cake with $\alpha = 3/4\pi$, $c = 0.7 + 0.2i$, with the boundary parameterization following the edges at $[0, 1]$ and $[1, 2]$, and the arc from $[2, 2 + 3/2\pi]$. For both, the parameterization s starts at the top left corner and goes counterclockwise.

are biased. In contrast, our conformal first passage is free from bias and matches the analytic ground truth closely. Similar observations can be made for the cake in (d)-(f), where the parameter position $s = 1$ corresponds to the tip of the convex corner.

4.2 Circle Selection Strategy

Our greedy primitive placement strategy aims for the best possible coverage of the domain. An ablation study with and without the different primitives, and the resulting path lengths, is shown in Fig. 7, where the average path length decreased from 4.8 without cakes, to 3.8 without additional semicircles, down to 3.6 when using all primitives. While using semicircles reduces only the average path length, the cake primitives help especially in corners and at segment endings. Adding additional semicircles at places where two semicircles meet gives a modest performance improvement. In Fig. 8, we visualize the geometry placement for all scenes. The circles were omitted from these figures to reduce visual clutter.

4.3 Performance Measurements

In Fig. 8, we compare the runtime performance of the proposed method with the conventional WoS algorithm. Both methods have been implemented in CUDA on the GPU and utilize a bounding volume hierarchy (BVH) with surface-area-heuristic for faster closest-point queries. The BVH is constructed on the CPU using the open source implementation of Tian and Günther [2025]. On the GPU, a stack data structure is used for efficient traversal. The performance measurements were taken on a workstation equipped with an AMD Ryzen 9 7950X CPU and an NVIDIA RTX 4090 GPU. Even though the computation of one step with our method is more involved

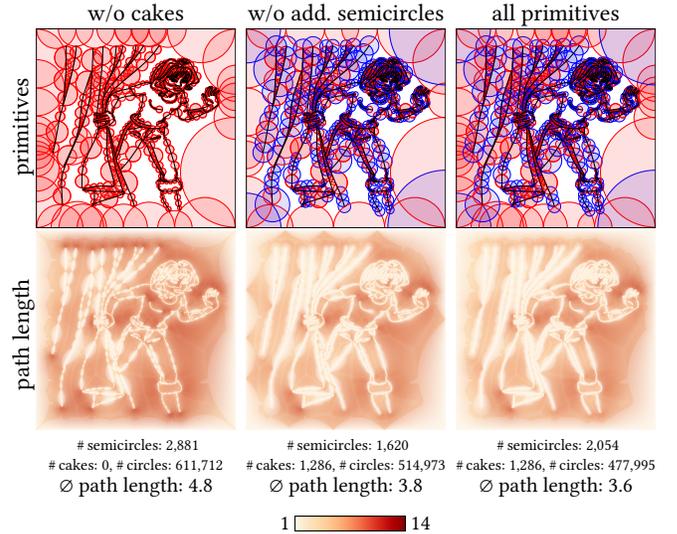


Fig. 7. Geometry placements without cakes, without additional semicircles, and with all primitives. Adding primitives lowers the average path length.

than in the conventional Walk-on-Spheres algorithm, we are able to outperform the WoS by requiring less steps to reach the boundary. Thus, our approach converges further in the same amount of time. To better show the tradeoff between quality and performance inherent to the Walk-on-Spheres epsilon-shell, we conduct a comparison with varying epsilon-shell thicknesses in Fig. 9. The epsilon value is always relative to the domain size. The BVH-accelerated WoS

