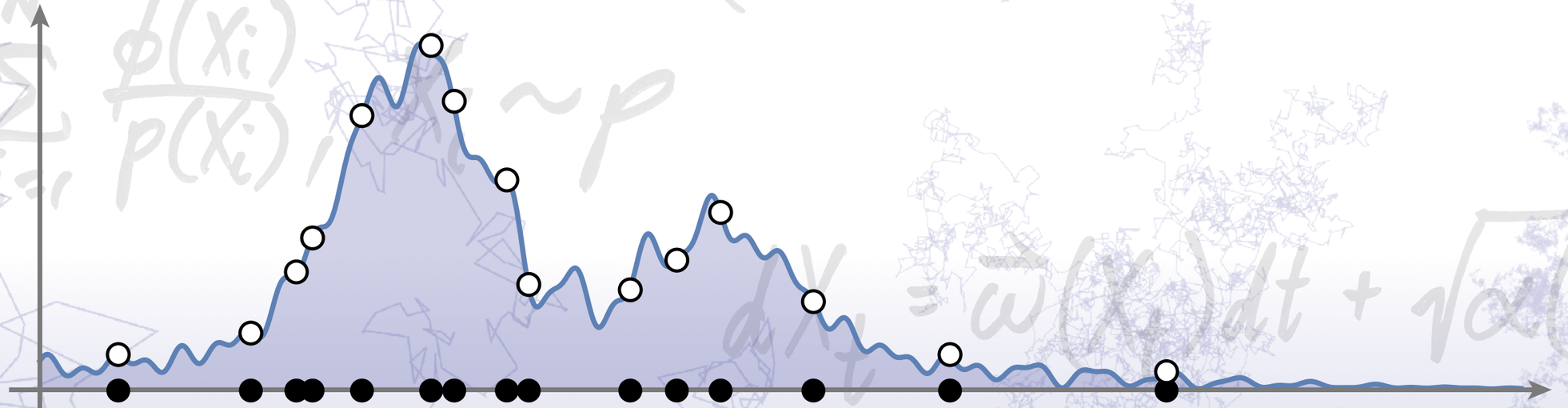
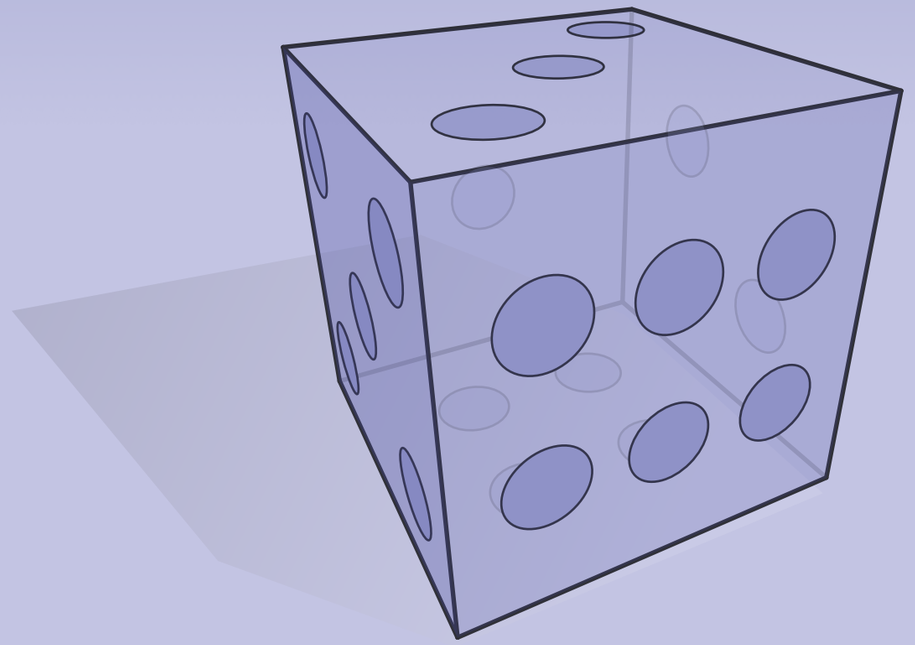
Monte Carlo Methods and Applications



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

Lecture 23

Walk on Spheres II



Monte Carlo Methods and Applications

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

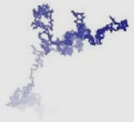
Walk on Spheres II—Overview

- **Last time:**
  - Motivated why we want to solve PDEs—and how Monte Carlo methods can help
  - Introduced basic *walk on spheres (WoS)* algorithm for Laplace equation
- **This time:**
  - Look at broader set of problems & algorithms
  - PDEs: Poisson, screened Poisson, convection-diffusion, heat …
  - Boundary conditions: Neuman, Robin, …
  - Applications & open questions

PDEs & Stochastic Processes: Beyond Brownian Motion

Brownian motion
$\Longleftrightarrow$ initial value problems
(e.g., heat equation)

branching random walks
$\Longleftrightarrow$ semilinear PDEs
(e.g., reaction-diffusion equations)



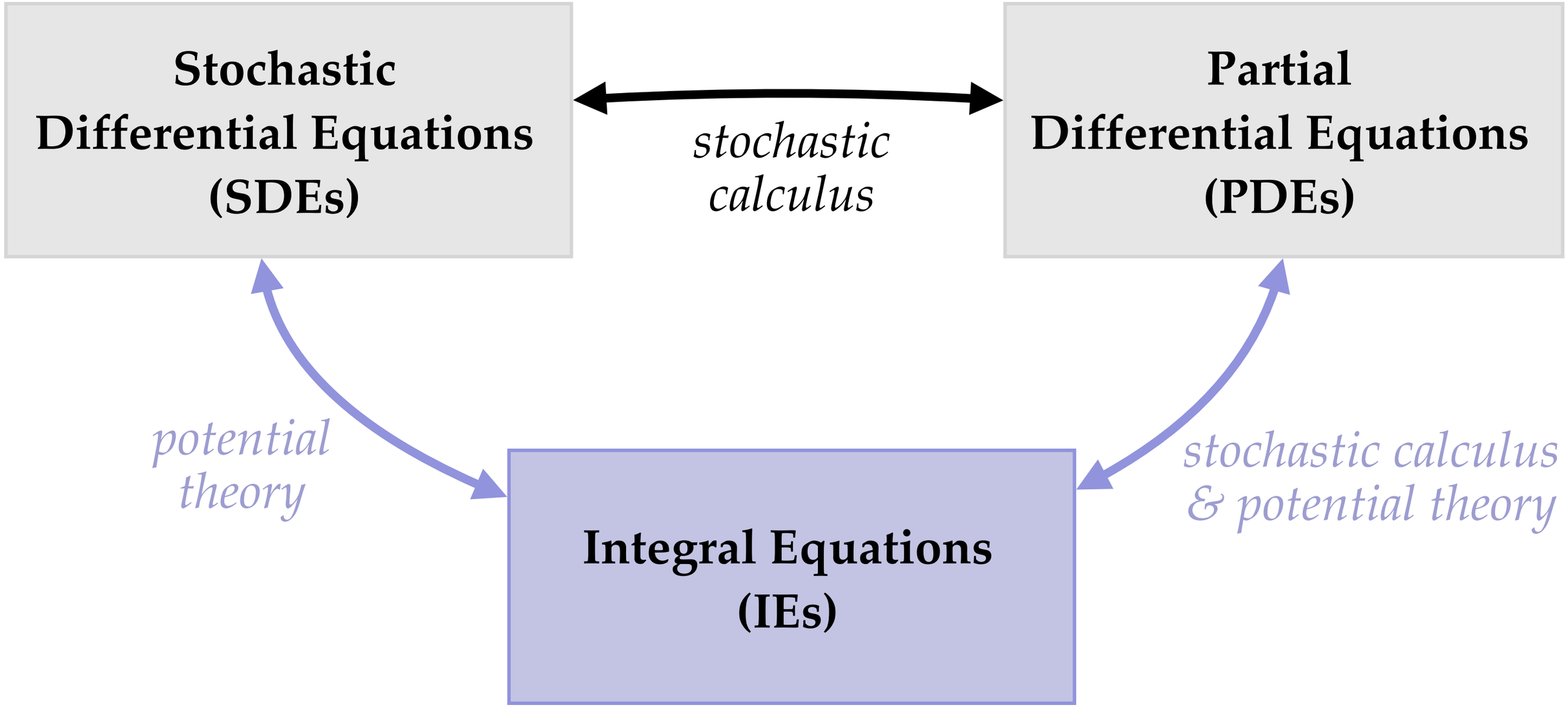
radially symmetric Levy process
$\Longleftrightarrow$ fractional order operators
(e.g., fractional Laplace equation)

stopped Brownian motion
$\Longleftrightarrow$ boundary value problems
(e.g., Laplace equation)

reflected Brownian motion
$\Longleftrightarrow$ Neumann boundary conditions
(i.e., prescribe normal derivative)

partially reflected Brownian motion
$\Longleftrightarrow$ Robin boundary conditions
(i.e., prescribe derivative + value)

Three Perspectives

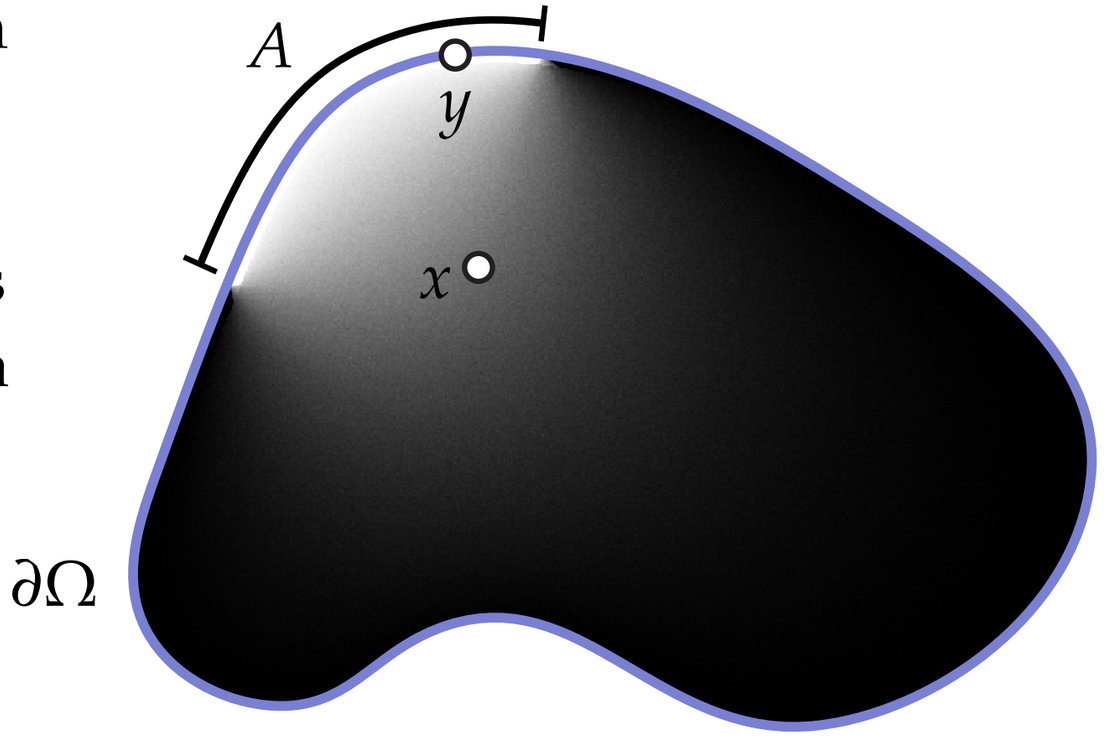


Basic Potential Theory

Review: 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"*):

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

Exit Distribution & Poisson Kernel

Stochastic Calculus

Exit distribution. For any point $x \in \Omega$, the *exit distribution* $P(x,y)$ describes how first hit locations of stopped Brownian motion X_t are distributed on $\partial\Omega$:

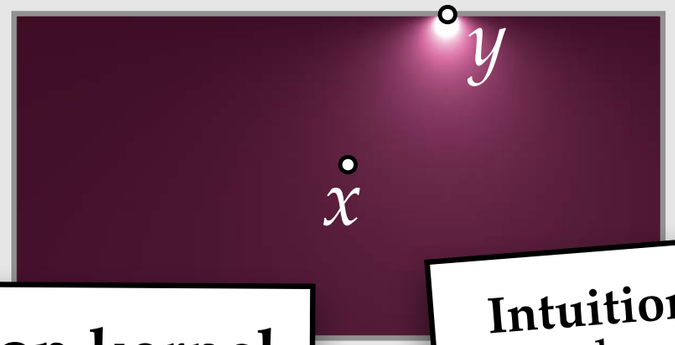
$$P(X_T^x \in A) = \int_A P(x,y)\, dy$$

Intuition: "how often does X_t exit through region A"?

Potential Theory

Poisson kernel. The equilibrium temperature at each point $x \in \Omega$ for a point source at boundary point y defines *Poisson kernel* $P(x,y)$:

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



Intuition: switch on a heater along wall of a cold room

Exit distribution & Poisson kernel are the same (up to normalization)

Walk Distribution & Green's Function

Stochastic Calculus

Location distribution. For any point $x \in \Omega$, let $G(x,y)$ be the probability that stopped Brownian motion X_t is found at y over the time interval $0 < t < T$:

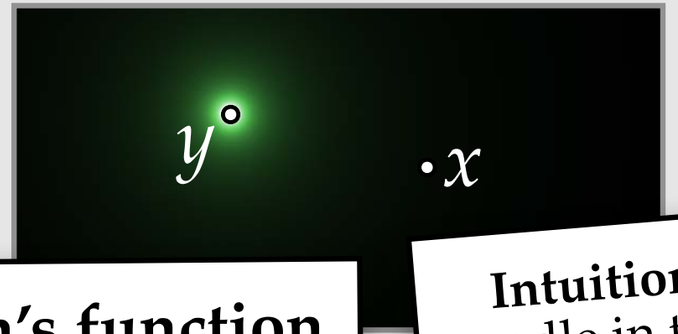
$$\mathbb{E}^x\left[\int_0^T \mathbf{1}_{X_t \in A} dt\right] = \int_A G(x,y)dy$$

Intuition: "how often does X_t visit region A"?

Potential Theory

Green's function. Let $G(x,y)$ be equilibrium temperature at x for a point source at interior point y, i.e., solution to Poisson equation

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



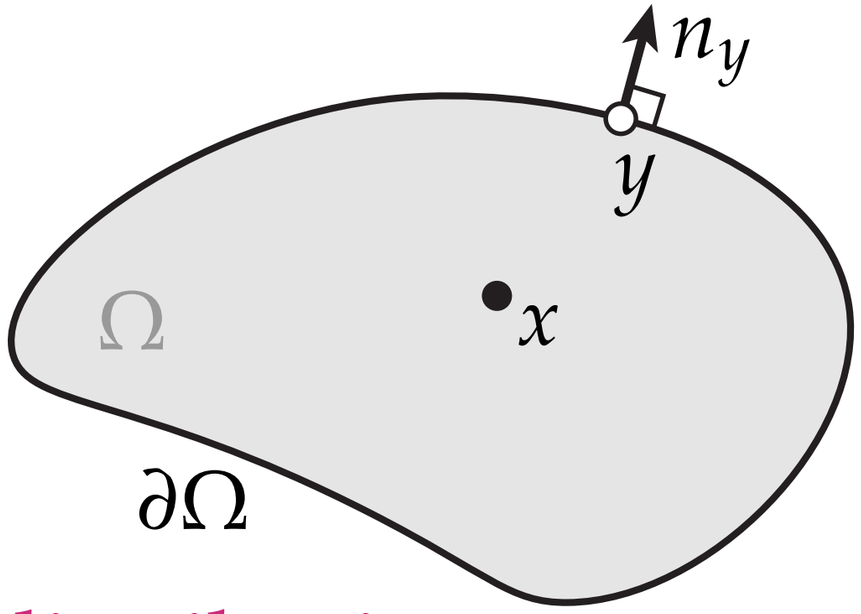
Intuition: light a candle in the middle of a cold room

Location distribution & Green's function are the same (up to normalization)

Green's Function & Poisson Kernel

STOCHASTIC CALCULUS ⟷ **POTENTIAL THEORY**

location distribution ⟷ Green's function G



$$P(x,y) = \frac{\partial G}{\partial n_y}(x,y)$$

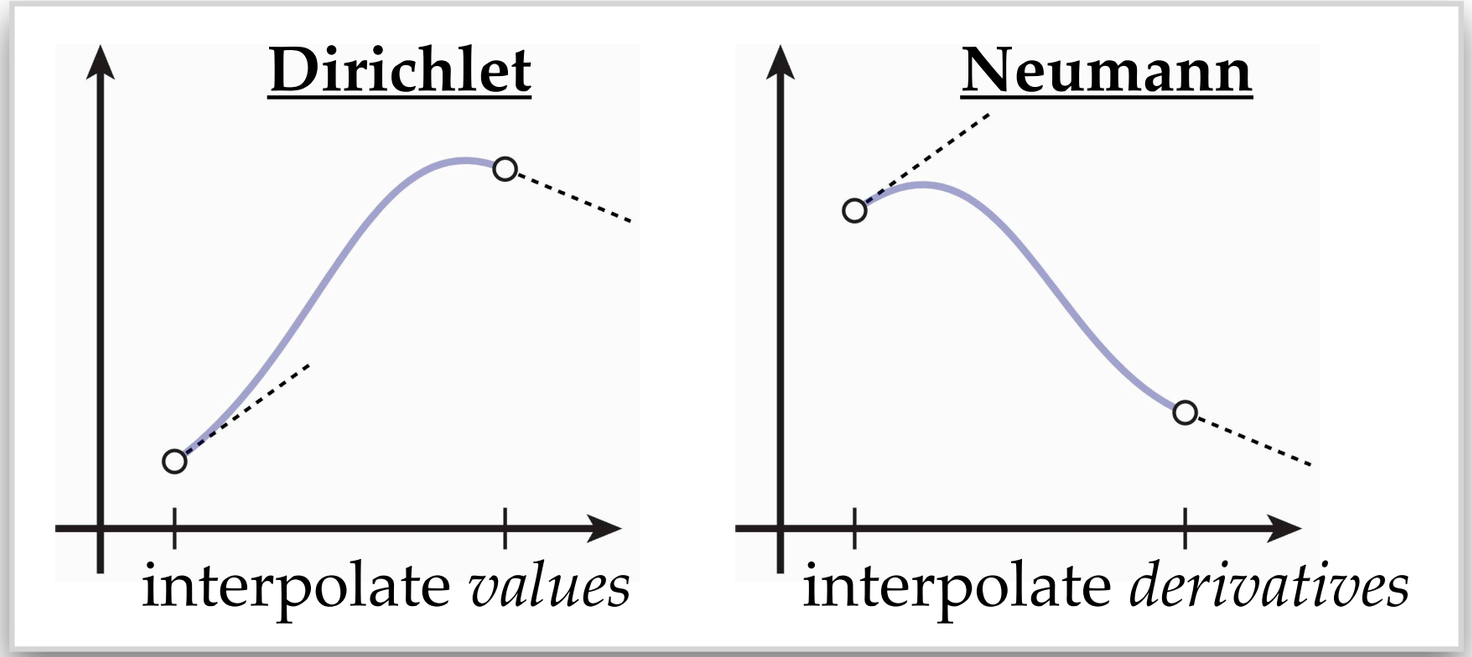
exit distribution ⟷ Poisson kernel P

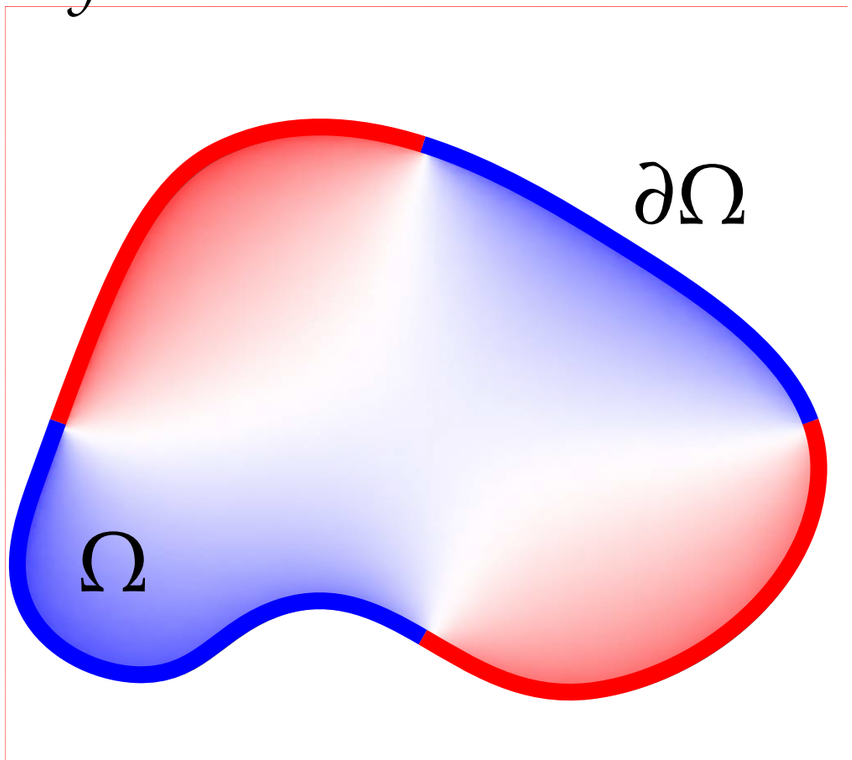
Boundary Integral Equation

In general, harmonic function u on domain Ω satisfies a *boundary integral equation (BIE)*

a.k.a *"Green's representation theorem"*

$$u(x) = \int_{\partial\Omega} P(x,y)u(y)\,dy - G(x,y)\frac{\partial u}{\partial n_y}\,dy$$





Boundary Integral Equation—Derivation

Start with a Laplace equation

$$\Delta u = 0$$

Multiply both sides by G and integrate to get

$$\int_\Omega G(x,y)\Delta u(y)\, dy = 0$$

Recall product rule for divergence operator:

$$\nabla \cdot (G\nabla u) = \nabla G \cdot \nabla u + G\Delta u$$

Integrating this relationship (*"integration by parts"*) yields

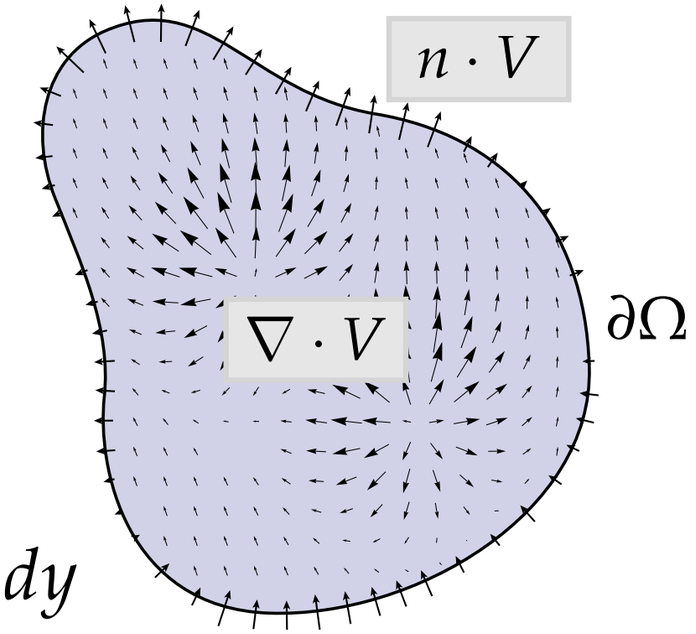
$$\int_\Omega \nabla \cdot (G(x,y)\nabla u(y))\, dy - \int_\Omega \nabla G(x,y) \cdot \nabla u(y)\, dy = 0$$

Boundary Integral Equation—Derivation (cont.)

Integrating this relationship (*"integration by parts"*) yields

$$\int_\Omega \nabla \cdot (G(x,y)\nabla u(y))\, dy - \int_\Omega \nabla G(x,y) \cdot \nabla u(y)\, dy = 0$$

$$\boxed{\int_\Omega \nabla \cdot V = \int_{\partial\Omega} n \cdot V}$$



Now apply divergence theorem to the first term:

$$\int_{\partial\Omega} G(x,y)\frac{\partial}{\partial n_y}u(y)\, dy - \int_\Omega \nabla G(x,y) \cdot \nabla u(y)\, dy = 0$$

Apply integration by parts again to second term:

$$\int_\Omega \Delta G(x,y)u(y)\, dy = \int_{\partial\Omega} P(x,y)u(y) - G(x,y)\frac{\partial}{\partial n_y}u(y)\, dy$$

By definition, $\Delta G(x,y) = \delta_x(y)$ and we get

$$u(x) = \int_{\partial\Omega} P(x,y)u(y) - G(x,y)\frac{\partial}{\partial n_y}u(y)\, dy$$

Boundary Integral Equation with Source

For linear-elliptic PDE $Lu = f$, can express solution via integral

$$u(x) = \int_{\partial\Omega} \underbrace{P(x,y)}_{\text{Poisson kernel}} u(y)\, dy \quad \text{Dirichlet boundary values}$$

$$- \int_{\partial\Omega} \underbrace{G(x,y)}_{\text{Green's function}} \frac{\partial u}{\partial n_y}\, dy \quad \text{Neumann boundary values}$$

$$+ \int_{\Omega} G(x,y) f(y)\, dy \quad \text{source term}$$

For a full derivation, see Section 3.3 of Sawhney et al,
"Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions" (2023)

Sounds pretty great:

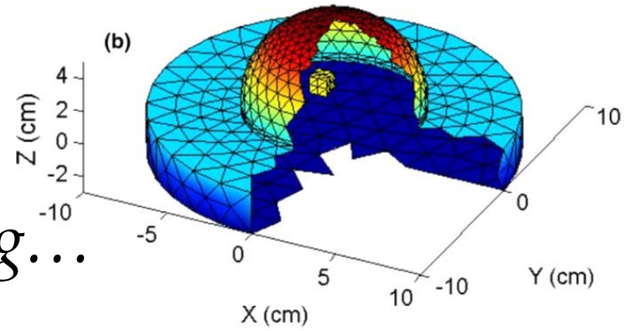
now we can just solve PDEs via integration!

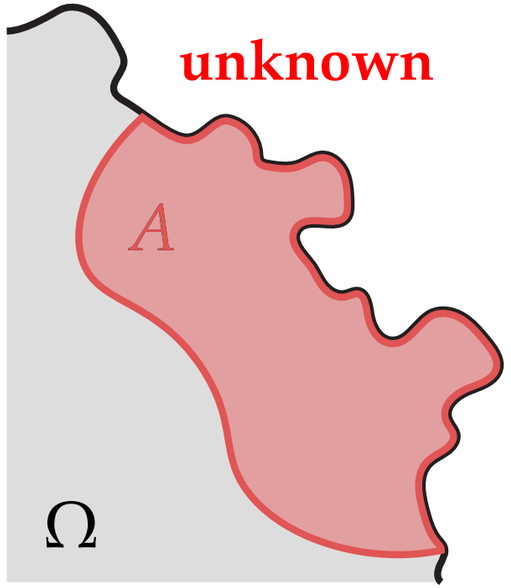
Problem:

…where do we get the Green/Poisson kernels?

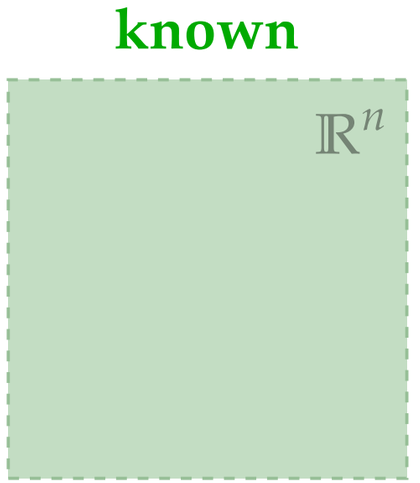
Walk on Subdomains

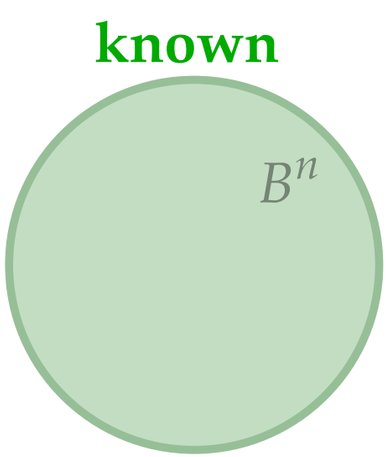
- For domains w/ complex geometry, *never* know Green/Poisson kernel!
- **Traditional approach:** boundary element method (BEM)
  - discretize & solve BIE
  - all the usual problems with *meshing*…

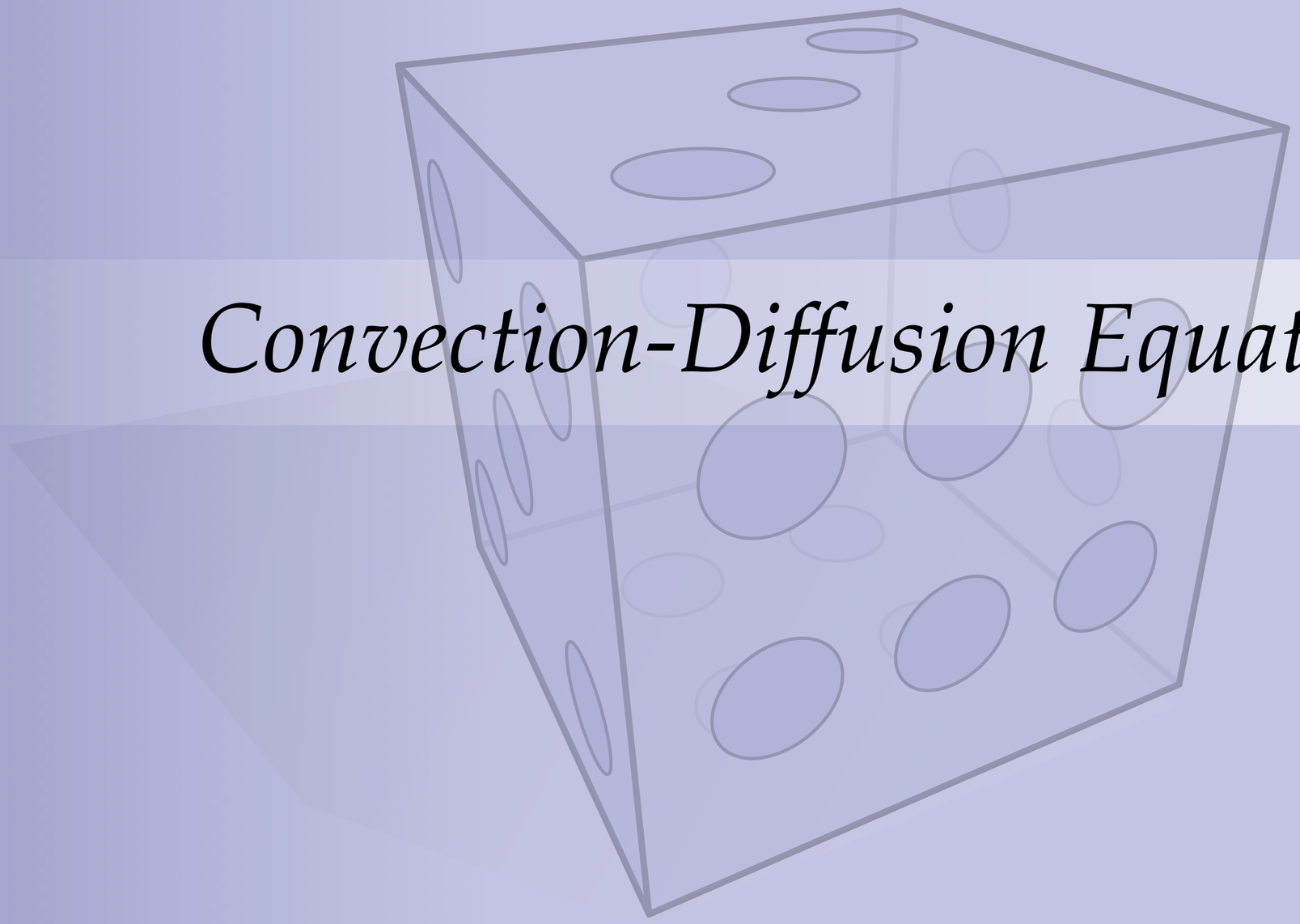




- **Monte Carlo approach:** *walk on X*
  - at least know kernels for simple domains
  - e.g., X = ball, rectangle, star-shaped region…
  - so, <u>recursively</u> estimate BIE for "easy" subdomains, via Monte Carlo







Convection-Diffusion Equations

Review: Convection-Diffusion Equation

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

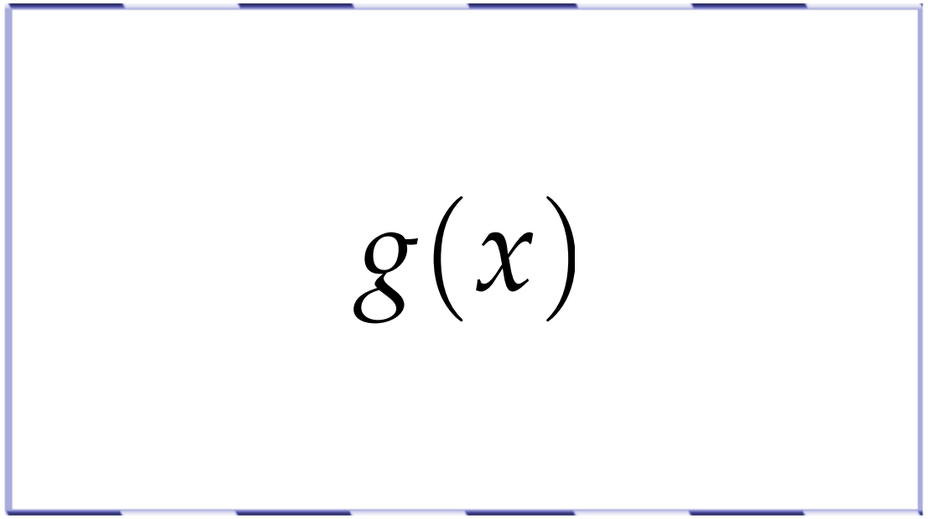
TEMPERATURE

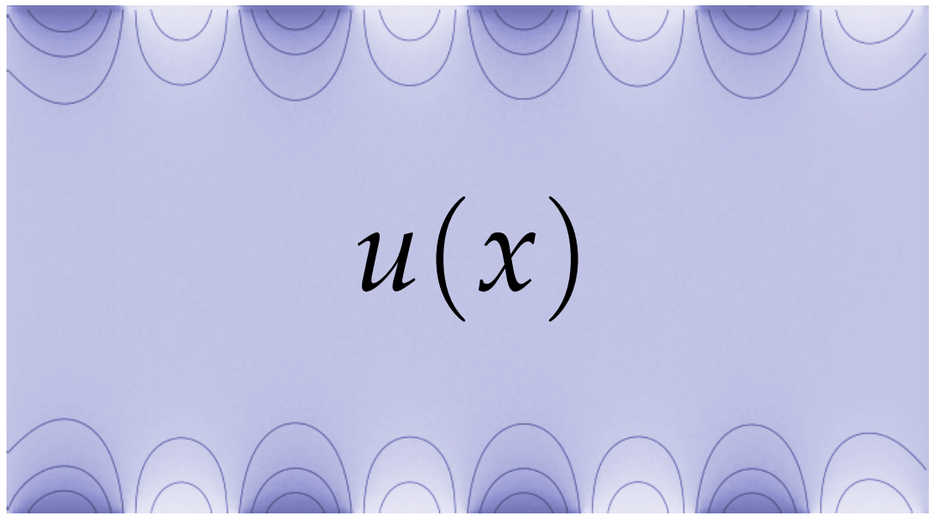
Review: Convection-Diffusion Equation

$$\nabla \cdot (\alpha \nabla u) + \vec{\omega} \cdot \nabla u - \sigma u = 0 \quad \text{on} \quad \Omega$$

$$u = g \quad \text{on} \quad \partial\Omega$$

BOUNDARY VALUE





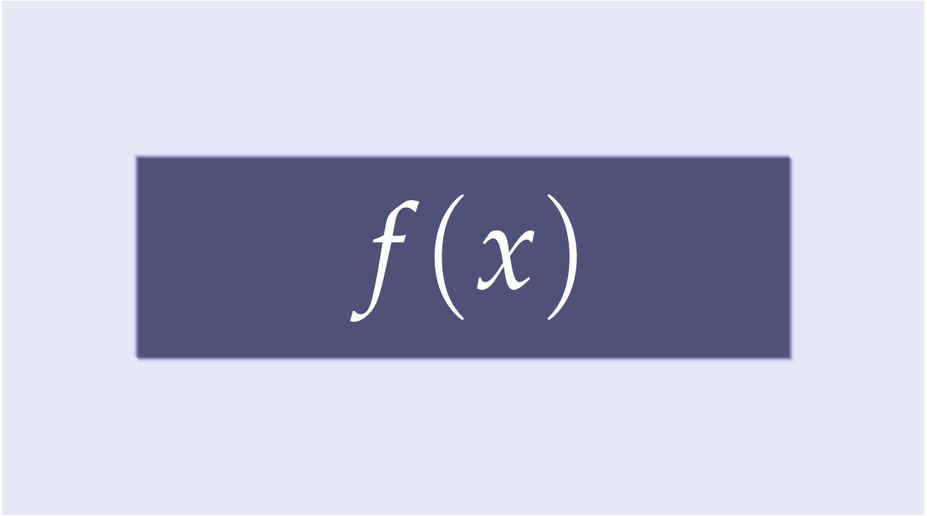
Intuition: temperature along boundary is fixed.

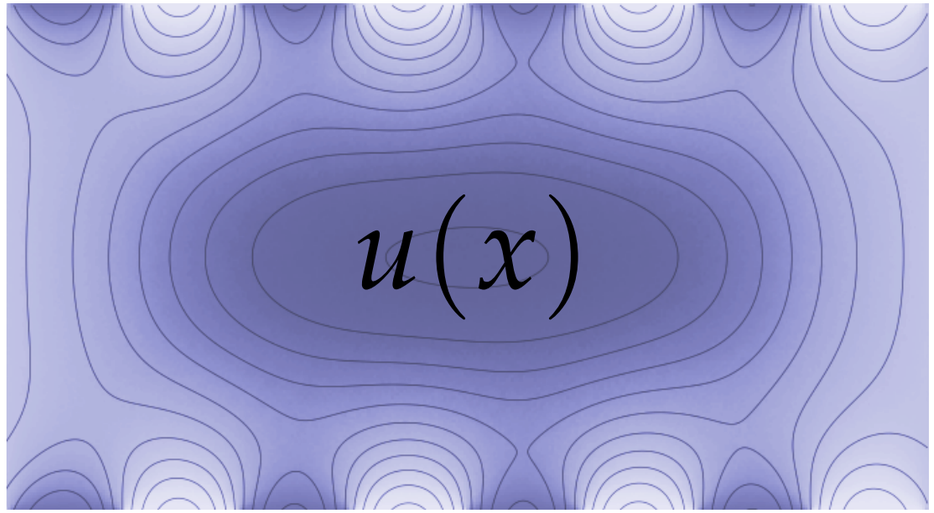
Review: Convection-Diffusion Equation

$$\nabla \cdot (\alpha \nabla u) + \vec{\omega} \cdot \nabla u - \sigma u = f \text{ on } \Omega$$

SOURCE

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



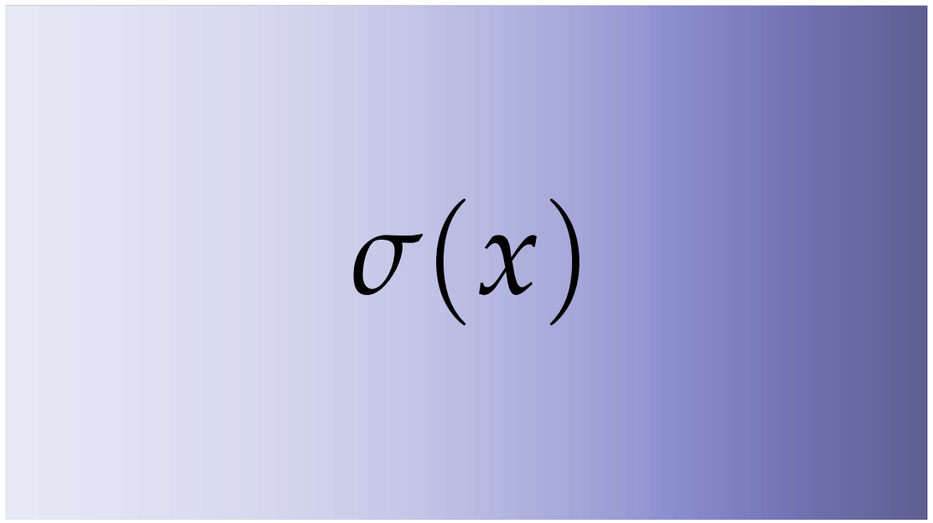


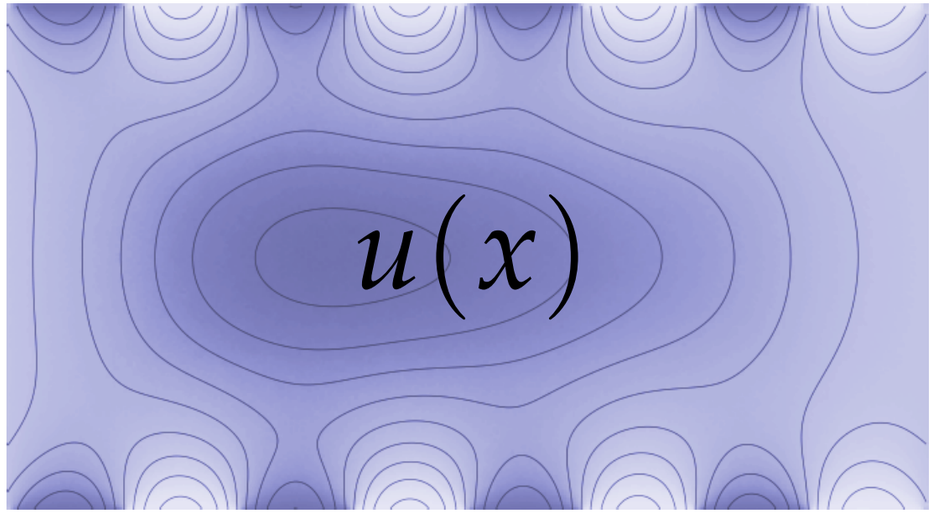
Intuition: adds additional "background temperature."

Review: Convection-Diffusion Equation

ABSORPTION

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

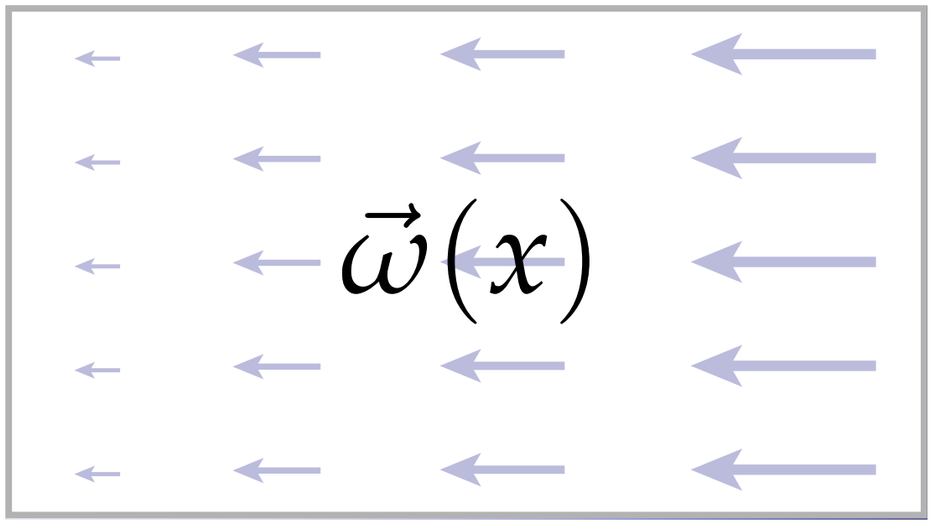


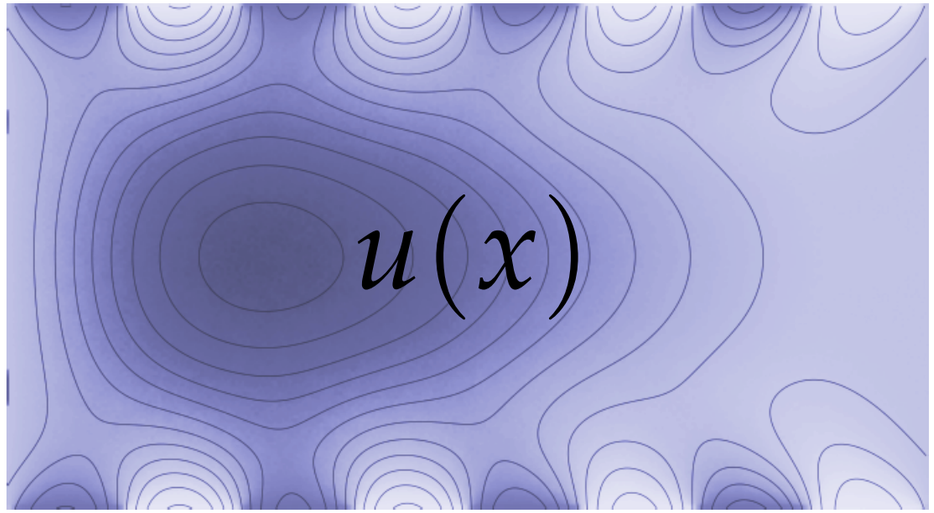


Intuition: "cooling" due to absorption into background medium.

