



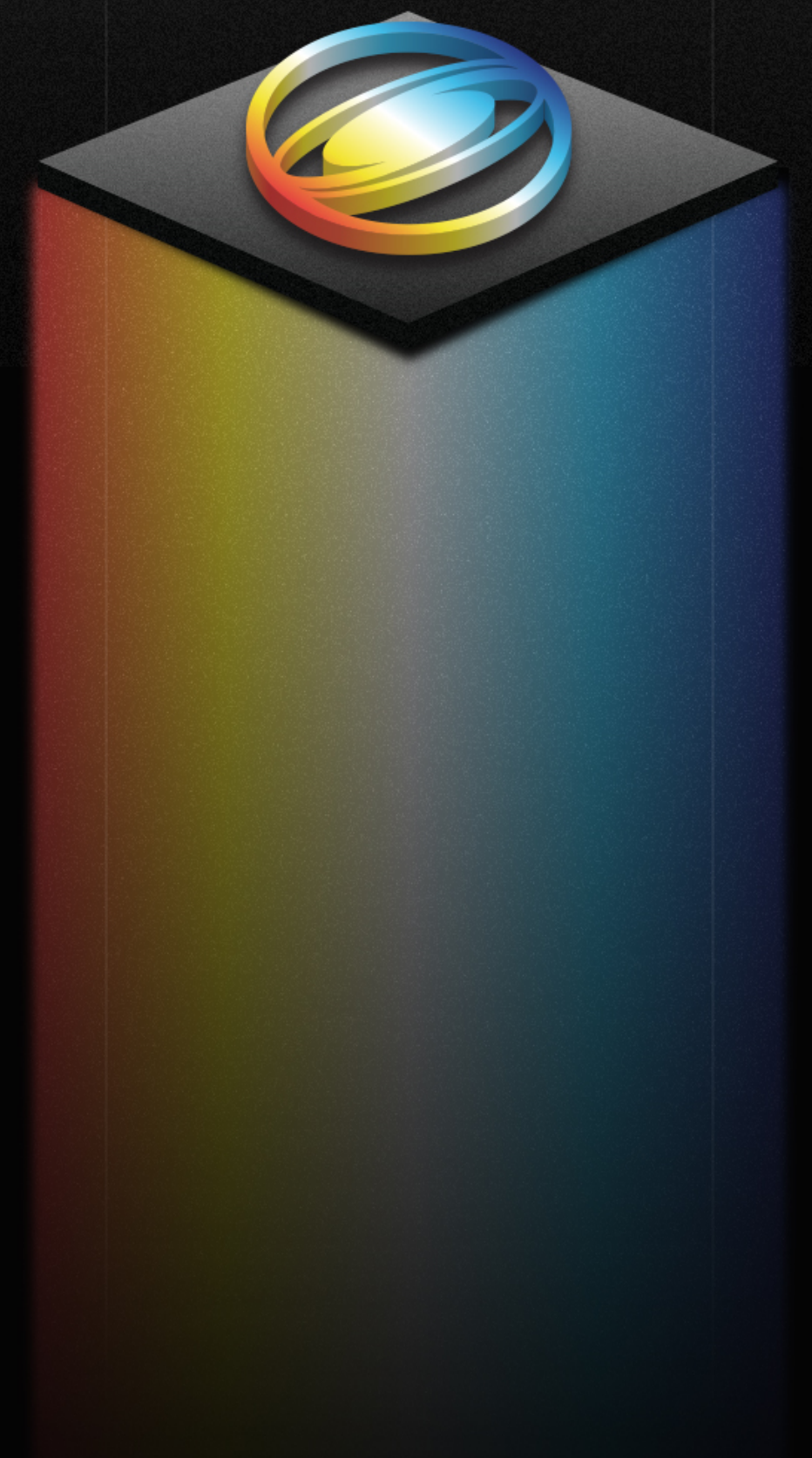
**SIGGRAPH 2024**

DENVER+ 28 JUL — 1 AUG

THE PREMIER CONFERENCE  
& EXHIBITION ON  
COMPUTER GRAPHICS &  
INTERACTIVE TECHNIQUES

# WALKIN' ROBIN: WALK ON STARS WITH ROBIN BOUNDARY CONDITIONS

BAILEY MILLER\*, ROHAN SAWHNEY\*,  
KEENAN CRANE†, IOANNIS GKIOULEKAS†







**Bailey Miller**  
CMU PhD



**Rohan Sawhney**  
High Fidelity Physics @ Nvidia

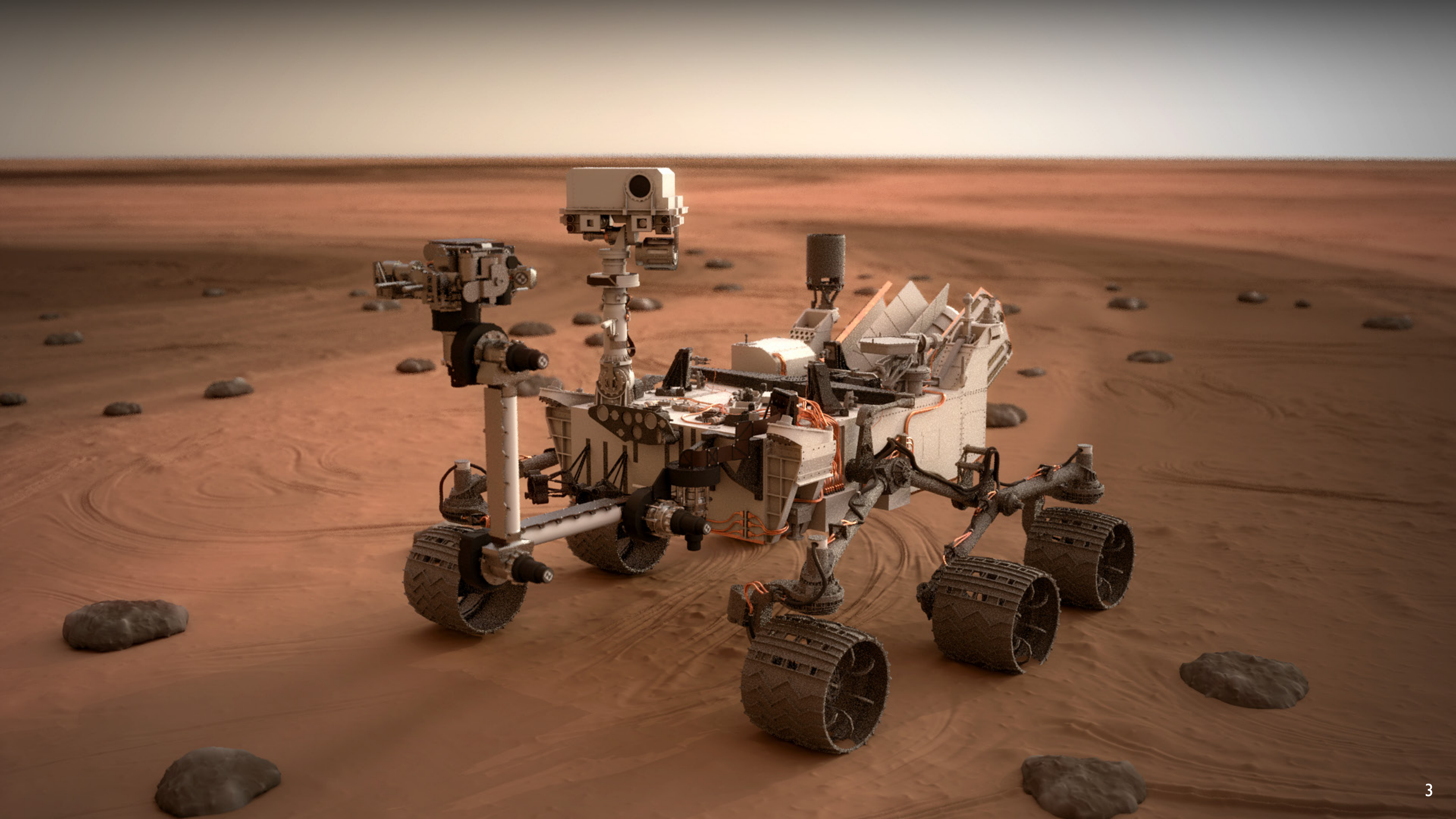


**Keenan Crane**  
CMU



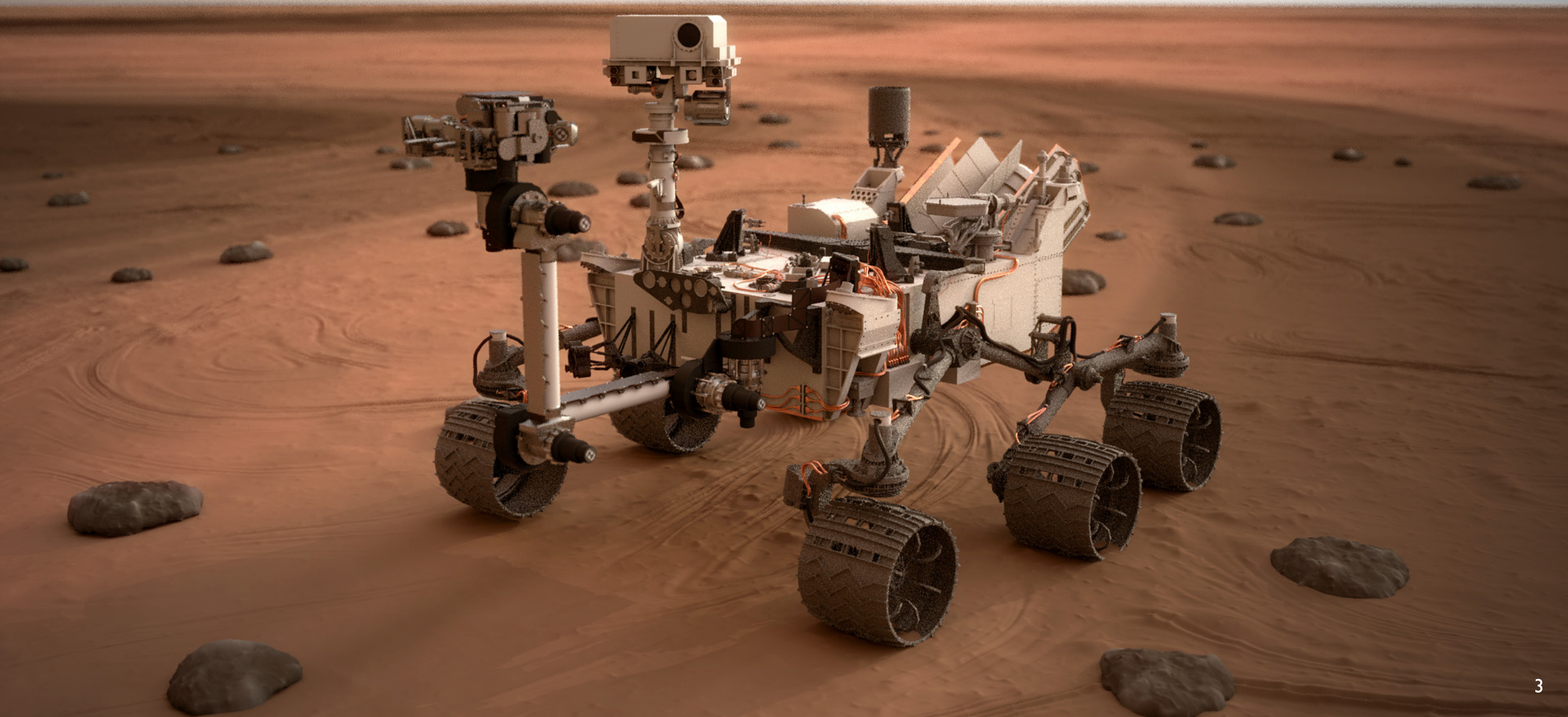
**Ioannis Gkioulekas**  
CMU





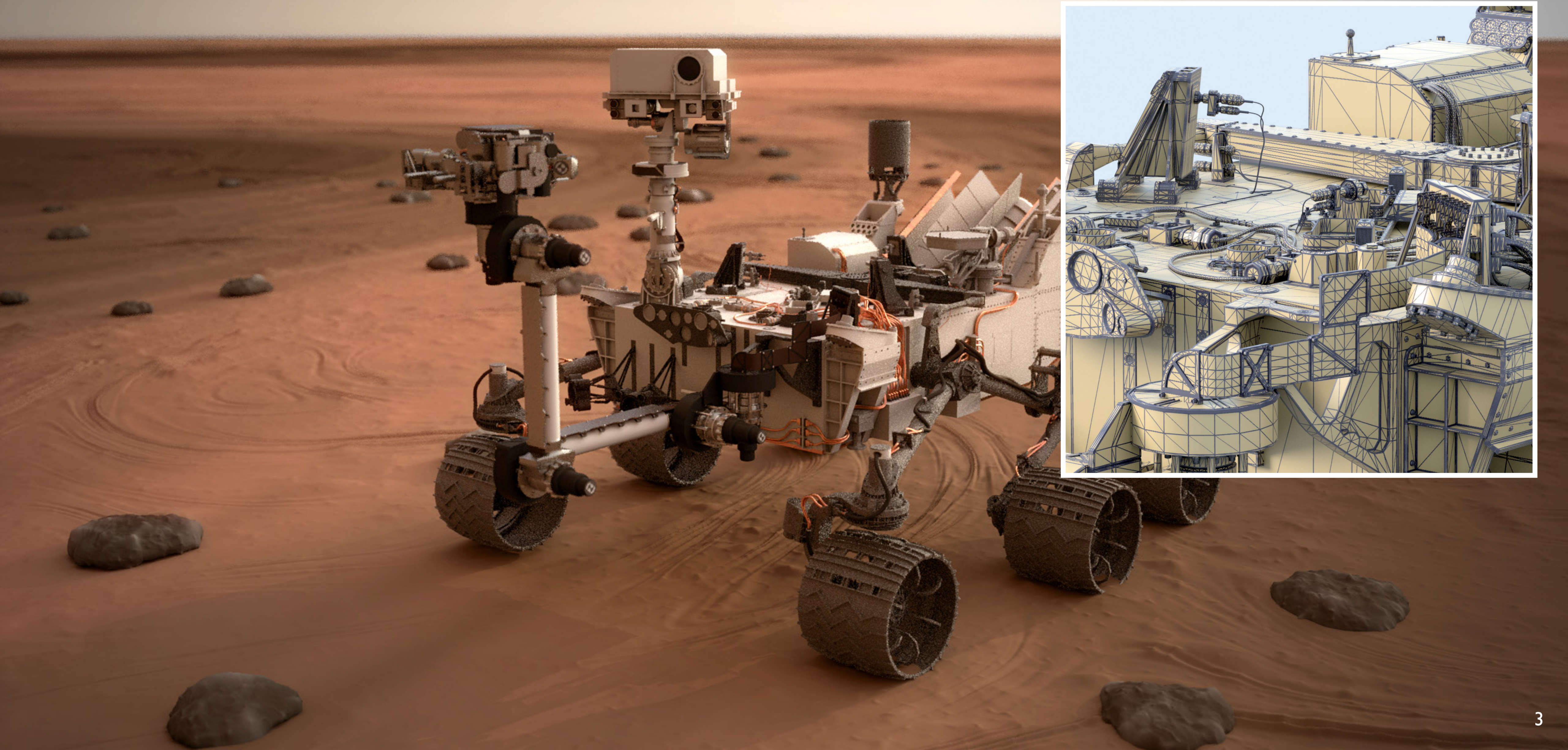


# NASA's *Curiosity* Mars Rover



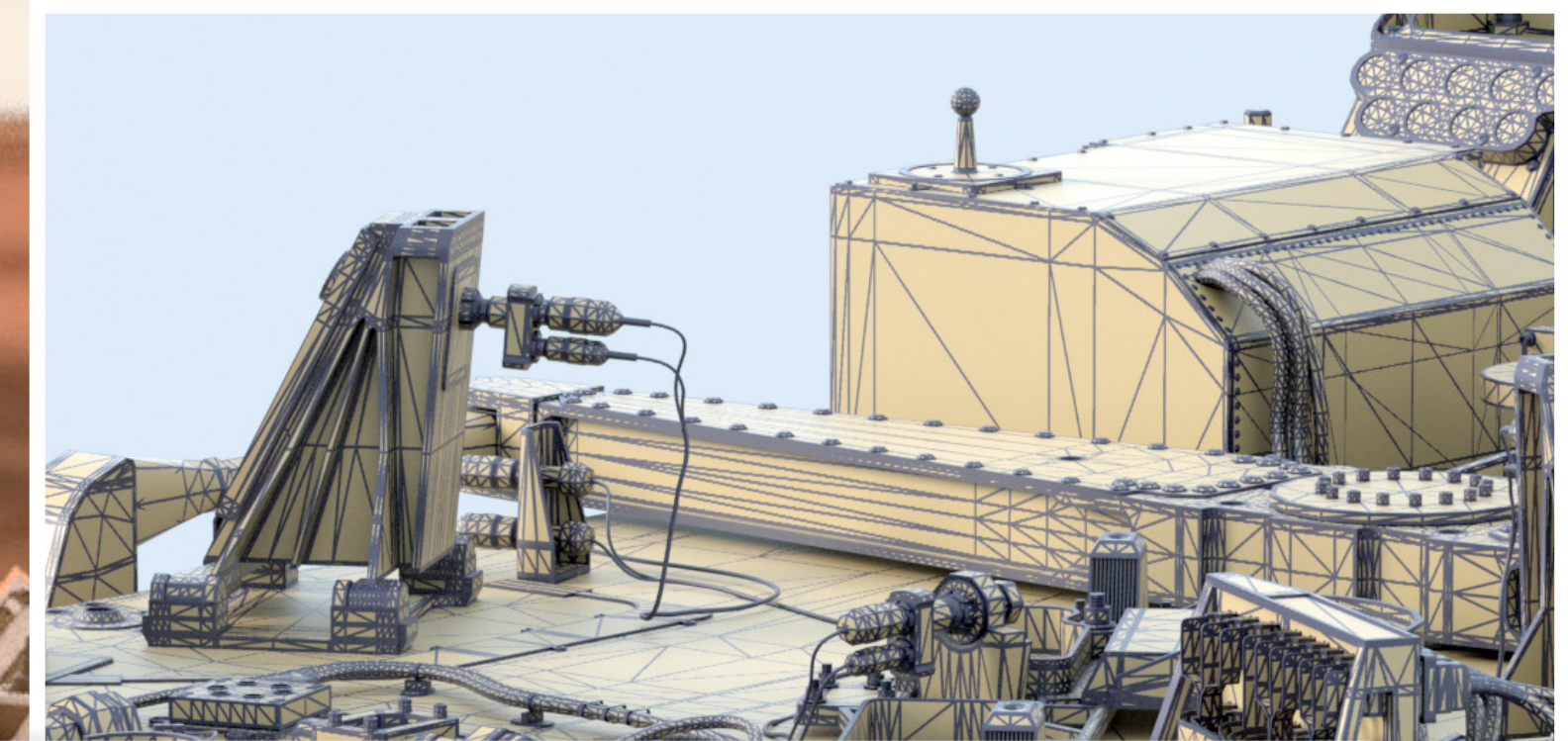
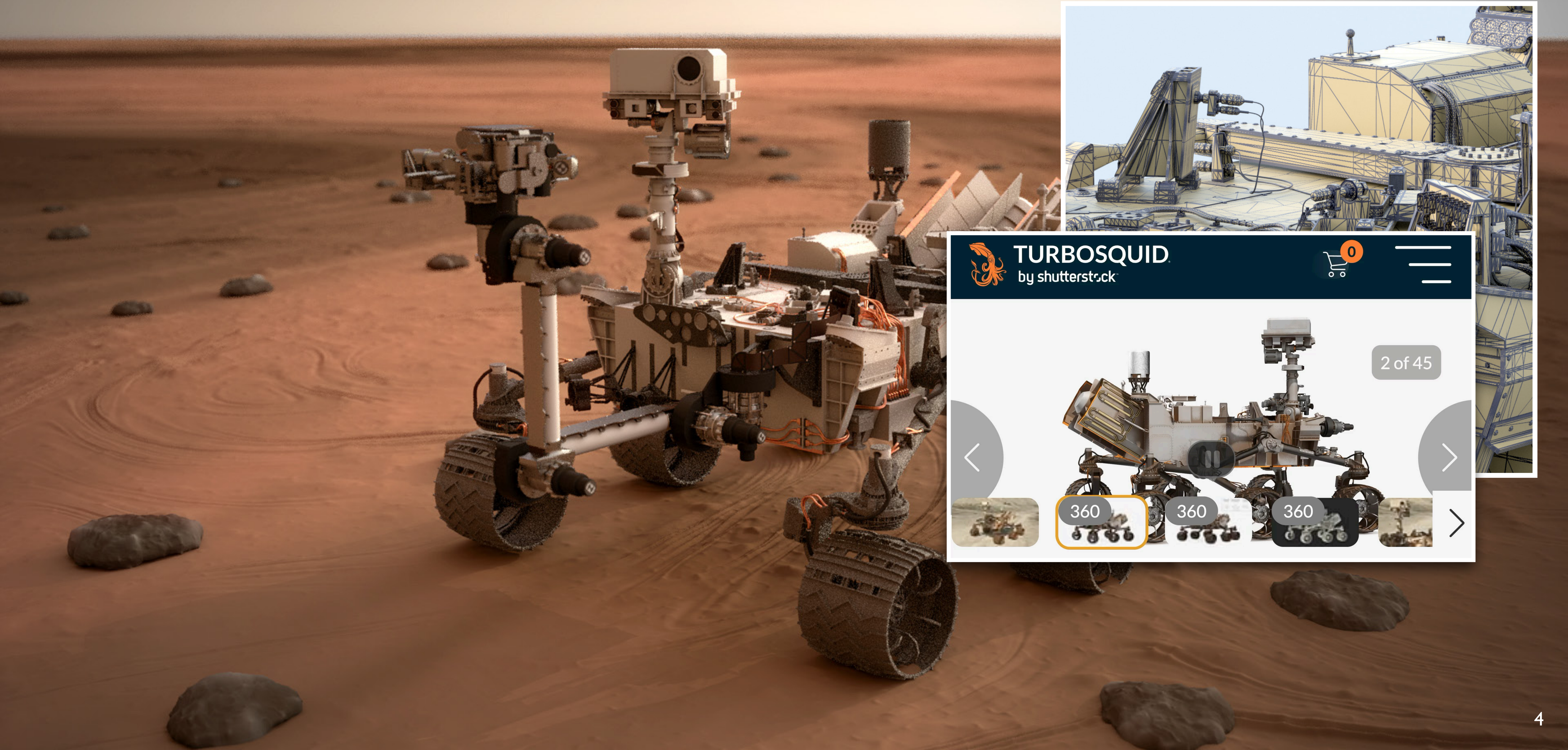