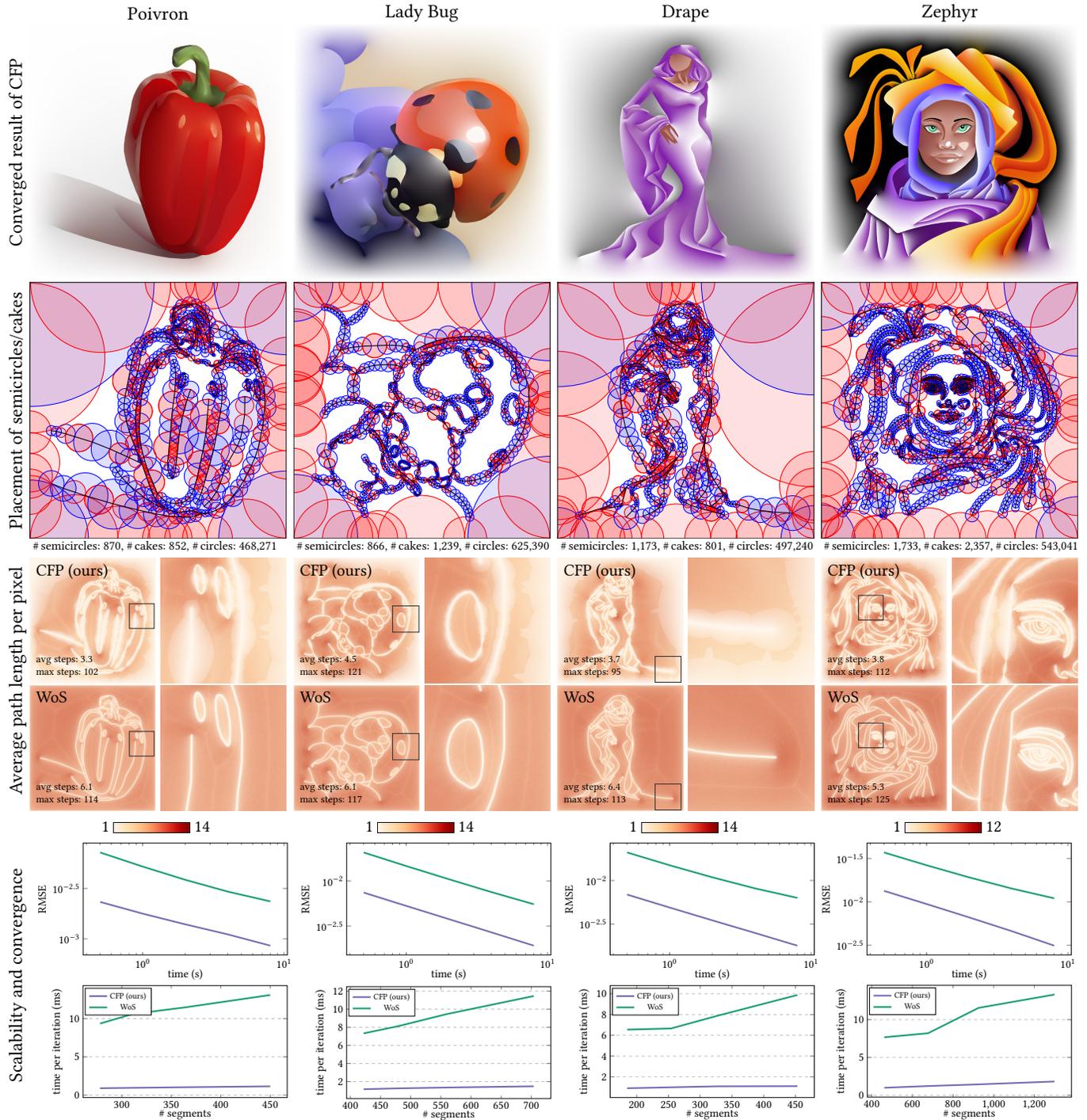


Fig. 8. Results of our method for different test scenes rendered at a resolution of 1024×1024 pixels. From left to right we see the Poivron (450 initial line segments), the Lady Bug (704 initial line segments), the Drape (453 initial line segments), and the Zephyr (1299 initial line segments). The first row shows the converged result of our conformal first passage. The second row depicts the semicircles (red) and cakes (blue) that have been placed in the scene. Note that the full circles cover the remaining space with plenty of overlap to offer many choices to our primitive selection. Below the images, the number of primitives in use is listed. The third row of images compares the average path length of our method with the one from a BVH-accelerated Walk-on-Spheres ($\epsilon = 0.0027$). It can be seen that the path length decreased (brighter colors). Using convergence plots in log-log domain, the fourth row shows that we converge faster than WoS at equal time. The RMSE is measured with respect to the reference in the first row computed with 16,384 Conformal First Passage walks per pixel. Note that the preprocessing time is excluded from this graph. The last row of plots shows the scalability of our method for a growing number of input segments.