Review: Convection-Diffusion Equation

$$\nabla \cdot (\alpha \nabla u) + \underbrace{\vec{\omega} \cdot \nabla u}_{\text{DRIFT}} - \sigma u = f \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$



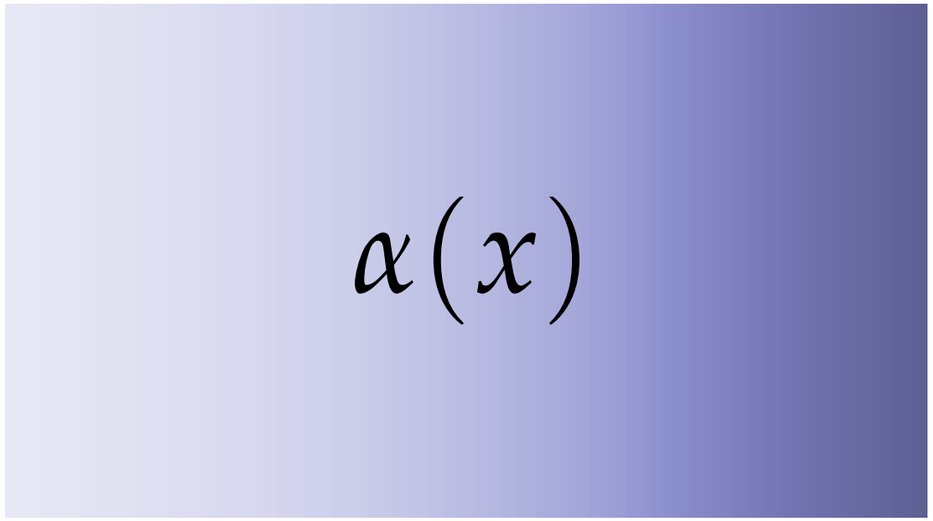


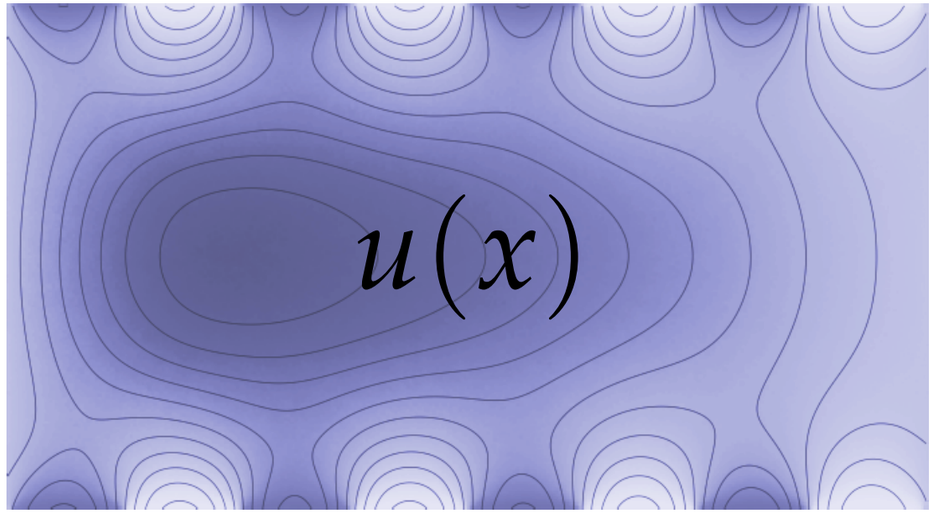
Intuition: heat is dragged along with a flowing river.

Review: Convection-Diffusion Equation

DIFFUSION

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





Intuition: how fast does heat "spread out"?

Review: Convection-Diffusion Equation

In general, coefficient functions α, ω, σ can vary over space:

$$\underbrace{\nabla \cdot (\alpha(x)\nabla u)}_{\text{Diffusion}} + \underbrace{\vec{\omega}(x) \cdot \nabla u}_{\text{Drift}} - \overbrace{\sigma(x)u}^{\text{Absorption}} = \overbrace{f}^{\text{Source}} \text{ on } \Omega$$

$$u = \underbrace{g}_{\text{Boundary}} \text{ on } \partial\Omega$$

Review: Convection-Diffusion Equation

In general, coefficient functions α, ω, σ can vary over space:

$$\underbrace{\Delta u}_{\text{Diffusion}} + \underbrace{\vec{\omega} \cdot \nabla u}_{\text{Drift}} - \underbrace{\sigma u}_{\text{Absorption}} = \underbrace{f}_{\text{Source}} \quad \text{on} \quad \Omega$$

$$u = \underbrace{g}_{\text{Boundary}} \quad \text{on} \quad \partial\Omega$$

For now: assume coefficients are constant.

Boundary Term

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

boundary

differential equation
(Laplace)

KAKUTANI'S PRINCIPLE

$$\mathbb{E}\left[g(W_T)\middle|W_0 = x\right]$$

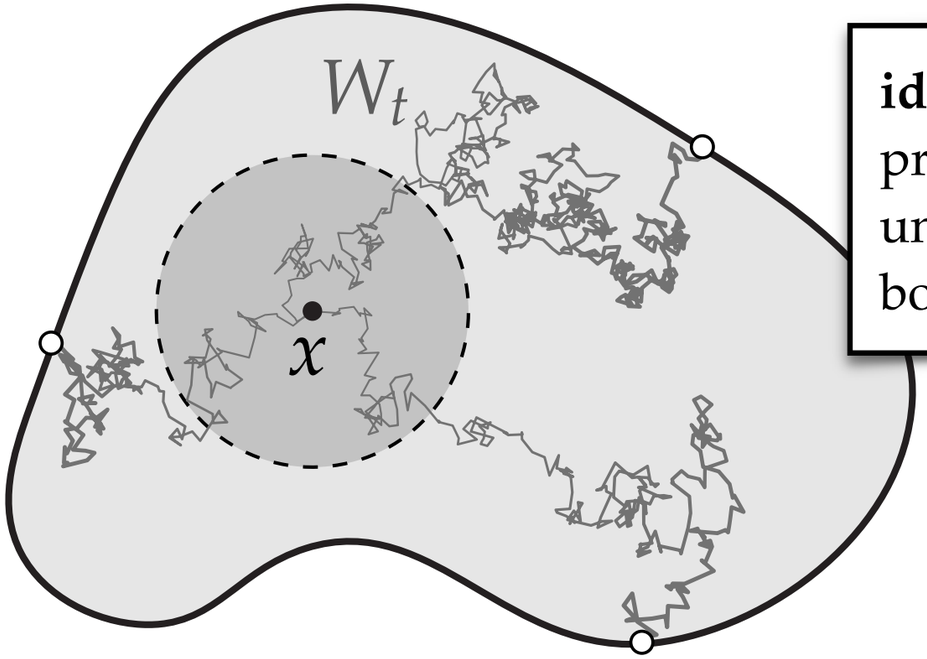
Brownian process

stochastic representation
(any domain)

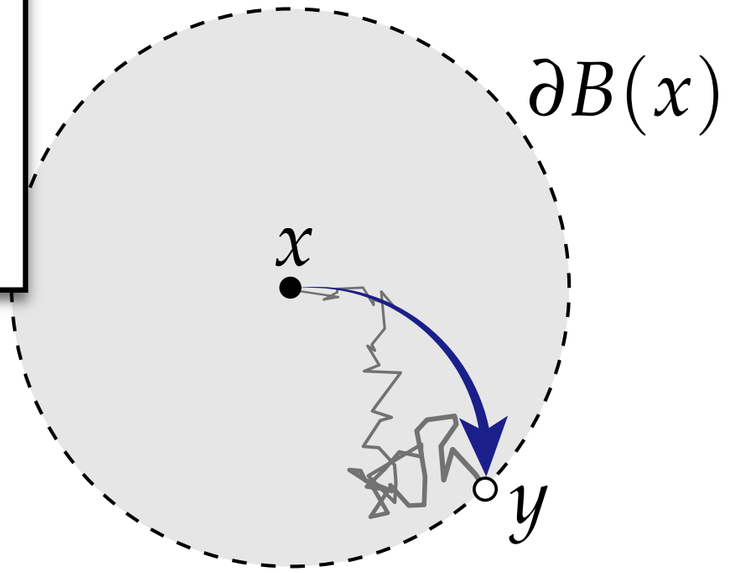
MEAN VALUE PROPERTY

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

integral representation
(on ball)



idea: apply Kakutani's principle in a ball; use unknown values u as boundary conditions g



Boundary Term — Estimator

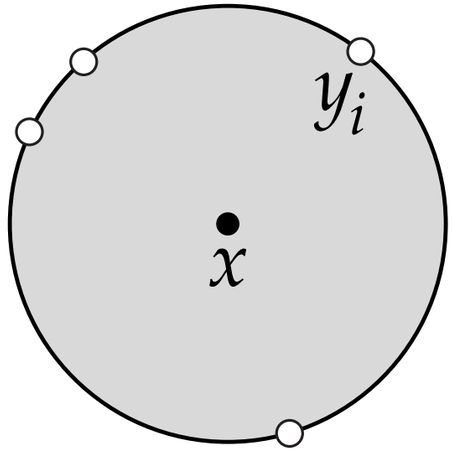
Monte Carlo estimator

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

uniform distribution on sphere

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

N=1



MEAN VALUE PROPERTY

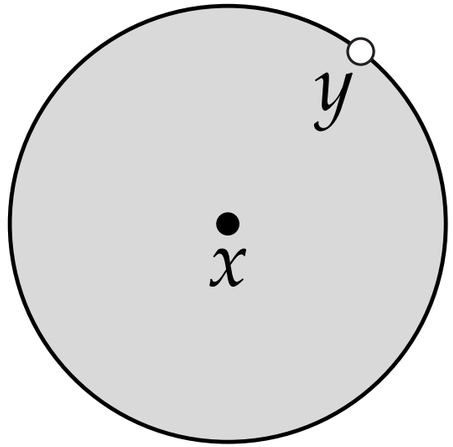
$$\frac{1}{|\partial B(x)|}\int_{\partial B(x)} u(y)dy$$

integral representation
(on ball)

Boundary Term — Estimator

Monte Carlo estimator

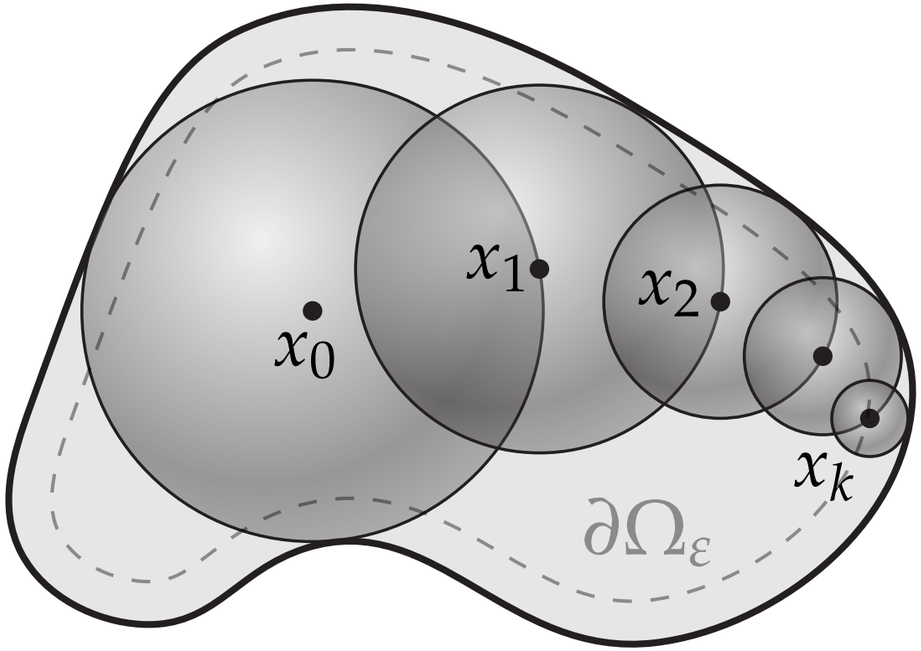
$$\widehat{u}(x) = \widehat{u}(y),\ y \sim \mathcal{U}_{\partial B(x)}$$



MEAN VALUE PROPERTY

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

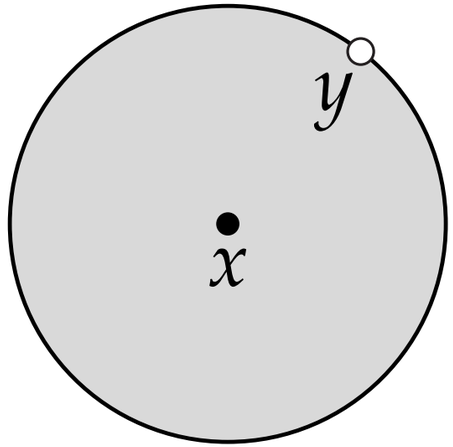
integral representation
(on ball)



Boundary Term — Estimator

Monte Carlo estimator

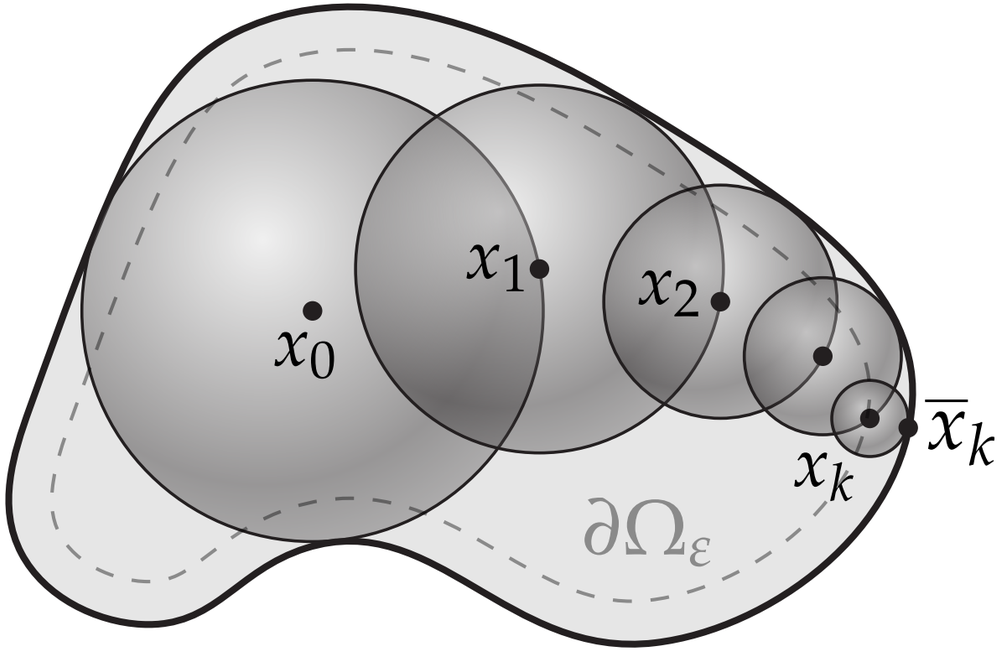
$$\widehat{u}(x) = \begin{cases} g(\overline{x}), & x \in \partial\Omega_\varepsilon \\ \widehat{u}(y), & \text{otherwise} \end{cases}$$



MEAN VALUE PROPERTY

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

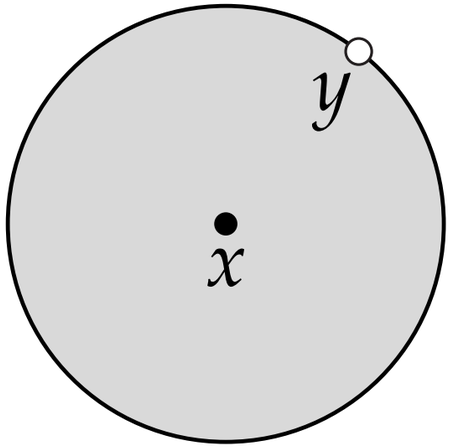
integral representation
(on ball)



Boundary Term — Estimator

Monte Carlo estimator

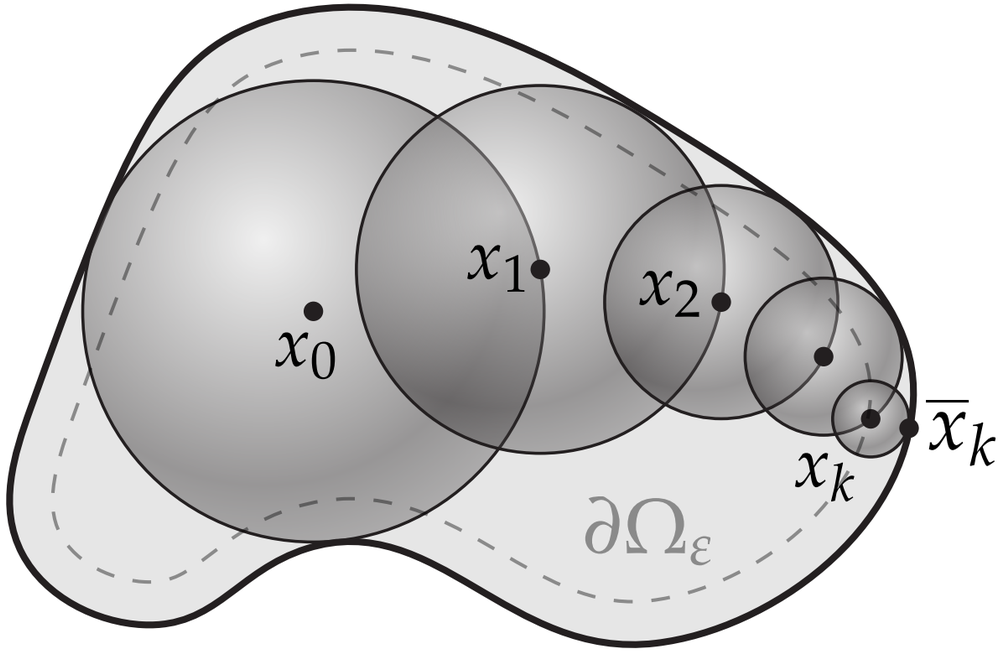
$$\widehat{u}(x) = \begin{cases} g(\overline{x}), & x \in \partial\Omega_\varepsilon \\ \widehat{u}(y), & \text{otherwise} \end{cases}$$



MEAN VALUE PROPERTY

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

integral representation
(on ball)




```
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
```

Source Term

SOURCE

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

differential equation
(Poisson)

TIME TO REACH BOUNDARY

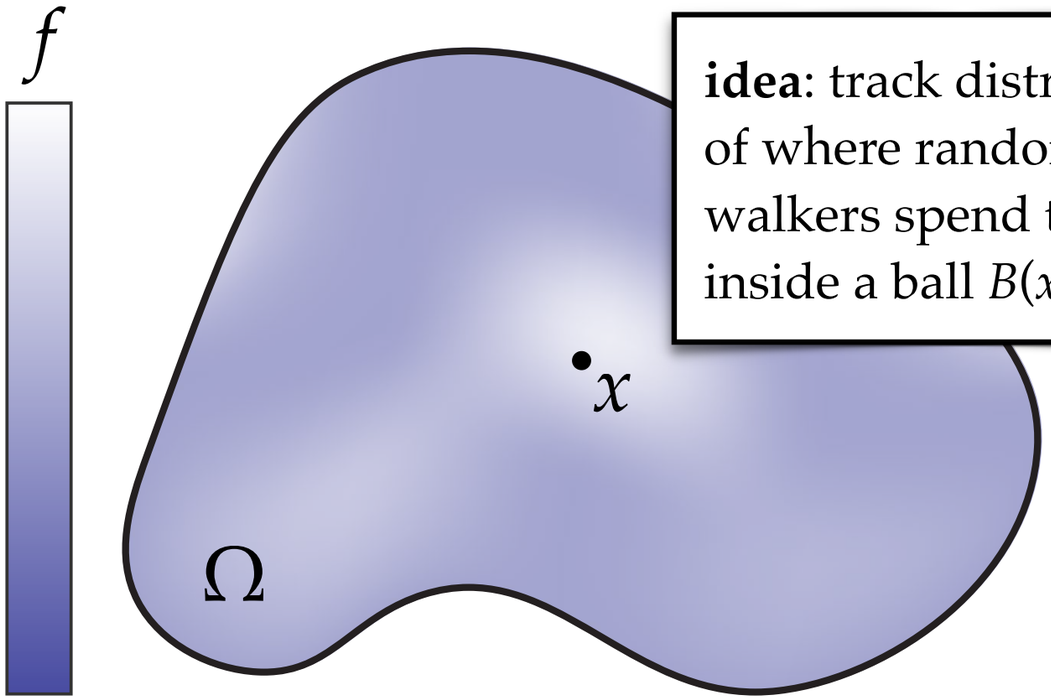
$$\mathbb{E}\left[\int_0^T f(W_t)dt \,\middle|\, W_0 = x\right]$$

stochastic representation
(any domain)

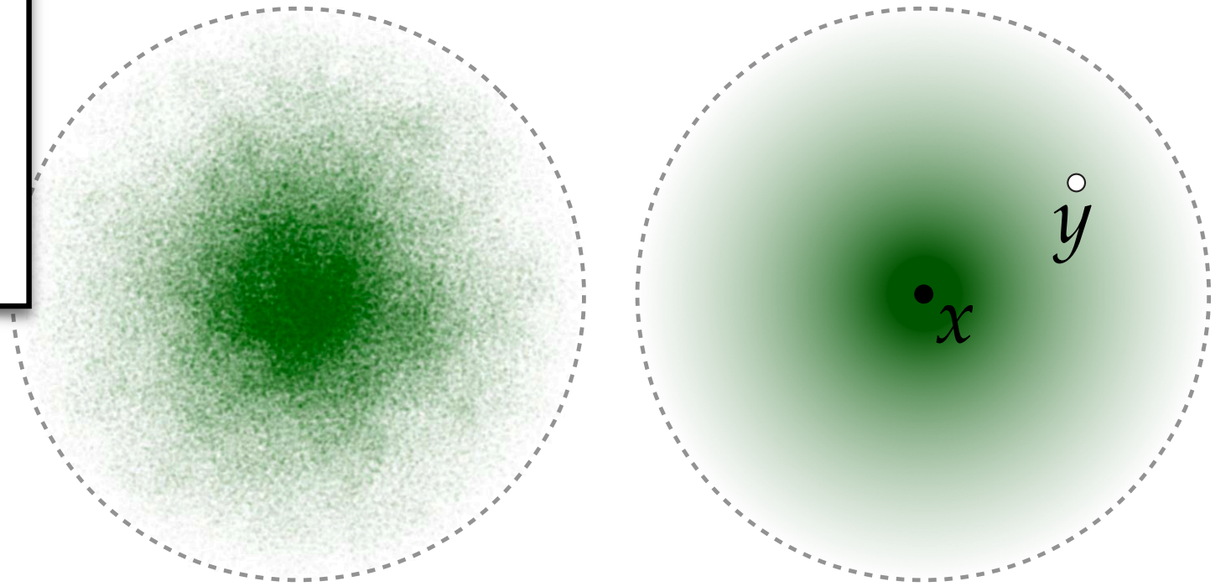
GREEN'S FUNCTION

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

integral representation
(source term on ball)



idea: track distribution of where random walkers spend time inside a ball $B(x)$



HARMONIC GREEN'S FUNCTION $G(x,y)$

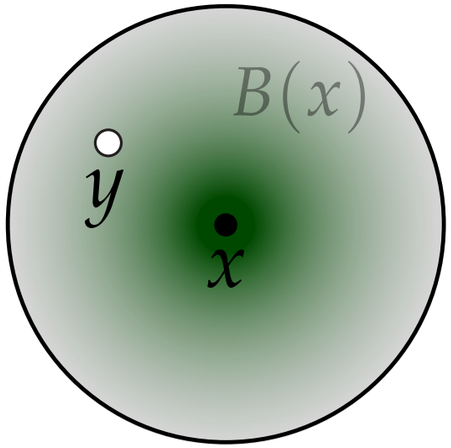
Source Term

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*



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

integral representation
(source term on ball)

[Epelov & Mikhailov 1969]

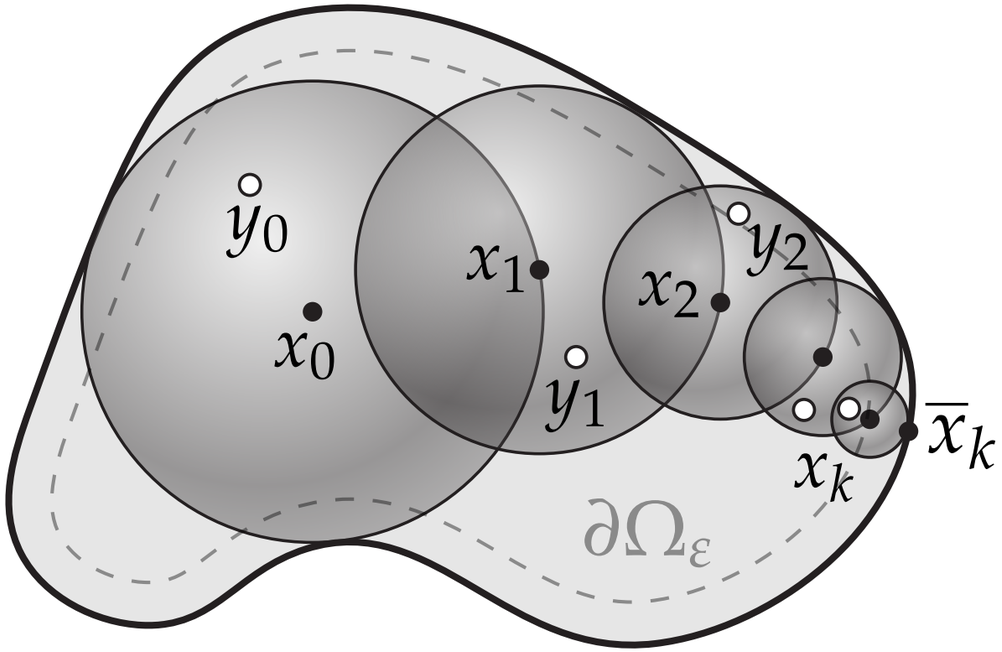
Source Term

Monte Carlo estimator

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

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

integral representation
(source term on ball)



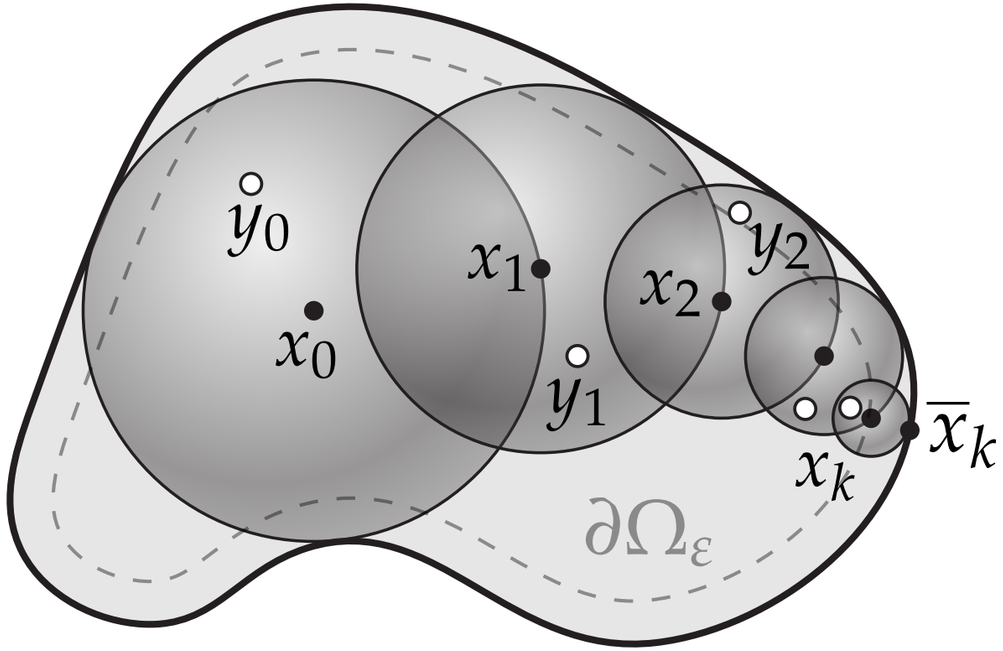
Source Term

Monte Carlo estimator

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

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

integral representation
(source term on ball)




```
u = 0 // solution estimate
for i=1,..,nWalks {
   x = x0 // start a new walk
   do {
      r = distance(x,∂Ω)
      y = randomBall(x,r) // random point inside ball
      u += π*r*r * G(x,y,r)*f(y) // source contribution
      x = randomSphere(x,r)
   } while(r > ε) // close enough!
   u += g(closestPoint(x,∂Ω)) // sample boundary value
}
return u/nWalks // return average boundary value
```

Absorption Term

ABSORPTION

$$\Delta u + \sigma u = f \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

differential equation
(screened Poisson)

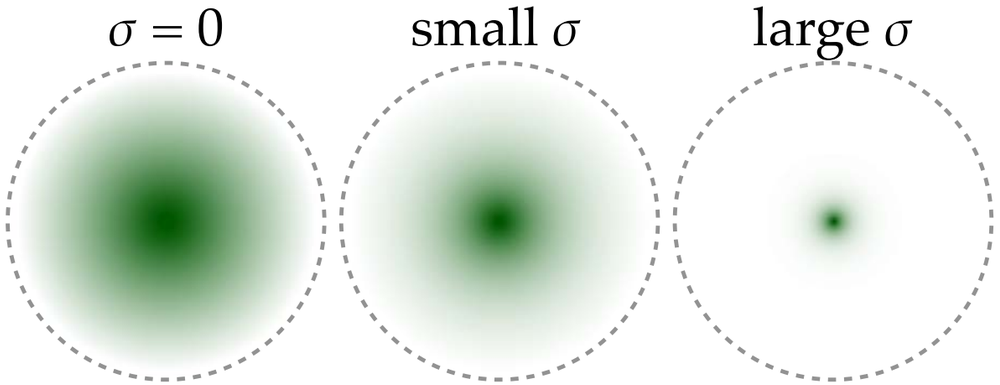
EXPONENTIAL DECAY

$$\mathbb{E}\left[\int_0^T e^{-\sigma t} f(W_t)\,dt + e^{-\sigma T} g(W_T)\right]$$

source term *boundary term*

stochastic representation
(any domain)

YUKAWA POTENTIAL



some walkers get "absorbed"
before exiting ball

Absorption Term

ABSORPTION

$$\Delta u + \sigma u = f \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

differential equation
(screened Poisson)

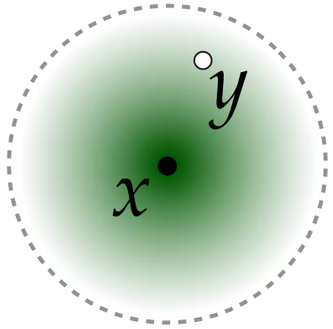
EXPONENTIAL DECAY

$$\mathbb{E}\left[\int_0^T e^{-\sigma t} f(W_t)\,dt + e^{-\sigma T} g(W_T)\right]$$

source term *boundary term*

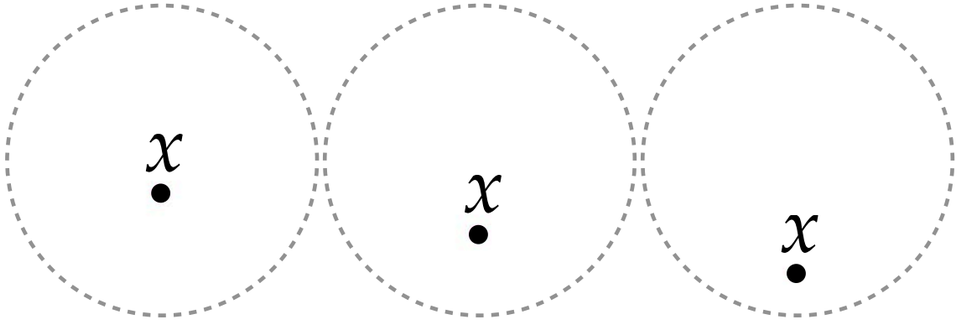
stochastic representation
(any domain)

YUKAWA POTENTIAL



$G^\sigma(x,y)$

POISSON KERNEL



$P^\sigma(x,y)$

GREEN'S FUNCTION

$$\int_{B(x)} G^\sigma(x,y) f(y)\,dy + \int_{\partial B(x)} P^\sigma(x,y) u(y)\,dy$$

POISSON KERNEL

integral representation
(source term on ball)

Drift Term—Same Story, Different Green's Function

DRIFT

$$\Delta u + \vec{\omega}\cdot\nabla u = f \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

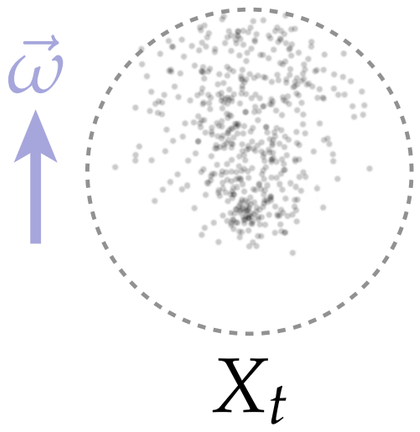
differential equation
(convection-diffusion)

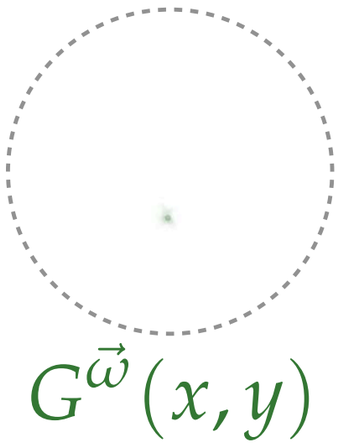
DIFFUSION PROCESS WITH DRIFT

$$dX_t = dW_t + \vec{\omega}(X_t)dt$$
$$\mathbb{E}\left[\int_0^T f(X_t)dt + g(X_t)\right]$$

stochastic representation
(any domain)

$P^{\vec{\omega}}(x,y)$ (VON-MISES FISHER)





$$P = \left.\frac{\partial G}{\partial n}\right|_{\partial B}$$

GREEN'S FUNCTION

$$\int_{B(x)} G^{\vec{\omega}}(x,y) f(y)dy + \int_{\partial B(x)} P^{\vec{\omega}}(x,y) u(y)dy$$

POISSON KERNEL

integral representation
(source term on ball)

[Sabelfeld 2016]

Summary (Constant Coefficients)

Differential Equation

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

Stochastic Representation $\quad dX_t = \vec{\omega} dt + \sqrt{\alpha} dW_t$

$$u(x) = \mathbb{E}\left[\int_0^T e^{-\int_0^t \sigma(X_s)ds} f(X_t)dt \;+\; e^{-\int_0^t \sigma(X_t)dt} g(X_t)\right]$$

Integral Formula $\quad \tilde{\sigma} := \sigma + \frac{1}{2}|\omega|^2$

$$u(x) = \int_{B(x)} G^{\tilde{\sigma}}(x,y) e^{\omega\cdot(y-x)} f(y)dy + \int_{\partial B(x)} P^{\tilde{\sigma}}(x,y) e^{\omega\cdot(y-x)} u(y)dy$$

Monte Carlo Estimator

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

Summary (Convection-Diffusion)

variable coefficients
derivative estimators
variance reduction
...

DIFFERENTIAL EQUATION

$$\Delta u + \vec{\omega} \cdot \nabla u - \sigma u = f \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

STOCHASTIC REPRESENTATION $dX_t = \vec{\omega} dt + \sqrt{\alpha} dW_t$

$$u(\quad \left[\int^T \quad \int^t \quad \right]$$

INTEGRAL FORMULA $\tilde{\sigma} := \sigma + \frac{1}{2}|\omega|^2$

$$u(x) = \int_{B(x)} G^{\tilde{\sigma}}(x,y) e^{\omega \cdot (y-x)} f(y) dy + \int_{\partial B(x)} P^{\tilde{\sigma}}(x,y) e^{\omega \cdot (y-x)} u(y) dy$$

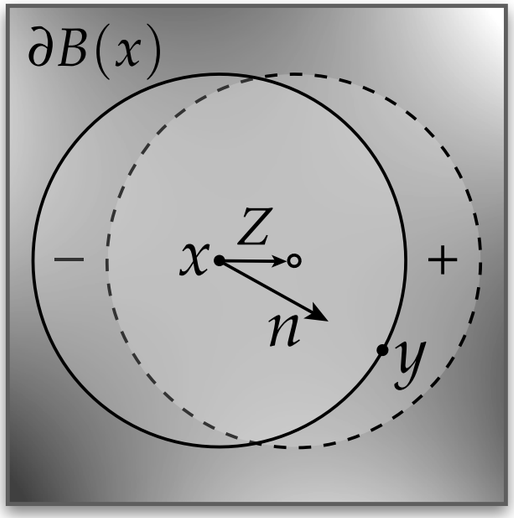
MONTE CARLO ESTIMATOR

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

Derivative Estimators

Laplace equation

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



directional derivative

$$D_Z u = Z \cdot \nabla u$$

$\Longrightarrow$ curl, divergence, …

value — mean value principle

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

gradient — Reynolds transport theorem

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

NORMAL

Hessian — Malliavin calculus

$$\nabla^2 u(x) = \frac{1}{|\partial B|} \int_{\partial B(x)} \frac{m^2}{R^4} g(y)(y-x) \otimes (y-x) - \frac{m}{R^2} g(y) I\, dy$$

For other PDEs, see [Sawhney & C. 2020]

(m=dimension, R=ball radius)

Composing Estimators

Can solve higher-order PDEs by "nesting" estimators.

Example. *(Biharmonic equation)*

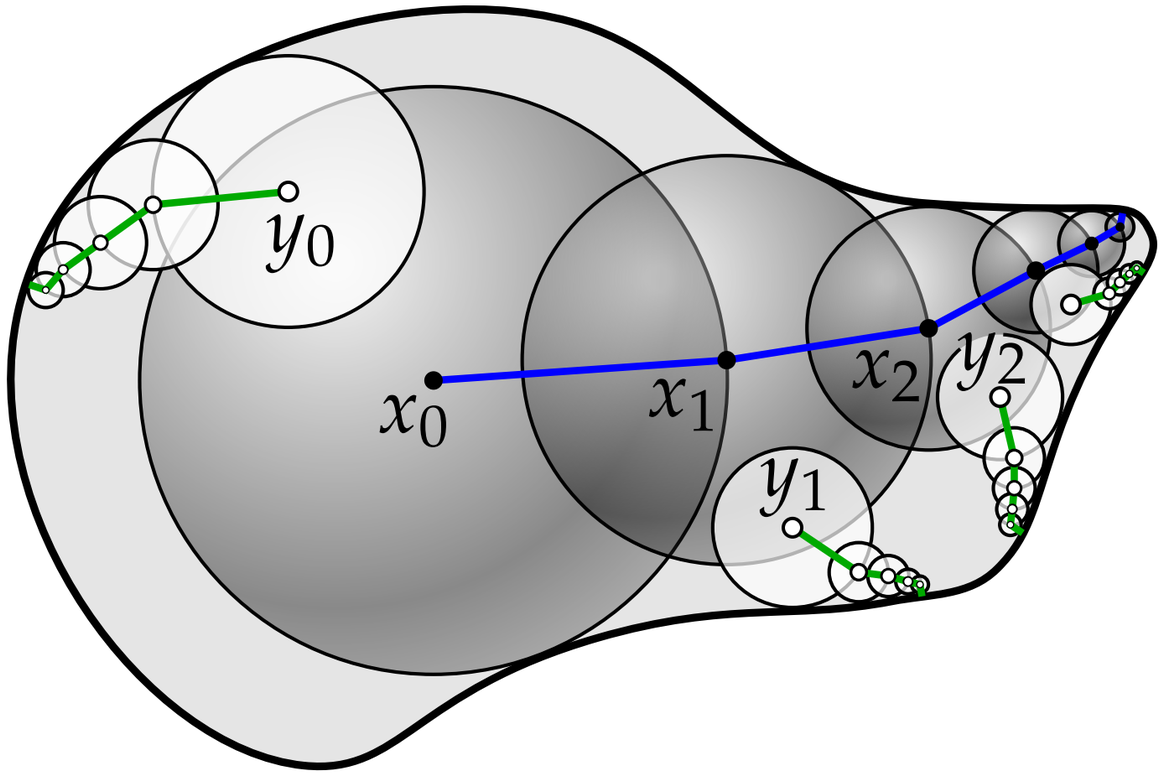
$$\Delta^2 u = f$$

$$\Longrightarrow \Delta u = v$$

invoke WoS estimator for v

$$\Delta v = f$$

NESTED WALK ON SPHERES

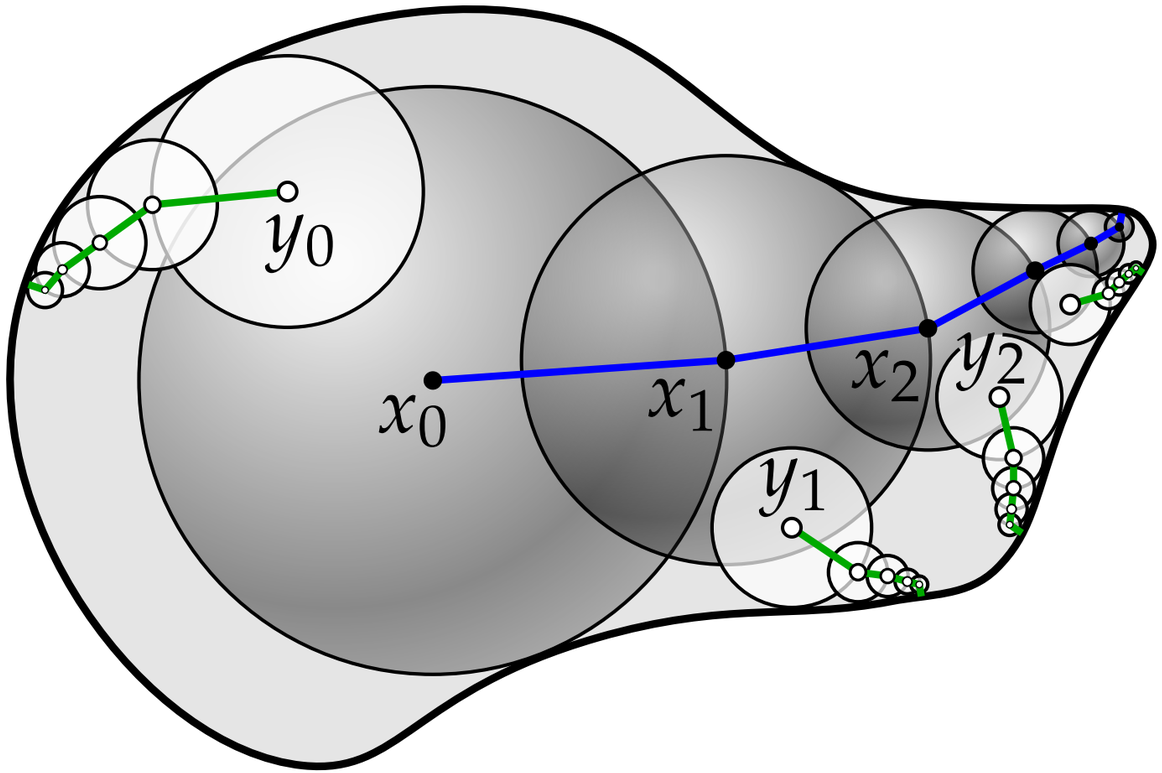


Can accelerate naïve scheme $O(n^2)$ to $O(n)$ by re-using walks

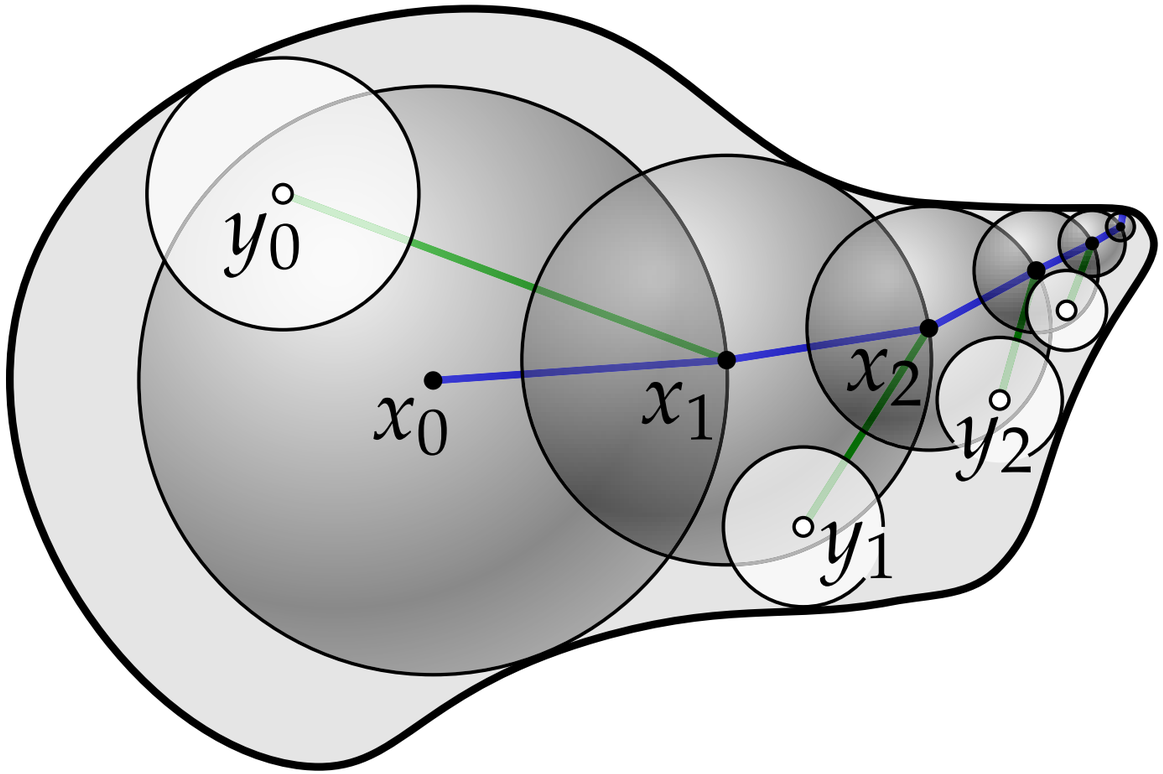
Composing Estimators

Can solve higher-order PDEs by “nesting” estimators.