# a rendering of NASA's *Curiosity* Mars Rover





# a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover



**TURBOSQUID**  
by shutterstock

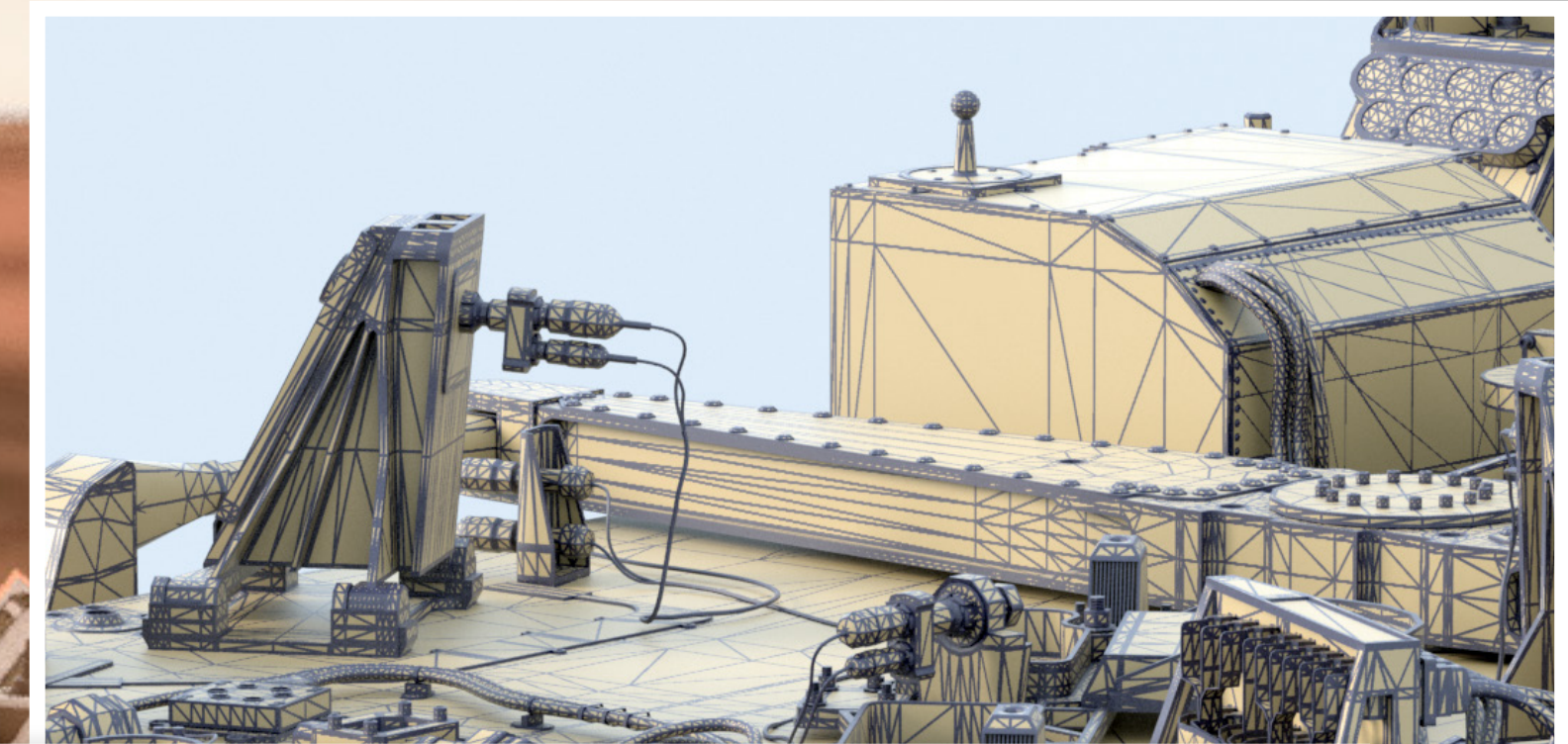
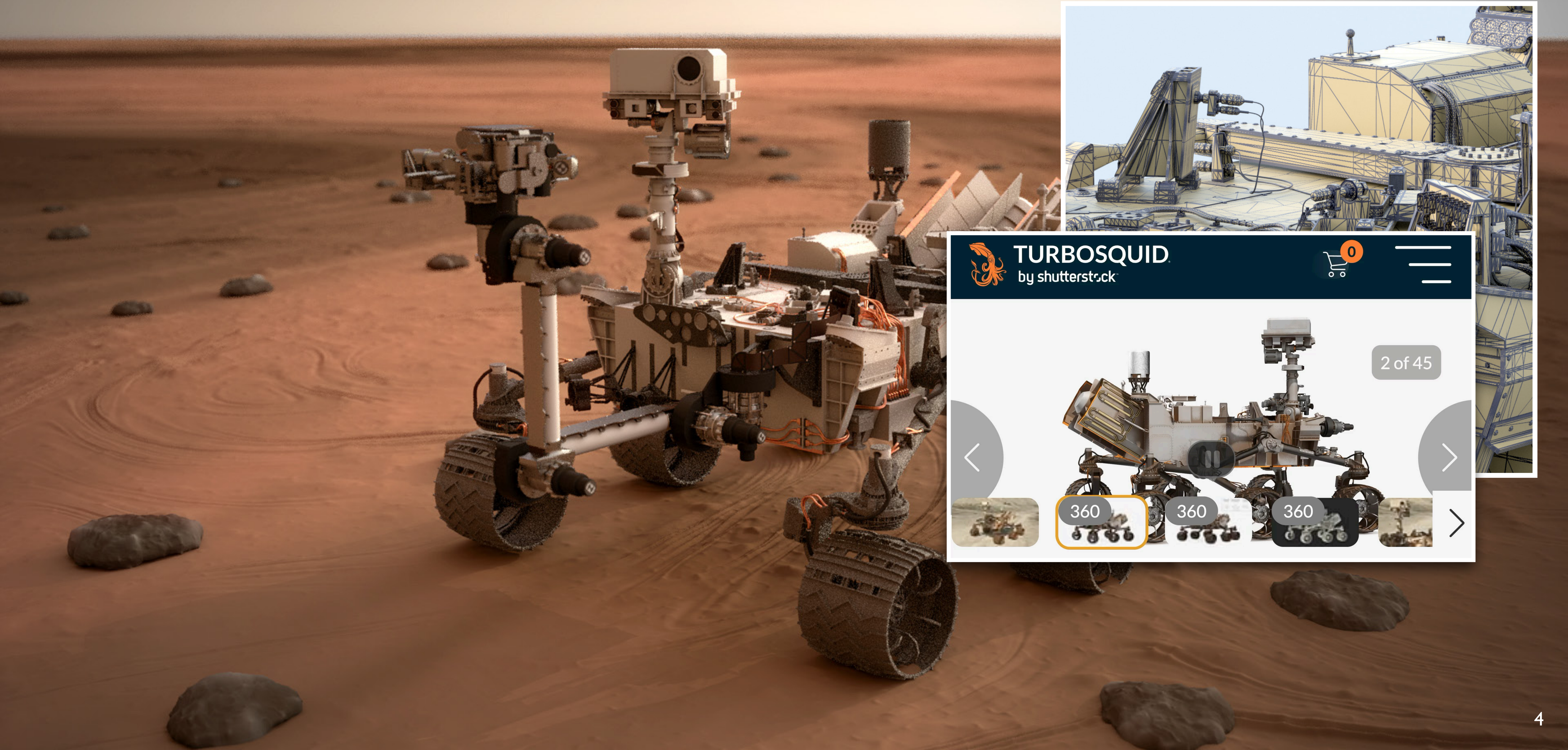
0

2 of 45

360 360 360



# a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover



**TURBOSQUID**  
by shutterstock

0

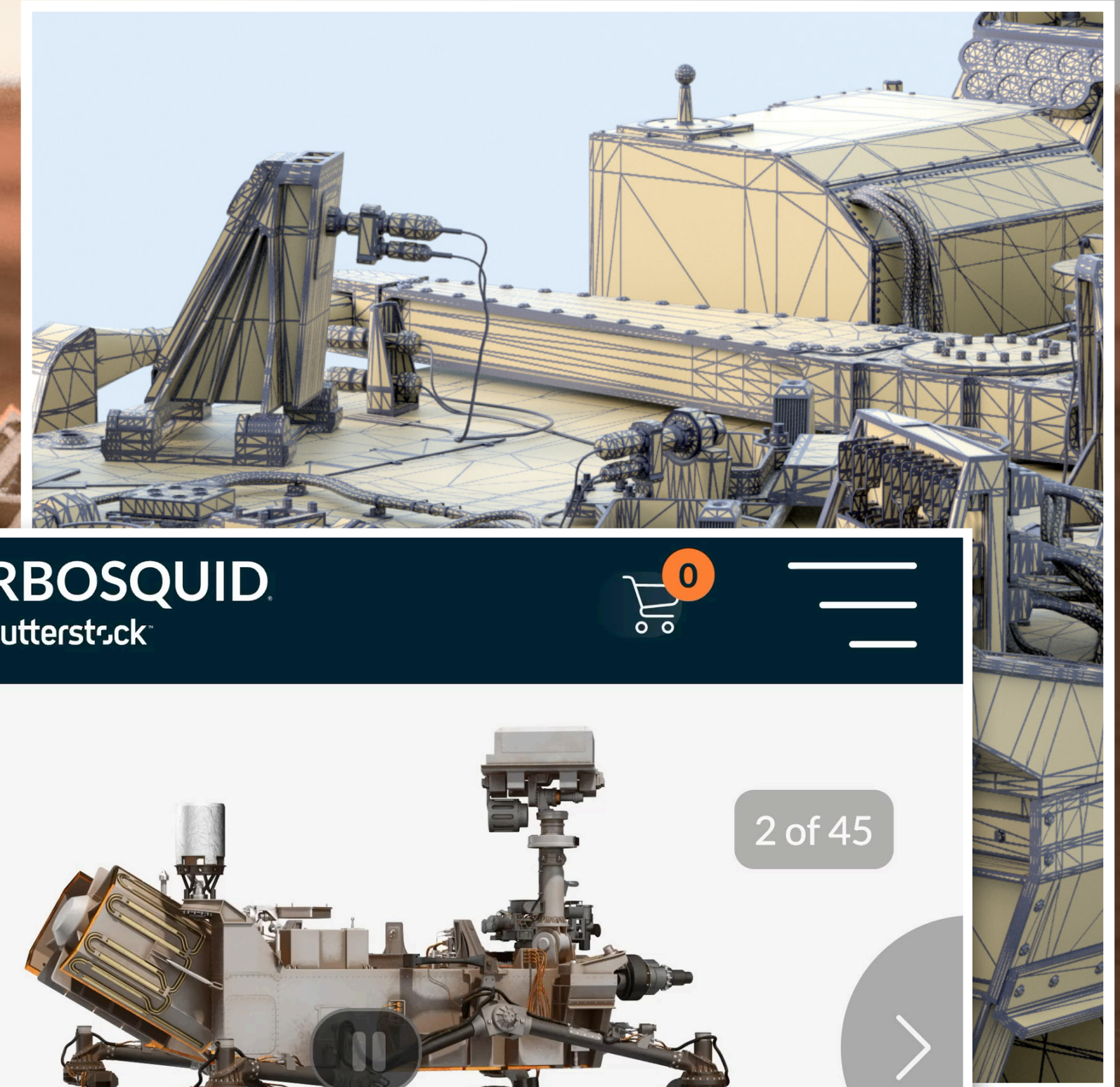
2 of 45

360 360 360



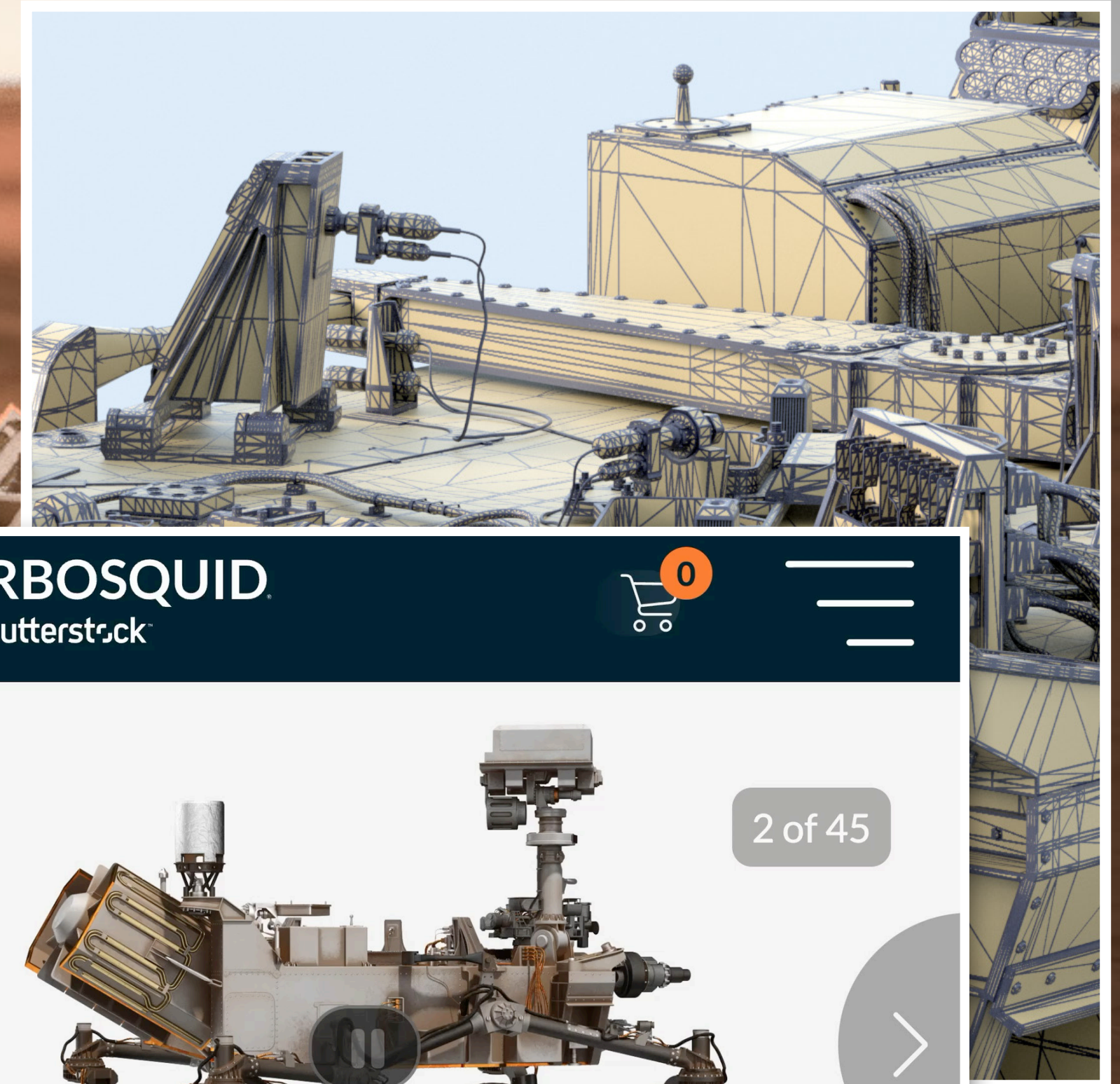
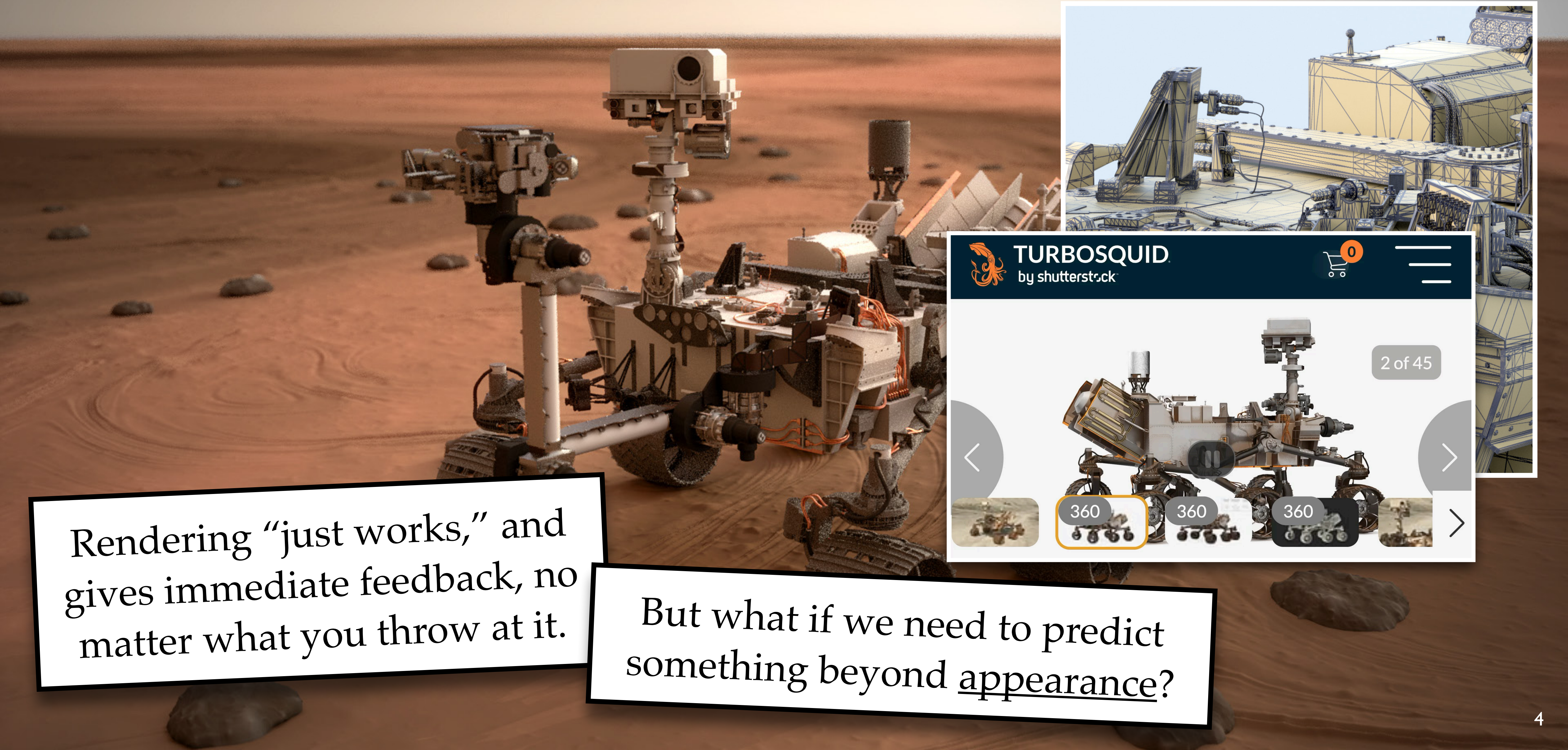
# a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover

Rendering "just works," and gives immediate feedback, no matter what you throw at it.





# a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover

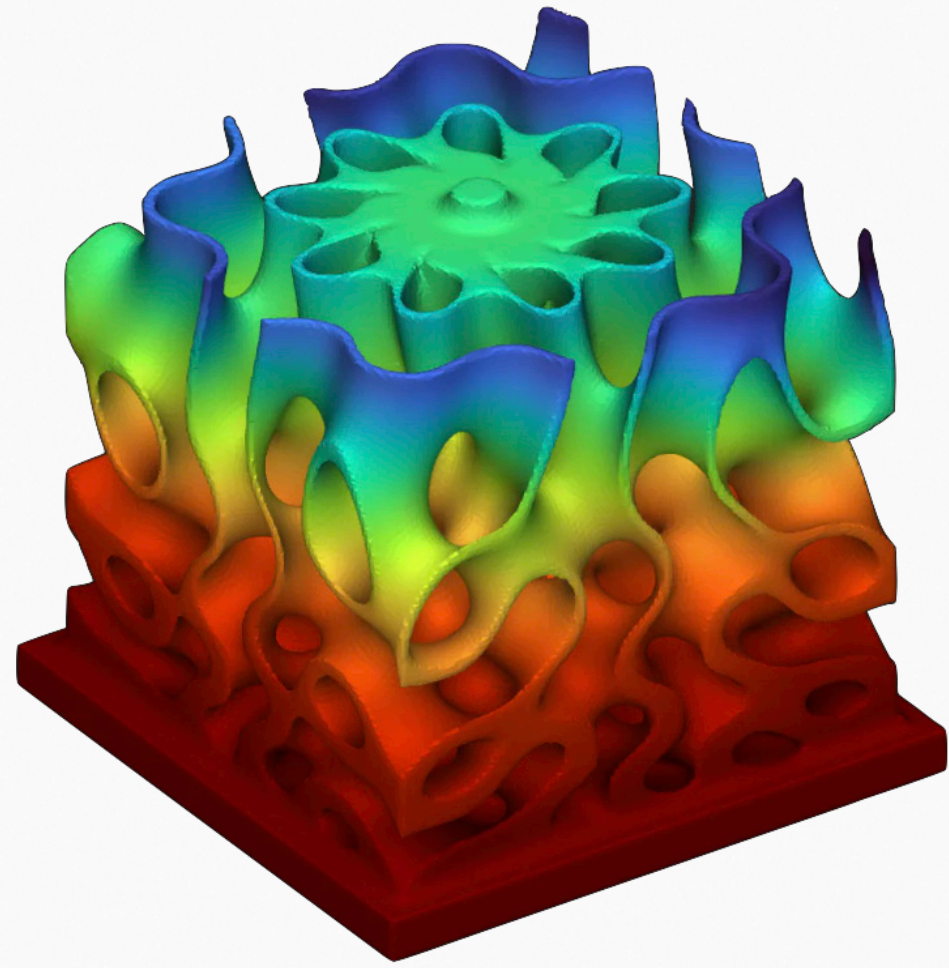


Rendering “just works,” and gives immediate feedback, no matter what you throw at it.

