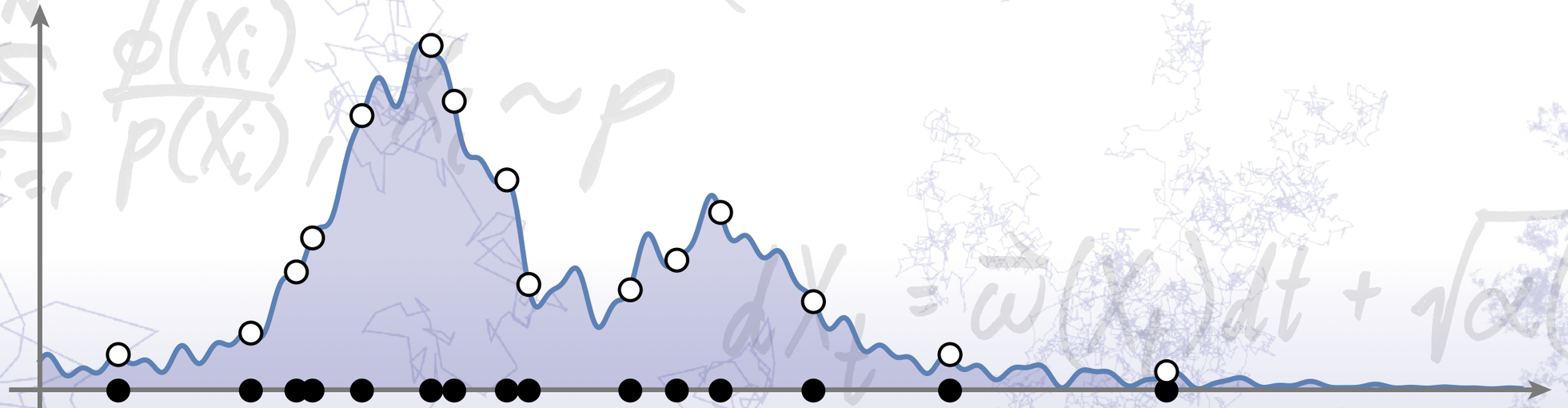
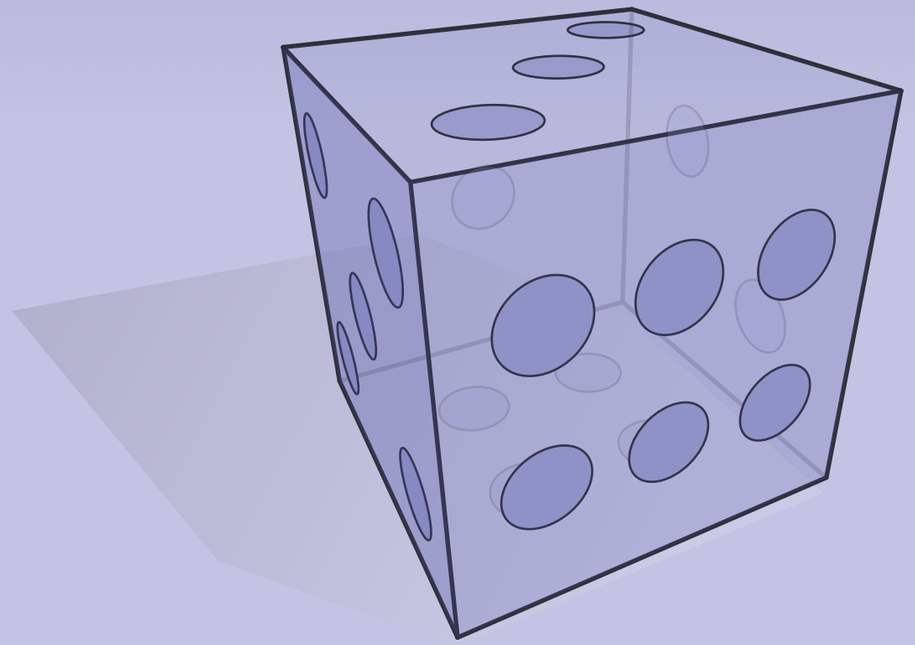
Monte Carlo Methods and Applications



Carnegie Mellon University | 21-387 / 15-327 / 15-627 / 15-860 | Fall 2023

Lecture 22

Walk on Spheres



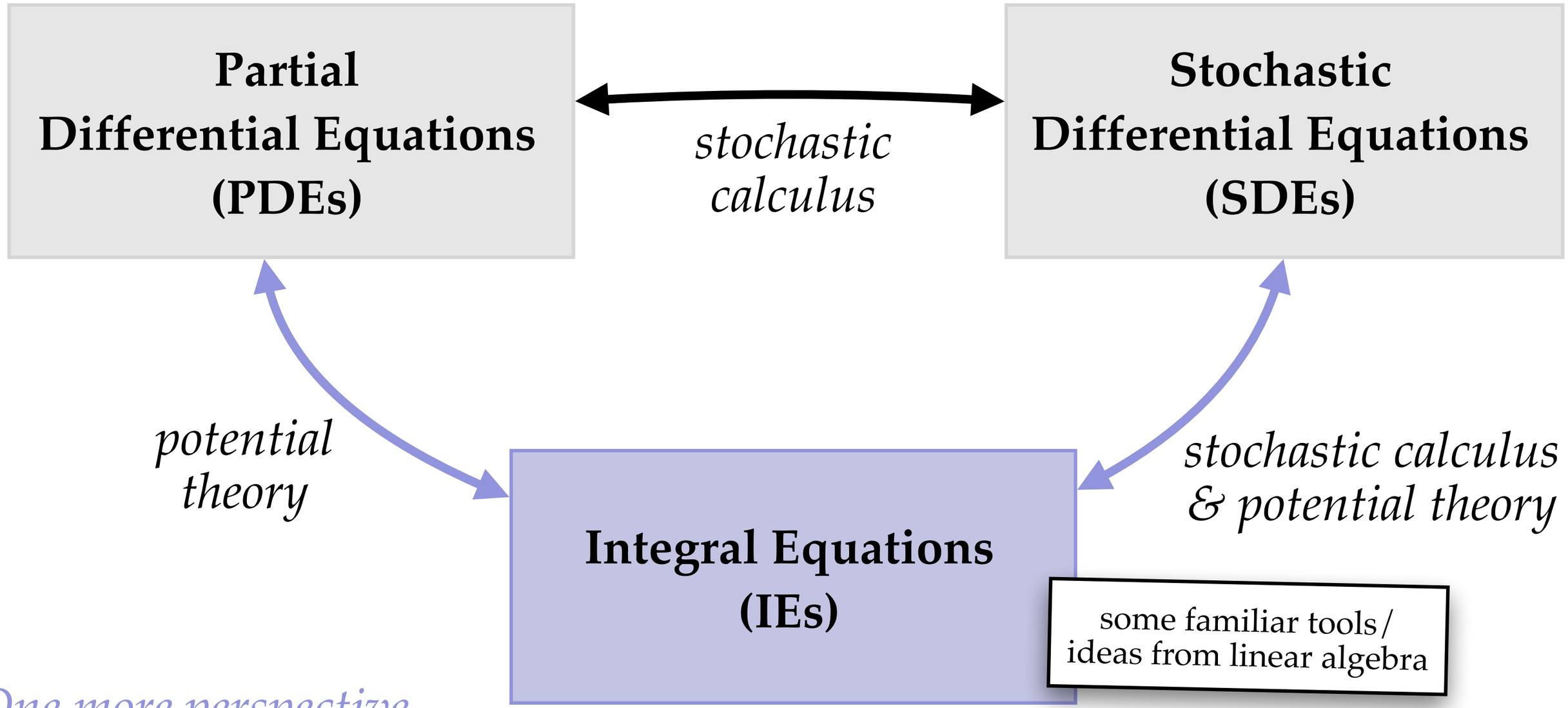
Monte Carlo Methods and Applications

Carnegie Mellon University | 21-387 | 15-327/627 | 15-860 | Fall 2023

Walk on Spheres—Overview

- Application goal has been to solve **sampling problems**, using *PDEs as a tool*
- Today, our application goal is to **solve tough PDEs**, using *sampling as a tool*
  - Why do we want to solve PDEs?
  - What are the computational challenges?
  - *"Walk on Spheres (WoS)"* as interesting new tool **(research @ CMU!)**
- This lecture: basic Walk on Spheres Algorithm (Laplace)
- Next lecture: "Walk on *X*" algorithms for broader PDEs

Overview—Mathematical Tools



One more perspective…

Overview—Numerical Tools

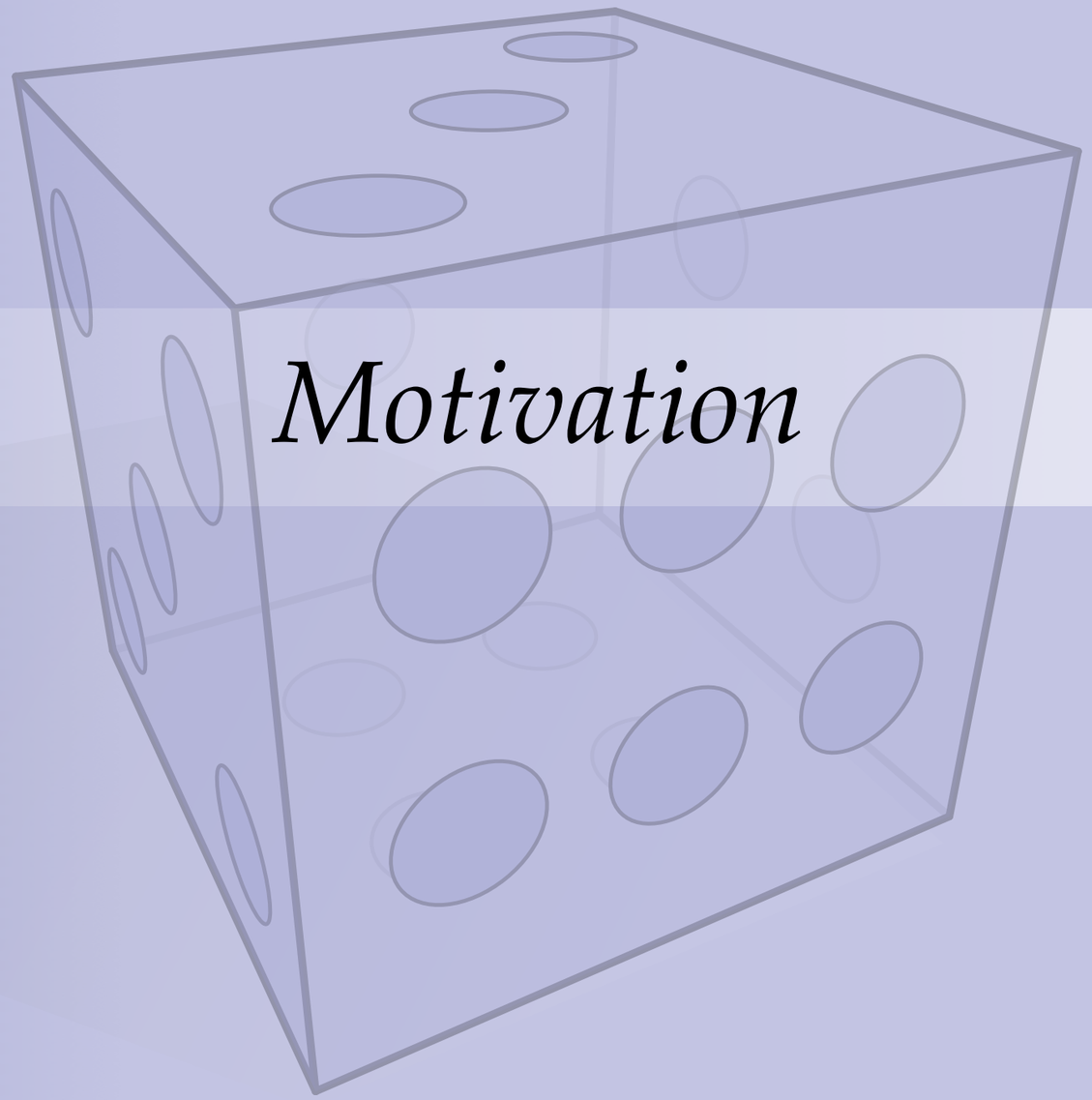
Deterministic techniques

- numerical quadrature
- finite differences
- finite elements
- finite volumes
- spectral methods
- …

Stochastic techniques

- Monte Carlo integration
- Euler-Maruyama
- Metropolis-Hastings
- MALA
- Langevin Monte Carlo
- Walk on Spheres

We're going to add one more trick to the bag…



Motivation

Partial Differential Equations are Everywhere

Schrödinger equation
quantum mechanics
computational chemistry
material science

wave equation
acoustics
seismology

convection-diffusion
heat flow
thermal engineering
mathematical finance

Maxwell's equations
electromagnetism
electric power / motors
electronics / optics

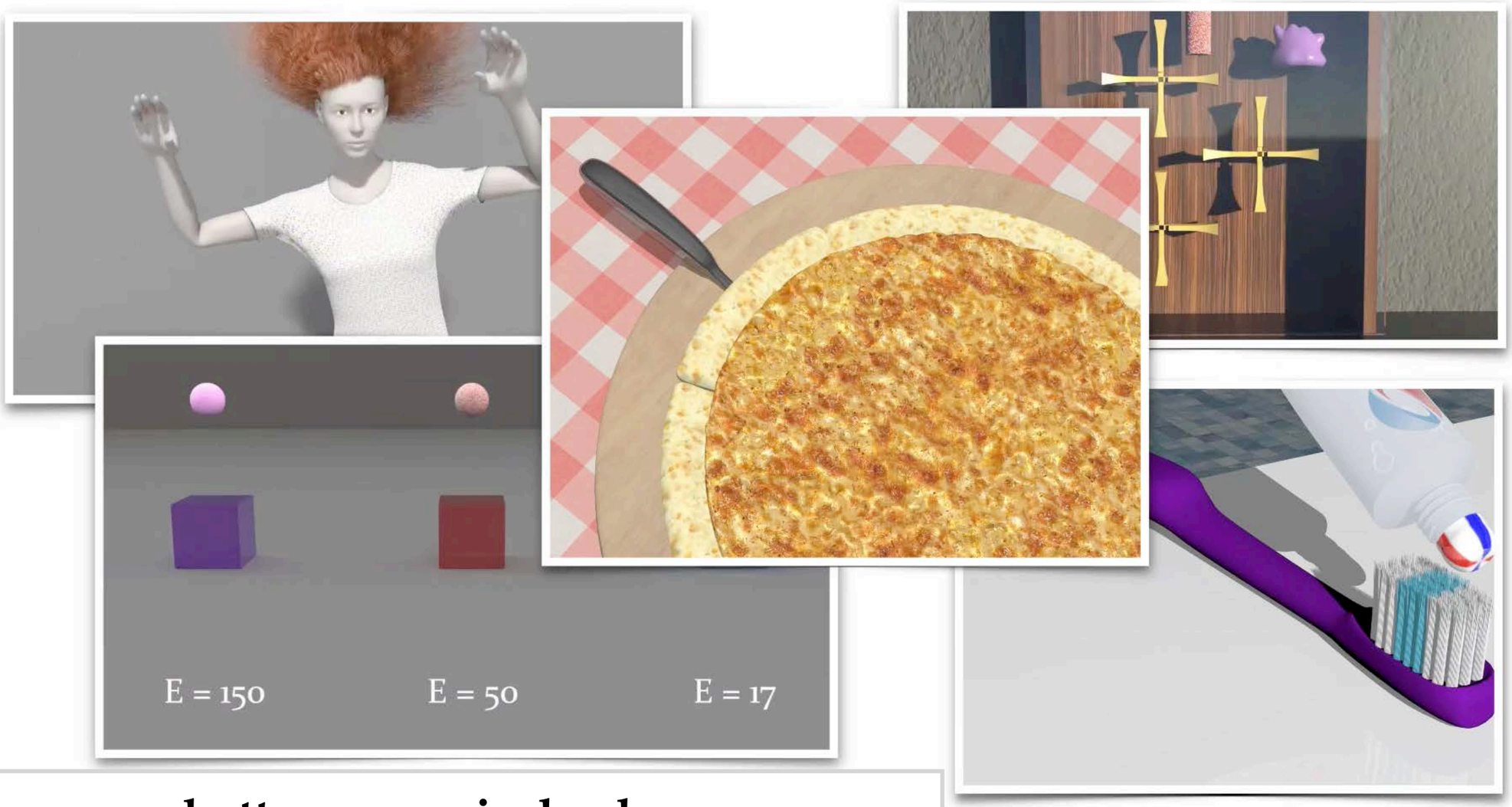
Poisson equation
electrostatics, gravitation
signal processing

reaction-diffusion
chemical concentrations
cell biology / morphogenesis
population dynamics

Navier-Stokes
fluid flow
aerospace / automotive
climate / weather

Einstein equations
general relativity
cosmology / astrophysics
gravitational lensing

So, Numerical PDE Solvers are Everywhere



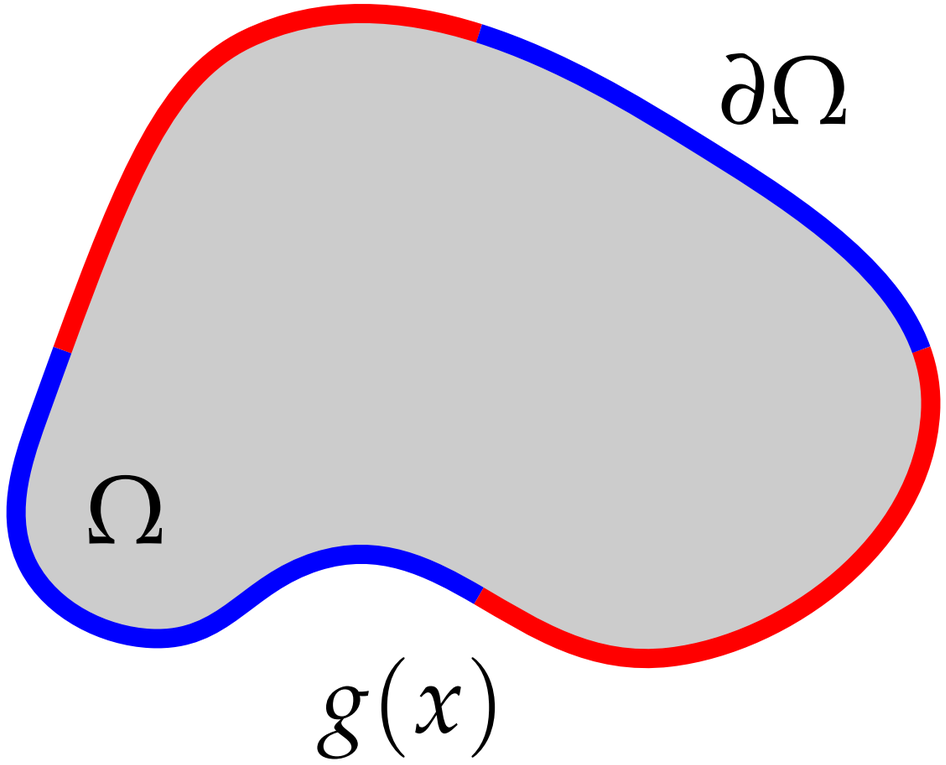
better numerical solvers
⇒ better ability to predict reality

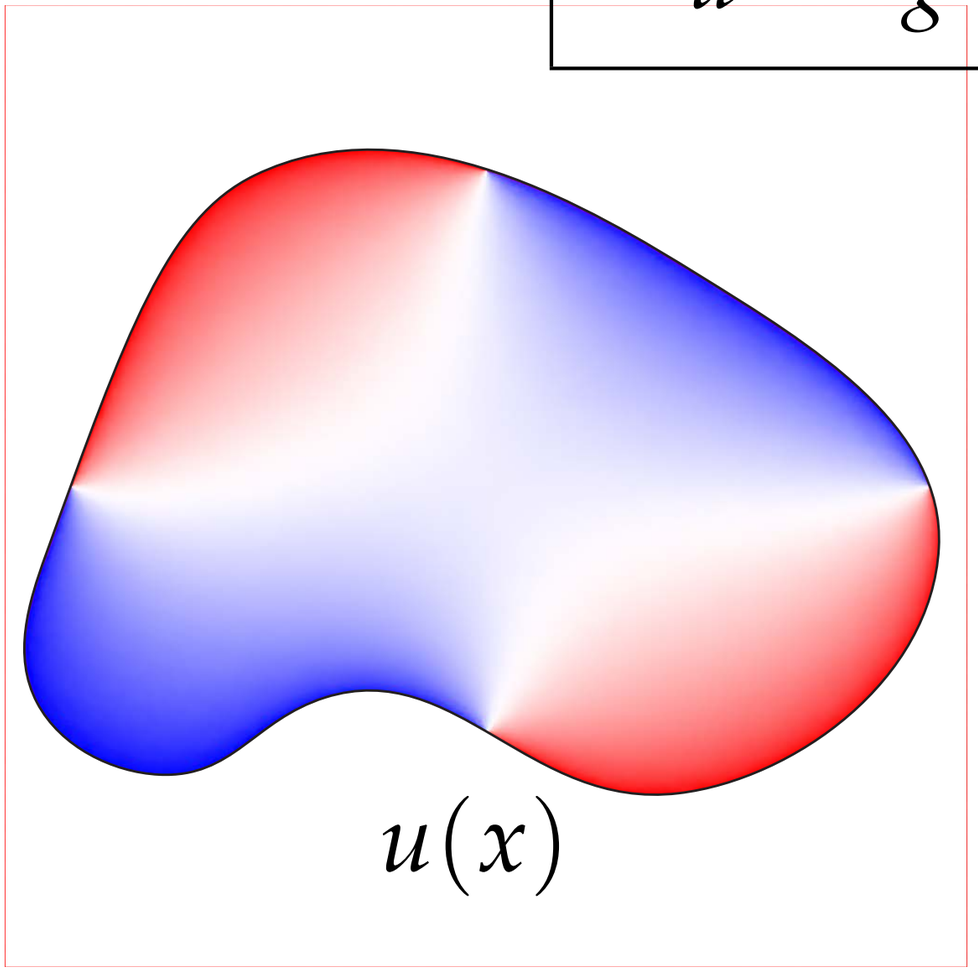
Ok, but let’s start simple. :-)

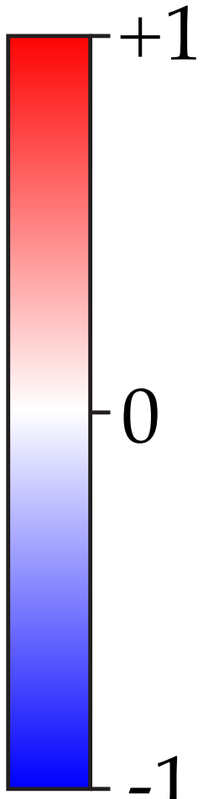
GIVEN: values on boundary $\partial\Omega$ of a region Ω

FIND: smooth interpolation into interior

$$\begin{aligned} \Delta u &= 0 \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$

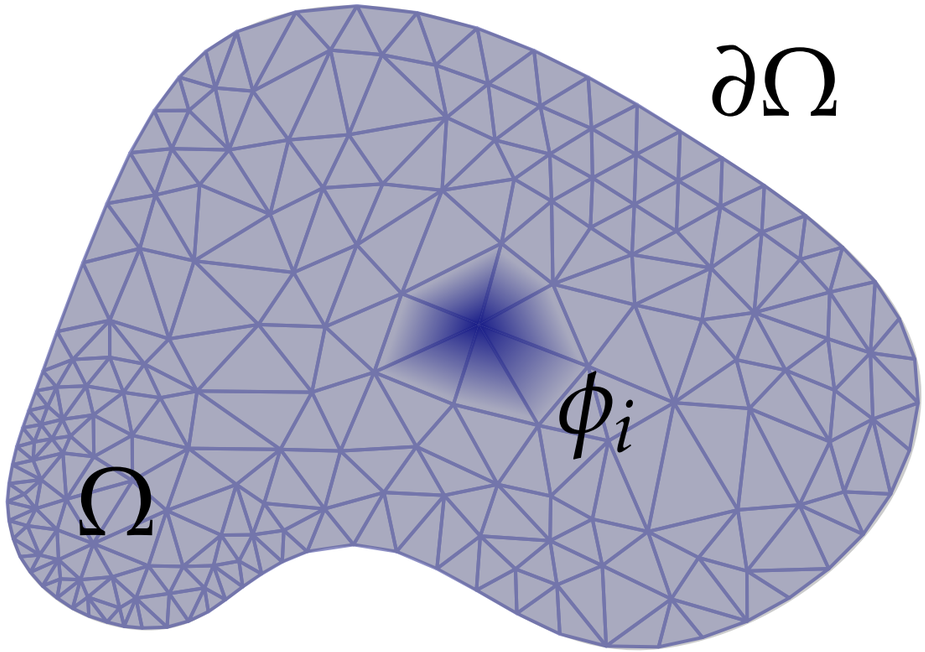




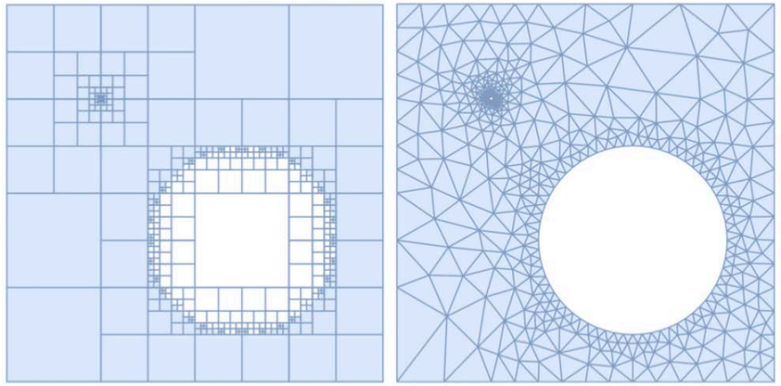


Interpolation Problem—Traditional Approach

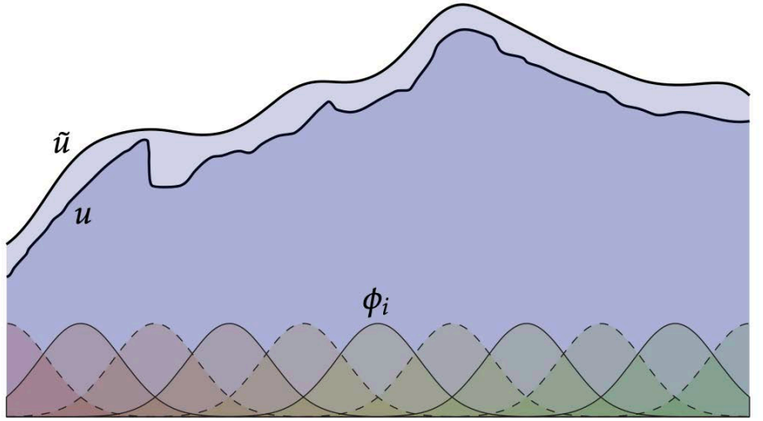
Traditional methods* compute a finite-dimensional approximation:



$$\langle \Delta u, \phi_i \rangle = 0, \ \forall i$$



error in approximation of <u>geometry</u>

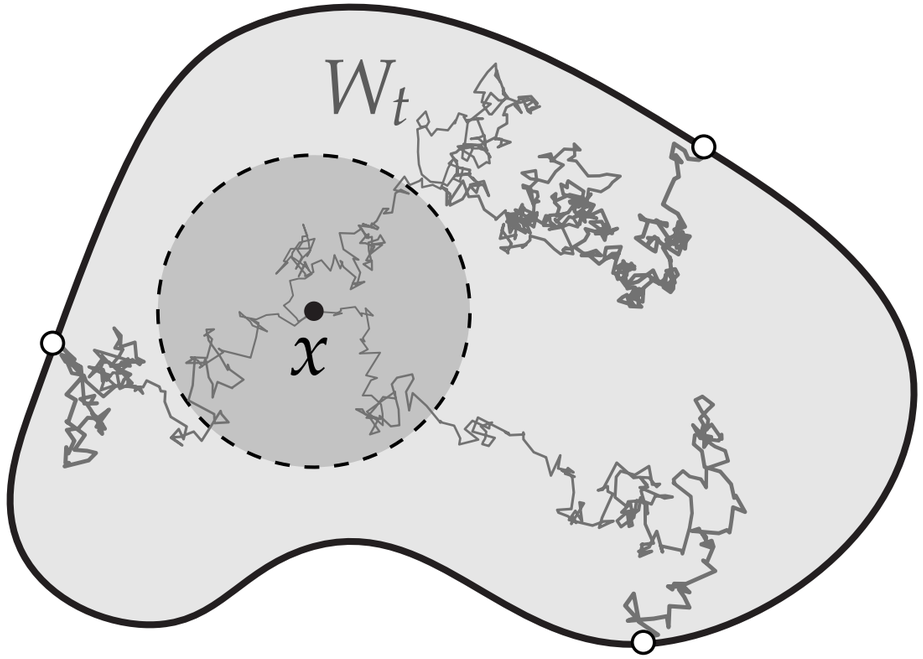


error in approximation of <u>functions</u>

*e.g., finite differences, FEM, BEM, "meshless" FEM, …

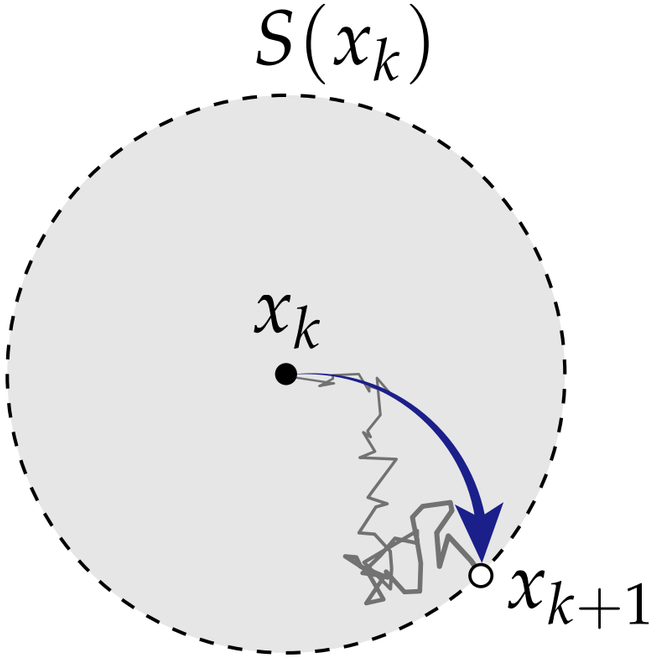
Interpolation Problem—Monte Carlo Approach

Can avoid finite-dimensional approximation **completely**, via Monte Carlo:



KAKUTANI'S PRINCIPLE

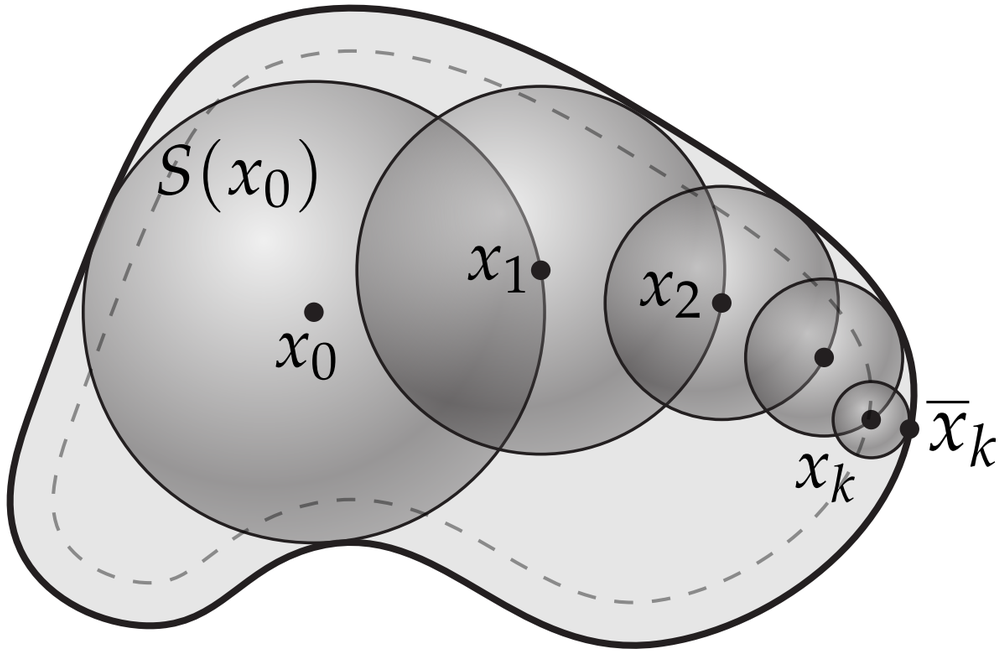
$$u(x) = \mathbb{E}[g(W_T)]$$



idea: to simulate a step of a (Brownian) random walk, sample any "empty" sphere $S(x_k) \subset \Omega$ around x_k uniformly at random.

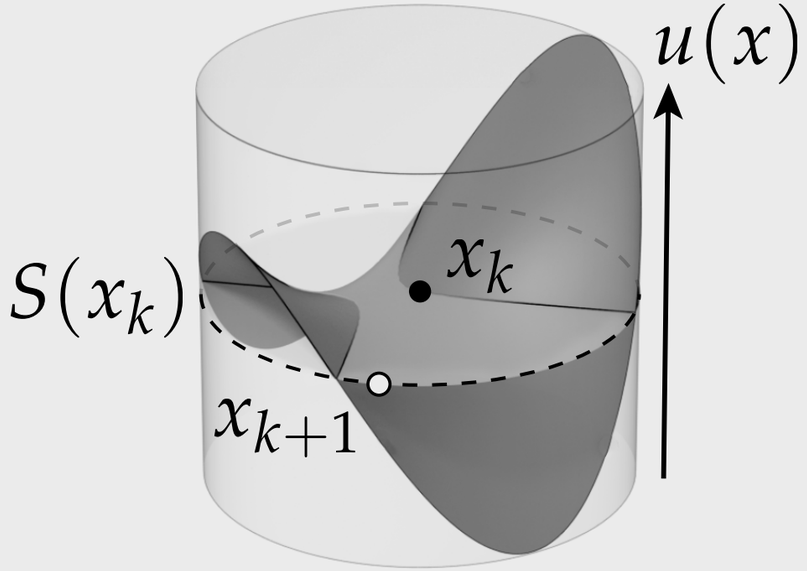
Interpolation Problem—Monte Carlo Approach

Can avoid finite-dimensional approximation **completely**, via Monte Carlo:



WALK ON SPHERES
[MULLER 1956]

MEAN VALUE PROPERTY



$$u(x_k) = \frac{1}{|S(x_k)|} \int_{S_k} u(y)\, dy$$

apply 1-point Monte Carlo estimate

Why would we want to do it this way?

"I hate meshes.
I cannot believe how hard this is.
Geometry is hard."