Nested Walk on Spheres



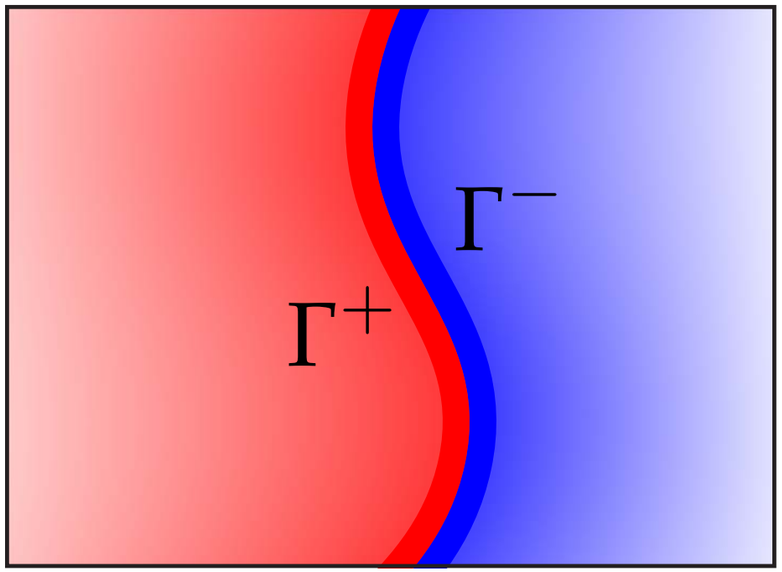
“Tree Walking” Estimator



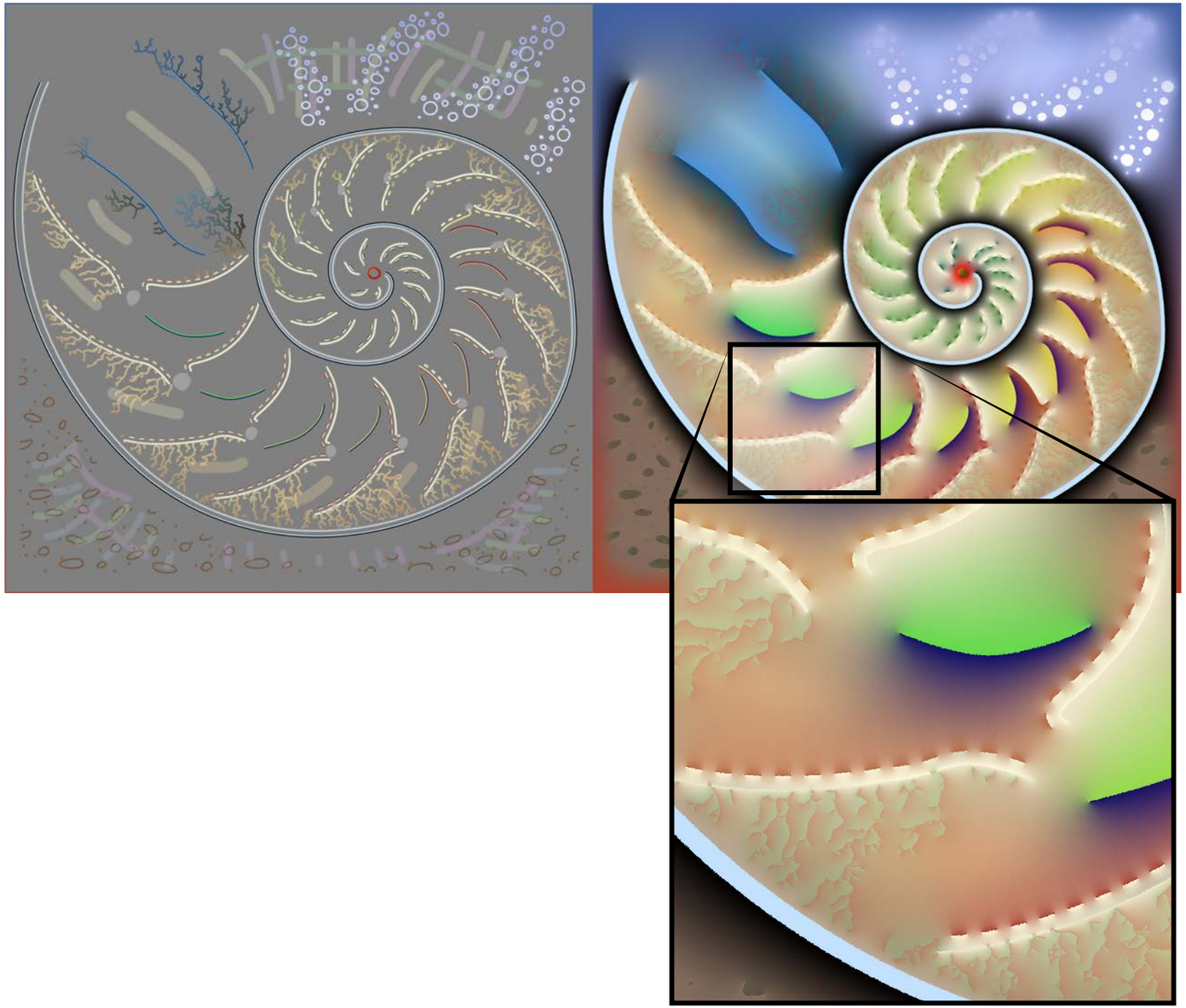
Can accelerate naïve scheme $O(n^2)$ to $O(n)$ by re-using walks

Diffusion Curves

$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \setminus \Gamma \\ u &= g^{+} && \text{on } \Gamma^{+} \\ u &= g^{-} && \text{on } \Gamma^{-} \end{aligned}$$

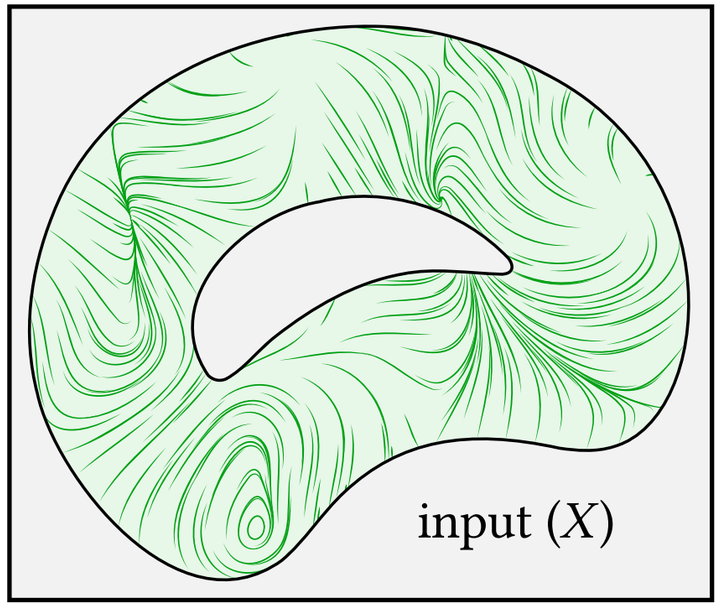


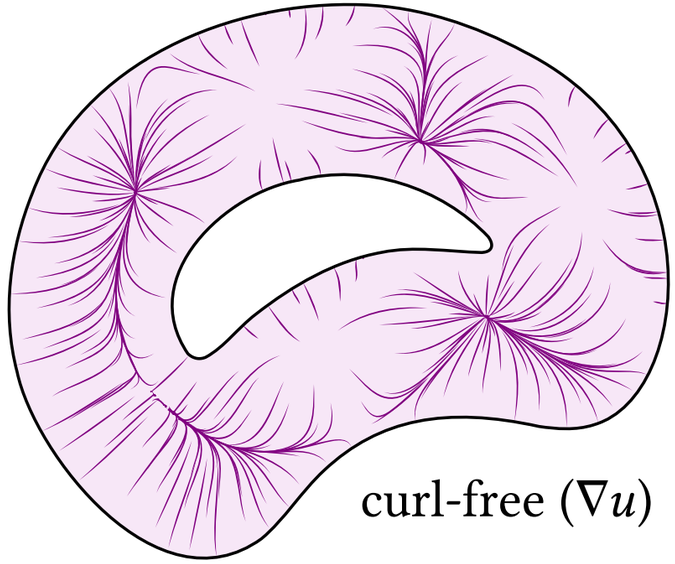
Γ — *collection of open & closed curves*

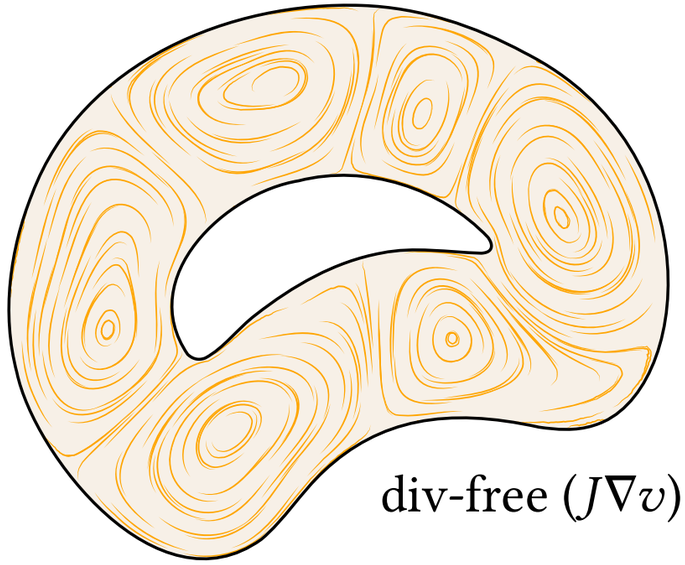


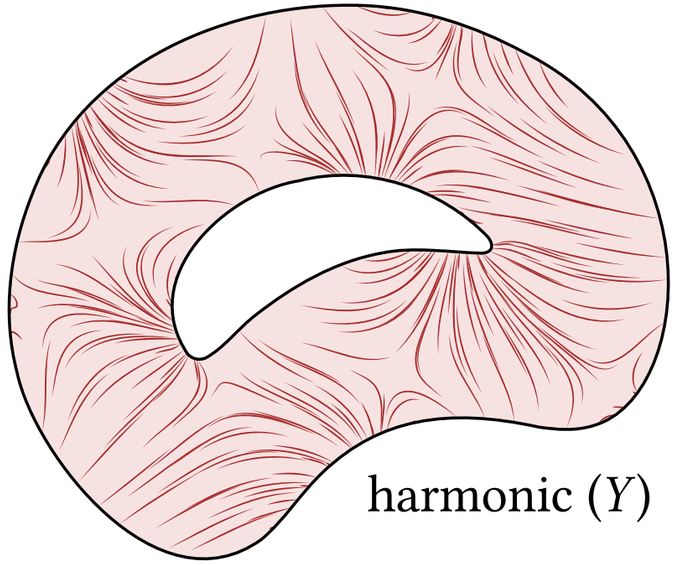
Helmholtz Decomposition (2D)

PDEs solved
only
at points along
integral curves

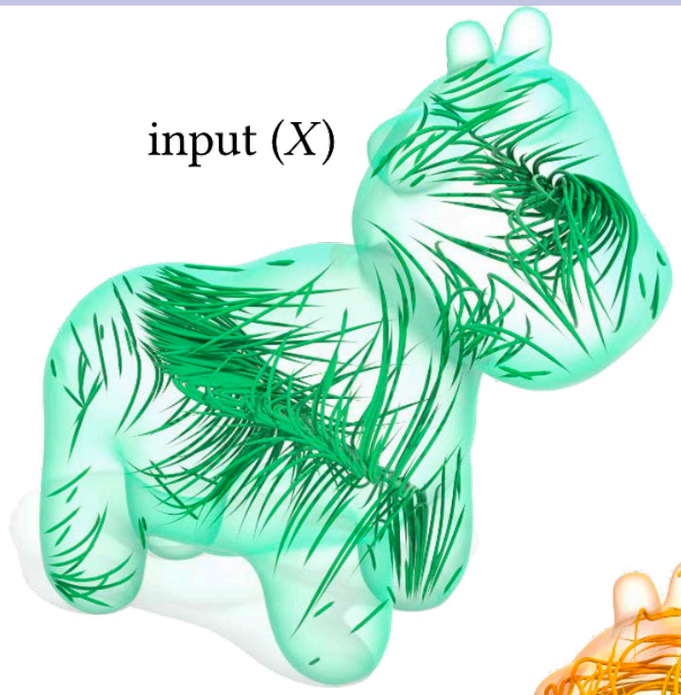


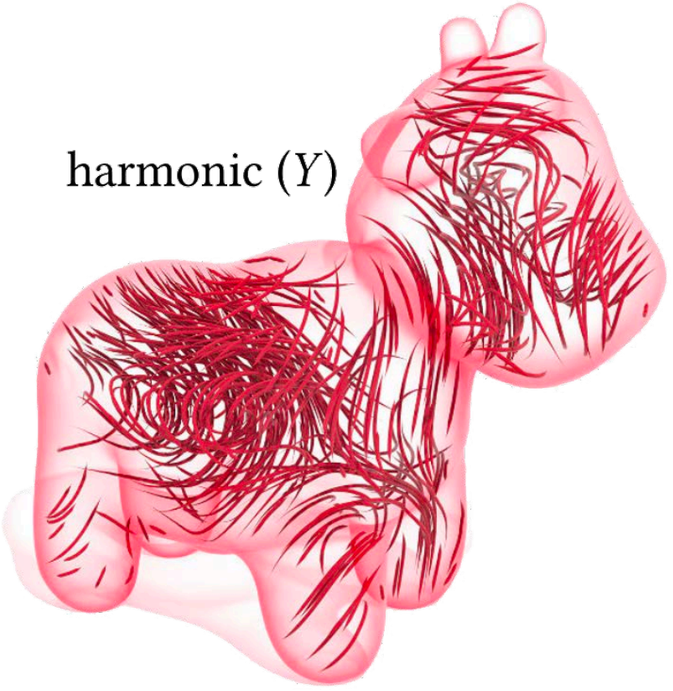






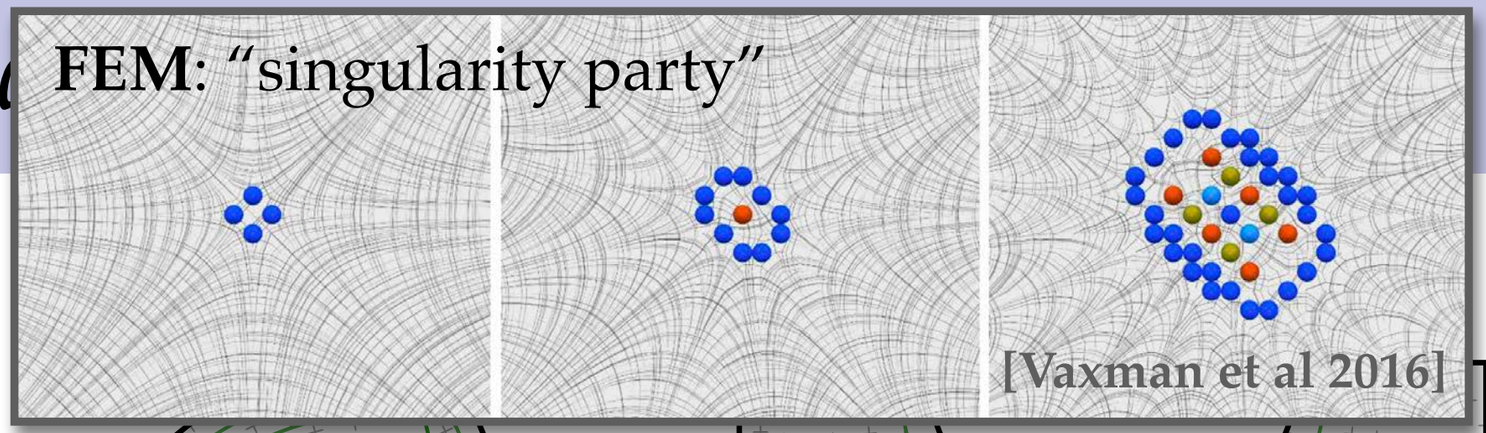
Helmholtz Decomposition (3D)





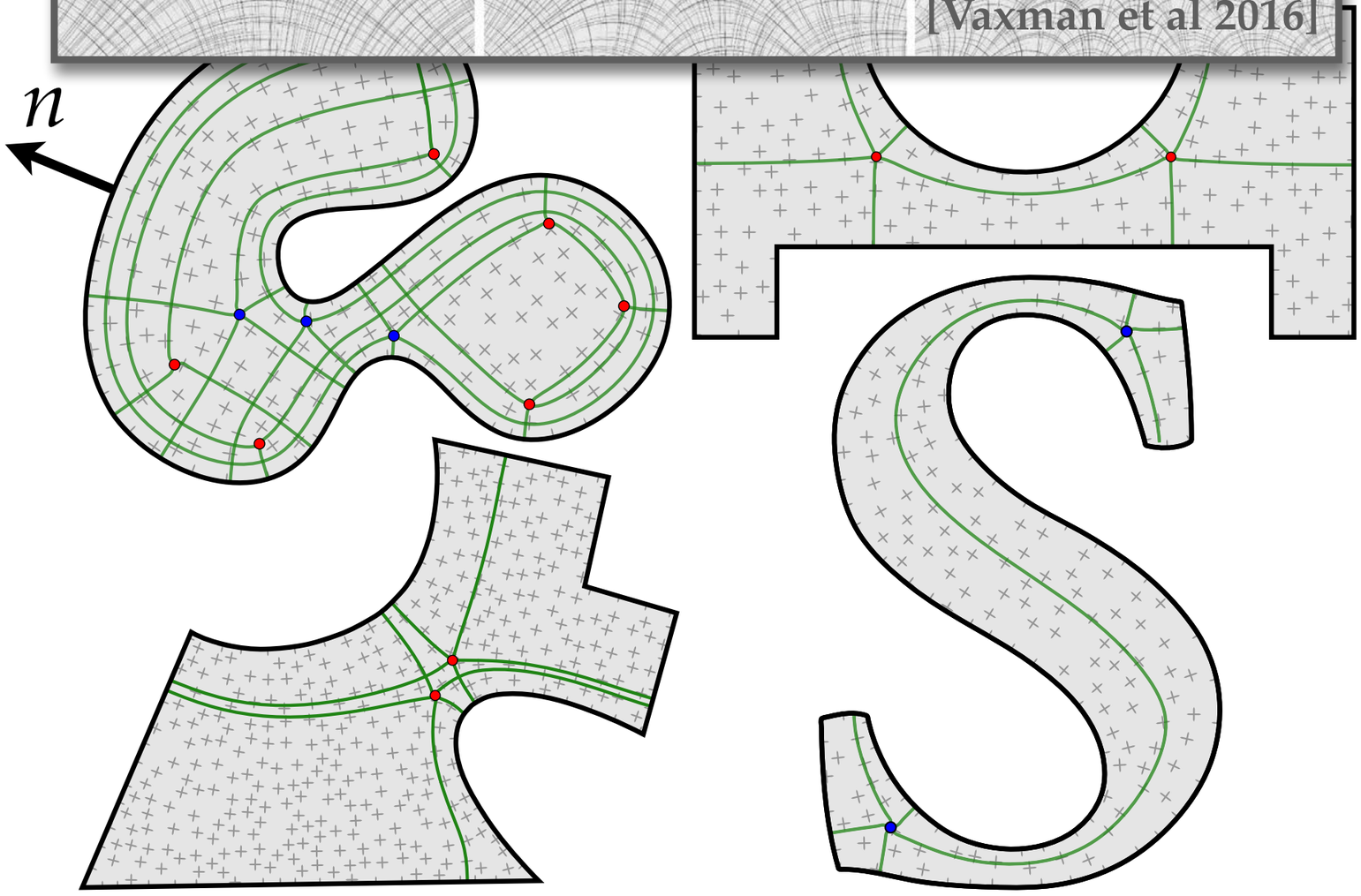
Cross Fields & Quad

FEM: "singularity party"



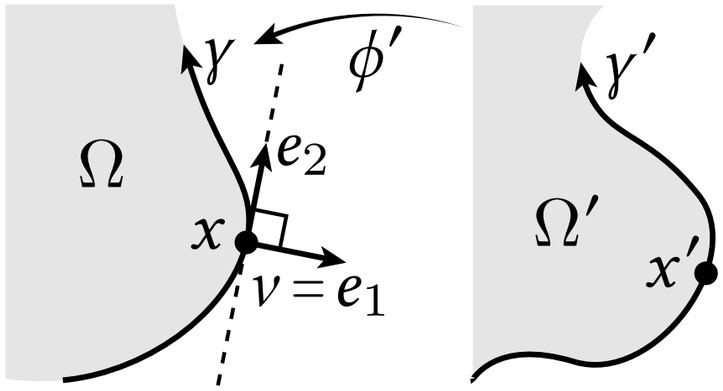
[Vaxman et al 2016]

No mesh used at any stage *(directly from Bézier curves)*



Shape Deformation (2D)

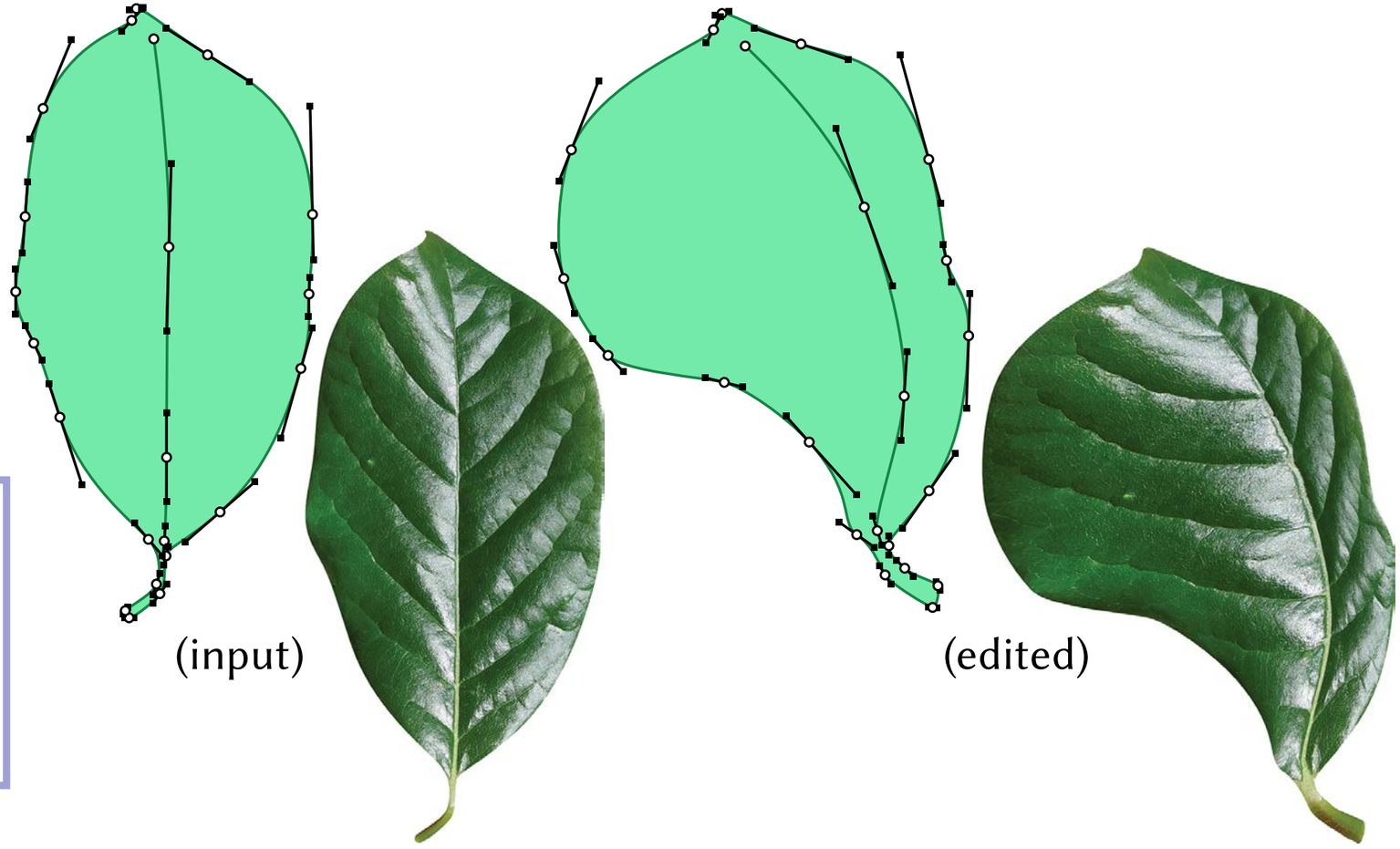
Deform images with Bézier control curves & 2nd-order boundary conditions.

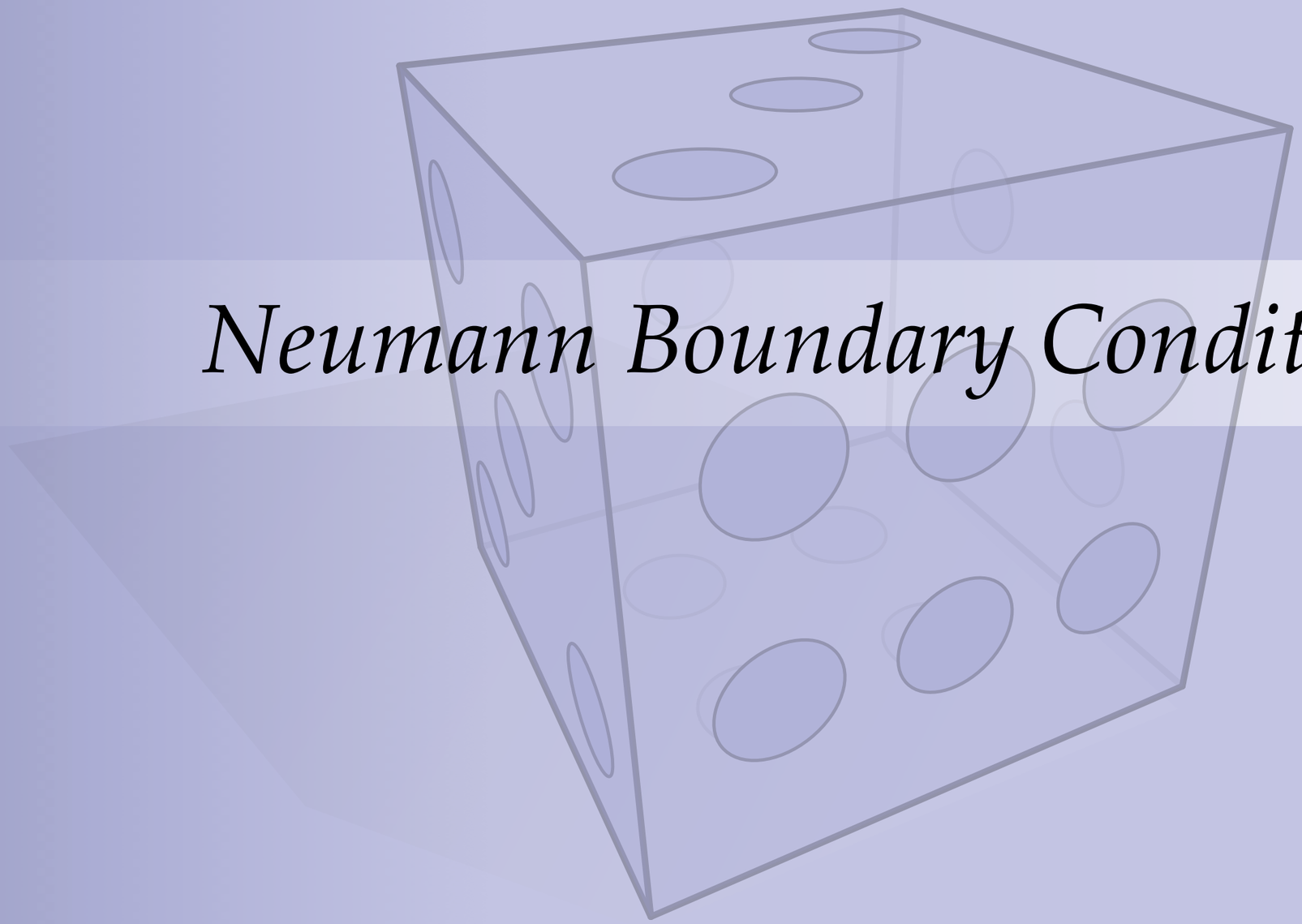


TREE WALKING ESTIMATOR

$$\begin{aligned} \Delta^2\phi' &= 0 && \text{on } \Omega \\ \phi' &= \phi && \text{on } \partial\Omega \\ \Delta\phi' &= \partial^2\phi/\partial e_2^2 && \text{on } \partial\Omega \end{aligned}$$

"as linear as possible"

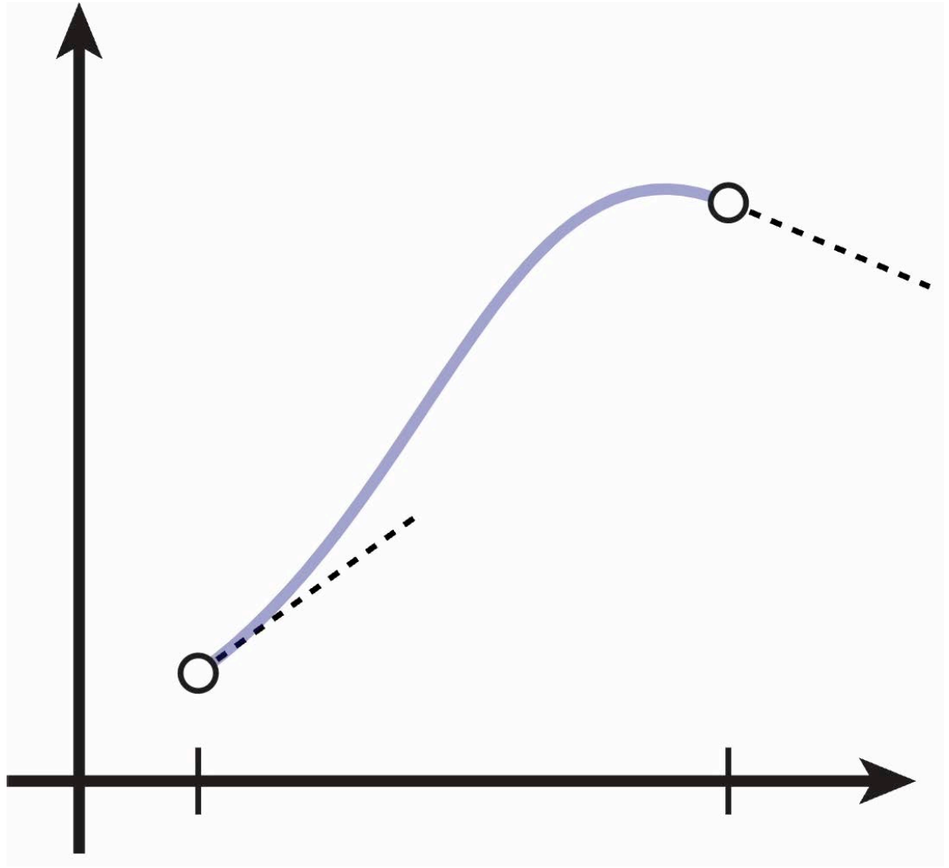




Neumann Boundary Conditions

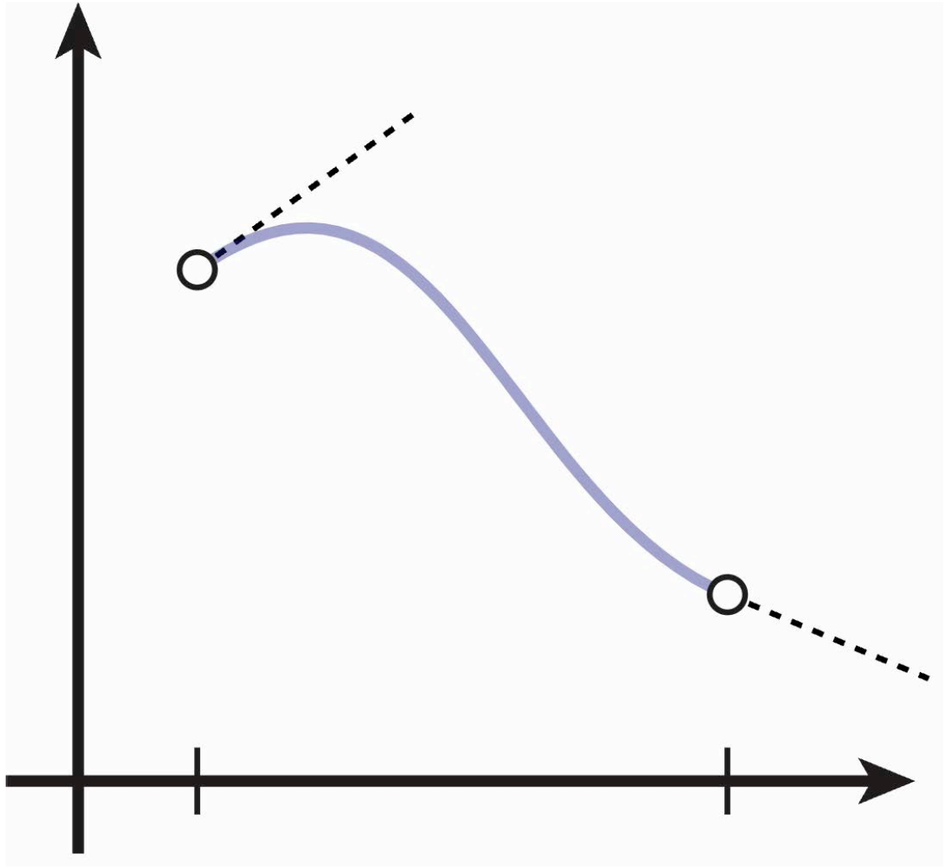
Dirichlet vs. Neumann Boundary Conditions

Dirichlet



interpolate given *values*

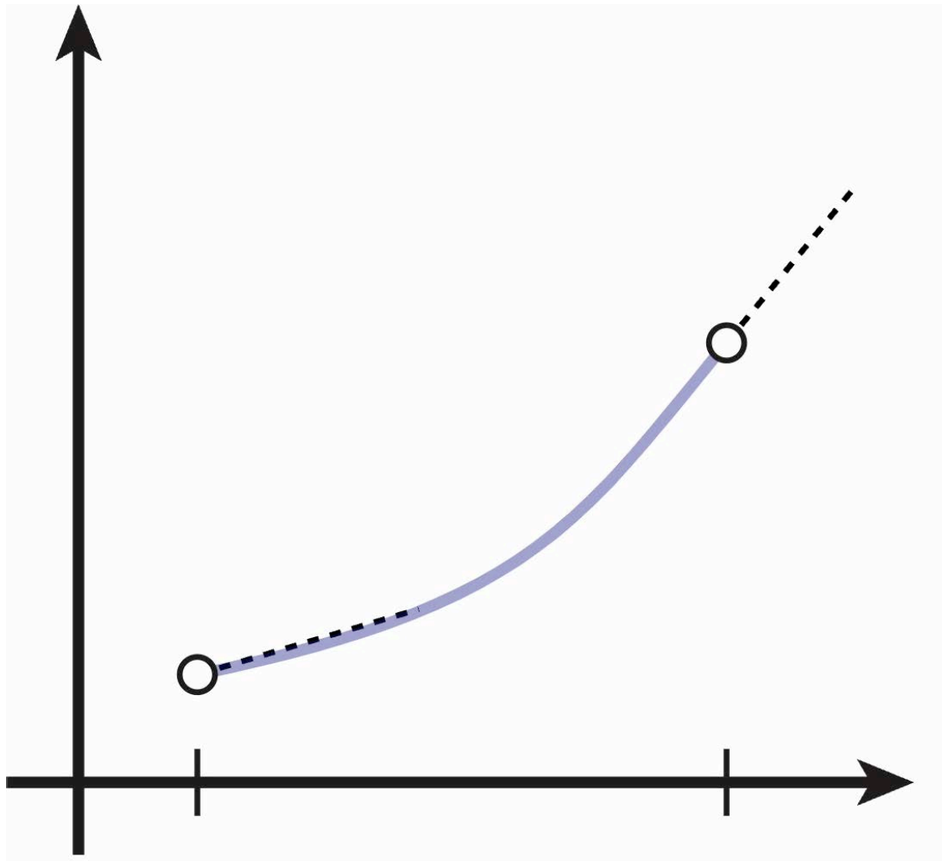
Neumann



interpolate given *derivatives*

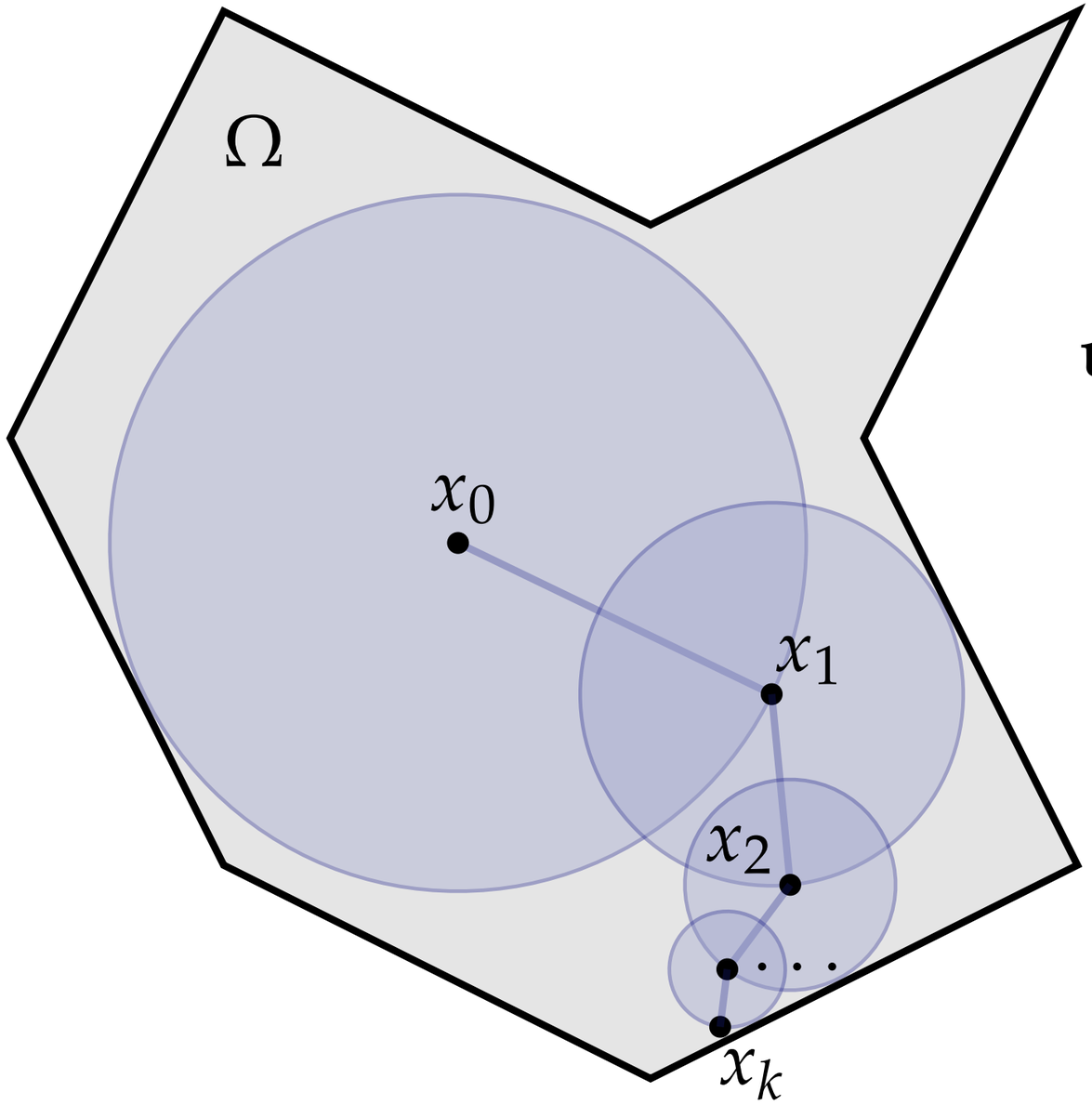
Dirichlet vs. Neumann Boundary Conditions

Mixed



fixed values *or* derivatives

Walk on Spheres—Recap



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

Laplace

Dirichlet

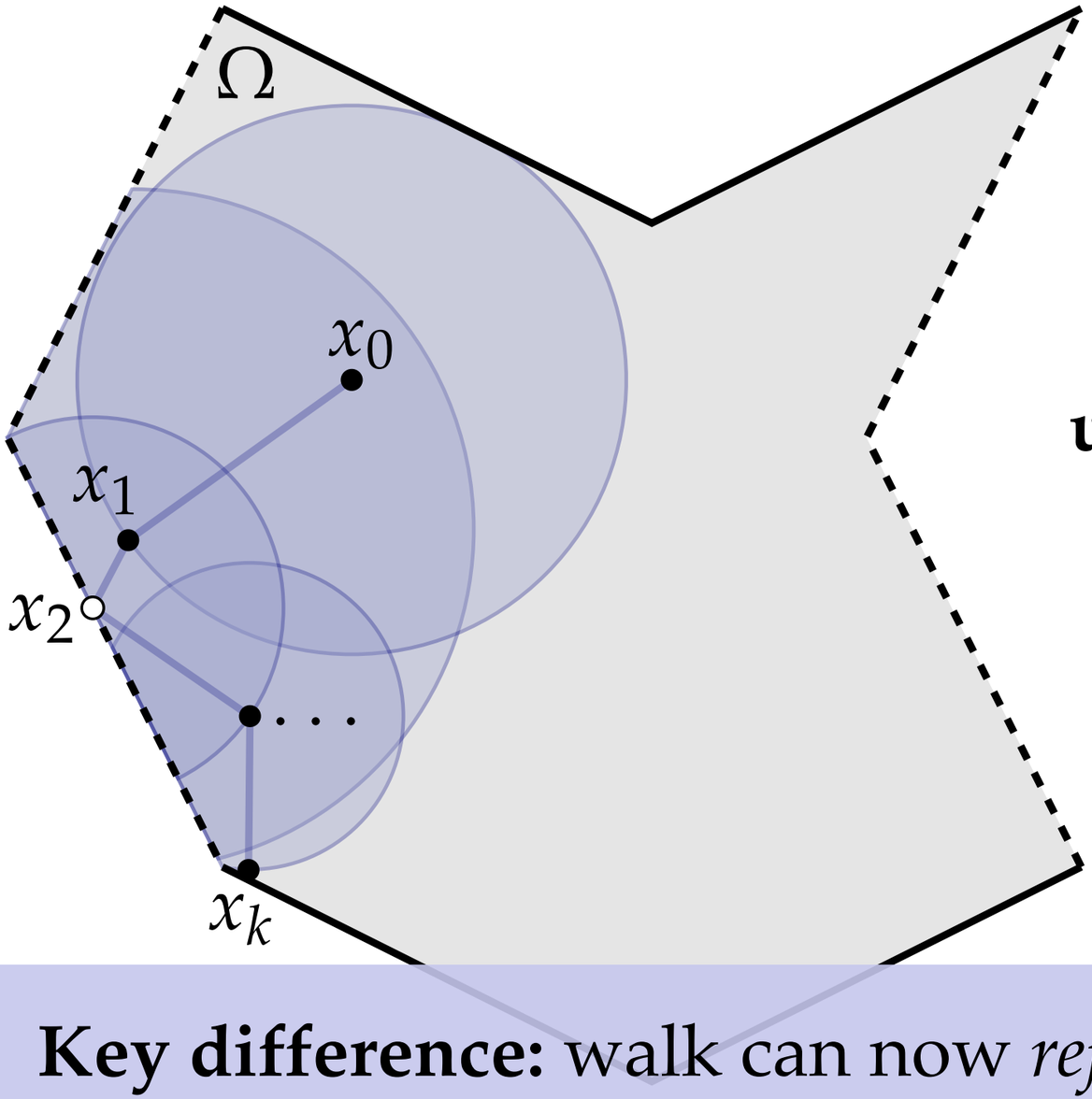
until we reach domain boundary $\partial\Omega$:

- find the largest ball B around x_k
- sample x_{k+1} from ∂B

add boundary value $g(x_k)$ to average

(repeat N times)

Walk on Stars



$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \\ u &= g && \text{on } \partial\Omega_D \\ \frac{\partial u}{\partial n} &= h && \text{on } \partial\Omega_N \end{aligned}$$

Laplace

Dirichlet ——

Neumann - - - -

until we reach Dirichlet boundary $\partial\Omega_D$:

- find star-shaped region St around x_k
- sample x_{k+1} from ∂St

add boundary value $g(x_k)$ to average

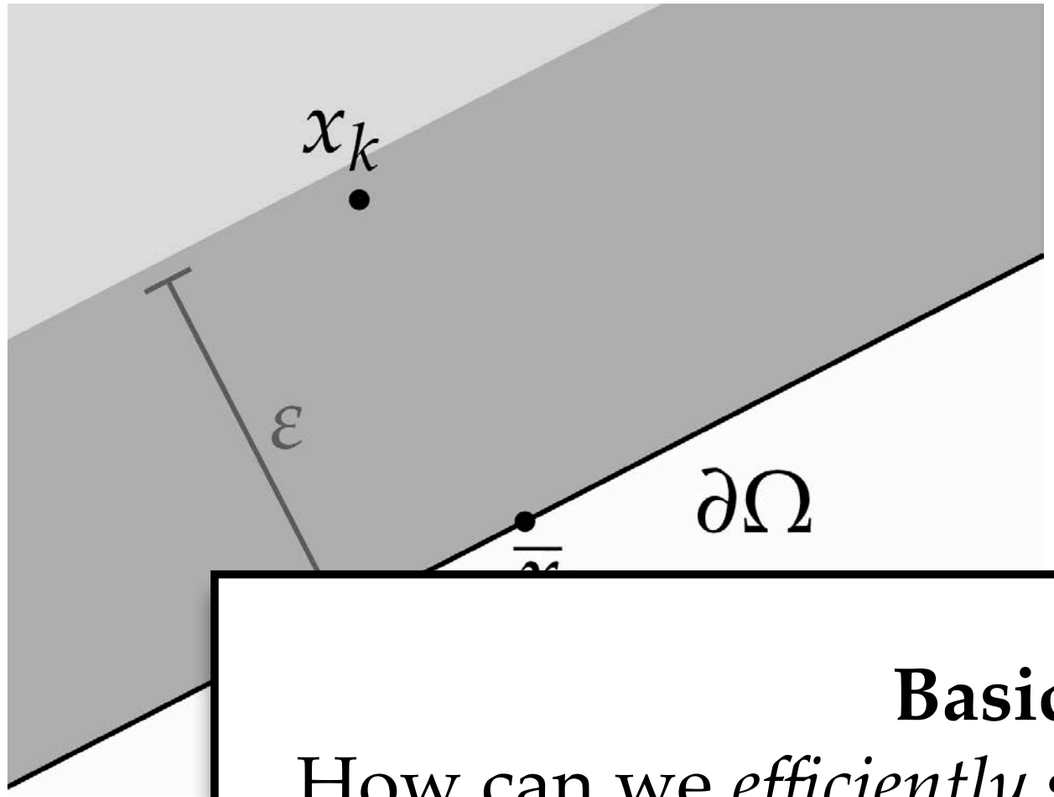
(repeat N times)

