Monte Carlo Geometry Processing A Grid-Free Approach to PDEs

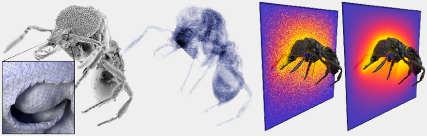
Rohan Sawhney, Keenan Crane

Baran Celik Swann Cordier Antoine Le Maguet

Master MVA : Geometric data analysis

January 5, 2026

- **Goal:** solve PDEs on complex domains **without volumetric meshing** (Monte Carlo).
- **Why avoid meshes?**
  - Meshing is costly and brittle (hours; holes / noisy scans).
  - A mesh **approximates** geometry $\Rightarrow$ loss of detail + geometric error.
- **Takeaway:** work directly on raw geometry $\Rightarrow$ **Walk-on-Spheres**.



From Brownian Motion to Walk on Spheres

Problem

$$\begin{cases} \Delta u(x) = 0 & x \in \Omega \\ u(x) = g(x) & x \in \partial\Omega \end{cases}$$

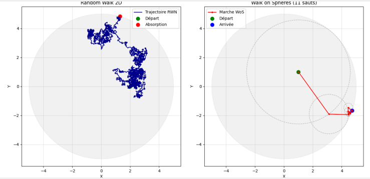
(Laplace equation)

Kakutani's principle

$$u(x) = \mathbb{E}[g(W_\tau) \mid W_0 = x]$$

Mean Value Property

$$u(x) = \frac{1}{|\partial B(x)|} \int_{\partial B(x)} u(y)\, dy$$



2D Random walk VS WoS

Idea

Brownian motion $\rightarrow$ many small steps

Walk on Spheres $\rightarrow$ direct jumps to sphere boundaries

Monte Carlo Estimator

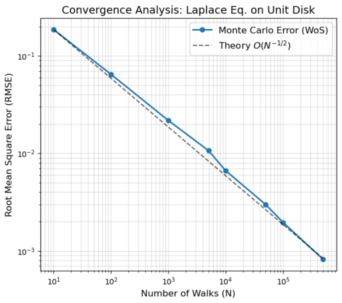
We simulate n WoS trajectory : $x_0, x_{1_i}, ..., x_{s_i}$ for $1 \leq i \leq n$.

Our estimator is then:

$$\hat{u}_n(x) = \frac{1}{n}\sum_{i=1}^{n} g(x_{s_i}).$$

Monte Carlo convergence :

$$\begin{aligned}\mathrm{RMSE} &= \sqrt{\mathbb{E}[(\hat{u}-u)^2]} \\ &= \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)\end{aligned}$$



RMSE convergence of the Monte Carlo estimator

Algorithm

Require: Query point x_0, Boundary $\partial\Omega$, Tolerance ϵ, Number of walks N

$u_{sum} \leftarrow 0$
for $i \leftarrow 1$ **to** N **do**
    $x \leftarrow x_0$ ▷ Start a new walk
    while $d(x, \partial\Omega) > \epsilon$ **do**
        $R \leftarrow d(x, \partial\Omega)$
        $\vec{v} \leftarrow$ UniformSampleUnitSphere$()$
        $x \leftarrow x + R \cdot \vec{v}$ ▷ Jump to sphere surface
    end while
    $x_b \leftarrow$ ClosestPoint$(x, \partial\Omega)$
    $u_{sum} \leftarrow u_{sum} + g(x_b)$ ▷ Accumulate boundary value
end for
return u_{sum}/N ▷ Return average

Extending to Poisson: handling source terms

Problem

$$\begin{cases} \Delta u(x) = f(x) & x \in \Omega \\ u(x) = g(x) & x \in \partial\Omega \end{cases}$$

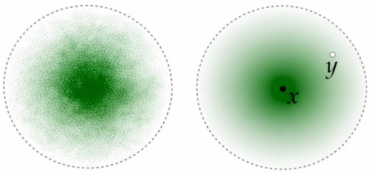
(Poisson equation)

Stochastic representation (any domain)

$$\mathbb{E}\Big[\int_0^T f(W_t)dt \Big| W_0 = x\Big]$$

Integral representation (source term on ball)

$$\int_{B(x)} f(y)G(x,y)dy$$

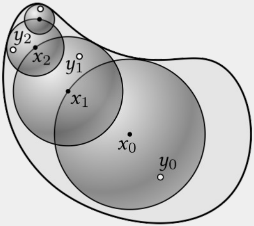


Harmonic Green's Function $G(x,y)$

Monte Carlo estimator

$$\hat{u}(x_k) = \begin{cases} g(\bar{x}_k), & x_k \in \partial\Omega_\varepsilon, \\ \hat{u}(x_{k+1}) + |B(x_k)|\, f(y_k)\, G(x_k, y_k), \end{cases}$$