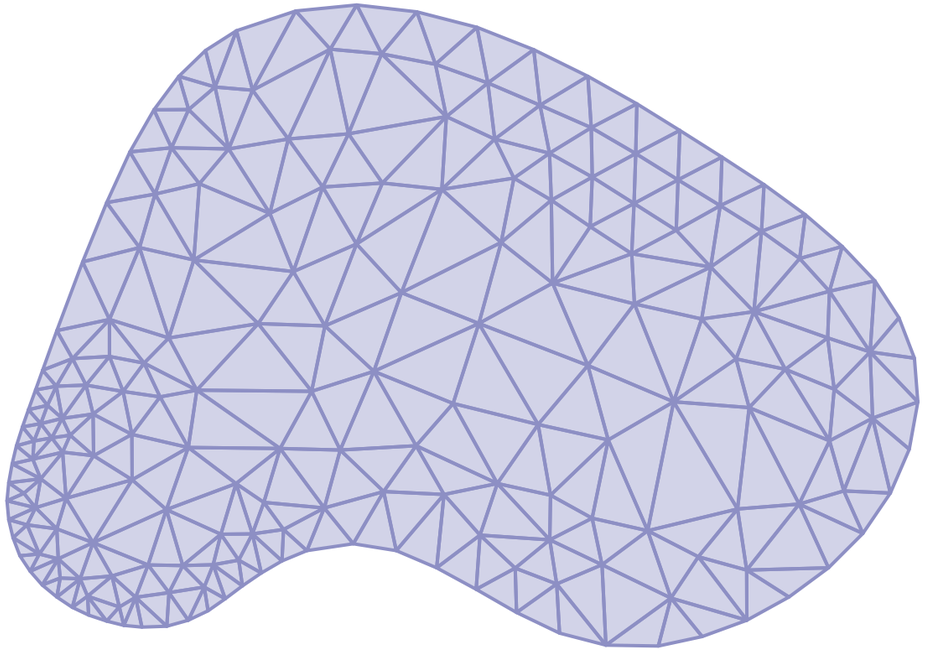
—David Baraff

Senior Research Scientist

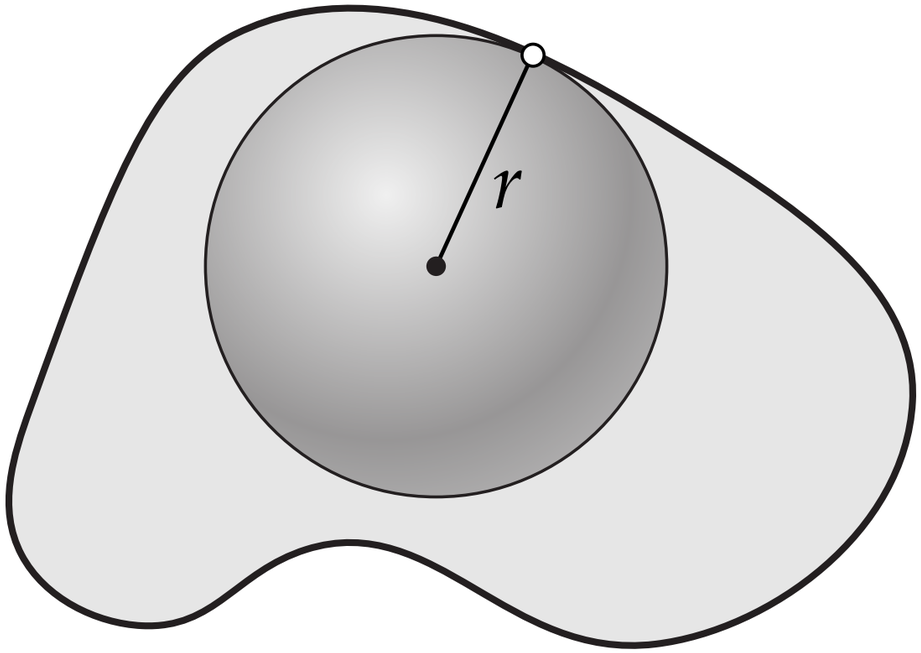
Pixar Animation Studios

Meshing is hard… finding closest point is easy!

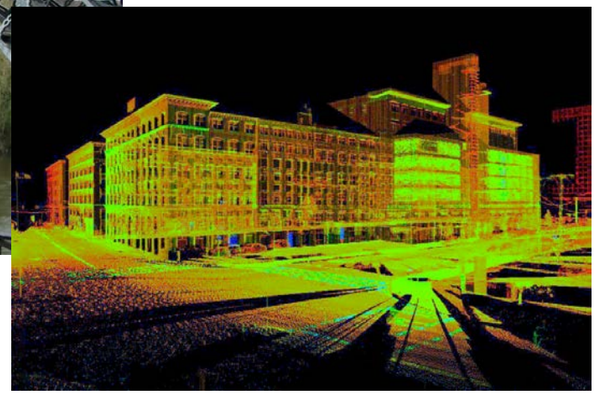
HARD



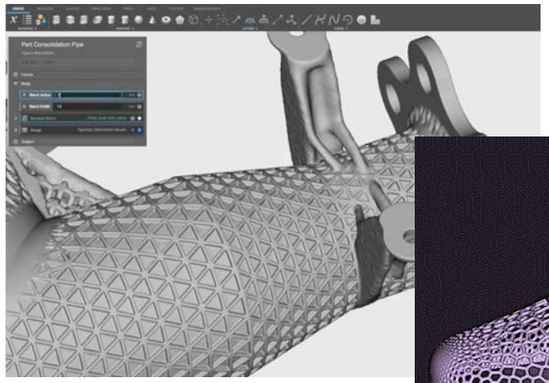
EASY

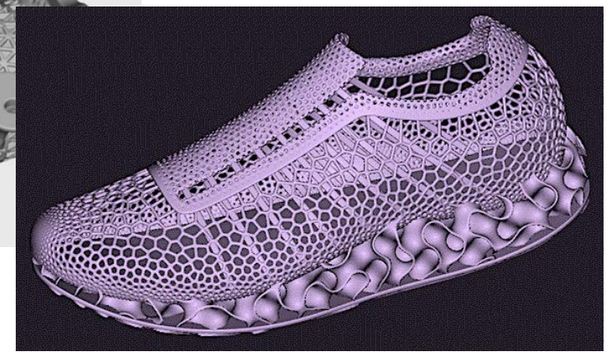


Need to analyze geometry of increasing complexity

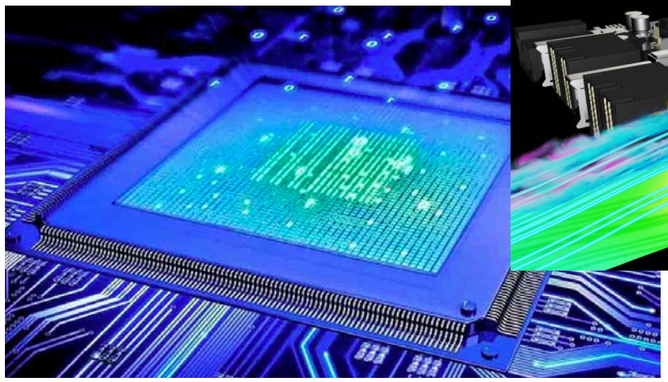


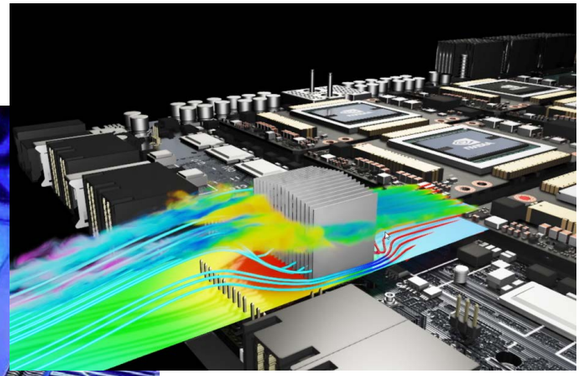
laser scanning / LiDAR



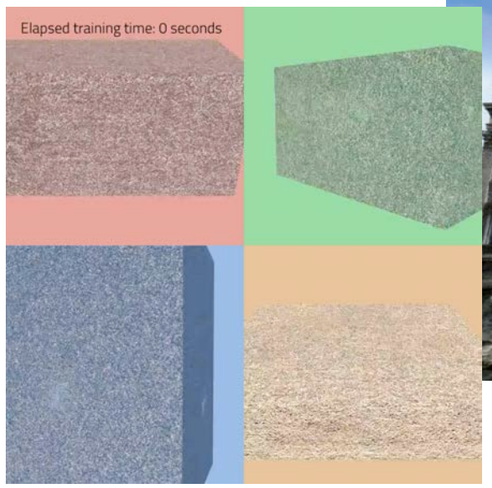


advanced manufacturing





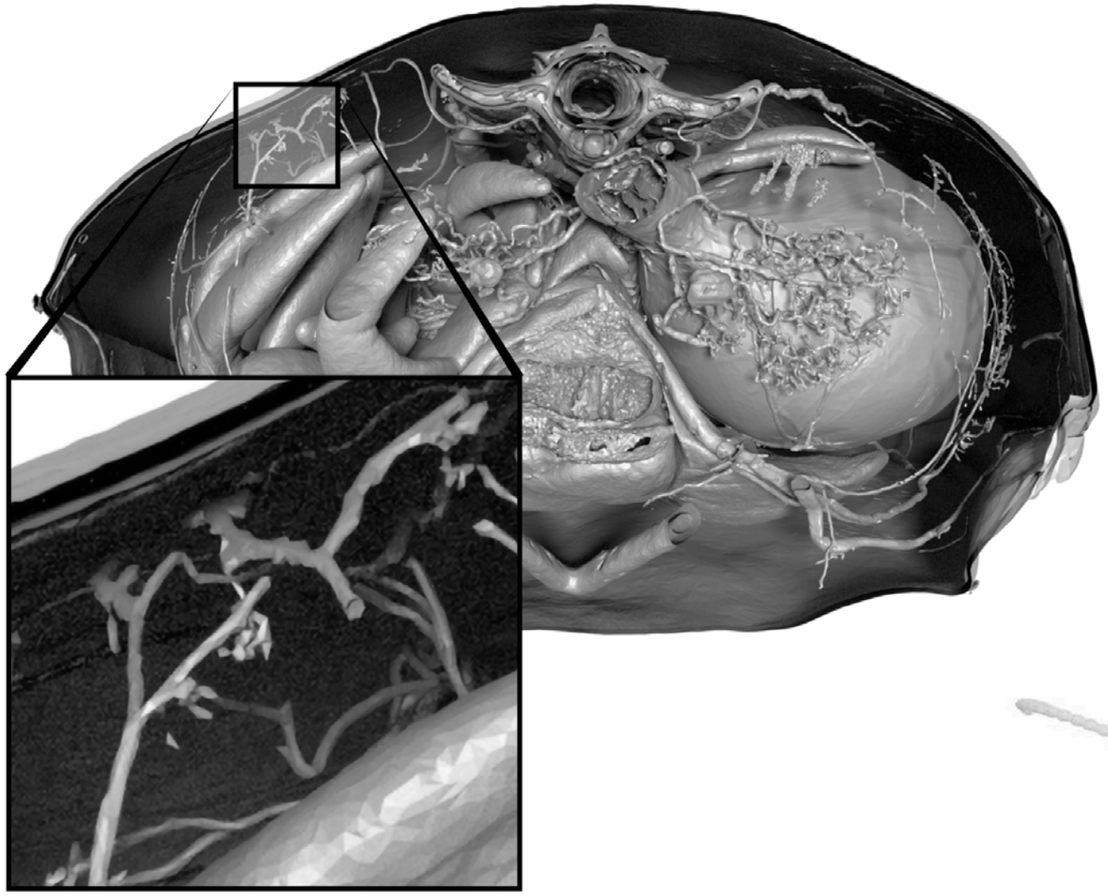
integrated circuit design

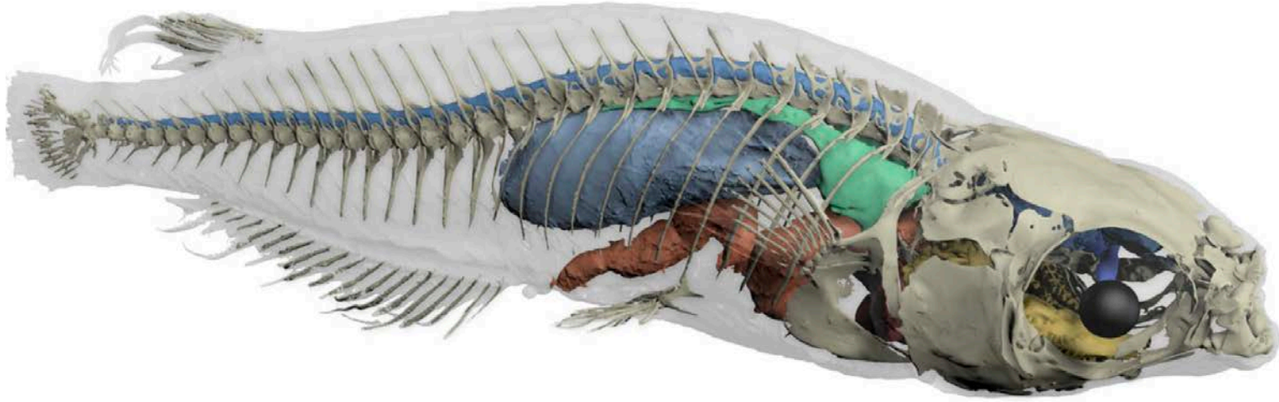


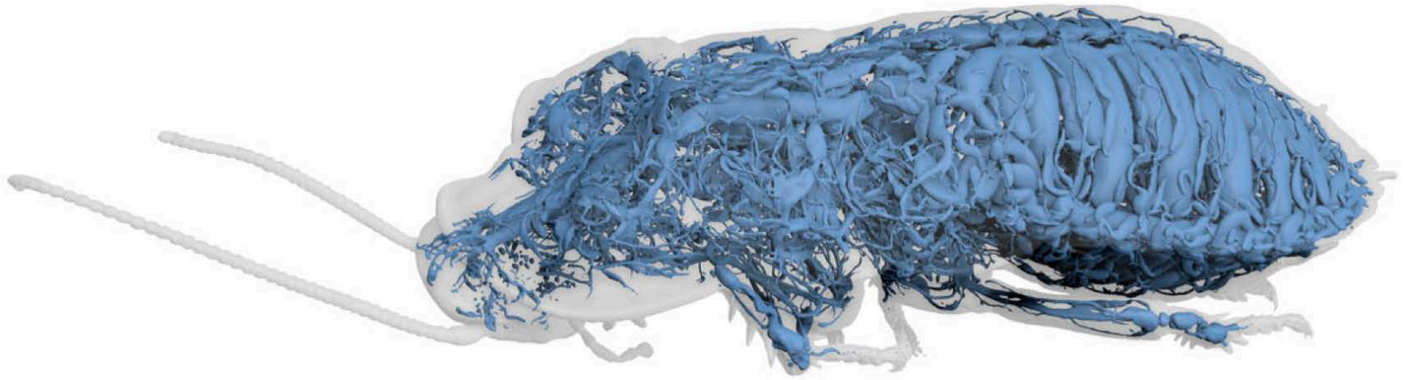
ML/generative modeling

Geometry found in nature is extremely complex…

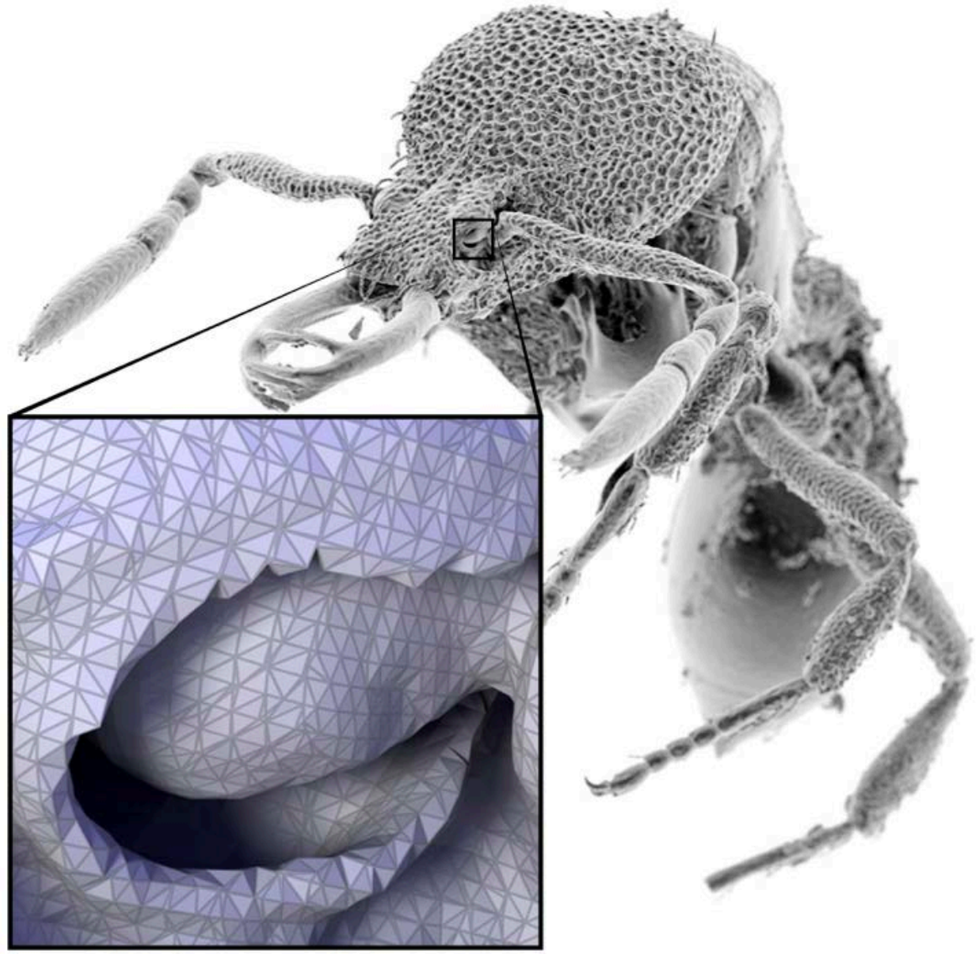
Example: high-resolution microCT







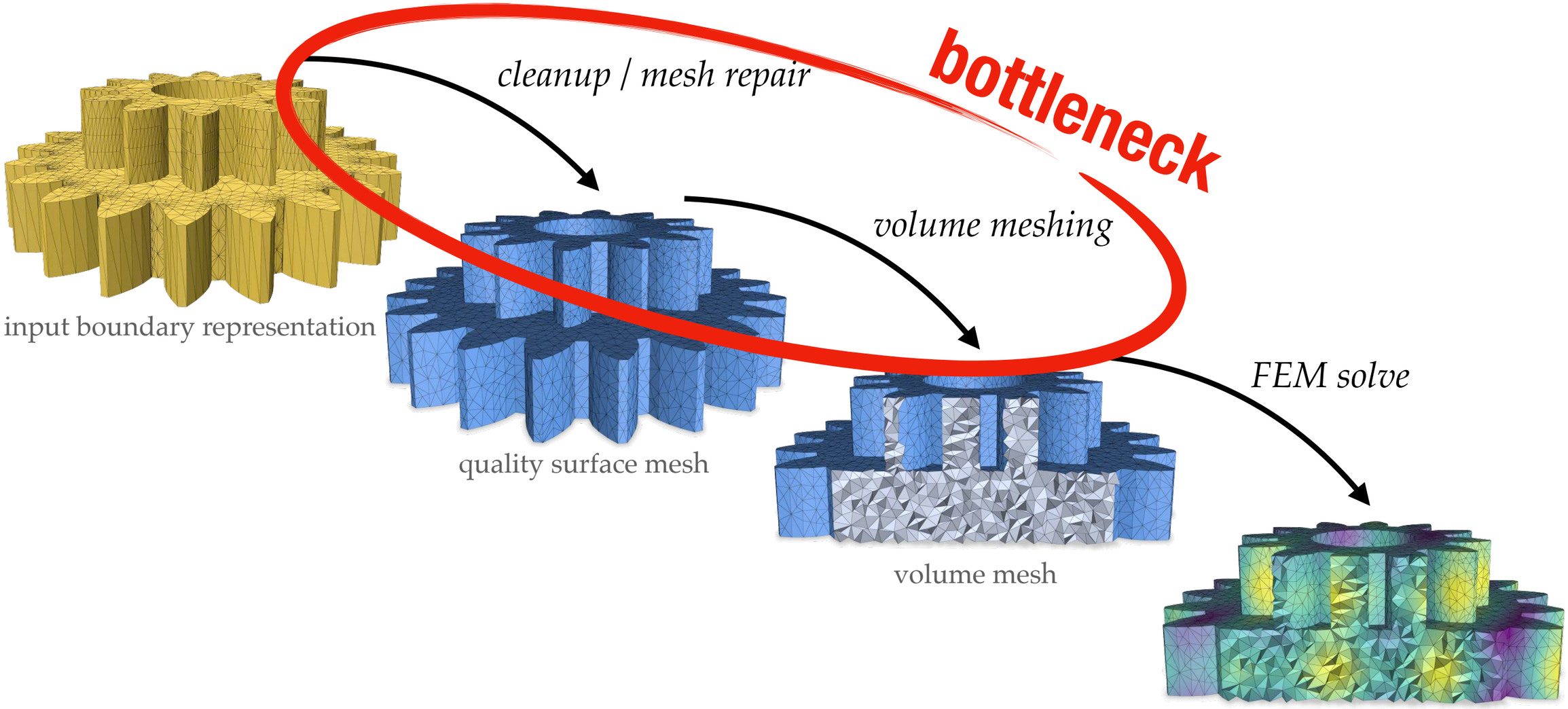
…and can take an extremely long time to mesh!



14 hrs / 30 GB RAM

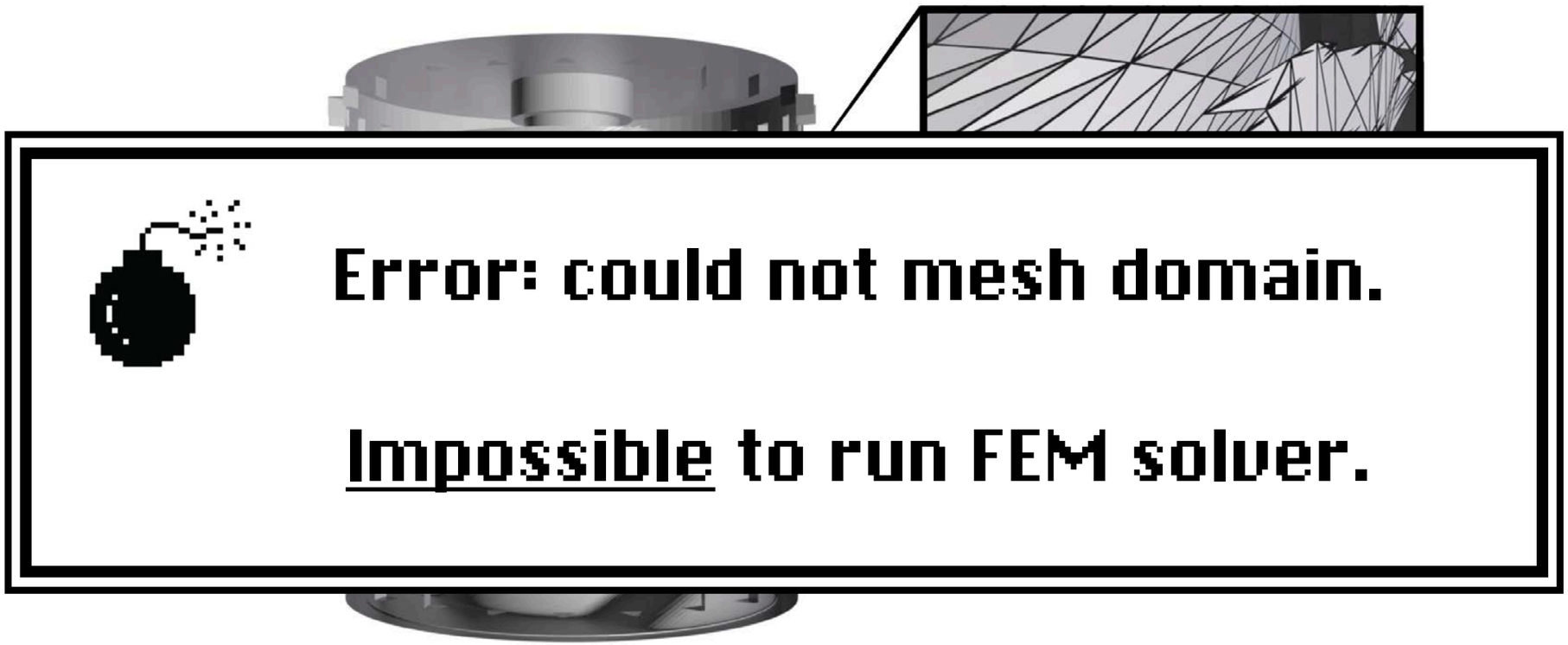
to generate FEM mesh

If meshing is slow, who cares if solver is fast?



finite element method (FEM) pipeline

Most geometry is not suitable for simulation

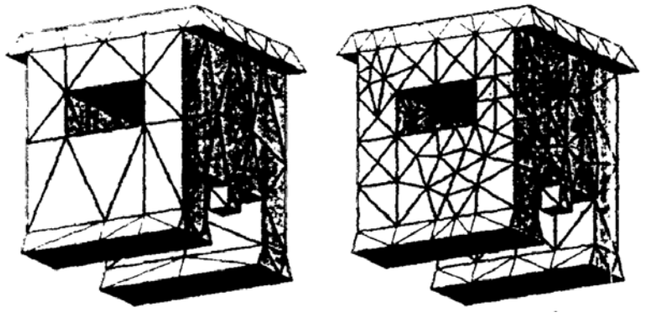


"Hell is other people's meshes."

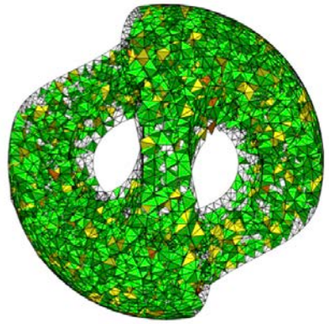
—Jean-Paul Sartre

Robust meshing is still hard, even after 20+ years

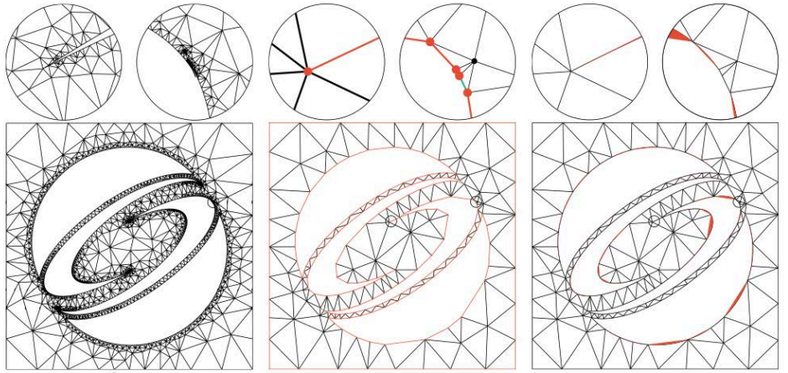
Not likely to ever be completely "solved":



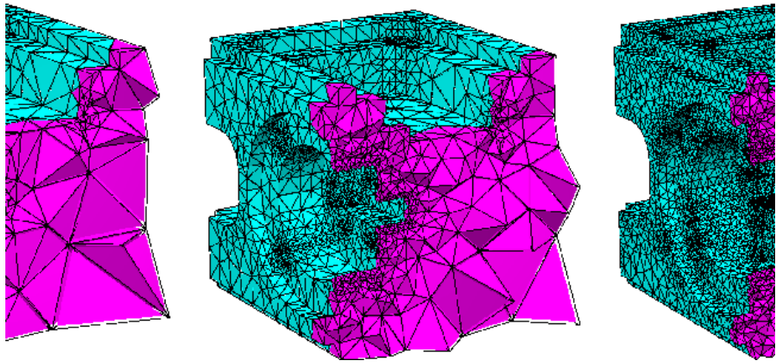
Tetrahedral Mesh Generation by Delaunay Refinement
[Shewchuk 1998]



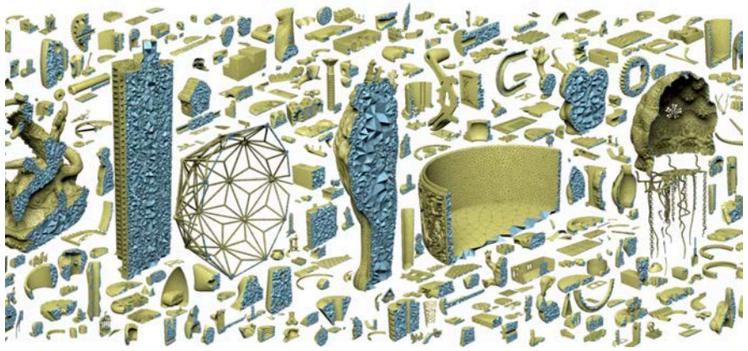
Robust Tetrahedral Meshing of Triangle Soups
[Spillman et al. 2006]



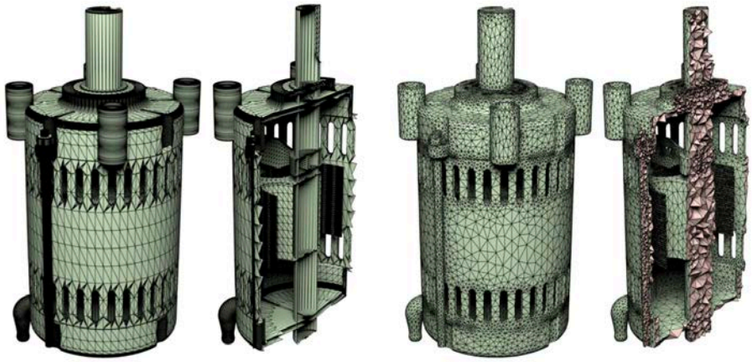
TriWild: Robust Triangulation With Curve Constraints
[Hu et al. 2019]



A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator
[Si 2006]



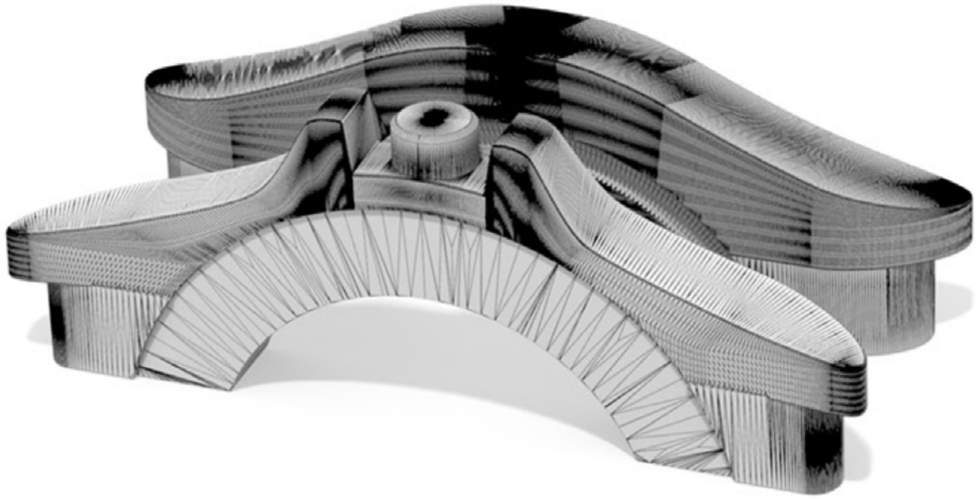
Tetrahedral Meshing in the Wild
[Hu et al. 2018]



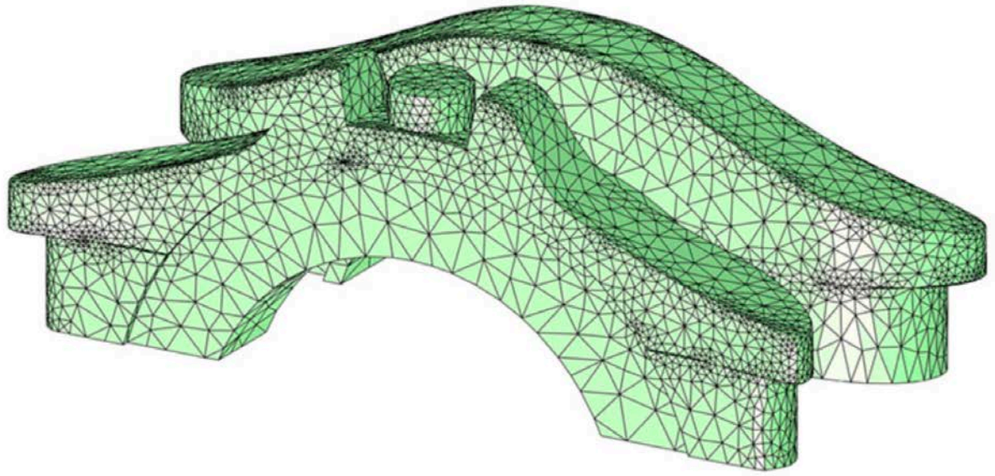
Fast Tetrahedral Meshing in the Wild
[Hu et al. 2020]

Robust meshing can be wildly unpredictable

Even very simple geometry can take hours to mesh:



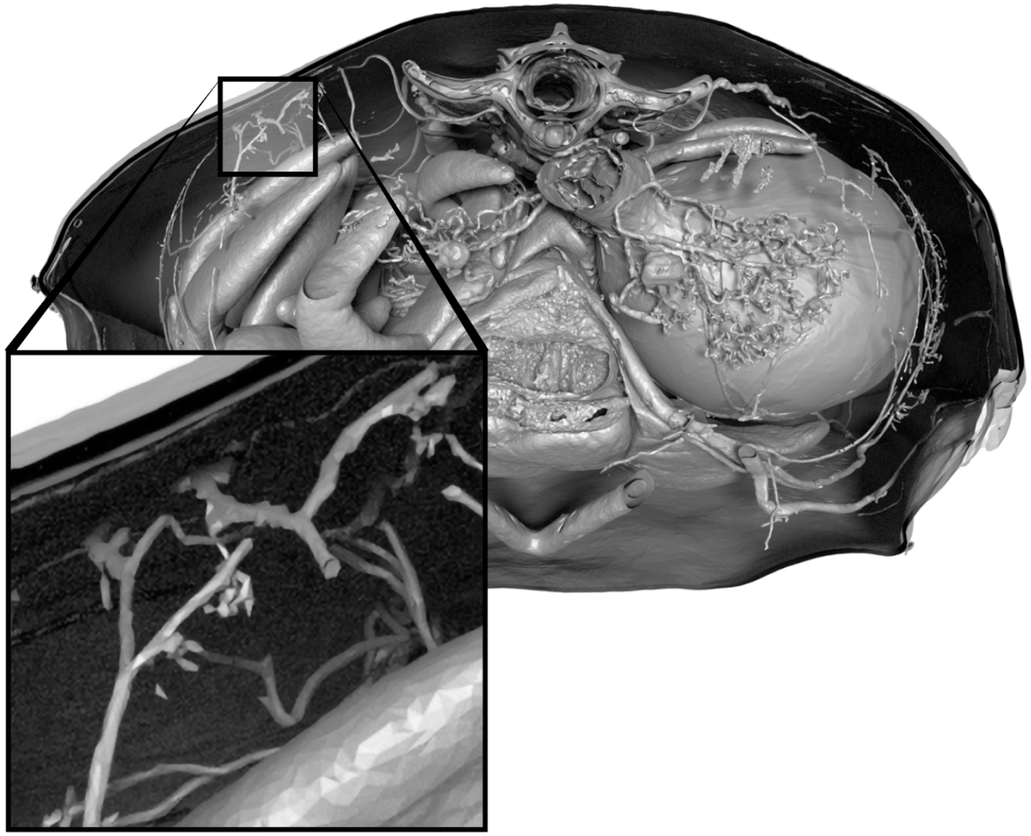
Input



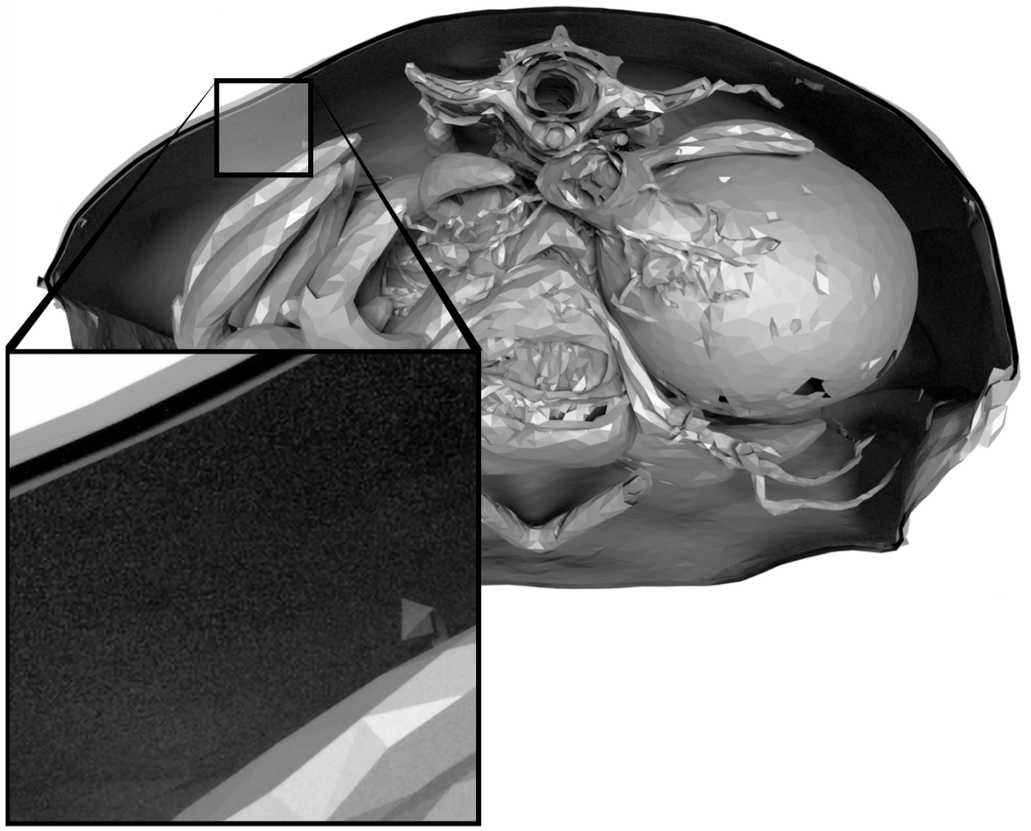
FASTTETWILD, **1 hour 25 minutes**

Hu et al, *"Fast Tetrahedral Meshing in the Wild"* (2020)

