CS482 Nguyen Minh Hieu - 20210722

Monte Carlo Geometry Processing

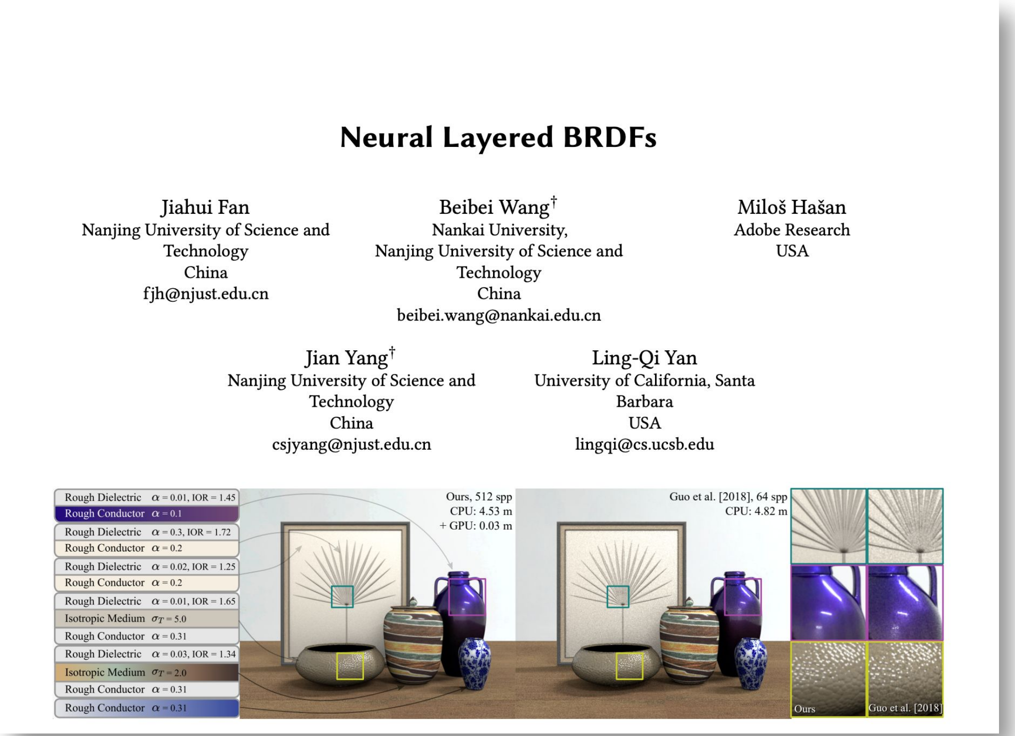
Rohan Sawhney, **Keenan Crane**
Carnegie Mellon University

*certain images that are taken from the authors' paper and video

Review of Last Week

Neural Layered BRDFs

- Parametrize Spatially-varying BRDFs via AutoEncoder
- Predict Consecutive Layering via Neural Network
- Layering is achieved by recursively predict layering effect



Neural Layered BRDFs

Jiahui Fan
Nanjing University of Science and Technology
China
fjh@njust.edu.cn

Beibei Wang[†]
Nankai University,
Nanjing University of Science and Technology
China
beibei.wang@nankai.edu.cn

Miloš Hašan
Adobe Research
USA

Jian Yang[†]
Nanjing University of Science and Technology
China
csjyang@njust.edu.cn

Ling-Qi Yan
University of California, Santa Barbara
USA
lingqi@cs.ucsb.edu

Overview

1. Motivation
2. Methods
3. Experiments
4. Main Takeaways

1. Motivation

Motivation - PDE is Everywhere!

Rendering

- Wave-based Light Transport
- Quantum Optical Simulation

Animation

- Fluid Simulation
- Sound Propagation
- Sound Synthesis
- Fracture Generation

Modeling

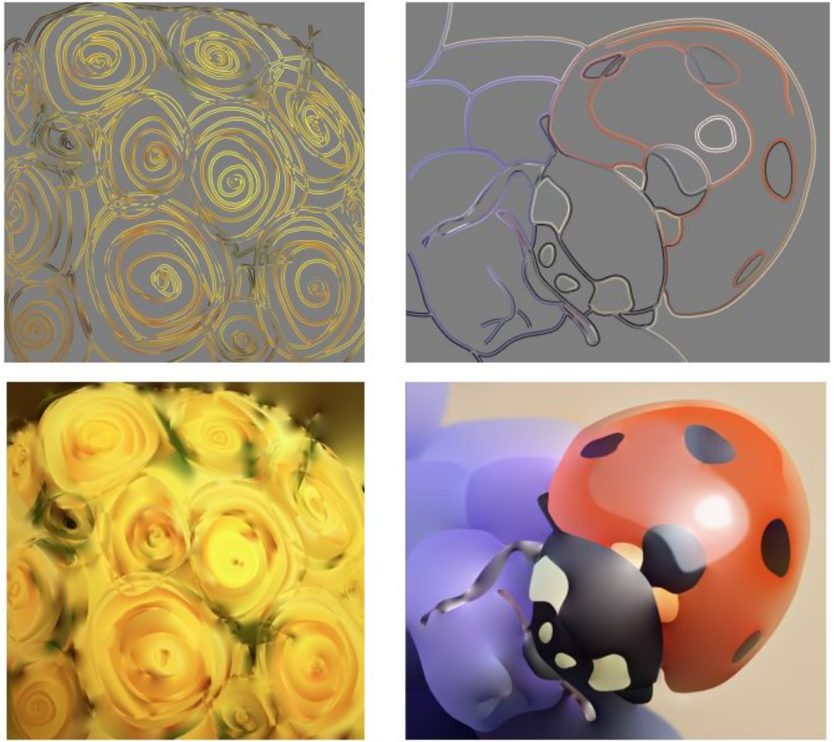
- Shape Deformation
- Physically-based Design

The Famous Interpolation Problem

given boundary color

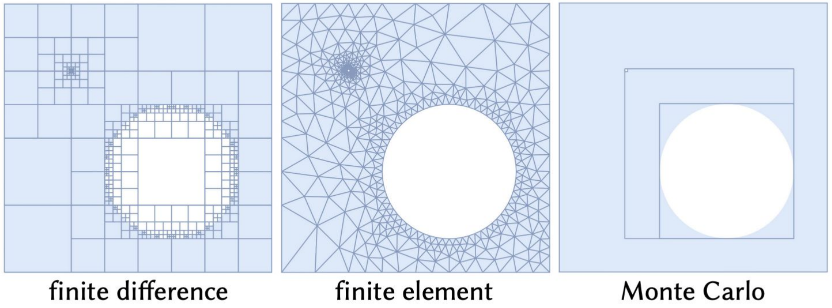
"heat diffuse" from
boundary to fill in colors

$$\Delta u = 0 \quad \text{on } \Omega,$$
$$u = g \quad \text{on } \partial\Omega.$$

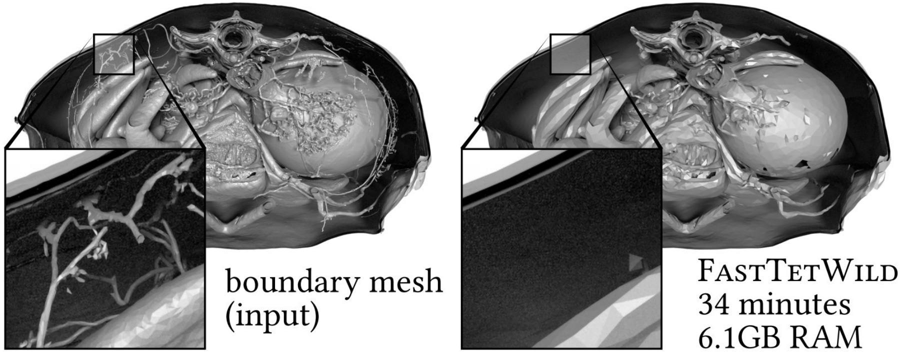


Motivation - Previous Approach

can't we just discretize?