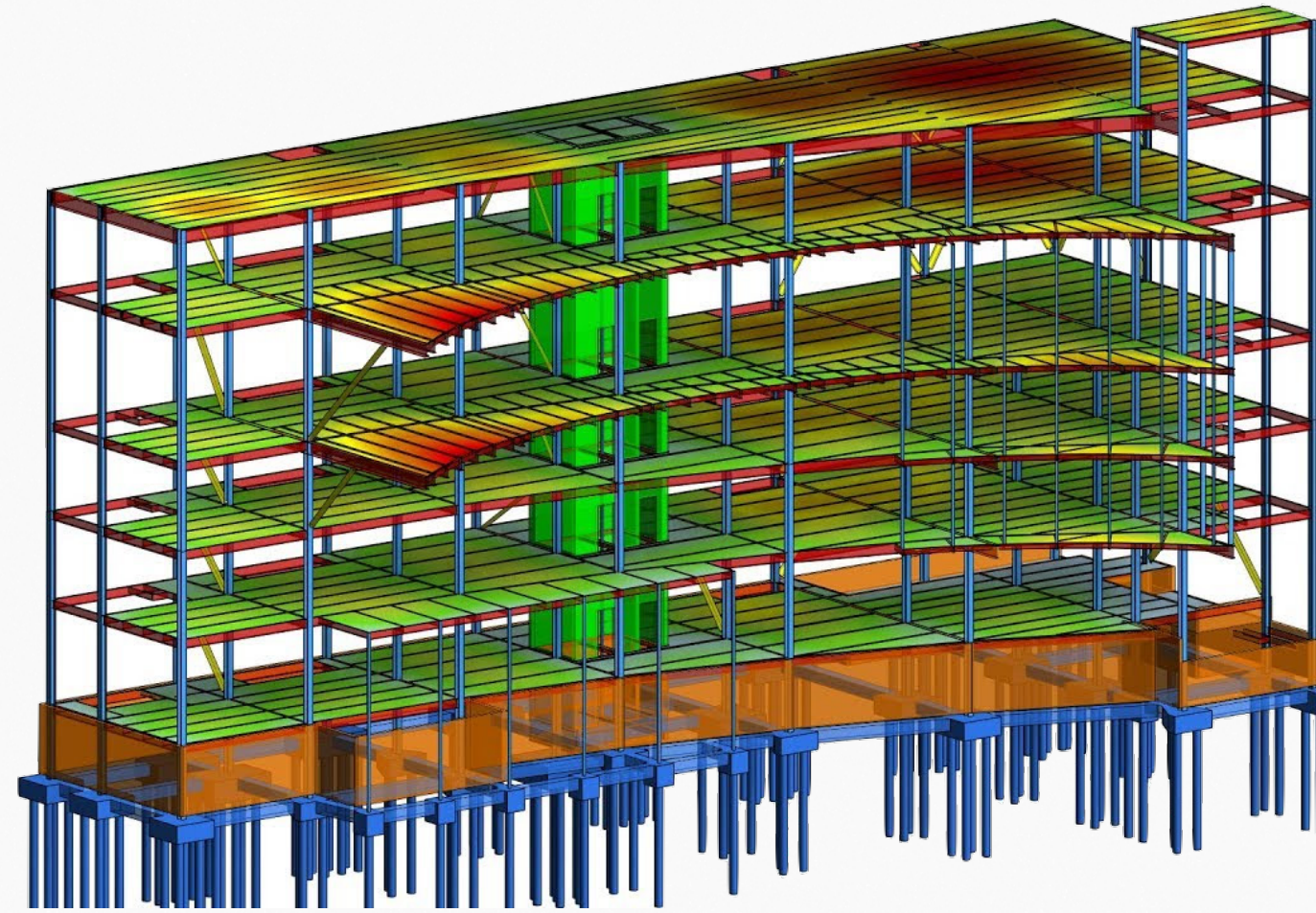
But what if we need to predict something beyond appearance?



# physics beyond light transport



thermal diffusion



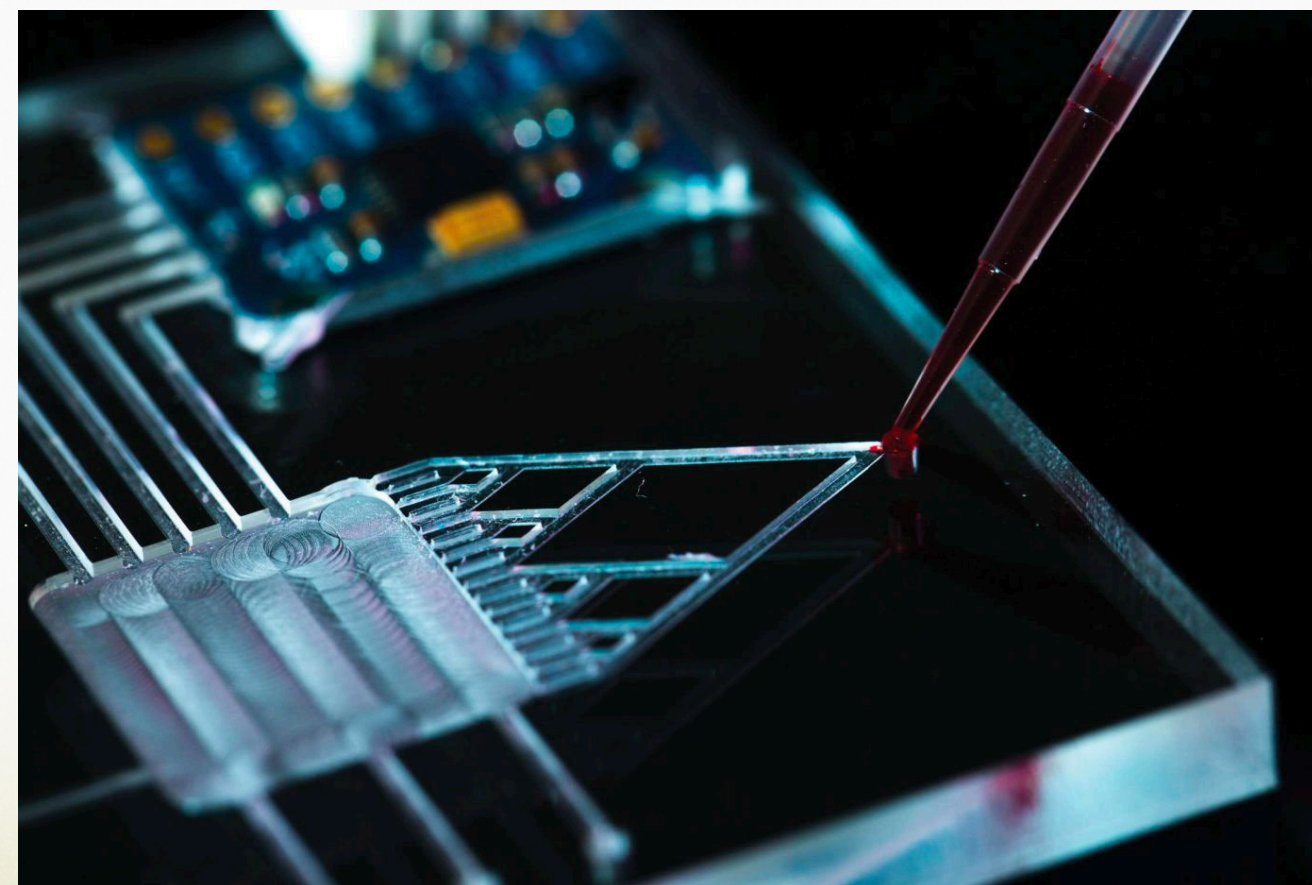
structural analysis



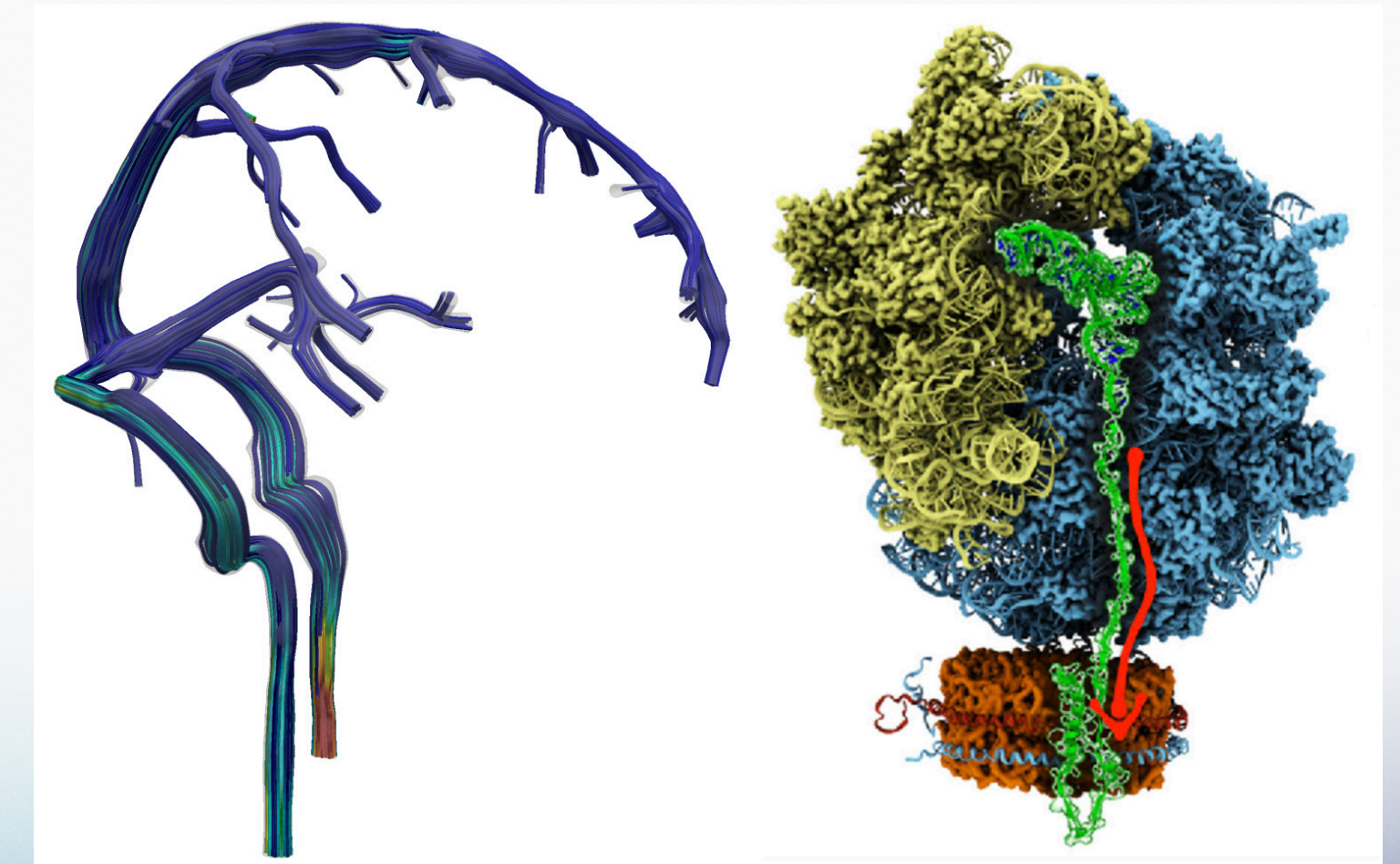
electrostatics



acoustic modeling



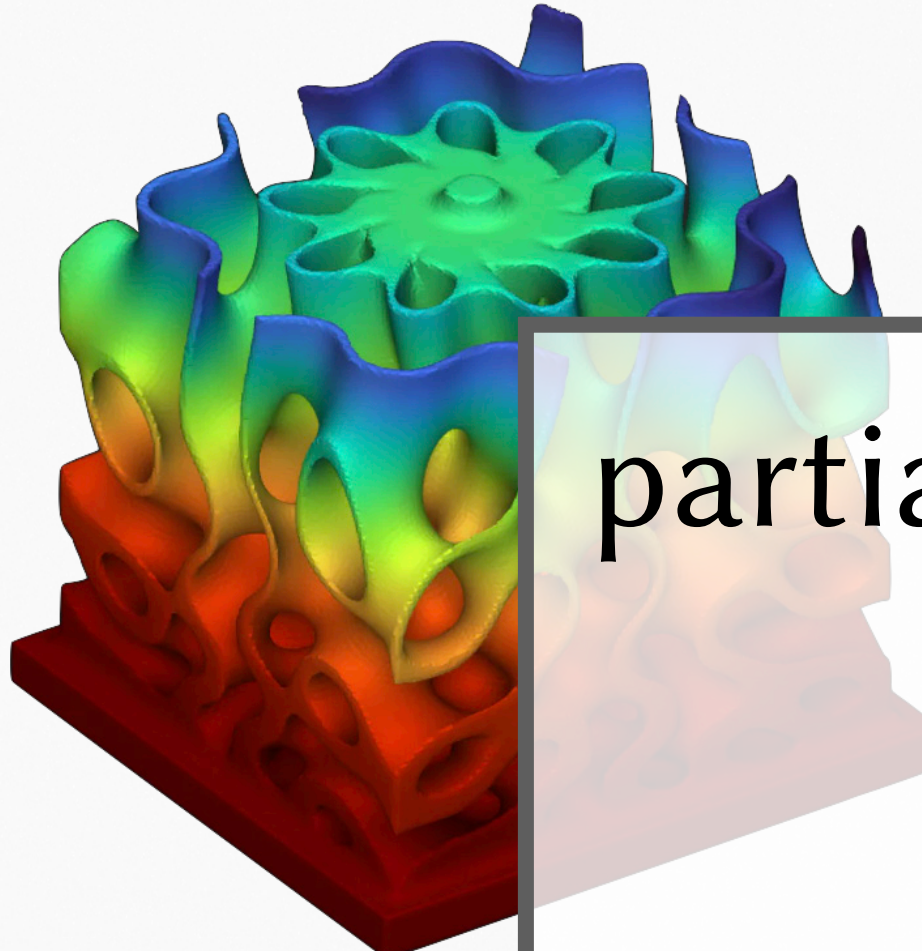
microfluidics



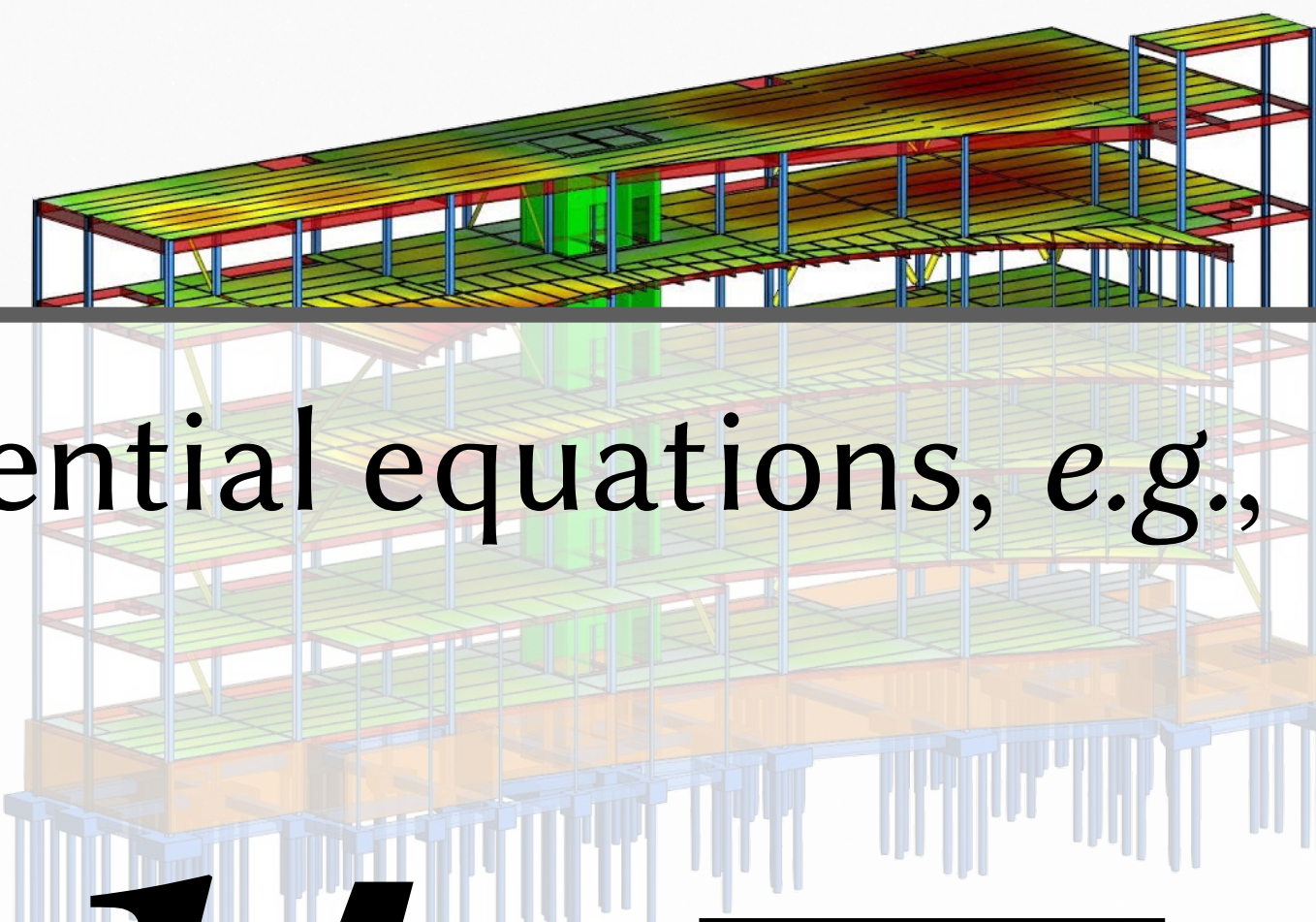
biophysics



# physics beyond light transport



thermal diffusion



structural analysis

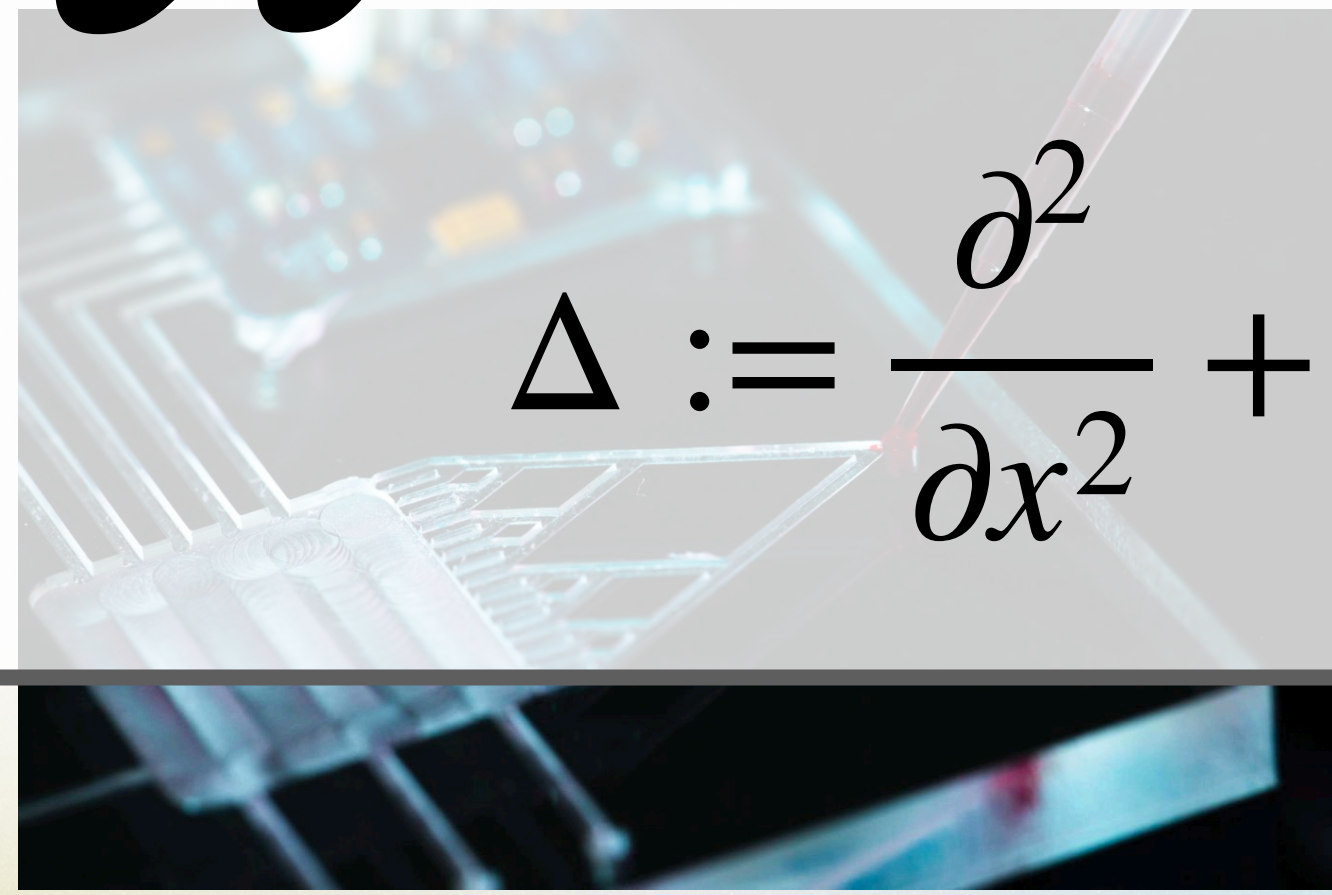


electrostatics

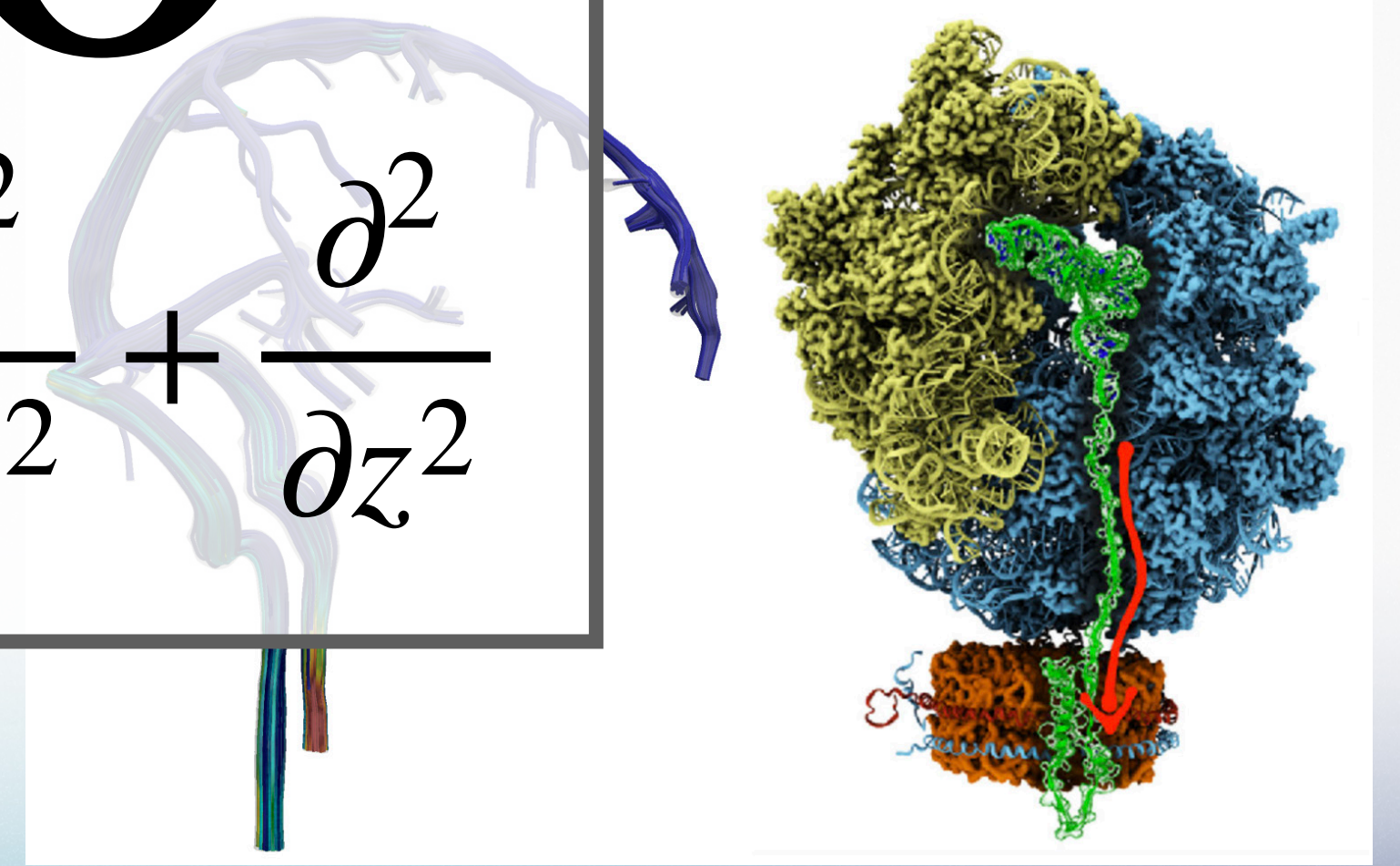
partial differential equations, e.g., Laplace eq.