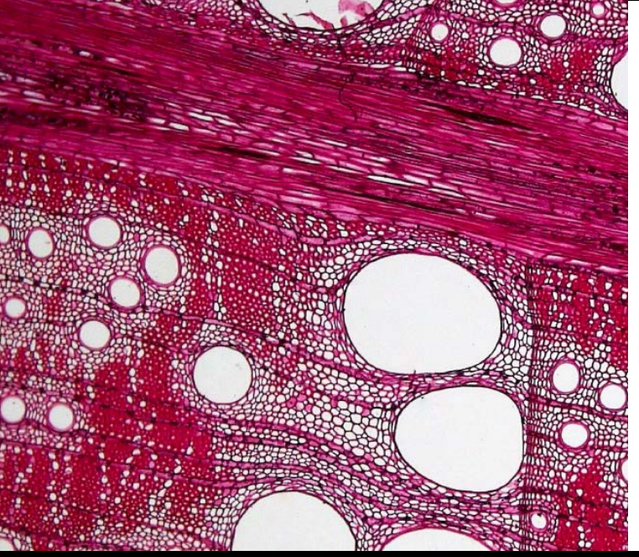
Even when meshing "succeeds", critical details can be lost

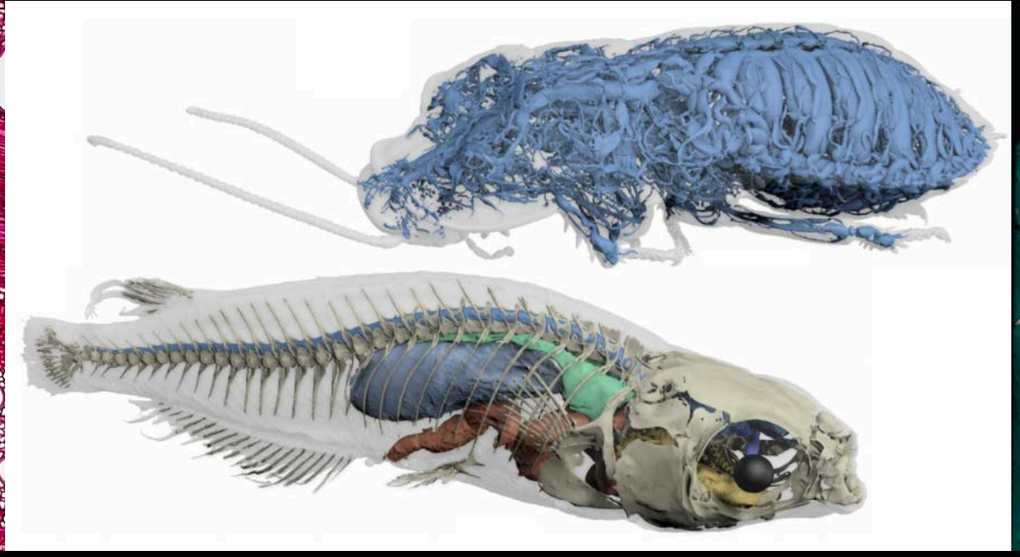


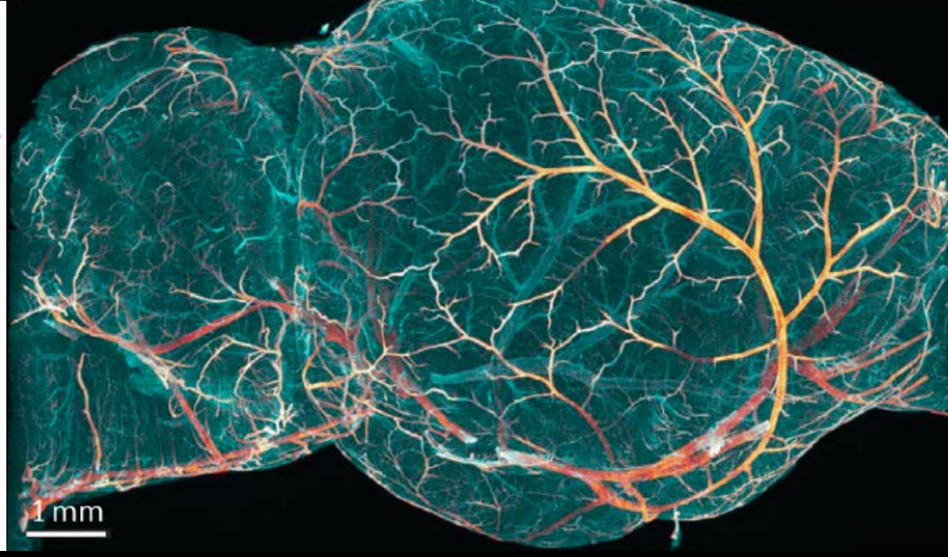
boundary mesh (*input*)



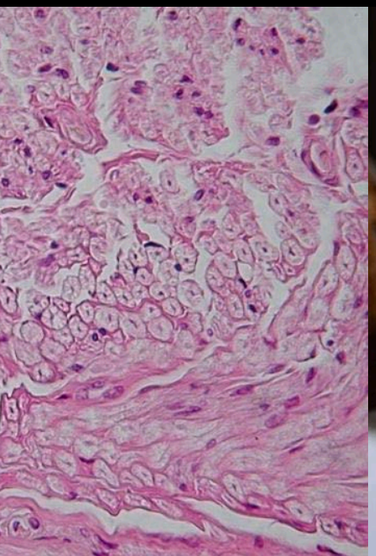
FASTTETWILD (*output boundary*)

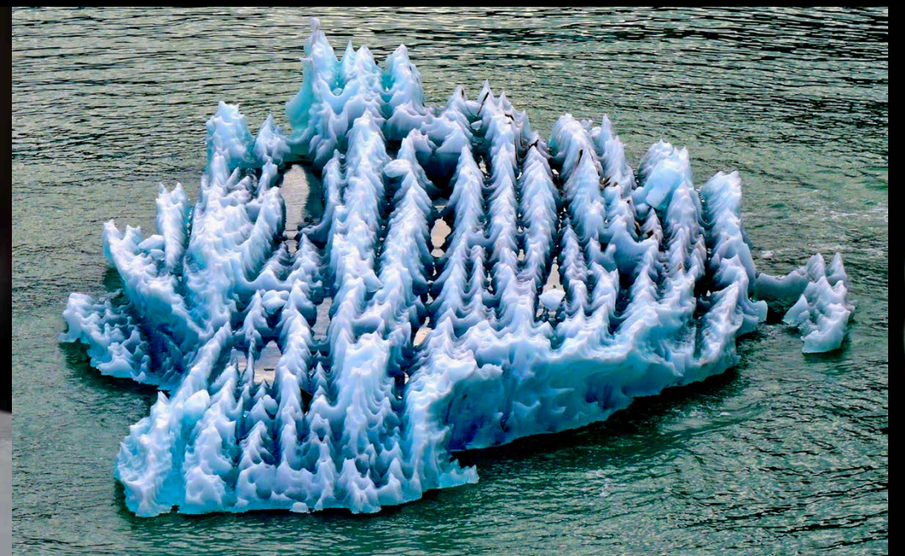






To faithfully simulate nature, we *must* be able to handle a much greater level of **geometric complexity**.





Inspiration from Photorealistic Rendering

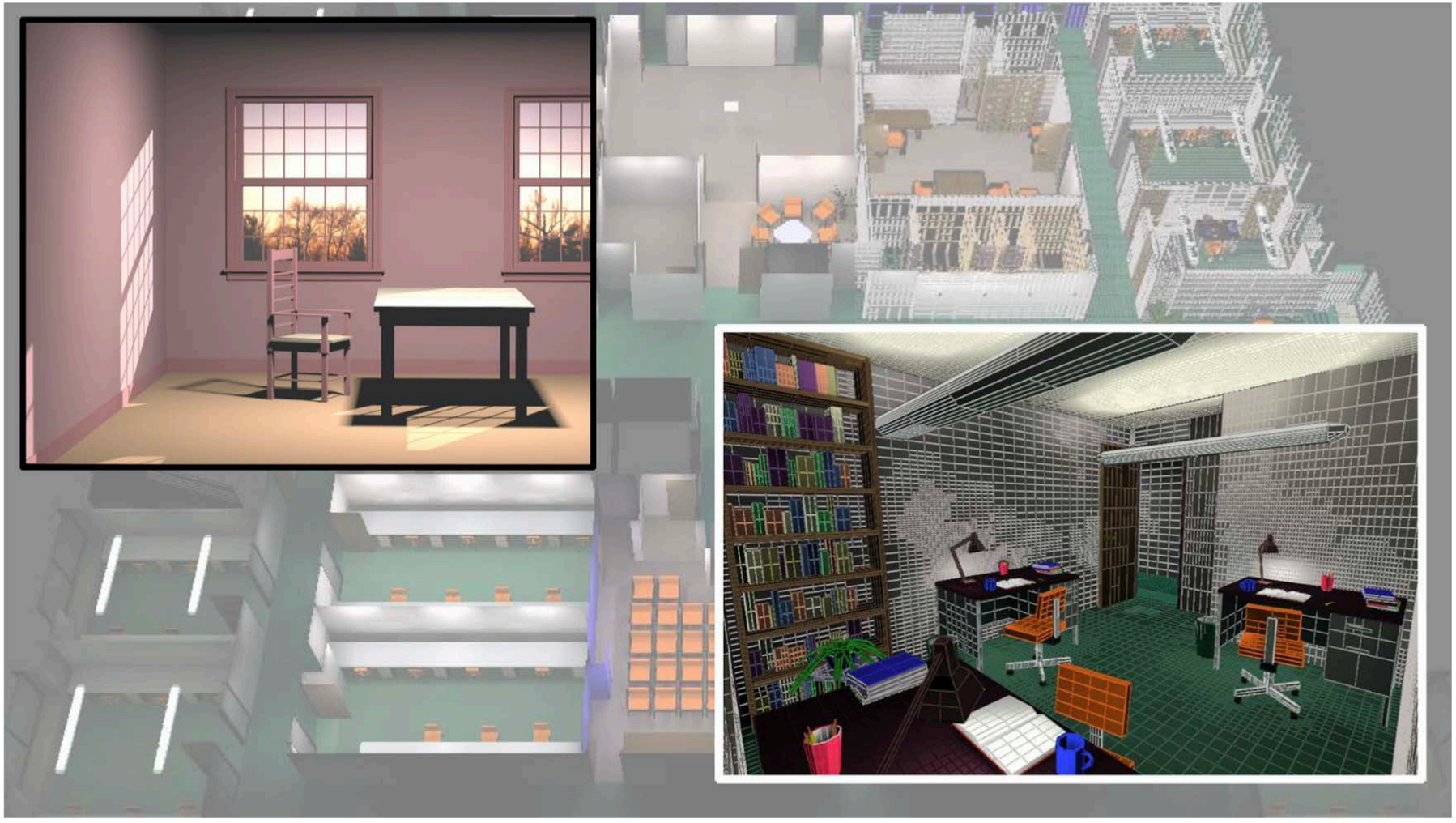
Problem: given a description of a 3D scene (geometry, materials, lights, camera), synthesize an image indistinguishable from a photograph.

3D model

synthesized image

Rendering: from Finite Elements to Monte Carlo

Early days of rendering: *finite element radiosity*



global & painful!

mesh scene

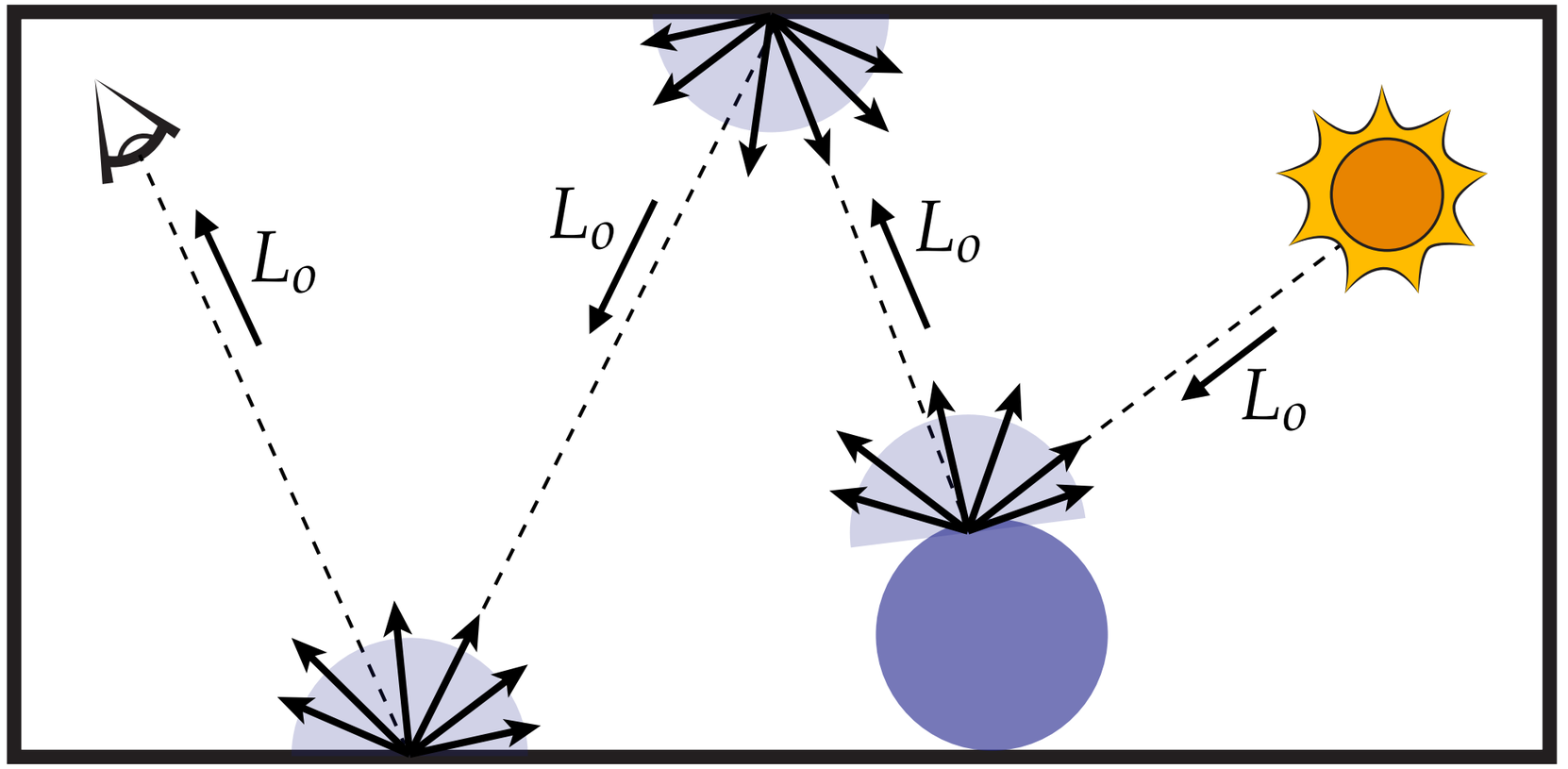
setup huge matrix

perform global solve

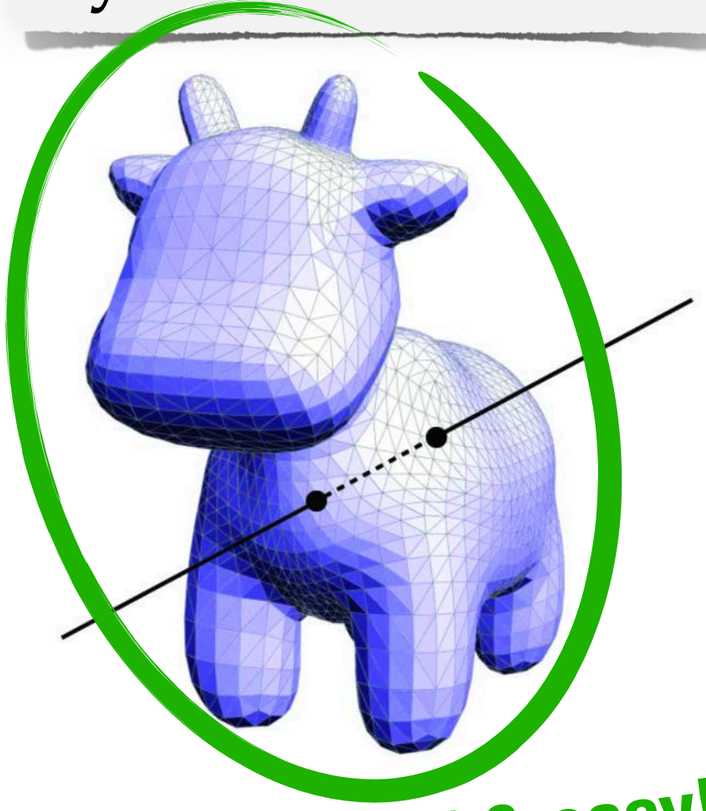
Rendering: from Finite Elements to Monte Carlo

Monte Carlo ray tracing skips meshing entirely, via repeated random sampling

recursively shoot rays

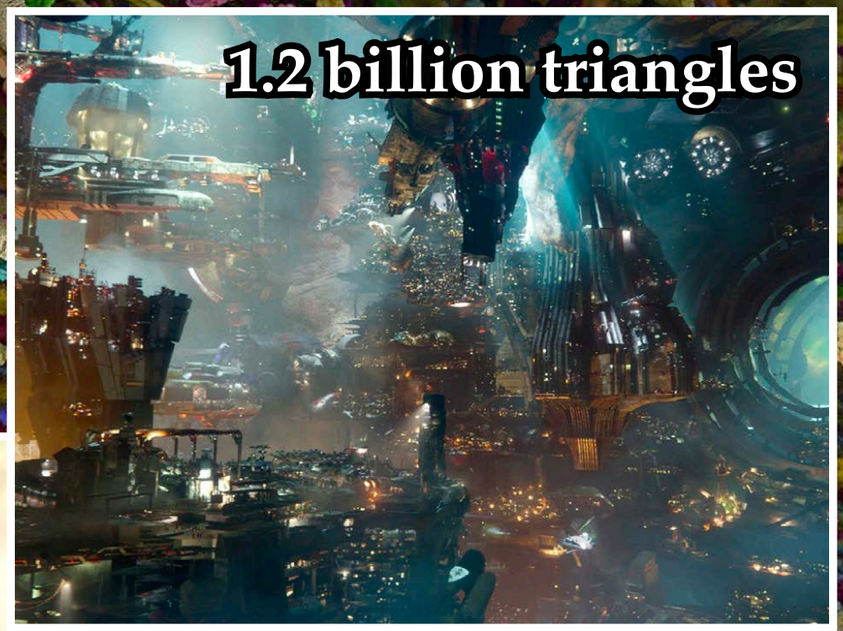


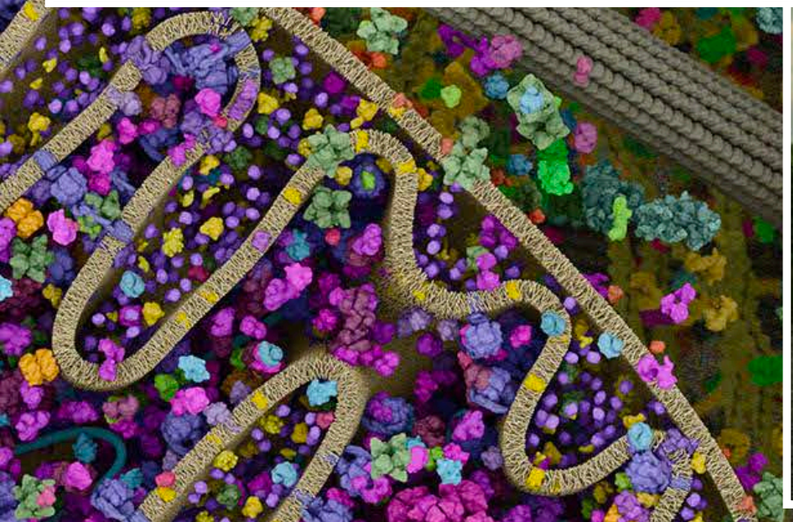
ray-scene intersections

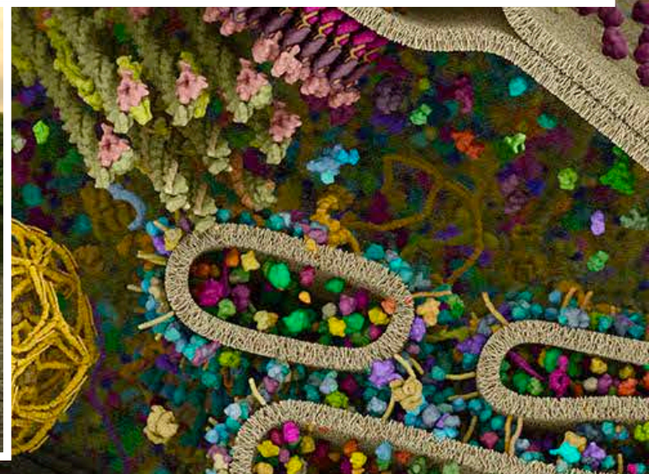


local & easy!

*Monte Carlo rendering can now handle **immense** geometric complexity*



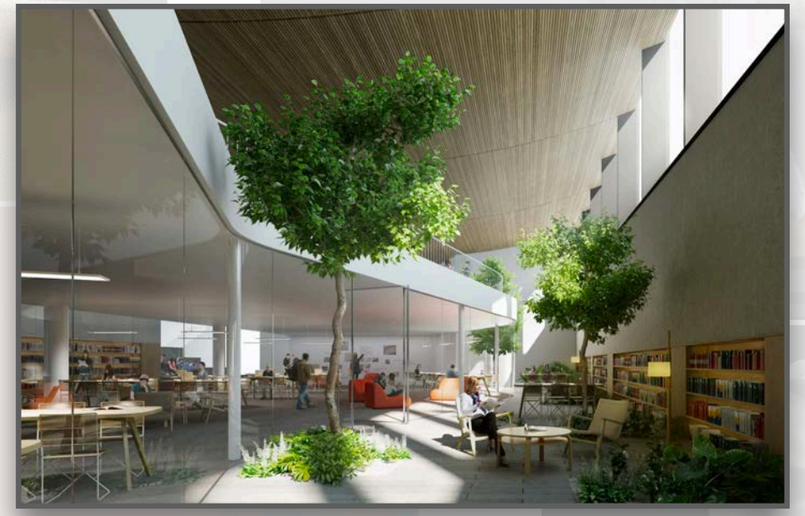




Rendering: from FEM to Monte Carlo

The move to Monte Carlo integration has enabled whole new industries:

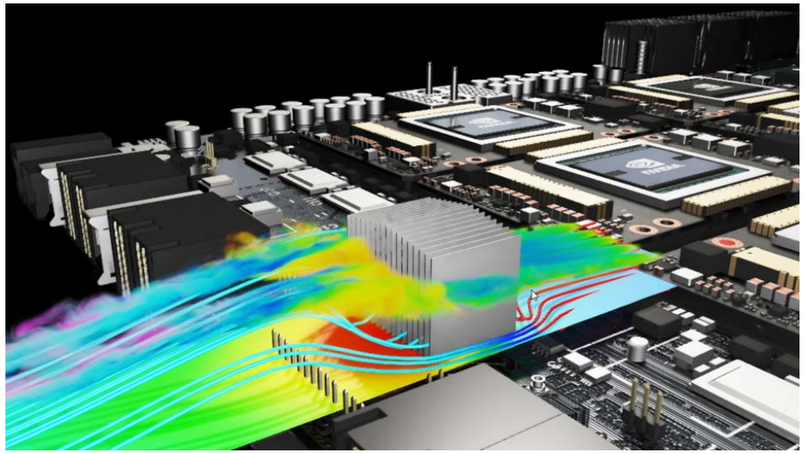
ENTERTAINMENT



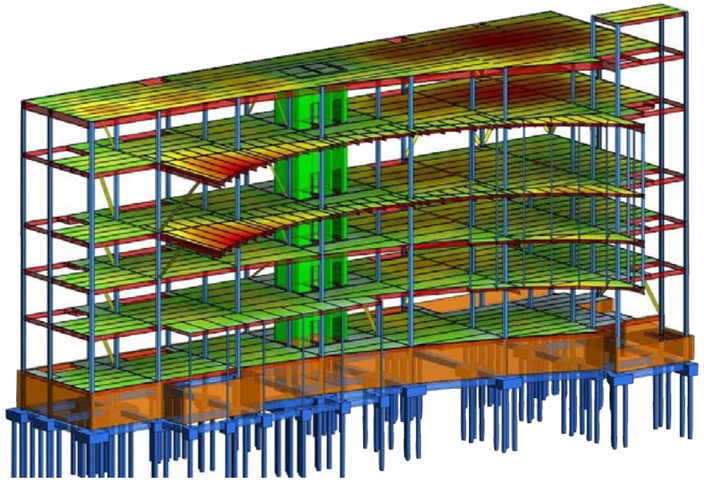
PRE-VISUALIZATION

COMPUTER VISION

There's a lot of physics beyond light transport!



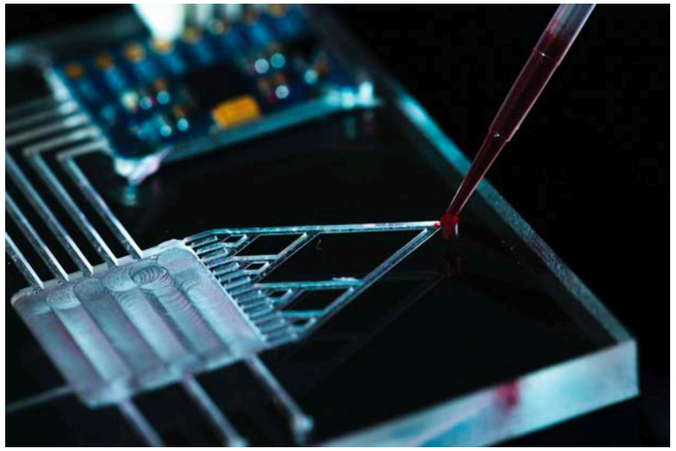
thermal analysis



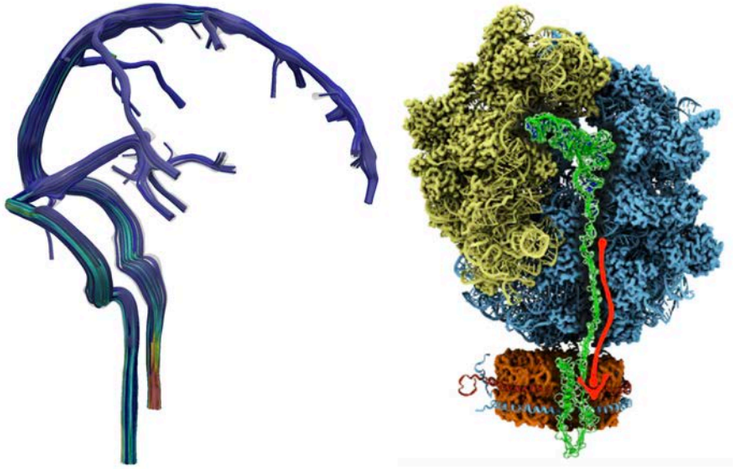
structural analysis

electrical capacitance

acoustic modeling



microfluidics



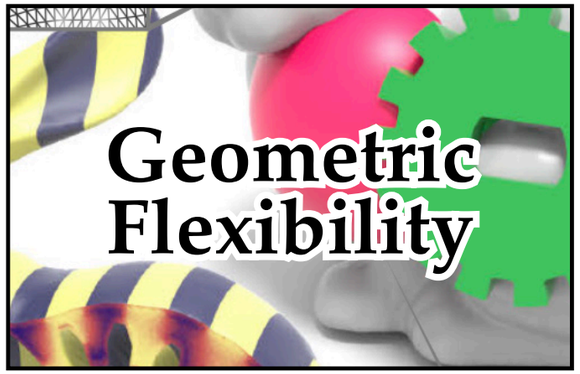
biophysics

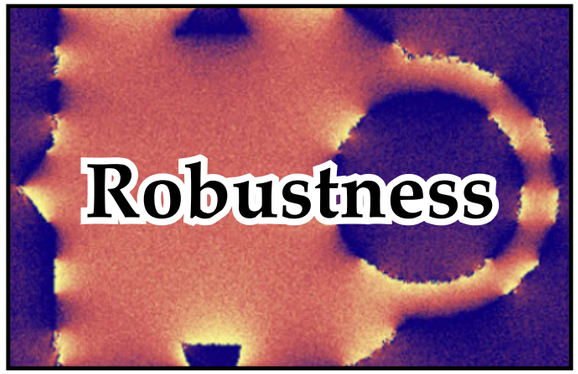
If Monte Carlo methods can make a physically accurate image of practically anything, can we use it to do other kinds of simulation too?

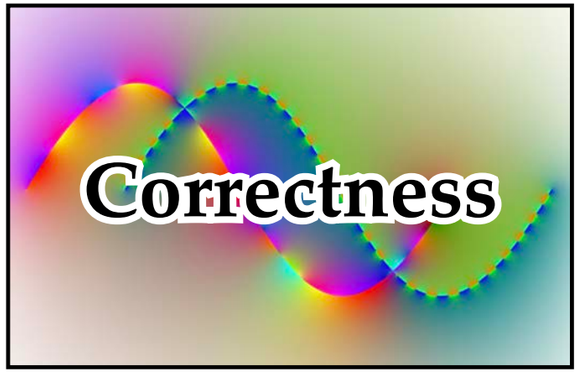
The answer is of course “yes.” :-)

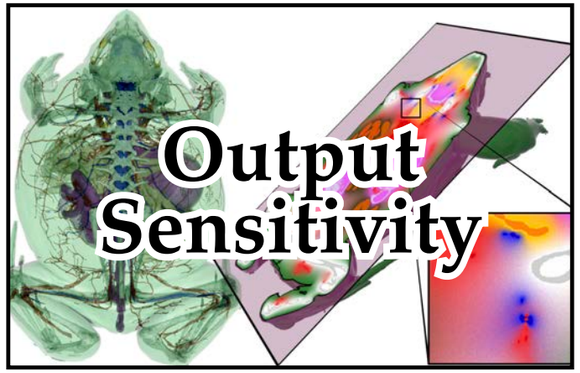
Benefits of Monte Carlo Methods

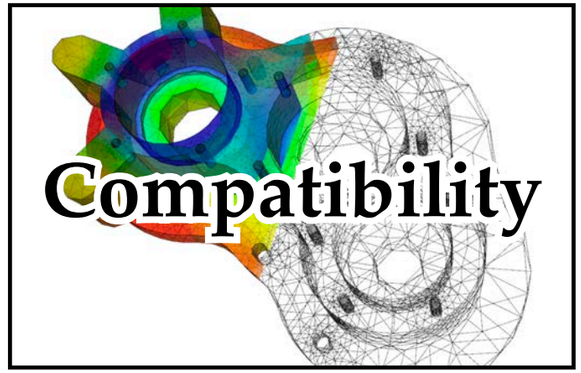
Walk on spheres provides many of the same benefits for science & engineering:









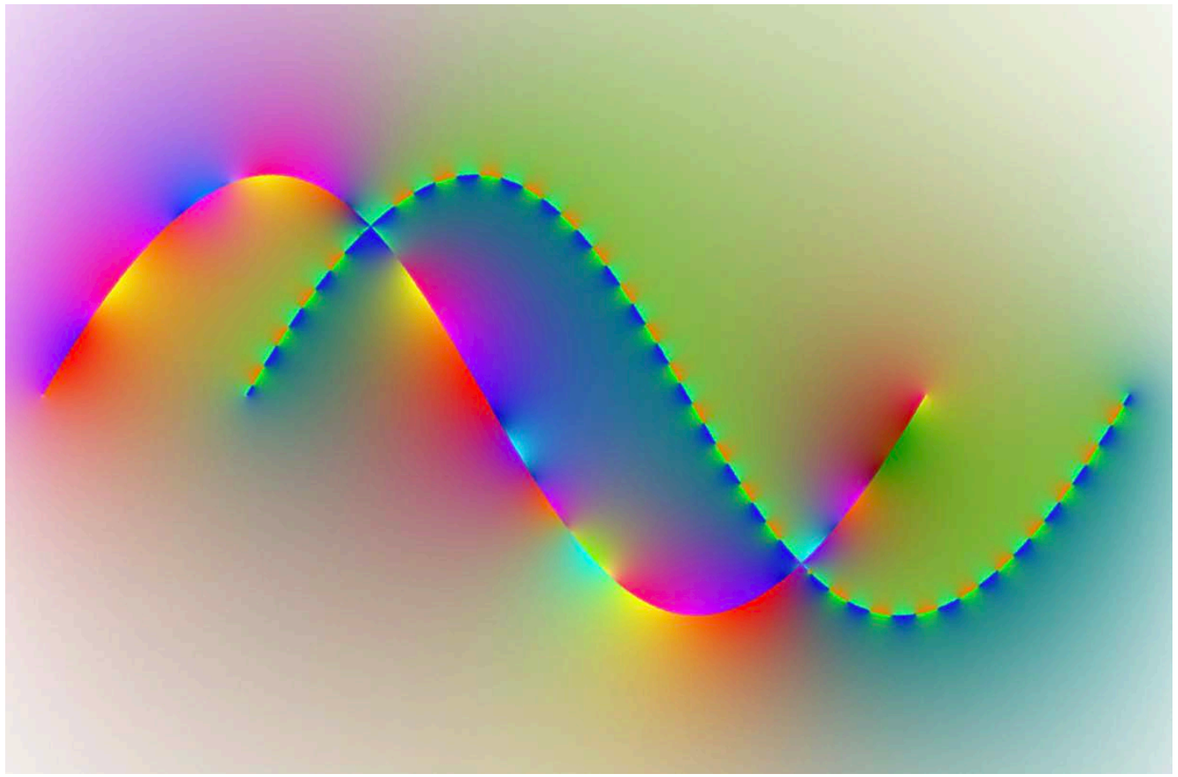


Benefits of Monte Carlo — Correctness

RENDERING

[Wann Jensen 1995]

PDEs



[Sawhney & C. 2020]

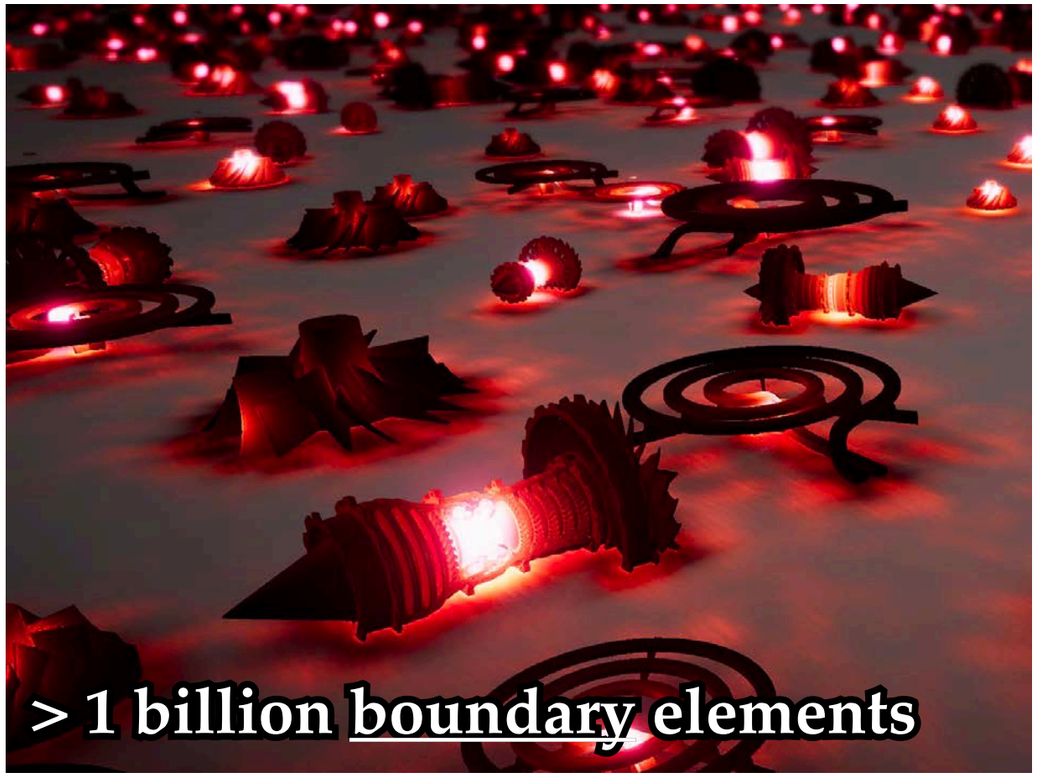
Key idea: as long as equation is well-posed, numerical solution *will* be correct.