Discretization Error



Complexity

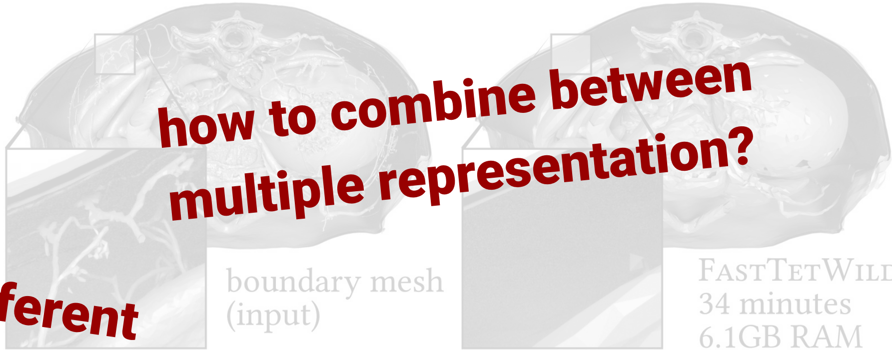
Motivation - Previous Approach

can't we just discretize?

Discretization Error

which discretization?

how to realize with different representation?



Complexity

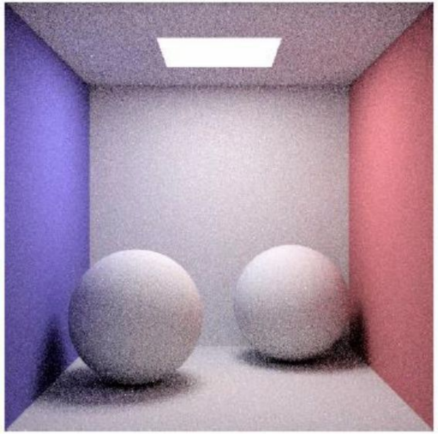
how to combine between multiple representation?

Motivation - Previous Approach

can't we just discretize?

From Donald Fong's s slides

Radiosity



Ray Tracing

Motivation - Monte Carlo for PDE

- agnostic to representation
- parallelizable
- easy to implement
- fast convergence
- no precompute
- easy to realize on most PDEs
- unbiased method
- robust to noise, numerically stable
- easy to compute div, grad, curl

2. The Main Idea

The Tales of Three Equations

Once, there was a parabolic monster …

$$\frac{\partial u}{\partial t}(x,t) + \mu(x,t)\frac{\partial u}{\partial x}(x,t) + \tfrac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2}(x,t) - V(x,t)u(x,t) + f(x,t) = 0$$

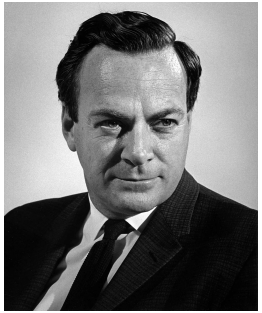
The Tales of Three Equations

Then a legendary duo appeared with a magical integral sword…

$$\frac{\partial u}{\partial t}(x,t) + \mu(x,t)\frac{\partial u}{\partial x}(x,t) + \tfrac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2}(x,t) - V(x,t)u(x,t) + f(x,t) = 0$$

$$u(x,t) = E\left[\exp(-\int_t^T V(X_\tau,\tau)\,d\tau)\psi(X_T) + \int_t^T \exp(-\int_t^s V(X_\tau,\tau)\,d\tau) f(X_s,s)\,ds \,\middle|\, X_t = x\right]$$

Feynman-Kac Theorem

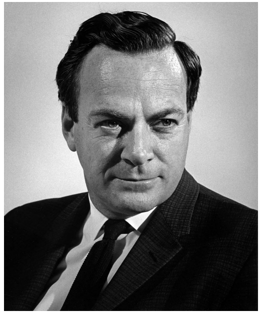


The Tales of Three Equations

Reducing the parabolic monster into a harmless form

$$dX_t = \mu(X,t)\,dt + \sigma(X,t)\,dW_t^Q$$

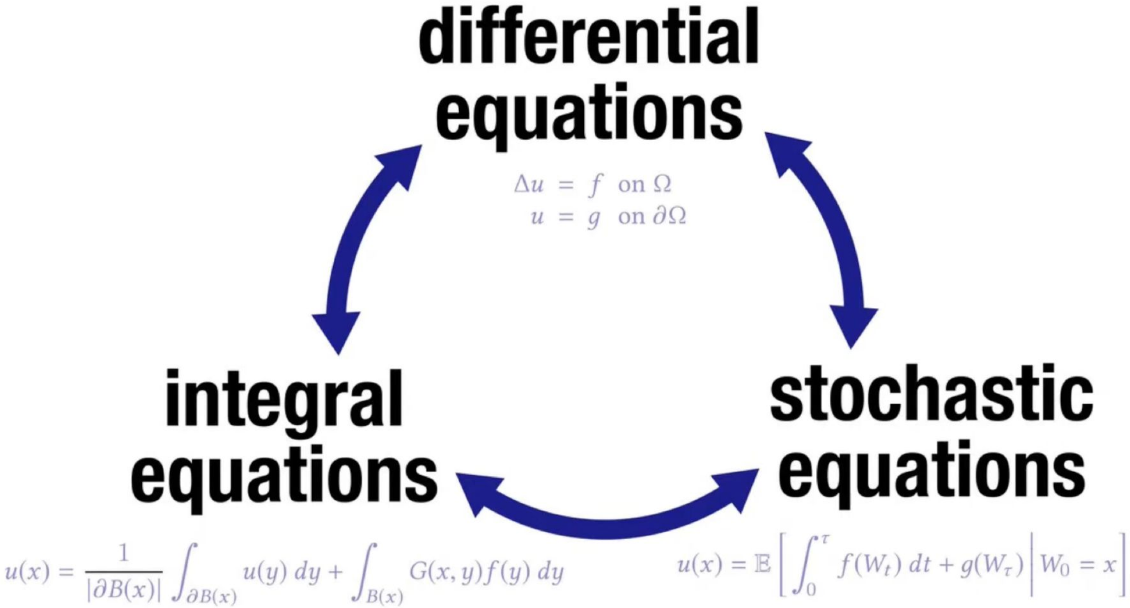
$$u(x,t) = E\left[\exp\left(-\int_t^T V(X_\tau,\tau)\,d\tau\right)\psi(X_T) + \int_t^T \exp\left(-\int_t^s V(X_\tau,\tau)\,d\tau\right) f(X_s,s)\,ds \,\middle|\, X_t = x\right]$$