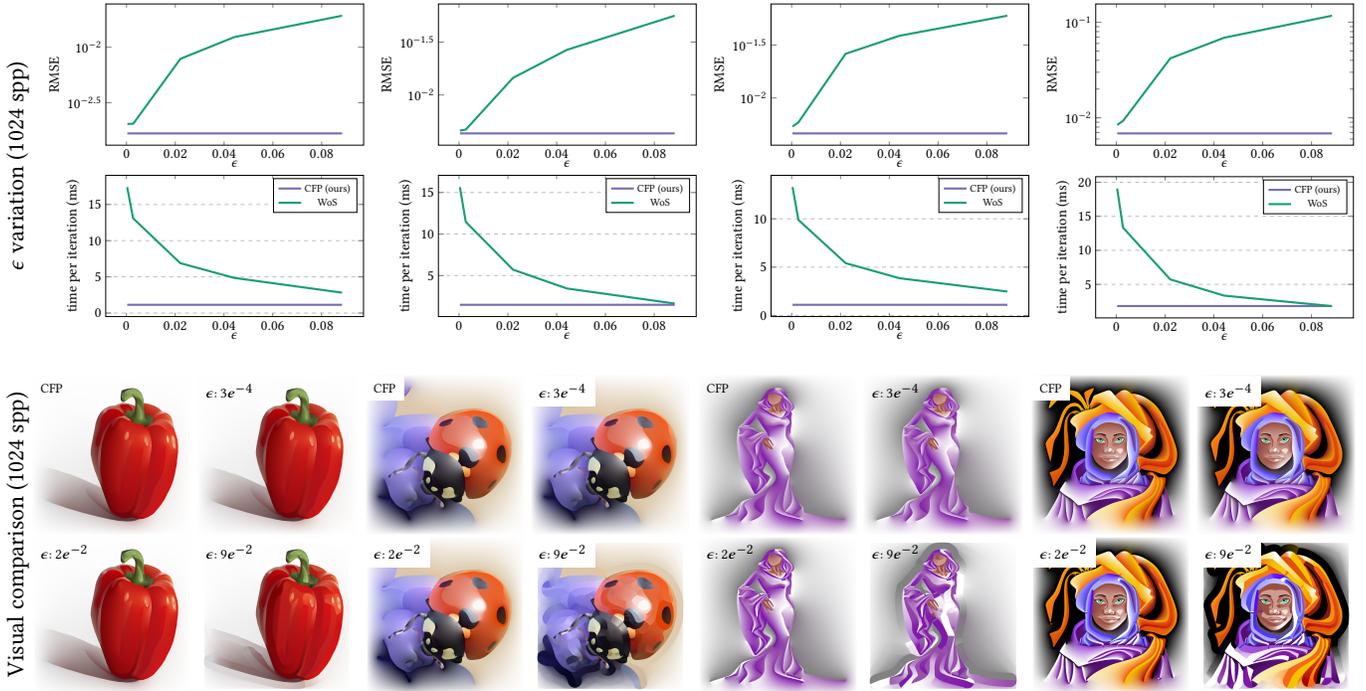


Fig. 9. In this equal time comparison, we vary the size of ϵ . While the comparisons in Fig. 8 were conducted with an epsilon-shell of $\epsilon = 0.0027$, we show here the impact of ϵ on the runtime per iteration, the RMSE, and the visual quality for a fixed sample count of 1,024. ϵ is varied between $3e^{-4}$ and $9e^{-2}$. In all scenes, the latter has approximately equal runtime compared to our approach, which shows the tradeoff between quality and performance. The RMSE is measured with respect to a reference computed with 16,384 Conformal First Passage walks per pixel. A large epsilon shell leads to significant visual discrepancies.

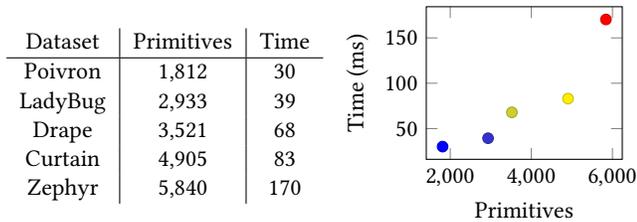


Fig. 10. The precomputation time (in ms) is determined by the number of considered semicircles and cakes, which are here referred to as primitives.

must increase the epsilon by an average of 32 times to close the performance gap with our approach. However, this results in large visual discrepancies at boundaries. In Fig. 10, we investigate what determines the runtime of the most expensive step, i.e., the search for ideal primitives per pixel. We see a direct relationship between the preprocessing time and the number of semicircles and cakes generated in the steps before.