Benefits of Monte Carlo — Scalability

RENDERING

[Georgiev et al 2018]

PDEs



[Sawhney, Seyb, Jarosz, C. 2022]

Key idea: cost of geometric detail grows like $O(n \log n)$.

Benefits of Monte Carlo — Parallelism

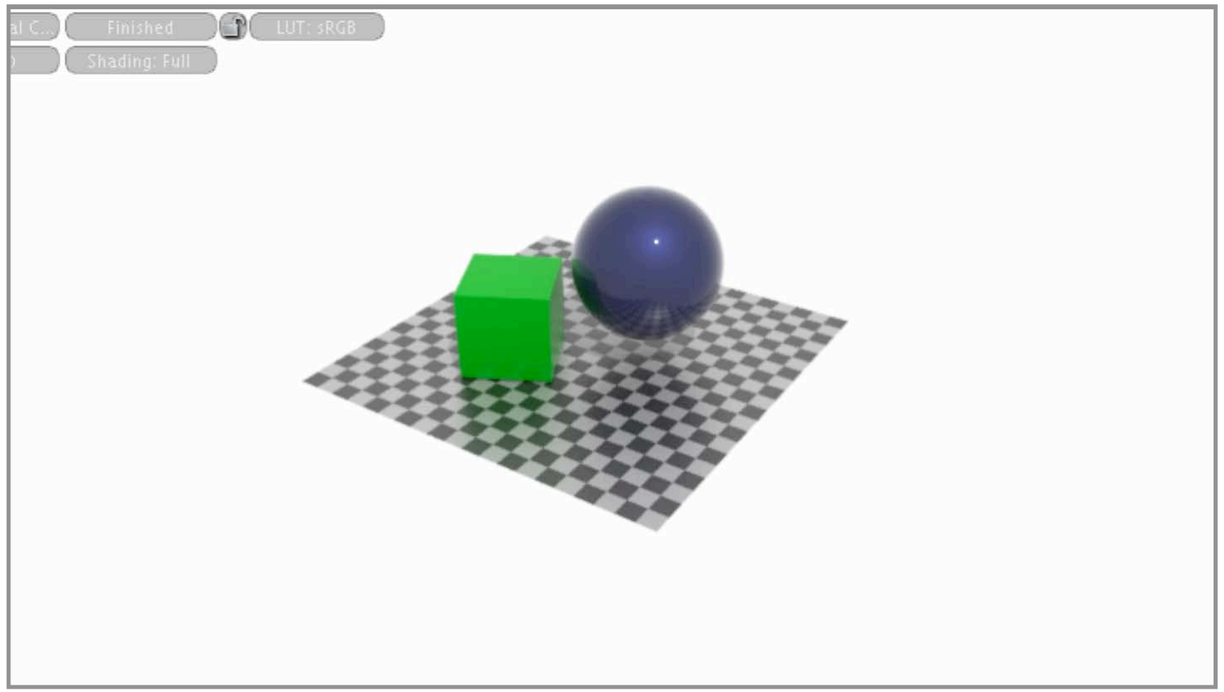
RENDERING

PDEs

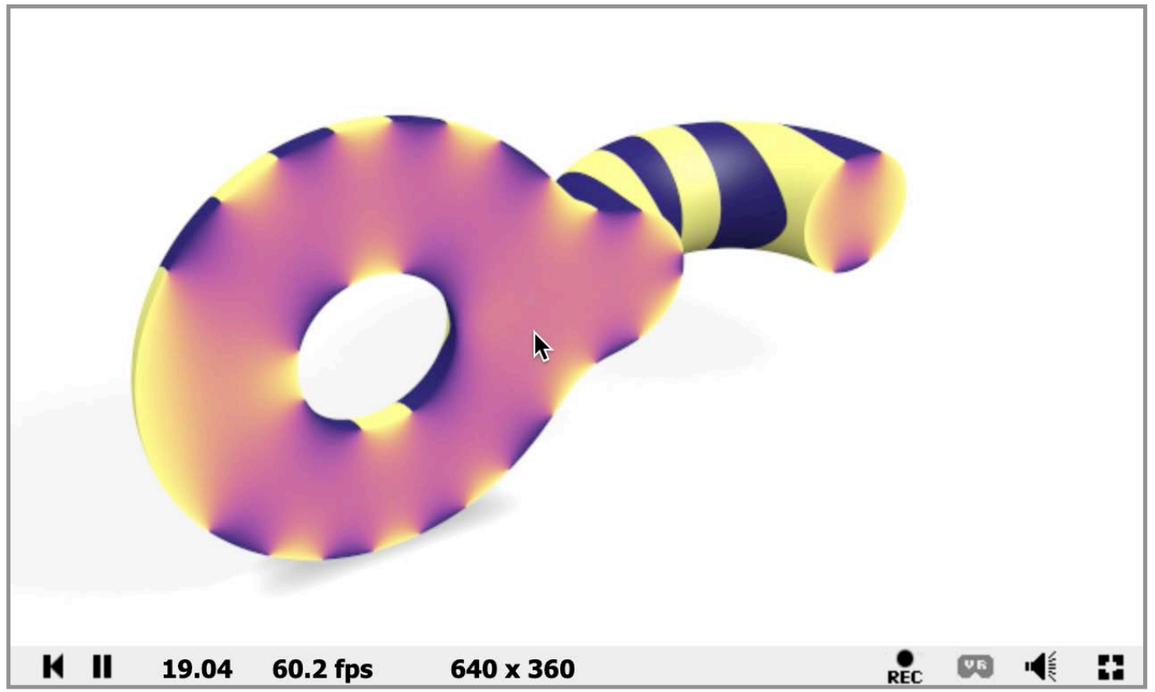
Key idea: just run on N processors, take average of N final estimates

Benefits of Monte Carlo — Progressive

RENDERING



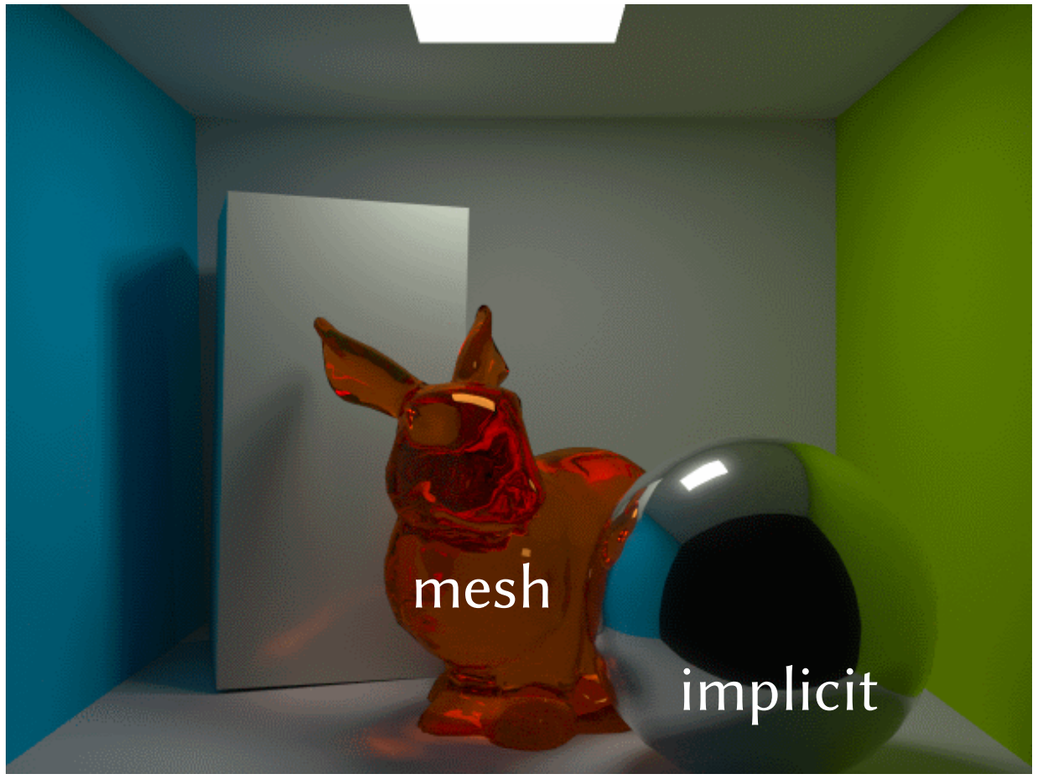
PDEs



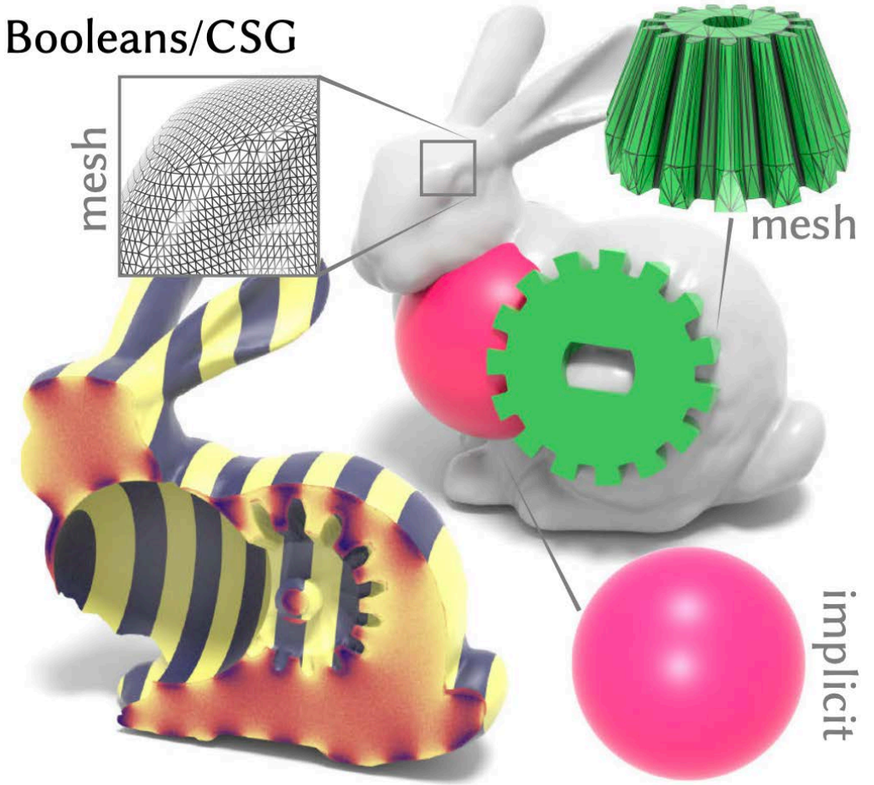
Key idea: fast-but-reliable “preview” enables rapid exploration

Benefits of Monte Carlo — Geometric Flexibility

RENDERING



PDEs

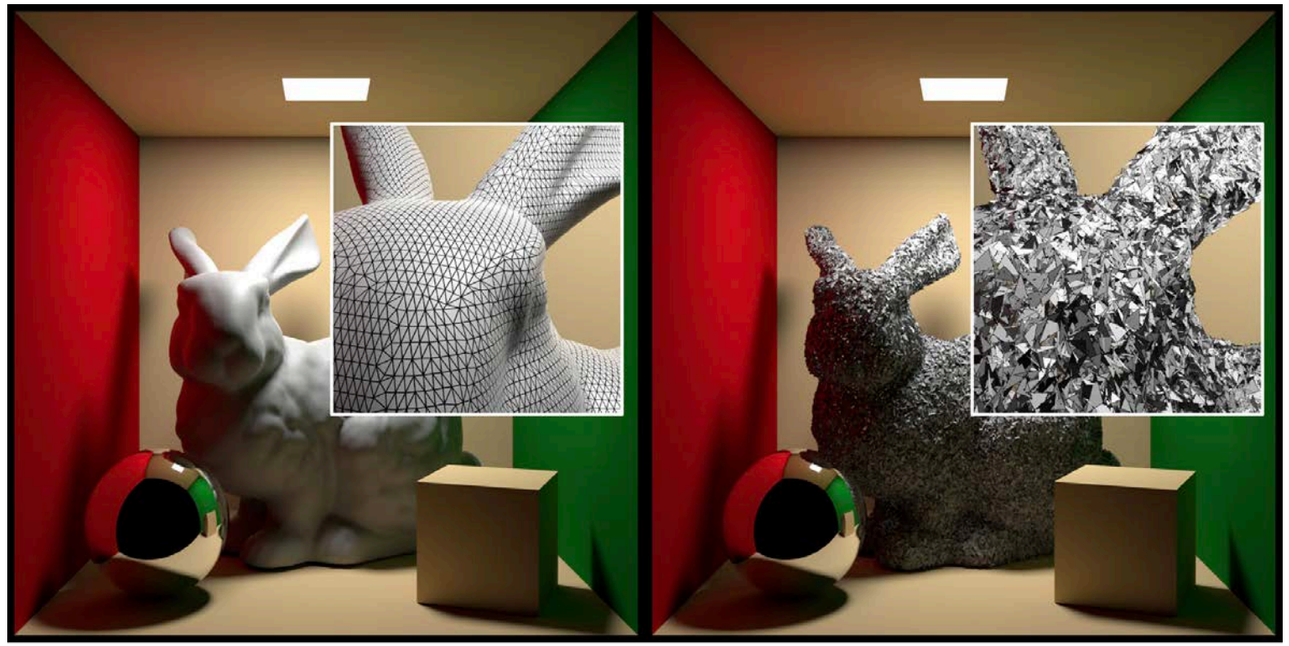


[Sawhney & C. 2020]

Key idea: can work directly with *heterogeneous* geometry, without conversion.

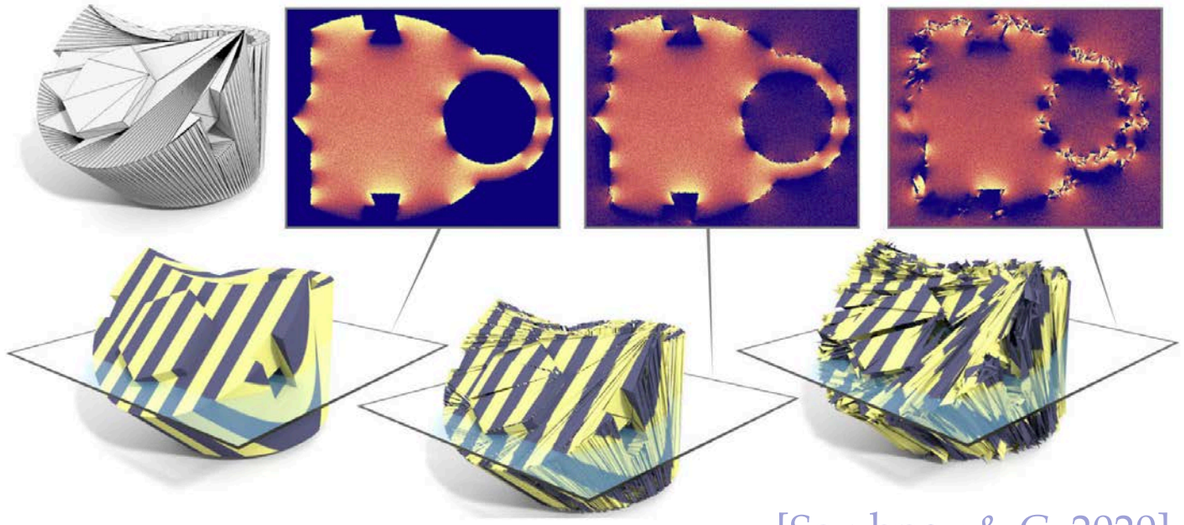
Benefits of Monte Carlo — Robustness

RENDERING



Beautiful. Still beautiful.

PDEs

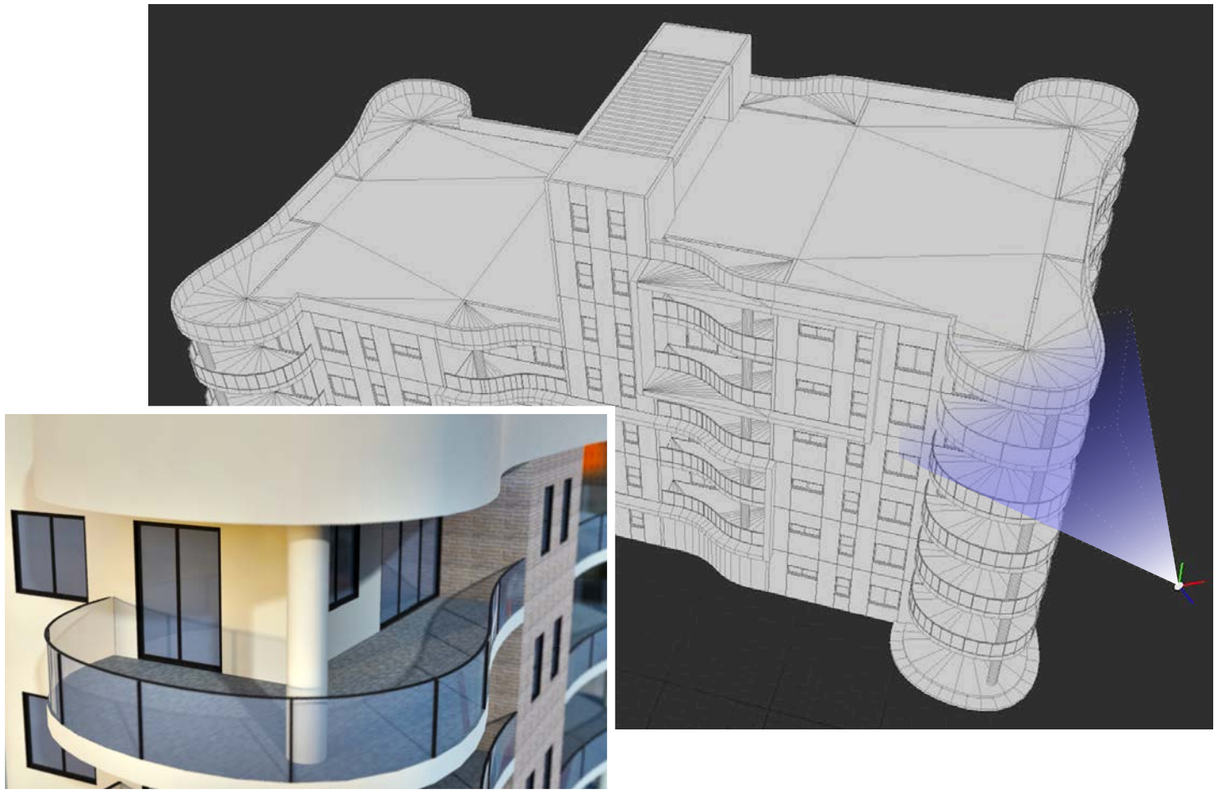


[Sawhney & C. 2020]

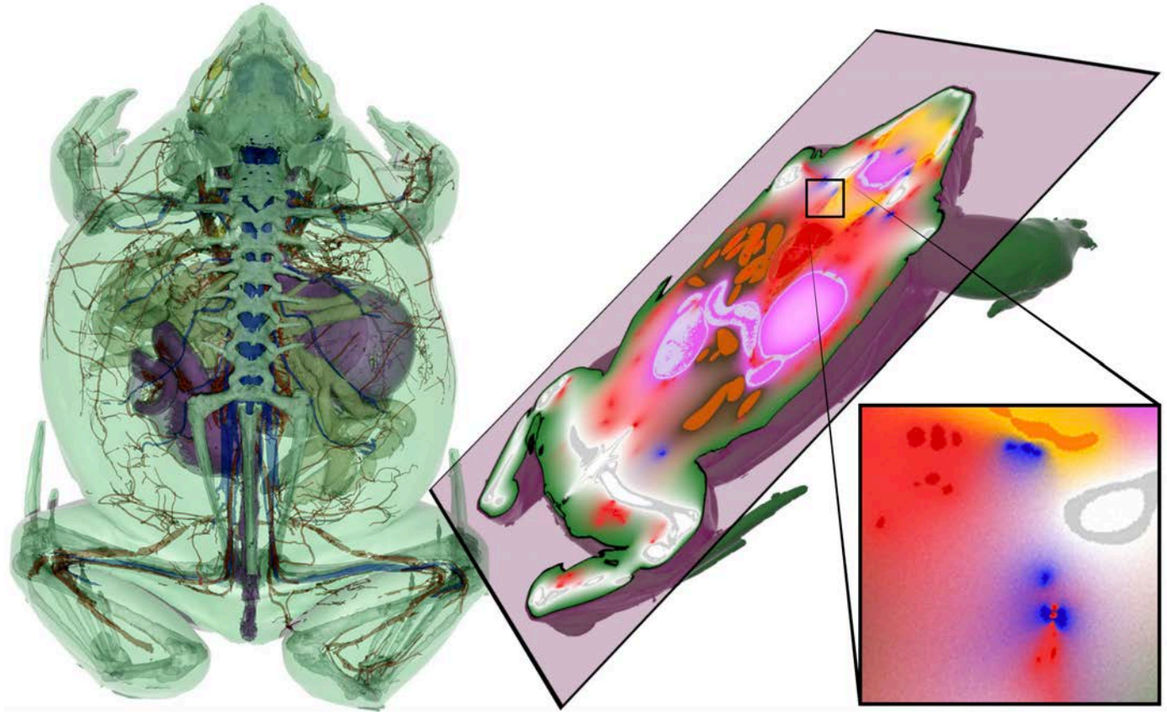
Key idea: solution quality degrades *gracefully*.

Benefits of Monte Carlo — Output Sensitivity

Rendering



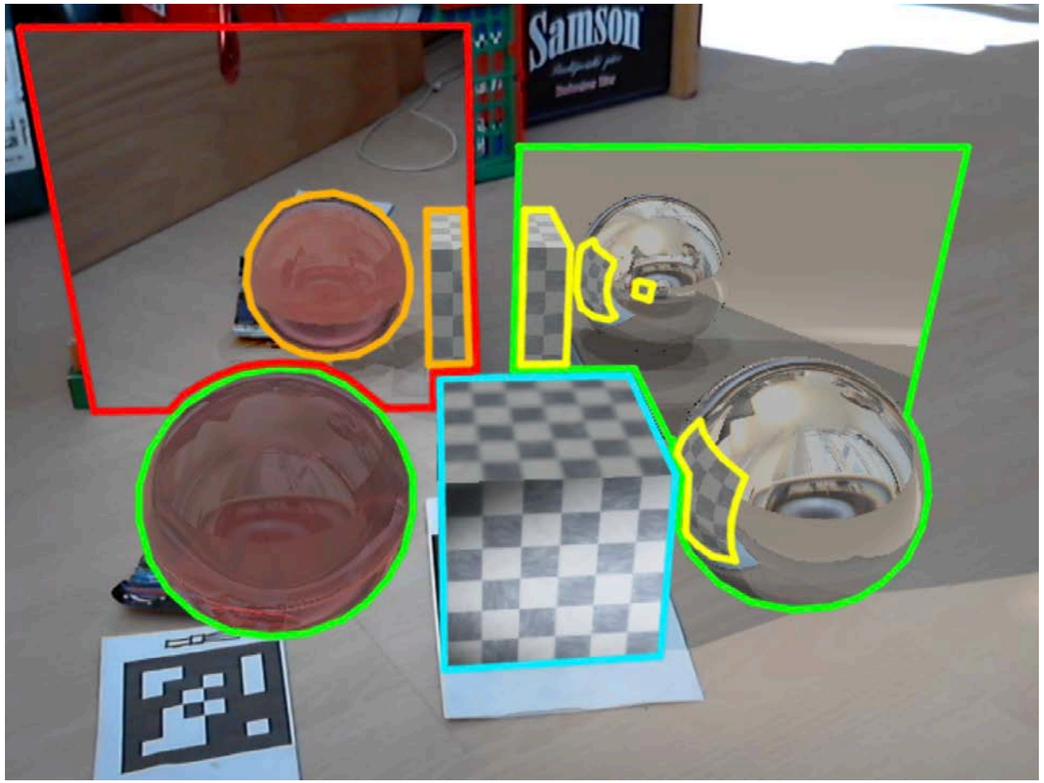
PDEs



Key idea: compute the solution only where you actually need it!

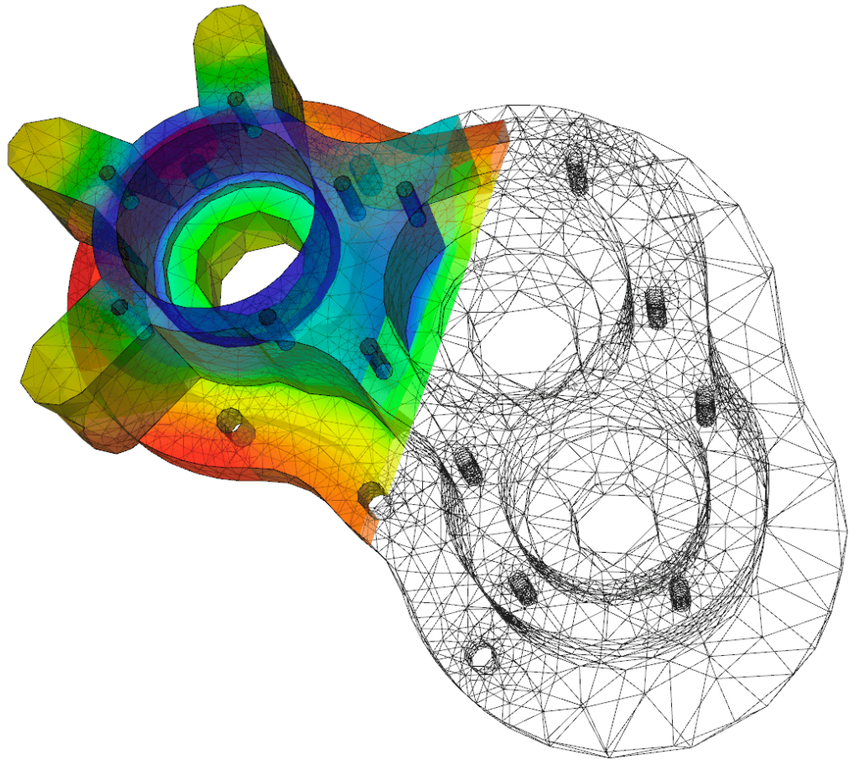
Benefits of Monte Carlo — Compatibility

RENDERING



e.g., use ray tracing inside rasterization

PDEs



e.g., use Monte Carlo inside mesh-based simulation

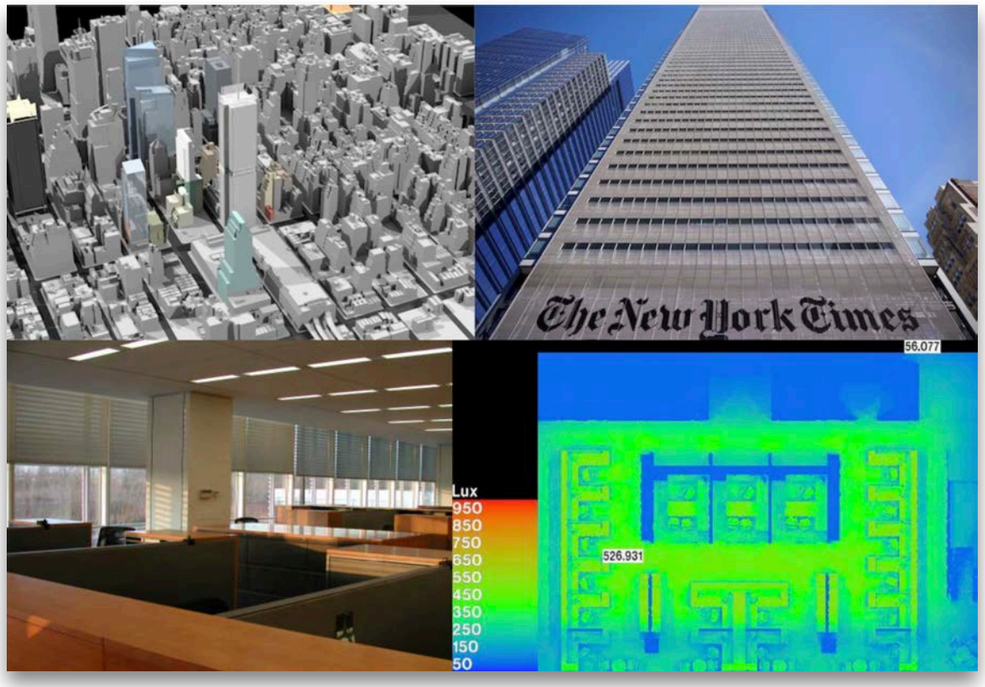
Key idea: not *forced* to use Monte Carlo everywhere

I'm not naïve

May **never** be able to solve certain PDEs via Monte Carlo…

Would be crazy not to see what we can build!

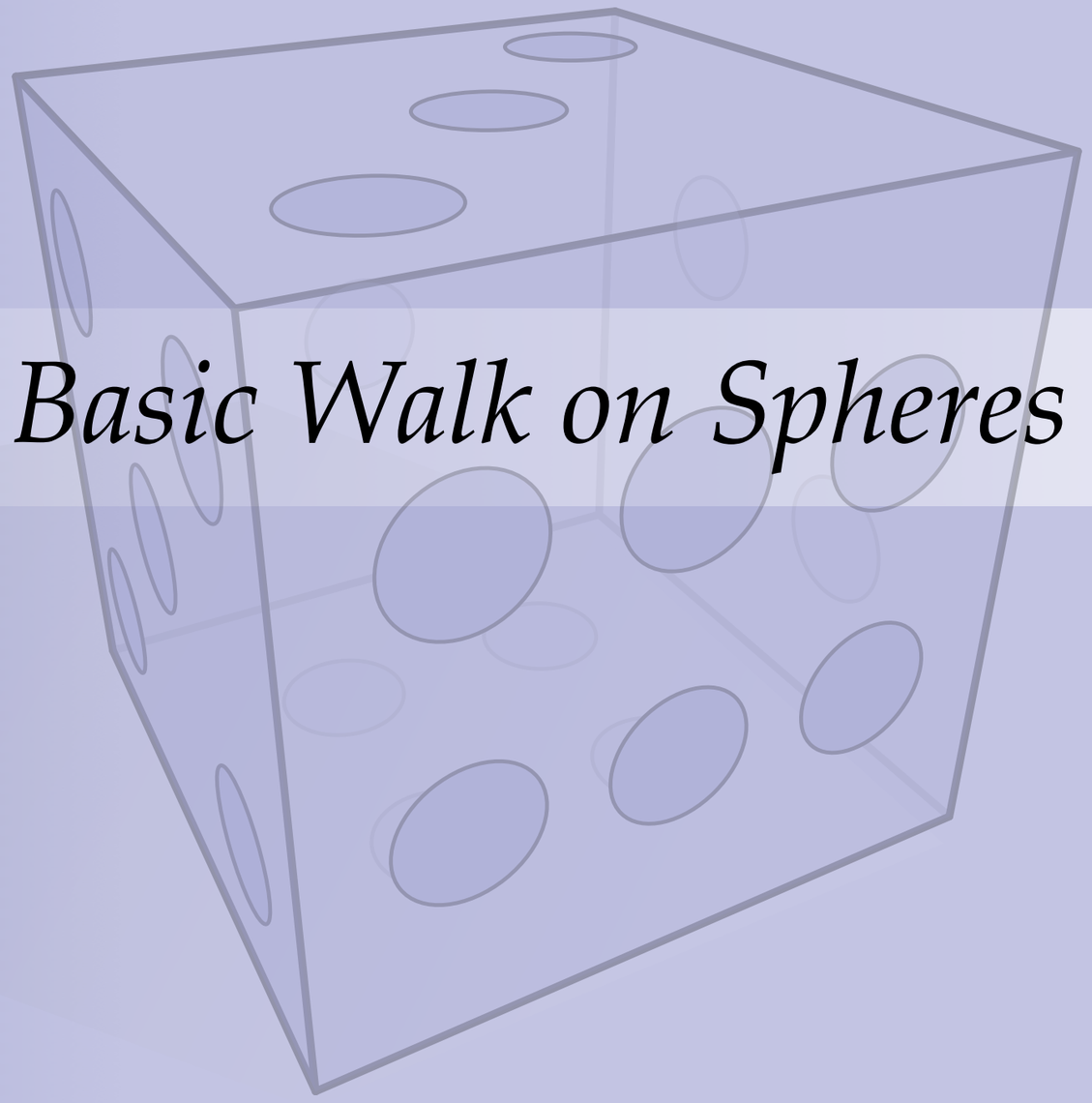
RENDERING



"Daylighting the New York Times Headquarters Building"
LBNL / New York Times / Anyhere Software (2015)

PDEs

Can we also predict energy efficiency directly from a detailed architectural model…?