$$\Delta u = 0$$
$$\Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$


acoustic modeling



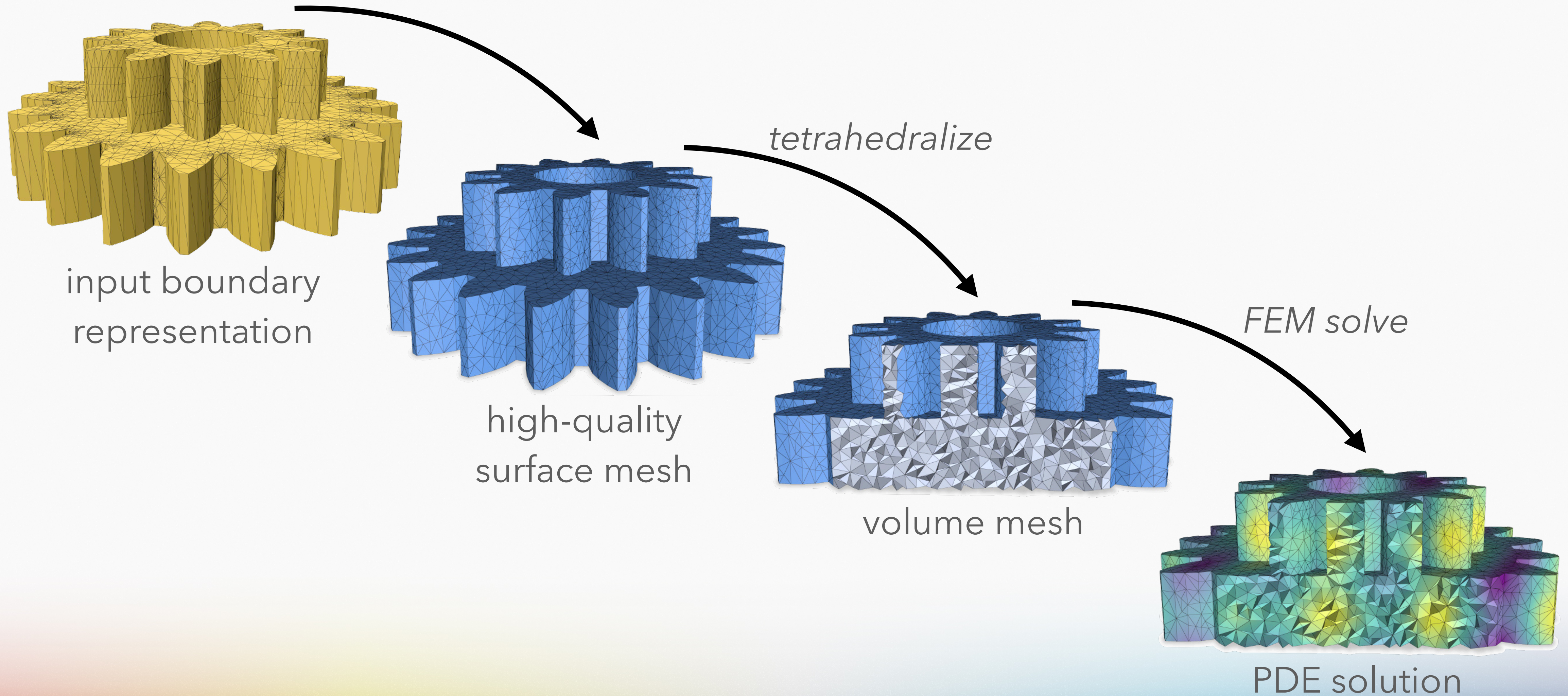
microfluidics



biophysics

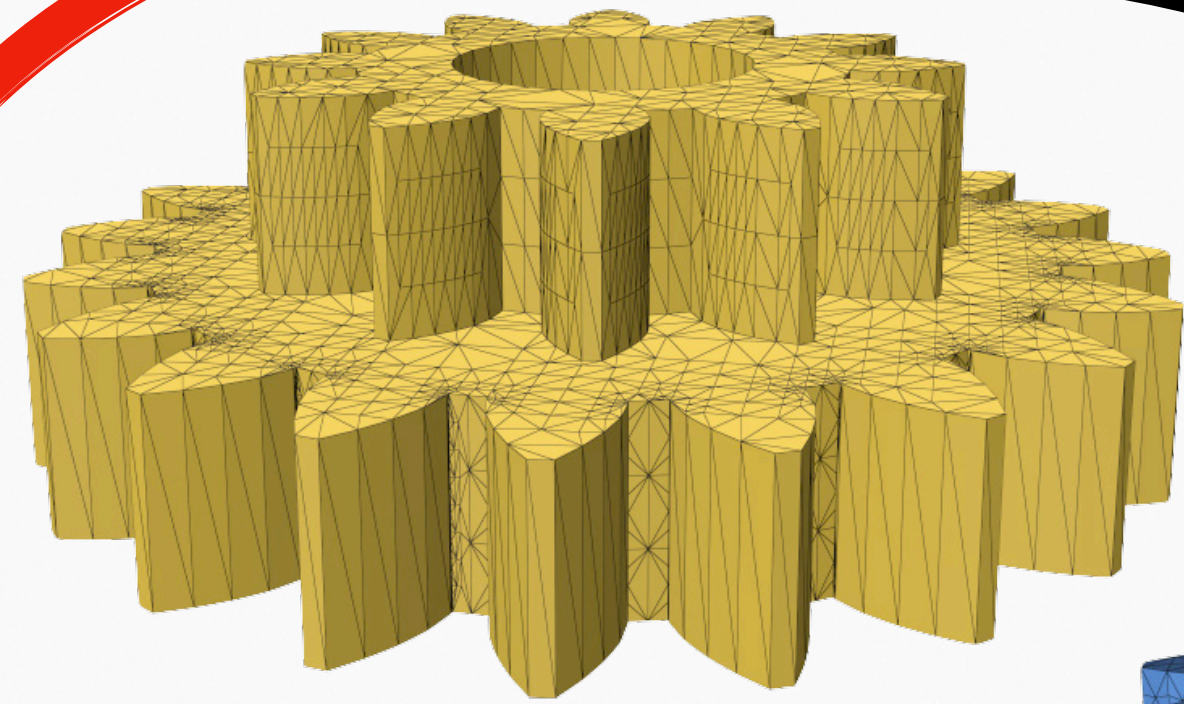


# finite element method (FEM) pipeline

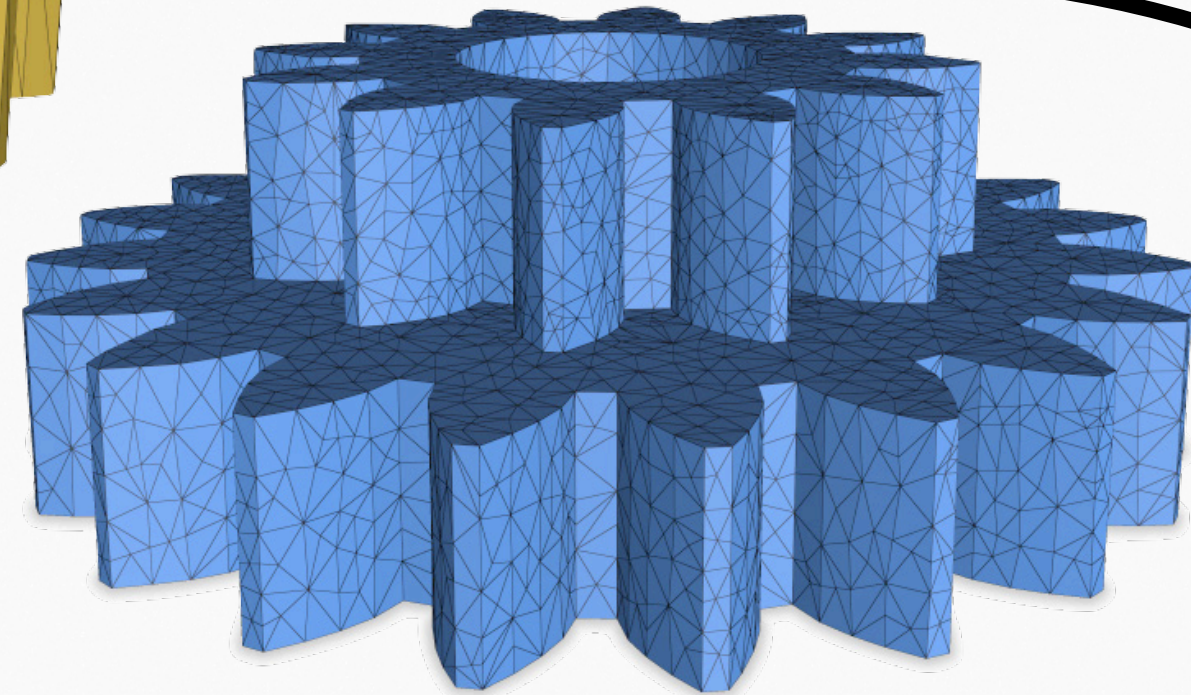




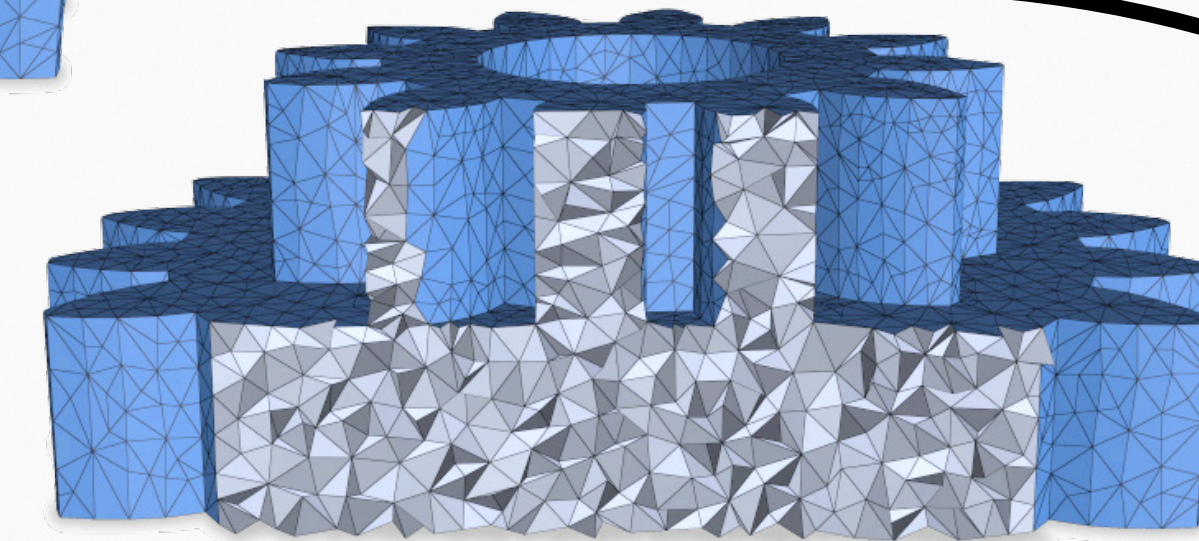
# finite element method (FEM) pipeline



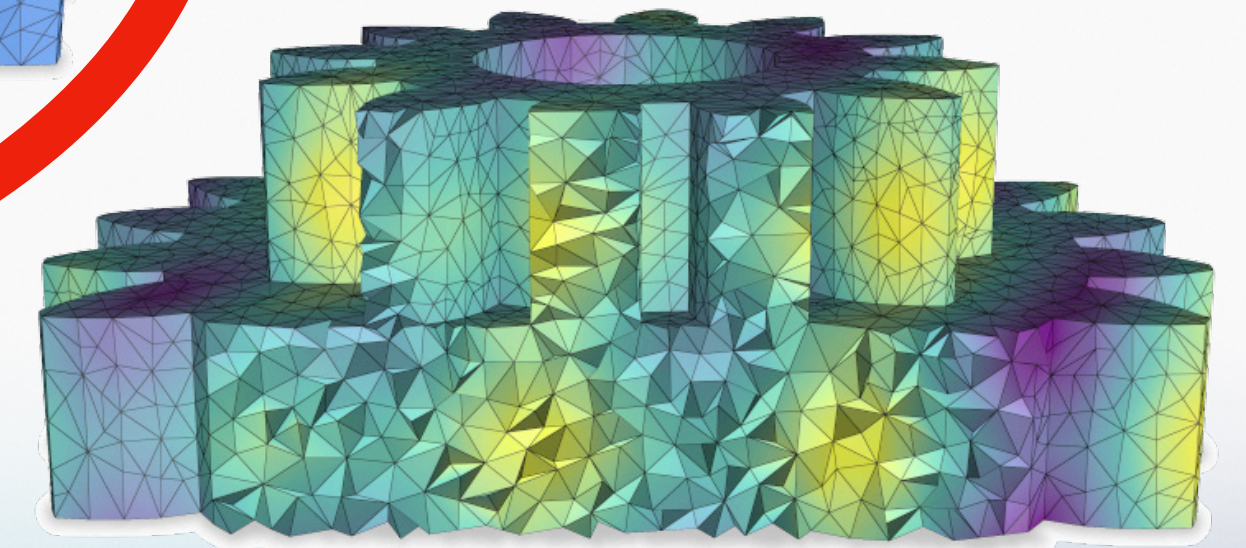
input boundary representation



high-quality surface mesh



volume mesh



PDE solution