Feynman-Kac Theorem

The Tales of Three Equations

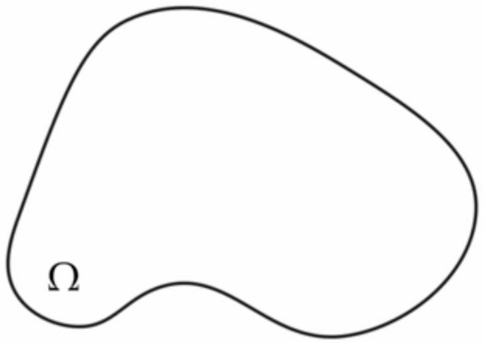
The Cycle continues



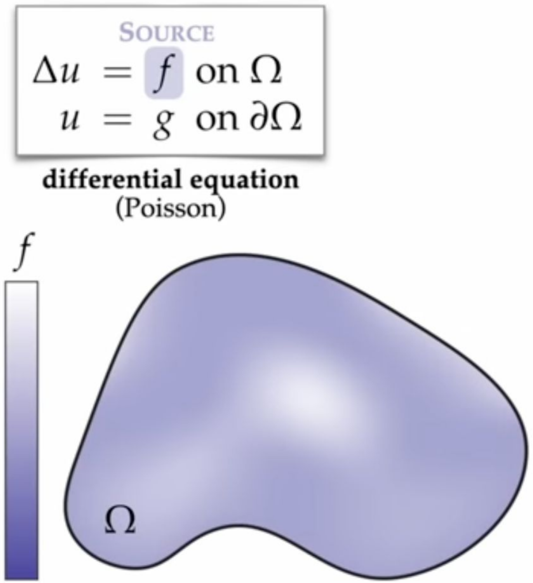
Walk-on-Sphere - An Example

$$\begin{aligned} \Delta u &= f \text{ on } \Omega \\ u &= g \text{ on } \partial\Omega \end{aligned}$$

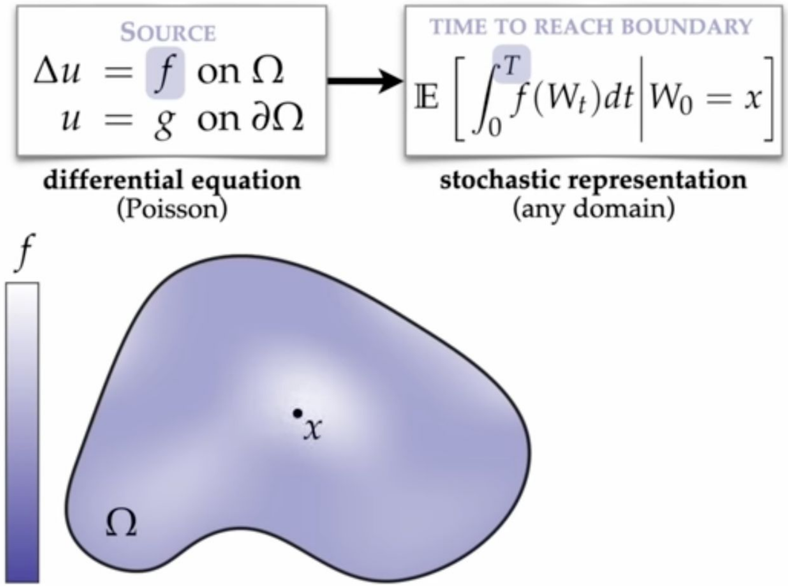
differential equation
(Poisson)