Basic Walk on Spheres

Basic Walk on Spheres — Overview

- Most basic walk on spheres algorithm solves *Laplace equation*
  - nicely illustrates general principle
  - (we'll see more sophisticated equations/algorithms next lecture)
- Two ways to motivate:
  1. **Stochastic processes** (*Kakutani's principle*)
  2. **Potential theory** (*mean value property*)
- In general: interplay between stochastic & boundary integral formulations helps understand theory, algorithms

Ball vs. Sphere

Natural language uses the word “ball” for rather distinct phenomena…

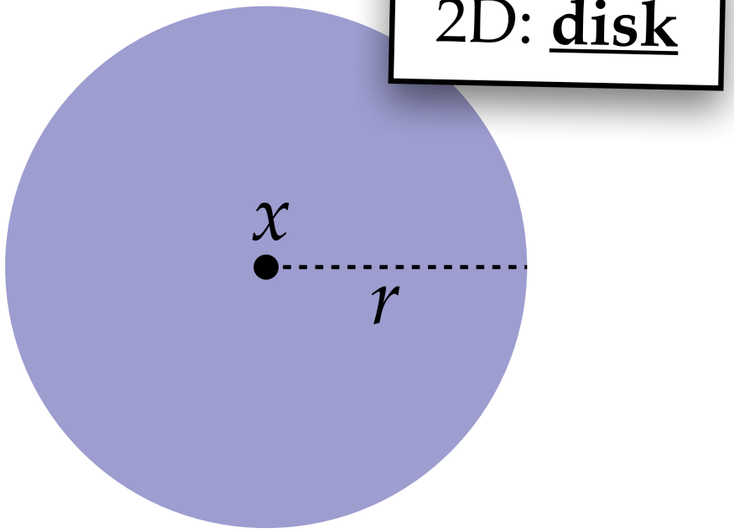
meatball
(solid)

soccer ball
(hollow)

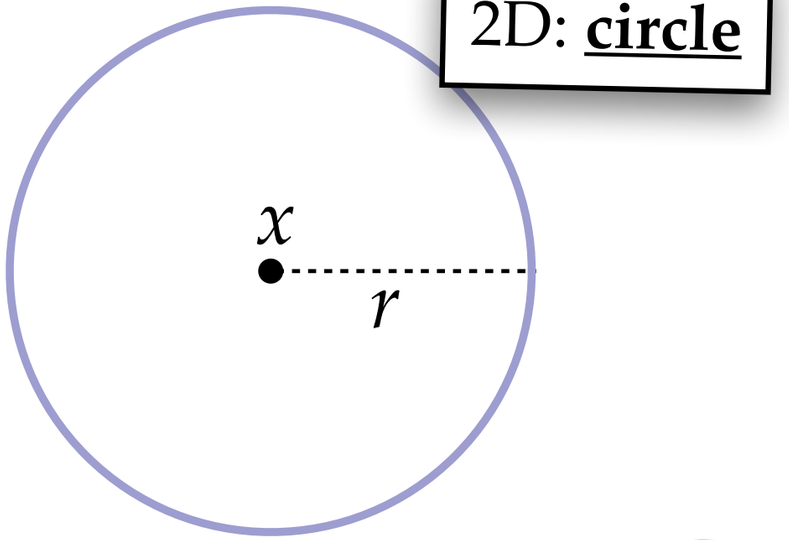
Ball vs. Sphere

Mathematics: a **ball** is a solid region; a **sphere** is the outer "shell."

BALL



SPHERE



$$B_r(x) := \{y \in \mathbb{R}^n : |y - x| \leq r\}$$

inequality

$$\partial B_r(x) := \{y \in \mathbb{R}^n : |y - x| = r\}$$

equality

In this lecture, won't need to be too careful about open vs. closed…

Dirichlet Problem

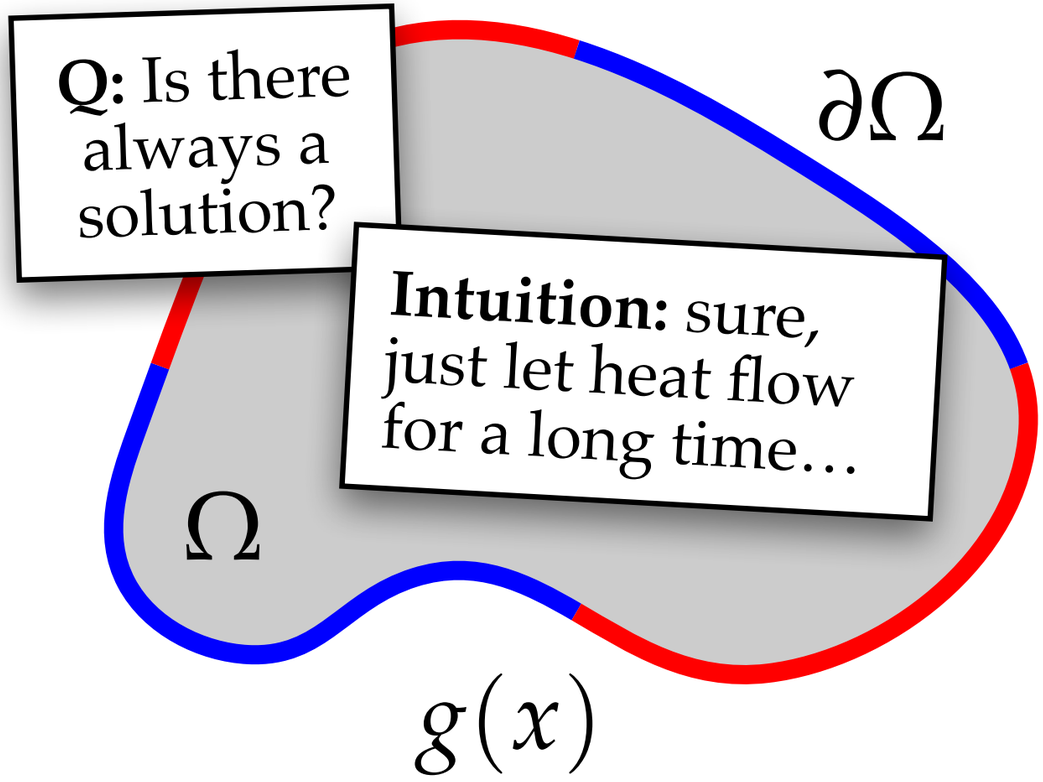
GIVEN: values on the boundary of a region Ω

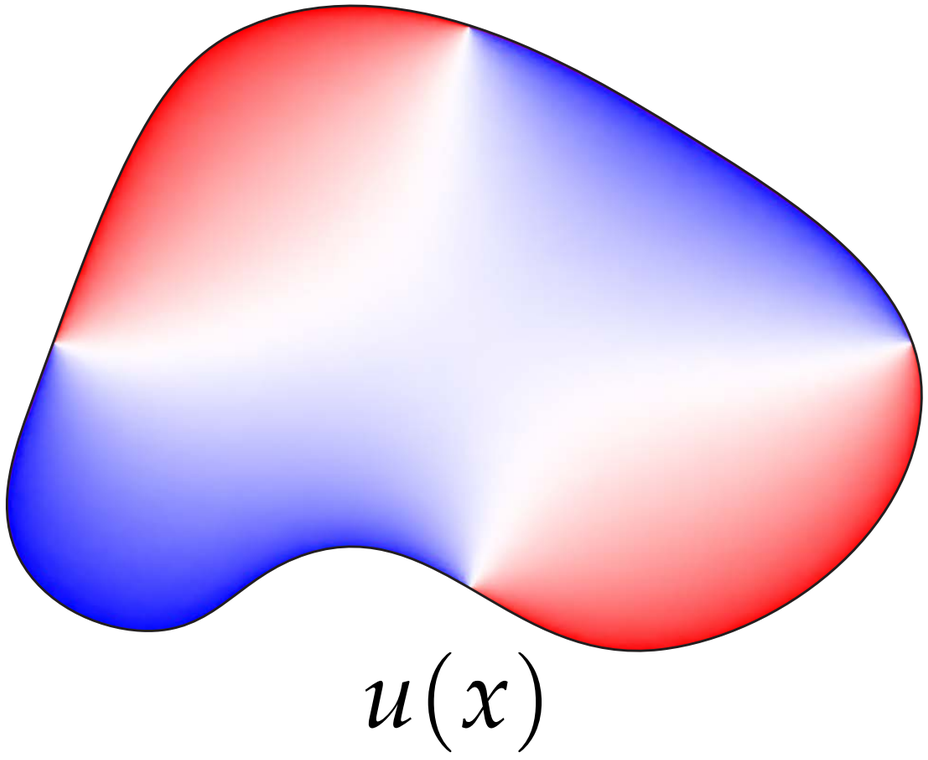
FIND: smooth interpolation into interior

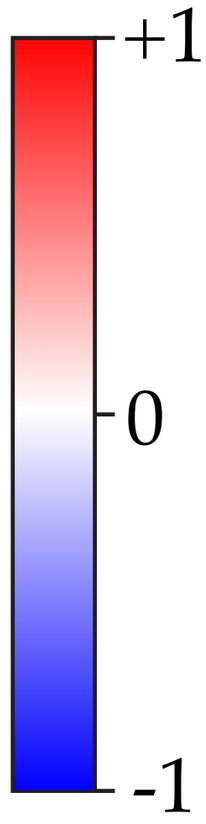
$$\begin{aligned} \Delta u &= 0 \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$

Q: Is there always a solution?

Intuition: sure, just let heat flow for a long time…







Aside: History of Dirichlet's Principle

The history of the Dirichlet principle is remarkable. Green, Dirichlet, Thomson, and others of their time regarded it as a completely sound method and used it freely. Then Riemann in his complex function theory showed it to be extraordinarily instrumental in leading to major results. All of these men were aware that the fundamental existence question was not settled, even before Weierstrass announced his critique in 1870, which discredited the method for several decades. The principle was then rescued by Hilbert and was used and extended in this [the 20th] century. Had the progress made with the use of the principle awaited Hilbert's work, a large segment of nineteenth-century work on potential theory and function theory would have been lost.

$$\min_u \int_\Omega |\nabla u|^2 \, dV$$

$$\updownarrow$$

$$\Delta u = 0$$

Green

Dirichlet

Riemann

Weierstrass

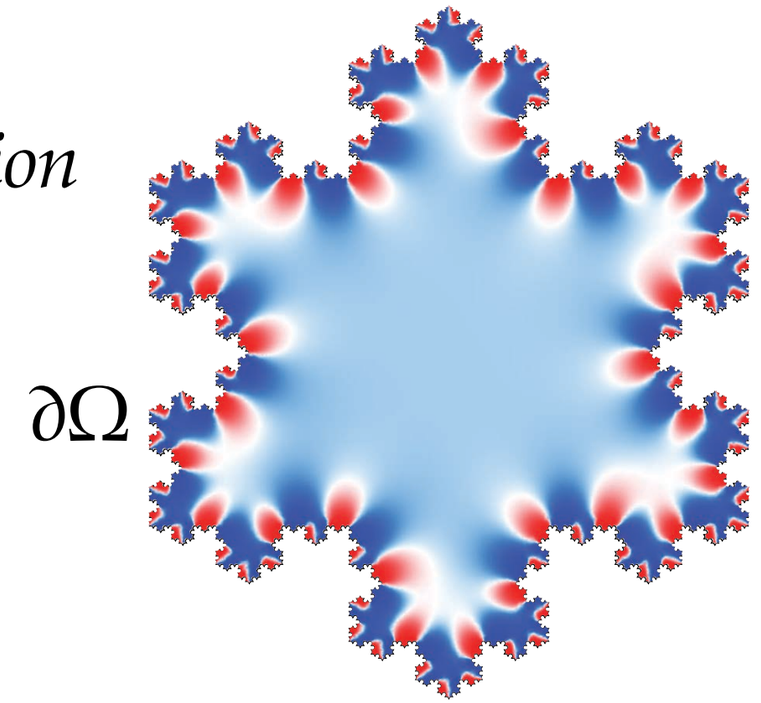
Hilbert

from Morris Kline, *"Mathematical thought from ancient to modern times"*, vol 3 (1972)

Dirichlet Problem & Harmonic Functions

- Consider a domain $\Omega \subset \mathbb{R}^n$ with sufficiently regular* boundary $\partial\Omega$, and a continuous** function $g : \partial\Omega \to \mathbb{R}$
- **Dirichlet problem** seeks solution to *Laplace equation*

$$\begin{aligned} \Delta u &= 0 \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$



- **Fact.** A unique solution always exists
- Solution is called a **harmonic function**—central object in *potential theory*

*Must be $C^{1,\alpha}$ (derivative is Hölder continuous)

**This assumption can be relaxed ("Perron solution")

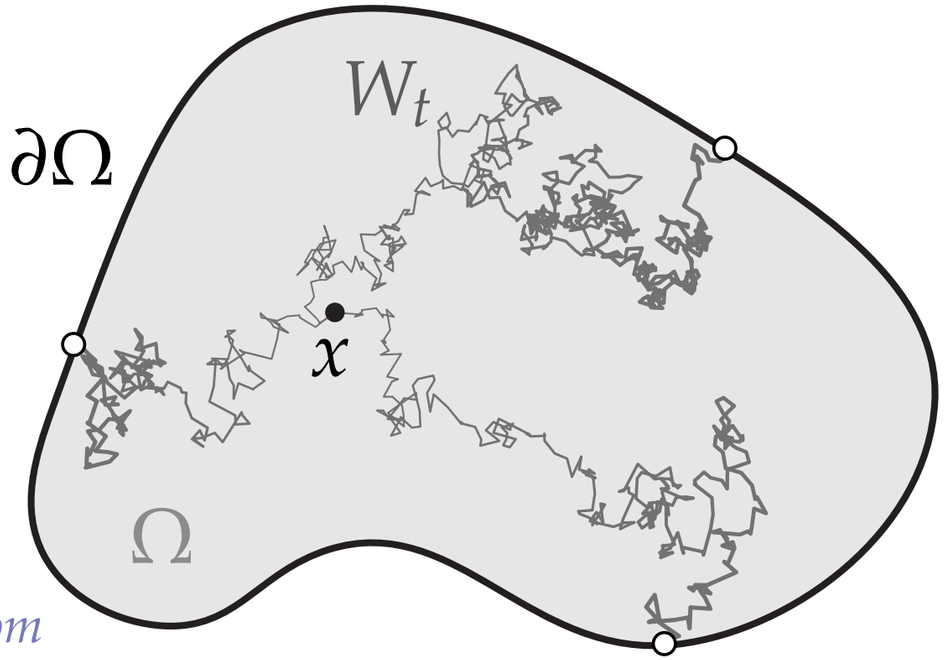
Stochastic Perspective: Kakutani's Principle

- Suppose we want to solve Dirichlet problem for domain Ω, boundary data g: $\partial\Omega \to \mathbb{R}$
- Consider the stochastic process $dX_t = dW_t$, starting at a point x, *i.e.*, pure Brownian motion
- Define the *hitting time* $T := \inf\{t > 0 \,|\, X_t \in \partial\Omega\}$
  - note that T itself is a random variable
  - analogous to an adapted process, T depends only on *past* events ("can't bet on future knowledge")
- **Kakutani's principle** then says that

$$u(x) = \mathbb{E}^x[g(X_T)] \xrightarrow{\text{Monte Carlo}} \frac{1}{N}\sum_{i=1}^{N} g((X_i)_{T_i})$$

ith random walk (over X_i); *ith hitting time* (over T_i)

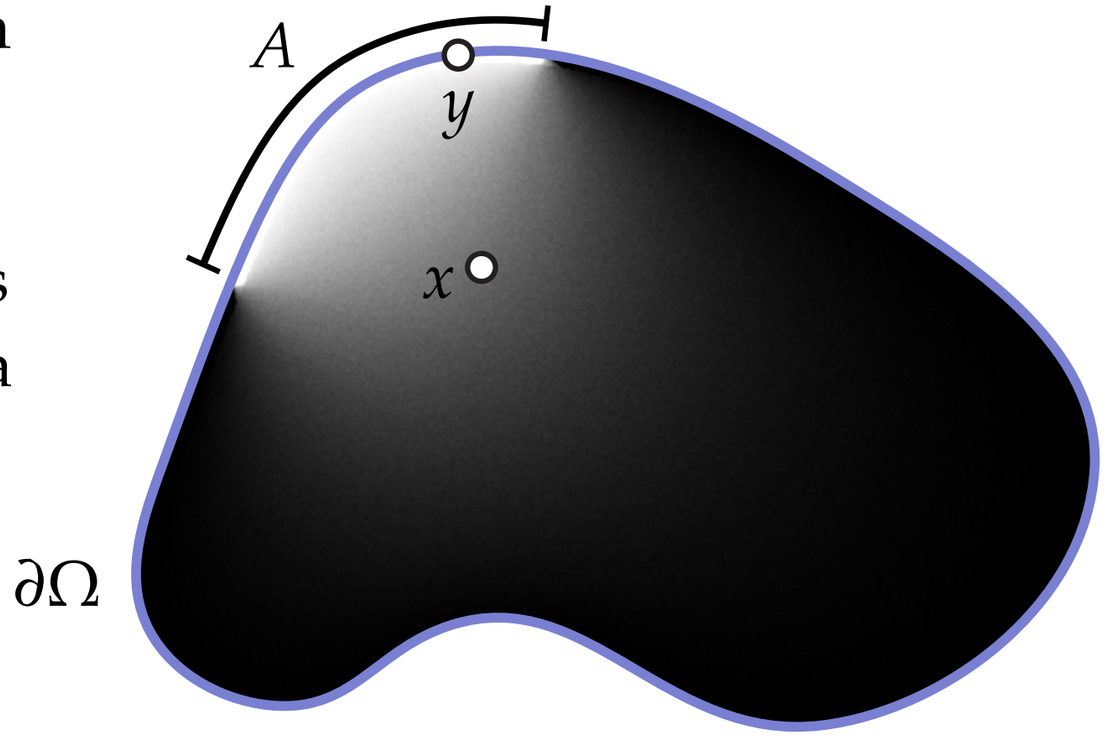
$$\begin{aligned} \Delta u &= 0 \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$



Kakutani's Theorem

Theorem (Kakutani 1944). Consider a domain Ω with sufficiently regular boundary $\partial\Omega$, and let $A \subset \partial\Omega$ be a region on the boundary. Then the probability $P(x, A)$ that a Brownian process X_t originating at x intersects A before $\partial\Omega \backslash A$ is a harmonic function satisfying

$$\lim_{x \to y} P(x, A) = \begin{cases} 1, & y \in A, \\ 0, & y \notin A\,. \end{cases}$$



The solution to the Dirichlet problem on Ω with boundary data $g : \partial\Omega \to \mathbb{R}$ is then given by the integral of g with respect to this probability (*"harmonic measure"*):

$$u(x) = \int_{\partial\Omega} P(x, dy) g(y)$$

Review: Solving Dirichlet Problem via Euler-Maruyama

① Simulate Brownian motion using Euler-Maruyama

$$dX_t = dW_t$$

$\widehat{u} \leftarrow 0$ initialize estimate of $u(x_0)$

for $i = 1, \ldots, N$:

  $x \leftarrow x_0$ start at evaluation point

  while $x \in \Omega$: still in the domain

    $\xi \sim \mathcal{N}^{2D}(0,1)$ sample direction

    $x \leftarrow x + \tau\xi$ take a step

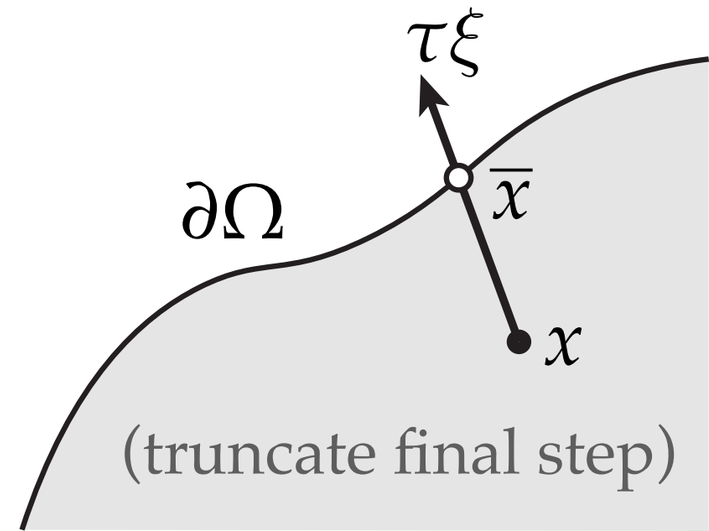
  $\widehat{u} \leftarrow \widehat{u} + g(\overline{x})$ add boundary value

return $\widehat{u}/N$ return the average

② Nothing to integrate in time

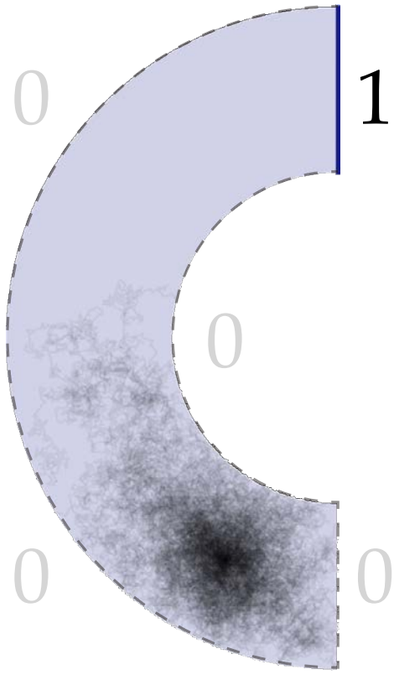
③ Overall estimate is average of boundary values at exit points

$$u(x) = \mathbb{E}[g(X_T)]$$

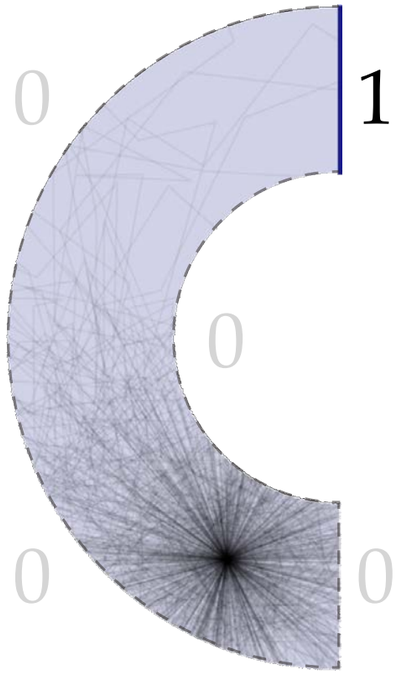


Euler-Maruyama—Numerical Challenges

- **small time steps** → accurate results, long compute times
- **large time steps** → shorter compute times, large *bias* (error)

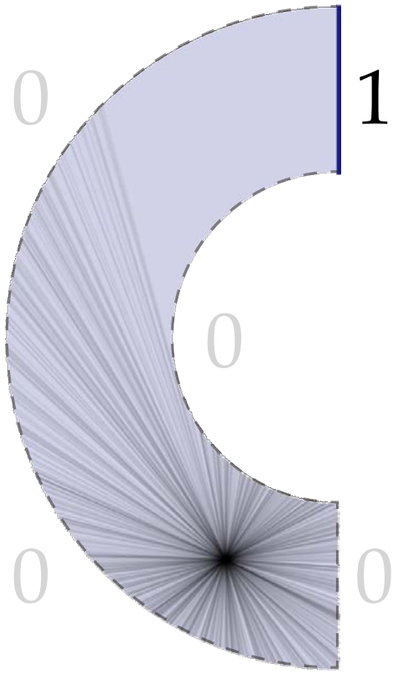


τ too small



τ just right…?

(still biased)



τ too large

Simulating Brownian Motion via Walk on Spheres

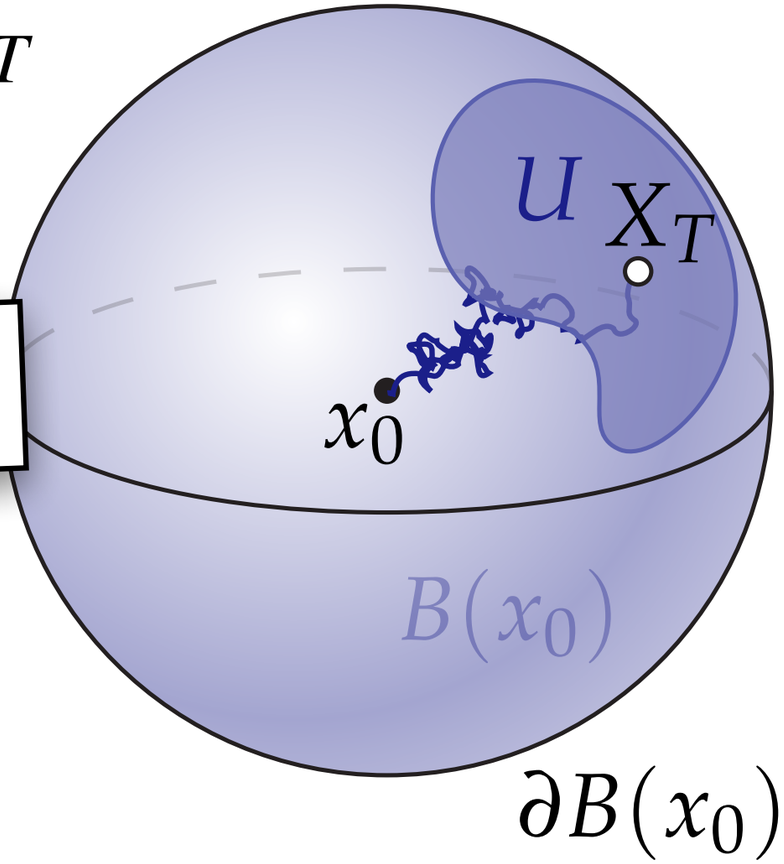
- Consider ball $B(x_0) := \{x \in \mathbb{R}^n : |x - x_0| \leq r\}$, any radius r
  - Let $\partial B(x_0) := \{x \in \mathbb{R}^n : |x - x_0| = r\}$ be the corresponding bounding sphere

$$dX_t = dW_t$$

$$X_0 = x_0$$

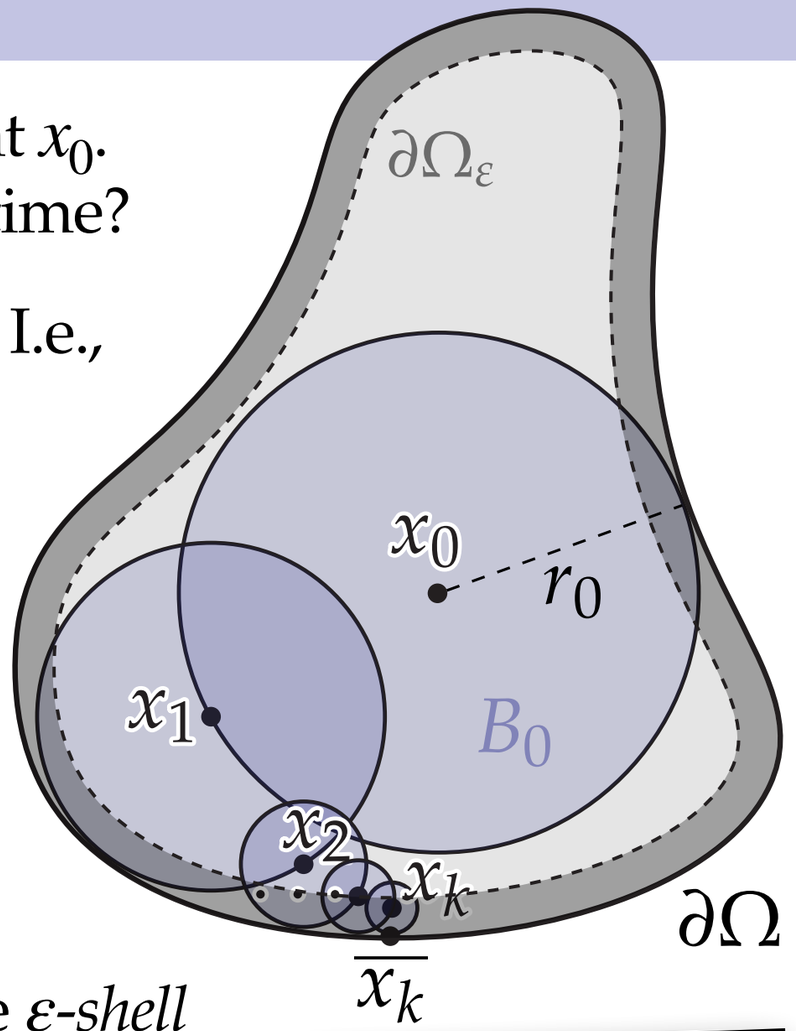
- **Claim.** Let $X_t^{x_0}$ be a Brownian process starting at x_0, and let T be the hitting time for $\partial B(x_0)$. Then X_T is random variable uniformly distributed over $\partial B(x_0)$.
- *Proof?*
  - Symmetry. ∎
- $\Longrightarrow$ Instead of simulating many steps inside ball, just uniformly sample bounding sphere.
- (**Q:** Do you remember how to uniformly sample a sphere?)

No time step τ; no bias / error due to time discretization.



Simulating Stopped Brownian Motion

- Suppose we're given a domain $\Omega \subset \mathbb{R}^n$ and a starting point x_0. How can we simulate Brownian motion up to first hitting time?
- Basic idea: iteratively apply walk on spheres starting at x_0. I.e., sample x_{k+1} from uniform distribution on $\partial B(x_k)$
  - To make quick progress, use *largest* possible radius $r_k := \min_{y \in \partial\Omega} |x_k - y|$
- **Q:** When do we stop?
  - note: the process x_k reaches $\partial\Omega$ with probability *zero!*
- **A:** Pick *stopping tolerance* $\varepsilon > 0$; stop if x_k is contained in the *ε-shell* $\Omega_\varepsilon := \{x \in \mathbb{R}^n \,|\, \mathrm{dist}(x, \partial\Omega) \leq \varepsilon\}$
  - will let $\bar{x} := \mathrm{argmin}_{y \in \partial\Omega} |x - y|$ denote closest boundary point



Note: doesn't give us the time T! Just the location X_T.

Solving Dirichlet Problem—Stochastic Perspective

- Putting these pieces together, we have:
  - **Kakutani's principle**
  - **Monte Carlo integration**
  - **walk on spheres**

tells us how to express solution as expected value of random variable

$$u(x_0) = \mathbb{E}[g(X_T^{x_0})]$$

tells us how to estimate expected value

$$u(x_0) \approx \frac{1}{N} \sum_{i=0}^{N} g((X_i^{x_0})_{T_i})$$

tells us how to simulate random walks

$$x_{k+1} \sim \mathcal{U}_{\partial B(x_k)}$$

Let's combine into a single estimator…

Solving Dirichlet Problem via Walk on Spheres

① Simulate Brownian motion using walk on spheres

$$dX_t = dW_t$$

$\widehat{u} \leftarrow 0$ initialize estimate of $u(x_0)$

for $i = 1, \ldots, N$:

  $x \leftarrow x_0$ start at evaluation point

  while $\text{dist}(x, \partial\Omega) > \varepsilon$: still in domain

    $r \leftarrow \min_{y \in \partial\Omega} |x - y|$ ball radius

    $x \leftarrow \xi \sim U_{\partial B_r(x)}$ sample ball

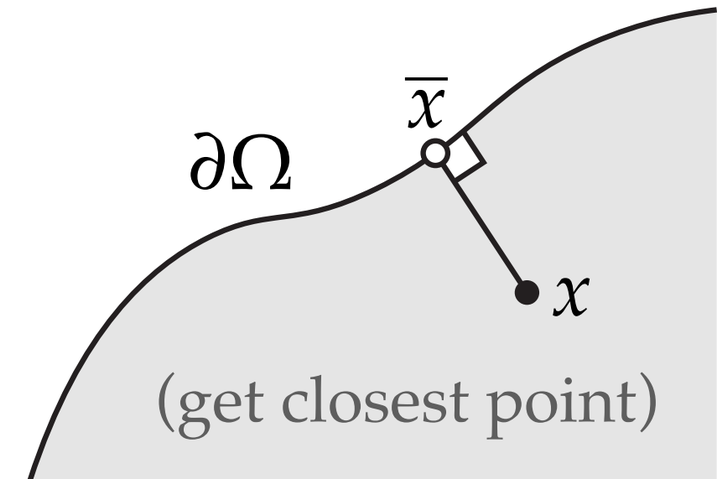
  $\widehat{u} \leftarrow \widehat{u} + g(\bar{x})$ add boundary value

return $\widehat{u}/N$ return the average

② Nothing to integrate in time

③ Overall estimate is average of boundary values at exit points

$$u(x) = \mathbb{E}[g(X_T)]$$



Closest Point Queries

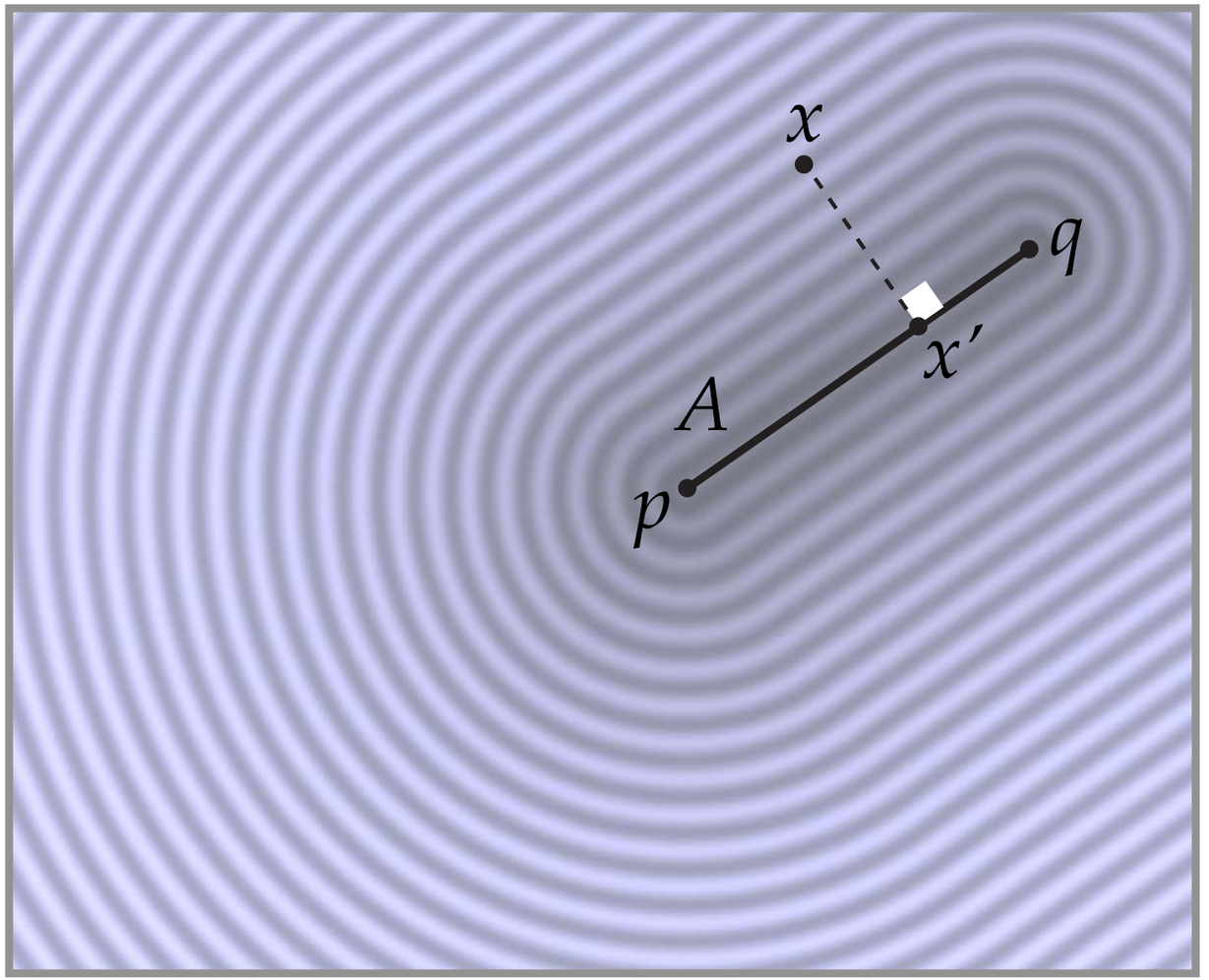
For query point x, find closest point x' on domain boundary $\partial\Omega$.