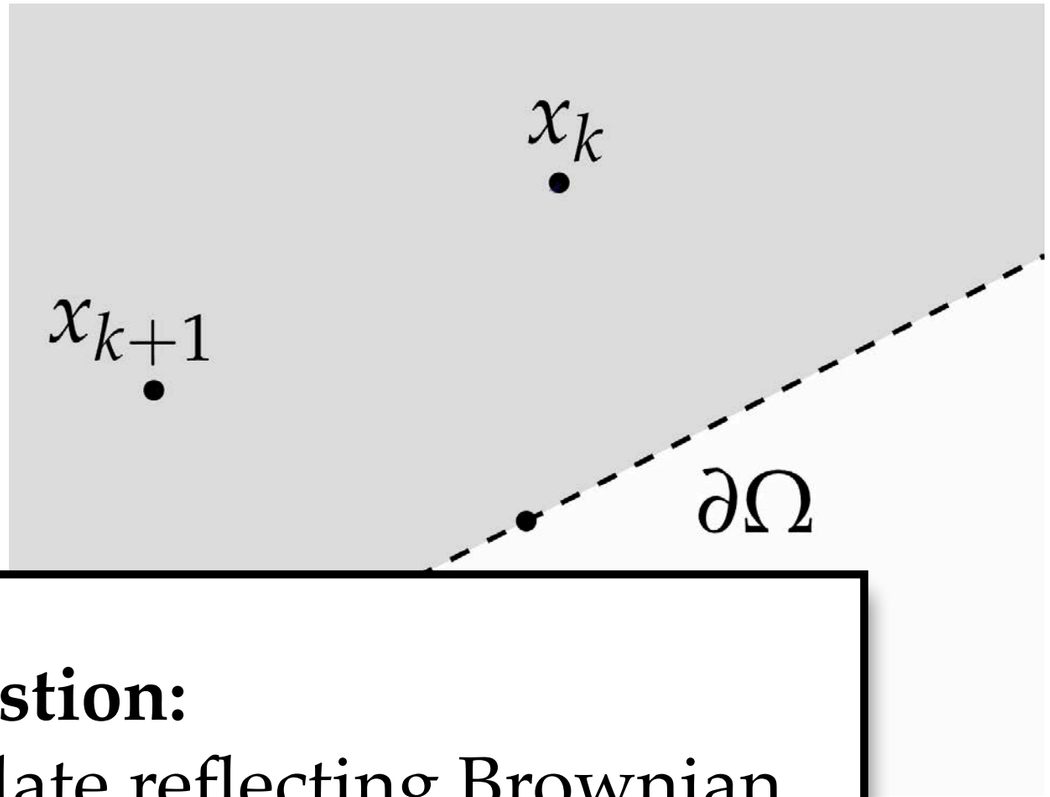
Key difference: walk can now *reflect* off Neumann boundary

Absorbed vs. Reflected Brownian Motion

Dirichlet ↔ absorbed

Neumann ↔ reflected



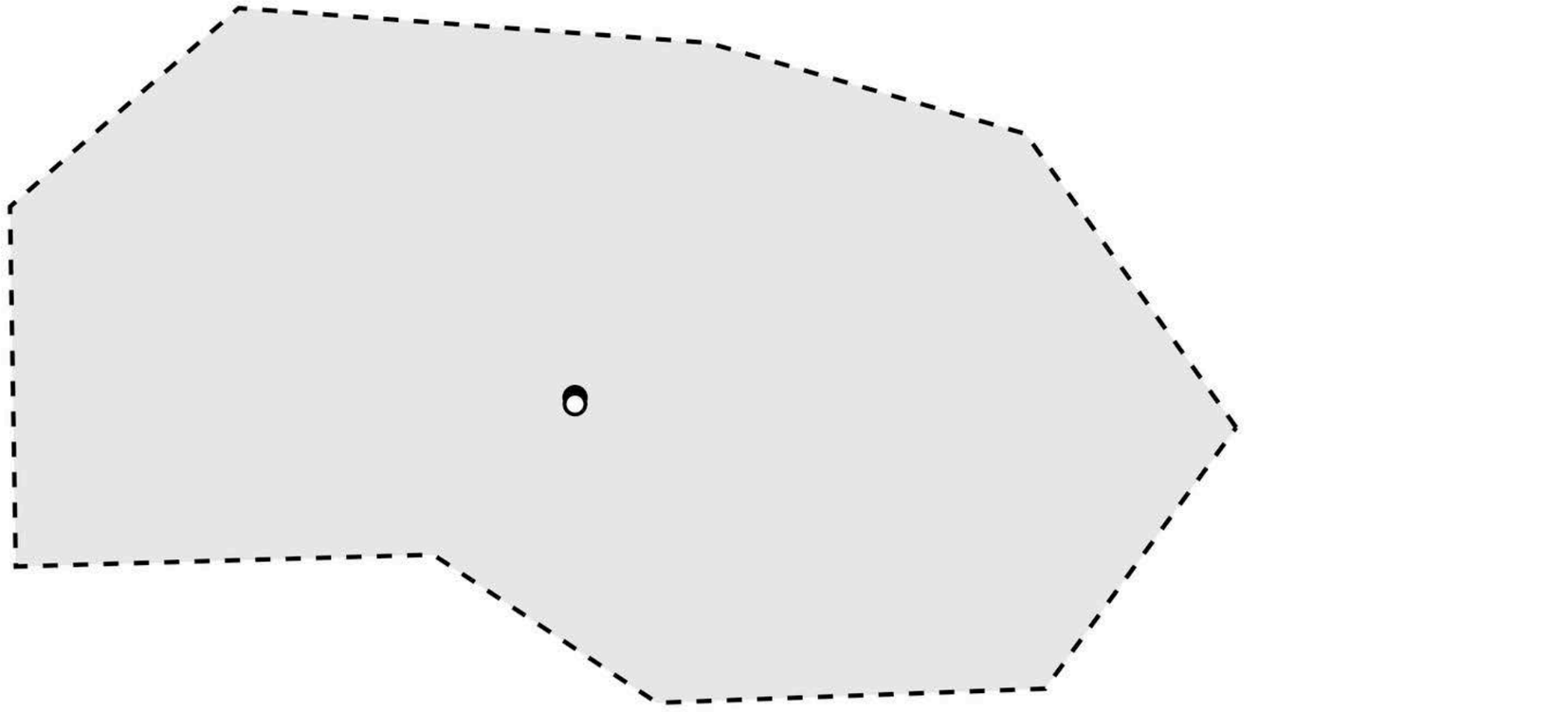


Basic question:
How can we *efficiently* simulate reflecting Brownian motion on domains with <u>complex geometry</u>?

Reflecting Brownian Motion — 1D

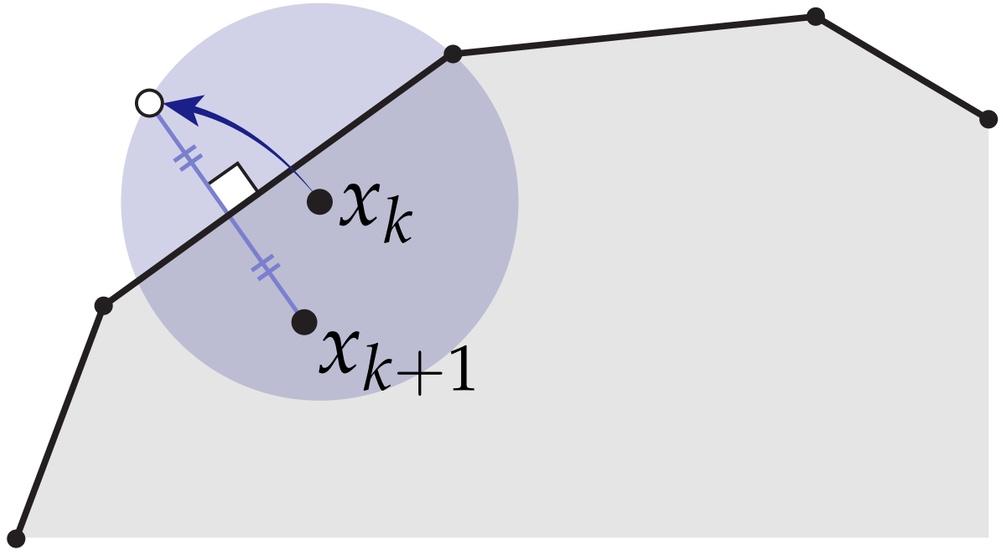
$$Y_t := |X_t|$$

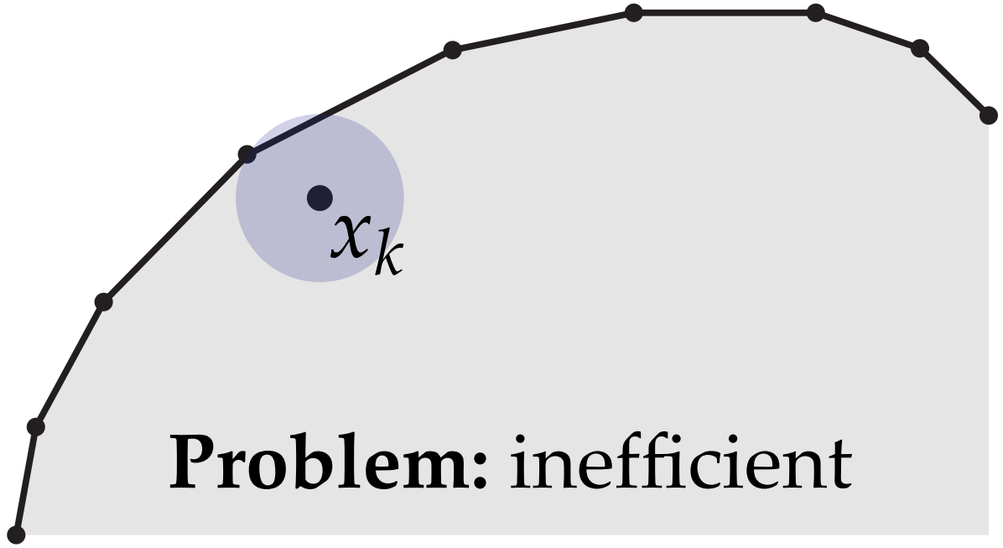
Reflecting Brownian Motion — nD / polyhedral



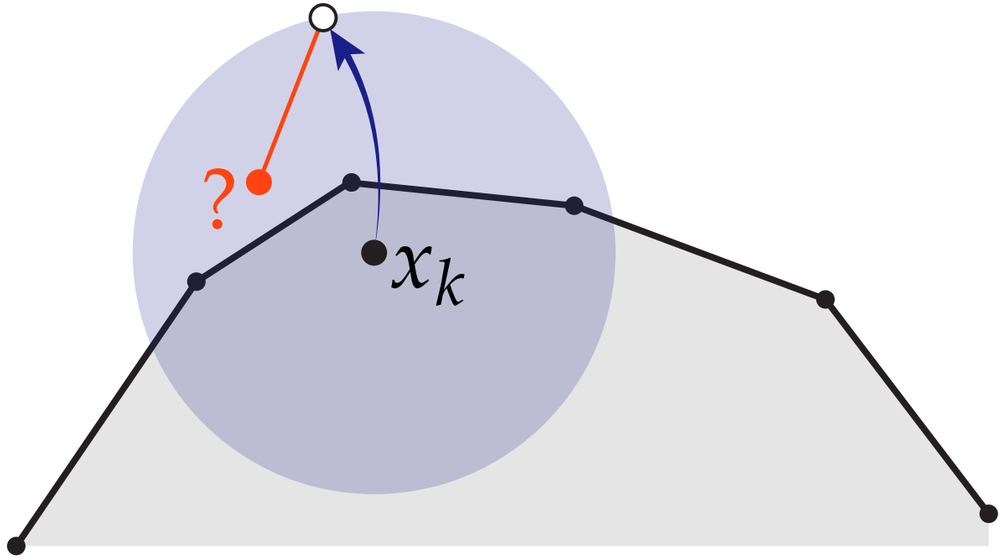
Naïve Simulation of Reflecting Brownian Motion

Naïve idea: sample biggest sphere around x crossing just one edge; reflect if necessary.





Problem: inefficient

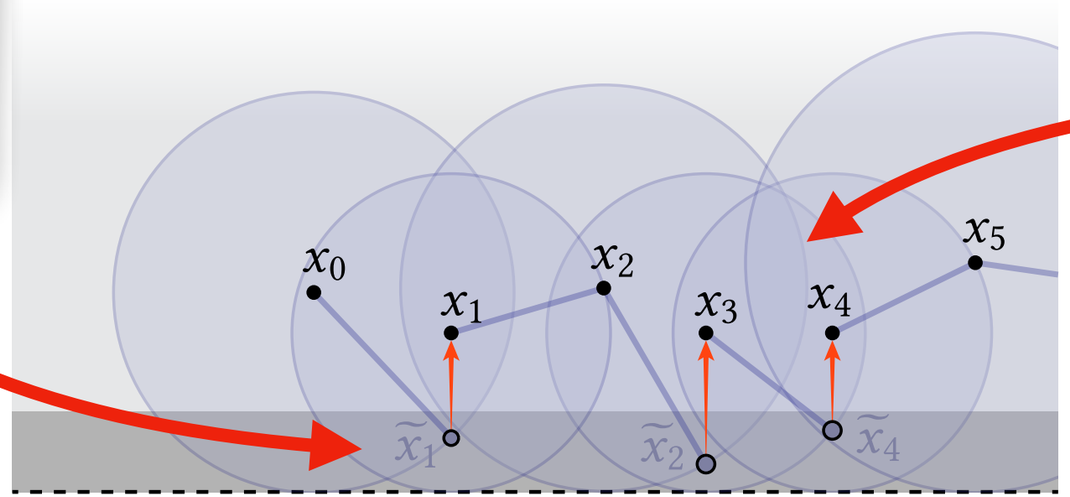


Problem: incorrect / biased

"Nudging" Walks Back Into Domain

Also tempting: *"nudge"* walk back into domain if we get too close to Neumann part of the boundary

accumulate error with each nudge *(biased solution)*



walks "cling" to Neumann boundary *(very long walks)*

ε-shell

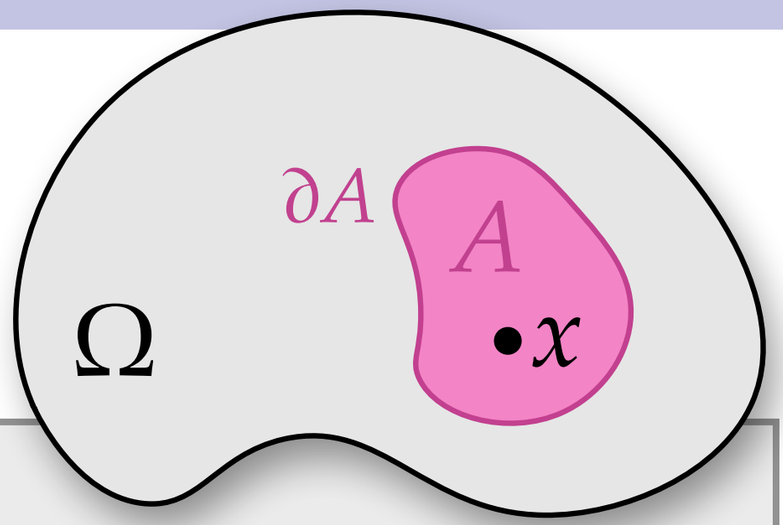
$\partial\Omega_N$

Mascagni & Simonov, *"Monte Carlo Methods for Calculating Some Physical Properties of Large Molecules"* (2004)

Boundary Integral Equation

LAPLACE EQUATION

$$\Delta u = 0 \quad \text{on } \Omega$$



BOUNDARY INTEGRAL EQUATION

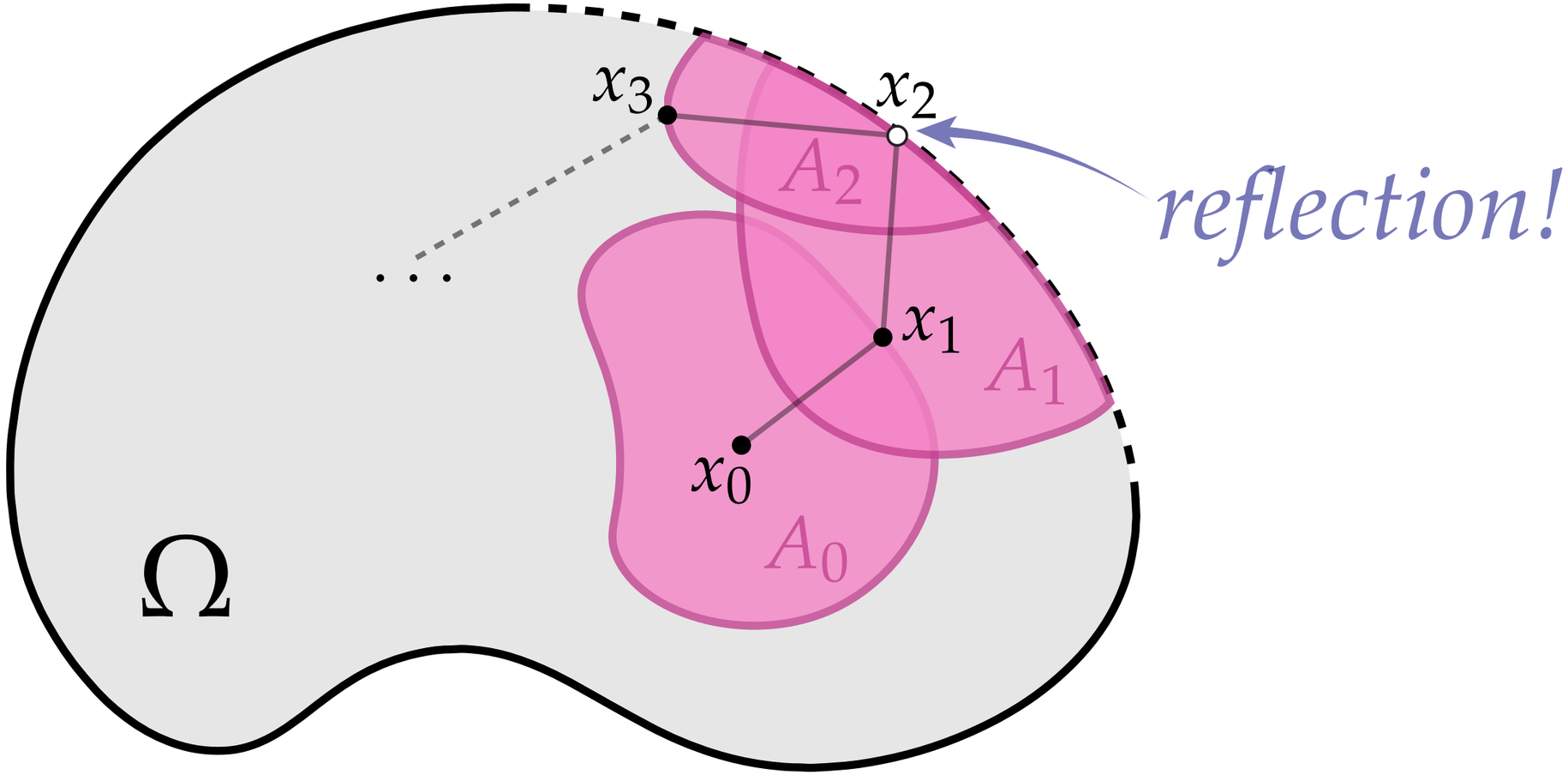
unknown value at x | Poisson kernel of A | unknown value at y

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

Intuition: "generalized mean value principle"

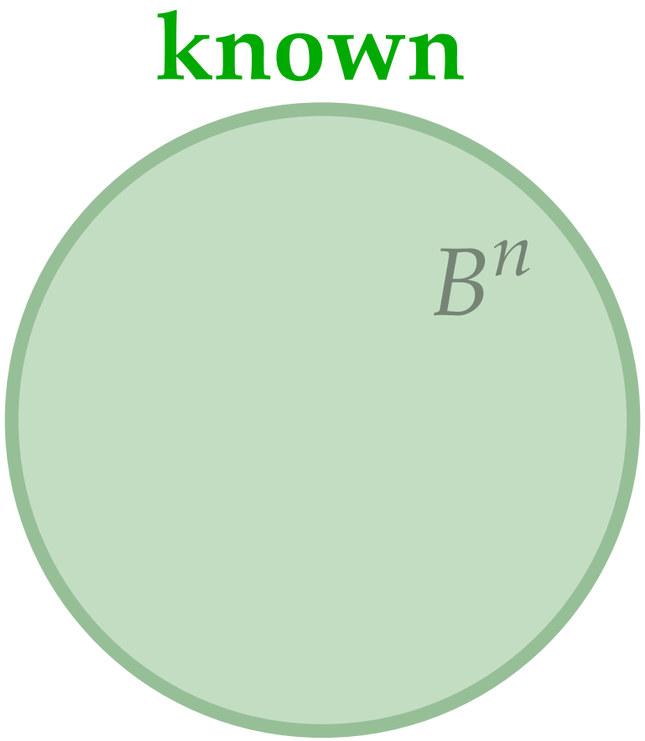
Walk on Subdomains

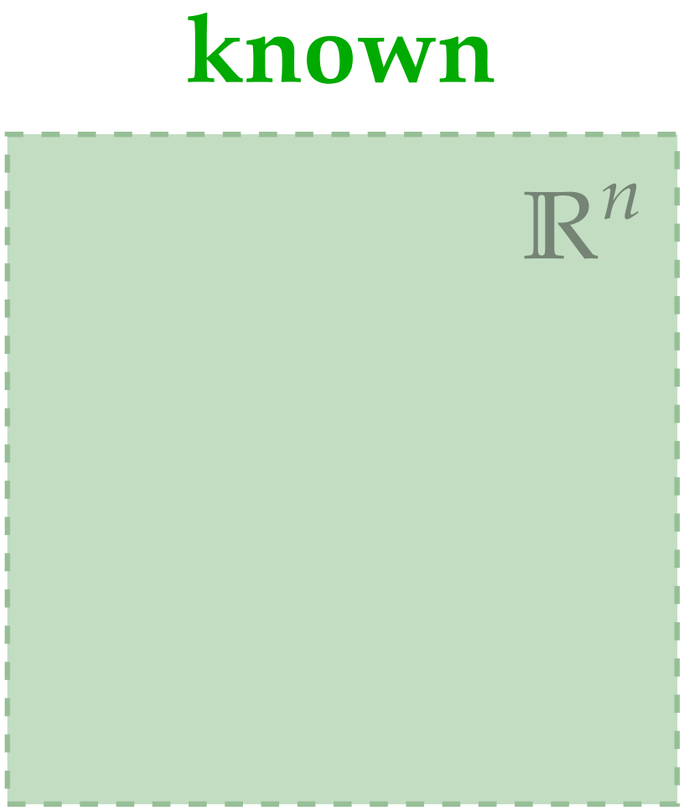
Idea: take a "walk" on subdomains that can *contain pieces of the Neumann boundary* (by sampling Poisson kernel)

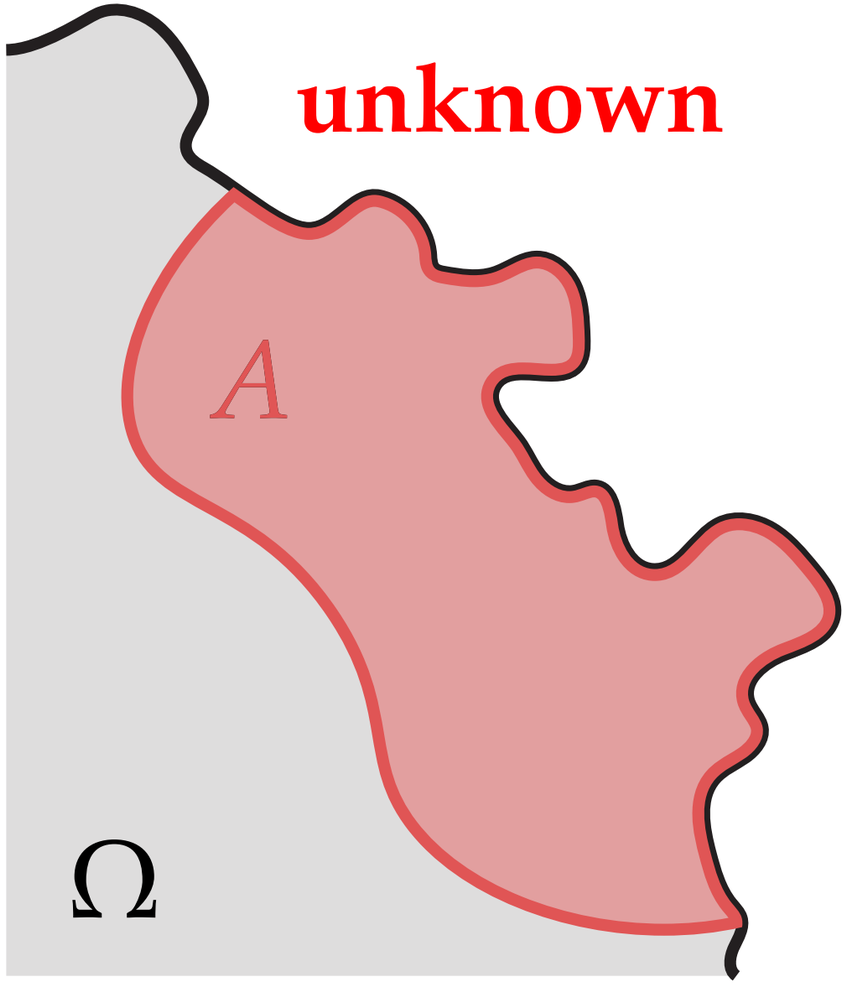


Sampling the Poisson Kernel

Problem: Poisson kernel known only for very simple regions

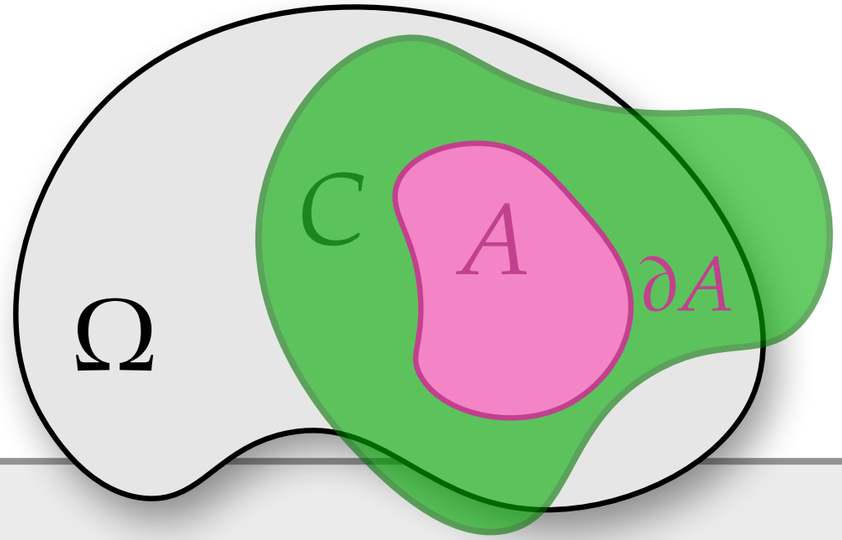






Boundary Integral Equation (General)

$$\Delta u = 0 \quad \text{on } \Omega$$
$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega$$



Boundary Integral Equation (general form)

unknown value at x

Poisson kernel of C (not A)

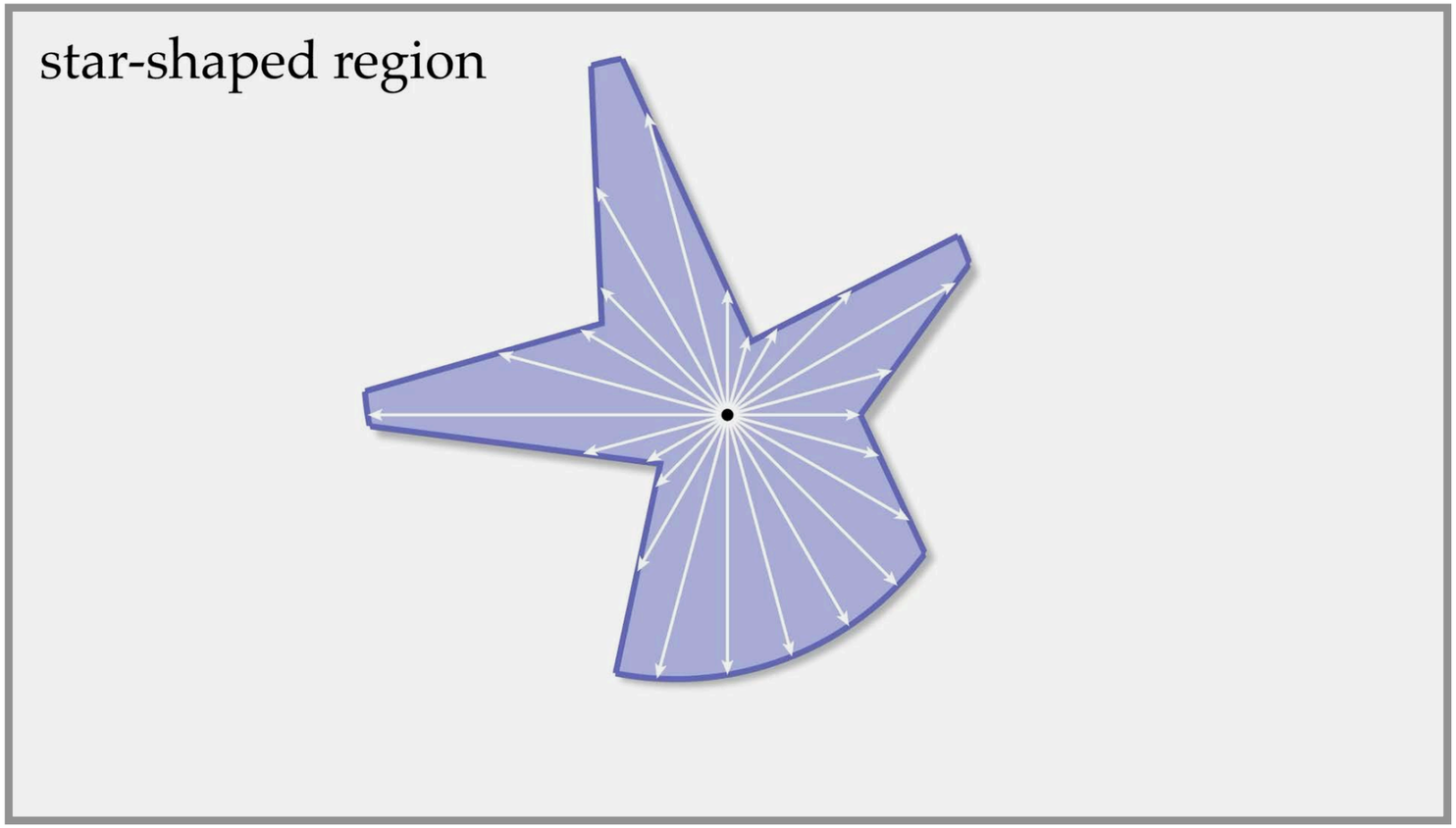
unknown value at y

$$u(x) = \int_{\partial A} P^C(x,y)\,u(y)\,dy$$

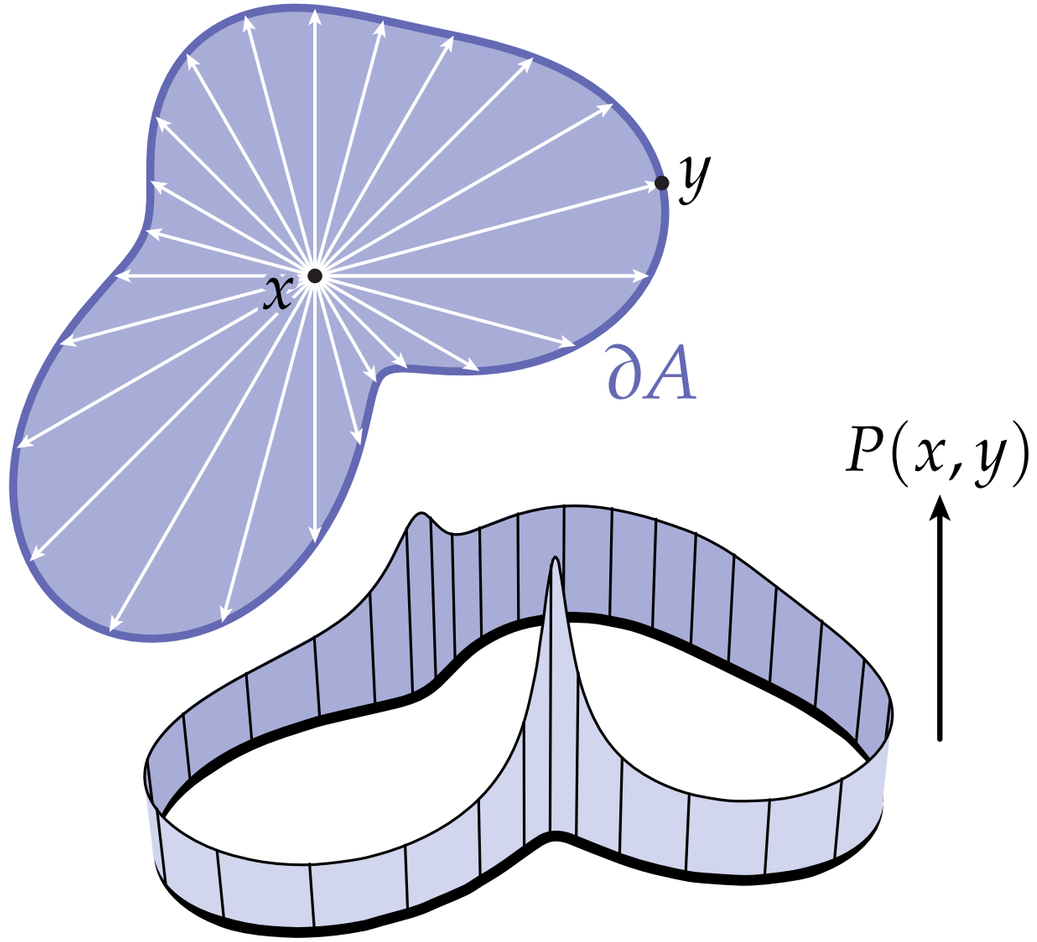
integrate over boundary of A (not C)

Walk on Stars

$A =$ **star-shaped region**



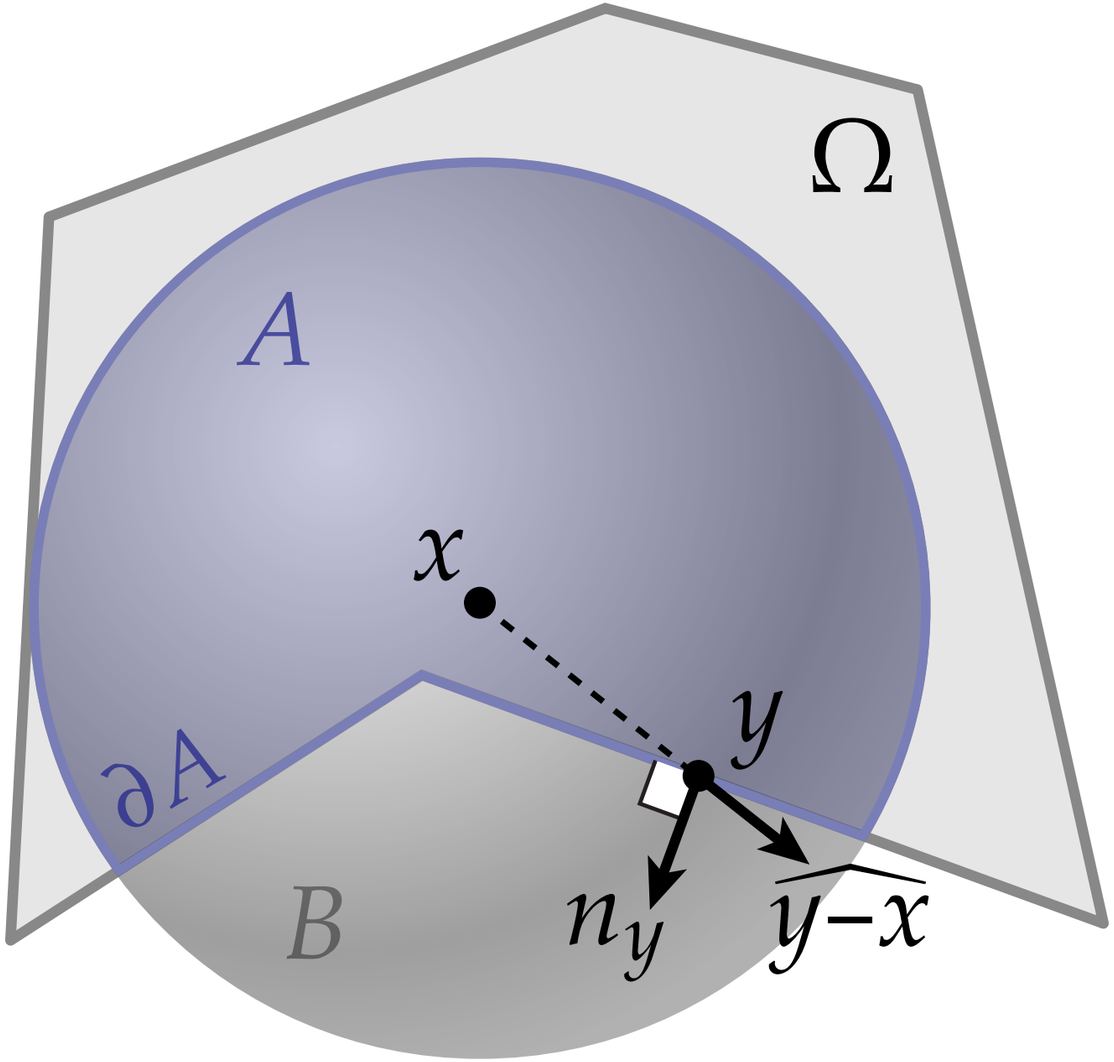
$C =$ **ball**



Poisson kernel from ball, restricted to ∂A

Sampling Star-Shaped Regions

Why use star-shaped region for A (and ball B for Poisson kernel)?



$$P^B|_{\partial A} = \frac{n_y \cdot \widehat{y-x}}{4\pi r^2}$$

normal at y

direction toward y

distance from x to y

signed solid angle

Signed Solid Angle

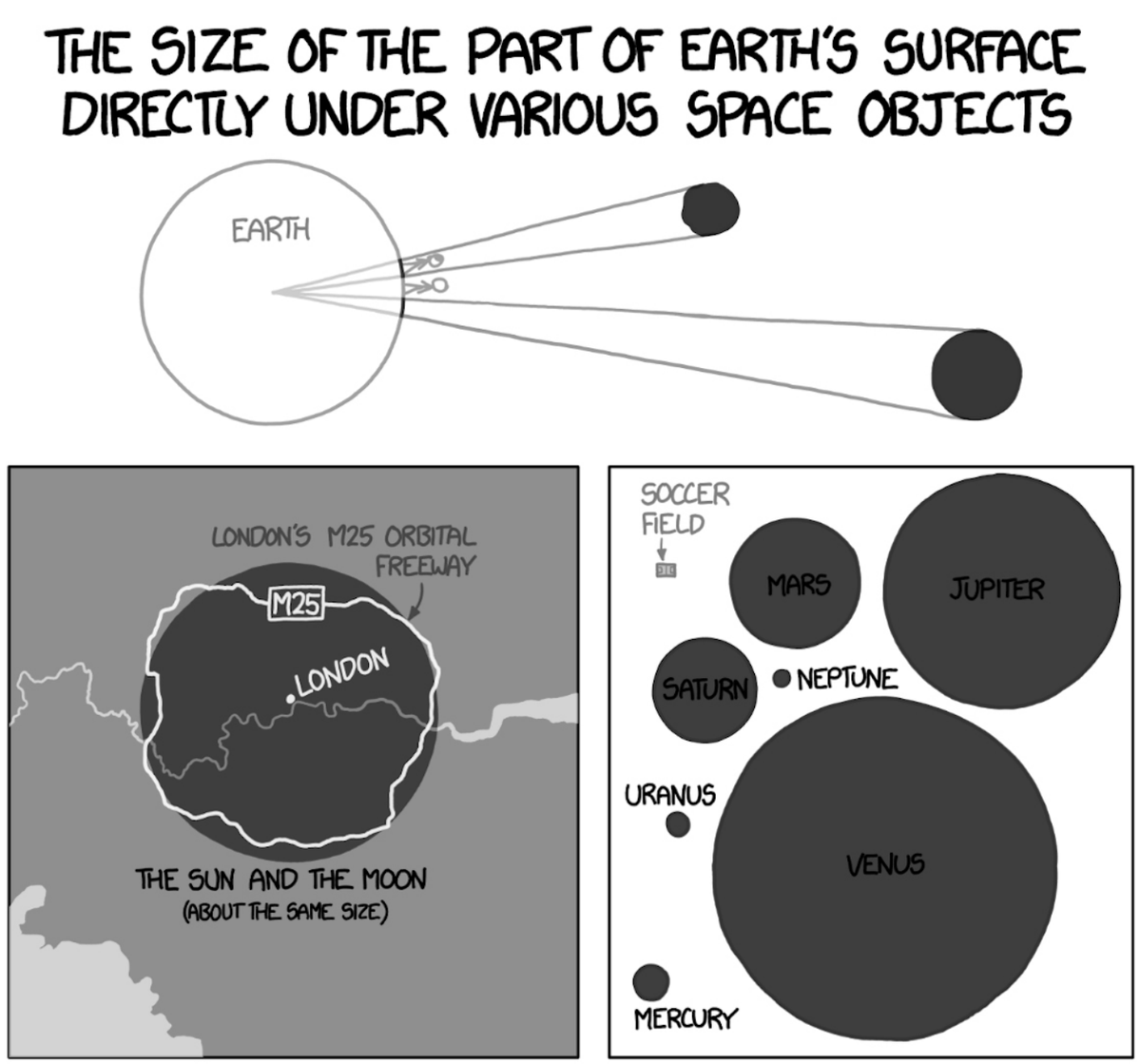
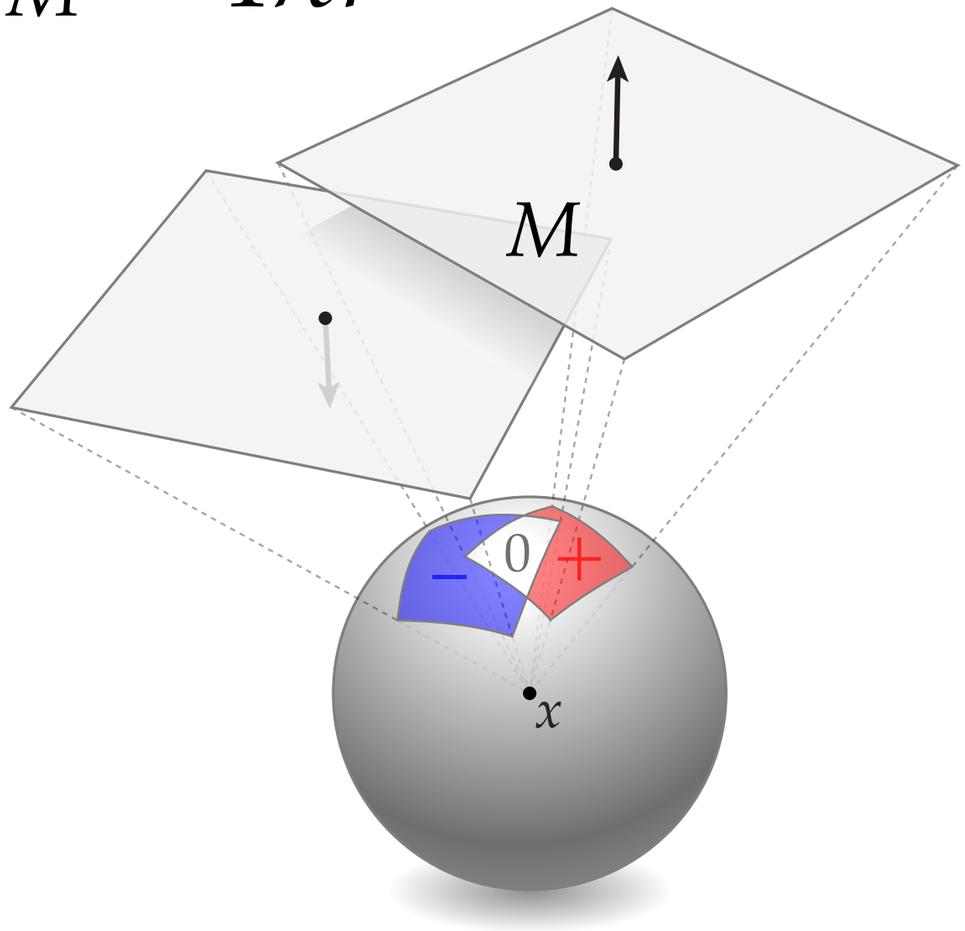