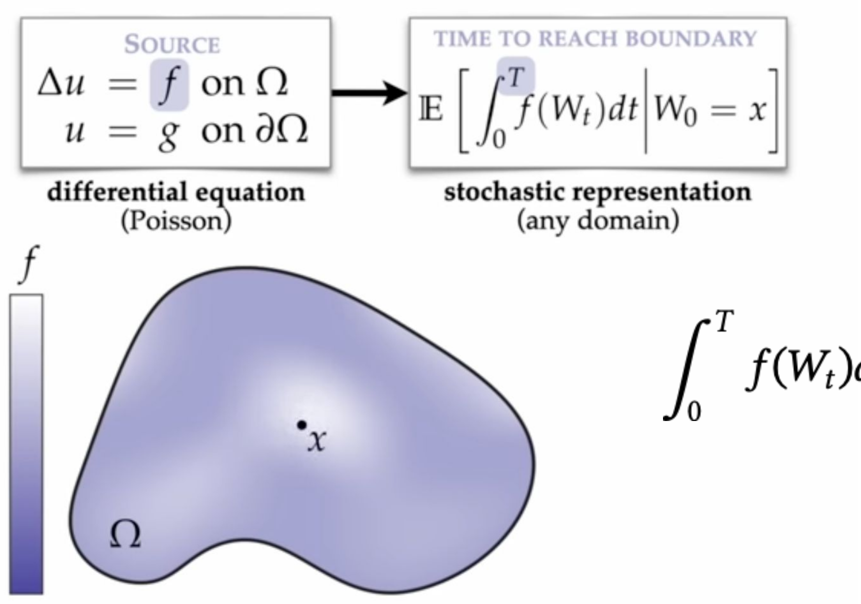
Walk-on-Sphere - An Example



Walk-on-Sphere - An Example



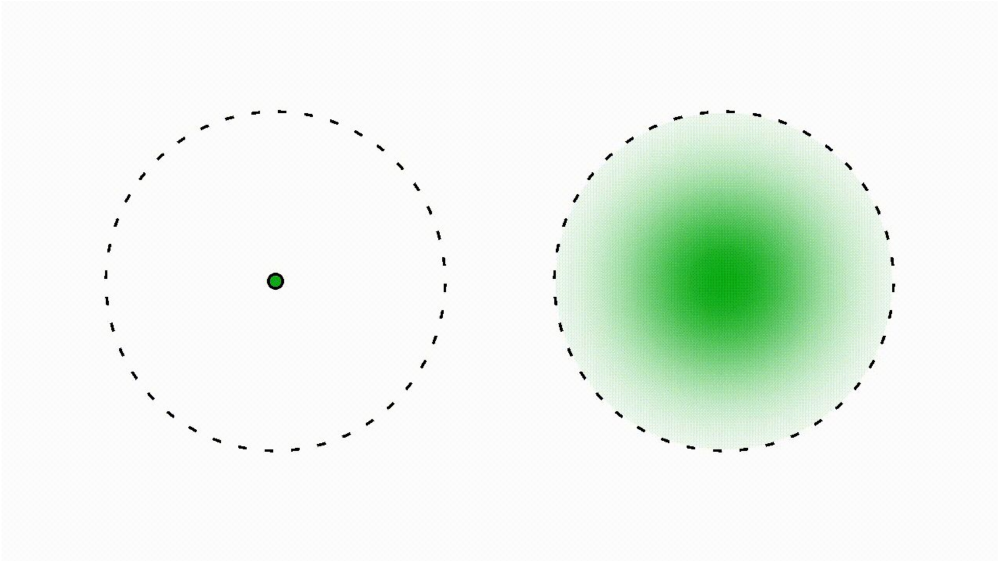
Walk-on-Sphere - An Example



$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$

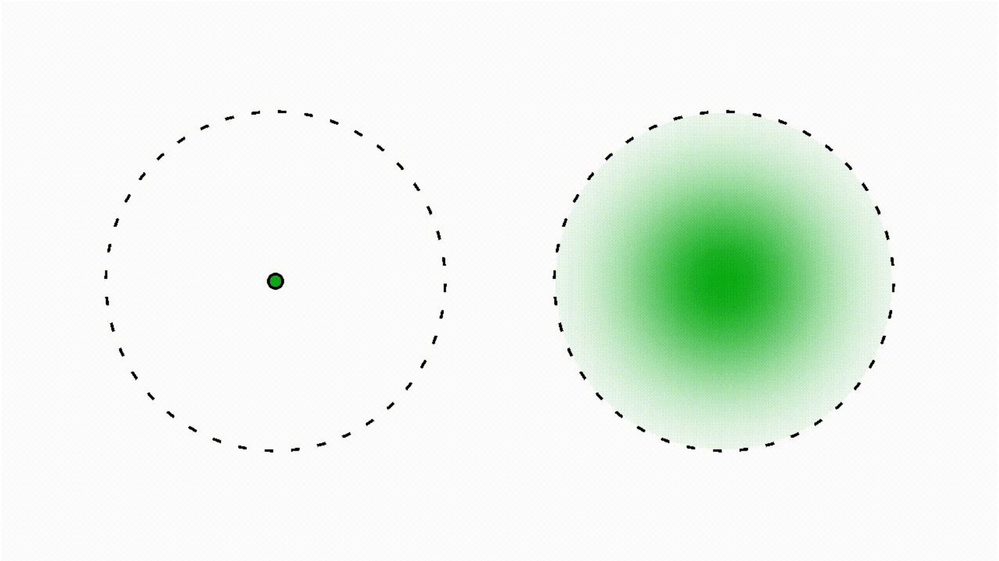
Walk-on-Sphere - Simple Domain

$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$



Walk-on-Sphere - Simple Domain

$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$



Analytic Solution!

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

also known as the Green's Function

Walk-on-Sphere - Simple Domain

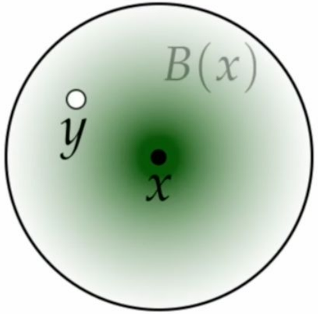
$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$

<u>**Monte Carlo estimator**</u>

N=1

$$|B(x)| f(y) G(x,y), \quad y \sim \mathcal{U}_{B(x)}$$

volume of ball — Green's function — uniform distribution on ball



Analytic Solution!

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

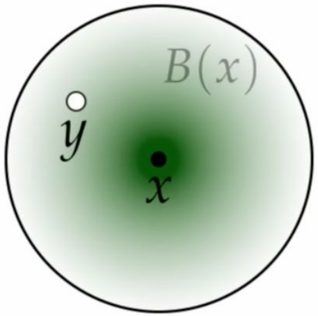
Walk-on-Sphere - Simple Domain

$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$

Monte Carlo estimator N=1

$$|B(x)|f(y)G(x,y), \quad y \sim \mathcal{U}_{B(x)}$$

volume of ball — Green's function — uniform distribution on ball



Analytic Solution!

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

We can solve this in O(1)!
(STILL UNBIASED)

Walk-on-Sphere - Simple Domain

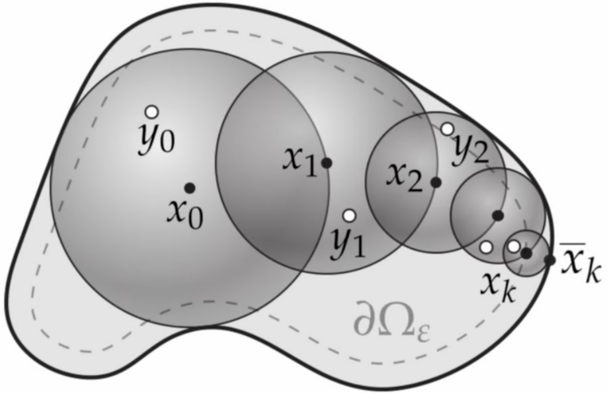
$$\int_0^T f(W_t)dt = \int_0^T f(dW_t) \text{ where } dW_t \sim \mathcal{N}(0; dt)$$

Can we decompose complex domain into simple domain?

Walk-on-Sphere - Simple Domain

Monte Carlo estimator

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



Yes!

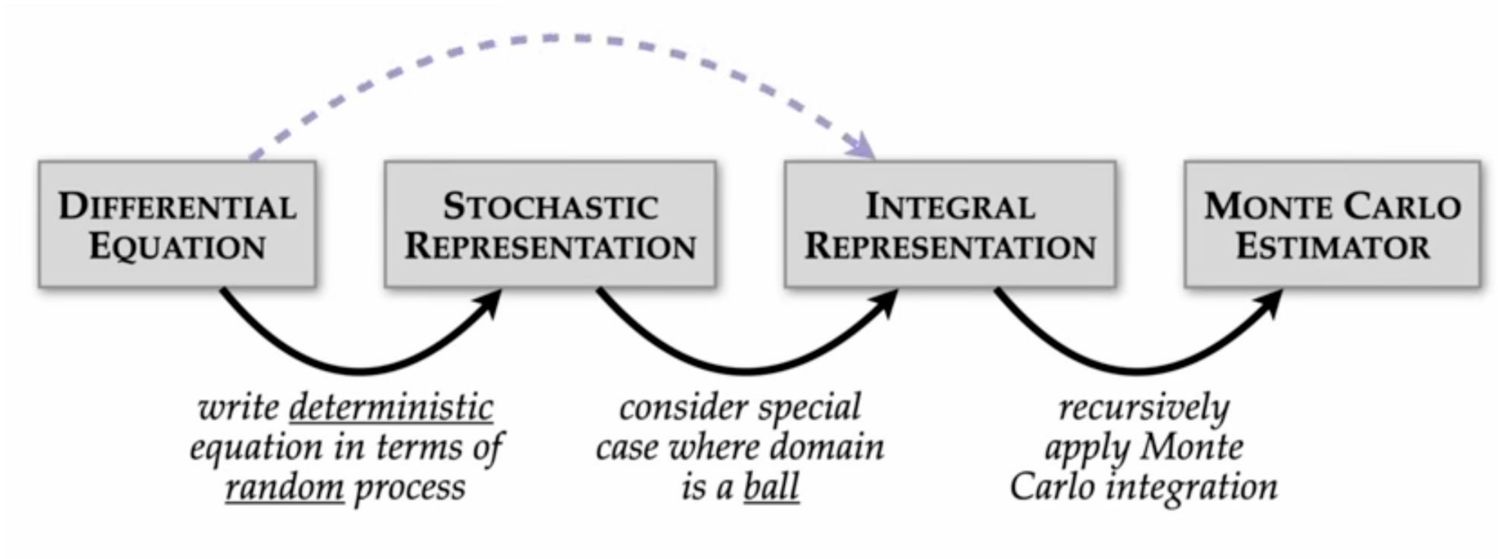
via Mean Value Theorem

solution (unknown!) | solution (unknown!)