Example. Line segment ($\partial\Omega = A$)

$$t := (x - p) \cdot (q - p) / |p - q|^2$$

$$x' = \begin{cases} p, & t < 0 \\ q, & t > 0 \\ (1-t)p + tq, & \text{otherwise} \end{cases}$$

$$d(x, A) = |x - x'|$$



Closest Point Queries

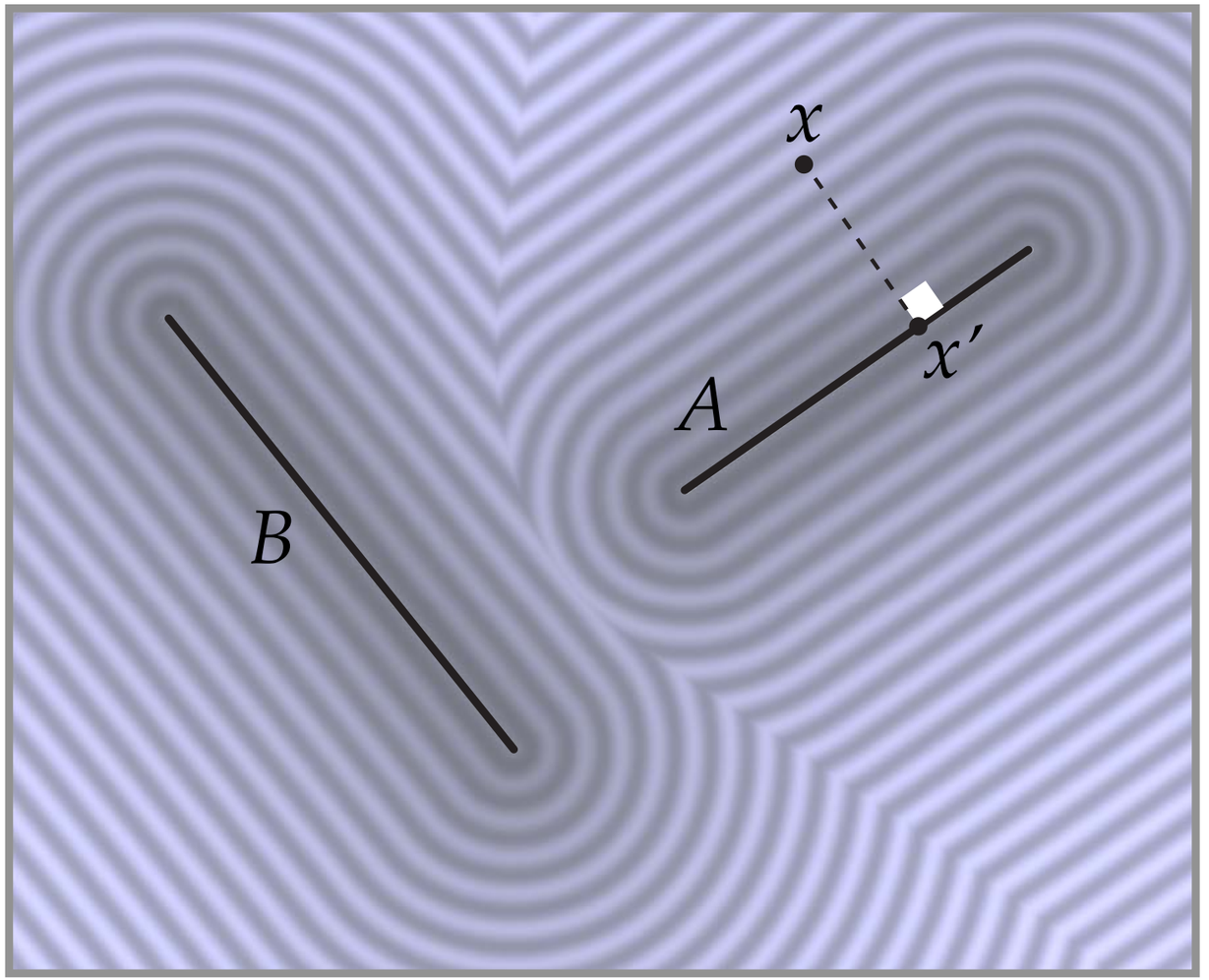
For query point x, find closest point x' on domain boundary $\partial\Omega$.

Example. Line segment ($\partial\Omega = A$)

$$d(x, A) = |x - x'|$$

Example. Two line segments ($\partial\Omega = A \cup B$)

$$d(x, A \cup B) = \min\left(d(x, A), d(x, B)\right)$$



Closest Point Queries

For query point x, find closest point x' on domain boundary $\partial\Omega$.

Example. Line segment ($\partial\Omega = A$)

$$d(x, A) = |x - x'|$$

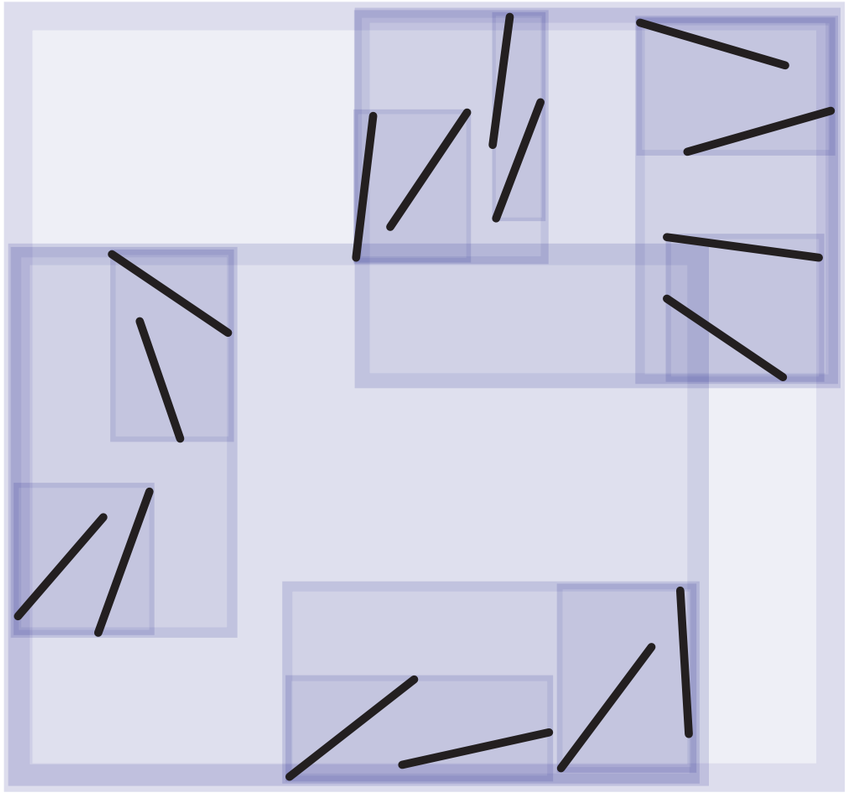
Example. Two line segments ($\partial\Omega = A \cup B$)

$$d(x, A \cup B) = \min\left(d(x, A), d(x, B)\right)$$

Example. Large number of line segments

Build *bounding volume hierarchy (BVH)*

amortized cost of one query: $O(\log n)$

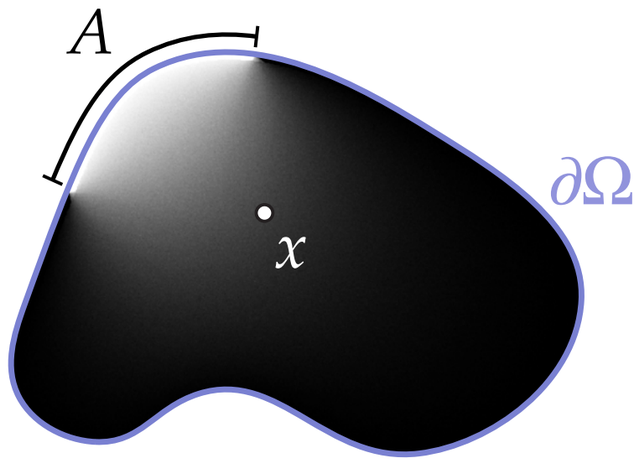


Correctness of Walk on Spheres

- Correctness of WoS for Dirichlet problem shown by Muller 1956 (*way* more than we can prove in class!):

- **Theorem.** For a domain Ω with boundary $\partial\Omega$, a walk on spheres starting at any point $x \in \Omega$ converges to a point of the boundary $\partial\Omega$ with probability 1.

- **Theorem.** For any* subset $A \subset \partial\Omega$, the probability $P_{WoS}(x_0, A)$ that a walk on spheres starting at a point $x \in \Omega$ will converge to a point in A without first hitting $\partial\Omega \backslash A$ is a harmonic function of x, and $\lim_{x \to y} P_{WoS}(x, A) = 1$ or 0 depending on whether y is contained in A or $\partial\Omega \backslash A$, respectively. The solution to the Dirichlet problem is then given by the expected boundary value at the end of the walk on spheres.



In short: the random process simulated by WoS has exactly the same statistics as continuous (stopped) Brownian motion.

*"elementary"

Efficiency of Walk on Spheres

Efficiency: what's the expected number of steps to reach boundary?

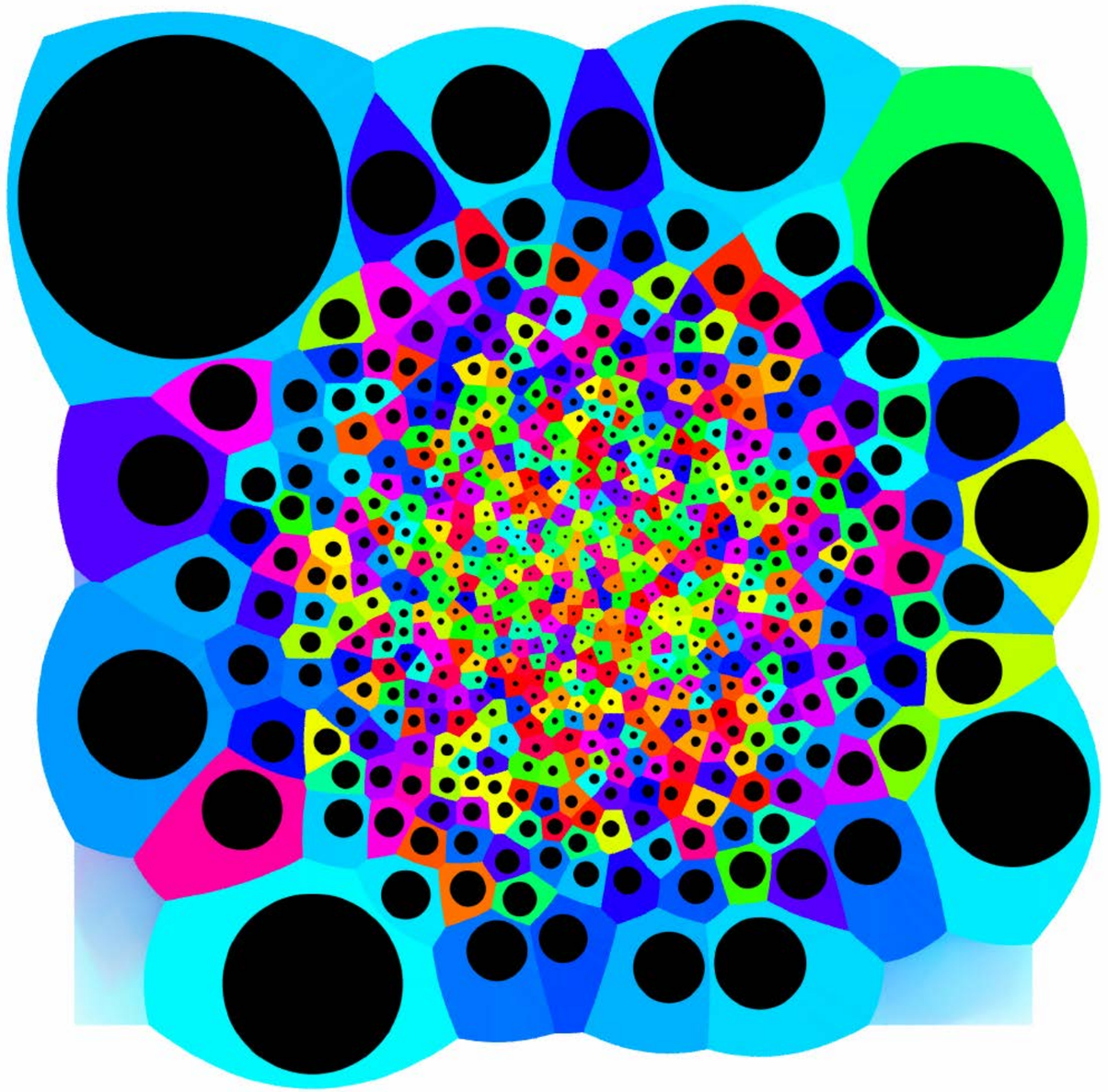
Theorem. If the domain **boundary $\partial\Omega$ is smooth**, *or* the **domain Ω is convex**, then WoS reaches the boundary in $O(\log 1/\varepsilon)$ steps, on average.

Alternatively: for **any 2D domain**, or for any **bounded 3D domain with connected complement** WoS reaches boundary in $O((\log 1/\varepsilon)^2)$ steps.

Either way: pretty darn fast.
(Compare to Euler-Maruyama…)

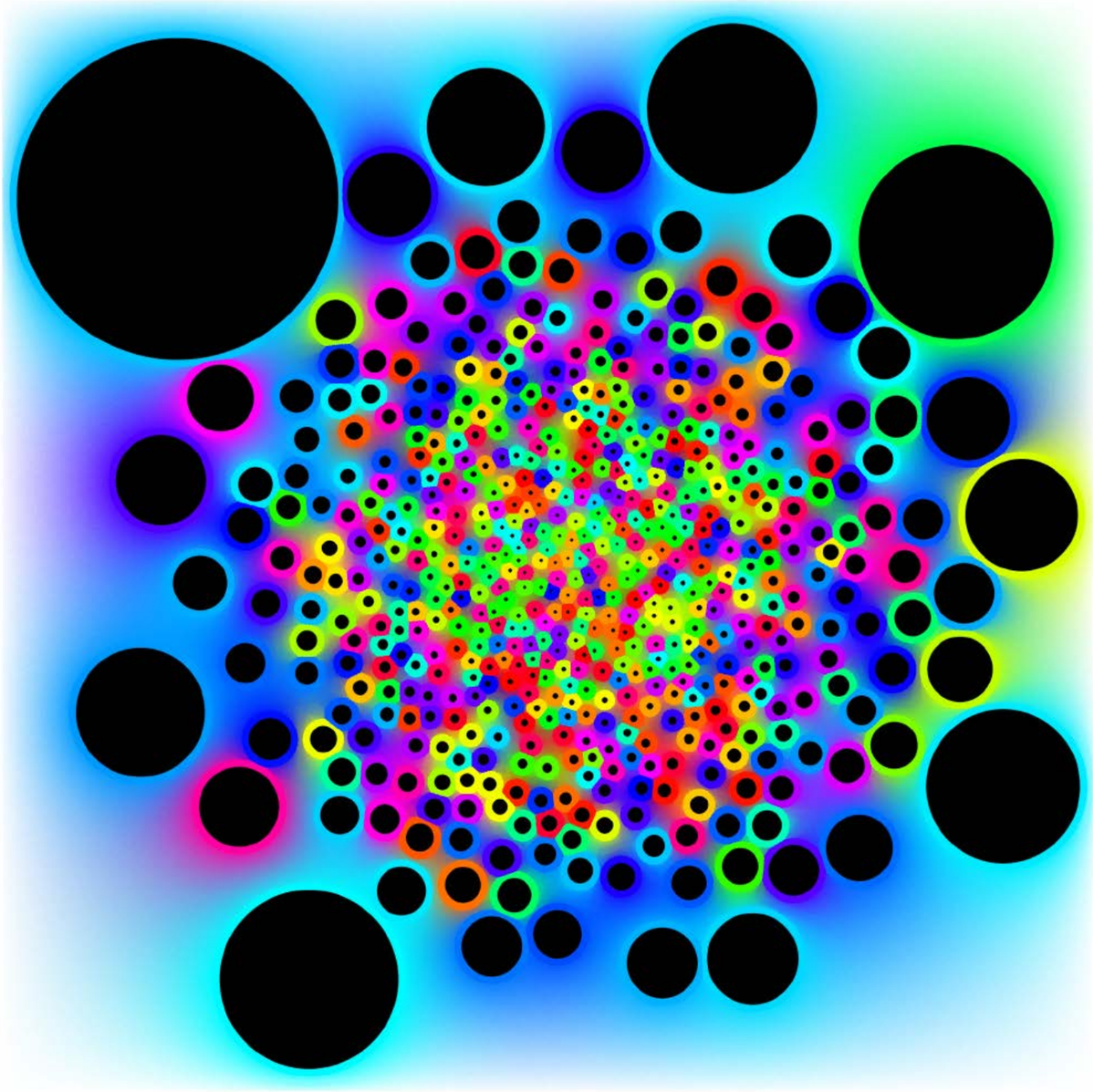
Binder & Braverman, *"The Rate of Convergence of the Walk on Spheres Algorithm"* (2012)

Effect of Stopping Tolerance ε



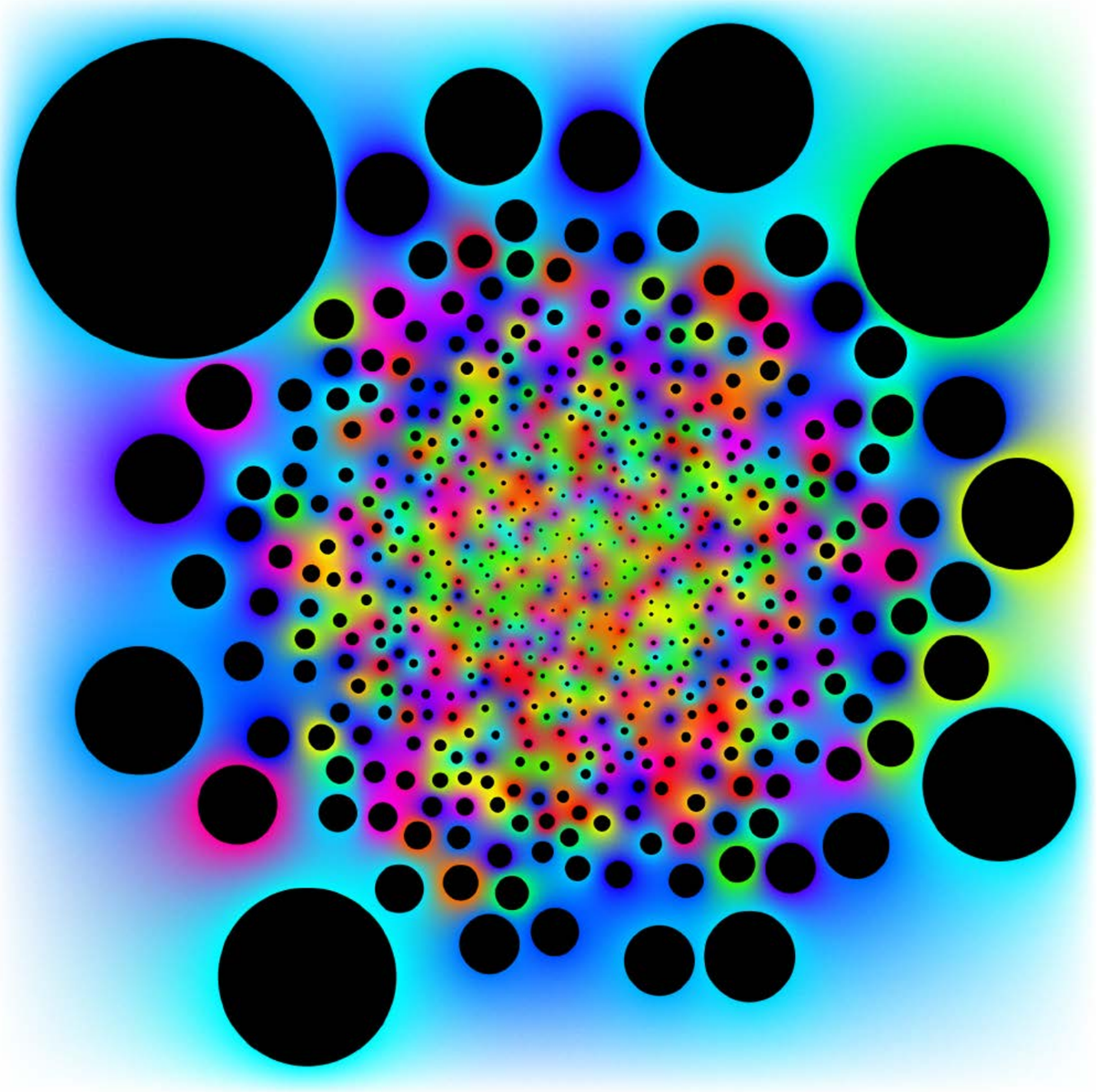
$\varepsilon = 10^{-1}$
steps/walk: 1.00

Effect of Stopping Tolerance ε



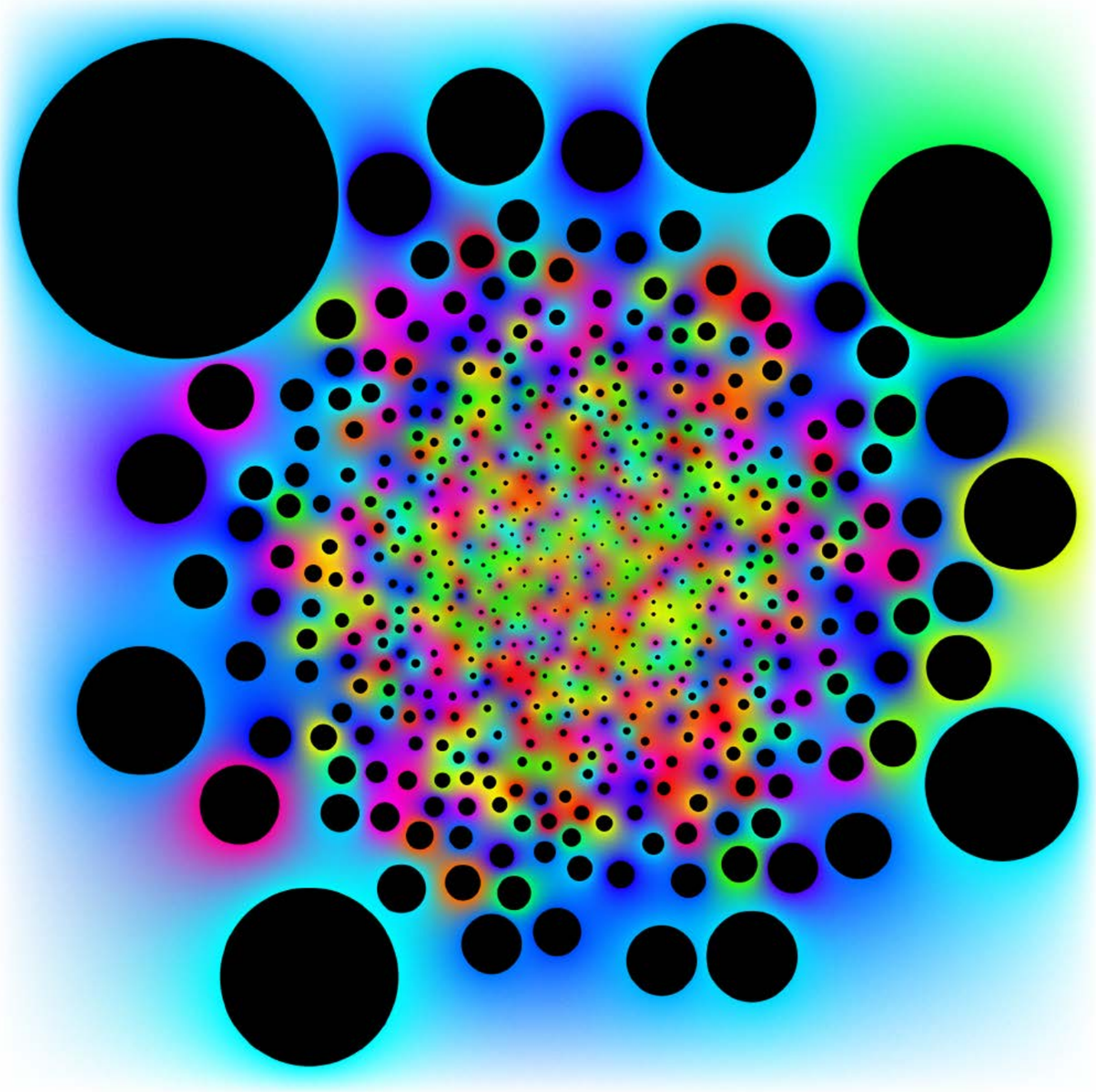
$\varepsilon = 10^{-2}$
steps/walk: 1.05

Effect of Stopping Tolerance ε



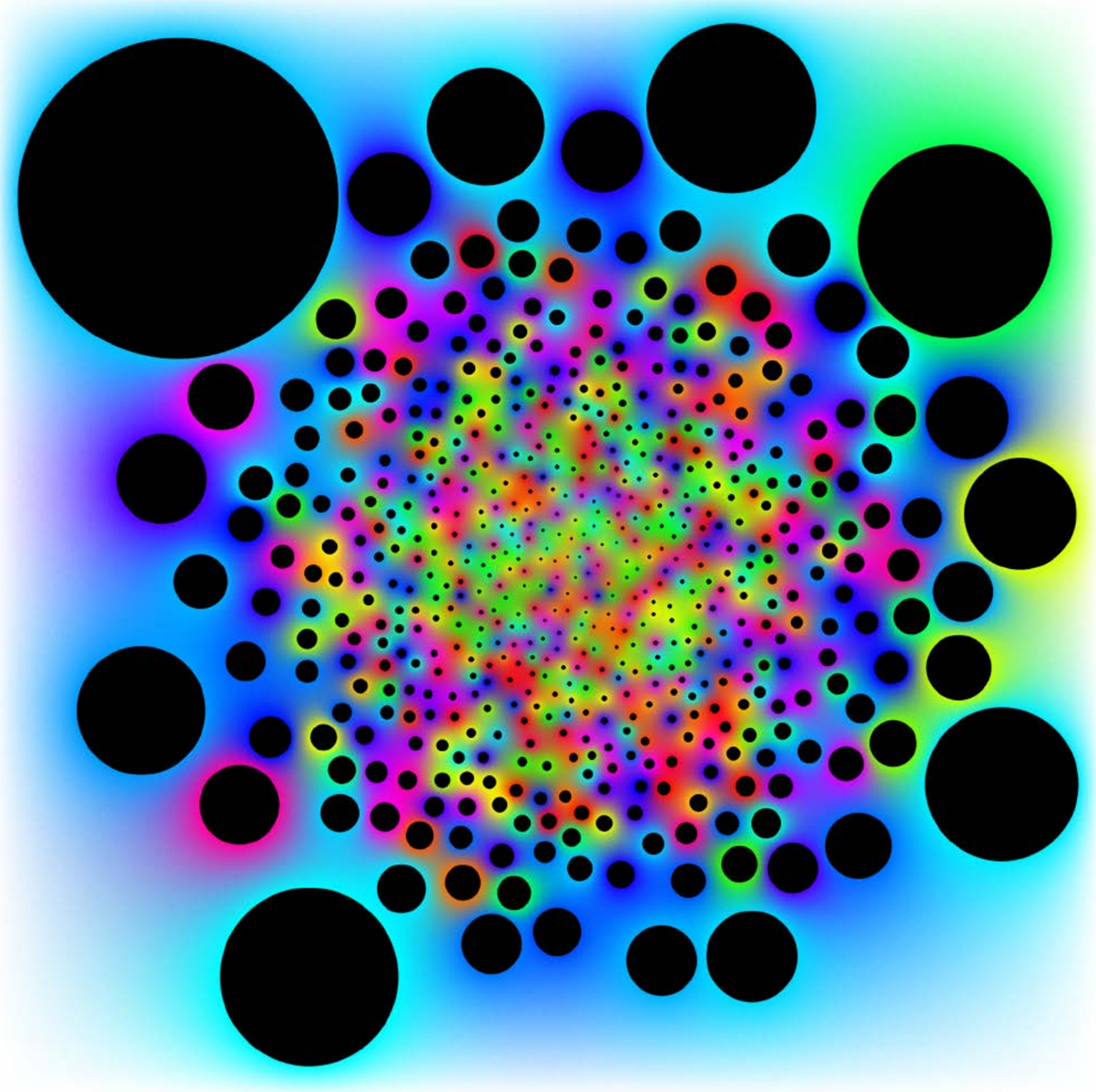
$\varepsilon = 10^{-3}$
steps/walk: 3.08

Effect of Stopping Tolerance ε



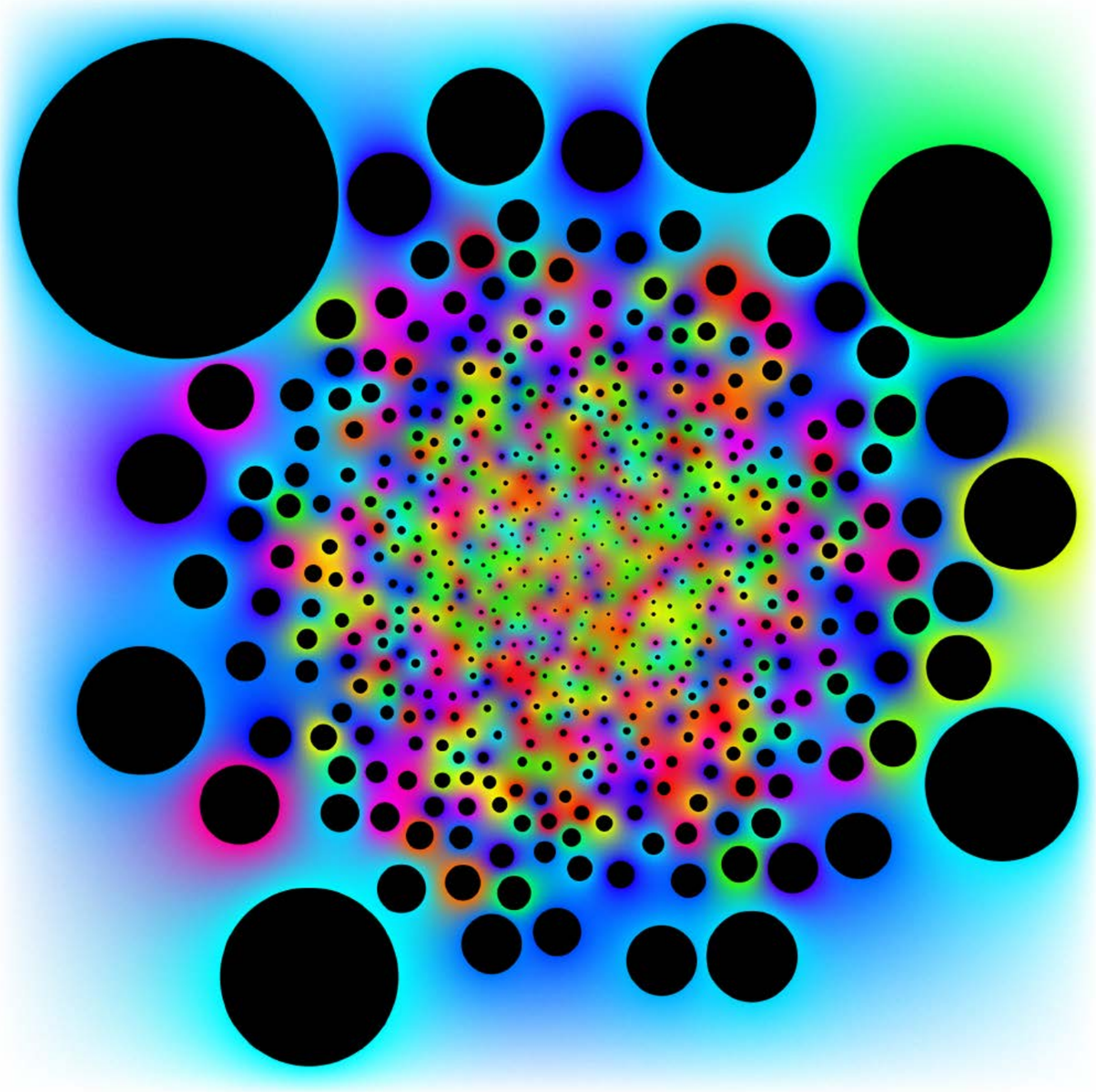
$\varepsilon = 10^{-4}$
steps/walk: 10.5

Effect of Stopping Tolerance ε



$\varepsilon = 10^{-5}$
steps/walk: 13.8

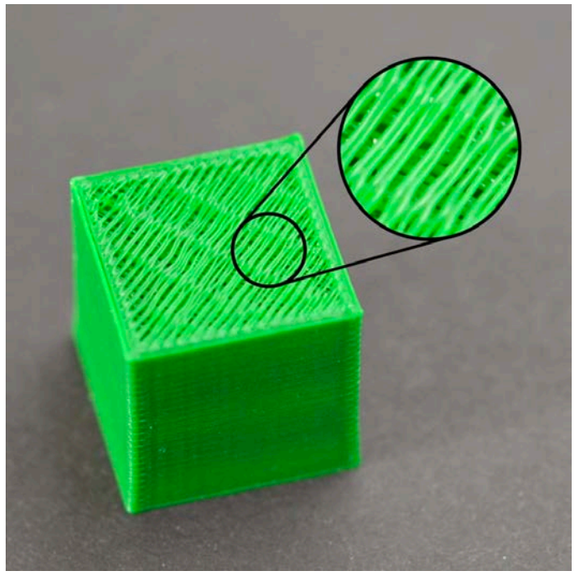
Effect of Stopping Tolerance ε

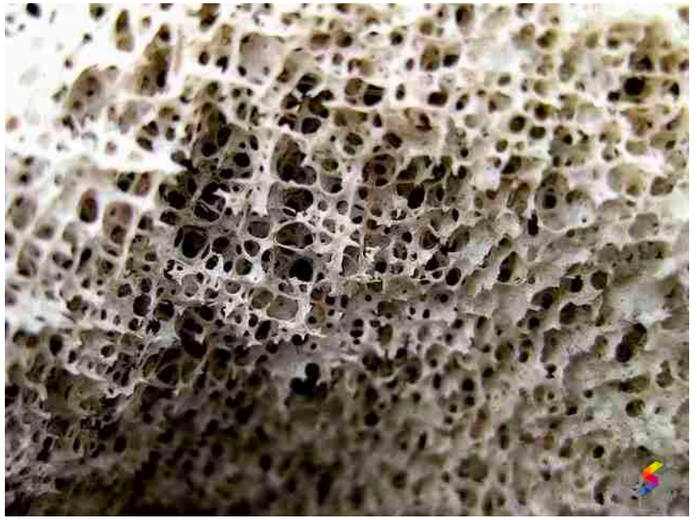


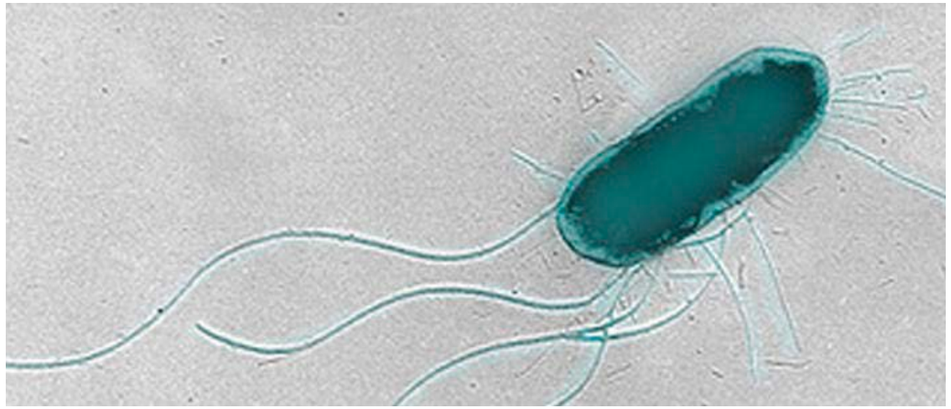
$\varepsilon = 10^{-6}$
steps/walk: 17.2

Discussion: Stochastic vs. Deterministic Methods

How much does it cost to capture fine-scale features, of size $O(\varepsilon)$?