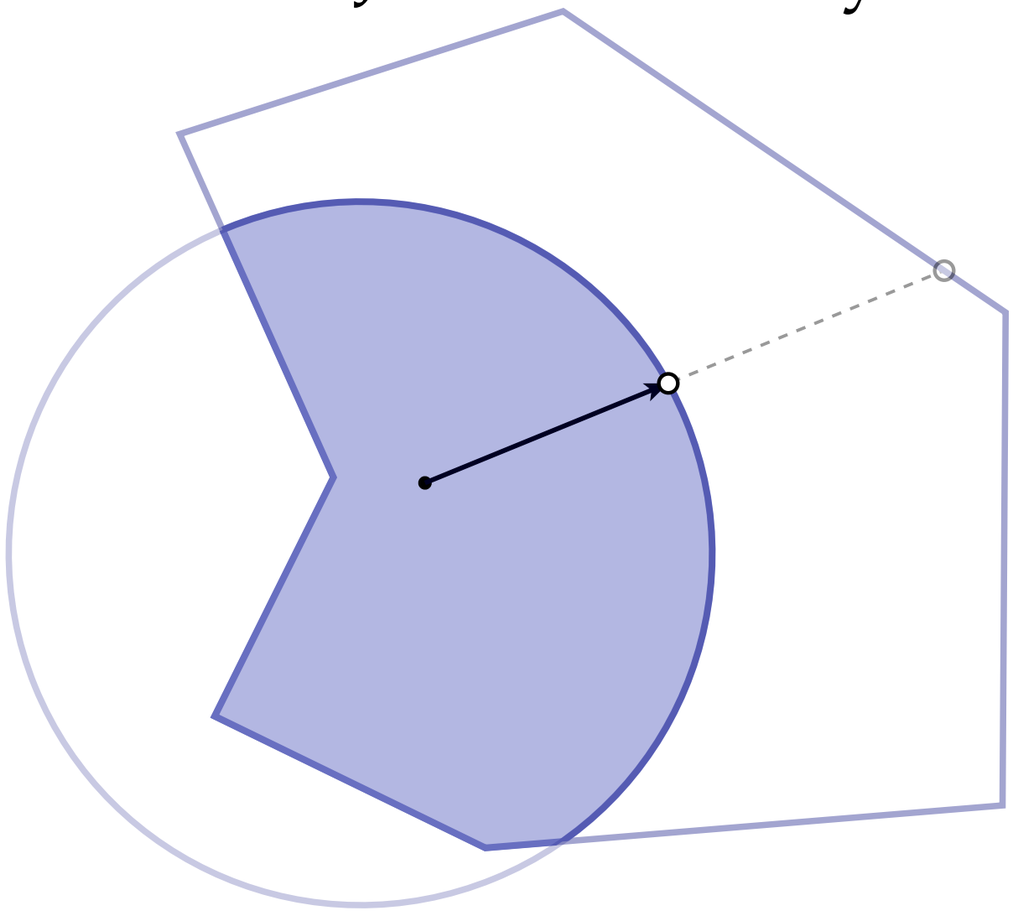
image credit: Randall Monroe / xkcd

$$\int_M \frac{n_y \cdot \widehat{y - x}}{4\pi r^2} \, dA_M$$

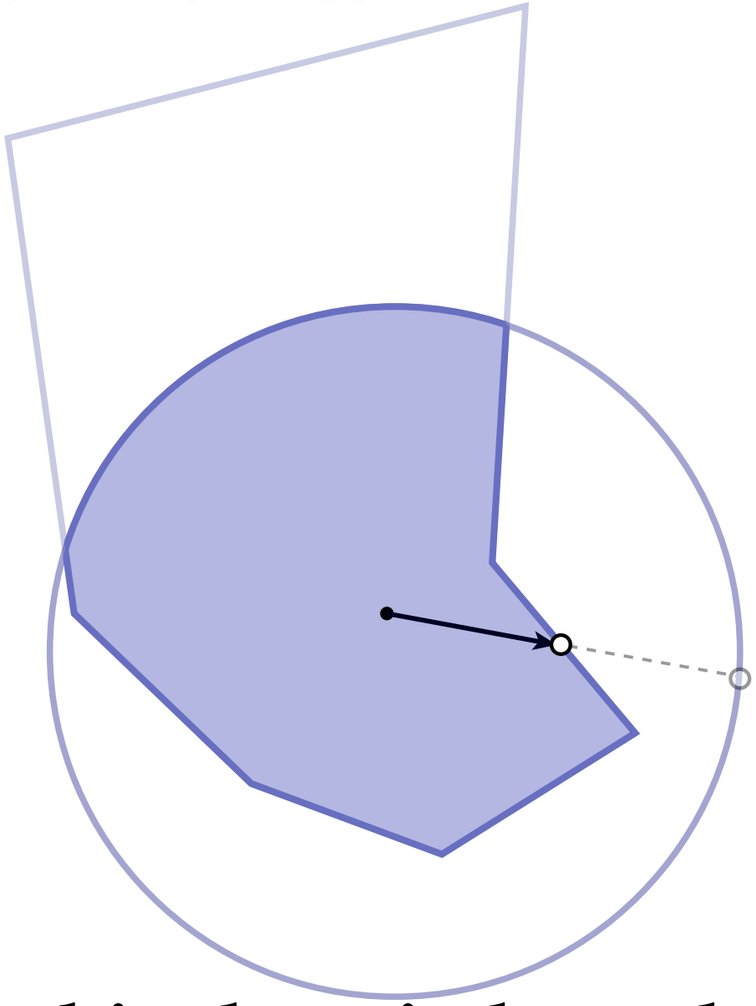


Sampling Poisson Kernel

Just shoot a ray in a random direction.



first hit: ball

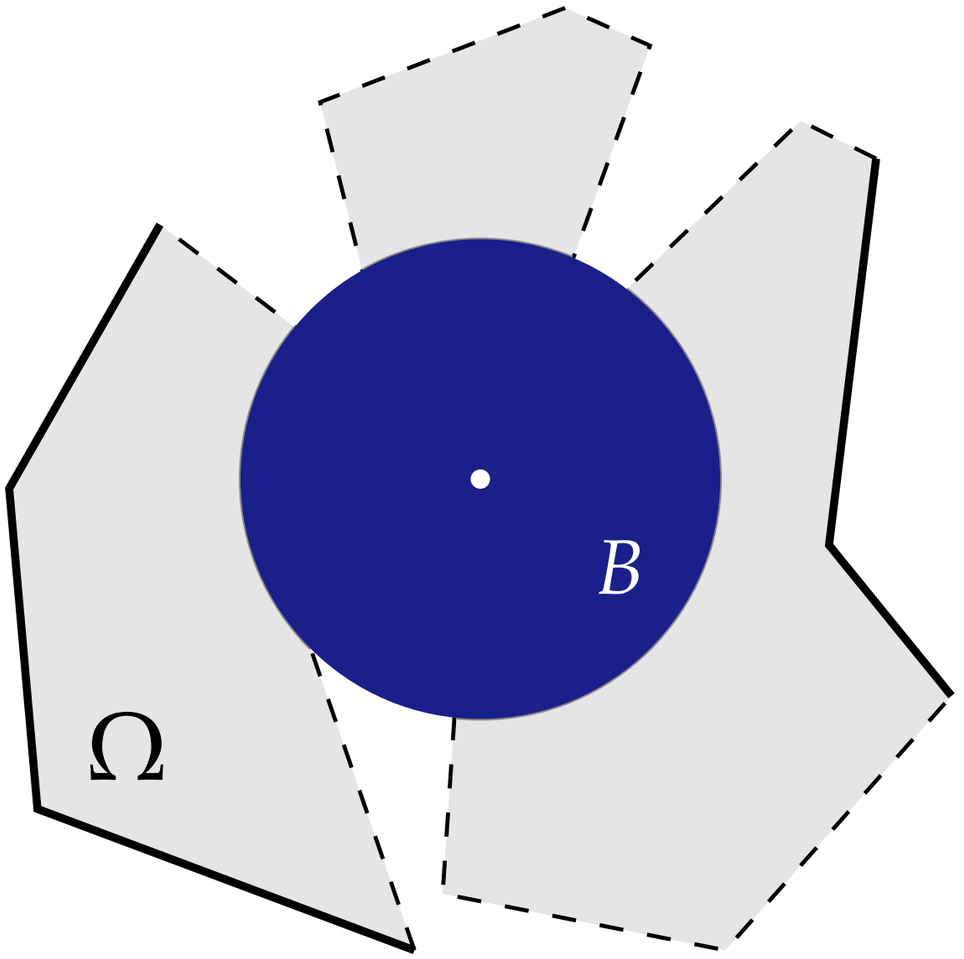


first hit: domain boundary

[Veach & Guibas 1995]

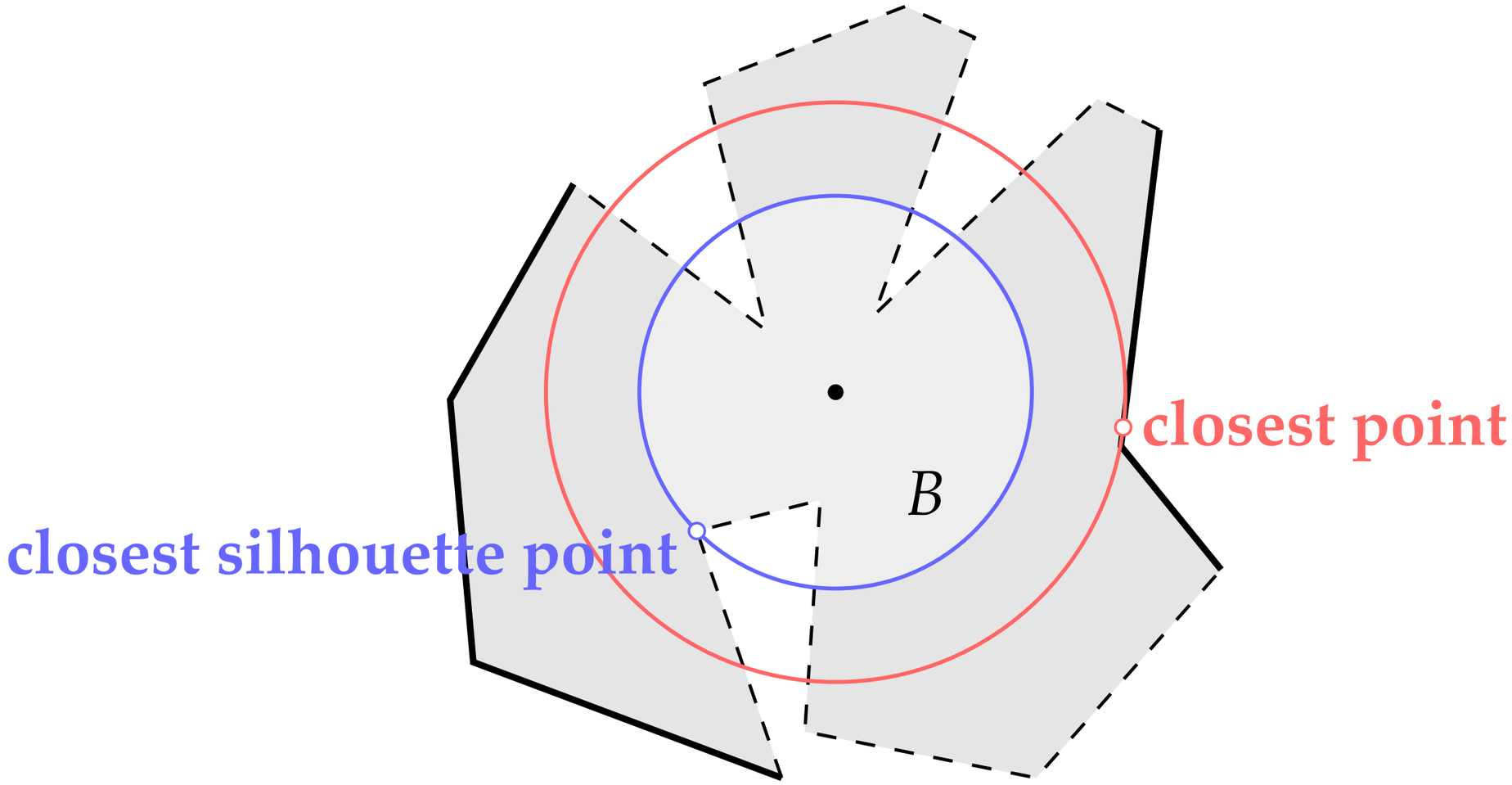
Finding Star-Shaped Regions

How do we find <u>big</u> star-shaped regions?

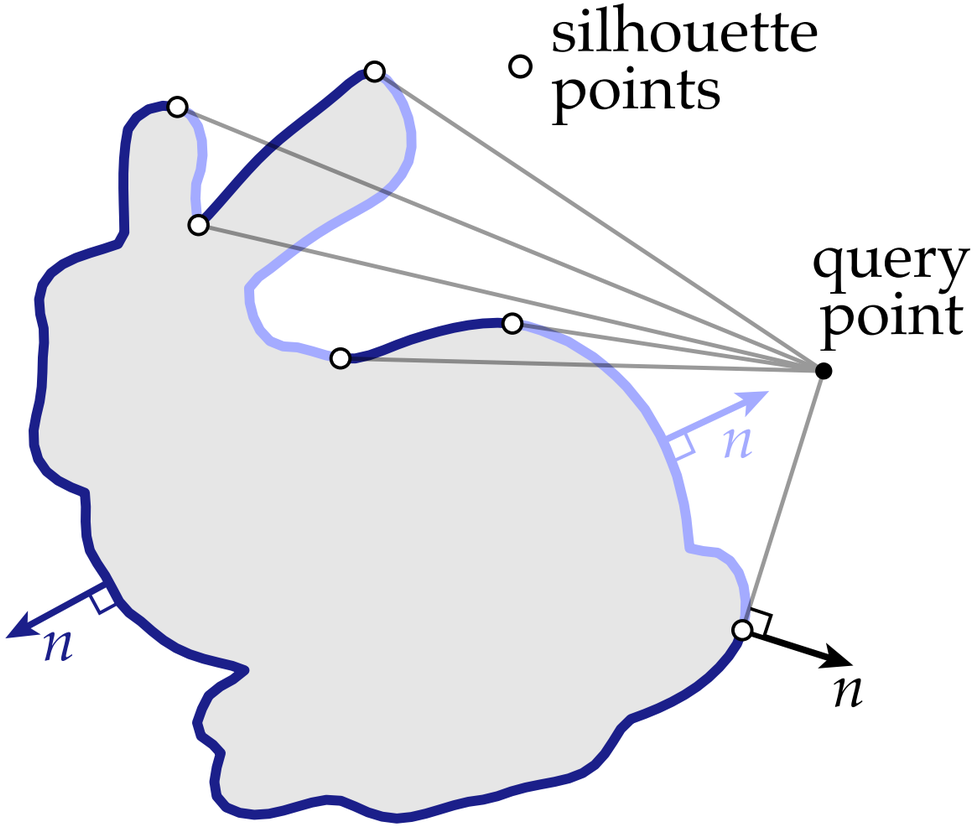


Finding Star-Shaped Regions

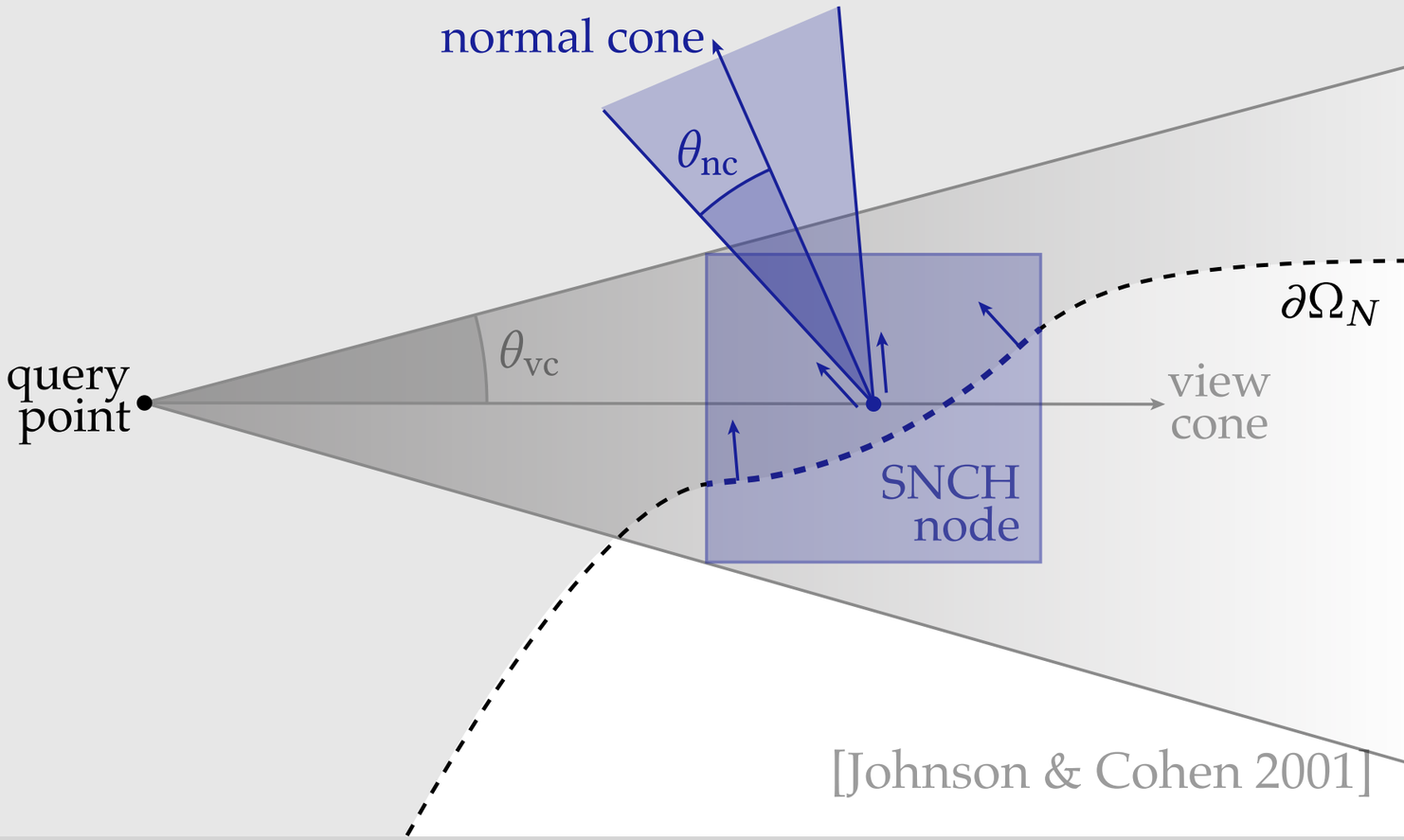
Surprisingly easy to do: take min distance to (i) Dirichlet boundary and (ii) *silhouette* of Neumann boundary.



Closest Silhouette Point Queries



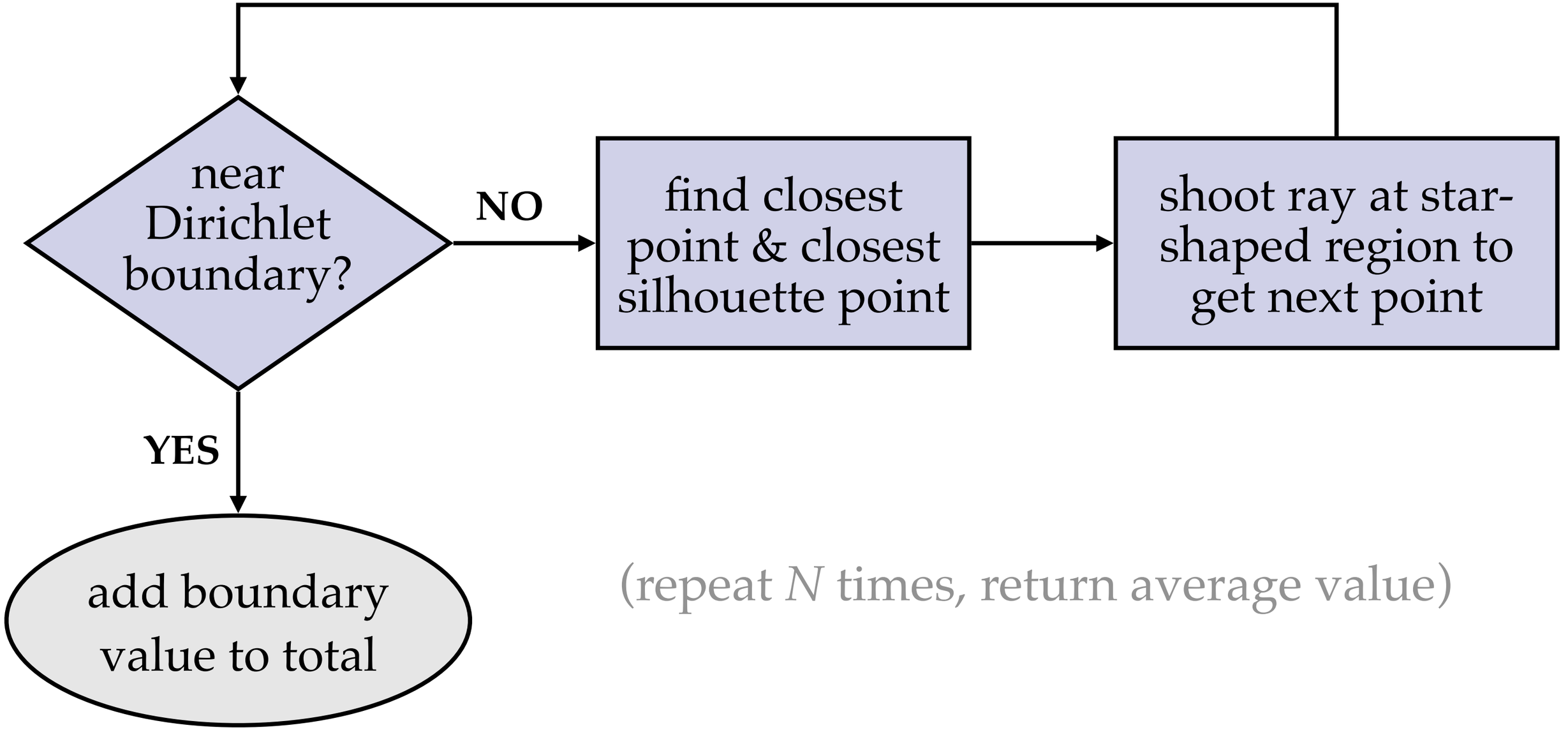
normal cone hierarchy



[Johnson & Cohen 2001]

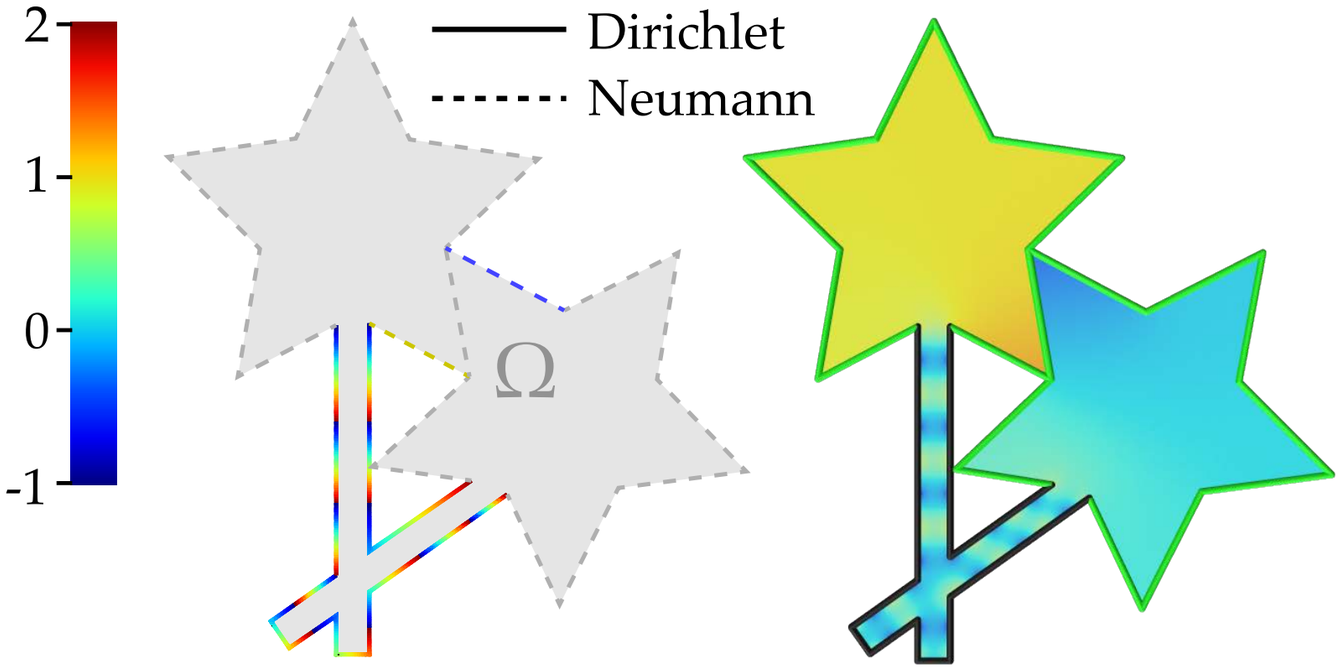
Can re-use same BVH already built for closest point queries.

Walk on Stars (Laplace, Dirichlet + zero-Neumann)



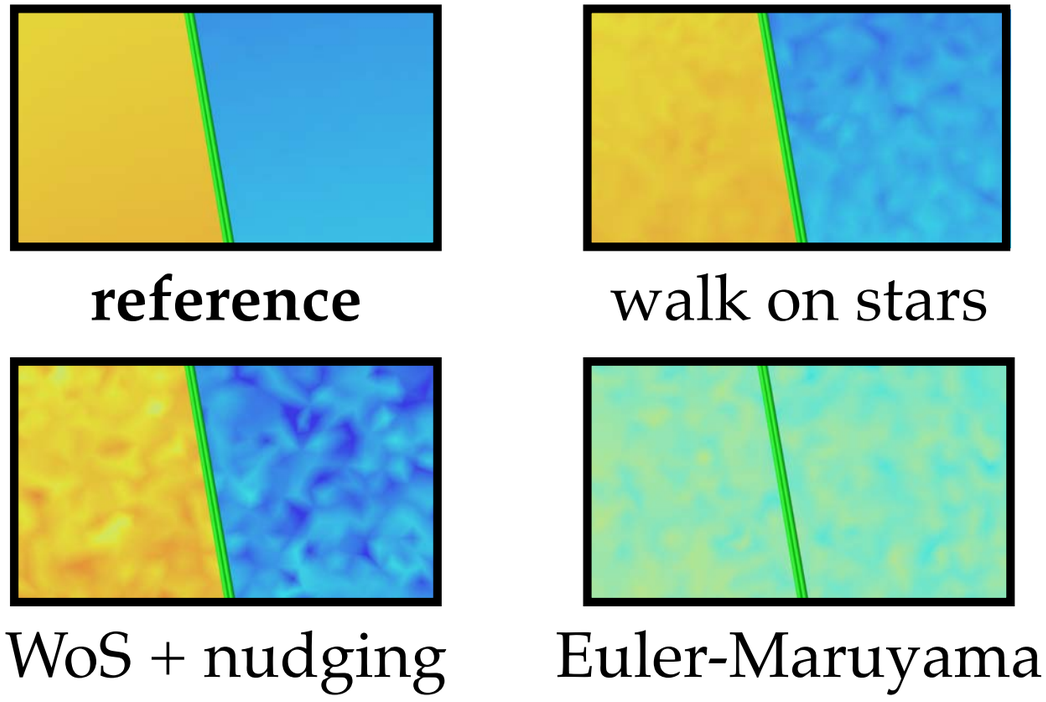
(repeat N times, return average value)

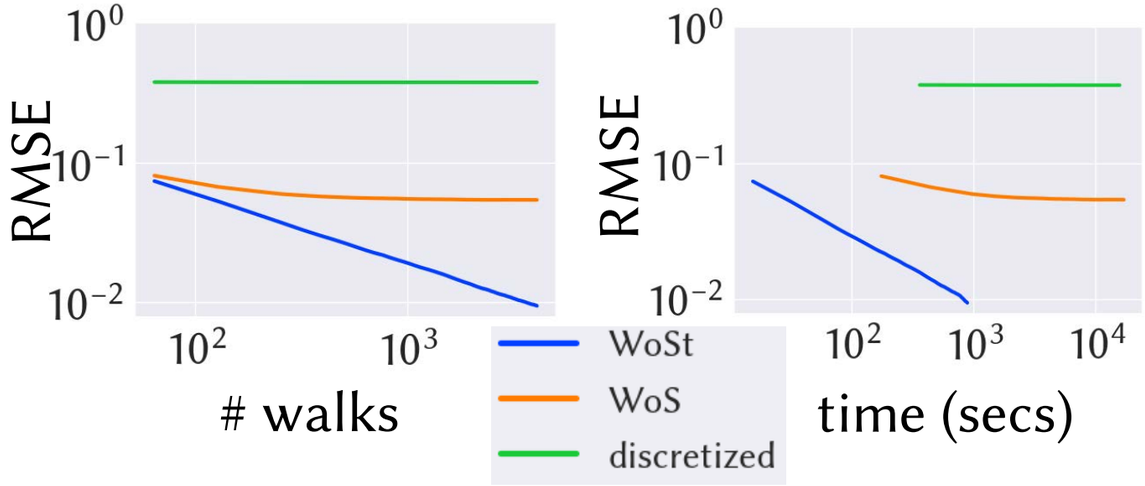
Comparison to Alternatives



$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \\ u &= g && \text{on } \partial\Omega_D \\ \frac{\partial u}{\partial n} &= h && \text{on } \partial\Omega_N \end{aligned}$$

same time





Radiative Transfer

THE HISTORY OF

Fourier and the Heat Equation

In the early 19th century, the French mathematician Jean Baptiste Joseph Fourier (1768–1830) developed a formula that describes how heat travels through solids by conduction. Now known simply as the heat equation, Fourier's

substances having different heat-transfer properties. Heat moves differently in muscle, bone, and fat, for example. Bread is sometimes more homogeneous, but the internal structure of the crumb is horrendously complex. It would

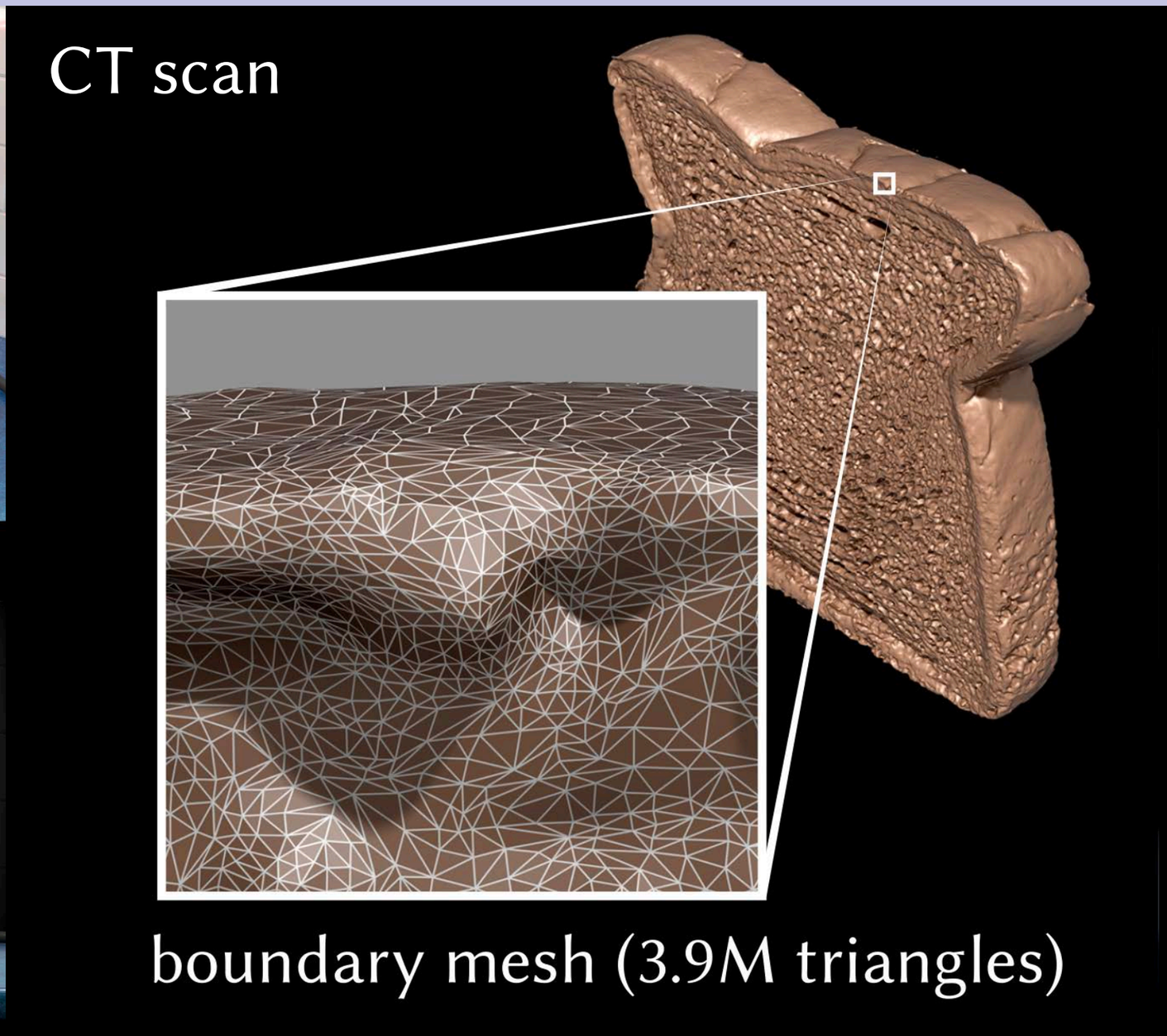
"The heat equation helps to answer a question: is it done yet? Or rather, it could, if only the complexity of food did not defy our ability to model it mathematically. ... It would take extraordinary effort to represent such intricate, highly-variable patterns in a heat-transfer model."

—Myhrvold & Migoya, "Modernist Bread"

defy our ability to model it mathematically. Solid foods typically consist of an elaborate assemblage of different

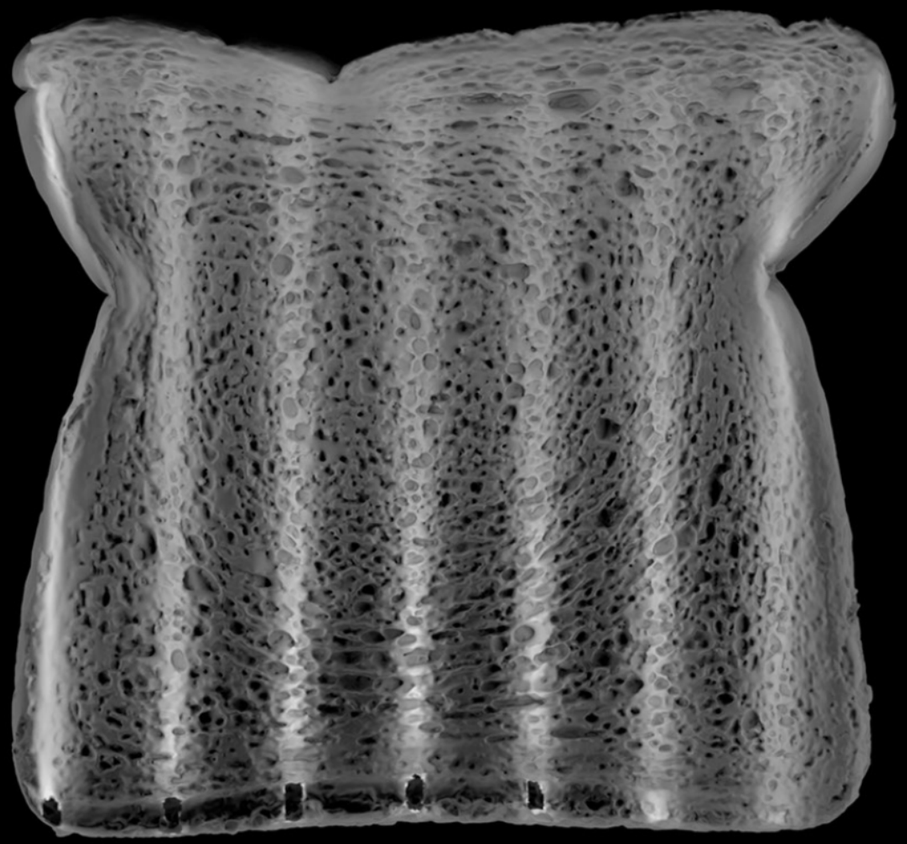
of oven temperature. Radiation skyrockets as wall temper-

Radiative Transfer

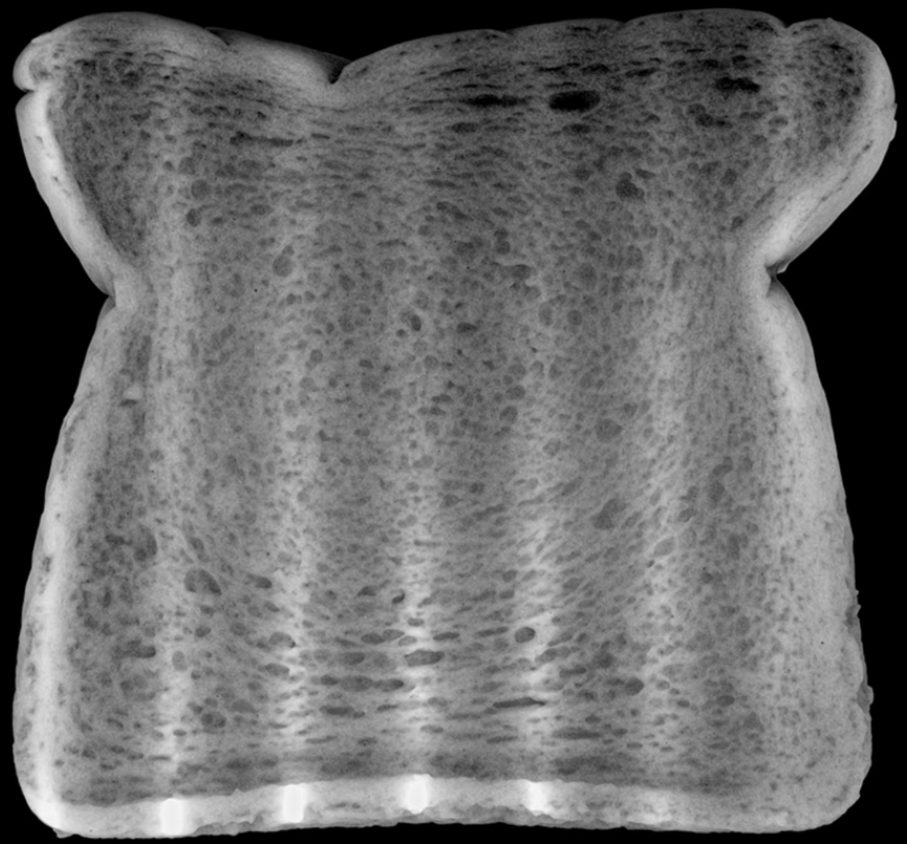


Radiative Transfer

heat transfer

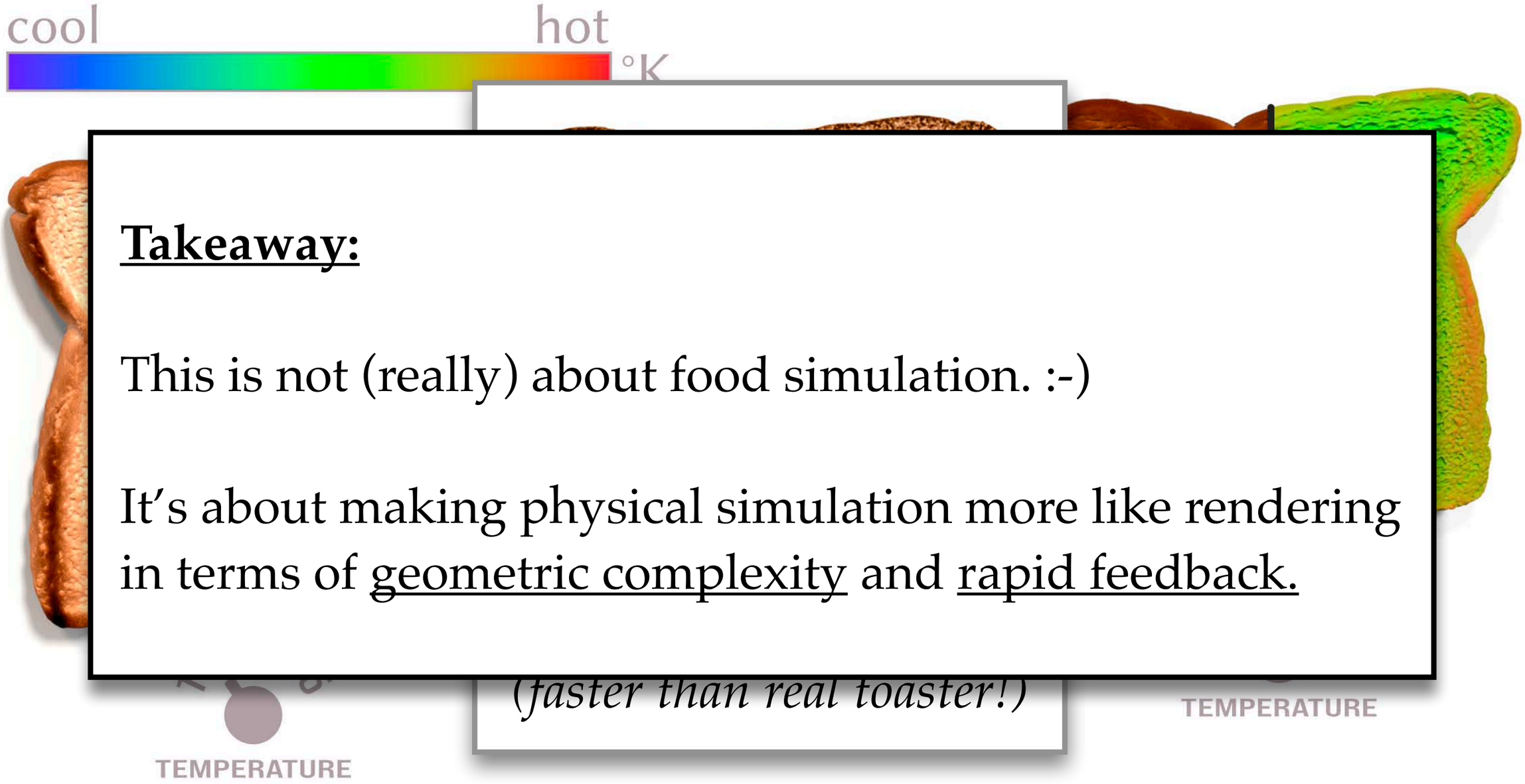


radiation
(ray tracing)



convection
(walk on stars)

Radiative Transfer



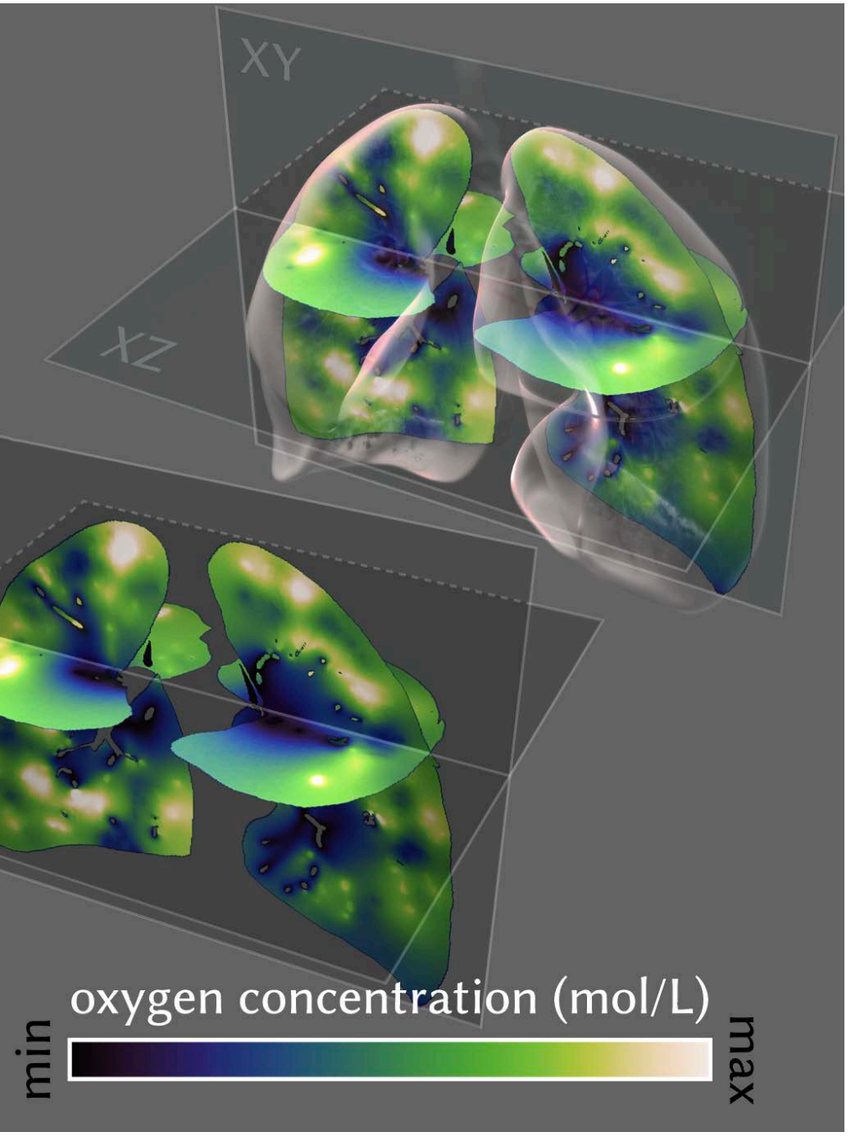
Takeaway:

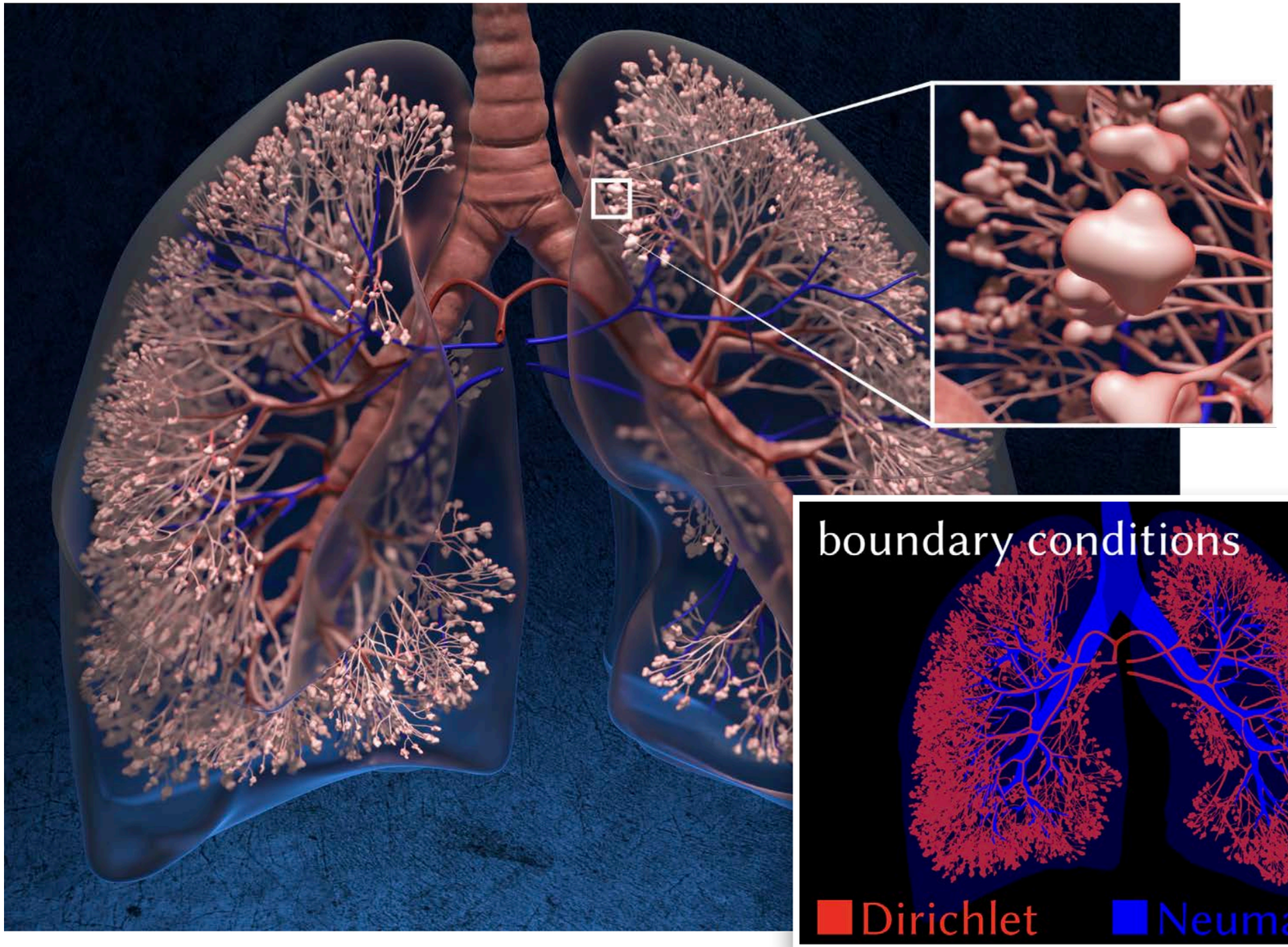
This is not (really) about food simulation. :-)

It's about making physical simulation more like rendering in terms of geometric complexity and rapid feedback.