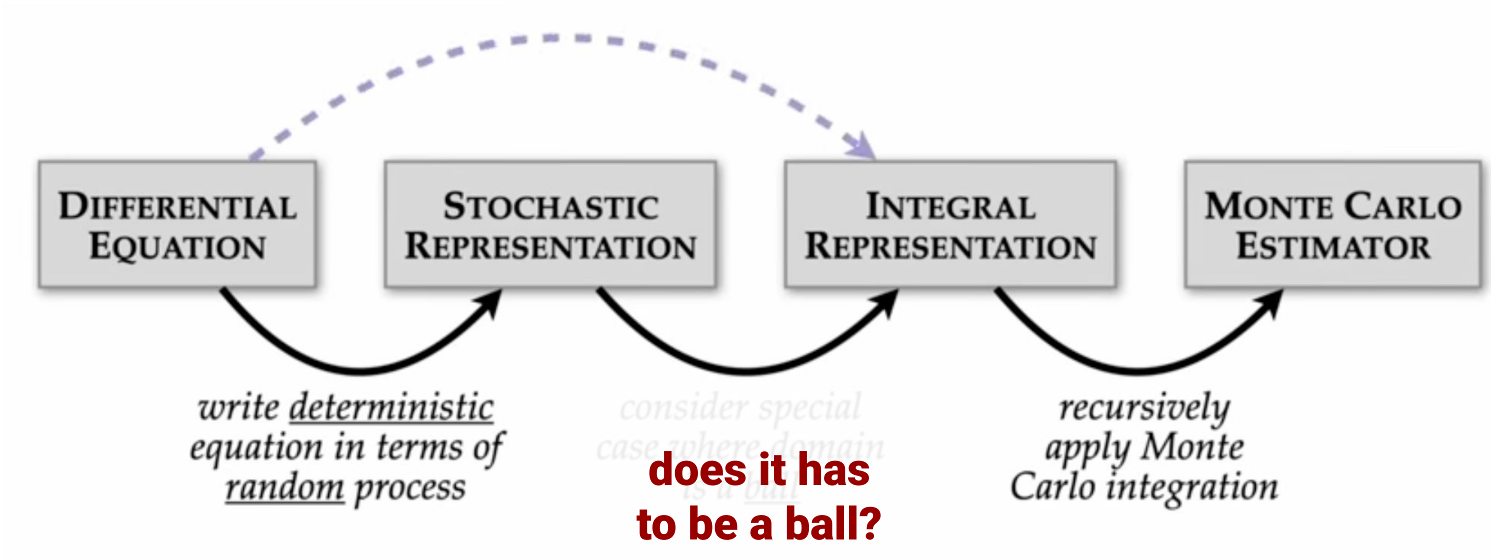
$$u(x) = \frac{1}{|\partial B(x)|} \int_{\partial B(x)} u(y)\, dy$$

volume of bounding sphere | ball around point x

Walk-on-Sphere - The Big Picture

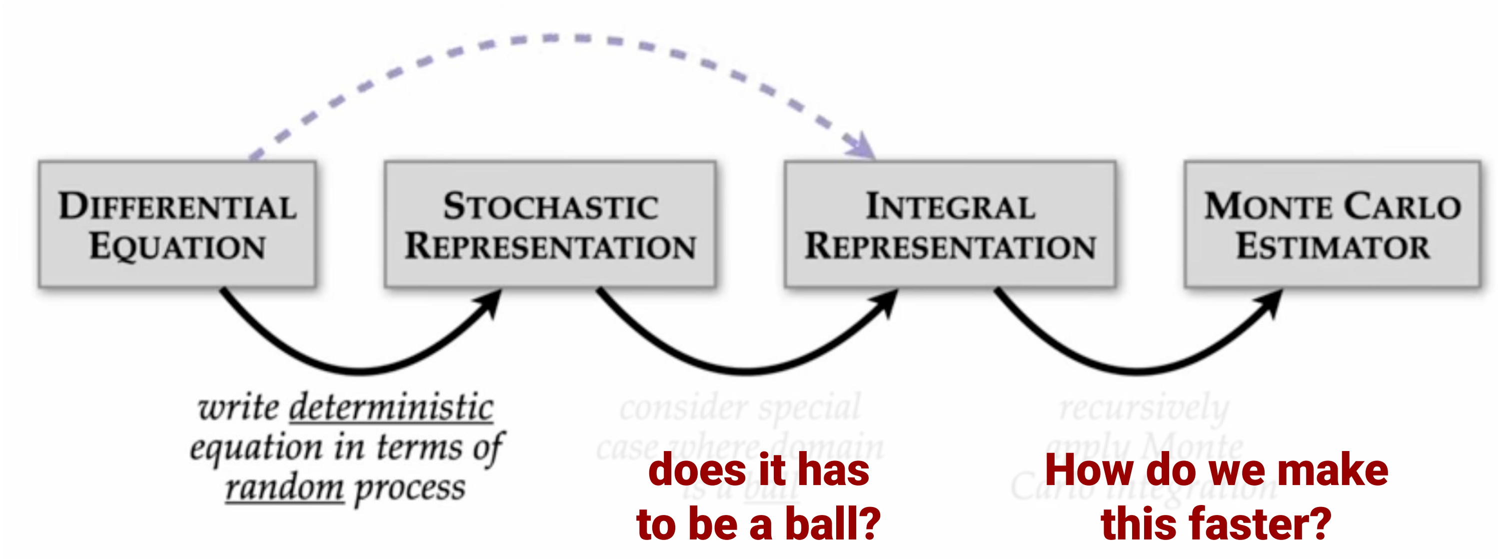


Walk-on-Sphere - The Big Picture



does it has to be a ball?

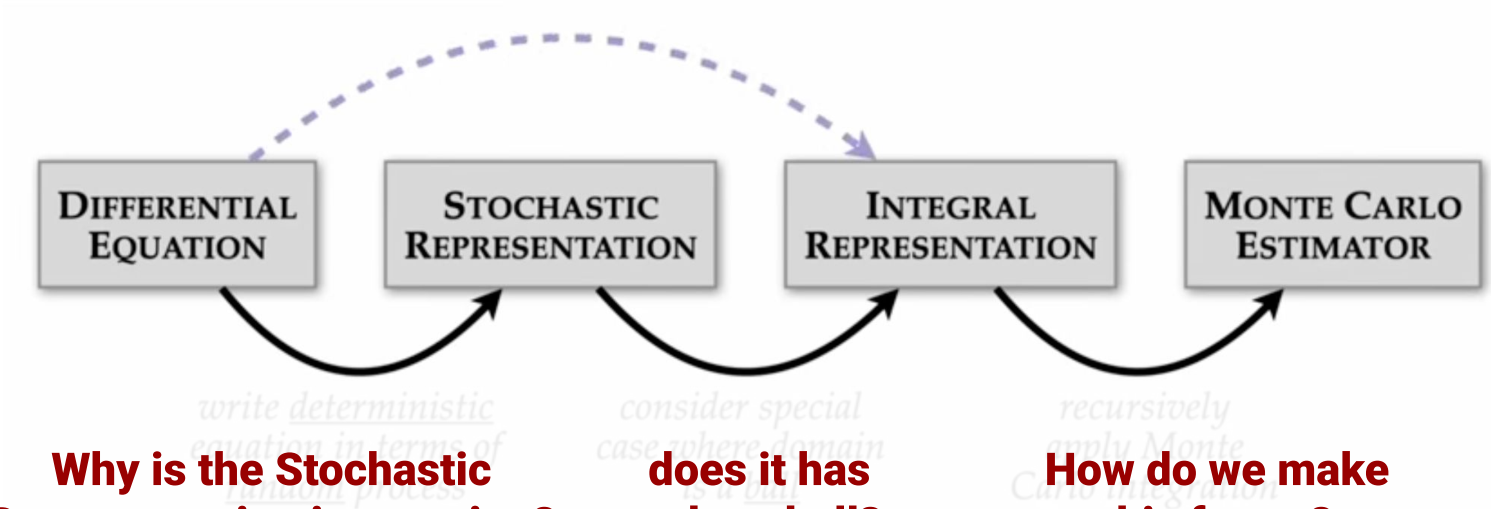
Walk-on-Sphere - The Big Picture



does it has to be a ball?