ALGORITHM



Require: Query point x_0, Boundary $\partial\Omega$, Tolerance ϵ, Number of walks N

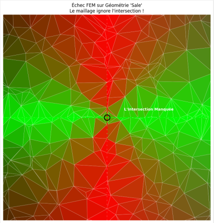
$u_{sum} \leftarrow 0$
for $i \leftarrow 1$ **to** N **do**
  $x \leftarrow x_0$ ▷ Start a new walk
  while $d(x, \partial\Omega) > \epsilon$ **do**
    $R \leftarrow d(x, \partial\Omega)$
    $y \leftarrow$ UniformSampleBall(x, R) ▷ Random point inside ball
    $u_{sum} \leftarrow u_{sum} + \pi R^2\, G(x, y, R)\, f(y)$ ▷ Source contribution
    $\vec{v} \leftarrow$ UniformSampleUnitSphere$()$
    $x \leftarrow x + R \cdot \vec{v}$ ▷ Jump to sphere surface
  end while
  $x_b \leftarrow$ ClosestPoint$(x, \partial\Omega)$
  $u_{sum} \leftarrow u_{sum} + g(x_b)$ ▷ Accumulate boundary value
end for
return u_{sum}/N ▷ Return average

Problem

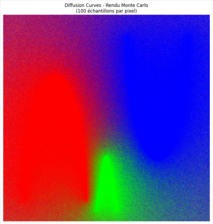
- Challenge: Impossible to mesh cleanly for FEM.
- If we extend this diffusion problem in 3D, the computational time and memory take by pre-processing will explode.

Result

- infinite zoom
- parameter ϵ
- mesh.



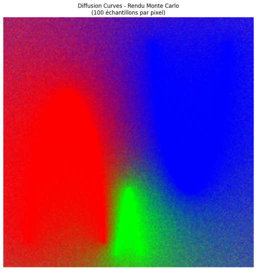
Problem with FEM



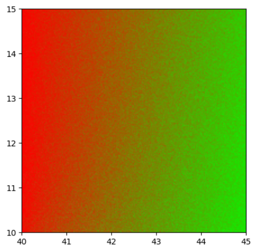
diffusion with Bézier curves

Local Evaluation Property

Unlike FEM, we can compute the solution at a specific window without solving the global system.



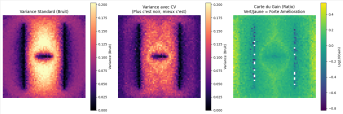
Global View



Zoom $\times 10$

Variance Reduction

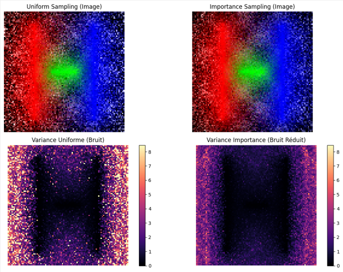
Problem: Monte Carlo noise is high for complex sources. We have to reduce variance.



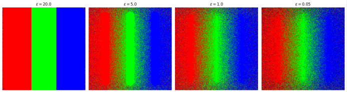
Techniques Implemented:

- **Control Variates**
- **Importance Sampling**

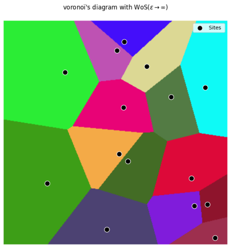
$\rightarrow$ Significant reduction in RMSE for the same computational budget.



PARAMETER ϵ



effet de ϵ



Effect when ϵ tends to infinity

- **Problem:** Will ϵ change our final figure?
- Computational cost = $O(log(1/\epsilon))$
- A large tolerance can be useful in other applications like Voronoï diagram.

Critical Limitations

We identified two major bottlenecks in our stress tests:

1. The "Blindness" Problem

- *Scenario:* Narrow tunnels (dumbbell shape).
- *Issue:* Random walkers get trapped, bouncing inside the tunnel.
- *Result:* Convergence takes minutes vs seconds for FEM.

2. Gradient Explosion

- *Scenario:* Evaluation near boundaries.
- *Issue:* Estimator contains a $1/R$ term. As $R \to 0$, variance $\to \infty$.
- *Result:* Extreme noise at the edges.

Conclusion

Summary

- **WoS is a powerful alternative** for geometry processing, specifically for "dirty" or non-manifold geometry.
- **Key strengths:** Implementation simplicity, infinite resolution, memory efficiency.

Future Work

- **Walk on Stars:** To solve the "tunnel" issue by allowing larger steps in star-shaped domains.