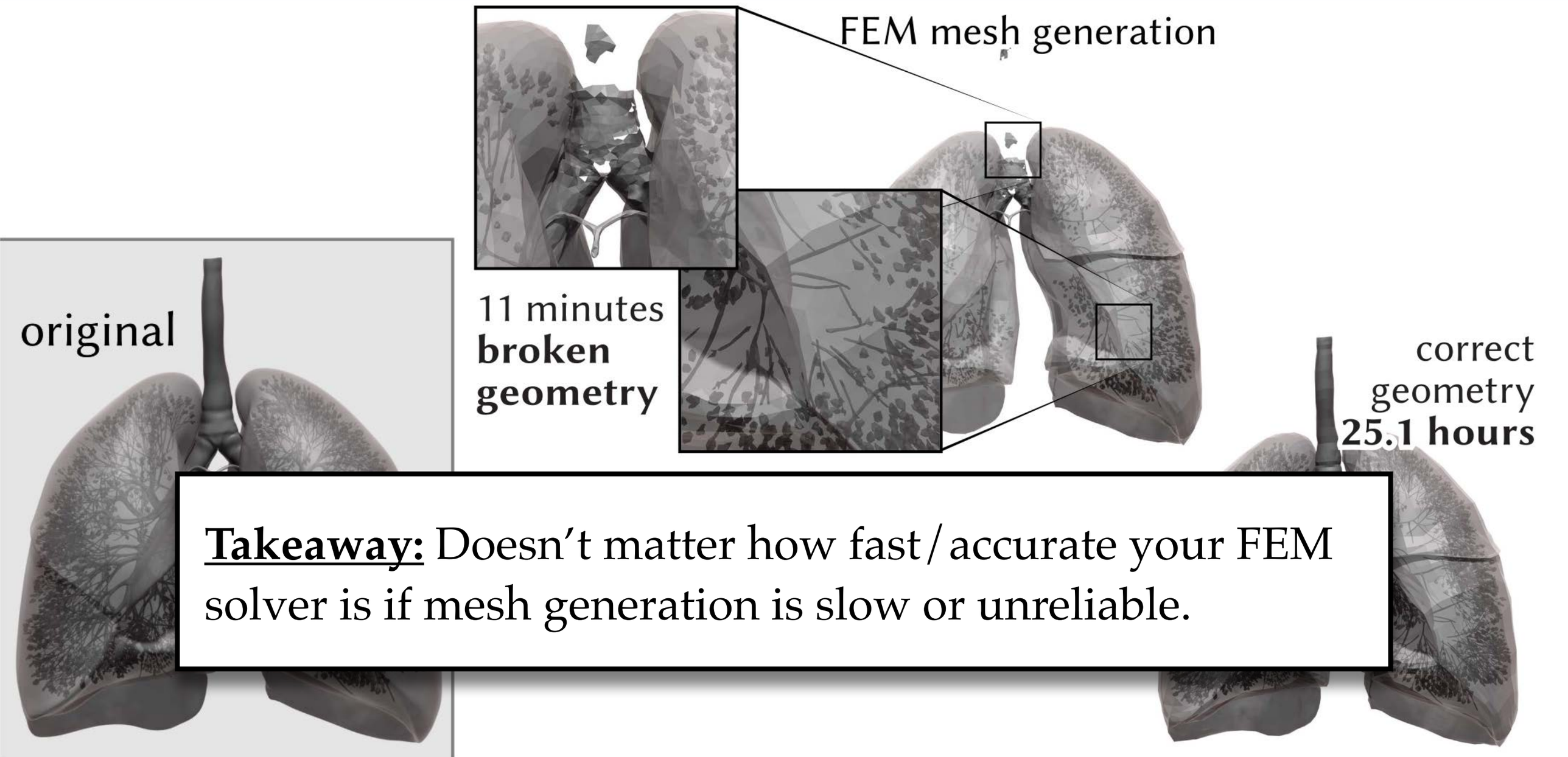
(faster than real toaster!)

Oxygen Diffusion



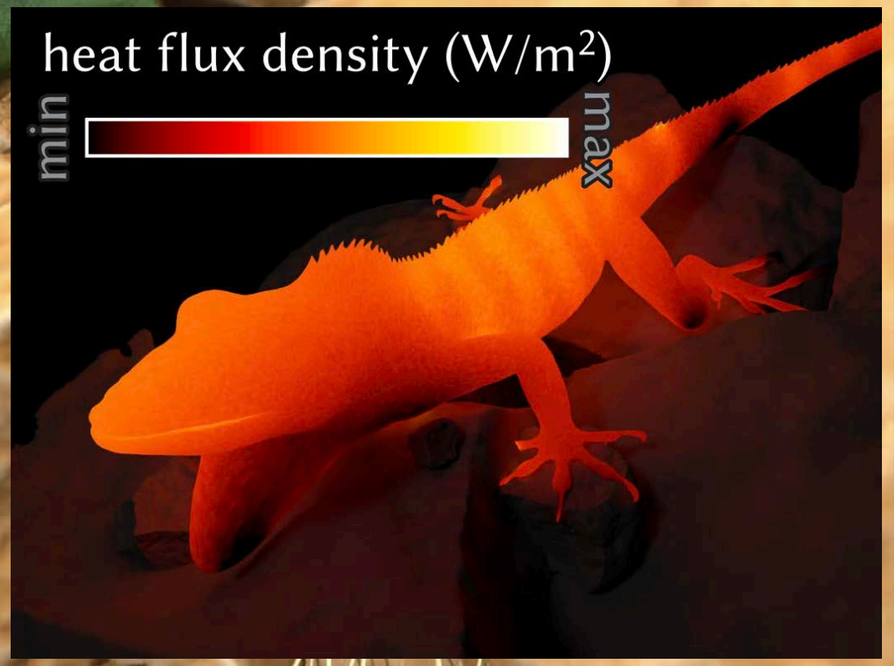


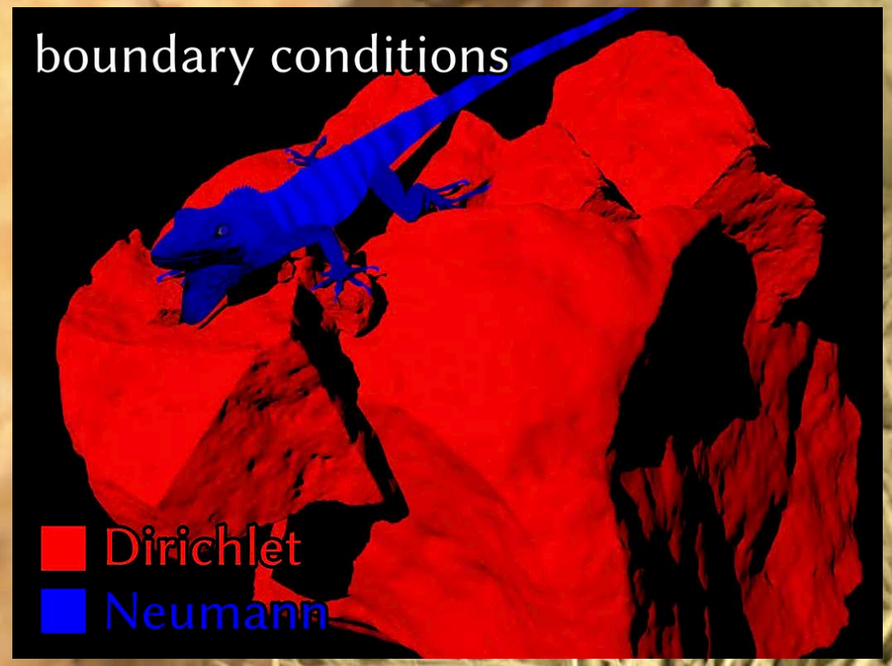
Oxygen Diffusion — FEM



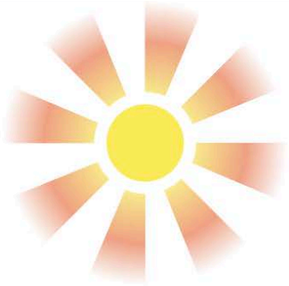
<u>Takeaway:</u> Doesn't matter how fast/accurate your FEM solver is if mesh generation is slow or unreliable.

Mixing Heat Transfer & Light Transport



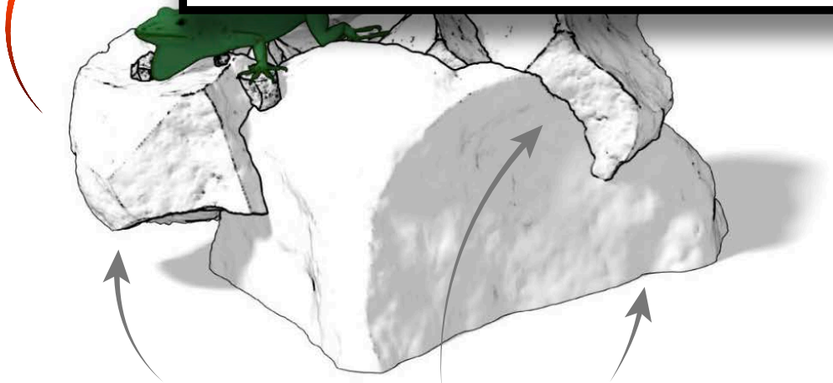


Mixing Heat Transfer & Light Transport

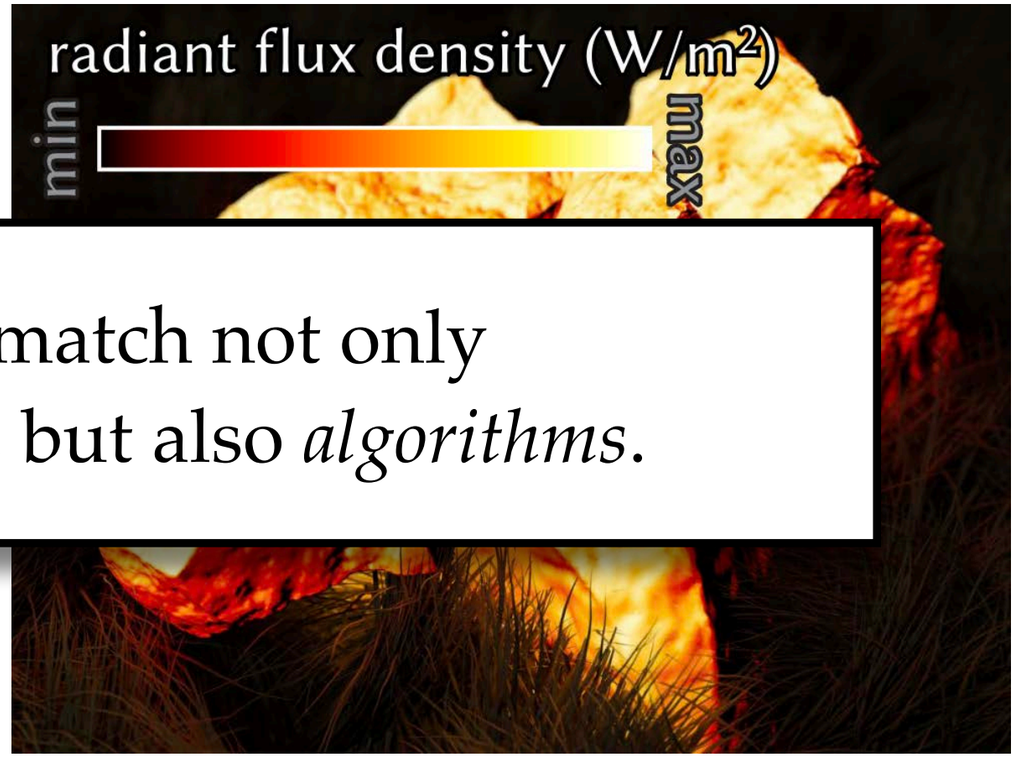


radiative heating
(ray tracing)

heat exch
(walk on s



signed distance functions

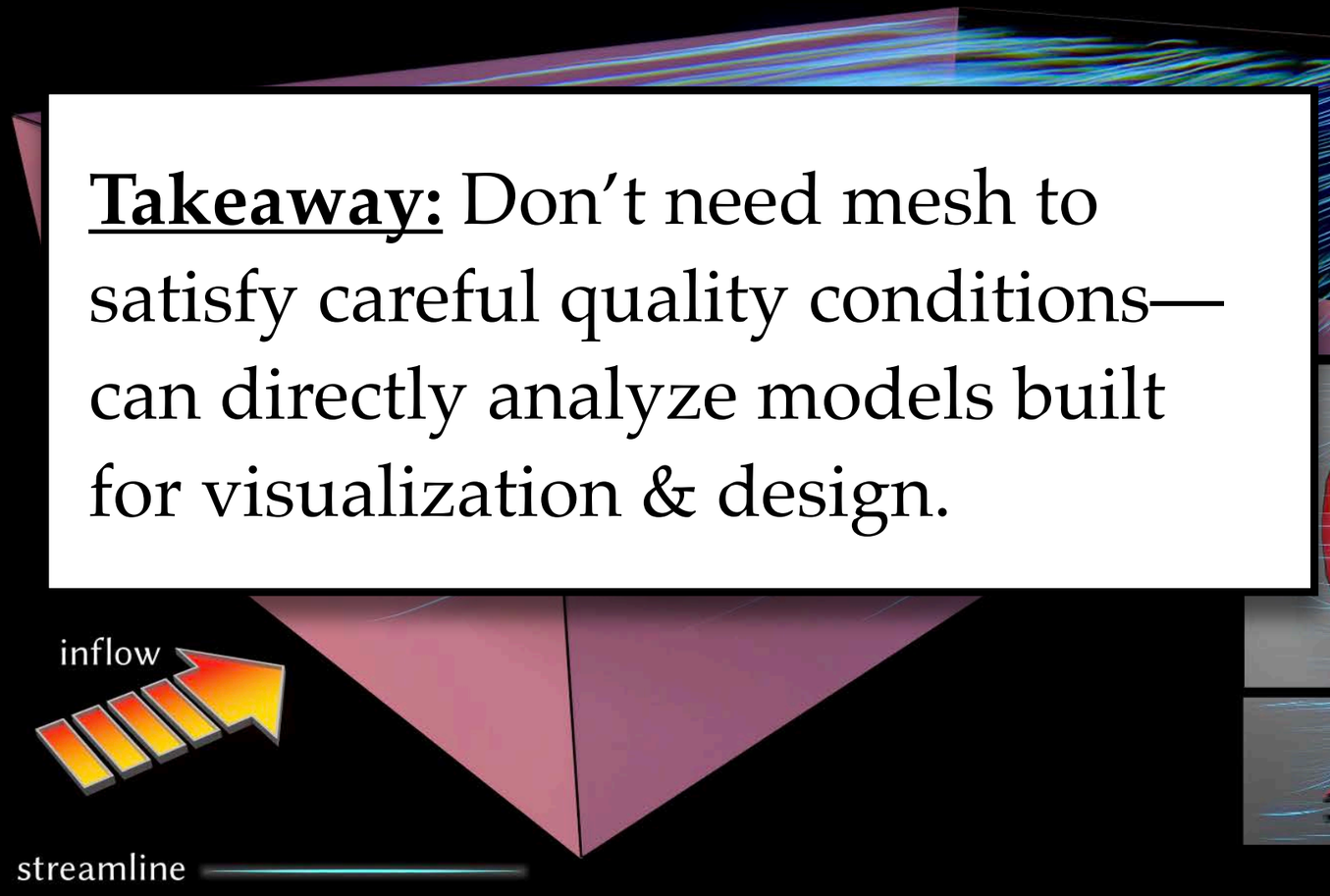


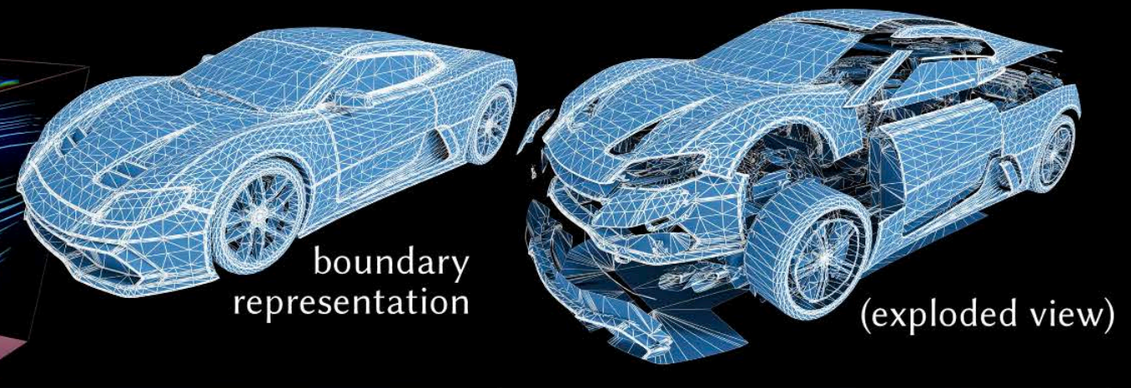
boundary conditions evaluated “on demand”

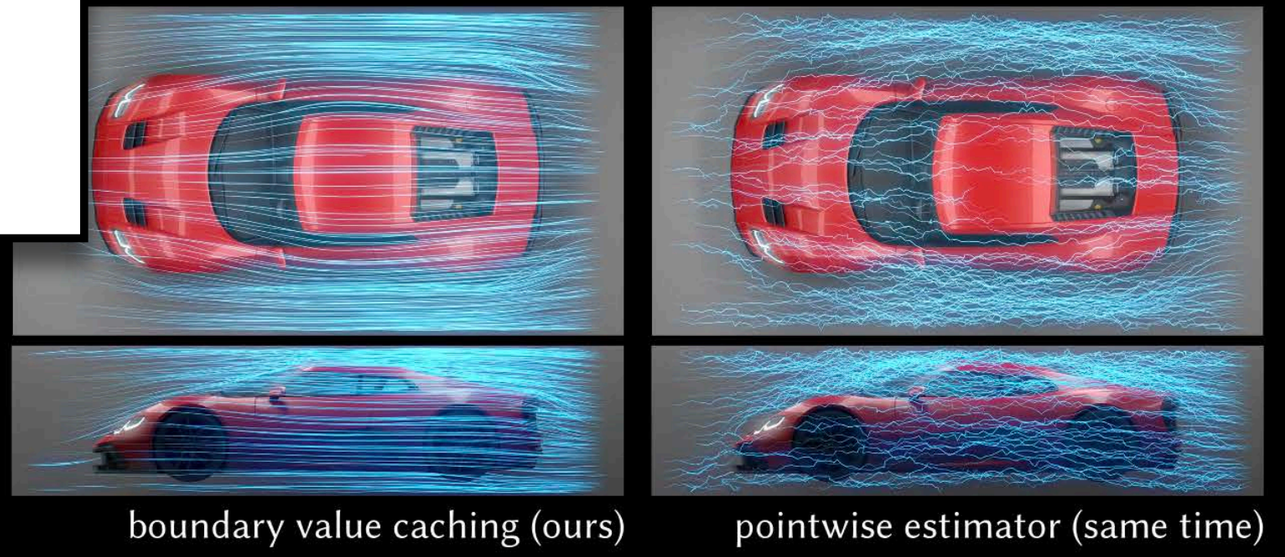
Takeaway: Easy to mix & match not only geometric representations, but also *algorithms*.

Fluid Simulation (Potential Flow)

Takeaway: Don't need mesh to satisfy careful quality conditions—can directly analyze models built for visualization & design.

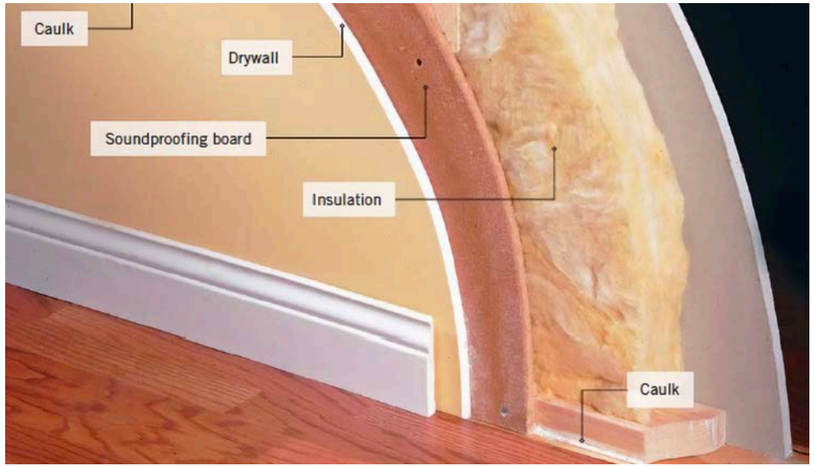




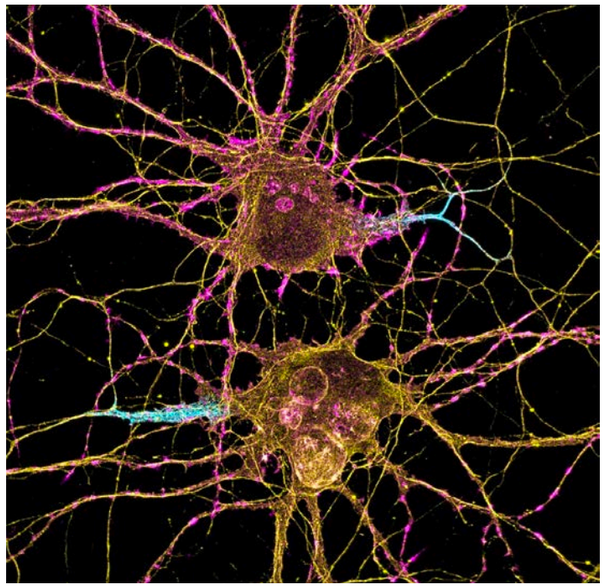


Spatially-Varying Coefficients

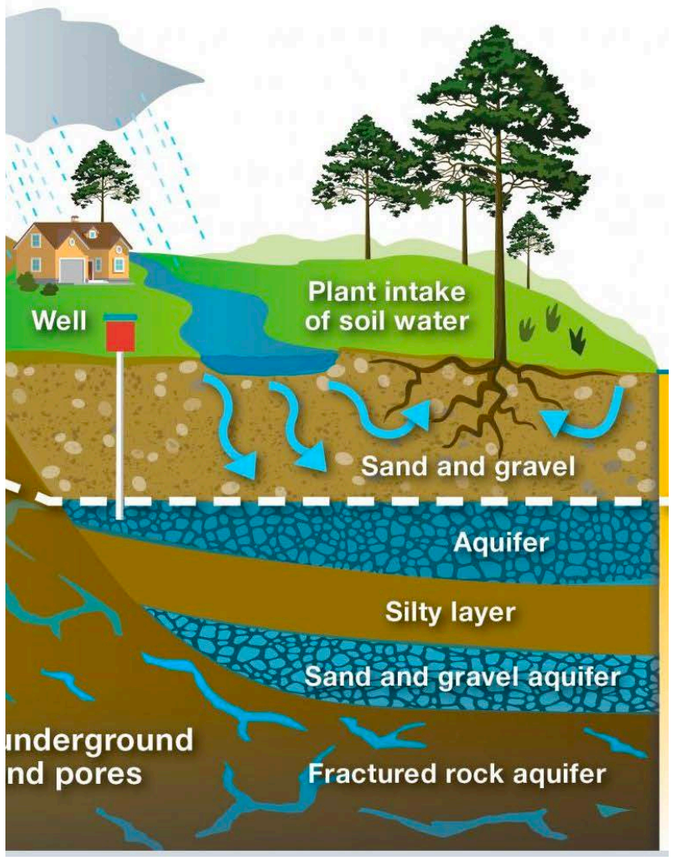
Real systems exhibit spatial variation



varying thermal diffusivity



varying electrical conductivity



varying permeability of porous media

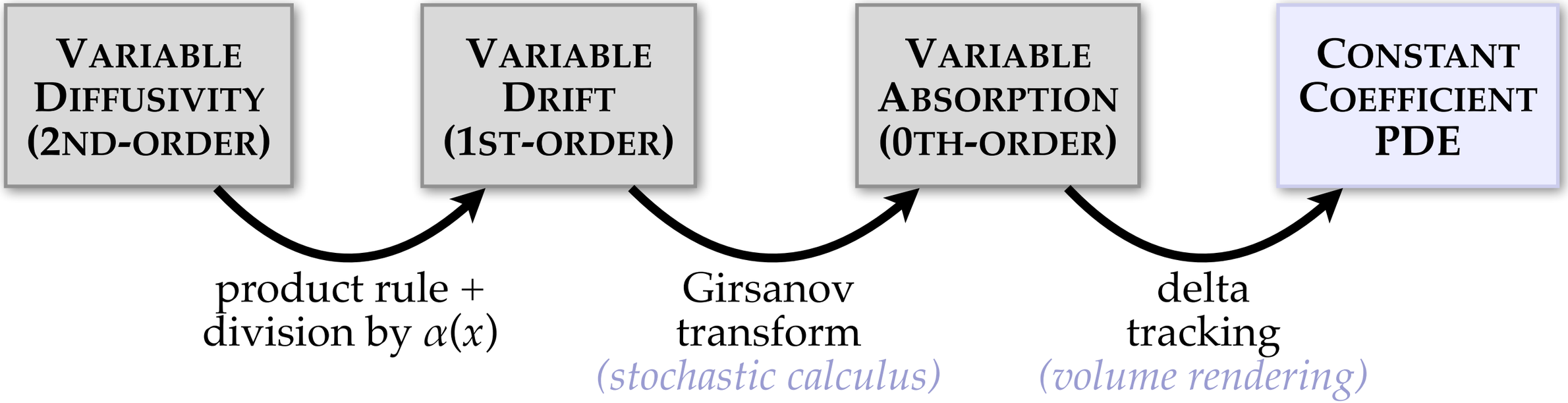
varying elastic response

Differential Equation—Variable Coefficients

In general, coefficient functions α, ω, σ can vary continuously over space:

$$\underbrace{\nabla \cdot (\alpha(x)\nabla u)}_{\text{DIFFUSION}} + \underbrace{\vec{\omega}(x) \cdot \nabla u}_{\text{DRIFT}} - \underbrace{\sigma(x)u}_{\text{ABSORPTION}} = f \quad \text{on} \quad \Omega$$
$$u = g \quad \text{on} \quad \partial\Omega$$

Strategy: transform into constant-coefficient PDE with all variability in source term:



Variable-Coefficient PDE—Estimator

constant coefficient PDE (screened Poisson)

$$\begin{aligned} \Delta U - \overline{\sigma} U &= f'(x, U) && \text{on } \Omega \\ U &= g'(x) && \text{on } \partial\Omega \end{aligned}$$

$$U := \sqrt{\alpha(x)}u \qquad \begin{aligned} f' &:= \frac{1}{\sqrt{\alpha(x)}} f \\ g' &:= \sqrt{\alpha(x)} g \end{aligned}$$

$$f'(x, U) := f'(x) + (\overline{\sigma} - \sigma'(x))U(x)$$

$$\sigma'(x) := \frac{\sigma(x)}{\alpha(x)} + \frac{1}{2}\left(\frac{\Delta\alpha(x)}{\alpha(x)} - \frac{|\nabla \ln(\alpha(x))|^2}{2}\right)$$

Variable-Coefficient PDE—Estimator

constant coefficient PDE (screened Poisson)

$$\begin{aligned} \Delta U - \overline{\sigma} U &= f'(x, U) && \text{on } \Omega \\ U &= g'(x) && \text{on } \partial\Omega \end{aligned}$$

Solution *U appears* on both sides of equation…?

FEM: didn't make life any easier!

Monte Carlo: recursive equations are no problem!

integral representation (recursive)

$$U(x) = \int_{\partial B(x)} U(y) P^{\overline{\sigma}}(x, y)\, dy + \int_{B(x)} f'(y, U) G^{\overline{\sigma}}(x, y)\, dy$$

always had to recursively estimate solution for boundary term

now we just estimate it for the source term as well

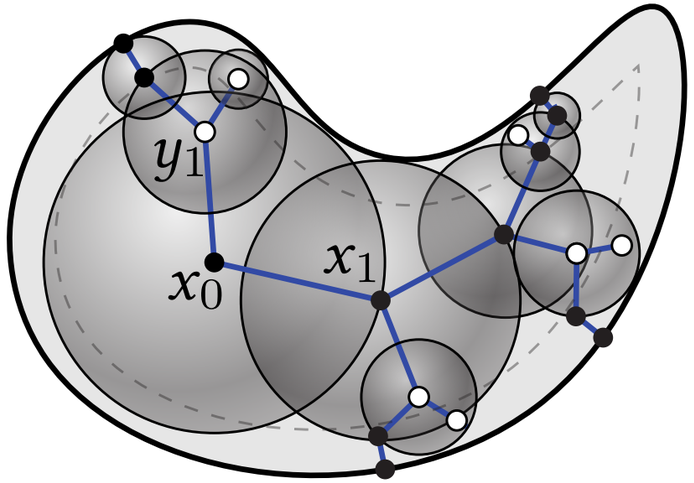
Monte Carlo estimator

$$\widehat{u}(x_k) = \underset{\text{term}}{\text{boundary}}(x_{k+1}) + \underset{\text{term}}{\text{source}}(y_k)$$

$$\widehat{u}(x_k) = \frac{1}{p} \underset{\text{term}}{\text{boundary}}(x_{k+1}) + \frac{1}{1-p} \underset{\text{term}}{\text{source}}(y_k)$$

evaluate with probability p

evaluate with probability $(1\text{-}p)$



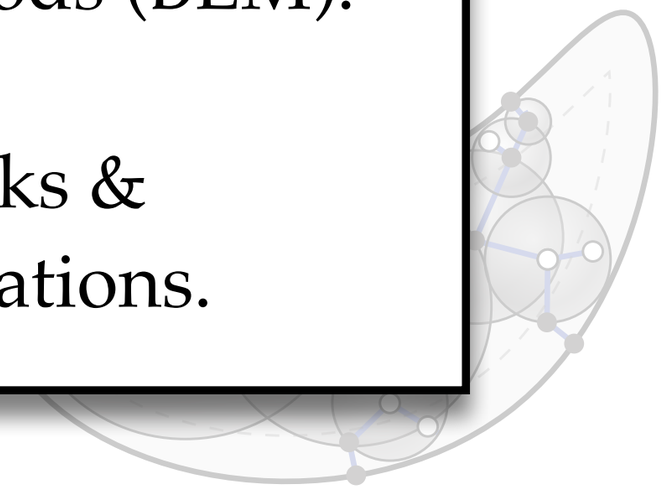
Problem: number of steps grows *exponentially!*

Variable-Coefficient PDE—Estimator

Takeaway:

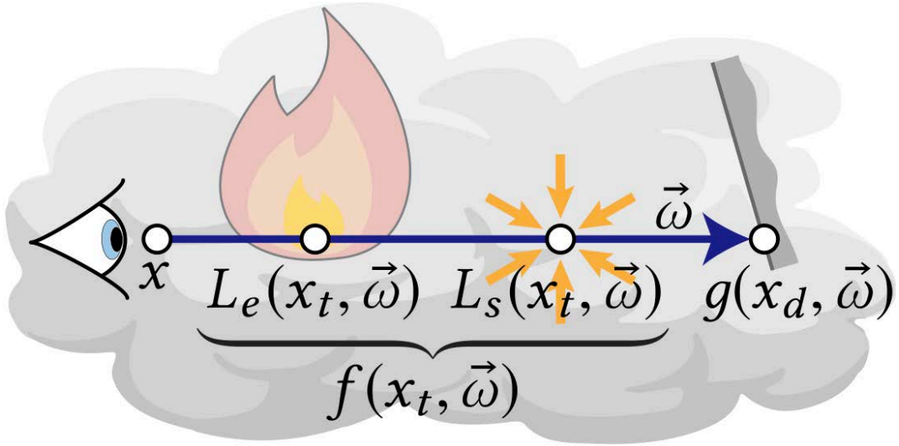
WoS is not just about integral formulations—which are already used by, *e.g.*, boundary element methods (BEM).

Rather, it enables fundamentally different tricks & transformations via **recursive** integral formulations.



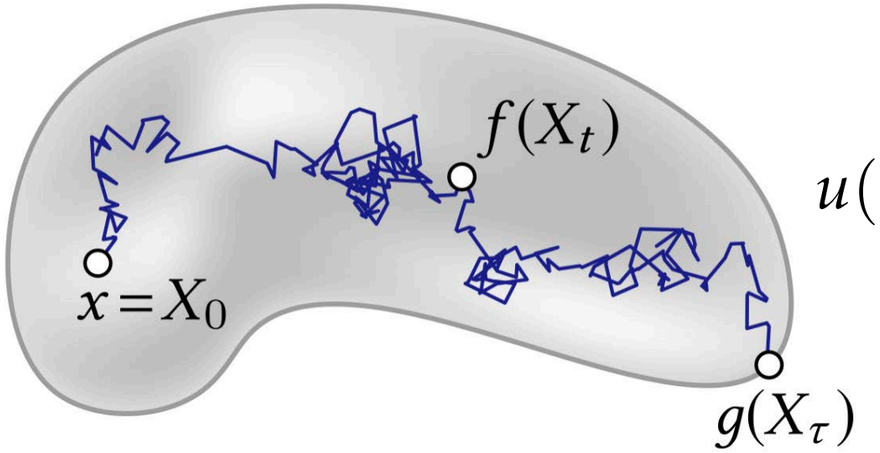
Variable Coefficient PDEs & Volume Rendering

Recursive, stochastic formulation also provides critical link to **rendering**.



$$L(x, \vec{\omega}) = \int_0^d e^{-\int_0^t \sigma(x_s) ds} f(x_t, \vec{\omega}, L) dt + e^{-\int_0^d \sigma(x_t) dt} g(x_d, \vec{\omega}, L)$$

VOLUME RENDERING EQUATION



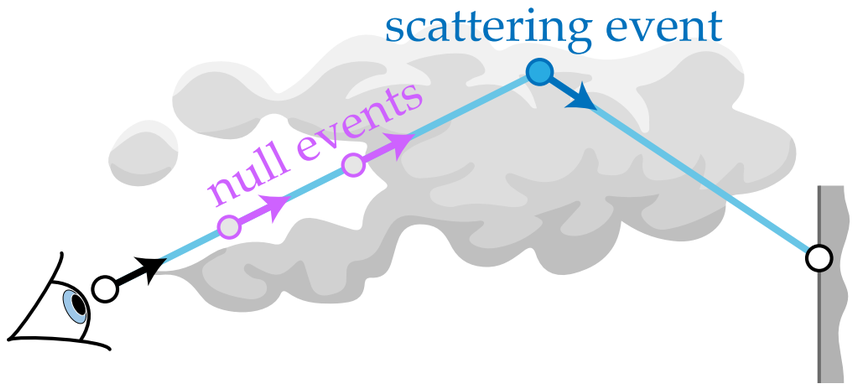
$$u(x) = \mathbb{E}\left[\int_0^T e^{-\int_0^t \sigma(X_s) ds} f(X_t) dt + e^{-\int_0^T \sigma(X_t) dt} g(X_t)\right]$$

FEYNMAN-KAC FORMULA

Can exploit this connection to reduce variance…

Acceleration — Volume Rendering

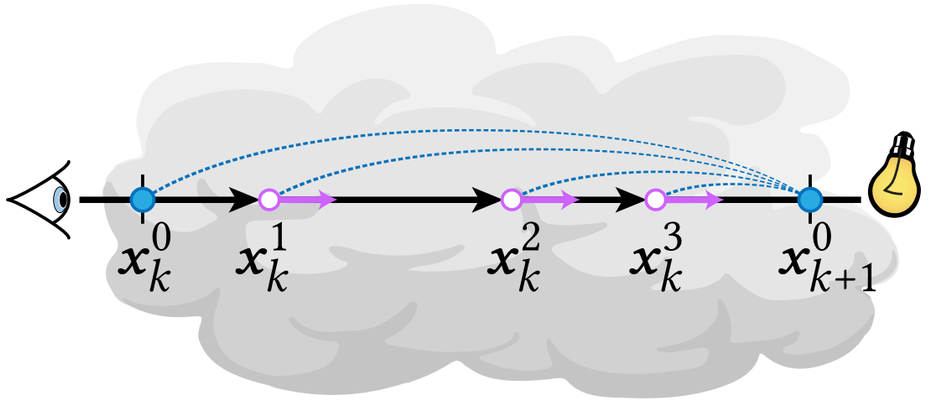
INPUT (4SPP)



Delta Tracking

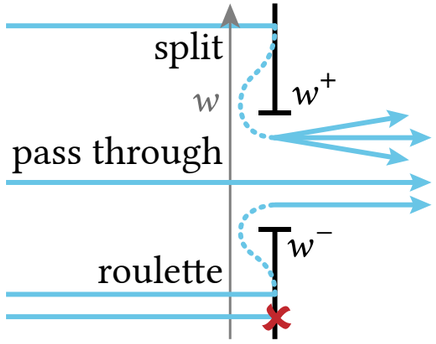
[Woodcock et al 1965; Raab et al 2008]

(4SPP)



Next Flight

[Cramer 1978]

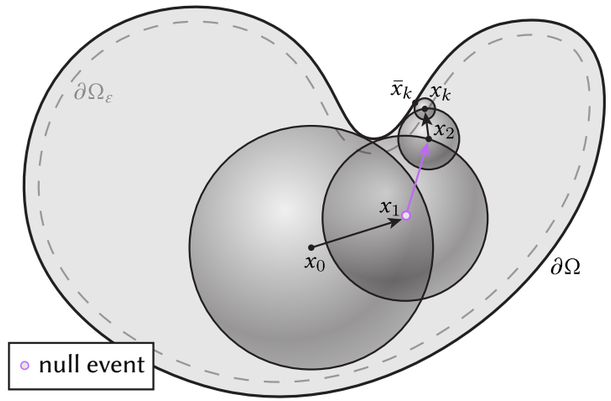


Weight Windows

[Hoogenboom and Légrády 2005]

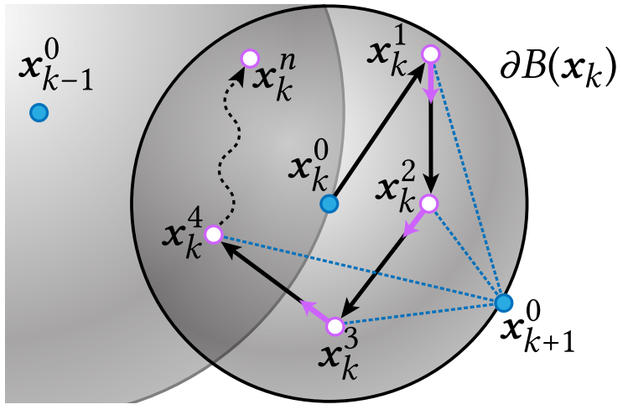
TARGET (4096SPP)

Acceleration — Spatially-Varying PDEs



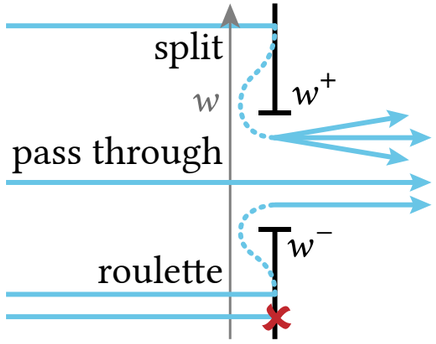
Delta Tracking

[Woodcock et al 1965; Raab et al 2008]



Next Flight

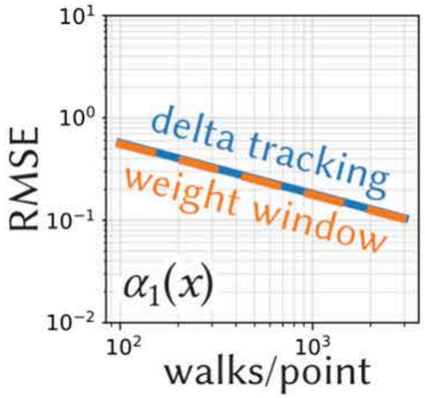
[Cramer 1978]



Weight Windows

[Hoogenboom and Légrády 2005]

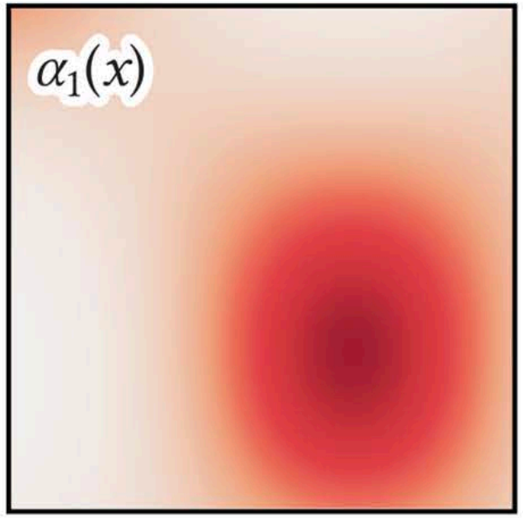
Reduction of Variance

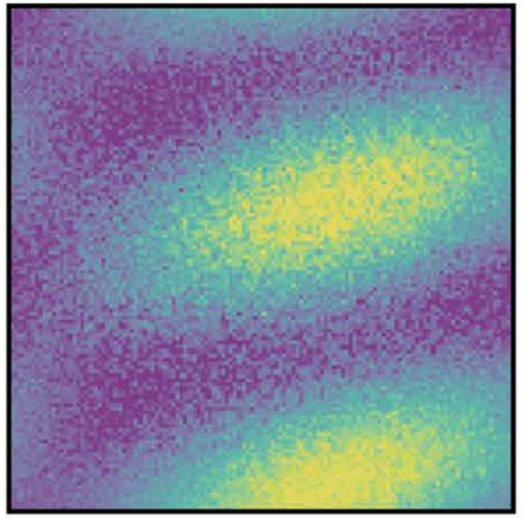


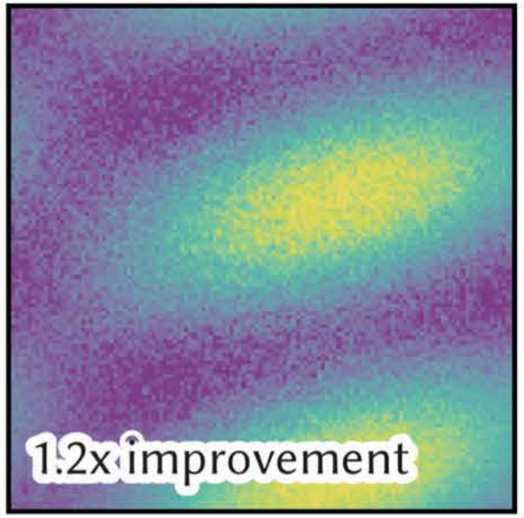
diffusion coefficient

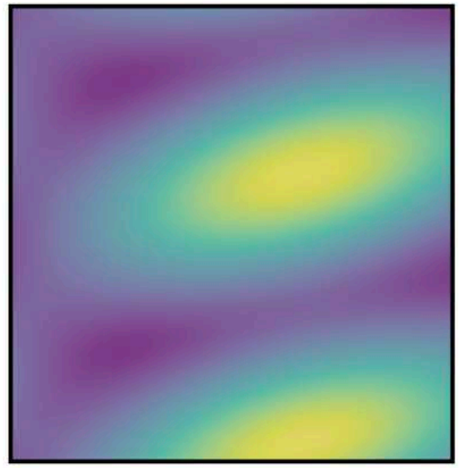
delta tracking (250 walks/point)

with weight window [0.5, 1.5]