How do we make this faster?

Walk-on-Sphere - The Big Picture

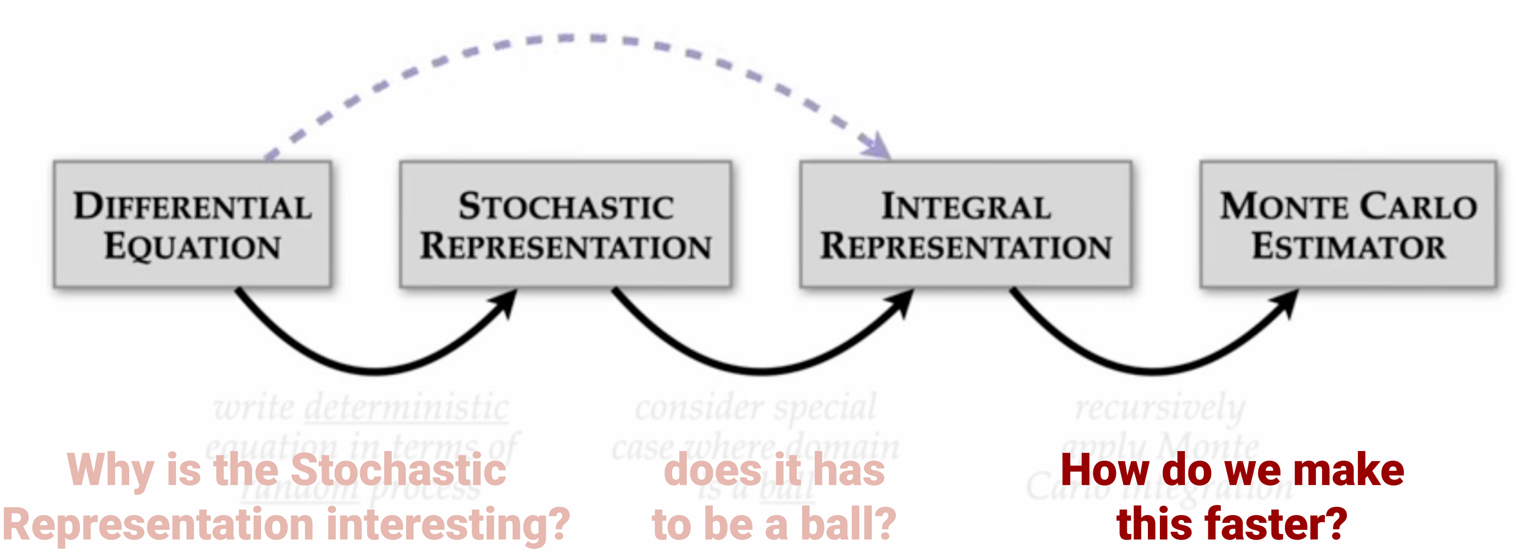


Why is the Stochastic Representation interesting?

does it has to be a ball?

How do we make this faster?

Walk-on-Sphere - The Big Picture

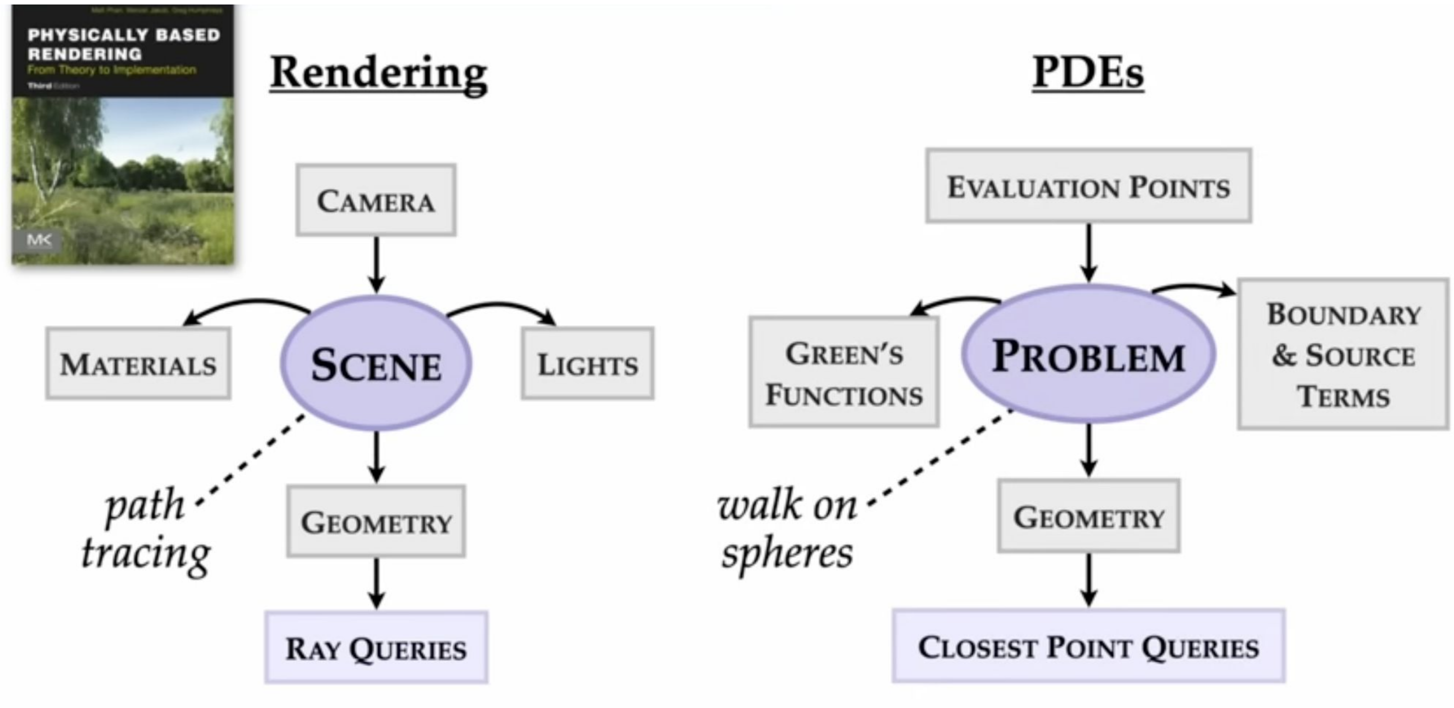


Why is the Stochastic Representation interesting?

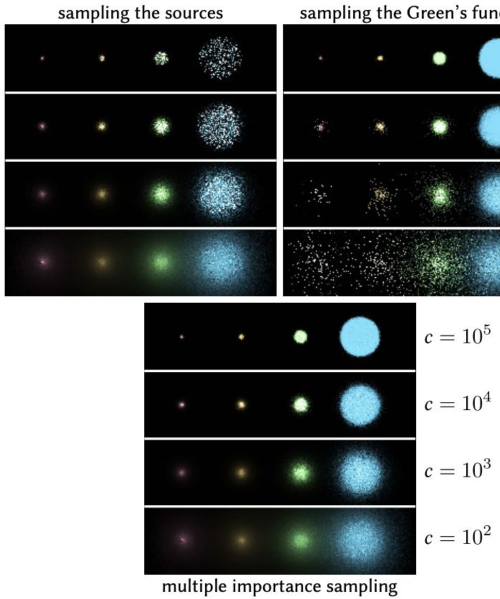
does it has to be a ball?