**bottleneck**

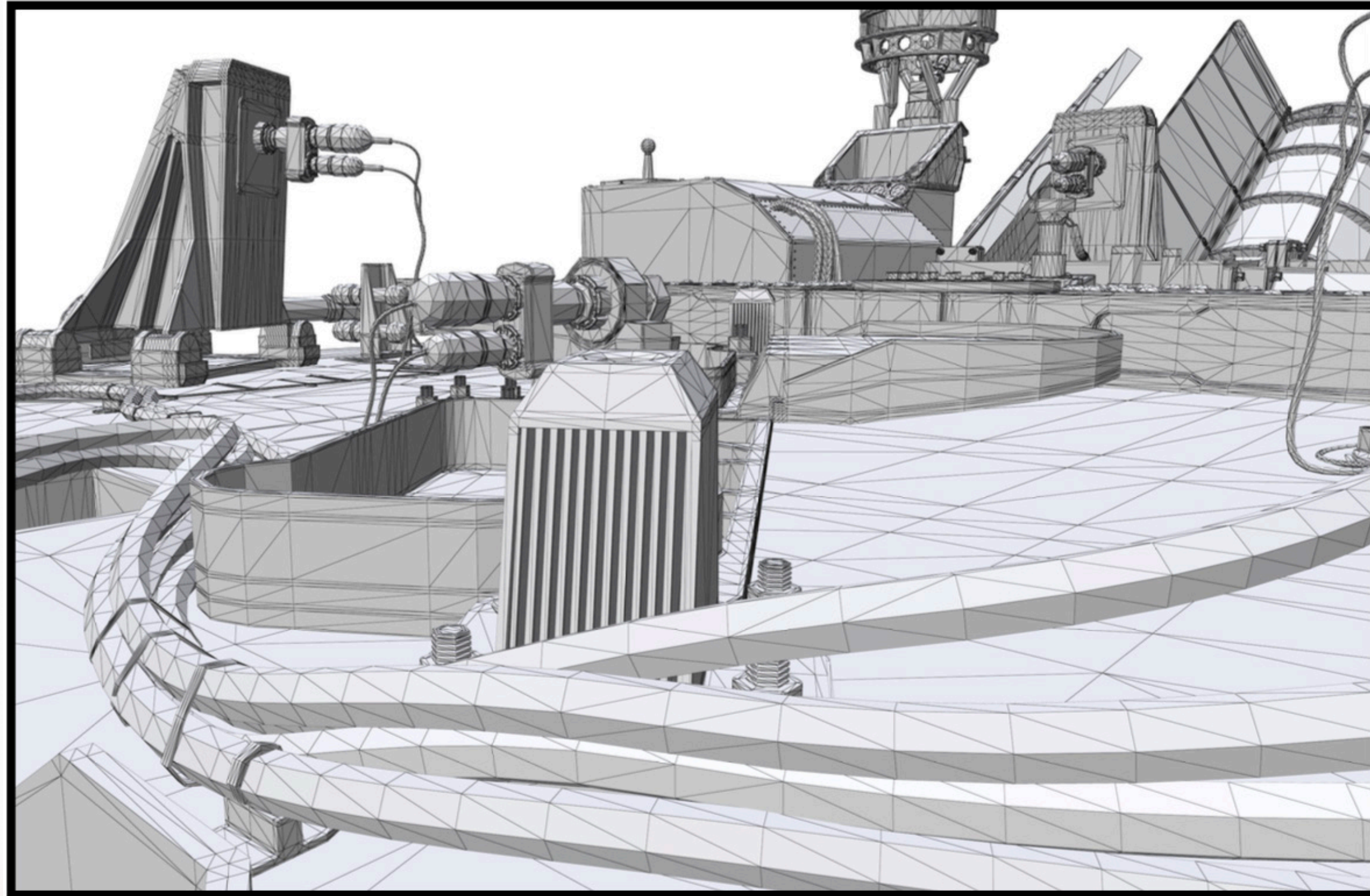
tetrahedralize

FEM solve



# meshing complex geometry is difficult + slow

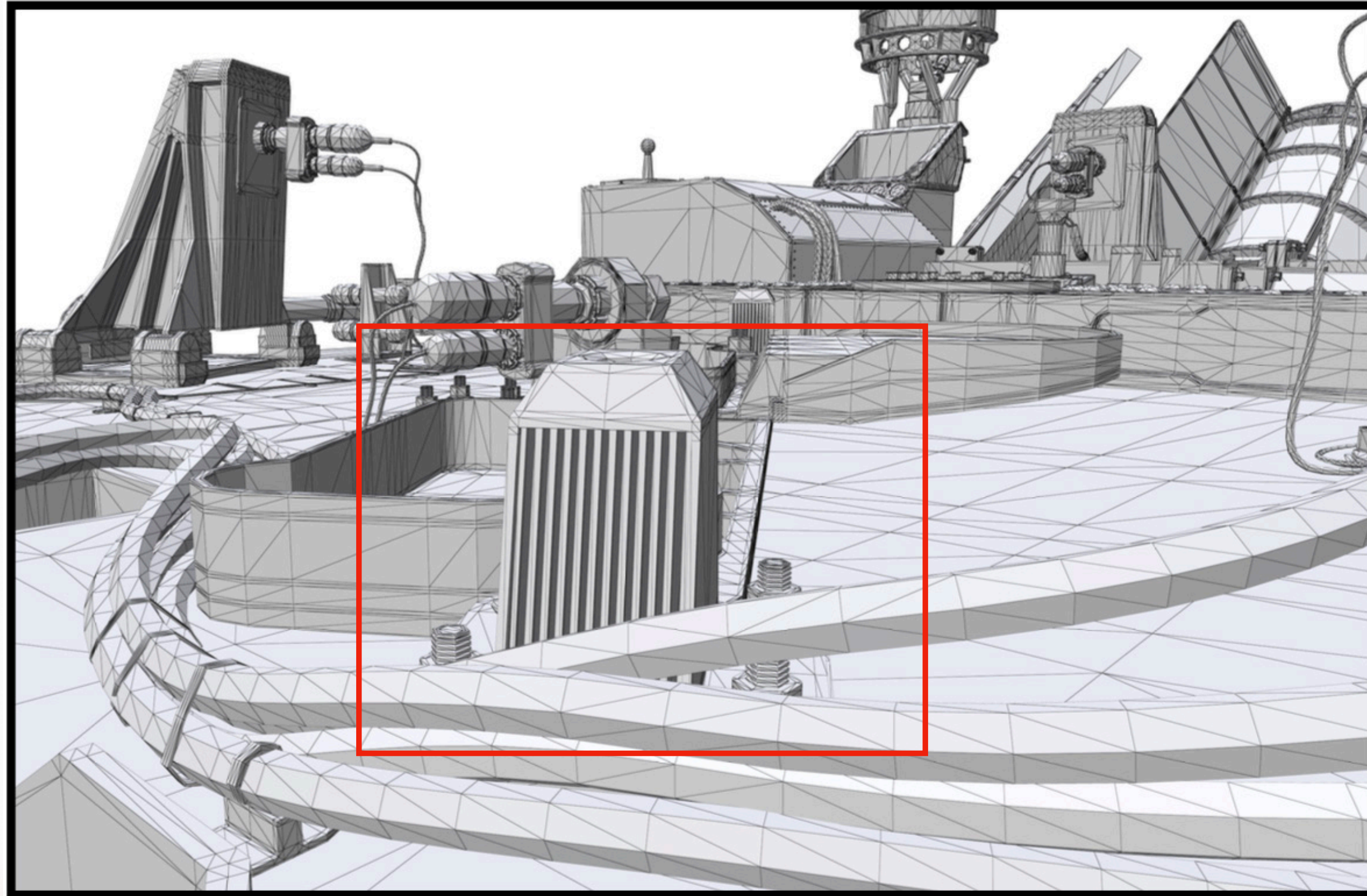
input boundary mesh



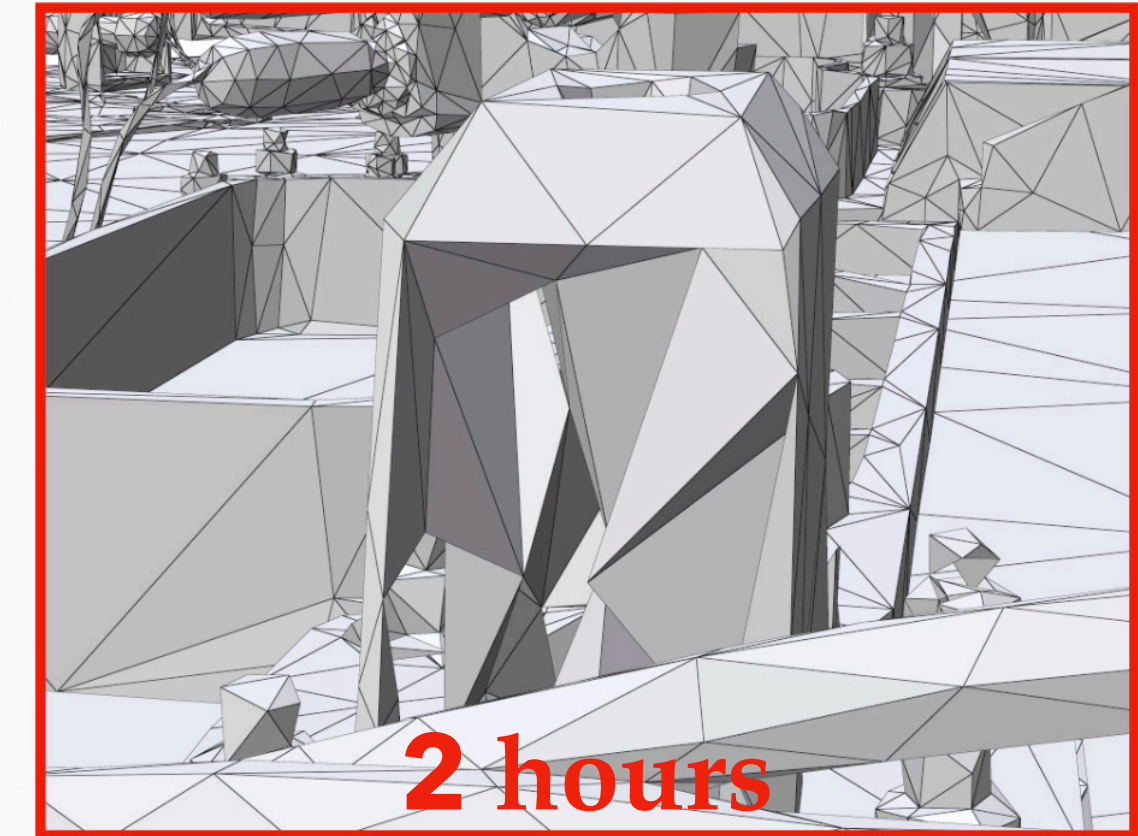
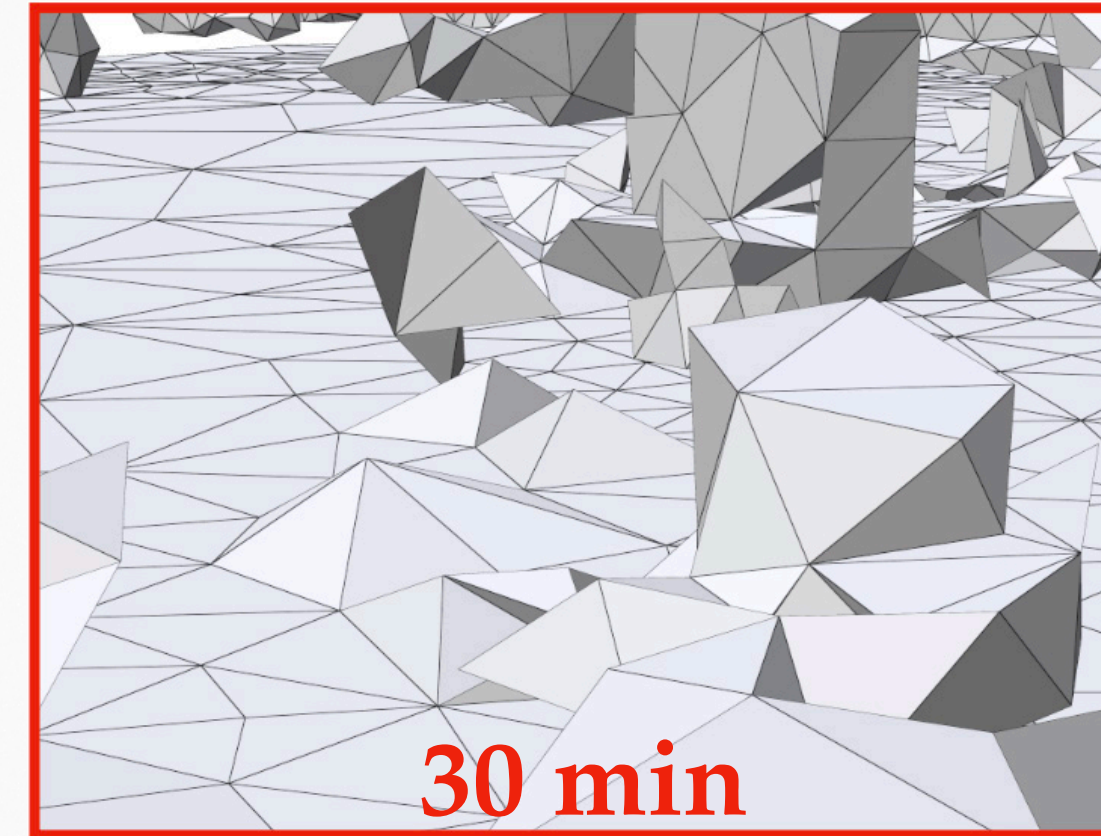


# meshing complex geometry is difficult + slow

input boundary mesh



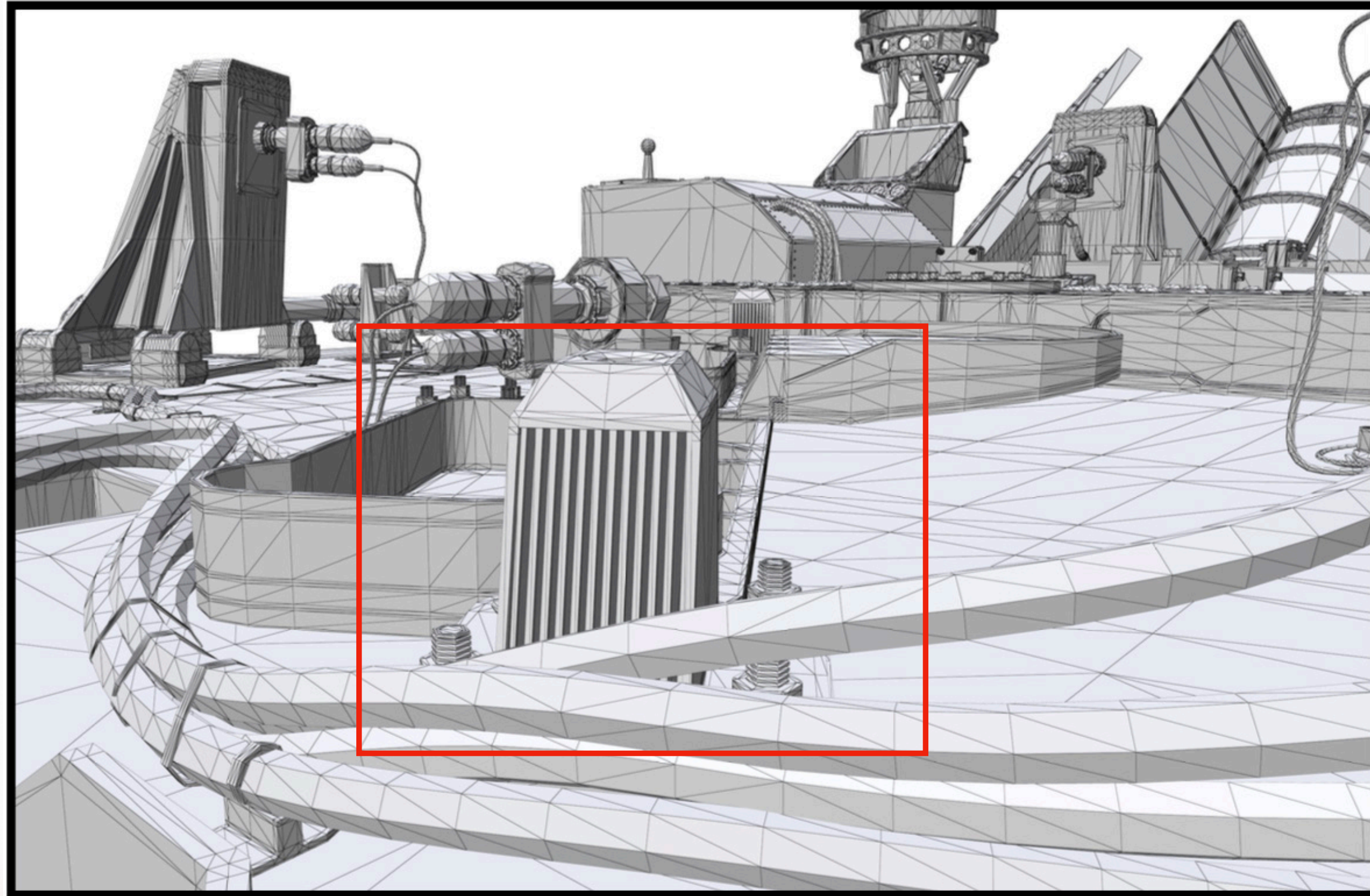
boundary of tetrahedral mesh



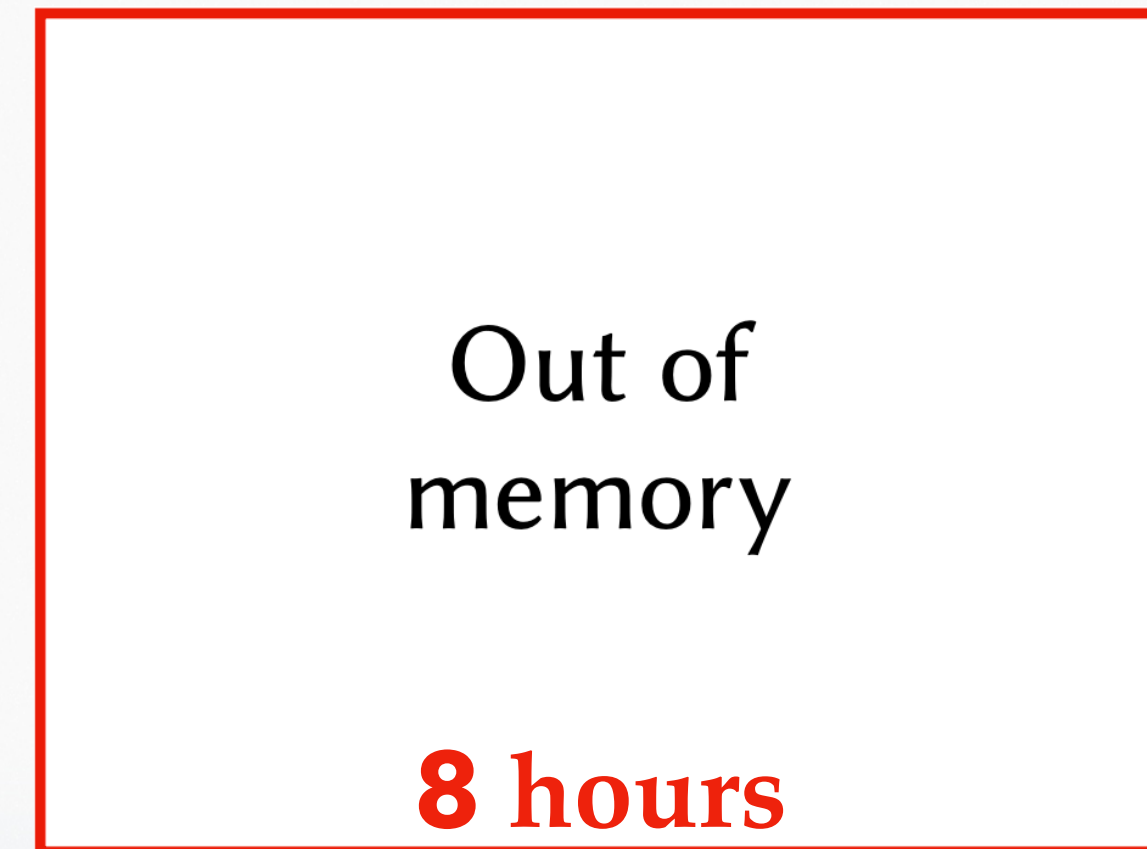
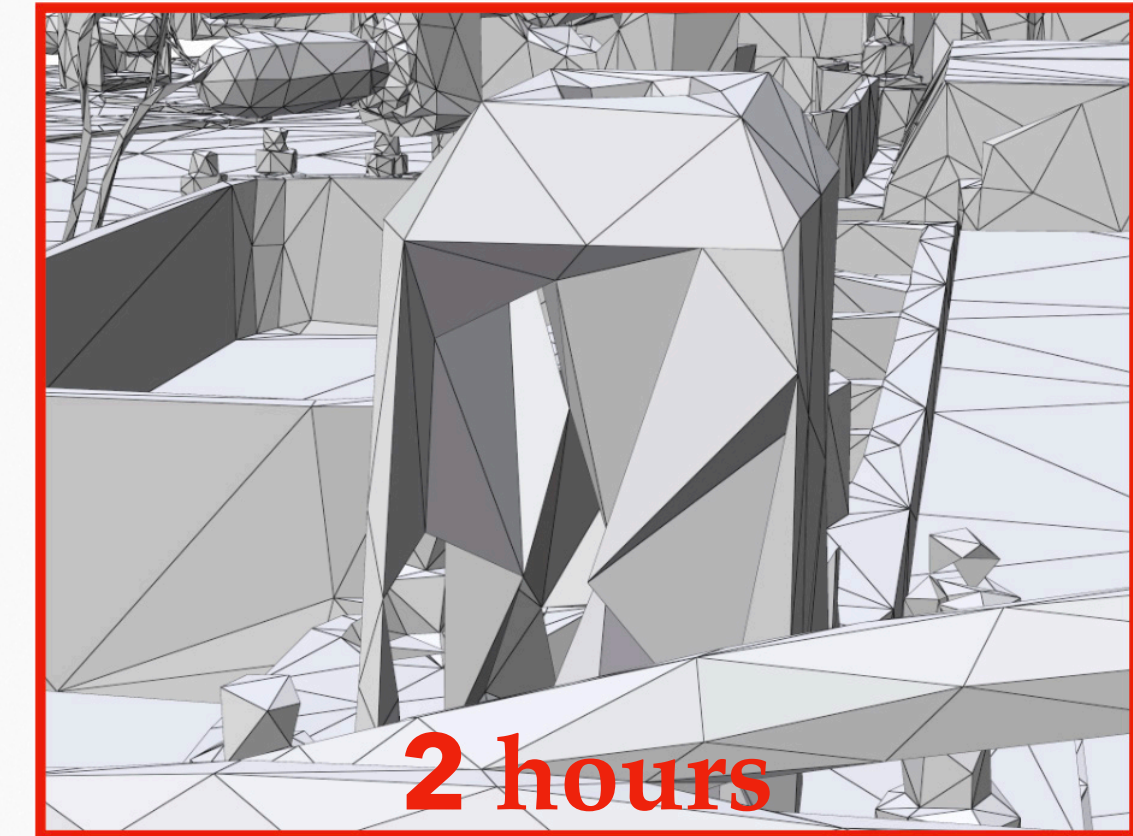
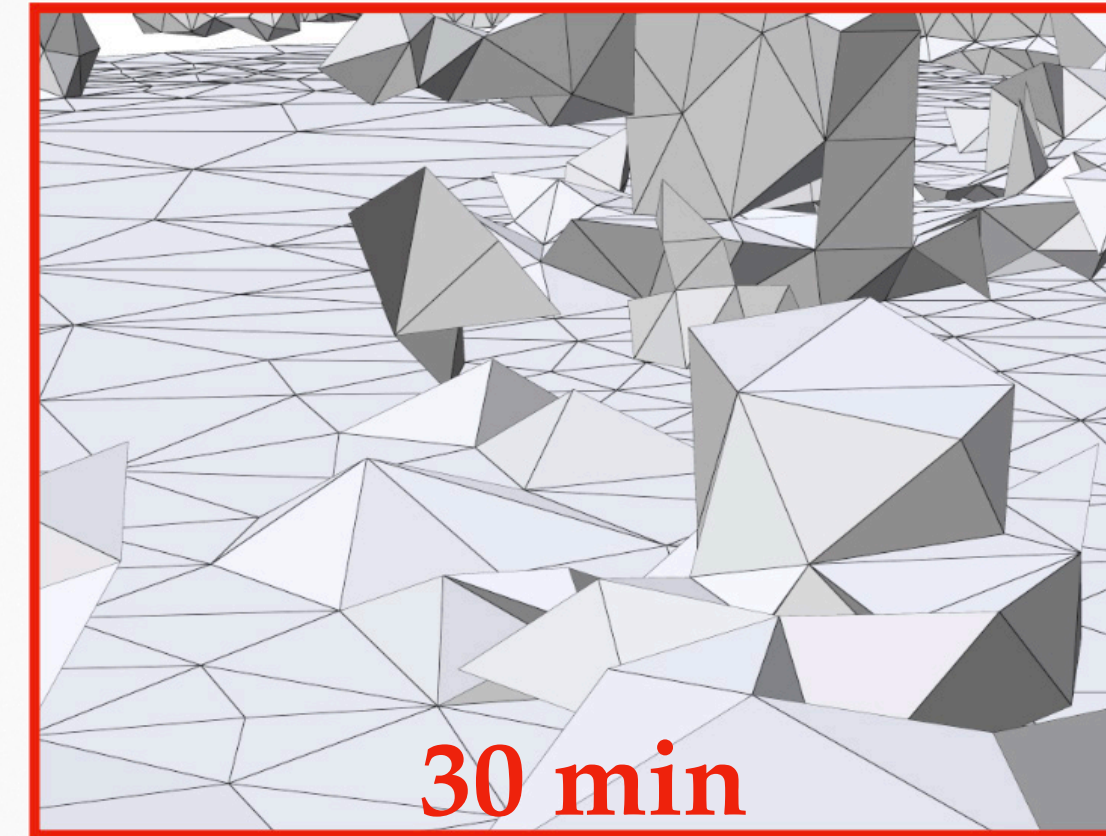