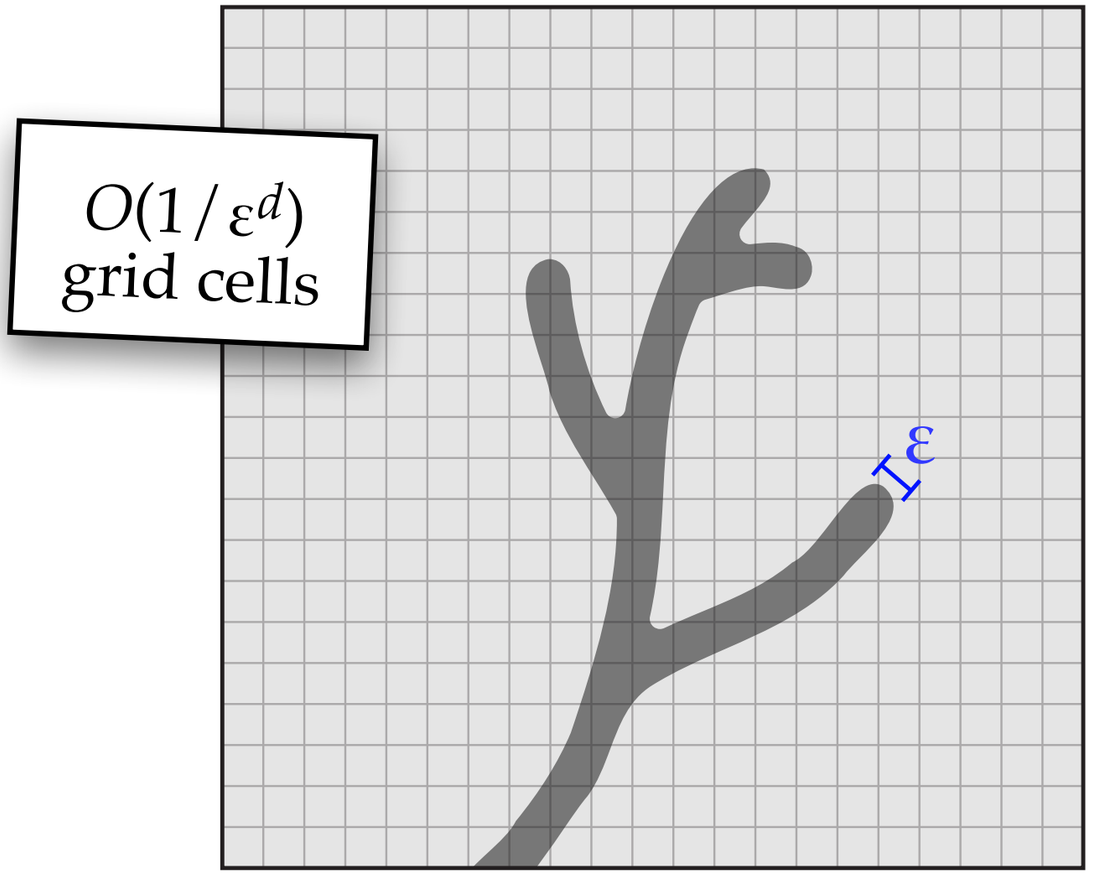




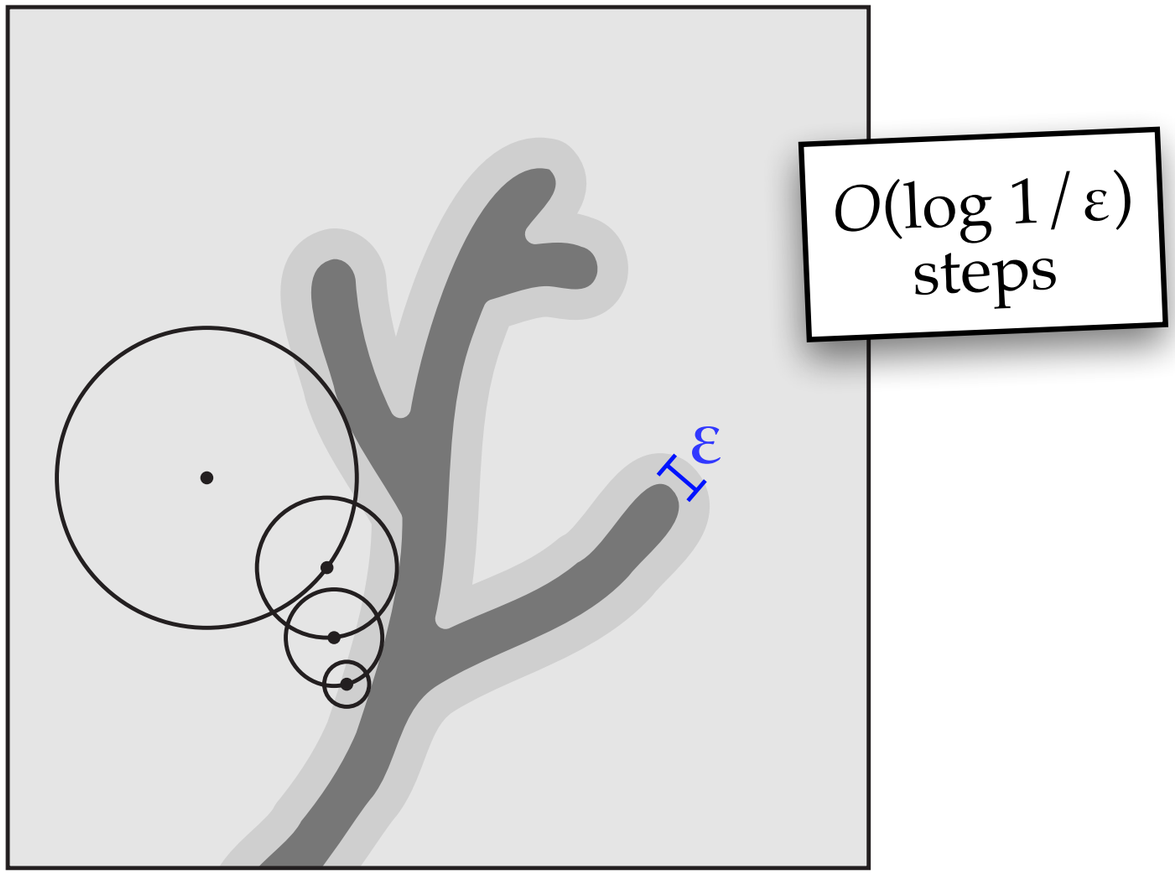


Discussion: Stochastic vs. Deterministic Methods

How much does it cost to capture fine-scale features, of size $O(\varepsilon)$?



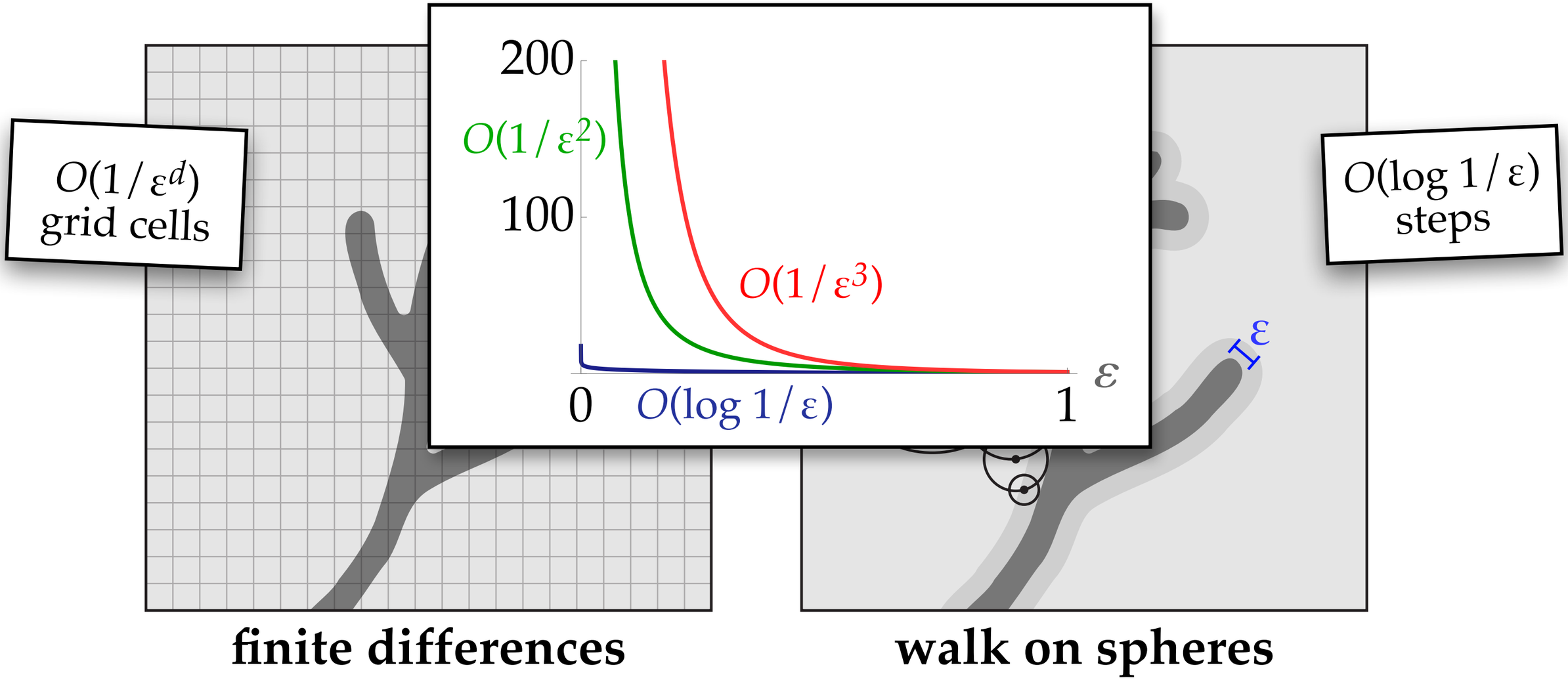
finite differences



walk on spheres

Discussion: Stochastic vs. Deterministic Methods

How much does it cost to capture fine-scale features, of size $O(\varepsilon)$?



finite differences **walk on spheres**

Let’s revisit this whole story
through the lens of *potential theory*…

Notation—Mean Integral

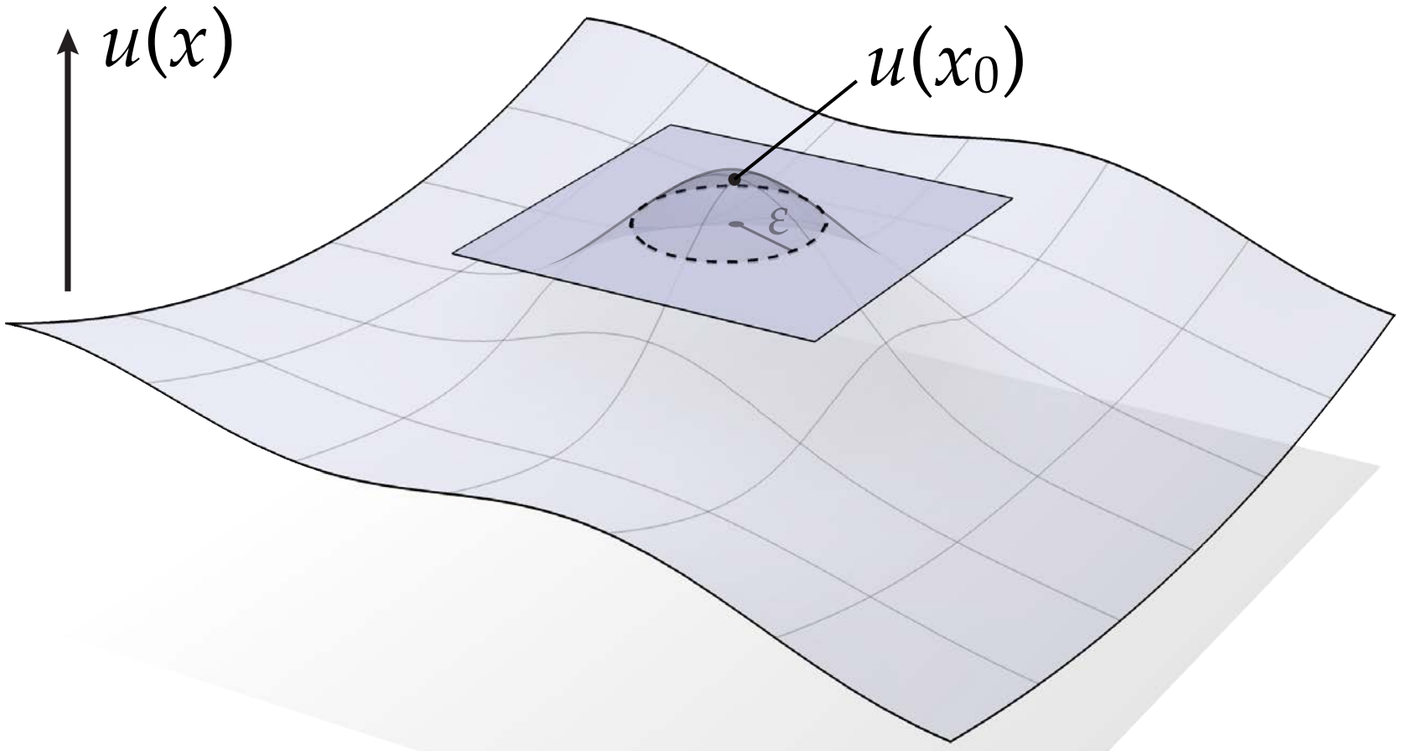
For brevity, will define some notation for the **mean** value of a function f over a region A:

$$\fint_A f(x)\,dx := \frac{1}{|A|}\int_A f(x)\,dx$$

(Won't use this notation at first, but will help shorten expressions later…)

Laplacian Gives Deviation from Local Average

Recall: can think of the Laplacian of a function u as difference between value at a point x_0, and the average value over a small sphere (or ball) around x_0.



$$\Delta u(x_0) \propto \lim_{\varepsilon \to 0} \frac{1}{\varepsilon^2} \left(\frac{1}{|S_\varepsilon(x_0)|} \int_{S_\varepsilon(x_0)} u(x)\, dx - u(x_0) \right)$$

sphere area — integral over sphere — value at center

Mean Value Property of Harmonic Functions

- **Q:** Given this interpretation of the Laplacian, what can we say about the behavior of a *harmonic function* (i.e., a function u satisfying $\Delta u = 0$?)
- **A:** Value at center and average over a ball/sphere are *equal*.

MEAN VALUE PROPERTY
(sphere)

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

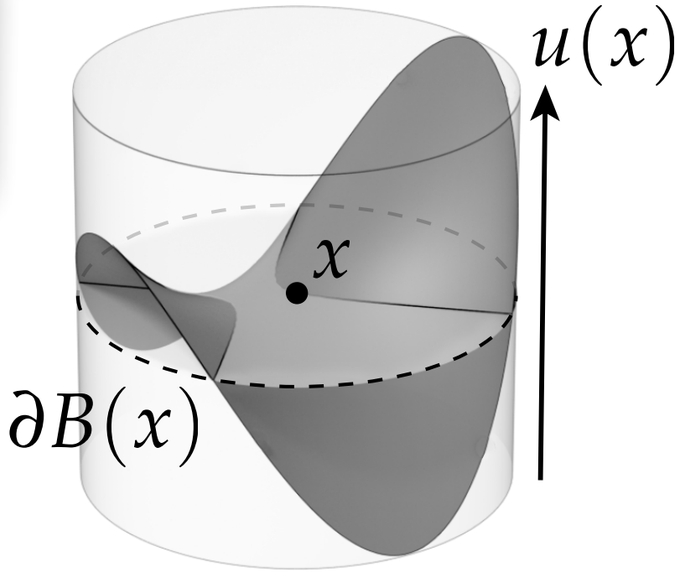
Not just an approximation: holds exactly for ball $B \subset \Omega$ of *any* radius

MEAN VALUE PROPERTY
(ball)

$$u(x) = \frac{1}{|B(x)|} \int_{B(x)} u(x)\, dx$$

Q: Why might the latter follow from the former?

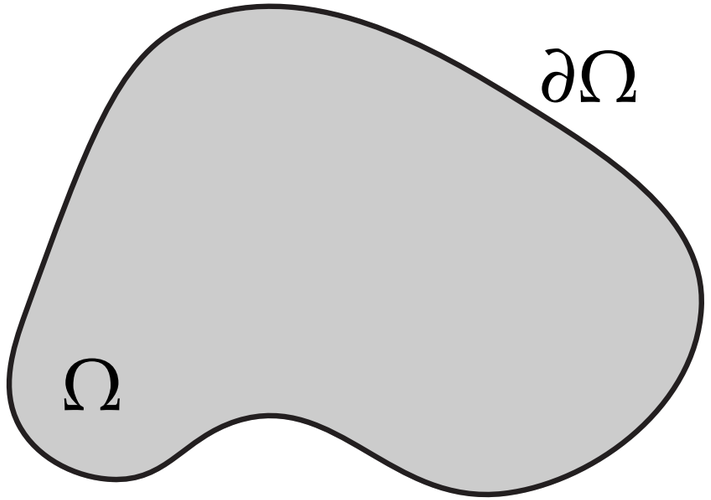
A: Can express integral over ball as integral over "nested" spheres.



Walk on Spheres via Potential Theory

- Let's return to our Dirichlet problem.
- Notice: the mean value property immediately tells us how to express the solution at any point as an **integral.**
- **Q:** What algorithm can you use to approximate integrals? 😄
- **A:** *Monte Carlo!*
- **Q:** How do you evaluate summands?
- **A:** *...Monte Carlo!*
- Leads us back to walk on spheres…

$$\begin{aligned} \Delta u &= 0 \quad \text{on} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$



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

Solving Dirichlet Problem—Potential Theory

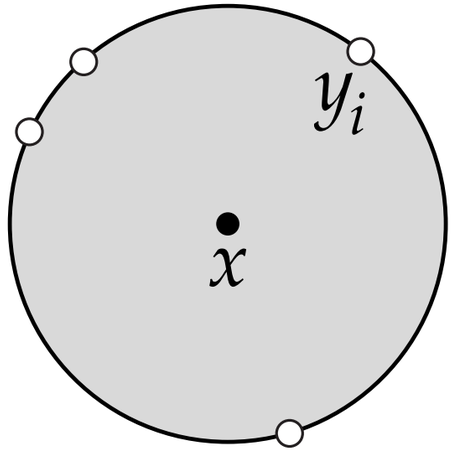
Monte Carlo estimator

$$\widehat{u}(x) = \frac{1}{N}\sum_{i=1}^{N} \widehat{u}(y_i),\ y_i \sim \mathcal{U}_{\partial B(x)}$$

uniform distribution on sphere

$$\mathbb{E}[\widehat{u}(x)] = u(x)$$

N=1



MEAN VALUE PROPERTY

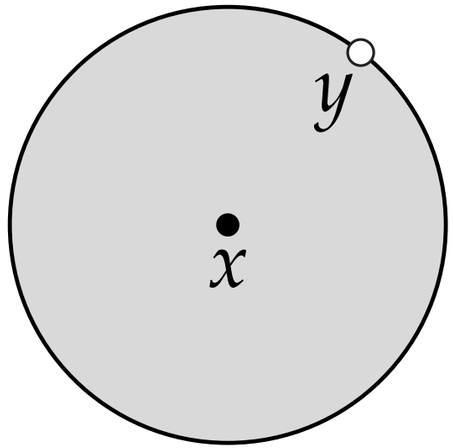
$$\frac{1}{|\partial B(x)|}\int_{\partial B(x)} u(y)dy$$

estimate via Monte Carlo

Solving Dirichlet Problem—Potential Theory

Monte Carlo estimator

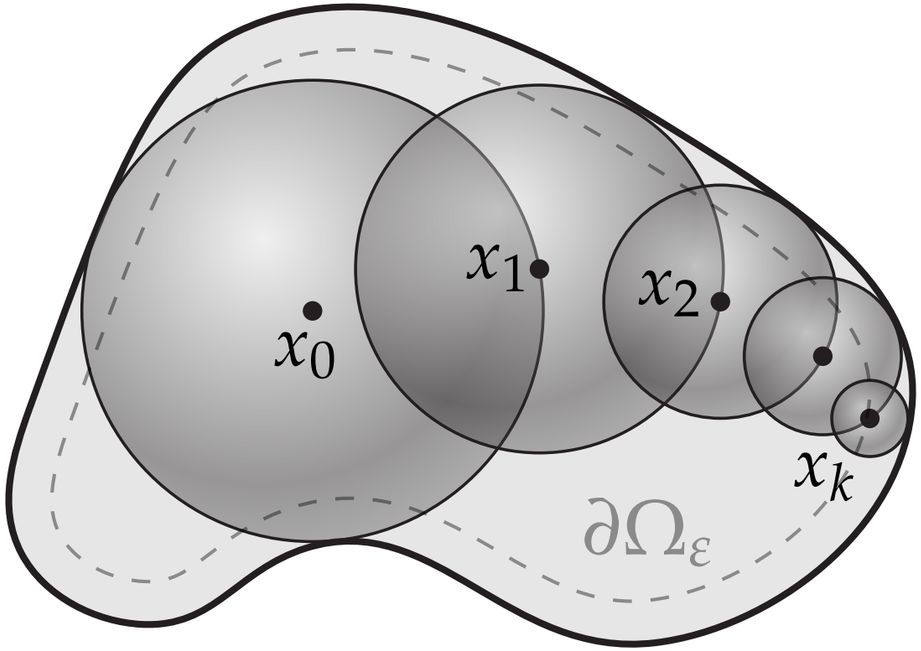
$$\widehat{u}(x) = \widehat{u}(y),\ y \sim \mathcal{U}_{\partial B(x)}$$



MEAN VALUE PROPERTY

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

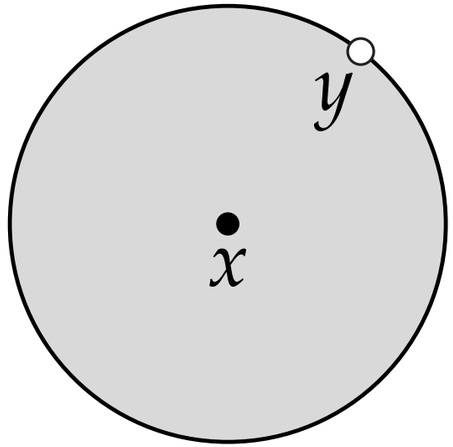
estimate via Monte Carlo



Solving Dirichlet Problem—Potential Theory

Monte Carlo estimator

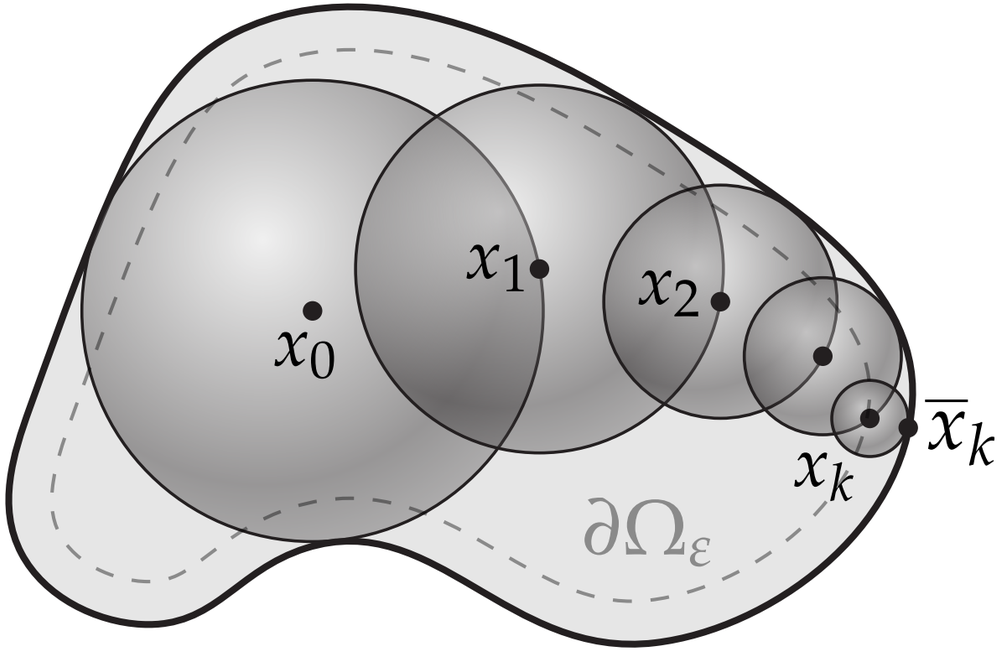
$$\widehat{u}(x) = \begin{cases} g(\overline{x}), & x \in \partial\Omega_\varepsilon \\ \widehat{u}(y), & \text{otherwise} \end{cases}$$



MEAN VALUE PROPERTY

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

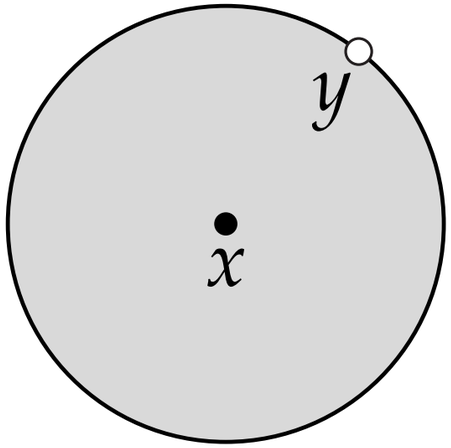
estimate via Monte Carlo



Solving Dirichlet Problem—Potential Theory

Monte Carlo estimator

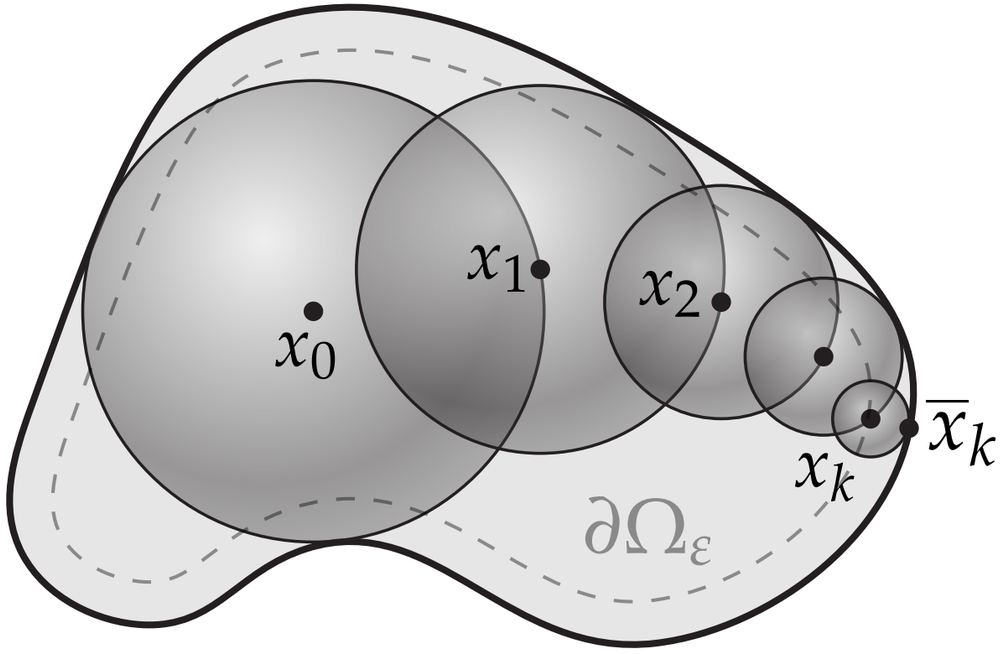
$$\widehat{u}(x) = \begin{cases} g(\overline{x}), & x \in \partial\Omega_\varepsilon \\ \widehat{u}(y), & \text{otherwise} \end{cases}$$



MEAN VALUE PROPERTY

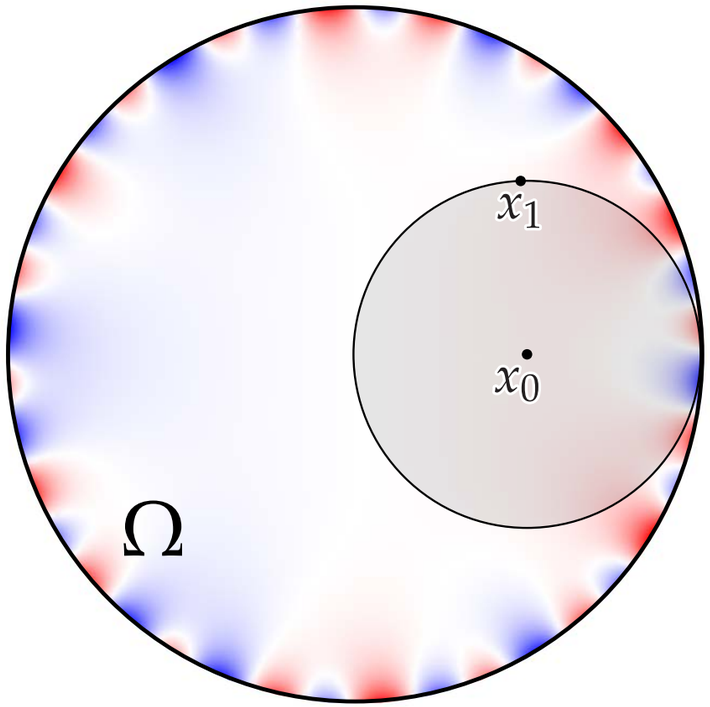
$$\frac{1}{|\partial B(x)|}\int_{\partial B(x)} u(y)dy$$

estimate via Monte Carlo



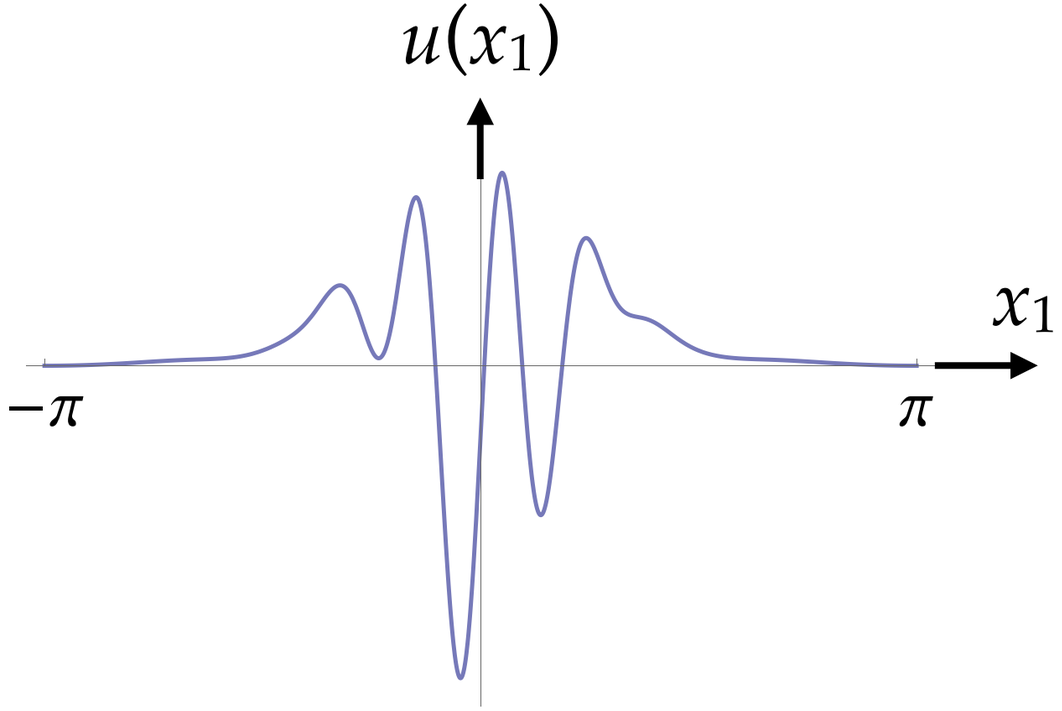

```
u = 0 // solution estimate
for i=1,..,nWalks {
   x = x0 // start a new walk
   do {
      // move to random point on biggest empty sphere
      r = distance(x,∂Ω)
      x = randomSphere(x,r)
   } while(r > ε) // close enough!
   u += g(closestPoint(x,∂Ω)) // sample boundary value
}
return u/nWalks // return average boundary value
```

Dirichlet Problem as Recursive Integral Equation

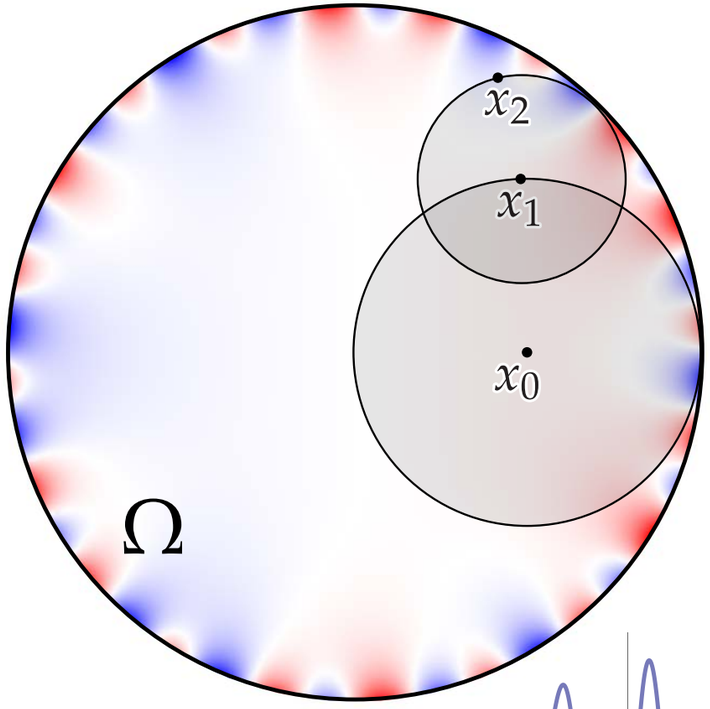


$$u(x_0) = \fint_{\partial B(x_0)} u(x_1)\, dx_1$$

mean value



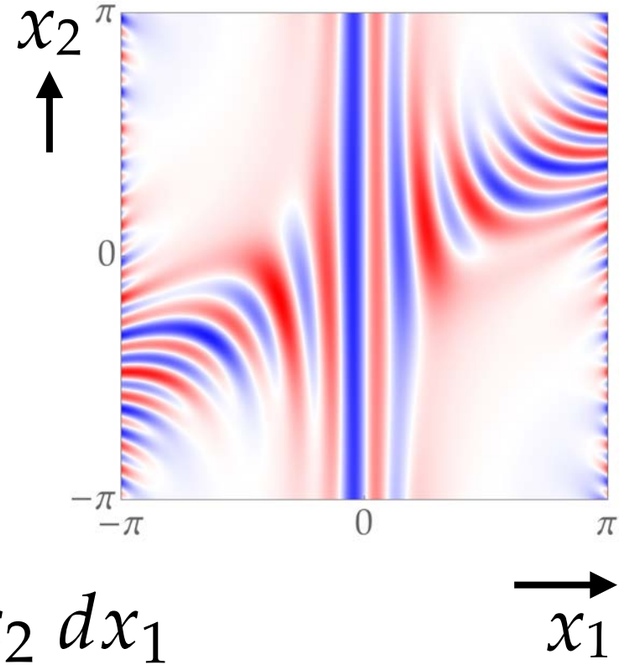
Dirichlet Problem as Recursive Integral Equation