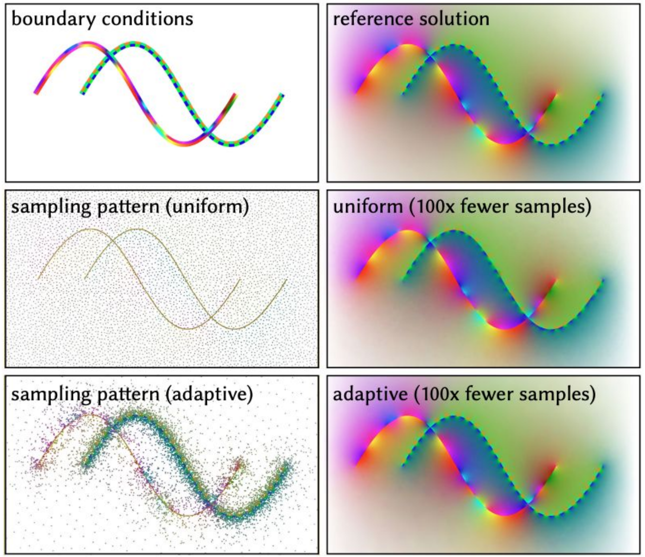
How do we make this faster?

Connection with Ray-Tracing



Ray-Tracing Methods in WOS

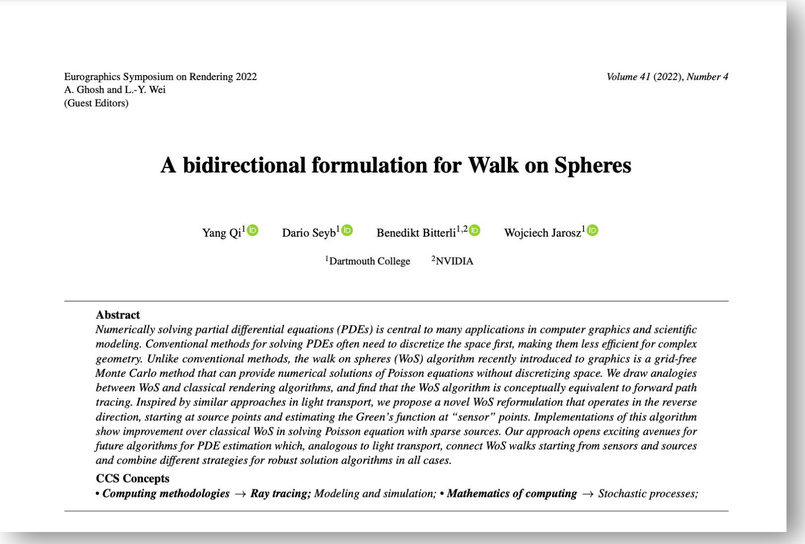




multiple importance sampling,
control variates

adaptive sampling
(sample near boundary)

Ray-Tracing Inspired Techniques on WoS


Eurographics Symposium on Rendering 2022
A. Ghosh and L.-Y. Wei
(Guest Editors)

Volume 41 (2022), Number 4

A bidirectional formulation for Walk on Spheres

Yang Qi[1] Dario Seyb[1] Benedikt Bitterli[1,2] Wojciech Jarosz[1]

[1]Dartmouth College [2]NVIDIA

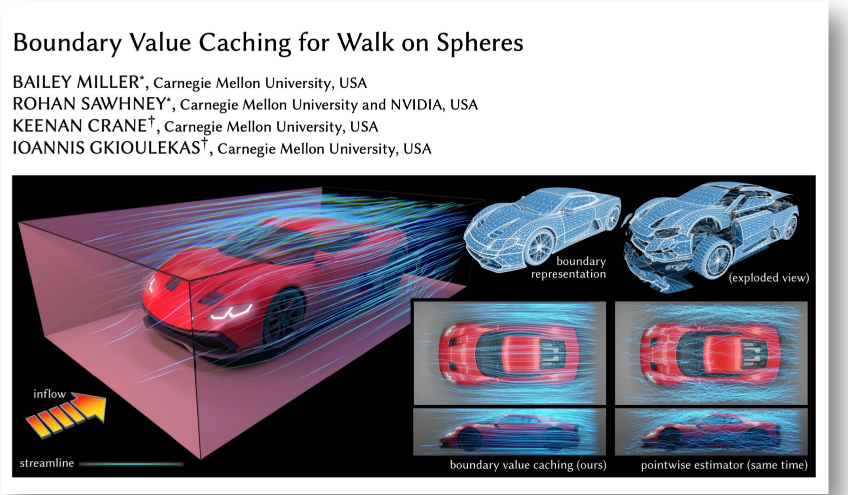
Abstract

Numerically solving partial differential equations (PDEs) is central to many applications in computer graphics and scientific modeling. Conventional methods for solving PDEs often need to discretize the space first, making them less efficient for complex geometry. Unlike conventional methods, the walk on spheres (WoS) algorithm recently introduced to graphics is a grid-free Monte Carlo method that can provide numerical solutions of Poisson equations without discretizing space. We draw analogies between WoS and classical rendering algorithms, and find that the WoS algorithm is conceptually equivalent to forward path tracing. Inspired by similar approaches in light transport, we propose a novel WoS reformulation that operates in the reverse direction, starting at source points and estimating the Green's function at "sensor" points. Implementations of this algorithm show improvement over classical WoS in solving Poisson equation with sparse sources. Our approach opens exciting avenues for future algorithms for PDE estimation which, analogous to light transport, connect WoS walks starting from sensors and sources and combine different strategies for robust solution algorithms in all cases.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;*** *Modeling and simulation;* • ***Mathematics of computing*** → *Stochastic processes;*

Bidirectional Ray-tracing


Boundary Value Caching for Walk on Spheres

BAILEY MILLER*, Carnegie Mellon University, USA
ROHAN SAWHNEY*, Carnegie Mellon University and NVIDIA, USA
KEENAN CRANE†, Carnegie Mellon University, USA
IOANNIS GKIOULEKAS†, Carnegie Mellon University, USA

Photon Mapping

4. Experiments

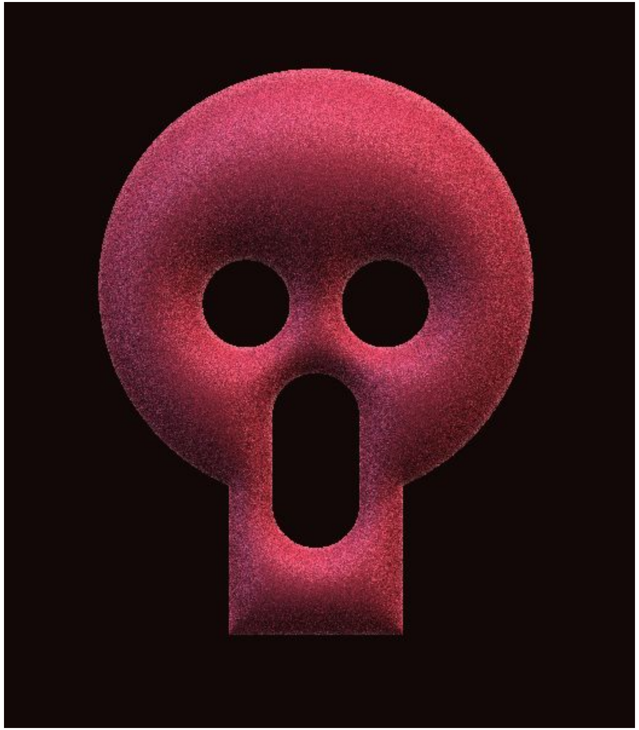
Curve Inflation - My Spooky Implementation