# meshing complex geometry is difficult + slow

input boundary mesh

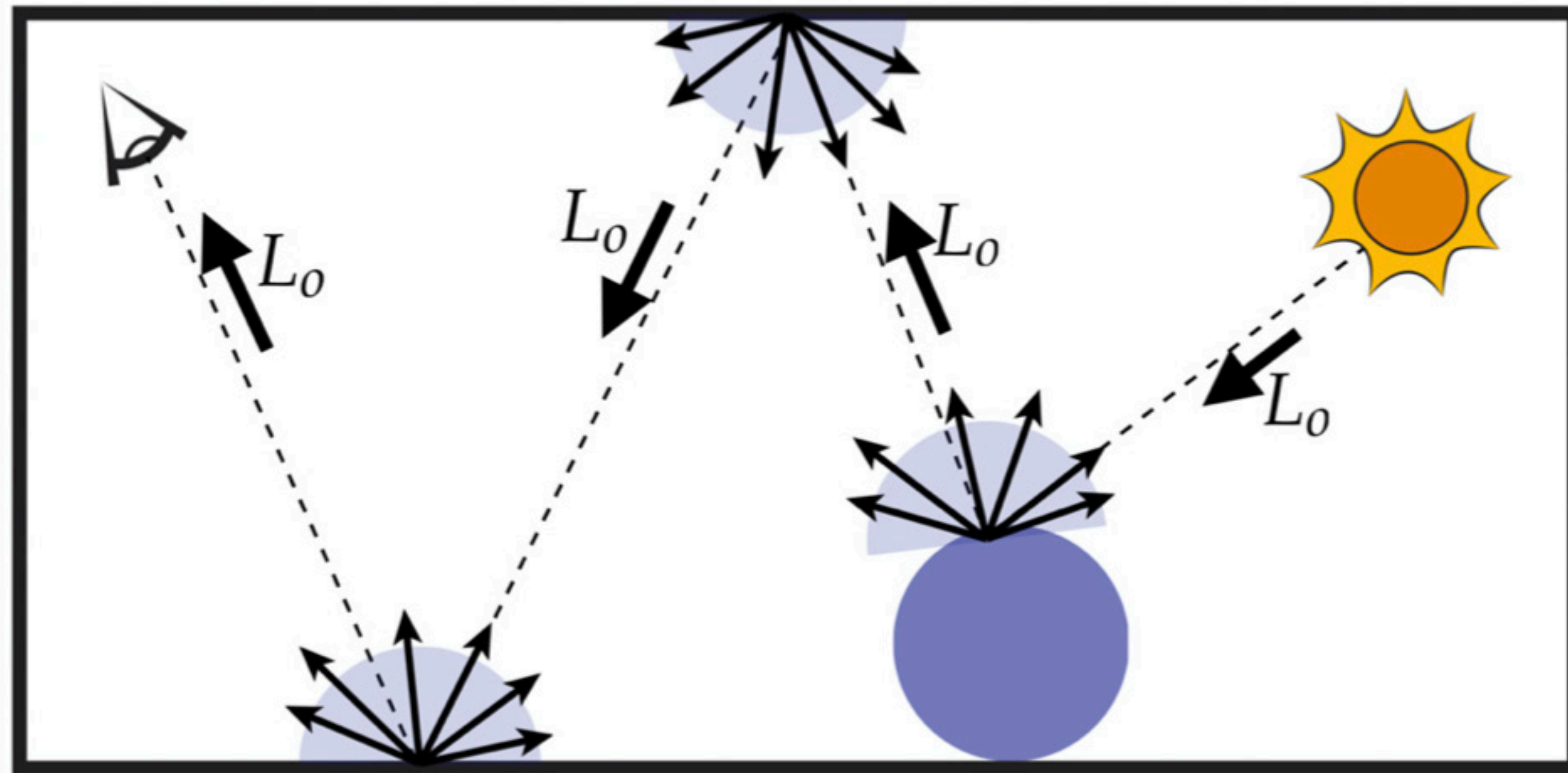


boundary of tetrahedral mesh

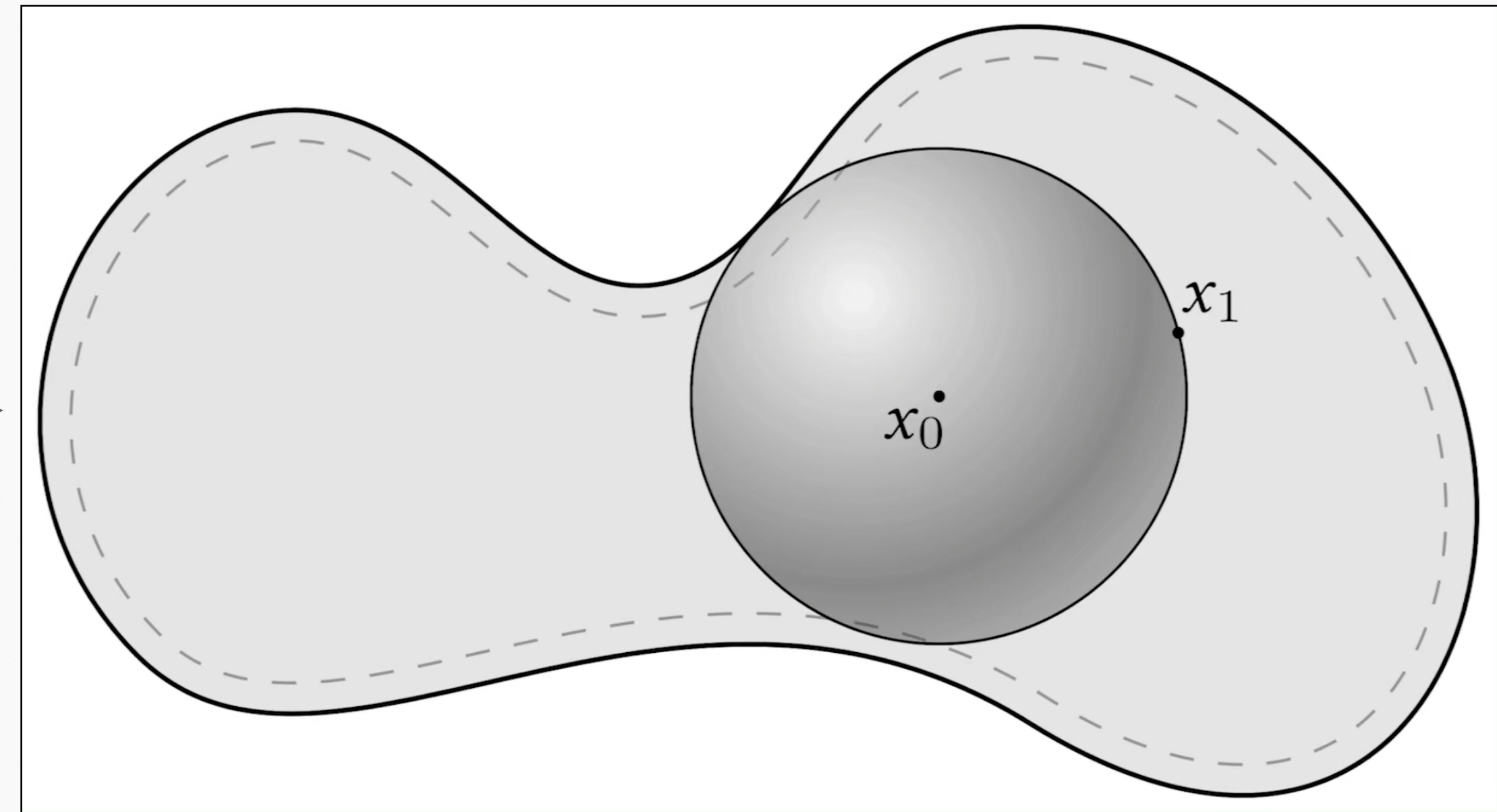
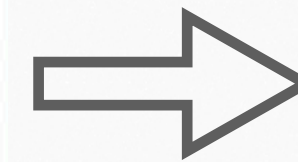




recursive random walks for solving the Laplace equation  $\Delta u = 0$



ray tracing

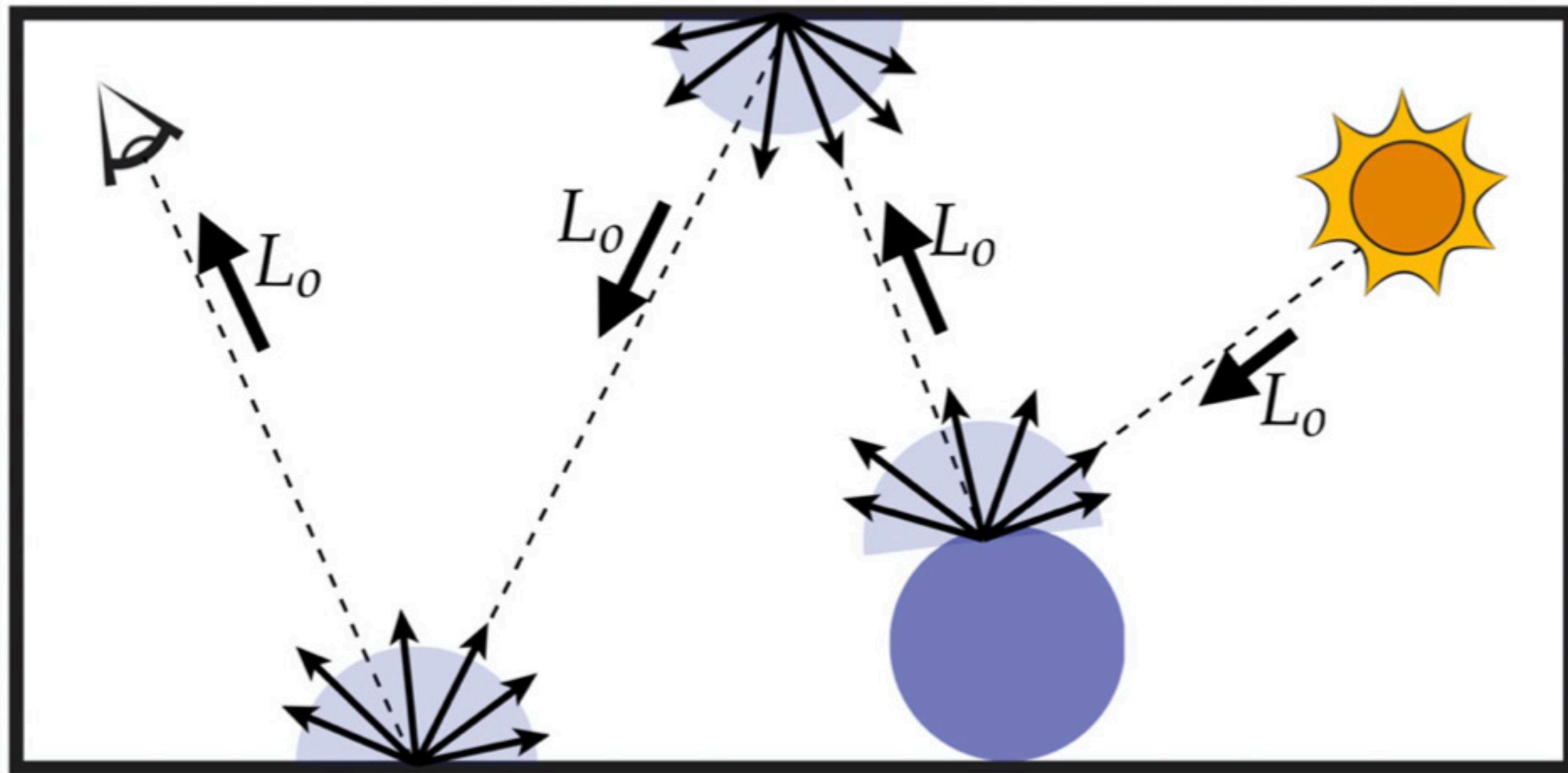


walk on spheres

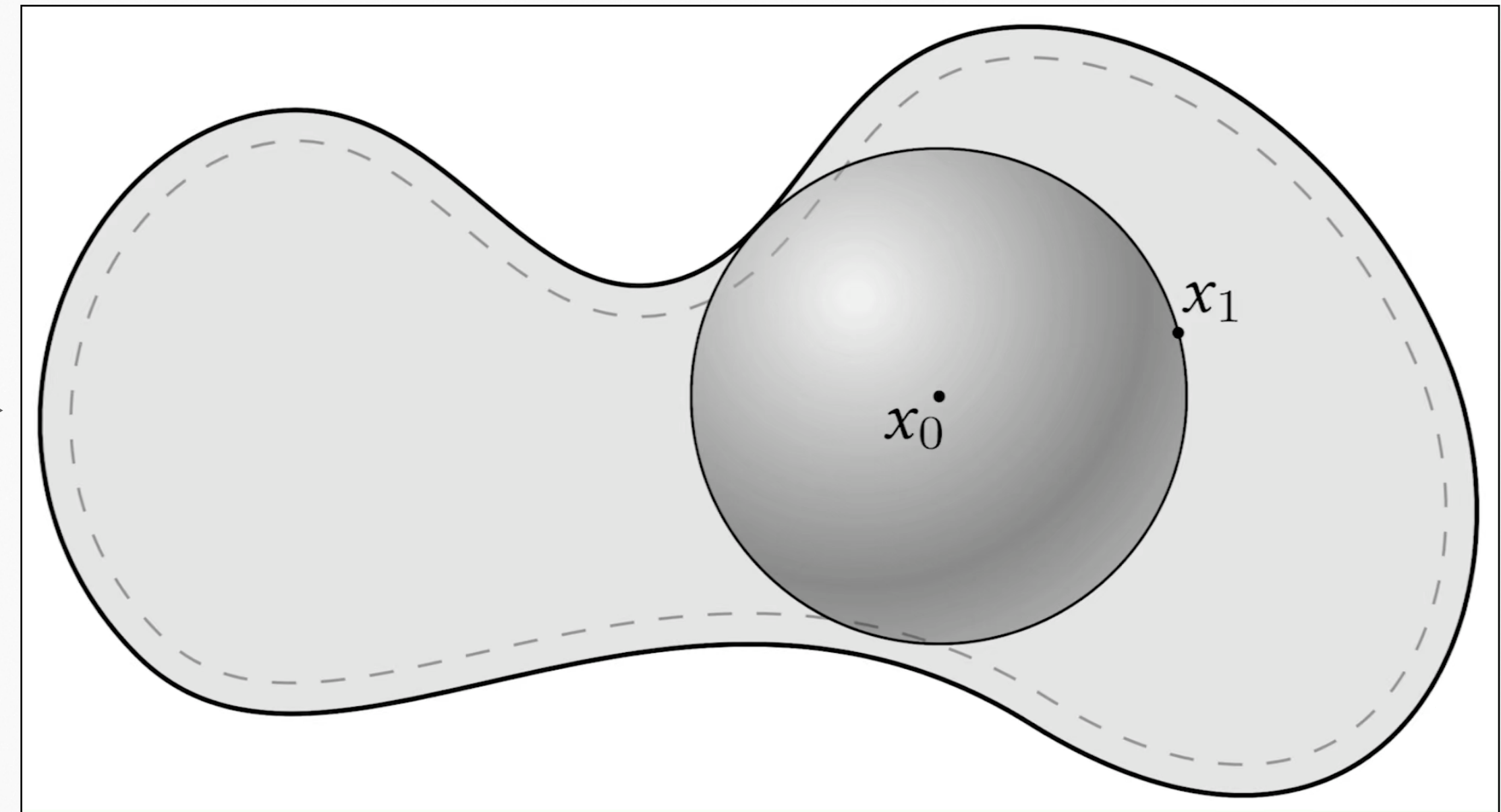
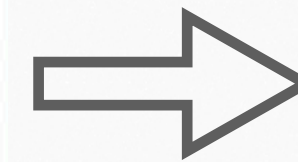
[Muller 1956, Sawhney and Crane 2020]



recursive random walks for solving the Laplace equation  $\Delta u = 0$



ray tracing

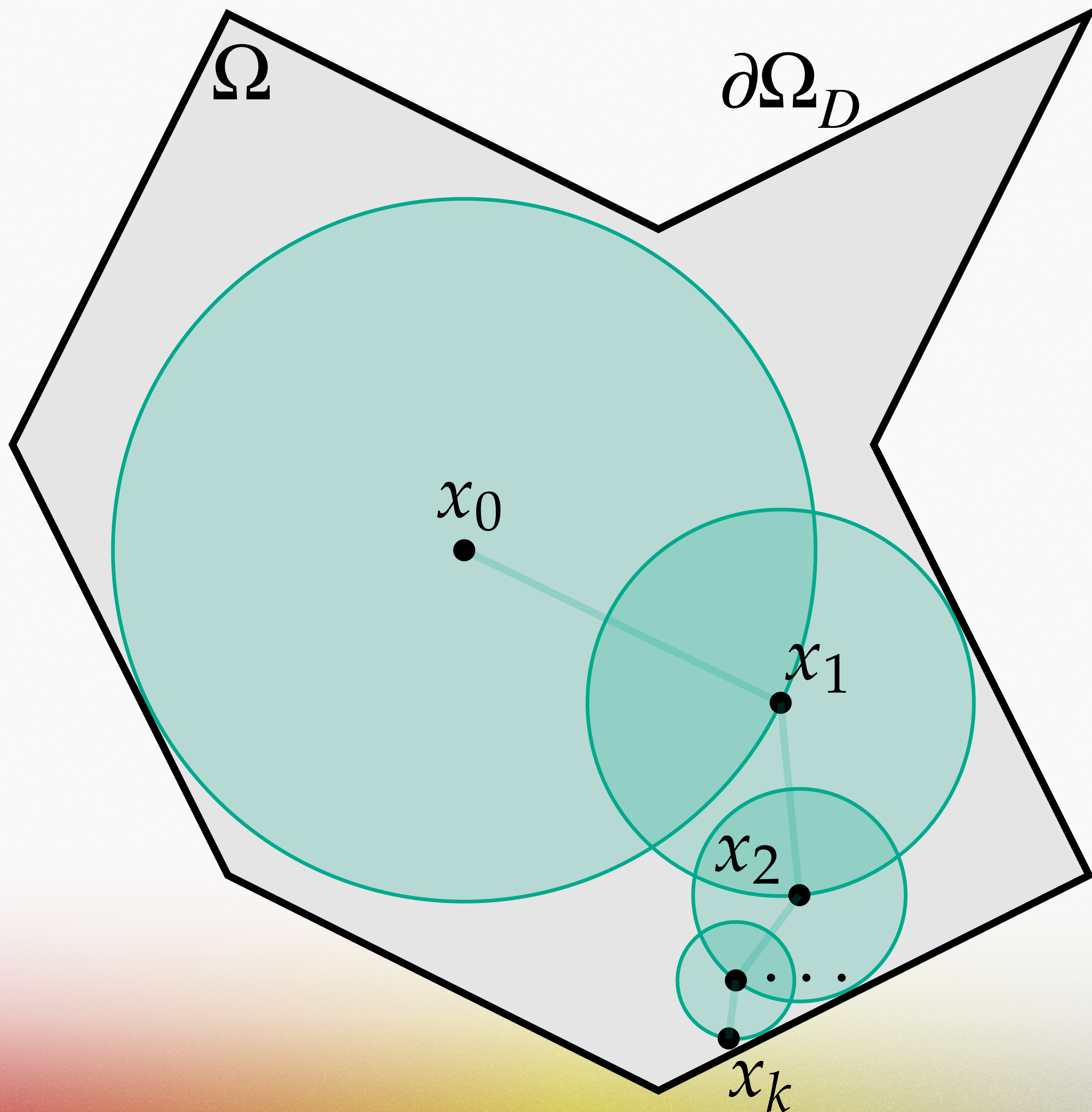


walk on spheres

[Muller 1956, Sawhney and Crane 2020]



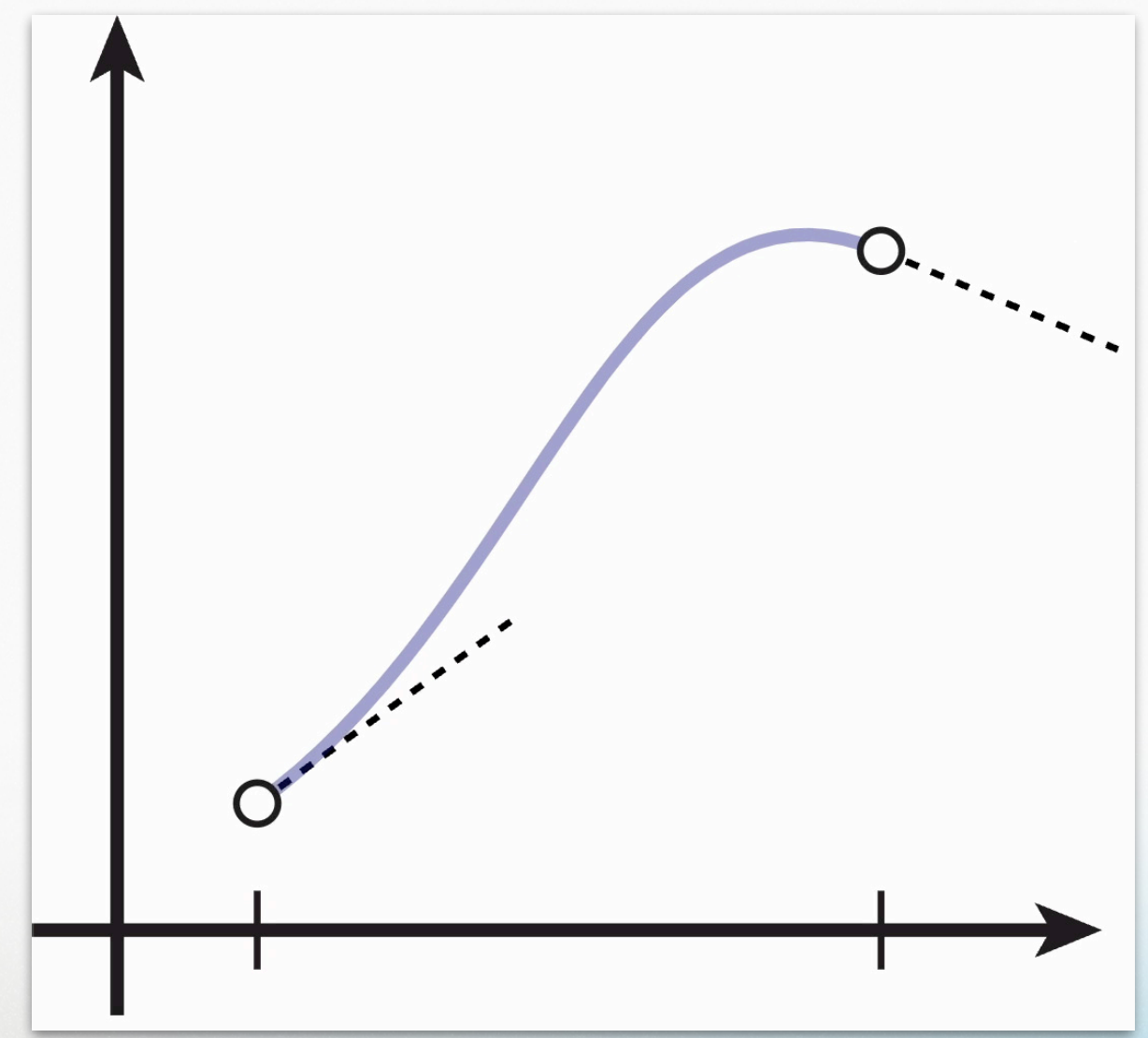
# walk on spheres [Muller 1956, Sawhney and Crane 2020]