4.4 Lower Number of Path Vertices

Since we walk on primitives that share edges with Dirichlet boundaries, we can take larger steps directly onto the boundaries. As a result, our method requires across all test scenes a lower maximum number of steps compared to the conventional Walk-on-Spheres, although the advantage is sometimes marginal. More importantly,

however, we demonstrate in Fig. 8 that our approach considerably reduces the average path length of the random walks. The visualizations of the path length demonstrate that our primitives facilitate direct jumps to boundaries, which is especially evident on longer boundary segments and at segment endpoints, e.g., in the Drape scene. However, also on scenes with shorter boundary segments, the evaluation demonstrates that the approach reduces the path length over the entire image.

4.5 Lower Error Near Boundaries

Contrary to the conventional Walk-on-Spheres, our conformal first passage formulation uses primitives that share edges with the Dirichlet boundary, allowing for a direct sampling without the need for an epsilon shell. In Fig. 11, we compare the two methods. The conformal first passage exhibits less bias, and interestingly also less numerical errors which would cause unintended color leaking.

4.6 Discretization of Boundaries

We applied our method to diffusion curves, for which the boundary geometry is given by cubic Bézier splines [Orzan et al. 2008]. Since the computation of distances to Bezier curves is expensive, it is not uncommon in the Monte Carlo PDE literature to approximate the boundary with piecewise linear segments. We discretized the curves adaptively using the Ramer-Douglas-Peucker algorithm [Douglas and Peucker 1973; Ramer 1972], choosing the split points directly

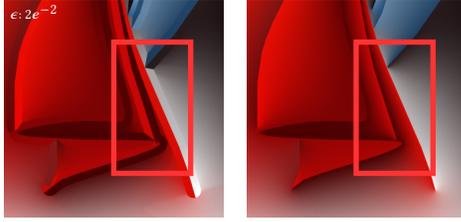


Fig. 11. Left: Using Walk-on-Spheres, artifacts are visible close to segments and in surrounding areas, due to the epsilon-shell (here $\epsilon = 2e^{-2}$). Right: Our approach does not require the epsilon-shell and provides better accuracy, especially close to segments.

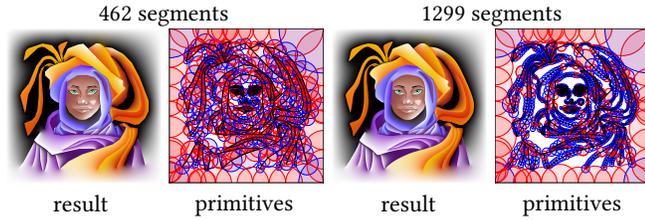


Fig. 12. The initial input curves are cubic Bézier splines. Discretizing them into a higher number of line segments, lowers the possible radius of semicircles and cakes. This results in larger empty areas, which are filled up with circle primitives, which are not visualized here.

from the continuous curve [Sederberg and Nishita 1990]. In the last row of Fig. 8, we evaluate how the curve discretization influences the runtime of our approach. When lowering the permitted approximation error, the number of segments increases, leading to smaller—and thereby more—semicircles and cakes. We observe that the runtime scales linearly in the scene complexity. Visual results of the generated primitives can be seen in Fig. 12.

4.7 Discussion

Poisson Equation. Unlike the Laplace equation, the Poisson equation is not invariant under conformal maps. Nevertheless, it is possible to extend our approach to planar Poisson equations in two dimensions and solve $\Delta u(\mathbf{x}) = f(\mathbf{x})$ with Eq. (2). To compute the interior integral in the BIE [Sawhney et al. 2023] at c , we sample a point $y \in \mathcal{X}$ in the unit disc, map it to the primitive $\mathbf{x}(c, y)$, and evaluate the source term $f(\mathbf{x}(c, y))$, cf. [Olver 2017, page 53]:

$$\mathbf{u}_{\text{int}}(c) = \int_{\mathcal{X}} \mathcal{G}^{\mathcal{X}}(c, y) f(\mathbf{x}(c, y)) \left| \frac{\partial \mathbf{x}(c, y)}{\partial y} \right| dy \quad (28)$$

where $\left| \frac{\partial \mathbf{x}(c, y)}{\partial y} \right|$ is the Jacobian determinant needed to model the change of variables. A result of our approach is shown in Fig. 13.

Piecewise Linear Boundaries. We assumed that the boundary is piecewise linear, since we only used conformal maps for straight boundaries. For circular or elliptic boundaries, conformal maps are likewise available [Kythe 2019], and could be added in the future.