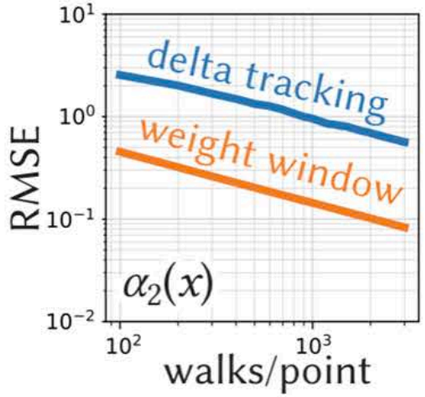


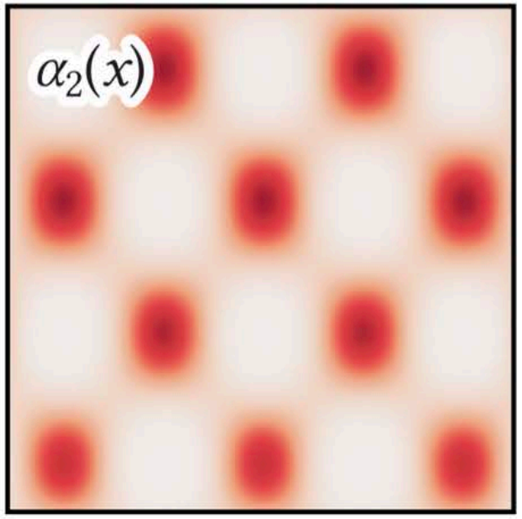


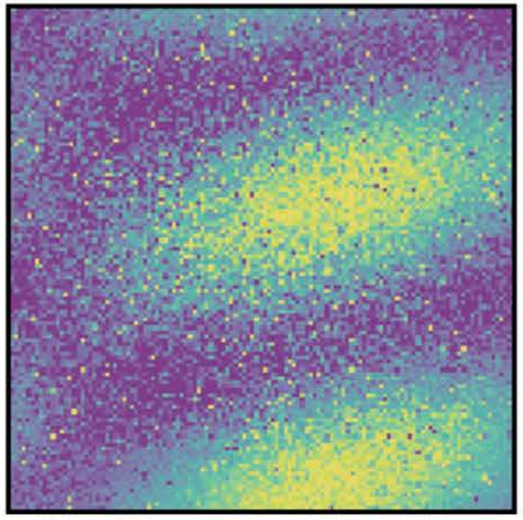


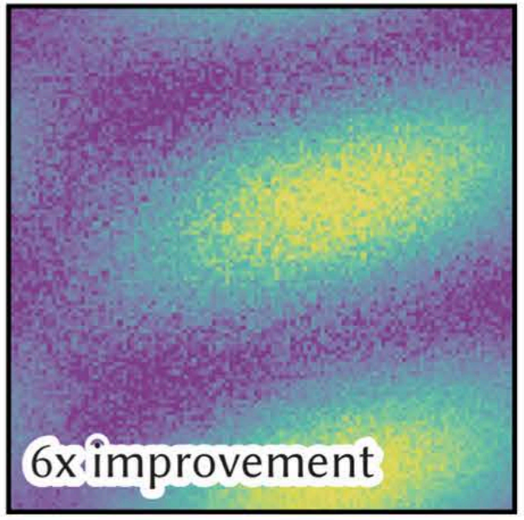


reference

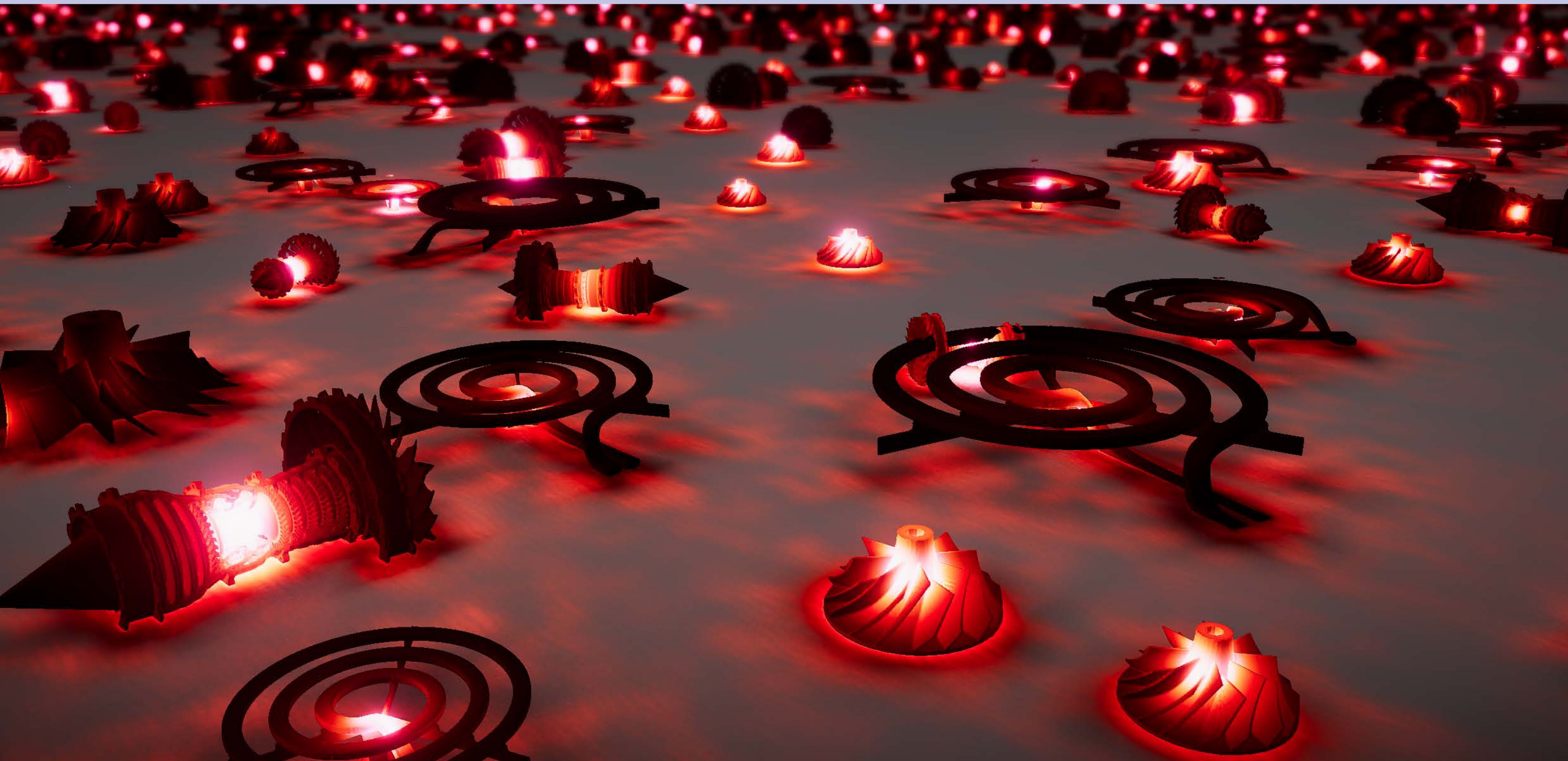






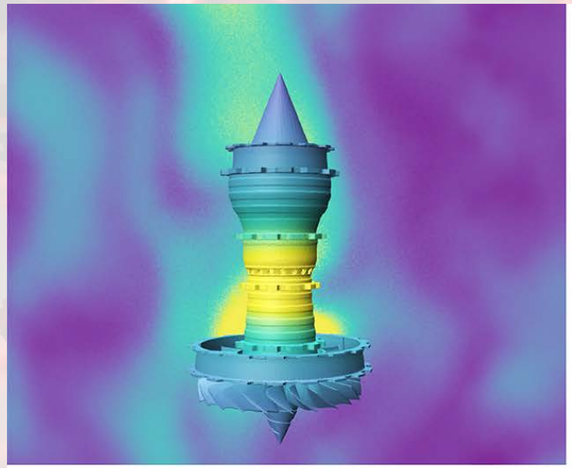


Example—Infinite Heterogeneous Environment

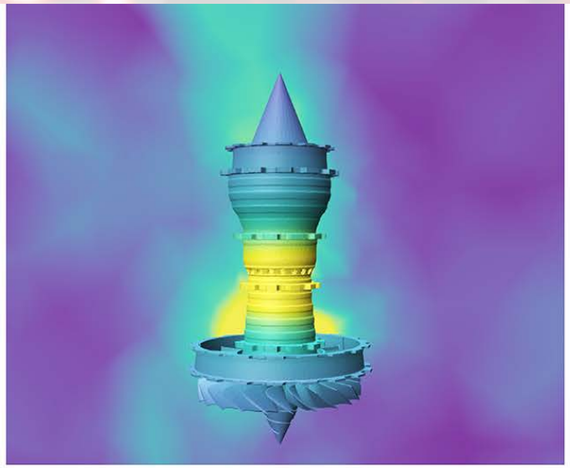


Comparison — Adaptive Mesh Refinement (AMR)

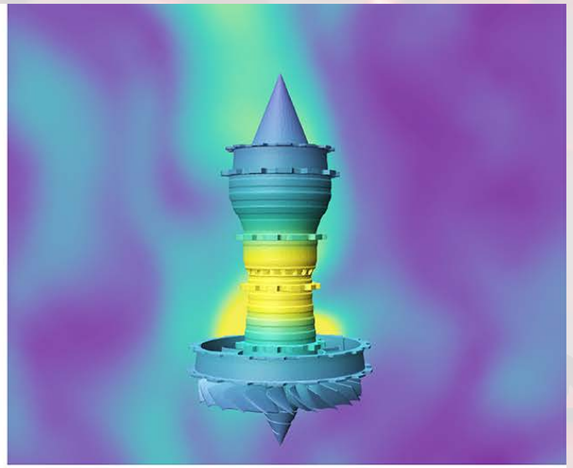
WoS — **10 minutes**



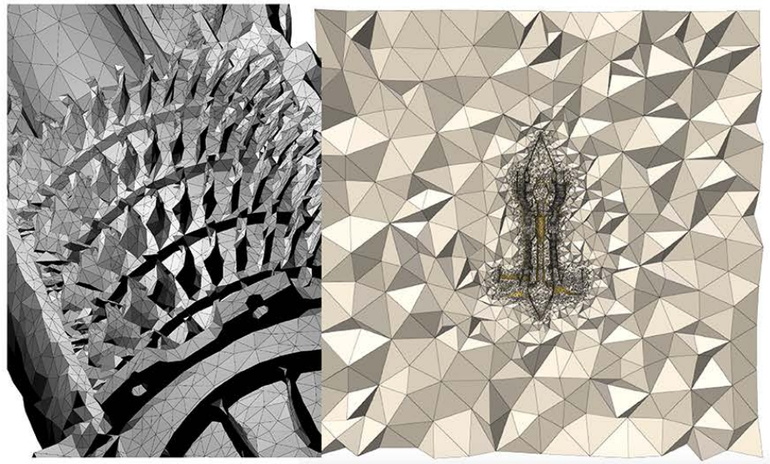
FEM — **1.5 hours**



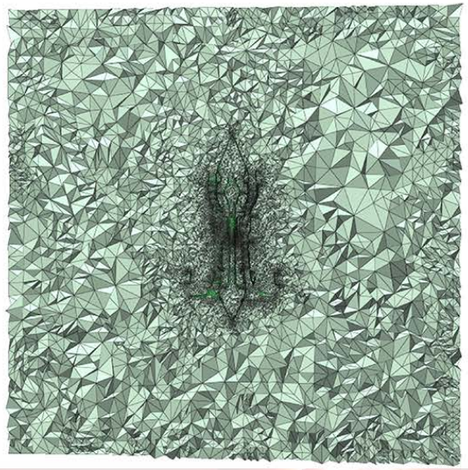
FEM+AMR — **2.5 hours**



input mesh
(used directly by WoS)



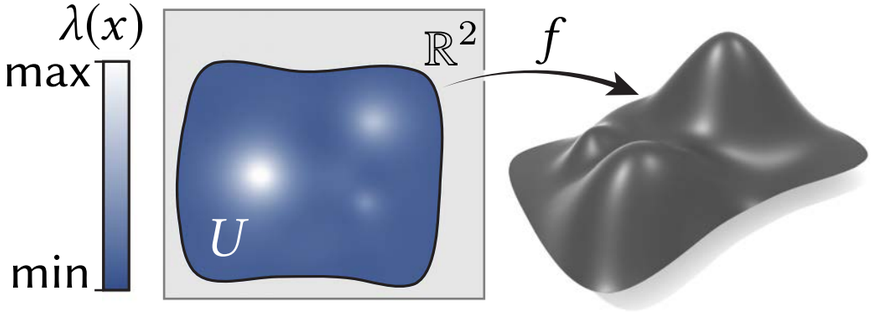
FEM mesh
(~700k tetrahedra)



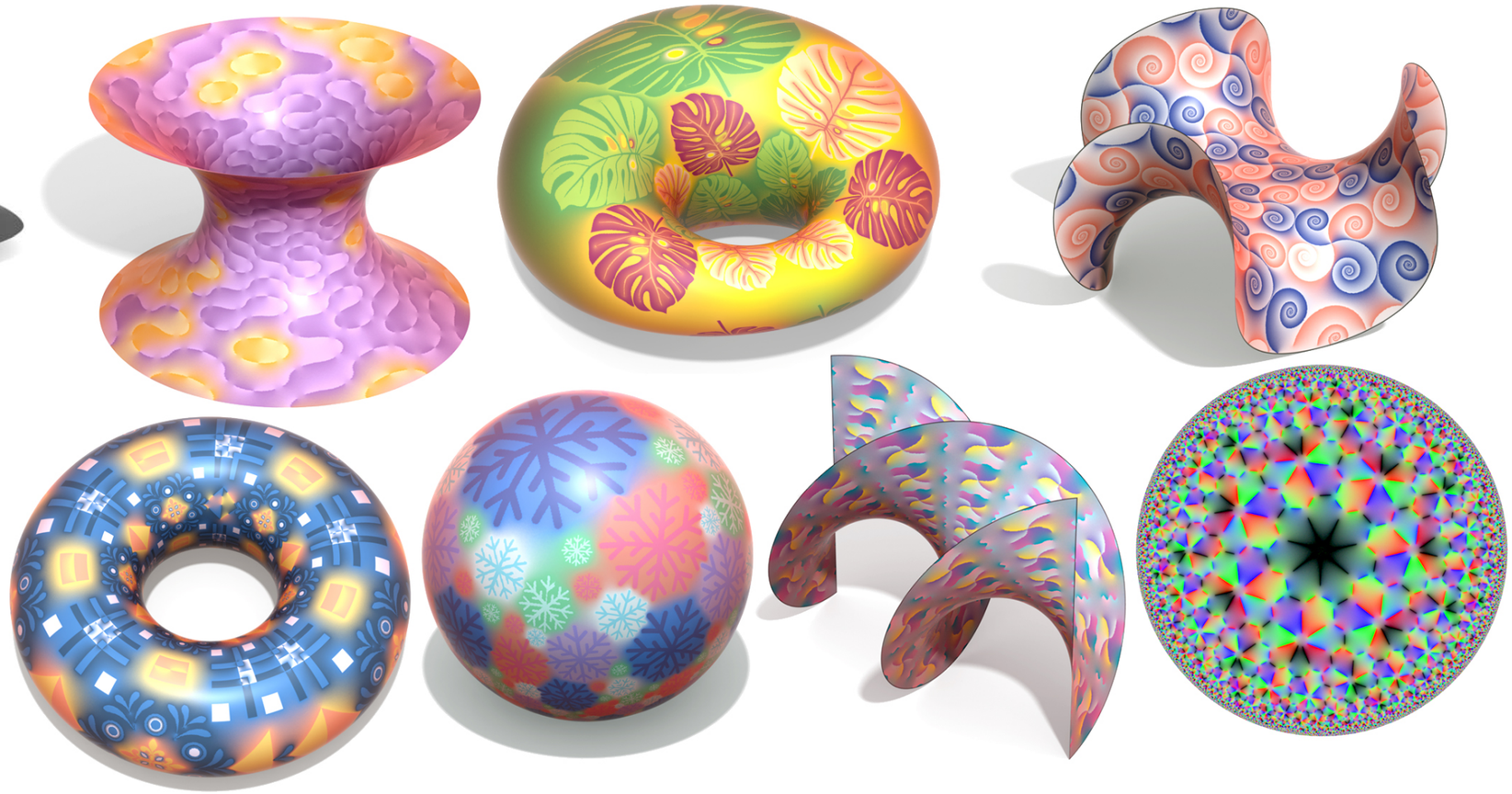
adaptive FEM mesh
(8.5 million tetrahedra)

Walk on Curved Surfaces

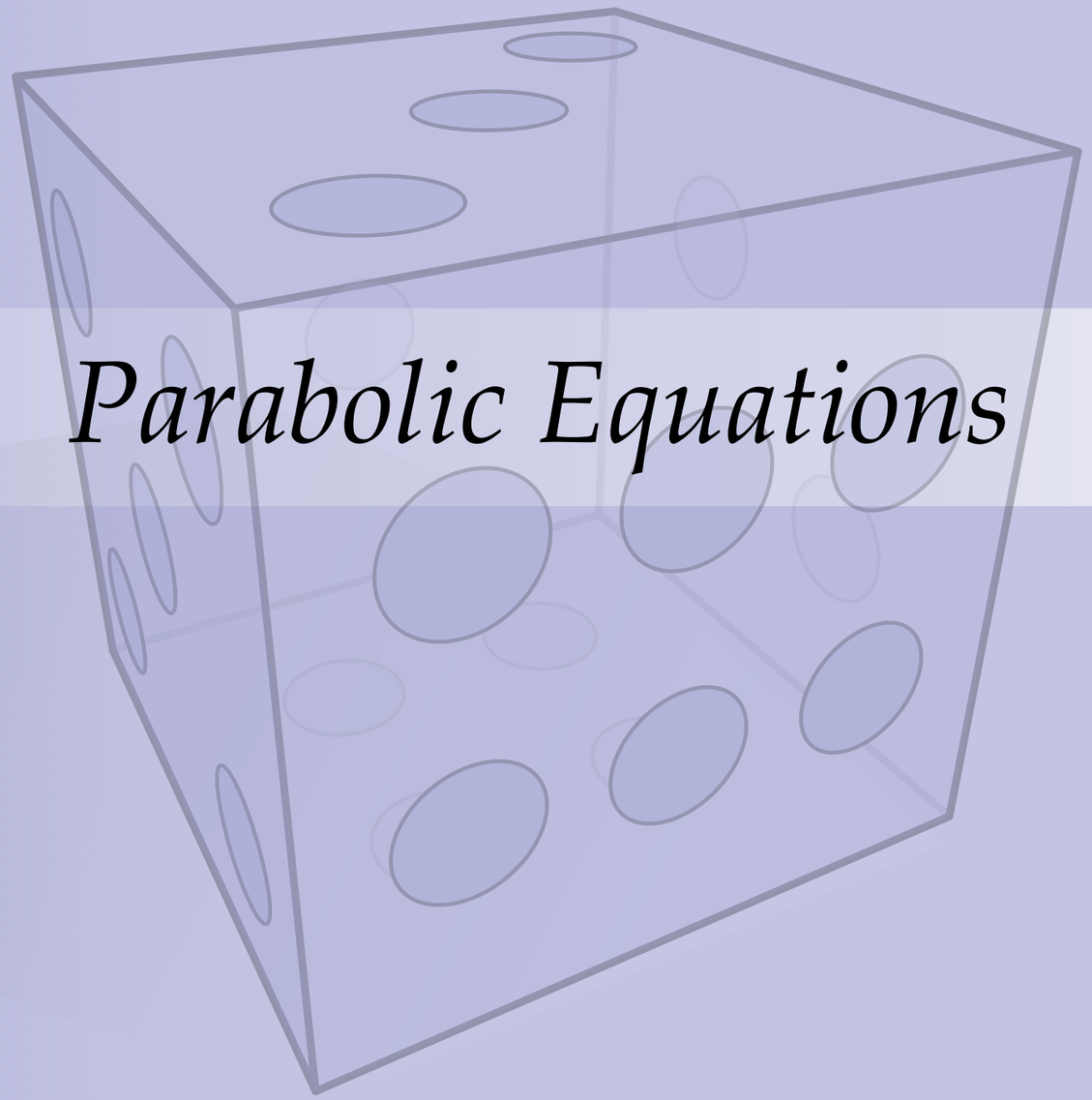
Key idea: express PDE on curved domain as variable-coefficient PDE on $\mathbb{R}^n$



Laplace-Beltrami $\Delta_f = \frac{1}{\lambda(x)}\Delta$ *Laplacian on* $\mathbb{R}^2$



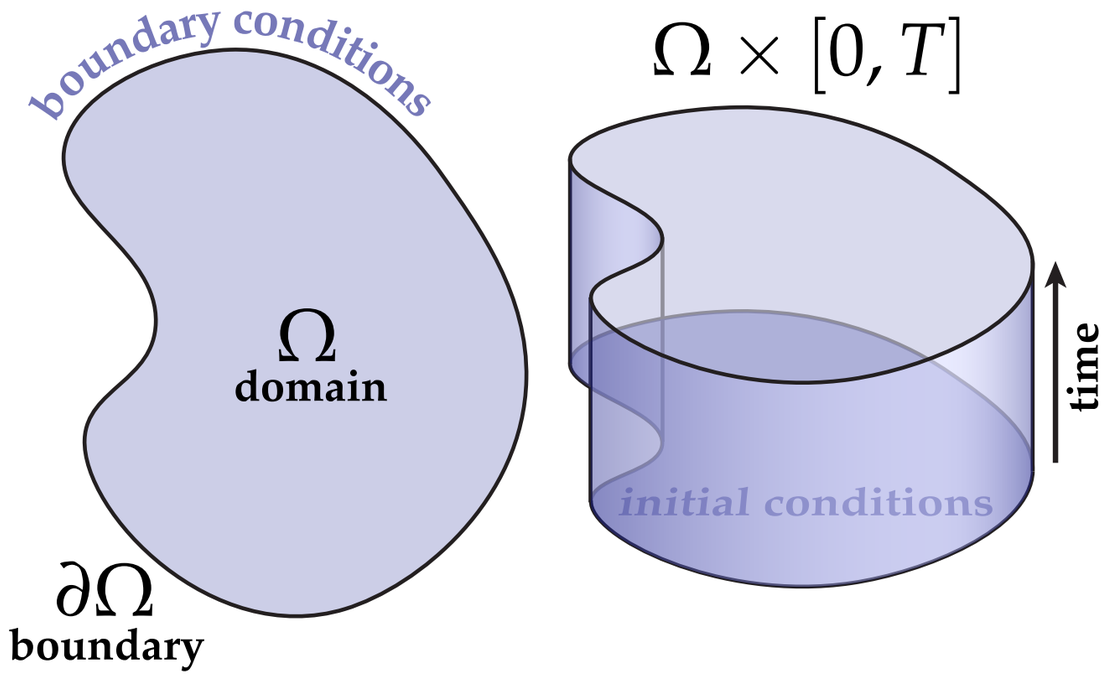
Future work: general *anisotropic* parameterizations (NURBS, subdivision surfaces, …)



Parabolic Equations

Parabolic Equations

- So far considered only *stationary solutions* / *time-independent equations*
- How can we simulate *parabolic* equations evolving over space & time?
- Recall we now have:
  - **boundary conditions** $g : \partial\Omega \to \mathbb{R}$
  - **initial conditions** $u_0 : \Omega \to \mathbb{R}$
- **ODE approach:** *"time stepping"*
  - incrementally update in time

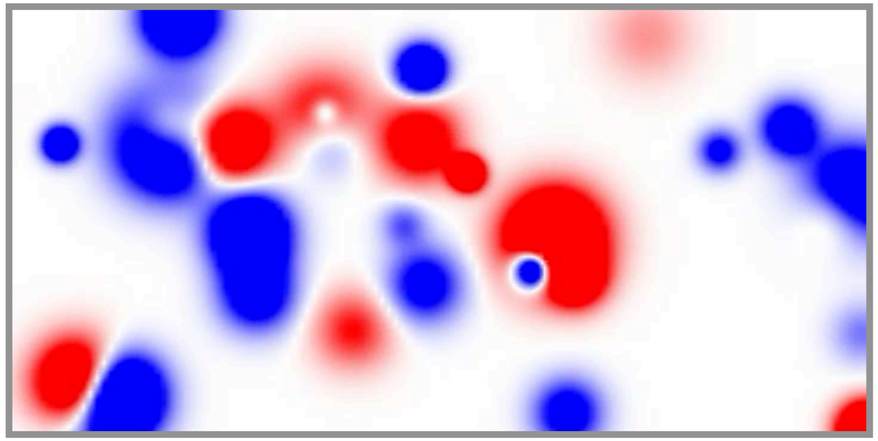


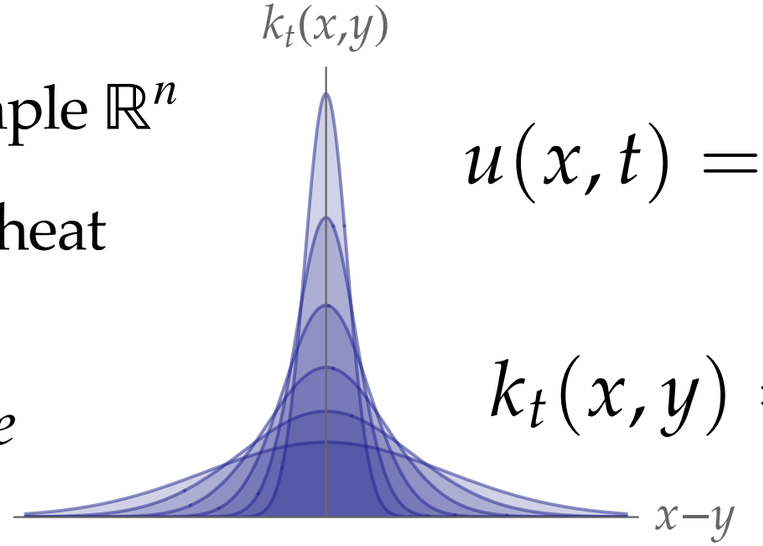
- **WoS approach:** solve for any $t > 0$ directly by *"walking on spacetime"*

Heat Equation—No Boundary (Easy)

- Consider heat equation on $\Omega = \mathbb{R}^n$ *(no boundary)*
- Solution to heat equation is then just <u>convolution</u> with heat kernel $k_t(x,y)$
- **Q:** What's our stochastic algorithm here?
- **A:** Just apply Monte Carlo integration! (No random walks needed.)
  - not meaningful to *uniformly* sample $\mathbb{R}^n$
  - instead, importance sample the heat kernel k_t or initial conditions u_0
  - …or both, via *multiple importance sampling (MIS)*

$$\frac{d}{dt}u(x,t) = \Delta u(x,t) \qquad u_0 : \mathbb{R}^n \to \mathbb{R}$$



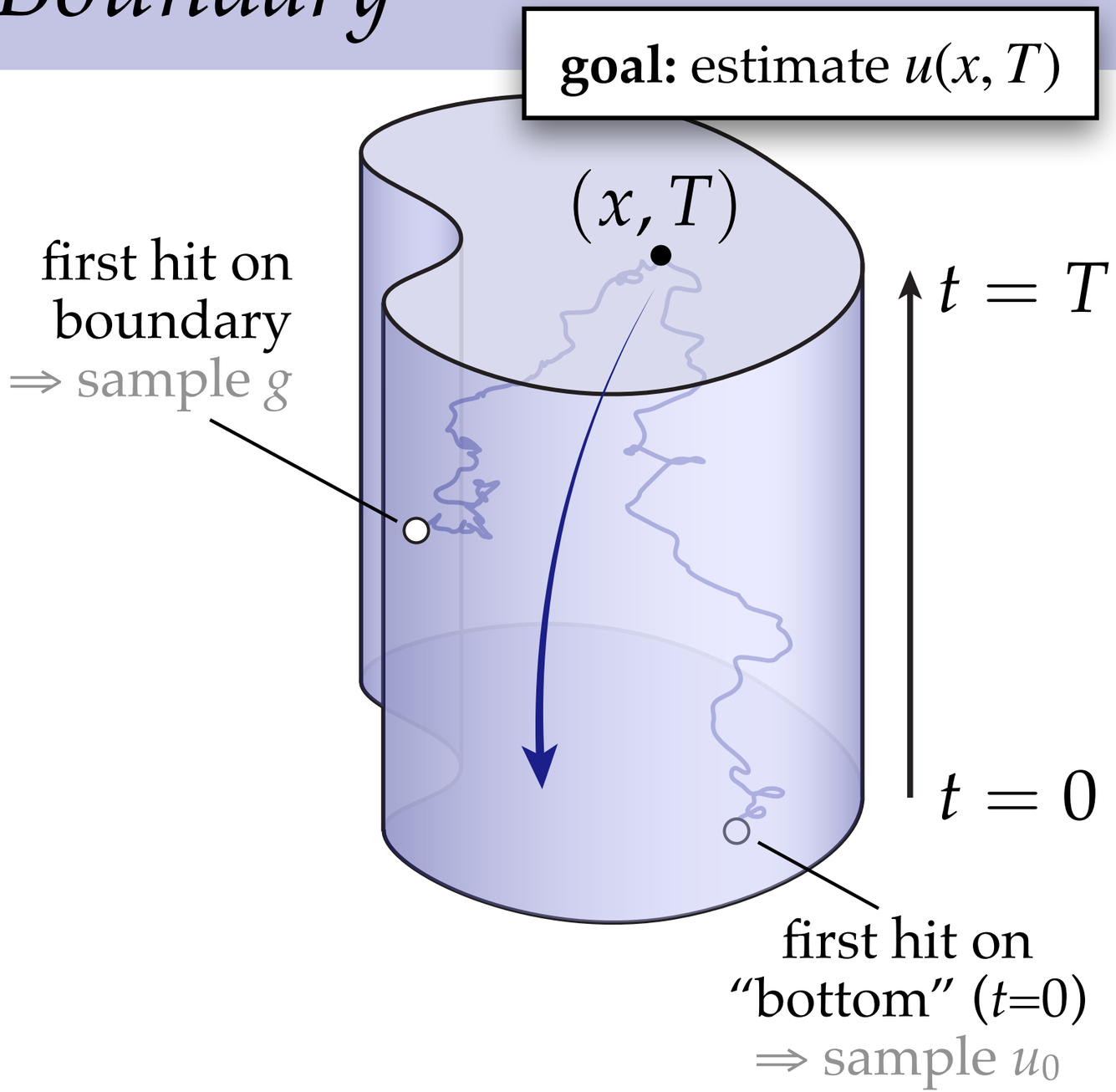


$$u(x,t) = \int_{\mathbb{R}^n} k_t(x,y)u_0(y)\,dy$$

$$k_t(x,y) = \frac{1}{(4\pi t)^{d/2}} e^{-|x-y|^2/(4t)}$$

Heat Equation—With Boundary

- For a domain with boundary, must instead simulate a random walk X_t
  - still Brownian motion $dX_t = dW_t$
- **Space:** take steps "as usual" (*e.g.*, sample from sphere *a la* WoS)
- **Time:** also keep track of time, *starting* at T and walking "backwards"
- If we hit boundary $\partial\Omega$ first, sample boundary conditions $g(X_t, t)$
- If we hit time t=0 first, sample initial conditions $u_0(X_0,0)$



Walk on Moving Spheres (simplified)—Heat Equation

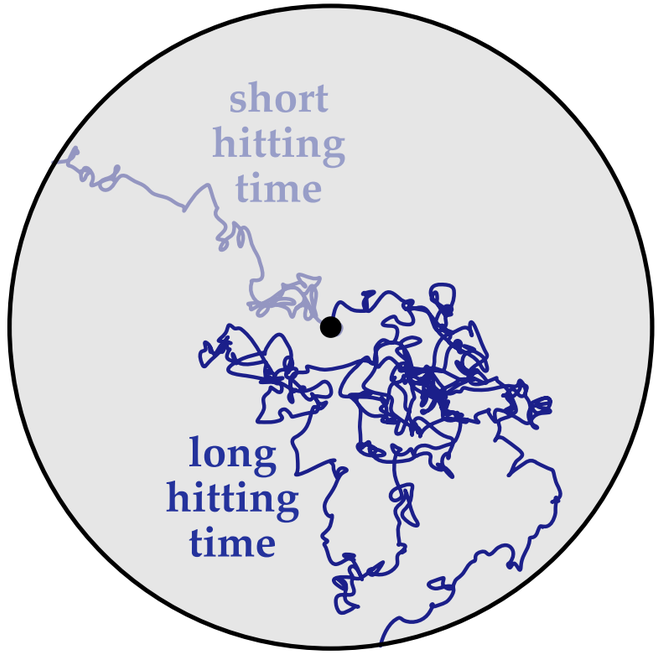
- $\hat{u} \leftarrow 0$ — *initialize estimate of $u(x_0, T)$*
- **for** $i = 1, \ldots, N$:
  - $(x, t) \leftarrow (x_0, T)$ — *start at evaluation point/time*
  - **while** $x \notin \Omega_\varepsilon$: — *still in the domain*
    - $r \leftarrow \min_{y \in \partial\Omega} |x - y|$ — *largest ball radius*
    - $\tau \leftarrow \text{SampleExitTime}(r)$ — *exit time*
    - **if** $t - \tau \leq 0$: — *time's up!*
      - **return** $u_0(x)$ — *evaluate initial conditions*
    - $x \sim U_{\partial B_r(x)}$ — *sample sphere around x*
    - $t \leftarrow t - \tau$ — *decrement clock*
  - $\hat{u} \leftarrow \hat{u} + g(\bar{x})$ — *evaluate boundary conditions*
- **return** $\hat{u}/N$ — *return the average*

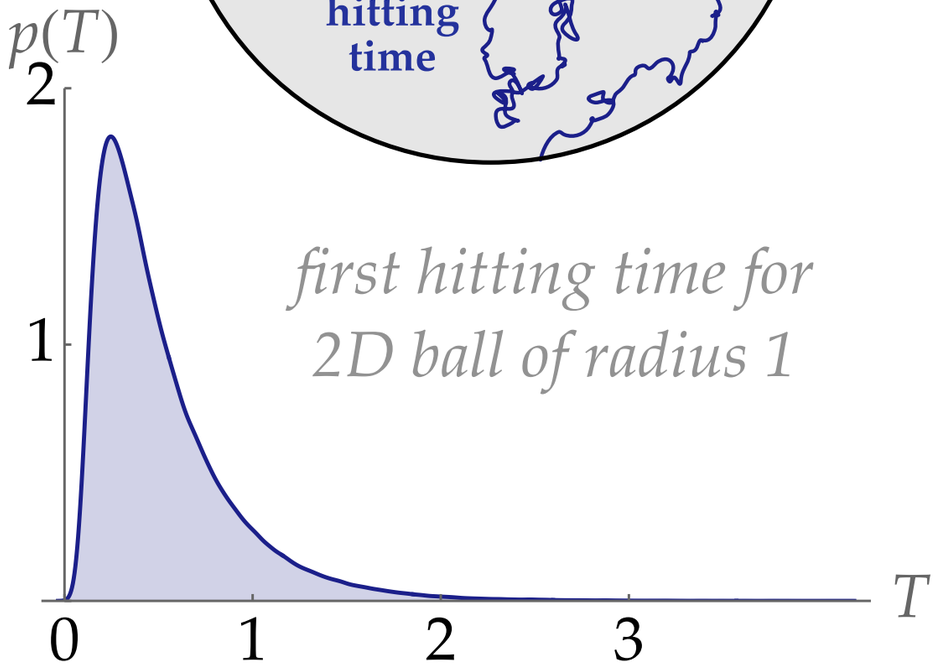
simulate Brownian motion using moving walk on spheres

overall estimate is average of boundary values or initial conditions

Exit Time Distribution

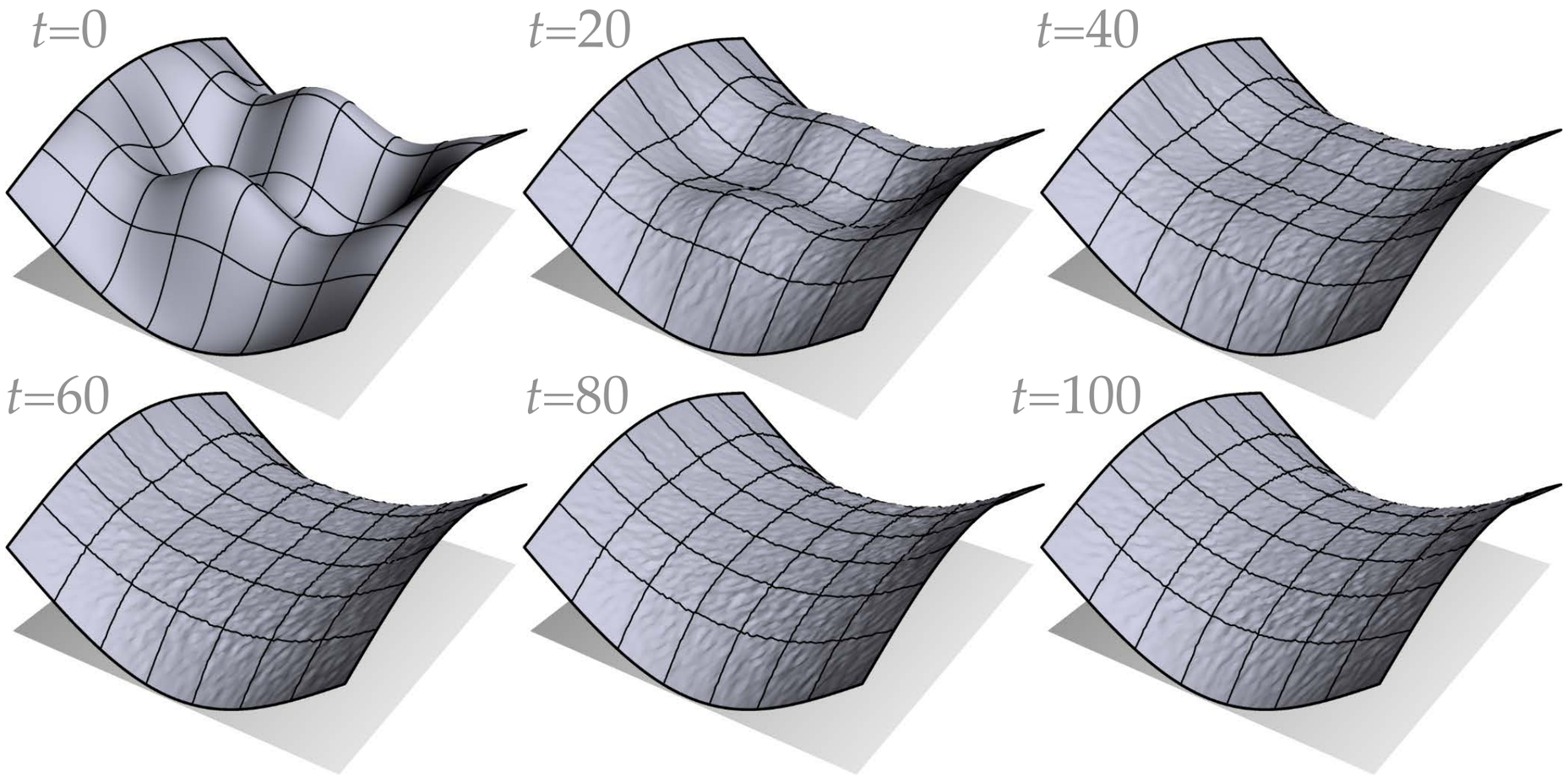
- **Bessel process** is the magnitude $R_t := |W_t|$ of a Brownian process W_t starting at $x = 0$
- In n dimensions, obeys SDE $dR_t = \frac{n-1}{2R_t}dt + dW_t$
  - **Q:** How could you show this?
  - **A:** Ito's lemma!
- **Q:** How can we simulate time taken by WoS step to hit a ball of radius r_k?
- **A:** Not nearly as easy as sampling closed-form Green's function/Poisson kernel
  - for an explicit procedure, see Deaconu & Herrmann, *"Walk on Moving Spheres"* (2013)

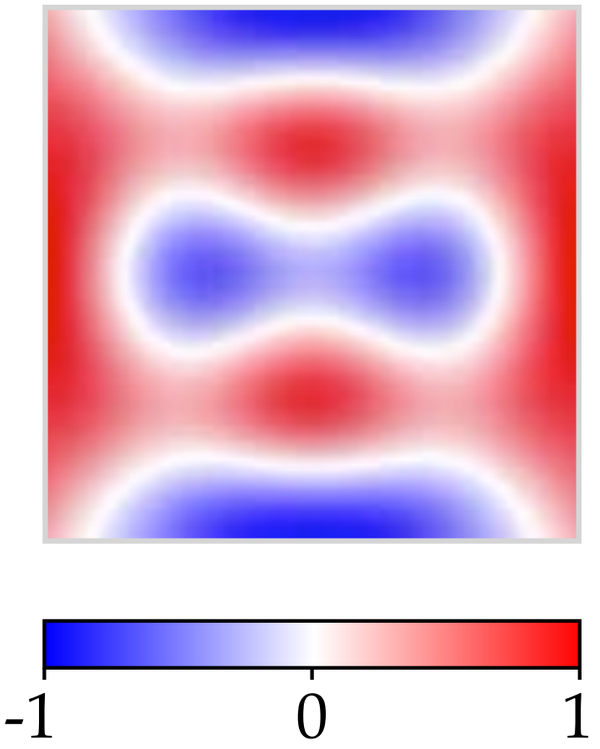




first hitting time for 2D ball of radius 1

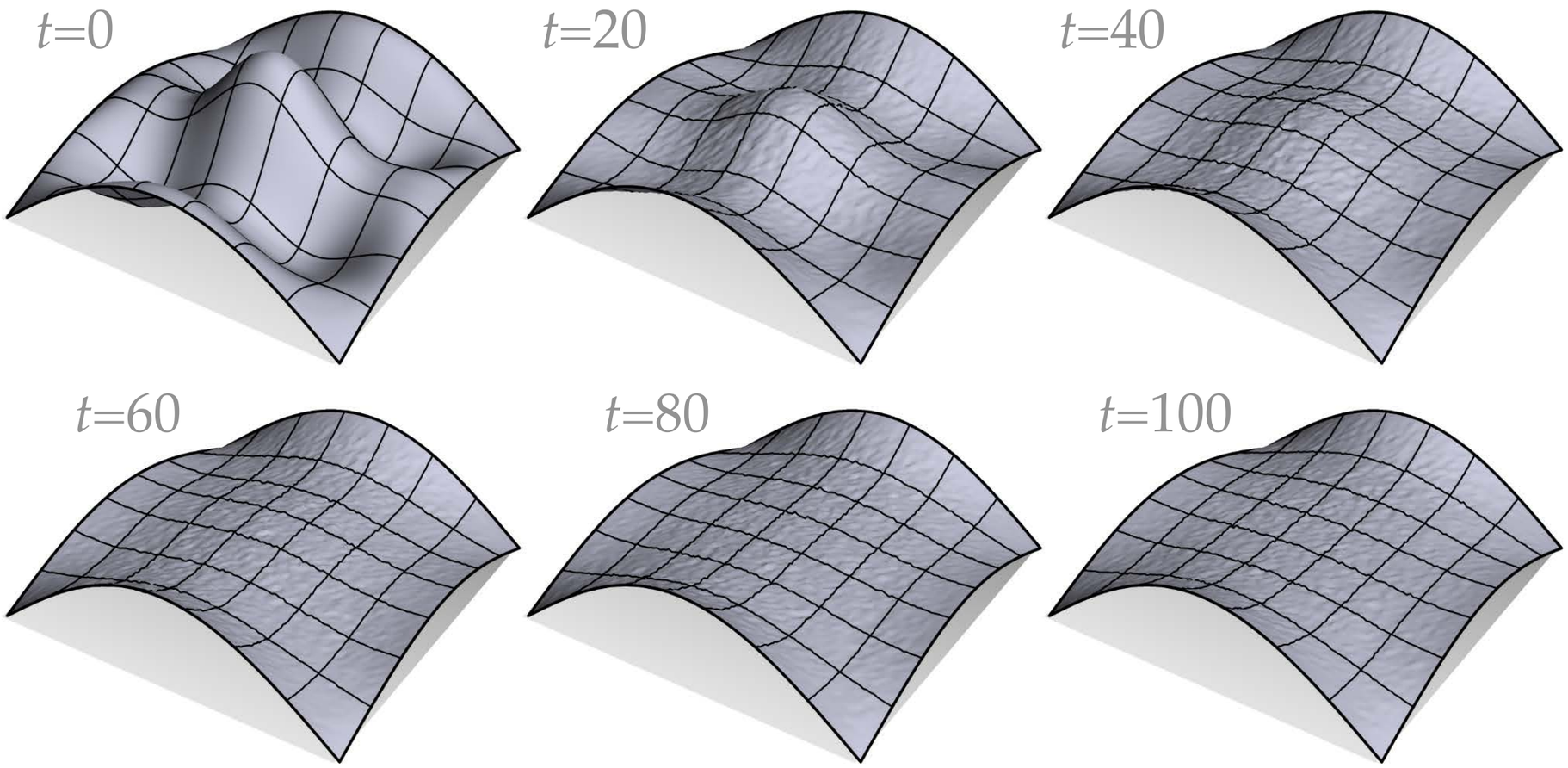
Heat Equation via Walk on Moving Spheres