$\Delta u = 0$  on  $\Omega$  Laplace eq

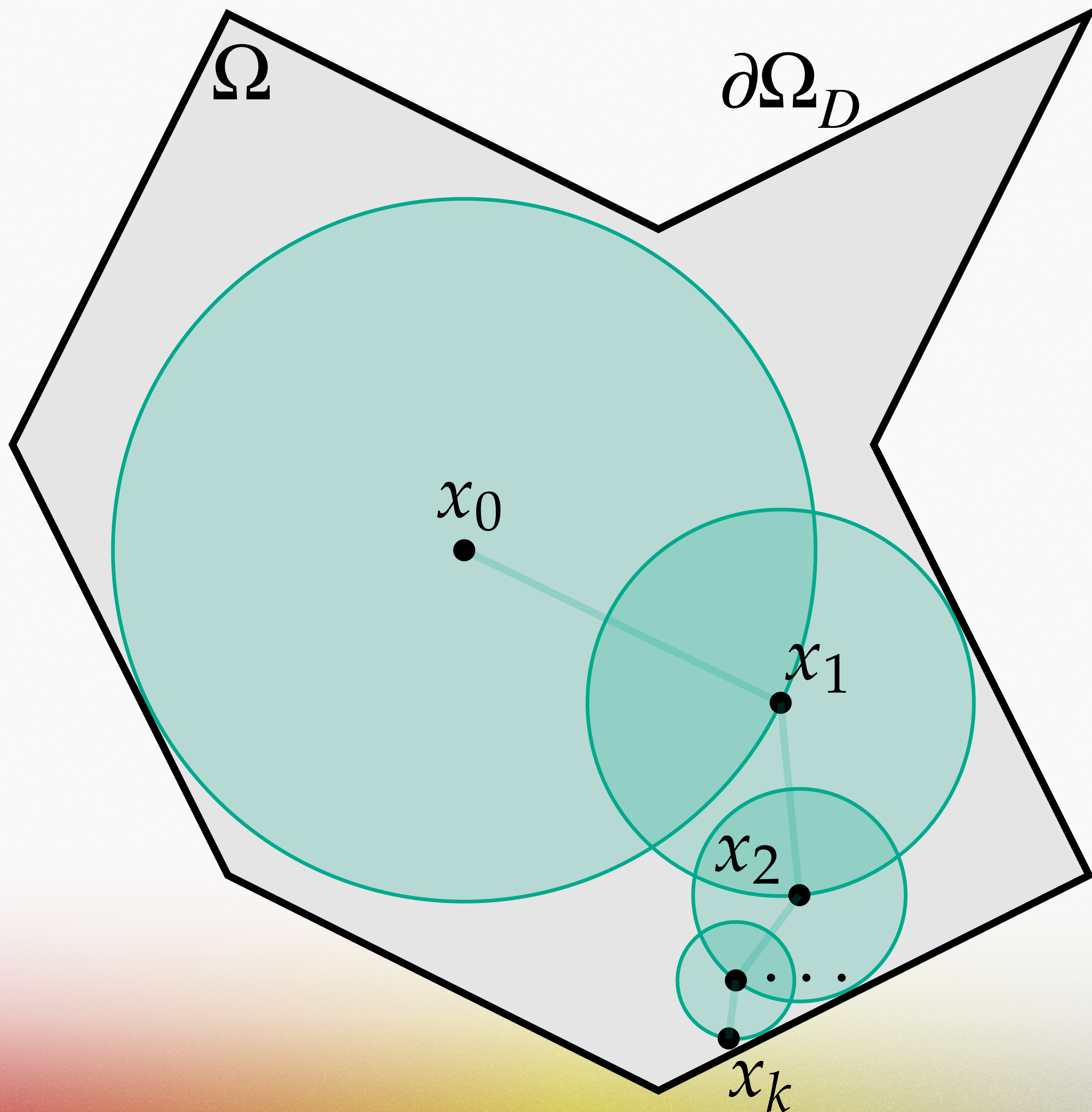
$u = g$  on  $\partial\Omega_D$  Dirichlet —

fixed value  $g$





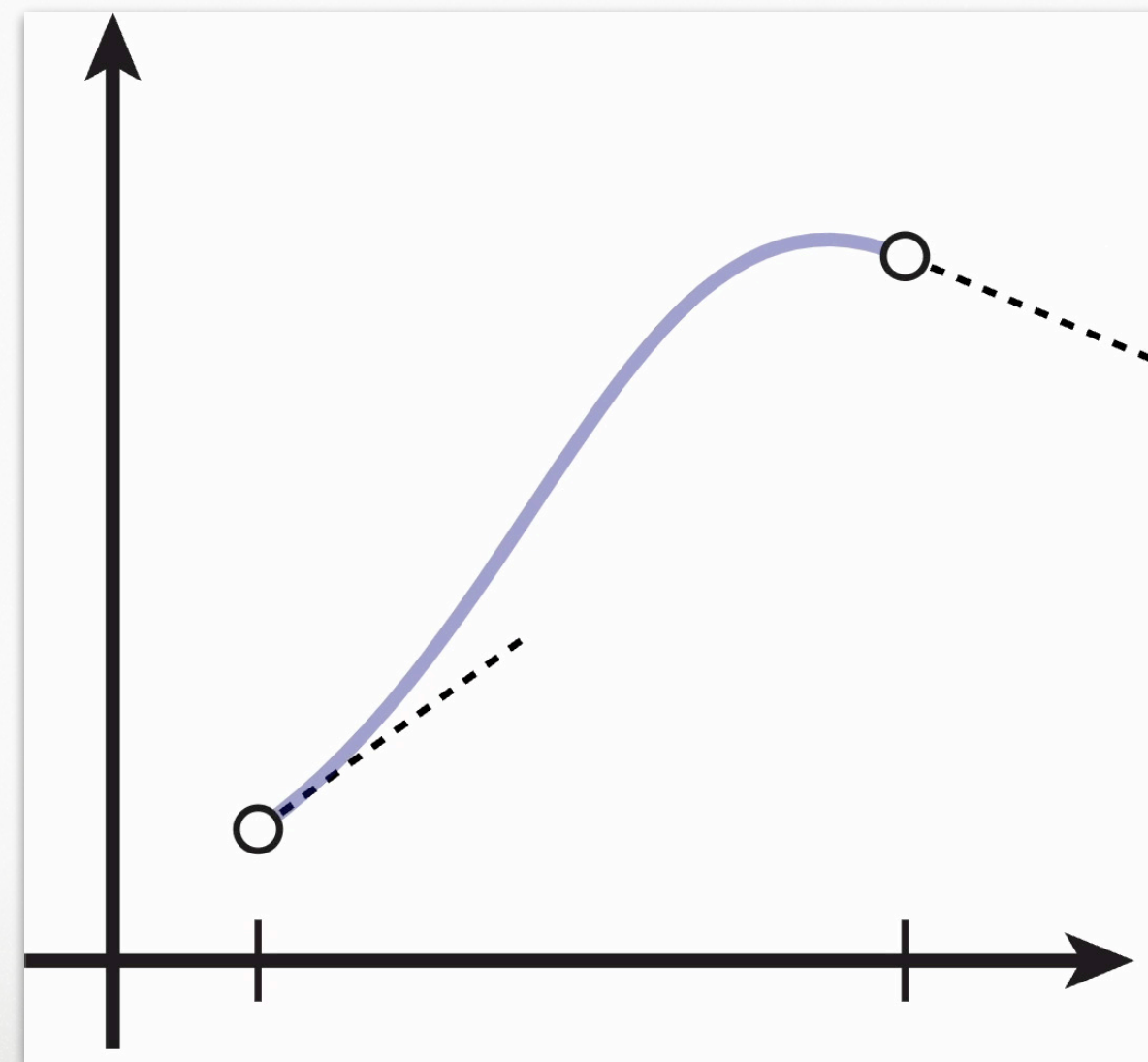
# walk on spheres [Muller 1956, Sawhney and Crane 2020]



$\Delta u = 0$  on  $\Omega$  Laplace eq

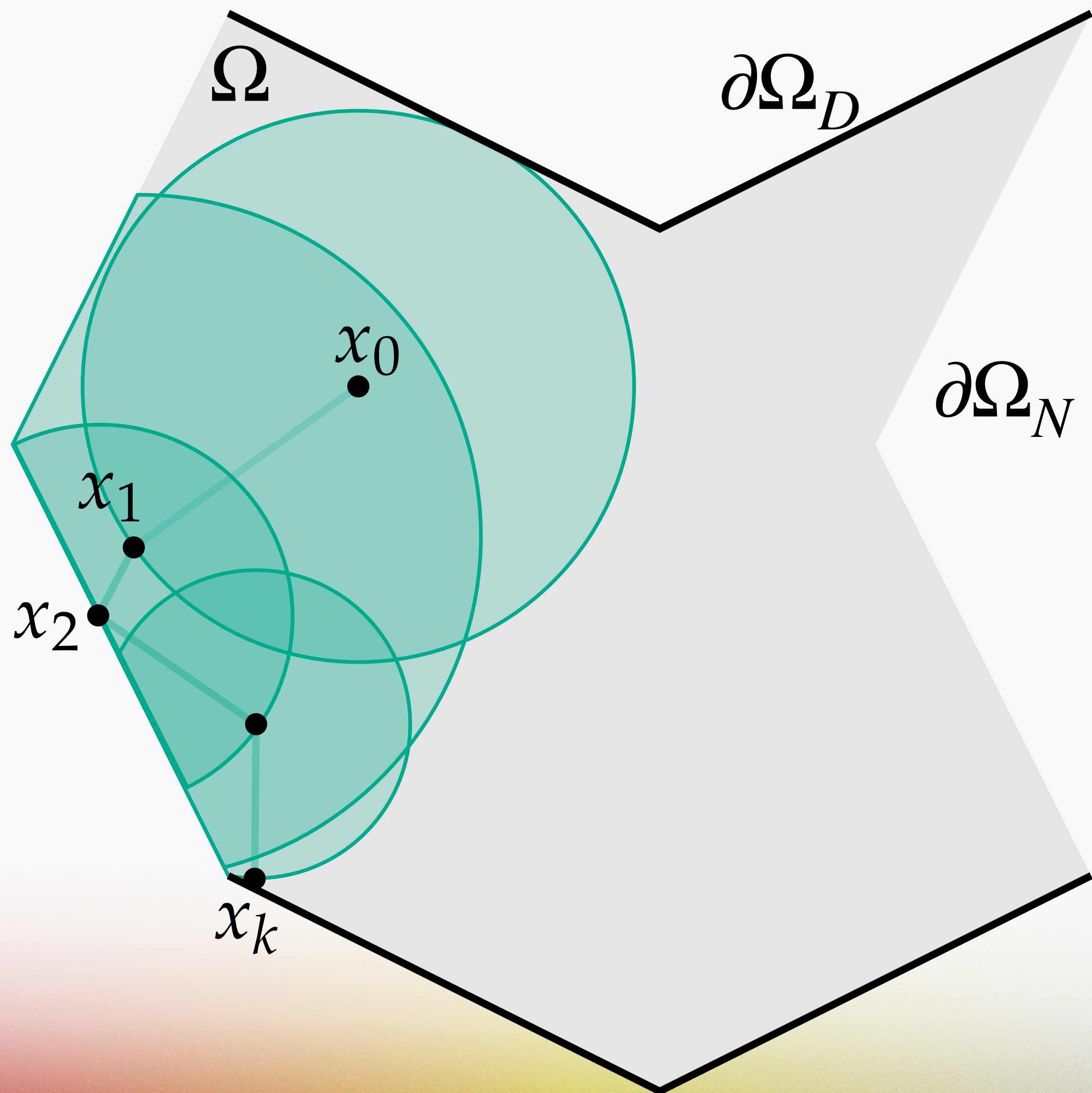
$u = g$  on  $\partial\Omega_D$  Dirichlet —

fixed value  $g$





# walk on stars [Sawhney et al. 2023]



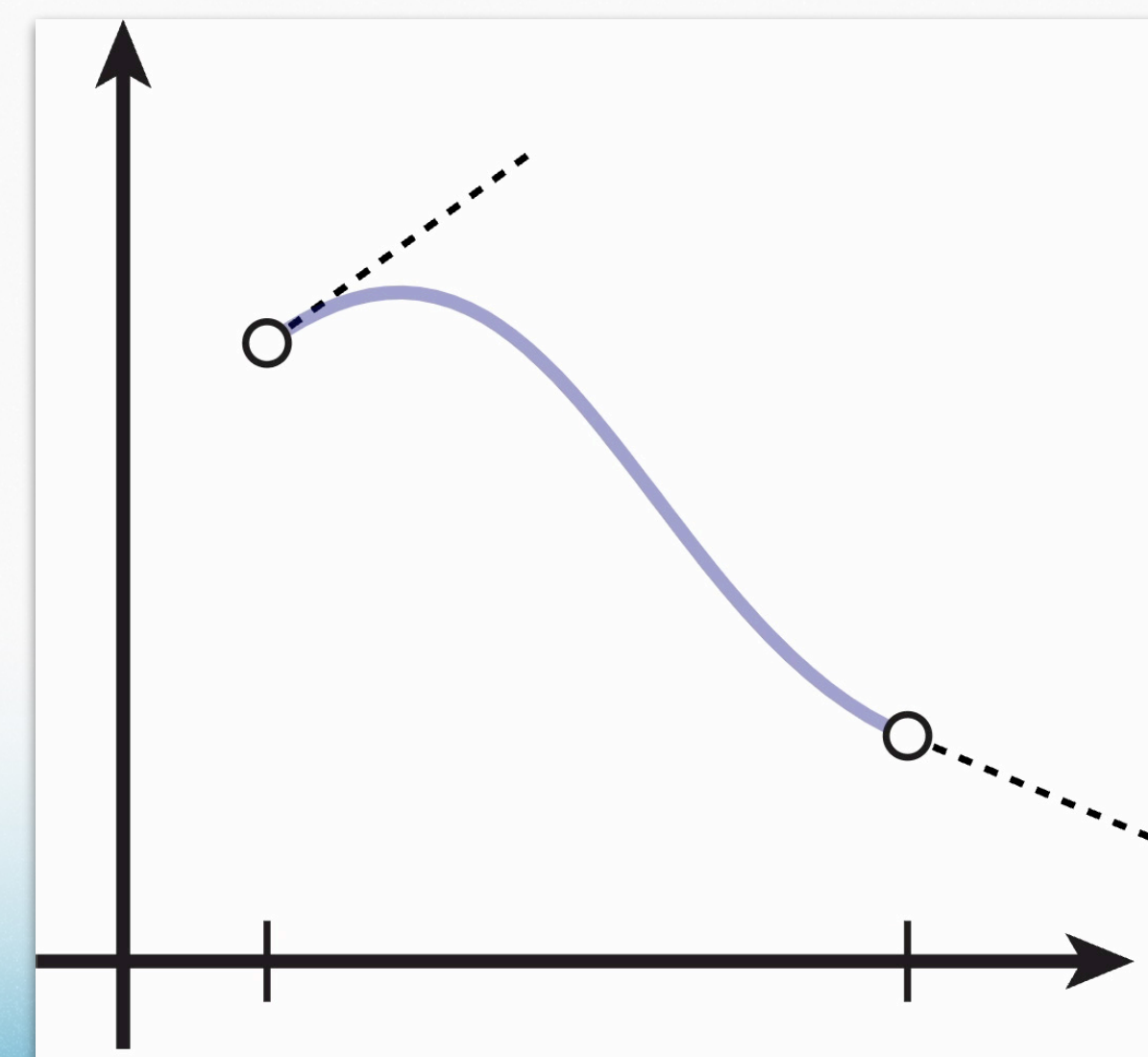
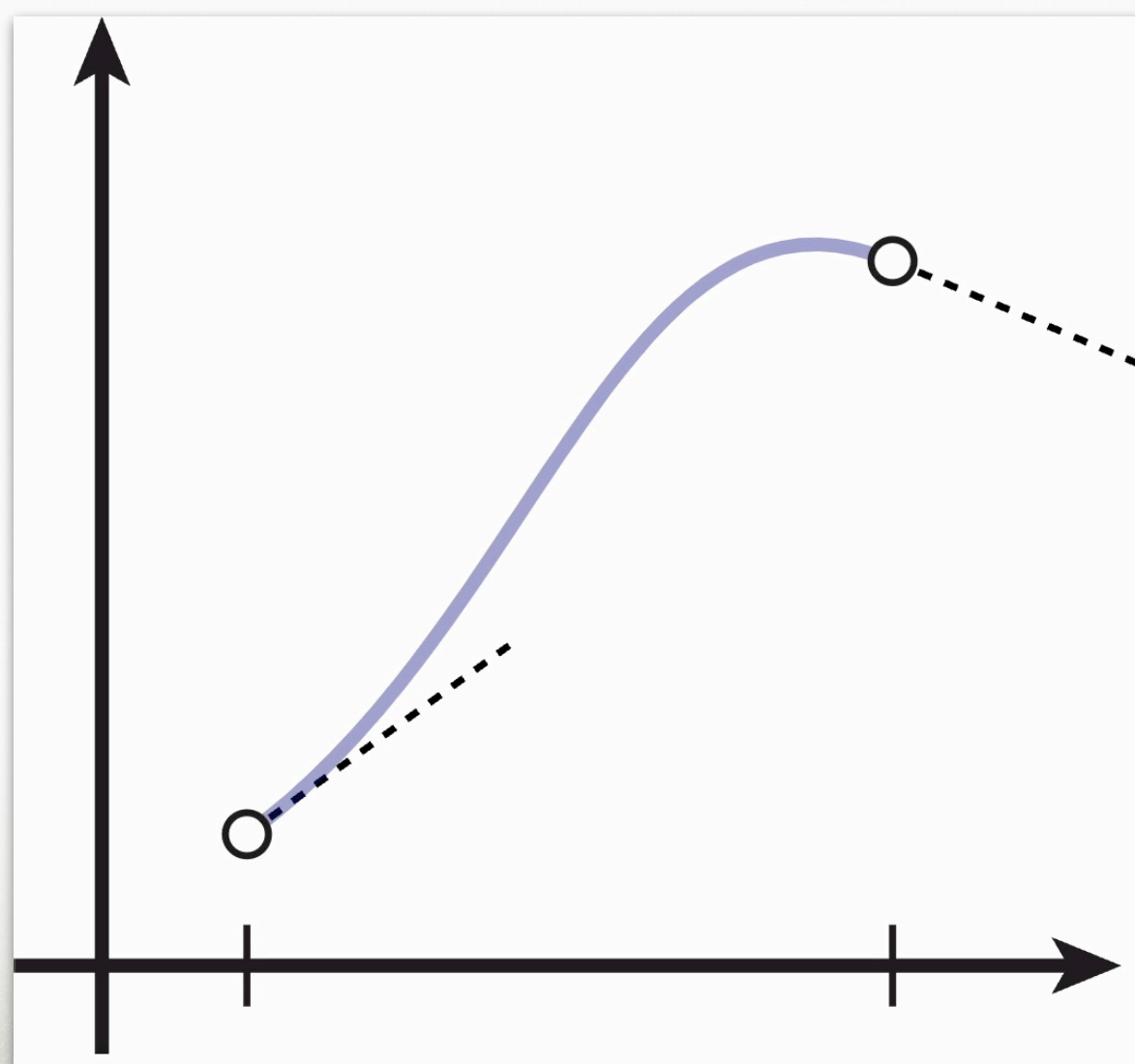
$\Delta u = 0$  on  $\Omega$  Laplace eq