Precomputation. While our approach iterates faster than the Walk-on-Spheres, our method requires a preprocessing to determine the primitives for the walk. For scenarios in which the colors of the

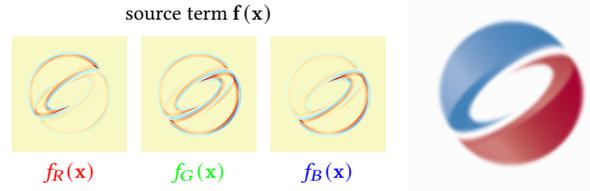


Fig. 13. As demonstrated here, our approach can be extended to the Poisson equation $\Delta u(\mathbf{x}) = f(\mathbf{x})$ in two dimensions.

diffusion curves are changed interactively, the primitives can be reused, since the geometry of the boundary curves has not changed.

Neumann Boundary Conditions. Similar to reflections along Neumann boundaries, it would be imaginable to reflect paths back into the domain. A smarter combination with the Walk-on-Stars is another interesting avenue for future work.

Extension to 3D. Our approach is fundamentally based on conformal maps. In 3D, the construction of conformal maps is significantly more constrained than in two dimensions, as demonstrated by Liouville's theorem. A generalization to 3D is therefore challenging.

Invariance to Primitive Placement. In contrast to discretization choices in boundary element methods (BEMs) and in mesh-based solvers, our primitive placement does not affect the quality of the result, i.e., our approach remains unbiased.

5 Conclusions

We presented a novel Monte Carlo approach to solve Laplace equations with Dirichlet boundary conditions in two dimensions. While Walk-on-Spheres [Muller 1956; Sawhney and Crane 2020] uses largest empty circles, which touch the boundary in only one point, we utilized semicircles and circle sectors (cakes) that can nestle against one or two boundary edges. Walks are thereby more likely to terminate after fewer steps. Unlike a previous first passage approach [Given et al. 1997], we used conformal maps to derive Green's functions and Poisson kernels, allowing us to utilize circle sectors that are well-suited for corners. Further, we precomputed the set of geometric primitives on which the walk is performed. We confirmed the correctness of our Monte Carlo walks by comparing with analytic first passage probability distributions, Brownian motion simulations, and with the Walk-on-Spheres. We demonstrated a reduced number of walking steps compared to previous work, resulting in faster computation times once the primitives are placed. We believe that our results show the potential of first passage algorithms in conjunction with conformal maps, which offers new avenues for future research. For example, we plan to continue the work on Poisson equations, and extend the method to handle Neumann boundary conditions, which likely results in a combination with existing methods for such problems [Miller et al. 2024b; Sawhney et al. 2023; Sugimoto et al. 2023]. Further, other primitives [Johnston et al. 2005] could be used to improve the domain coverage.

Acknowledgments

This work was supported by DFG grant no. GU 945/3-1.