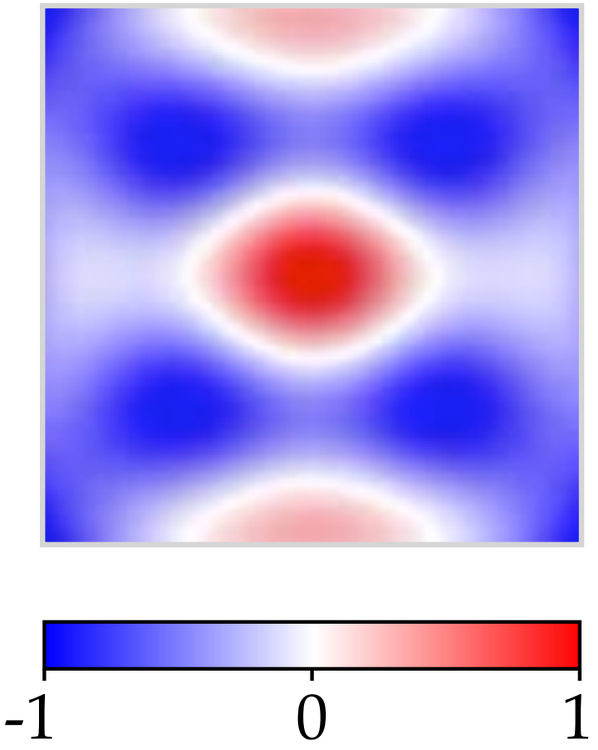




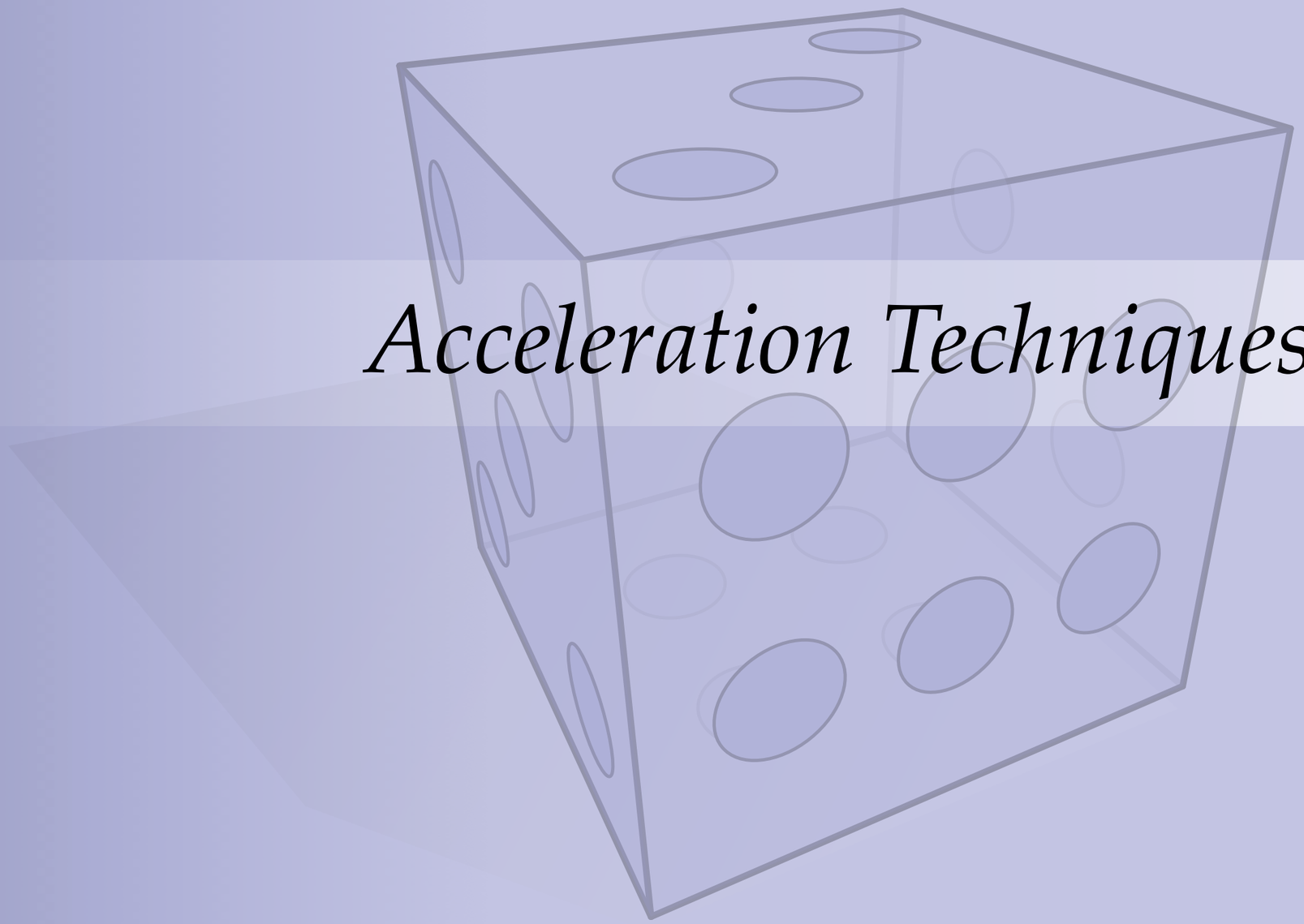
(using a more careful algorithm due to S. Schwarz)

Heat Equation via Walk on Moving Spheres





(using more a careful algorithm due to S. Schwarz)



Acceleration Techniques

Importance Sampling

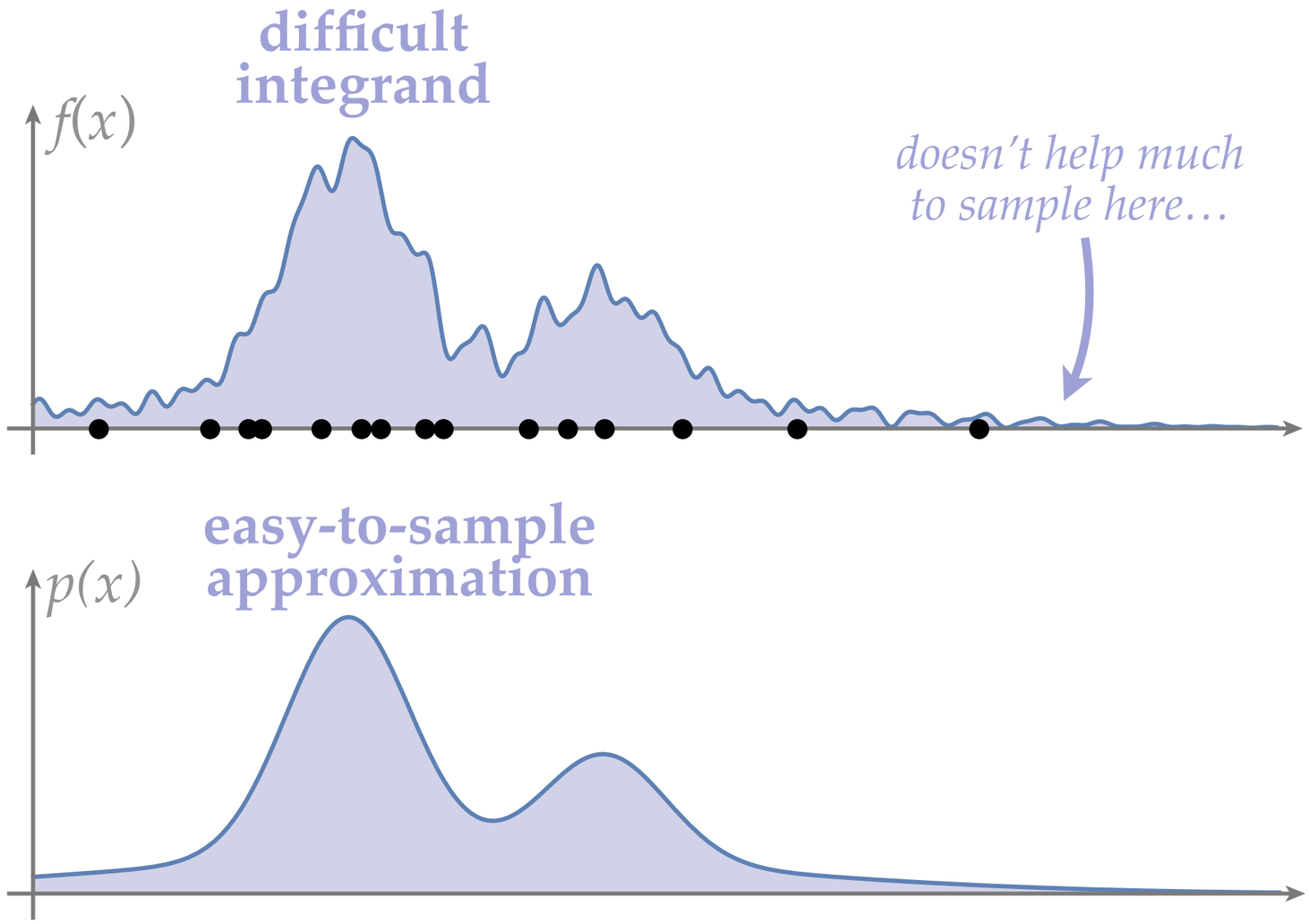
$$\int_\Omega f(x)\,dx$$

importance sampled Monte Carlo estimator

$$\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)},\quad X_i \sim p$$

INTEGRAND

IMPORTANCE DENSITY

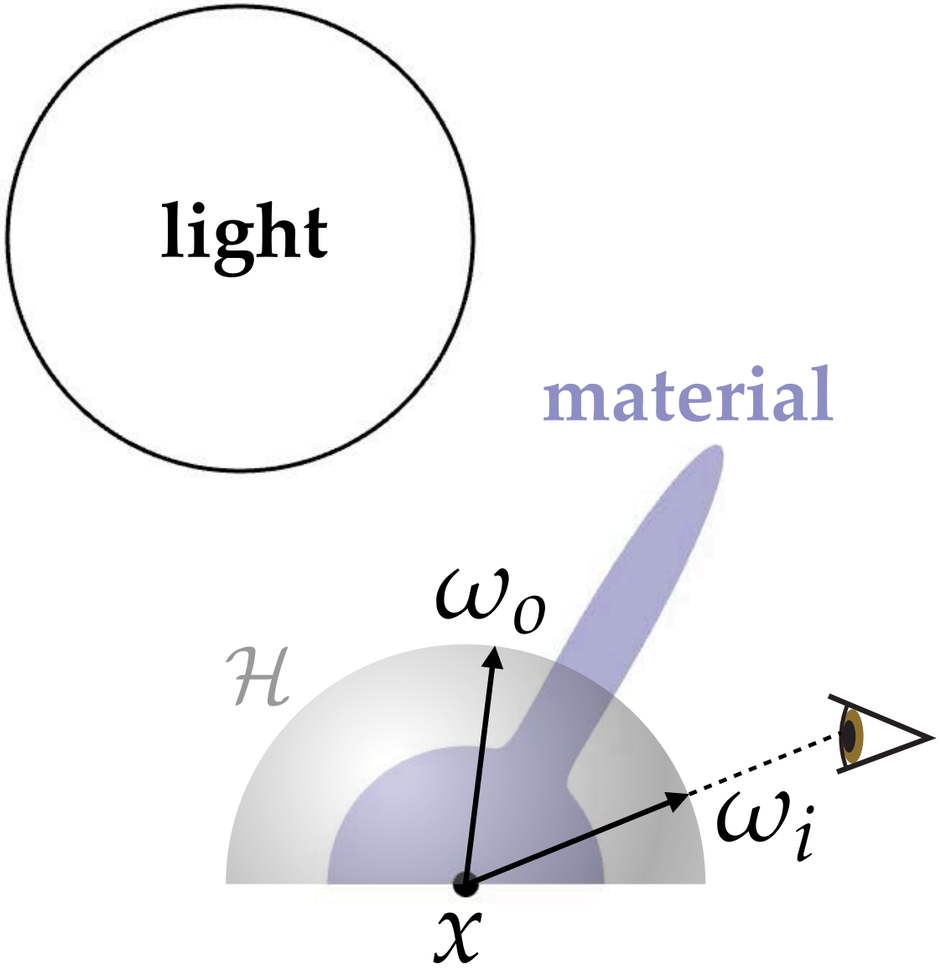


Idea: put more samples in regions that contribute to integral

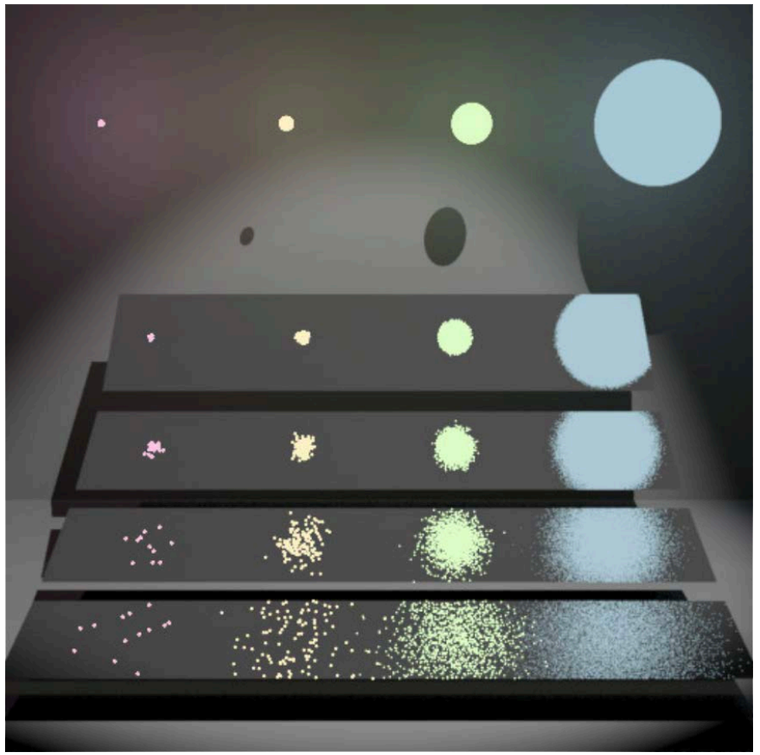
Importance Sampling for Rendering

$$L_o(x, \omega_o) = \int_{\mathcal{H}} f_r(\omega_i, \omega_o) L_i(x, \omega_i) \cos(\theta) d\omega_i$$

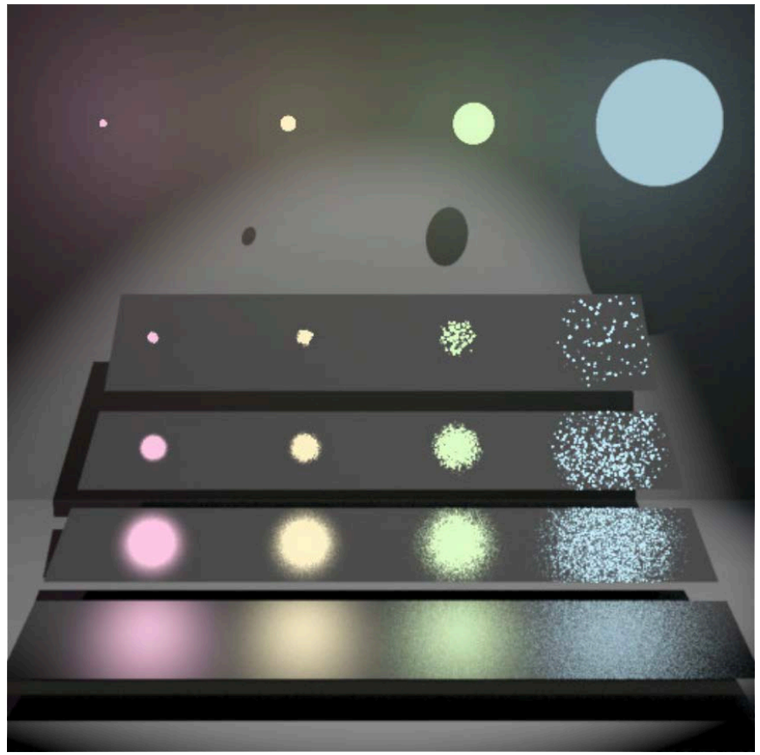
materials *lights*



SAMPLE LIGHTS



SAMPLE MATERIALS



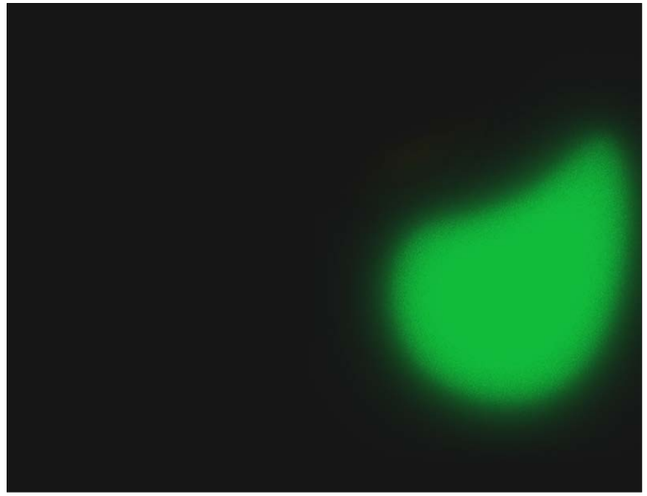
Importance Sampling for PDEs

SAMPLE SOURCES

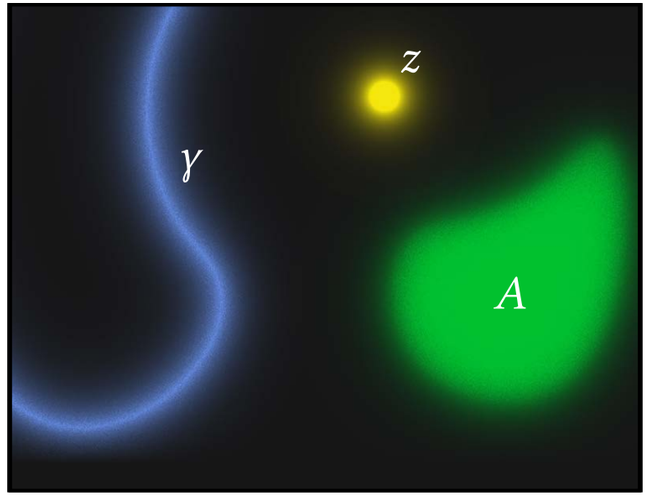
$$\int_{B(x)} G(x,y) f(y)\, dy$$

source term

uniform



sample sources

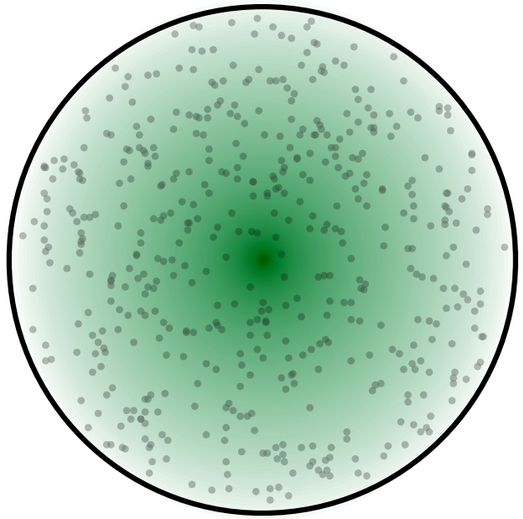


SAMPLE GREEN'S FUNCTION

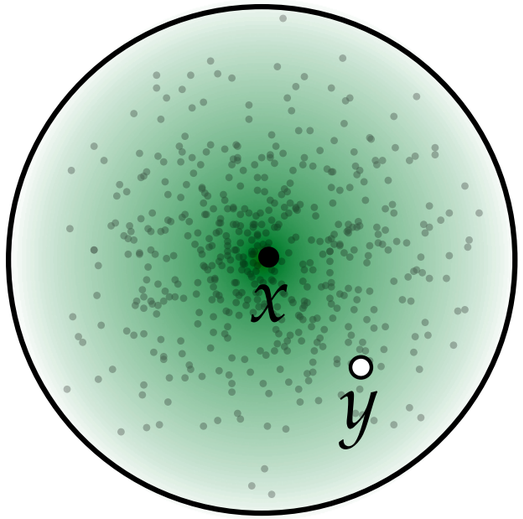
$$\int_{B(x)} G(x,y) f(y)\, dy$$

Green's function

uniform

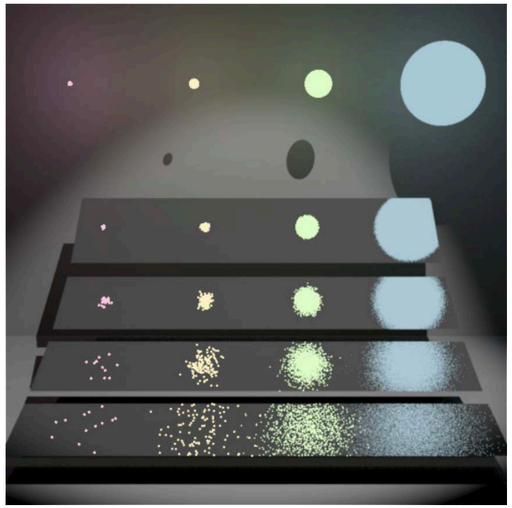


sample Green's function

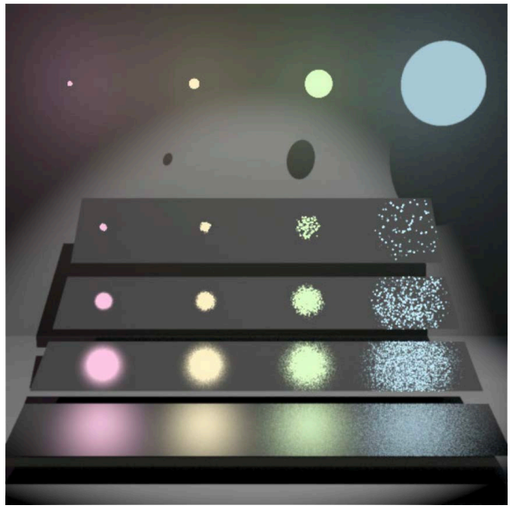


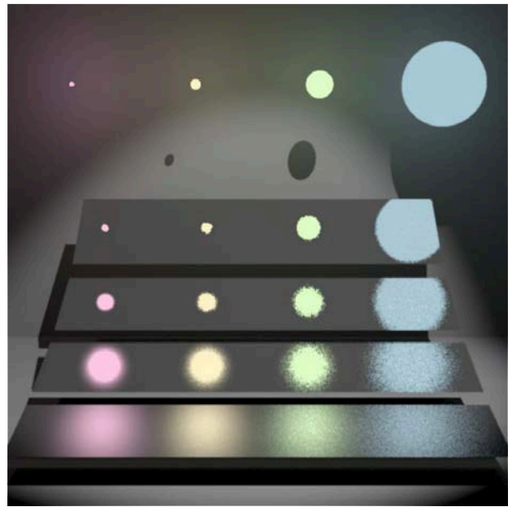
Multiple Importance Sampling—Rendering

sample lights



sample materials





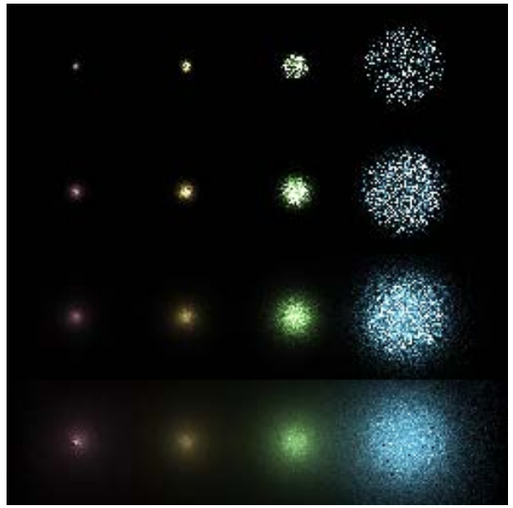
multiple importance sampling

"Optimally combining sampling techniques for Monte Carlo rendering"
Veach & Guibas (SIGGRAPH 1995)

Multiple Importance Sampling—PDEs

$$\Delta u + cu = f$$

sample sources



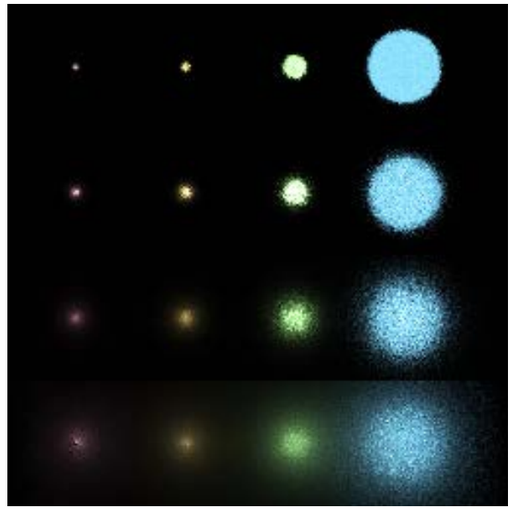
sample Green's function

$c = 10^5$

$c = 10^4$

$c = 10^3$

$c = 10^2$

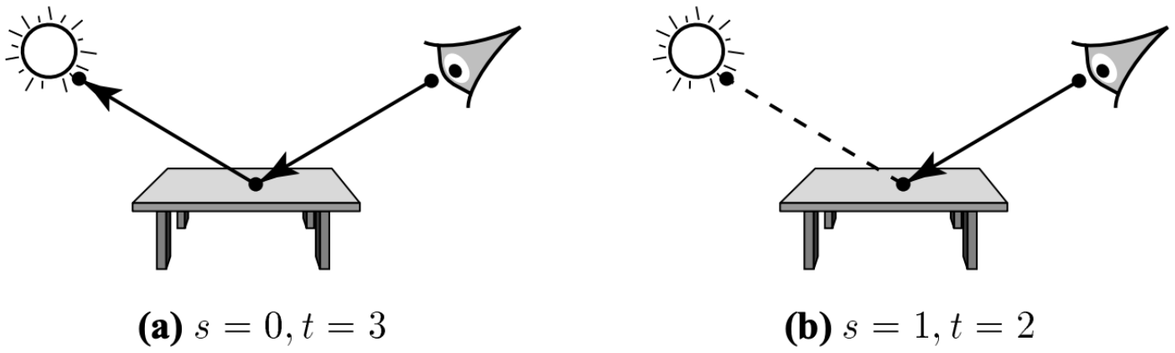


multiple importance sampling

[Sawhney & C. 2020]

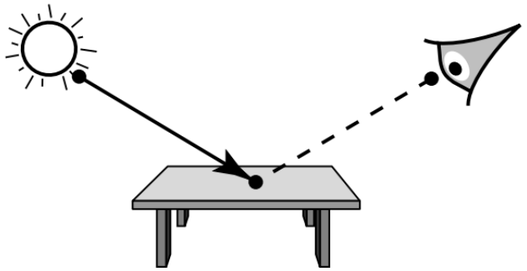
Path Reuse

"Bidirectional Estimators for Light Transport"
Veach & Guibas (1995)

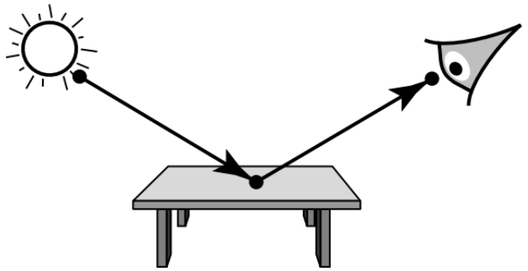


(a) $s = 0, t = 3$

(b) $s = 1, t = 2$



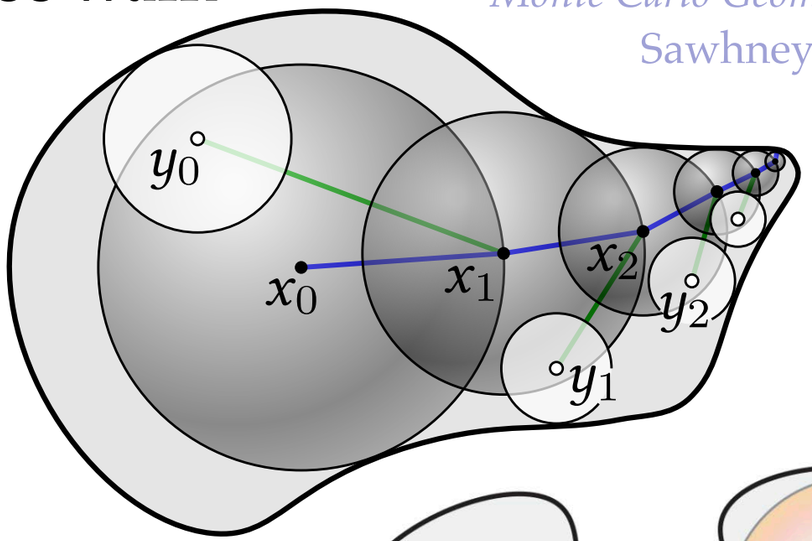
(c) $s = 2, t = 1$

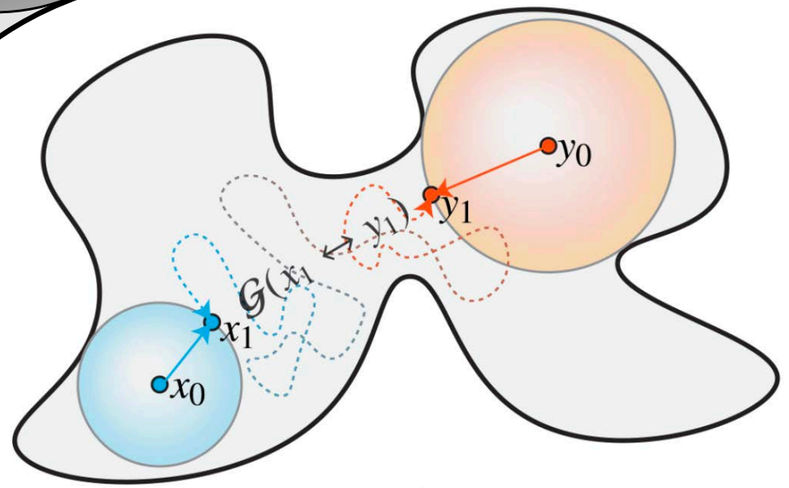


(d) $s = 3, t = 0$

tree walk

"Monte Carlo Geometry Processing"
Sawhney & Crane (2023)

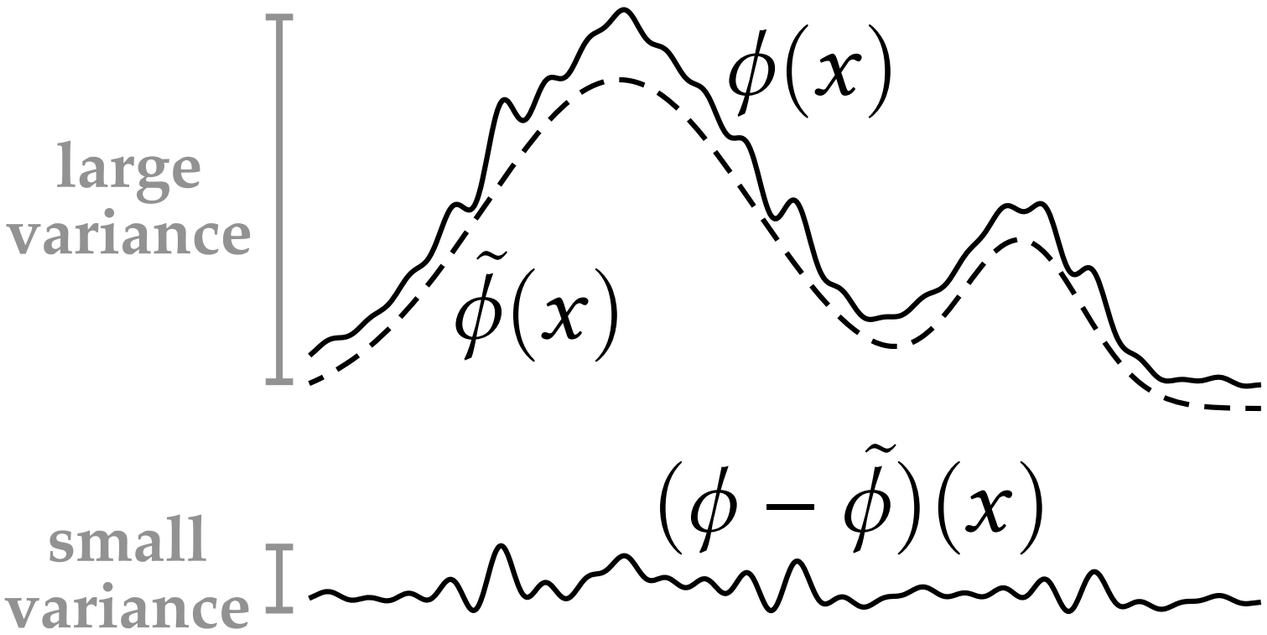


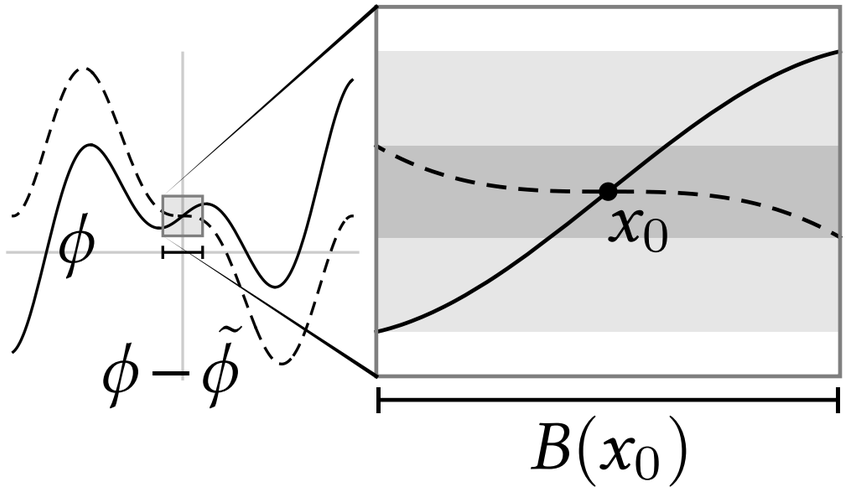


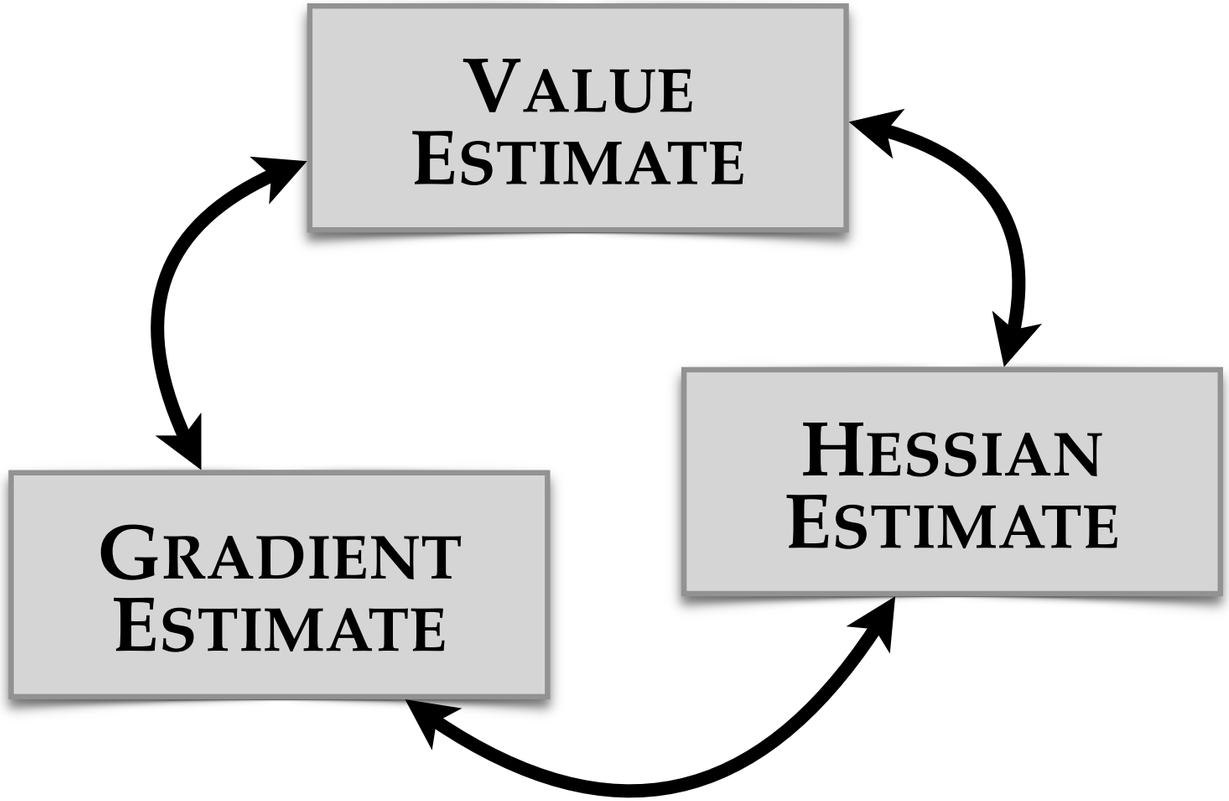
1 sensor segment, 1 source segment

Qi, Seyb, Bitterli, Jarosz,
"A Bidirectional Formulation for Walk on Spheres" (EGSR 2022)

Control Variates



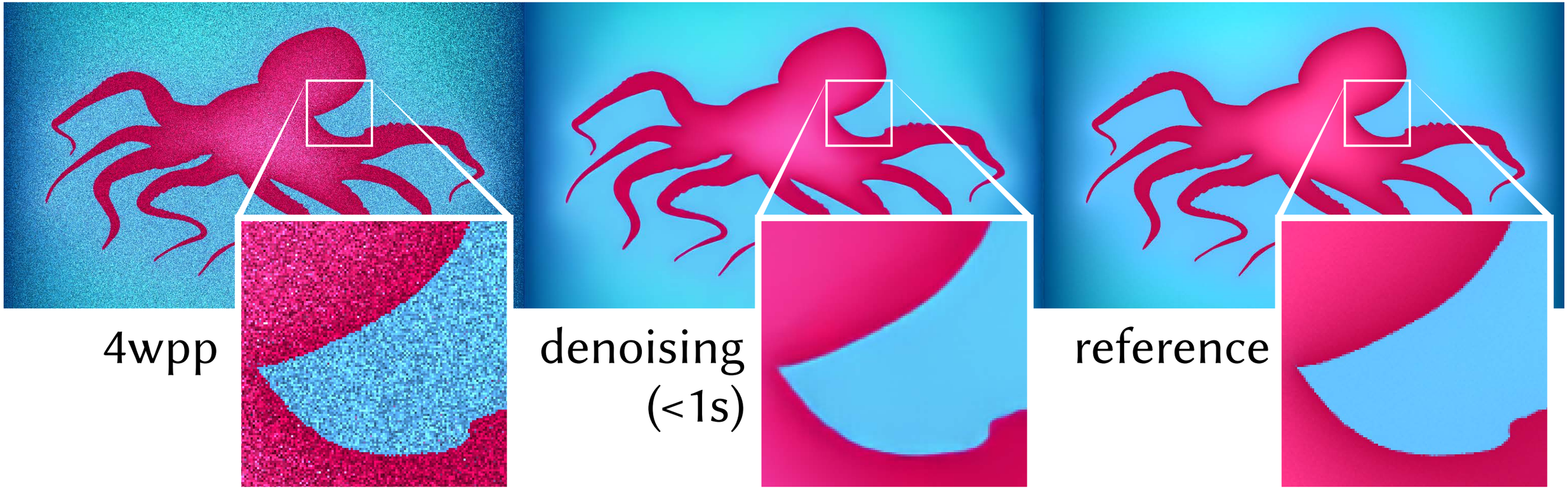


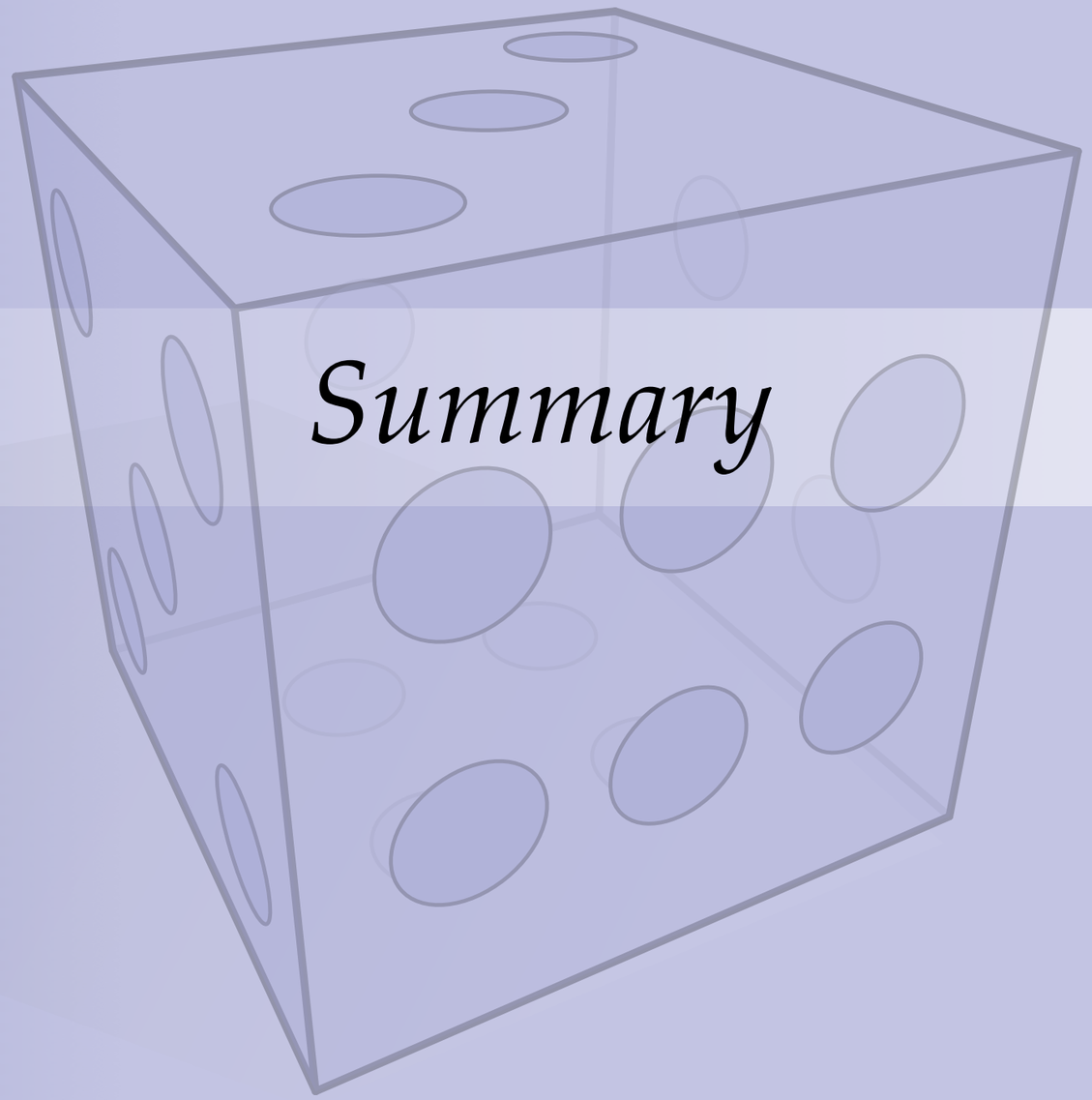


"Monte Carlo Geometry Processing"
Sawhney & Crane (2023)

Denoising

Can use some existing techniques from rendering directly for PDEs:



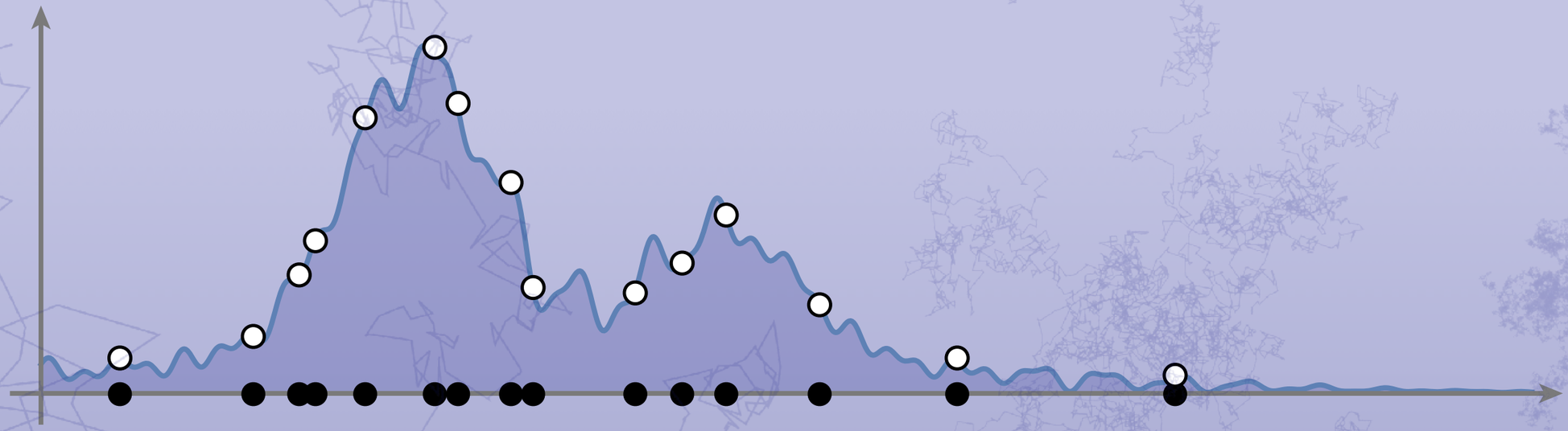


Summary

Summary

- **Walk on Spheres.** For simulation & PDEs, Monte Carlo/*walk on spheres* appears to be a powerful and very "different" alternative to classical methods like FEM
  - inherits many of the advantages of Monte Carlo rendering (scalable with geometric complexity, parallel scaling, output sensitivity, …)
  - not always the right tool for the job!
  - rather than trying to "beat FEM," should think about where unique advantages of WoS open the door to new problems/applications/ideas
- **Potential theory.** More broadly, PDE perspective provides link between stochastic calculus and potential theory
  - food for thought: where else might potential theory offer opportunity for improvements in existing stochastic/Monte Carlo/sampling tools?

Thanks!



Monte Carlo Methods and Applications

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