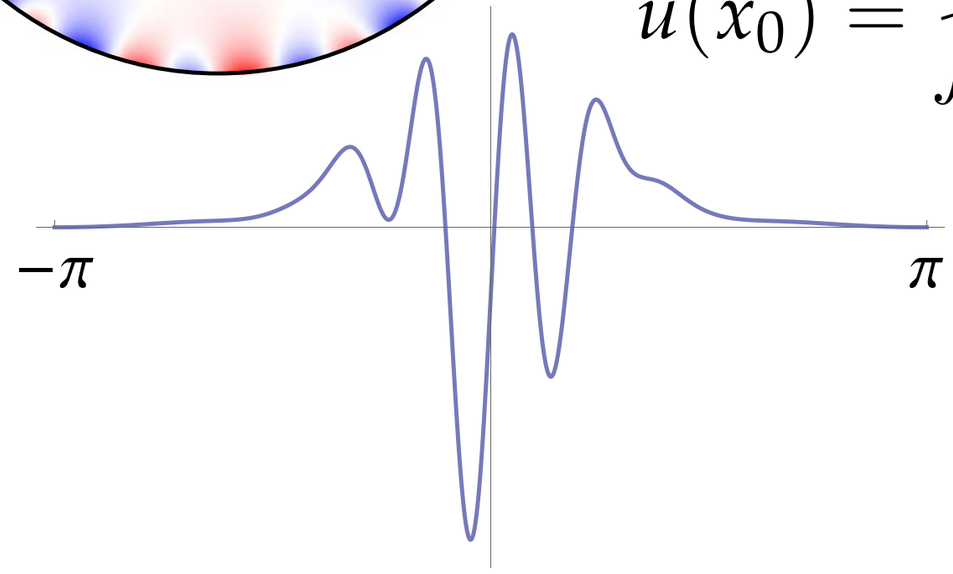


$$u(x_0) = \fint_{\partial B(x_0)} u(x_1)\, dx_1$$

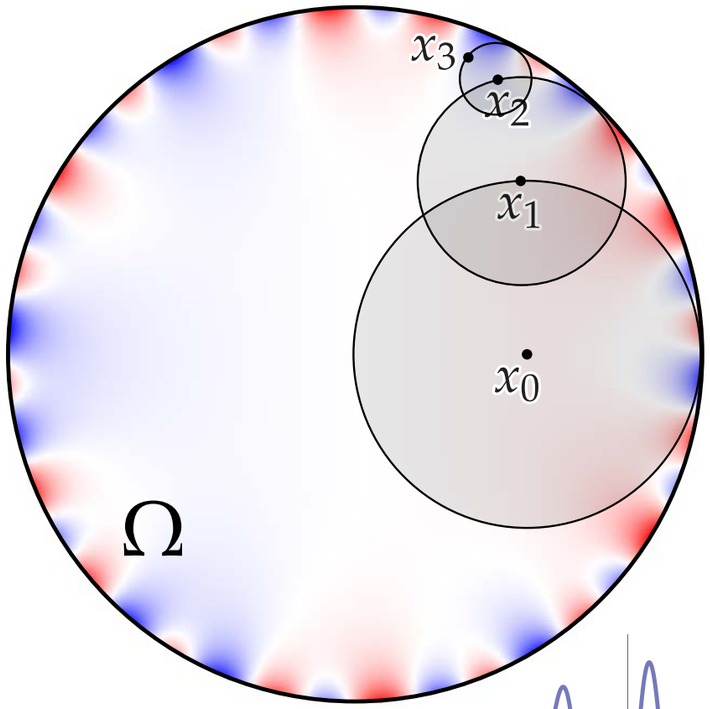
$$u(x_1) = \fint_{\partial B(x_1)} u(x_2)\, dx_2$$

$$u(x_0) = \fint_{\partial B(x_0)} \fint_{\partial B(x_1)} u(x_2)\, dx_2\, dx_1$$



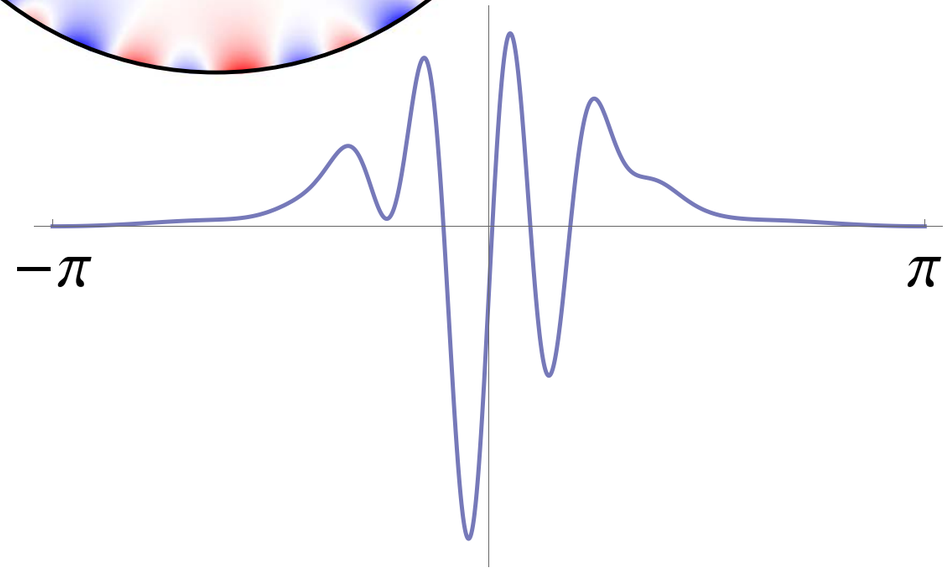


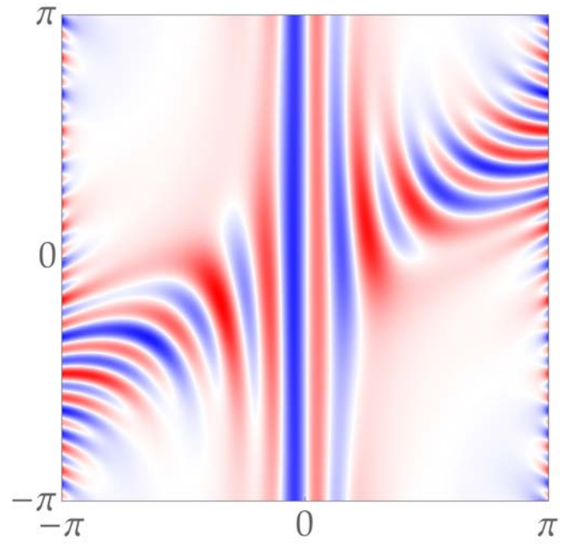
Dirichlet Problem as Recursive Integral Equation



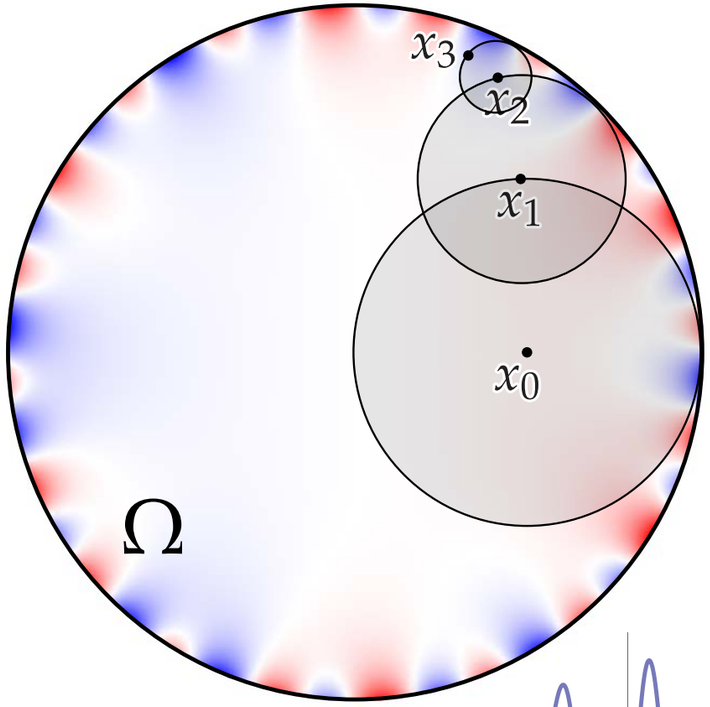
$$u(x_0) = \fint_{\partial B(x_0)} \fint_{\partial B(x_1)} u(x_2)\, dx_2\, dx_1$$

$$u(x_2) = \fint_{\partial B(x_2)} u(x_3)\, dx_3$$



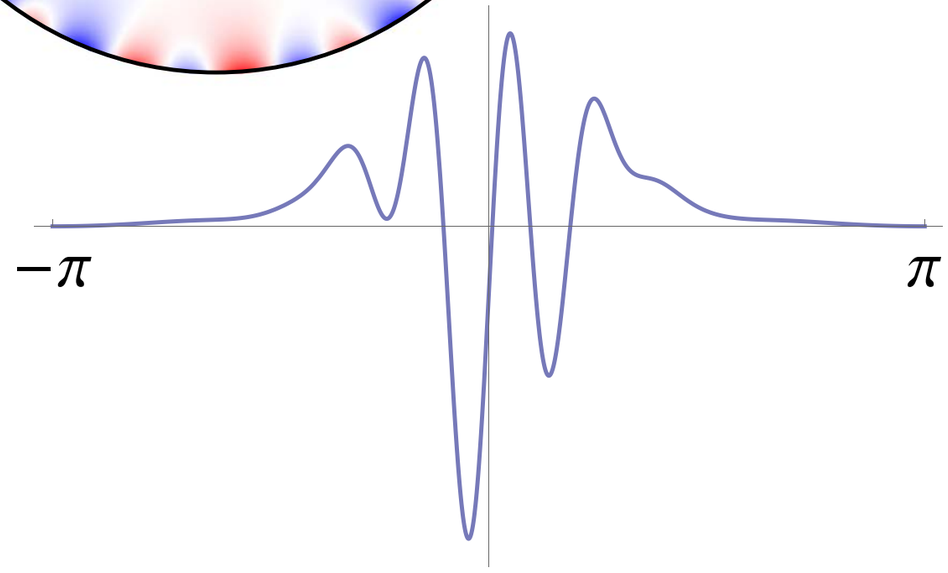


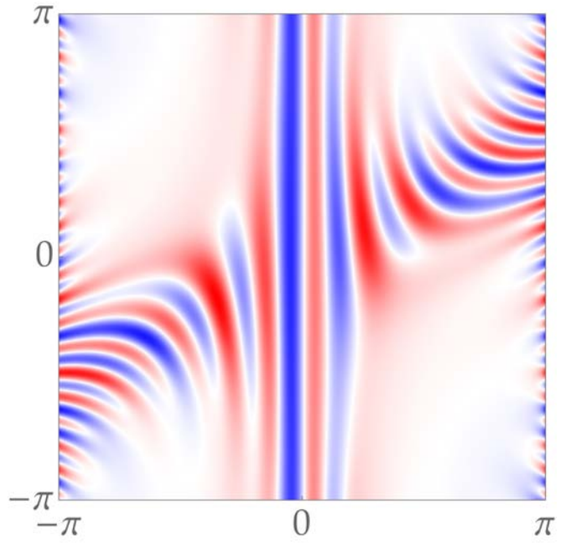
Dirichlet Problem as Recursive Integral Equation

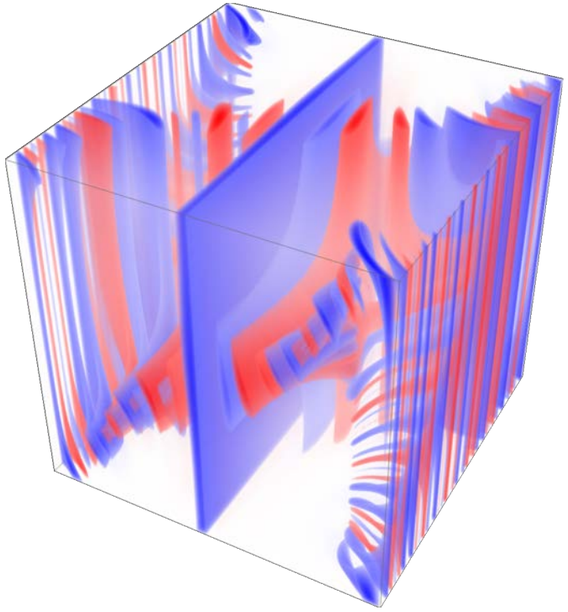


$$u(x_0) = \fint_{\partial B(x_0)} \fint_{\partial B(x_1)} \fint_{\partial B(x_2)} u(x_3)\, dx_3\, dx_2\, dx_1$$

One perspective: approximating an integral over an "infinite-dimensional" domain.





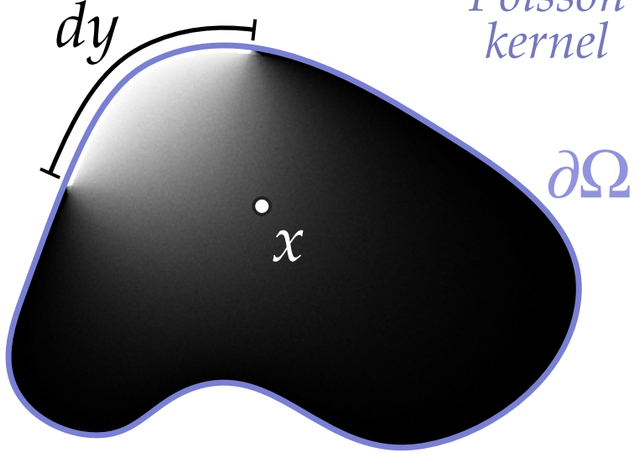


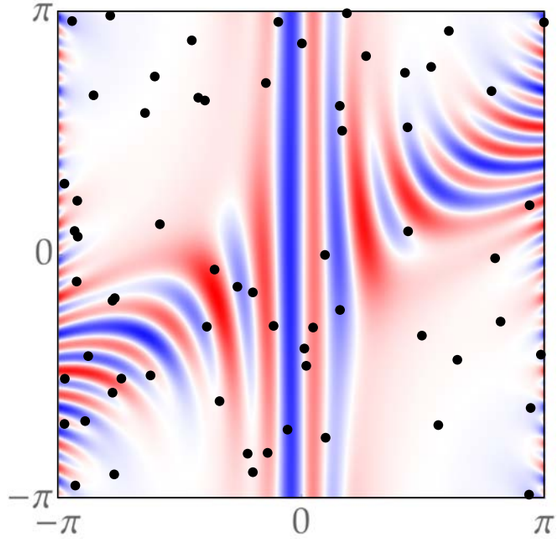
Dirichlet Problem as Recursive Integral Equation

- Walk on spheres algorithm is equivalent to Monte Carlo estimator for "infinite dimensional" integral
  - truncated to a finite number of steps by ε-shell
- …Didn't we already have a *low*-dimensional integral?
  - Huge "but": need to know harmonic measure P (a.k.a. *Poisson kernel*)
  - not known for most domains
  - estimating P <u>exactly</u> as hard as solving our "infinite-dimensional" integration problem!
- Food for thought: how can we *importance sample?*
  - Basic WoS corresponds to uniform sampling
  - Lots of other strategies—including MCMC!

$$u(x) = \int_{\partial\Omega} P(x, dy) g(y)$$

Poisson kernel





Warning About Convergence

- **Q:** What assumptions did we make in order to guarantee that basic Monte Carlo integration converges to the true integral?
- **A:** Not much: the integrand has to be a function (not a more general distribution), and the integral has to exist.
- **Q:** What if we now have arbitrarily many nested integrals?
- **A:** Still just need to make sure that the solution exists…

basic Monte Carlo

$$I = \int_\Omega f(x)\, dx$$

$$\widehat{I}_N = \frac{|\Omega|}{N} \sum_{i=1}^{N} f(X_i)$$

$$u(x_k) = \fint_{\partial B(x_k)} u(x_{k+1})\, dx_{k+1}$$

$$u(x_0) = \lim_{k\to\infty} \fint_{\partial B(x_0)} \fint_{\partial B(x_1)} \cdots \fint_{\partial B(x_k)} u(x_k)\, dx_k \cdots dx_2 dx_1$$

Convergence of WoS—Integral Viewpoint

- Stochastic perspective on WoS relied on Kakutani's theorem, analysis of (discrete) random walks
- Can also reason about correctness of WoS in terms of *integral operators*
  - map from functions to functions via integration
  - abstractly, just a linear map L (not *so* different from matrix multiplication…)

– **Q:** For a matrix $A \in \mathbb{R}^{n \times n}$, vector $x \in \mathbb{R}^n$ how do we check whether $\lim_{k \to \infty} A^k x$ exists?

– **A:** Just check that A cannot make any vector "bigger" at each step, *i.e.*, $\|Ax\|/\|x\| < 1$ for all x

– For integral operators, consider the *operator norm*

To really get this right, have to consider ε-shell…

integral operator

$$Lu(x) := \fint_{\partial B(x)} u(y)\, dy$$

$$L : (\text{functions}) \to (\text{functions})$$

operator norm

$$\|L\|_{\text{op}} := \inf\{c \geq 0 : \|Lu\|_1 \leq c\|u\|_1, \forall u\}$$

$$\|u\|_1 := \int_\Omega |u(x)|\, dx$$

L^1 norm

$$\boxed{\|L\|_{\text{op}} < 1}$$

(convergence criterion)

Review: Geometric Series

A **geometric series** is a sum of the form

$$b + ab + a^2b + a^3b + a^4b + \cdots$$

b — initial value

a — ratio between consecutive values

Fact. When $|a| < 1$, geometric series converges to

$$\lim_{n\to\infty} \sum_{k=0}^{n} a^k b = \frac{b}{1-a}$$

Fact. For a vector $b \in \mathbb{R}^d$ and matrix $A \in \mathbb{R}^{d\times d}$ with $\|A\|_{\text{op}} < 1$

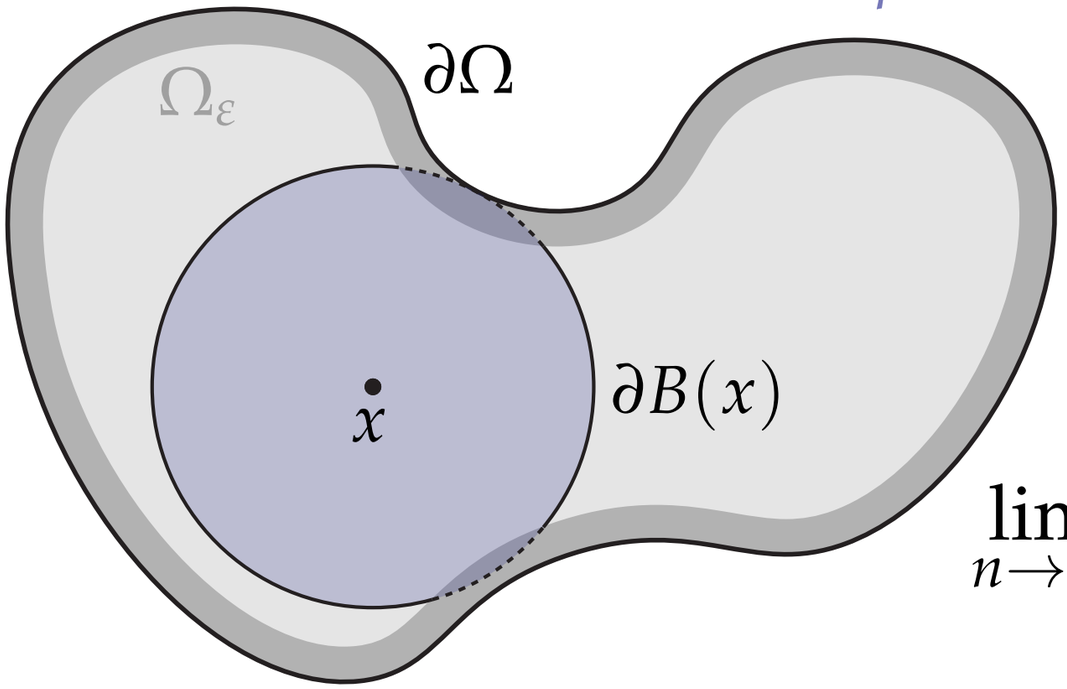
$$\lim_{n\to\infty} \sum_{k=0}^{n} A^k b = (I - A)^{-1} b$$

Basically same story with integral operators.

Convergence of WoS—Integral Viewpoint

Integral operator L for walk on spheres isn't *quite* just the "averaging operator"—must also account for contribution of Dirichlet boundary values from ε-shell:

$$Lu(x) := \underbrace{\frac{1}{|\partial B(x)|}\int_{\partial B(x)\setminus\Omega_\varepsilon} u(y)\,dy}_{\text{linear operator } L_\varepsilon u} + \underbrace{\frac{1}{|\partial B(x)|}\int_{\partial B(x)\cap\Omega_\varepsilon} g(y)\,dy}_{\text{constant term } b}$$



$$Lu = L_\varepsilon u + b$$

$$L^2 u = L_\varepsilon^2 u + L_\varepsilon b + b$$

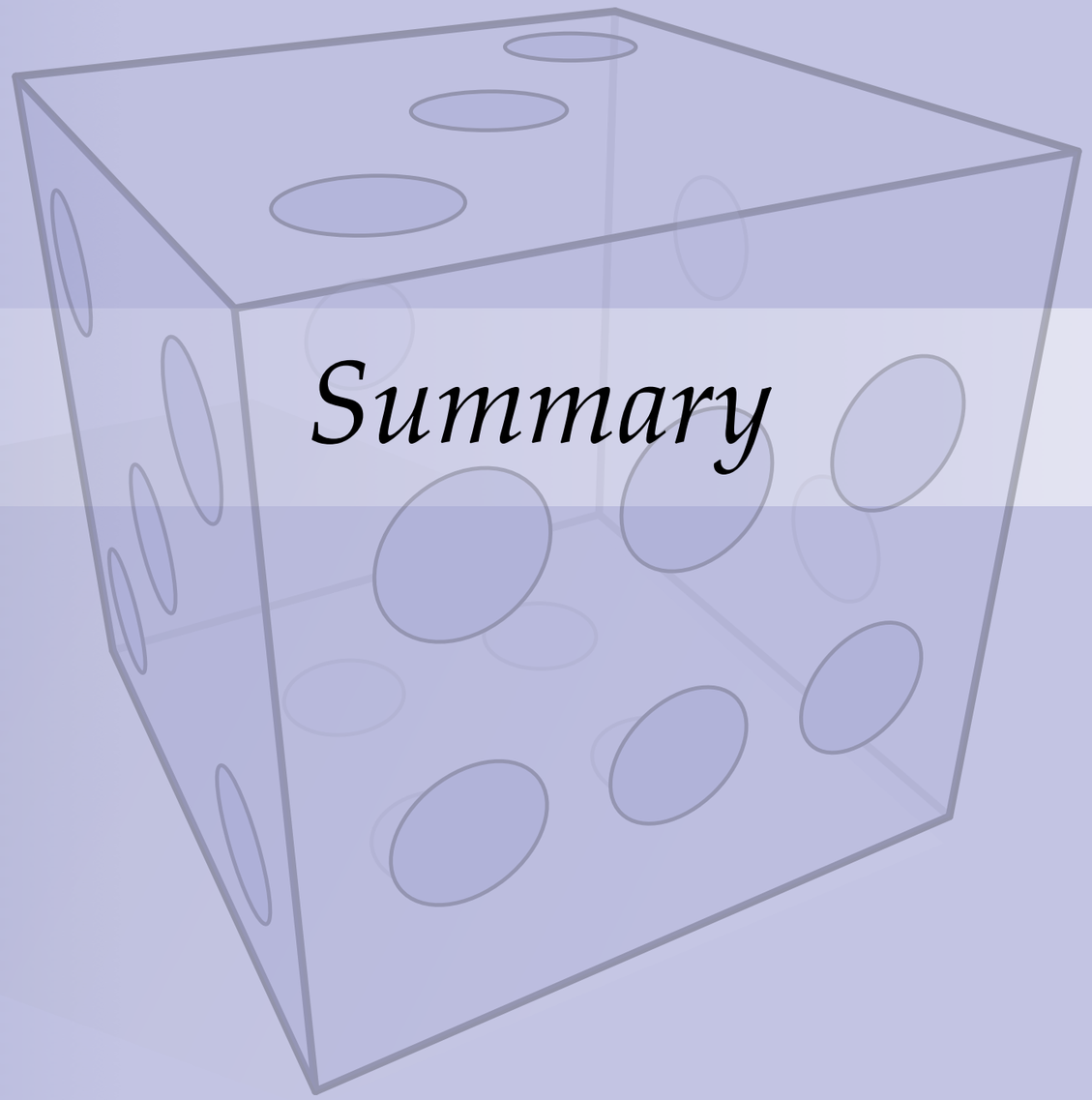
$$\cdots$$

$$\lim_{n\to\infty}\sum_{k=0}^{n} L_\varepsilon^k b = \underbrace{(I - L_\varepsilon)^{-1} b}_{\substack{\text{invertible since} \\ \|L_\varepsilon\|_{op} < 1}}$$

For constant function $u(x) := 1$, $Lu = u$.

$$\Rightarrow \|L\|_{\text{op}} = 1$$

$$\Rightarrow \|L_\varepsilon\|_{\text{op}} < 1$$



Summary

Basic Walk on Spheres—Summary

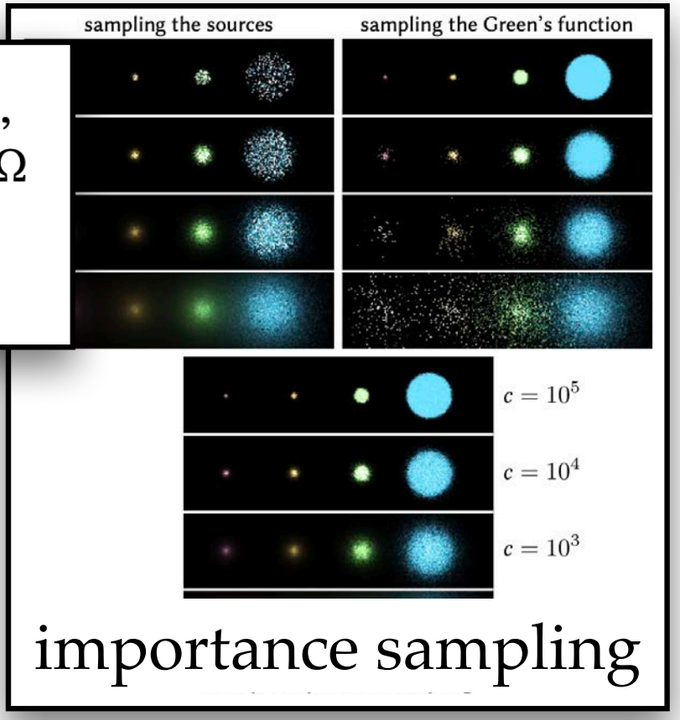
- Goal: simulate systems with very complex **geometry**
- Adopting a stochastic approach side-steps *meshing*
  - **walk on spheres (WoS)** makes stochastic approach fast enough to be practical (compared to **Euler-Maruyama**)
- Two perspectives:
  1. Stochastic calculus: WoS implements Kakutani's principle
  2. Potential theory: WoS estimates mean value property
- Both make use of *Monte Carlo integration*
  - lots of room for improvement (importance sampling, MCMC, …)

A lot more to say…

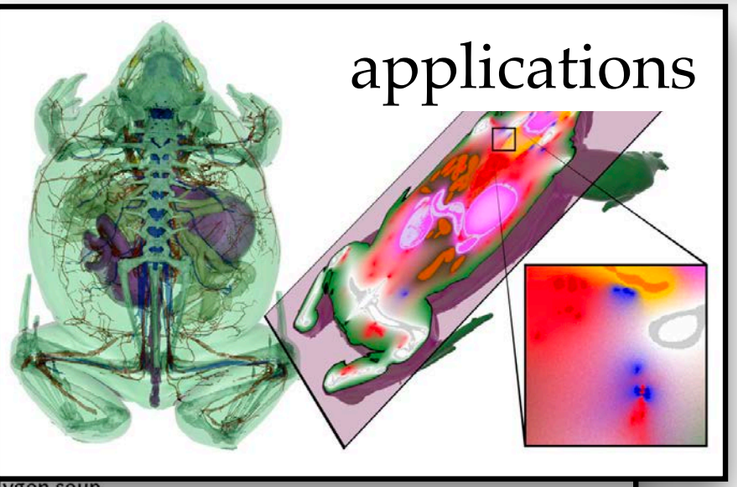
Will discuss some of this *next time!*

$$\begin{aligned} \nabla \cdot (\alpha \nabla u) + \vec{\omega} \cdot \nabla u - \sigma u &= -f \quad \text{on } \Omega, \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

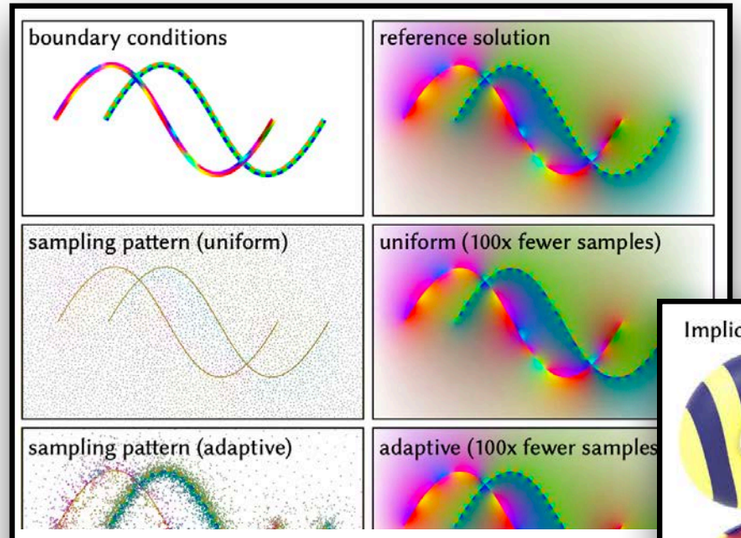
other PDEs



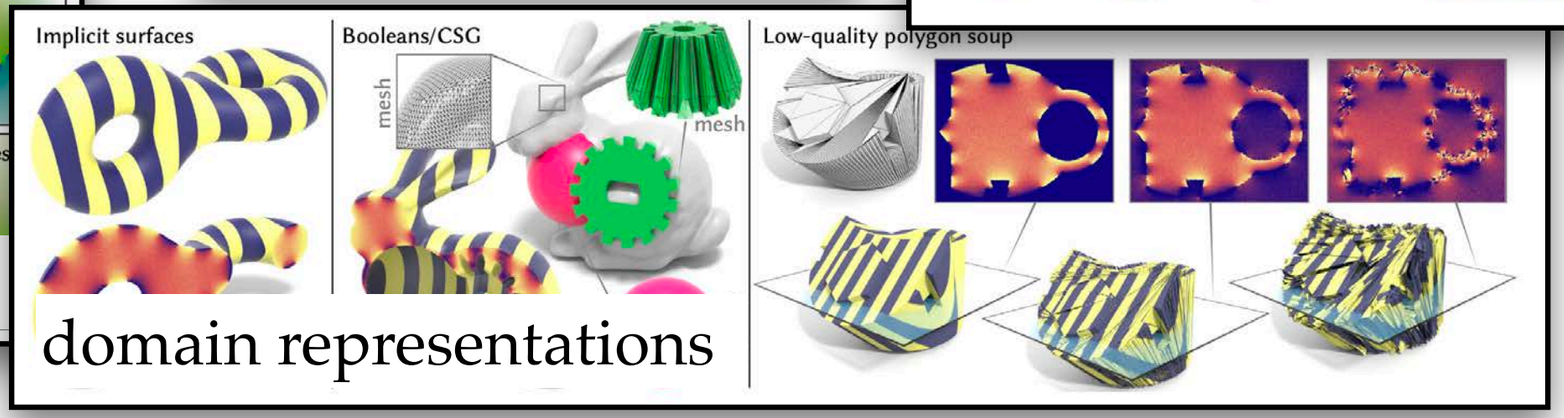
importance sampling



applications

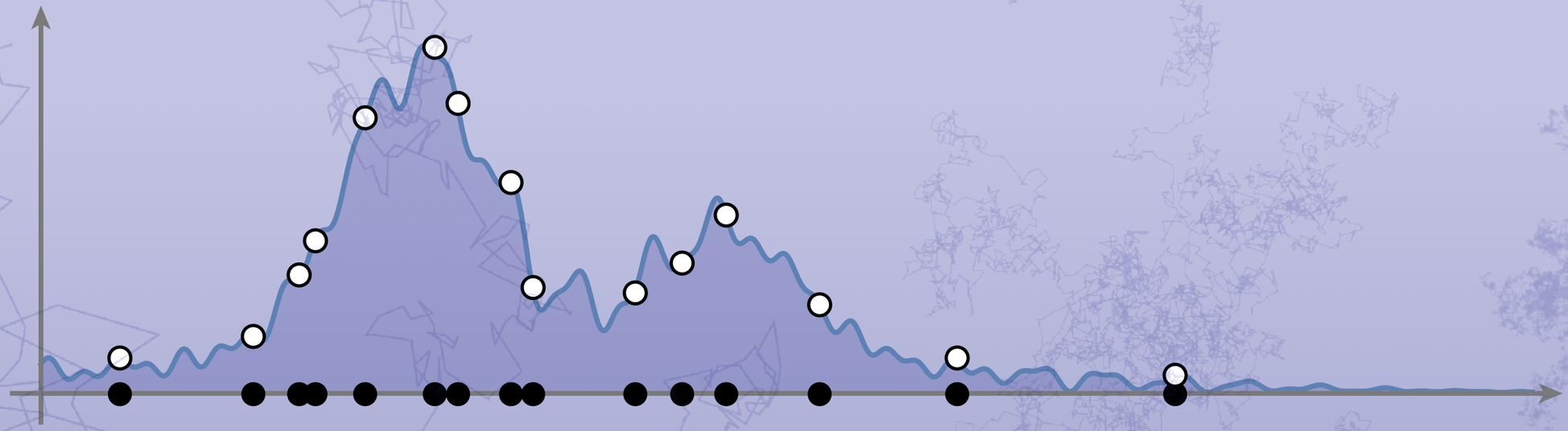


caching/adaptivity



domain representations

Thanks!



Monte Carlo Methods and Applications

Carnegie Mellon University | 21-387 / 15-327 / 15-627 / 15-860 | Fall 2023