References

- Seungbae Bang, Kirill Serkh, Oded Stein, and Alec Jacobson. 2023. An Adaptive Fast-Multipole-Accelerated Hybrid Boundary Integral Equation Method for Accurate Diffusion Curves. *ACM Trans. Graph.* 42, 6, Article 215 (Dec. 2023), 28 pages. doi:10.1145/3618374
- Jean-Michel Bergheau and Roland Fortunier. 2013. *Finite element simulation of heat transfer*. John Wiley & Sons, London, UK. doi:10.1002/9780470611418
- Iliia Binder and Mark Braverman. 2012. The rate of convergence of the walk on spheres algorithm. *Geometric and Functional Analysis* 22, 3 (2012), 558–587. doi:10.1007/s00039-012-0161-z
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. AK Peters/CRC Press, Natick, Massachusetts, USA. doi:10.1201/9781315266008
- Jiong Chen, Florian Schaefer, and Mathieu Desbrun. 2024. Lightning-fast Method of Fundamental Solutions. *ACM Trans. Graph.* 43, 4, Article 77 (July 2024), 16 pages. doi:10.1145/3658199
- Gary P. T. Choi, Yusan Leung-Liu, Xianfeng Gu, and Lok Ming Lui. 2020. Parallelizable Global Conformal Parameterization of Simply-Connected Surfaces via Partial Welding. *SIAM Journal on Imaging Sciences* 13, 3 (2020), 1049–1083. doi:10.1137/19M125337X
- Martin Costabel. 1987. Principles of boundary element methods. *Computer Physics Reports* 6, 1-6 (1987), 243–274. doi:10.1016/0167-7977(87)90014-1
- Michael Czekanski, Benjamin Faber, Margaret Fairborn, Adelle Wright, and David Bindel. 2024. Walking on Spheres and Talking to Neighbors: Variance Reduction for Laplace's Equation. arXiv:2404.17692 [physics.comp-ph]
- Auguste De Lambilly, Gabriel Benedetti, Nour Rizk, Chen Hanqi, Siyuan Huang, Junnan Qiu, David Louapre, Raphael Granier De Cassagnac, and Damien Rohmer. 2023. Heat Simulation on Meshless Crafted-Made Shapes. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Rennes, France). ACM, New York, NY, USA, Article 9, 7 pages. doi:10.1145/3623264.3624457
- David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization* 10, 2 (1973), 112–122. doi:10.3138/FM57-6770-U75U-7727
- Janick Martínez Esturo, Maik Schulze, Christian Röss, and Holger Theisel. 2013. Poisson-based tools for flow visualization. In *IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, Sydney, NSW, Australia, 241–248. doi:10.1109/PacificVis.2013.6596151
- 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 (03 1997), 3761–3771. doi:10.1063/1.473428
- P Glasserman. 2004. *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY, USA. doi:10.1007/978-0-387-21617-1
- Fei Hou, Qian Sun, Zheng Fang, Yong-Jin Liu, Shi-Min Hu, Hong Qin, Aimin Hao, and Ying He. 2018. Poisson vector graphics (PVG). *IEEE Transactions on Visualization and Computer Graphics* 26, 2 (2018), 1361–1371. doi:10.1109/TVCG.2018.2867478
- Chi-Ok Hwang, Sungpyo Hong, and Jinwoo Kim. 2015. Off-centered “Walk-on-Spheres” (WOS) algorithm. *J. Comput. Phys.* 303 (2015), 331–335. doi:10.1016/j.jcp.2015.10.002
- John David Jackson. 1998. *Classical electrodynamics* (3 ed.). John Wiley & Sons, Nashville, TN. ISBN:978-0-471-30932-1.
- Stefan Jeschke, David Cline, and Peter Wonka. 2009. A GPU Laplacian Solver for Diffusion Curves and Poisson Image Editing. *ACM Transactions on Graphics* 28, 5 (2009), 1–8. doi:10.1145/1618452.1618462
- Michelle E Johnston, Jan C Myland, and Keith B Oldham. 2005. A Green function for the equilateral triangle. *Zeitschrift für angewandte Mathematik und Physik ZAMP* 56 (2005), 31–44. doi:10.1007/s00033-004-3114-z
- Douglas Samuel Jones. 1964. *The theory of electromagnetism*. Elsevier, Oxford, UK.
- Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages. doi:10.1145/2766997
- Prem K Kythe. 2019. *Handbook of conformal mappings and applications*. Chapman and Hall/CRC, New York, NY, USA. doi:10.1201/9781315180236
- 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* (Sydney, NSW, Australia). ACM, New York, NY, USA, Article 34, 10 pages. doi:10.1145/3610548.3618141
- Michael Mascagni and Nikolai A Simonov. 2004. Monte Carlo methods for calculating some physical properties of large molecules. *SIAM journal on scientific computing* 26, 1 (2004), 339–357. doi:10.1137/S1064827503422221
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4, Article 82 (jul 2023), 11 pages. doi:10.1145/3592400
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024a. Differential Walk on Spheres. *ACM Trans. Graph.* 43, 6, Article 174 (Nov. 2024), 18 pages. doi:10.1145/3687913
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024b. Walkin' Robin: Walk on Stars with Robin Boundary Conditions. *ACM Trans. Graph.* 43, 4, Article 41 (jul 2024), 18 pages. doi:10.1145/3658153