$$\Delta u = -4 \text{ on } \Omega$$

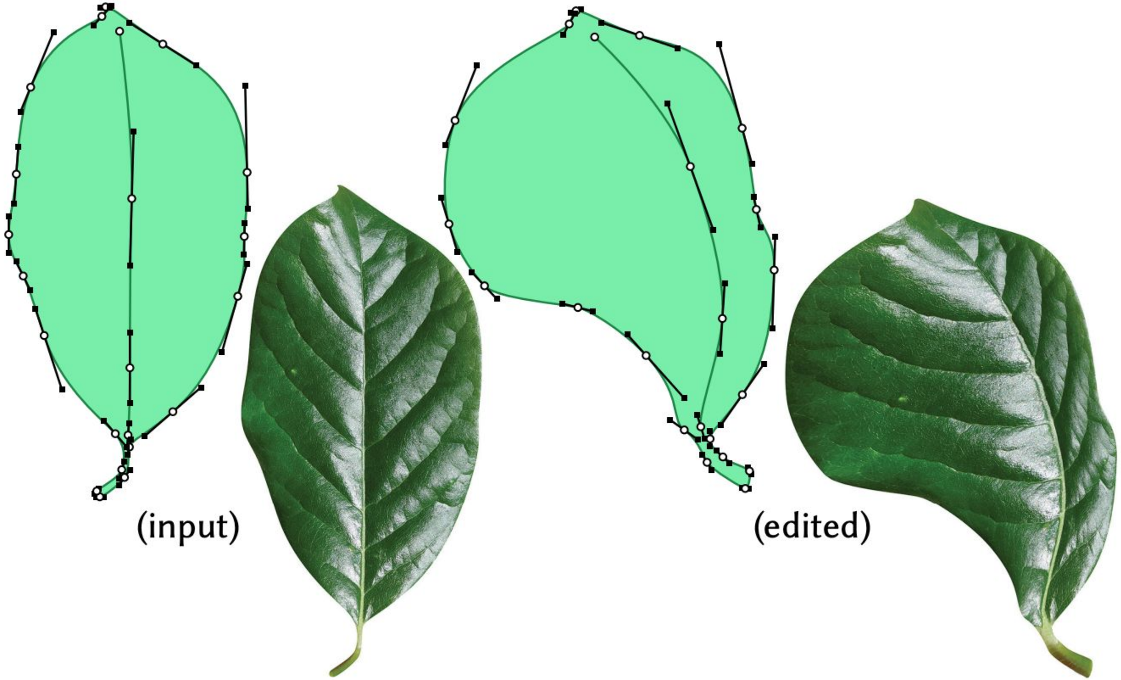
$$u = 0 \text{ on } \partial\Omega$$

$$z = \sqrt{u}$$ ← inflation

```
for (int k = 0; k < MAX_WALKS; k++) {
    if (R < EPS) break;     // terminate at boundary
    u += PI*R*R*G(R);       // evaluate Green function
    x += R * rand_circ();  // random walk
    R = main_sdf(x);        // distance query
}
```

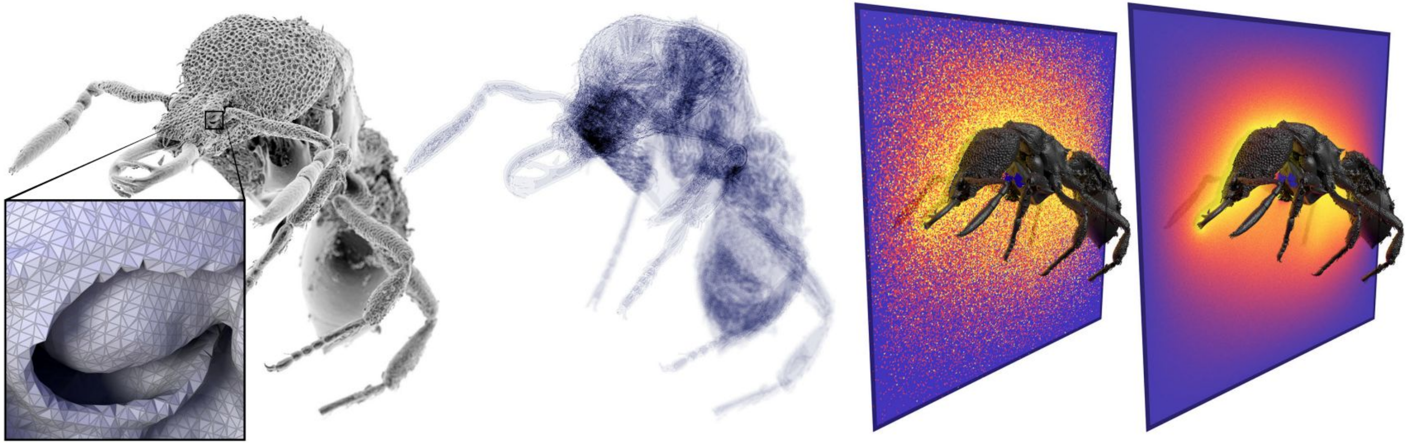


Real-time Deformation



FEM vs. WoS on Poisson

method	linear FEM	Monte Carlo
#triangles	2M	**10M**
#samples	**47k nodes**	23k pixels
precompute	14 hours	**0.4 seconds**
solve	**13 seconds**	57 seconds



5. Main Takeaways

There is no Free Lunch! Just discounted Lunch

Pros

- **agnostic to representation**
- parallelizable
- **easy to implement**
- fast convergence
- no precompute
- easy to realize on most parabolic and elliptic PDEs
- **unbiased method**
- robust to noise, numerically stable
- easy to compute div, grad, curl

Cons

- **hard to realize on hyperbolic PDEs**
  (eg. Wave Equation, Schrödinger Equation)
- requires bounded domain with Dirichlet (non-reflecting) boundary
- **requires analytic form of Green Functions**
- yet to handle spatially-varying conditions
- yet to handle time-dependent equation
- **only consider volumetric domain**

Quiz Time! :D

When performing walk-on-sphere on **bounded domain**. Is there a domain geometry that makes the algorithm never terminate in probability?

$$\lim_{n\to\infty} \mathbb{P}(\text{reaching boundary after } n \text{ walks}) \neq 1$$

(a) **True**
(b) **False**
(c) **Undecidable**

Which figure describe the **WRONG** walk-on-sphere algorithm?
(there is only one walker)

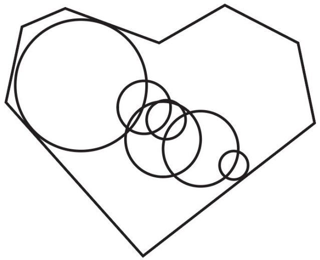


figure A

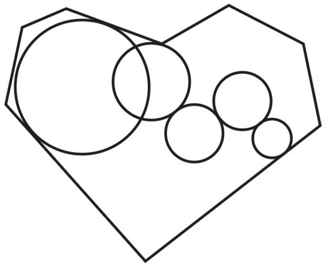


figure B

(a) **figure A**
(b) **figure B**
(c) **both wrong**
(d) **both correct**