- Mervin E Muller. 1956. Some continuous Monte Carlo methods for the Dirichlet problem. *The Annals of Mathematical Statistics* 27, 3 (1956), 569–589. doi:10.1214/aoms/1177728169
- Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin transformations for simulations on infinite domains. *ACM Trans. Graph.* 40, 4, Article 97 (July 2021), 15 pages. doi:10.1145/3450626.3459809
- Peter J Olver. 2017. Complex analysis and conformal mapping. Lecture Notes, University of Minnesota, available at https://www-users.cse.umn.edu/olver/ln_cml.pdf, accessed on April 17, 2025.
- Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion curves: a vector representation for smooth-shaded images. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–8. doi:10.1145/1360612.1360691
- Eric Poisson and Clifford M Will. 2014. *Gravity: Newtonian, post-newtonian, relativistic*. Cambridge University Press, Cambridge, UK. doi:10.1017/CBO9781139507486
- Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* 41, 4 (2022), 51–62. doi:10.1111/cgf.14586
- Urs Ramer. 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing* 1, 3 (1972), 244–256. doi:10.1016/S0146-664X(72)80017-0
- Damien Rioux-Lavoie, Ryusuke Sugimoto, Tümay Özdemir, Naoharu H. Shimada, Christopher Batty, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2022. A Monte Carlo Method for Fluid Simulation. *ACM Trans. Graph.* 41, 6, Article 240 (Nov. 2022), 16 pages. doi:10.1145/3550454.3555450
- Karl Karlovich Sabelfeld. 1982. Vector algorithms in the Monte-Carlo method for solving systems of second-order elliptic equations and Lamé's equation. *Doklady Akademii Nauk* 262, 5 (1982), 1076–1080. <https://www.mathnet.ru/eng/dan45069> (in Russian).
- Karl K. Sabelfeld and Nikolai A. Simonov. 1994. *Random Walks on Boundary for Solving PDEs*. De Gruyter, Berlin, Boston. doi:10.1515/9783110942026
- Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* 37, 1, Article 5 (Dec. 2017), 14 pages. doi:10.1145/3132705
- Rohan Sawhney and Keenan Crane. 2020. Monte Carlo geometry processing: a grid-free approach to PDE-based methods on volumetric domains. *ACM Trans. Graph.* 39, 4, Article 123 (Aug. 2020), 18 pages. doi:10.1145/3386569.3392374
- Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4, Article 80 (jul 2023), 20 pages. doi:10.1145/3592398
- T.W. Sederberg and T. Nishita. 1990. Curve intersection using Bézier clipping. *Computer-Aided Design* 22, 9 (1990), 538–549. doi:10.1016/0010-4485(90)90039-F
- Marie A Snipes and Lesley A Ward. 2016. Harmonic measure distributions of planar domains: a survey. *The Journal of Analysis* 24 (2016), 293–330. doi:10.1007/s41478-016-0019-0
- Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2024a. Velocity-Based Monte Carlo Fluids. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 8, 11 pages. doi:10.1145/3641519.3657405
- Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4, Article 81 (jul 2023), 16 pages. doi:10.1145/3592109
- Ryusuke Sugimoto, Nathan King, Toshiya Hachisuka, and Christopher Batty. 2024b. Projected Walk on Spheres: A Monte Carlo Closest Point Method for Surface PDEs. In *SIGGRAPH Asia 2024 Conference Papers* (SA '24). ACM, Tokyo, Japan, 1–10. doi:10.1145/3680528.3687599
- Xingze Tian and Tobias Günther. 2025. Unified Smooth Vector Graphics: Modeling Gradient Meshes and Curve-based Approaches Jointly as Poisson Problem. *IEEE Transactions on Visualization and Computer Graphics*, to appear. doi:10.1109/TVCG.2025.3558263
- Jasper Van de Gronde. 2011. *A High Quality Solver for Diffusion Curves*. Master's thesis. University of Groningen.
- Xiaokun Wang, Yanrui Xu, Sinuo Liu, Bo Ren, Jiri Kosinka, Alexandru C. Telea, Jiamin Wang, Chongming Song, Jian Chang, Chenfeng Zhang Li, Jun Jian, and Xiaojuan Ban. 2024. Physics-based fluid simulation in computer graphics: Survey, research trends, and challenges. *Computational Visual Media* 10 (2024), 803–858. Issue 5. doi:10.1007/s41095-023-0368-y
- Ofir Weber and Craig Gotsman. 2010. Controllable conformal maps for shape deformation and interpolation. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) (SIGGRAPH '10). Association for Computing Machinery, New York, NY, USA, Article 78, 11 pages. doi:10.1145/1833349.1778815
- Ekrem Fatih Yilmazer, Delio Vicini, and Wenzel Jakob. 2024. Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators. *ACM Trans. Graph.* 43, 6, Article 175 (Nov. 2024), 18 pages. doi:10.1145/3687990
- Zihan Yu, Lifan Wu, Zhiqian Zhou, and Shuang Zhao. 2024. A Differential Monte Carlo Solver For the Poisson Equation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. doi:10.1145/3641519.3657460