$u = g$  on  $\partial\Omega_D$  Dirichlet  $\text{—}$

$\frac{\partial u}{\partial n} = h$  on  $\partial\Omega_N$  Neumann  $\text{=}$

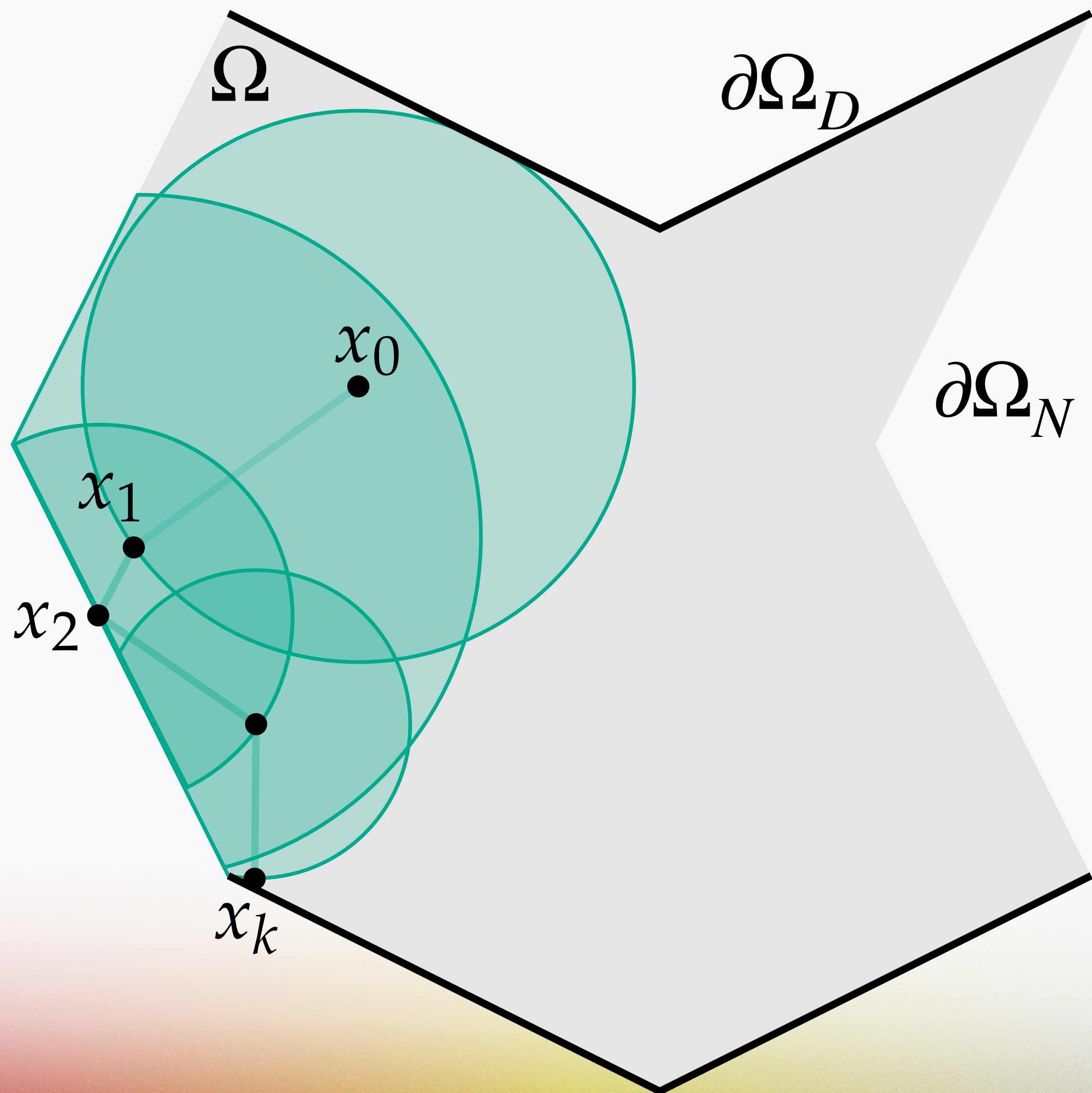
fixed value  $g$

fixed derivative  $h$





# walk on stars [Sawhney et al. 2023]

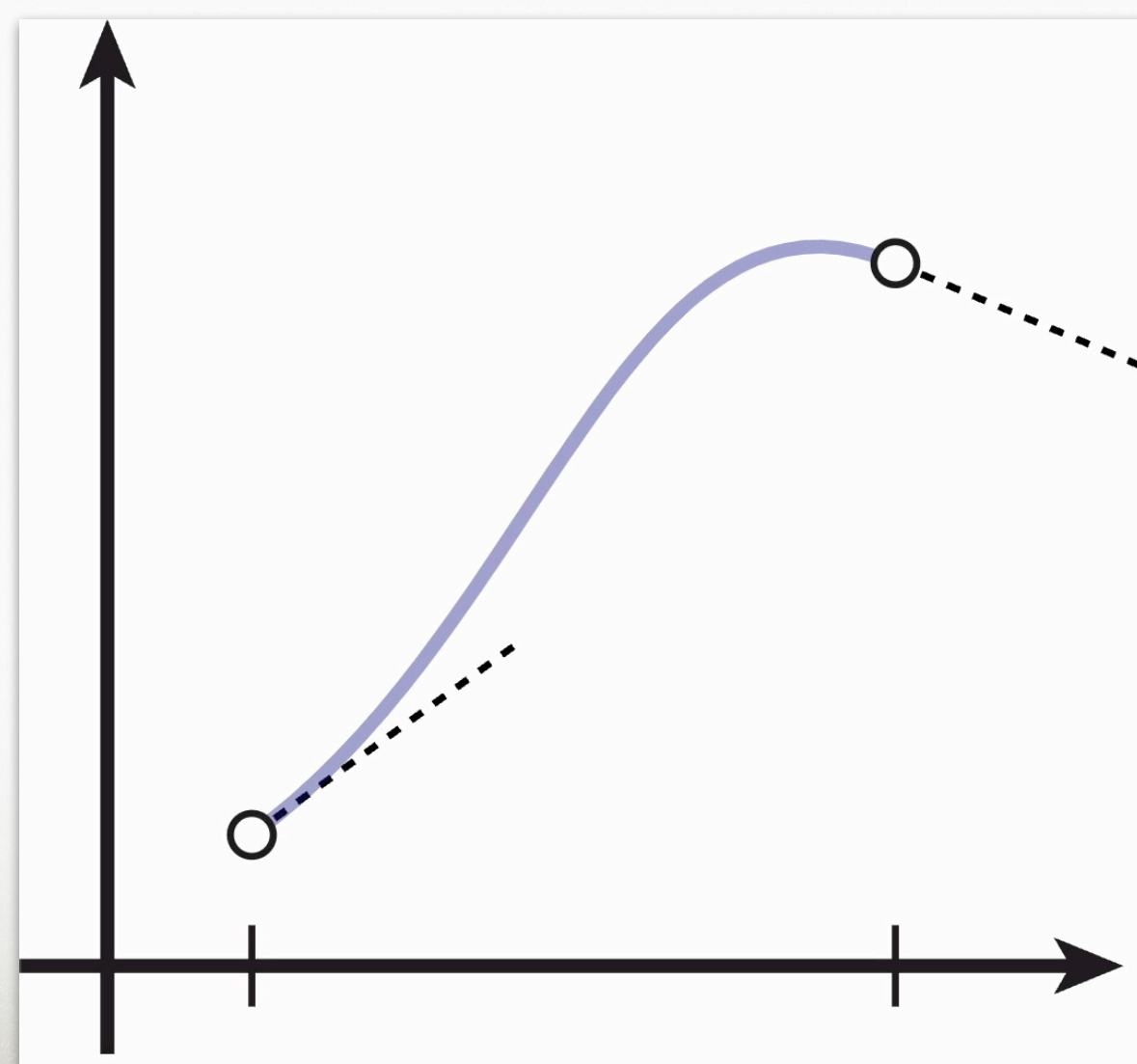


$\Delta u = 0$  on  $\Omega$  Laplace eq

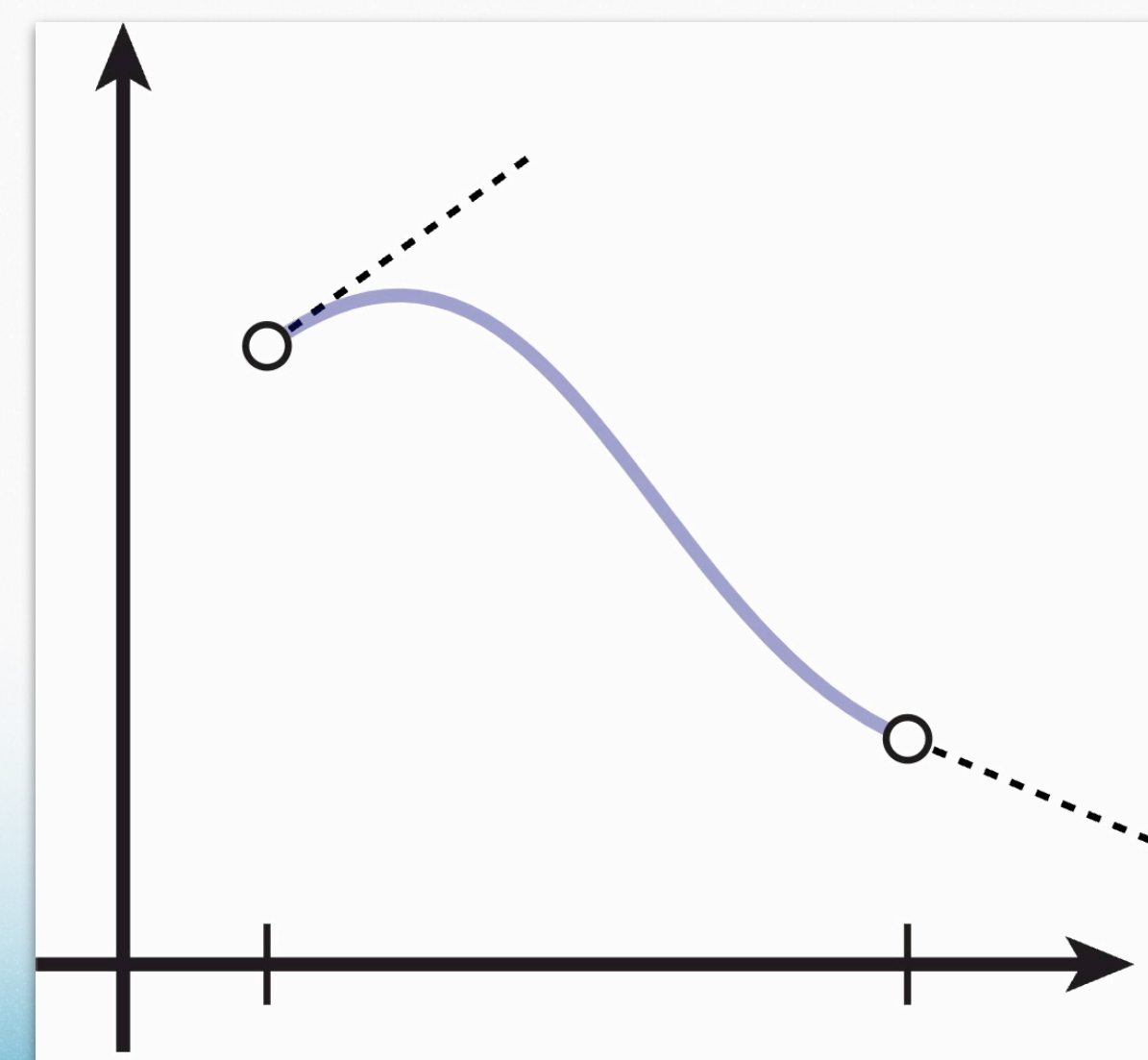
$u = g$  on  $\partial\Omega_D$  Dirichlet  $\text{—}$

$\frac{\partial u}{\partial n} = h$  on  $\partial\Omega_N$  Neumann  $\text{=}$

fixed value  $g$

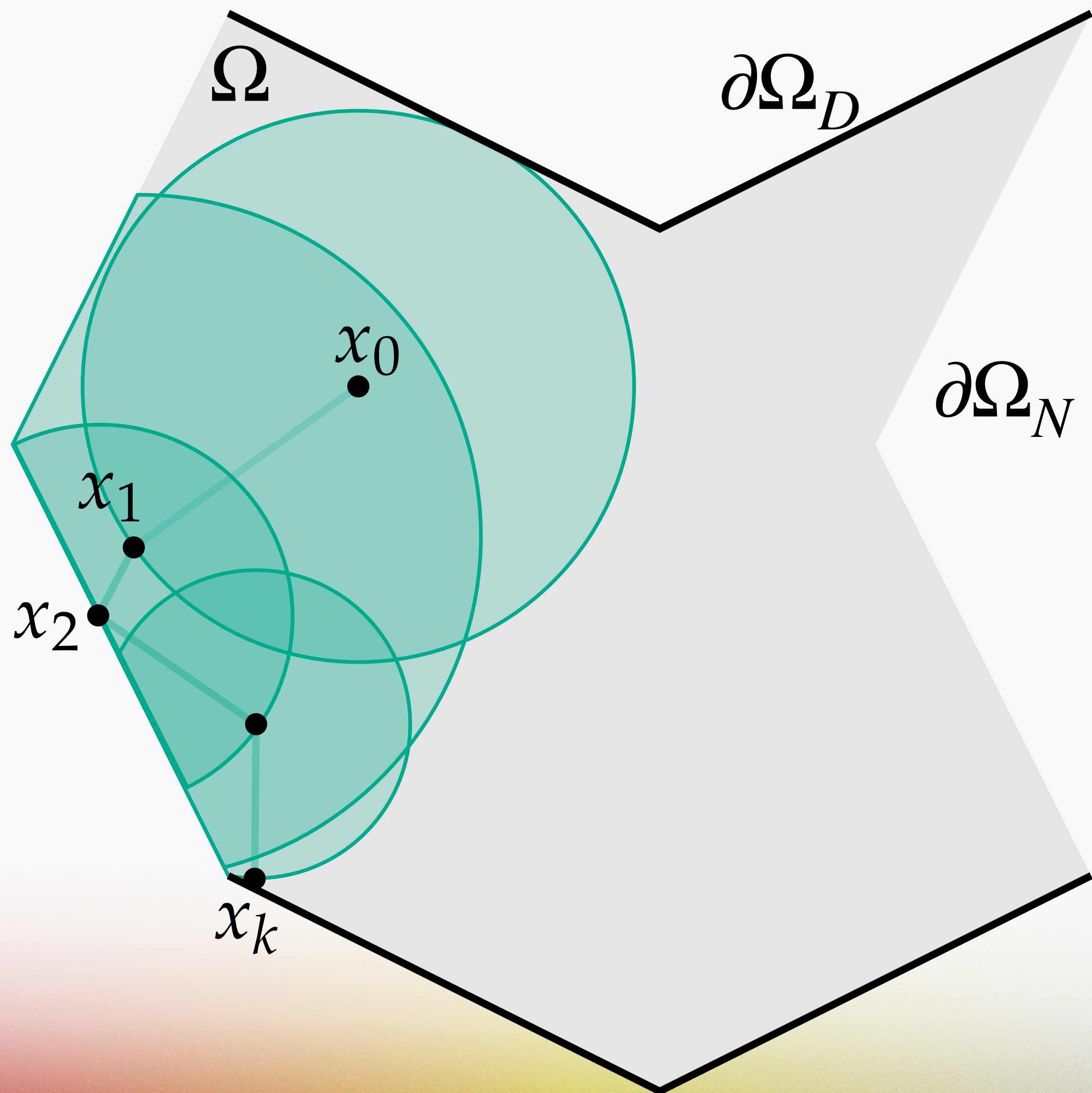


fixed derivative  $h$





# walk on stars [Sawhney et al. 2023]



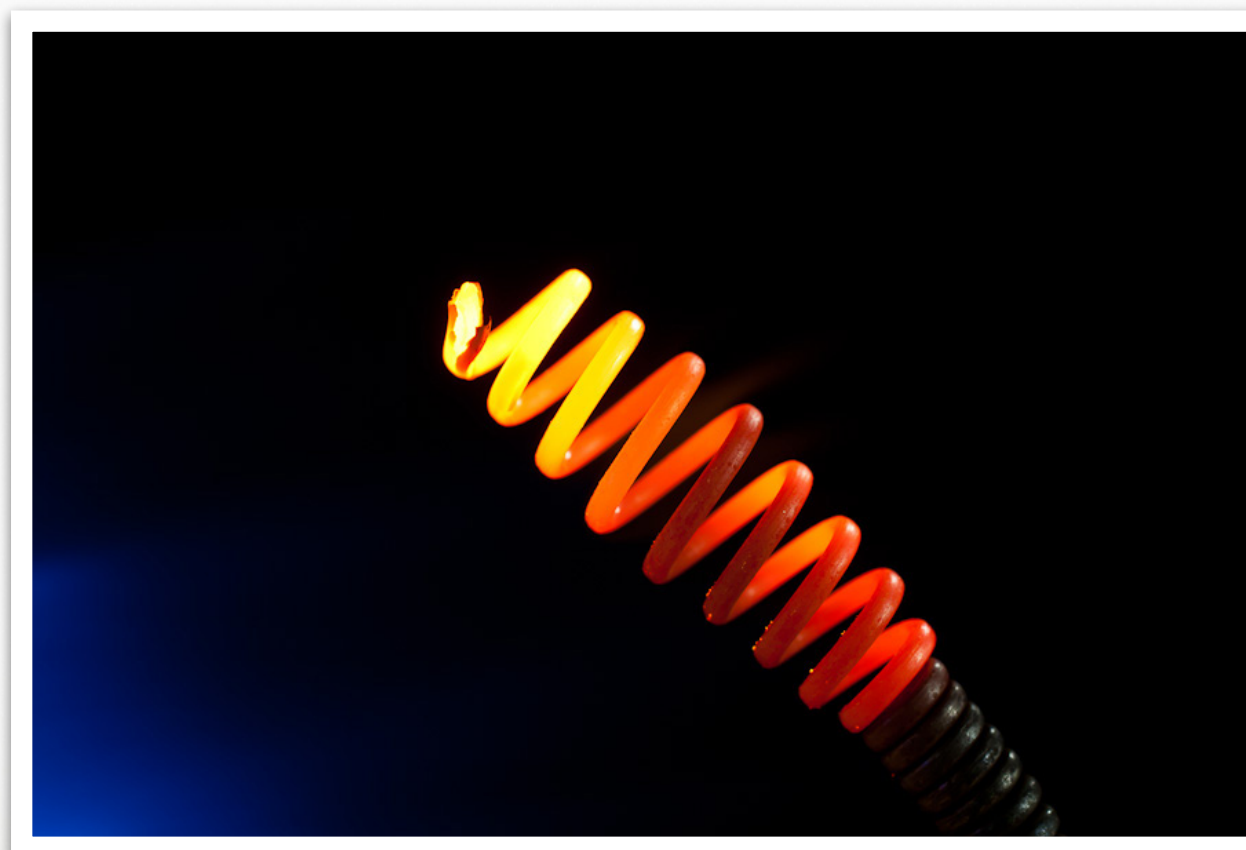
$$\Delta u = 0 \quad \text{on } \Omega \quad \text{Laplace eq}$$

$$u = g \quad \text{on } \partial\Omega_D \quad \text{Dirichlet } \text{—}$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N \quad \text{Neumann } \text{=}$$

fixed value  $g$

fixed **derivative**  $h$





# importance of materials in rendering

why are idealized materials not enough?

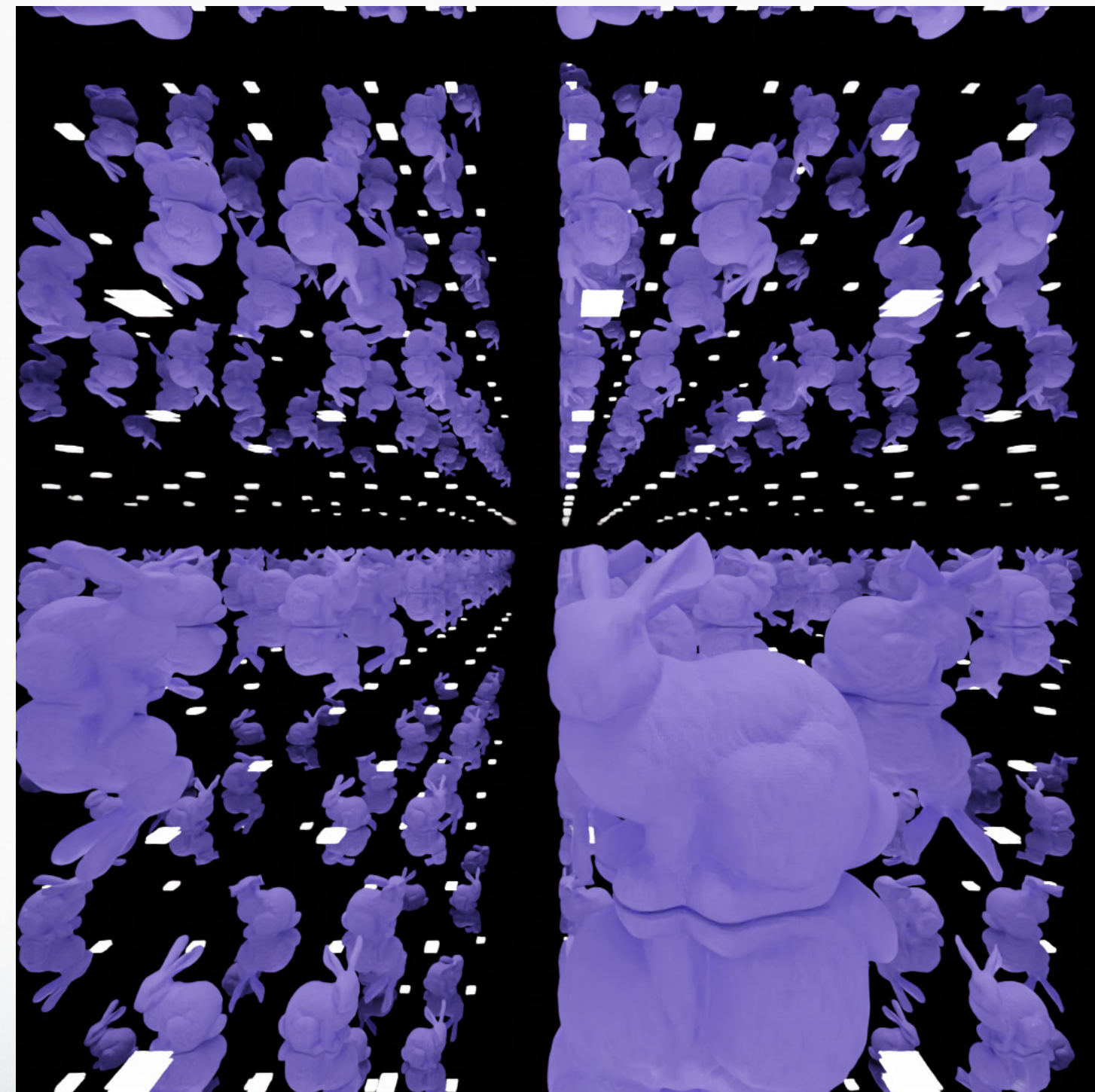
“Robin”

*partially* reflective walls



“Neumann”

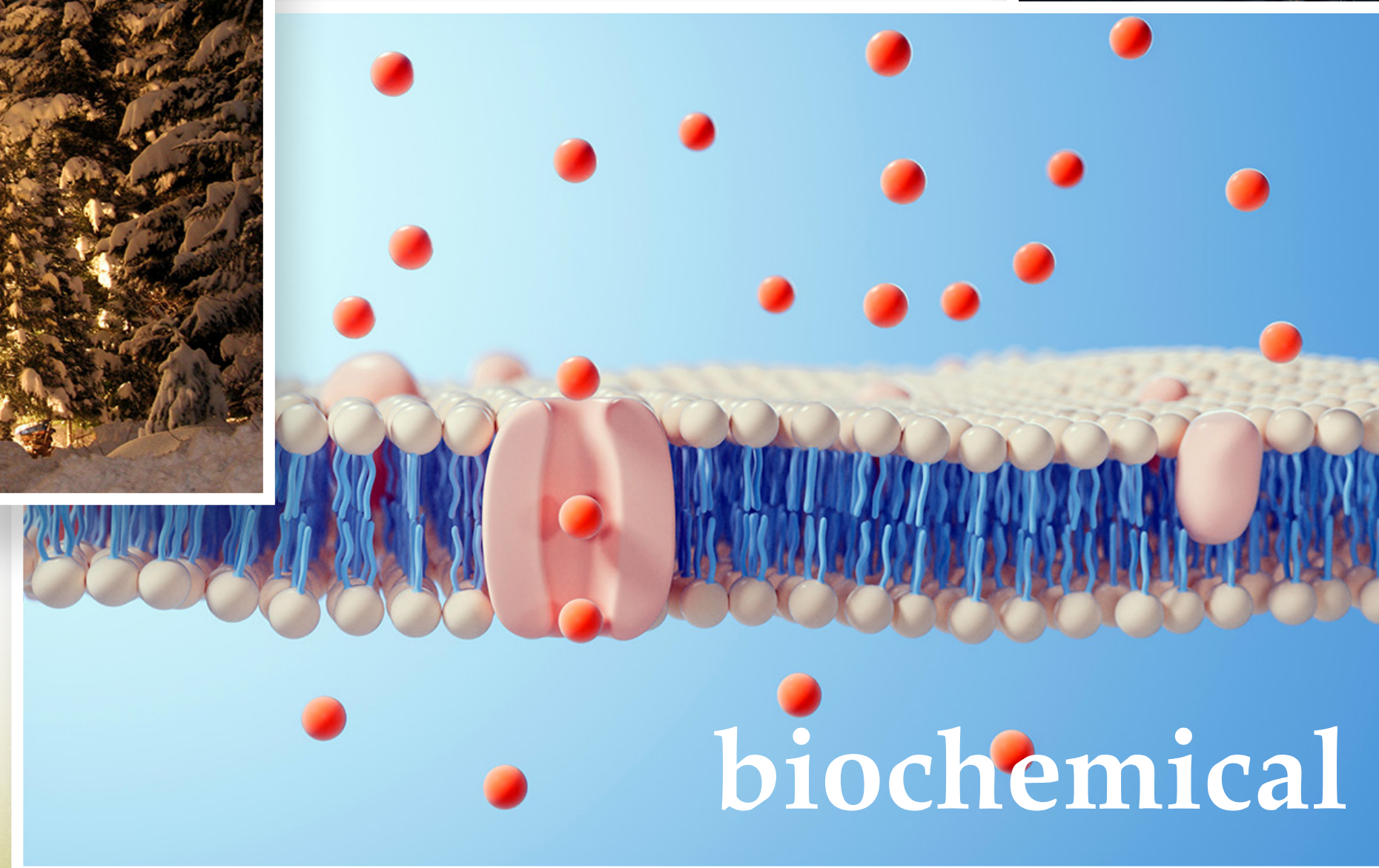
*perfectly* reflective walls





# Robin boundary conditions

real physical materials neither *perfectly* absorptive (**Dirichlet**)  
nor *perfectly* reflective (**Neumann**)





# Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \\ \frac{\partial u}{\partial n} - \mu u &= h && \text{on } \partial\Omega \end{aligned}$$



# Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \\ \frac{\partial u}{\partial n} - \mu u &= h && \text{on } \partial\Omega \end{aligned}$$

**Neumann**  
(reflective)

**Dirichlet**  
(absorbing)



# Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega \\ \frac{\partial u}{\partial n} - \mu u &= h && \text{on } \partial\Omega \end{aligned}$$

**Neumann**  
(reflective)

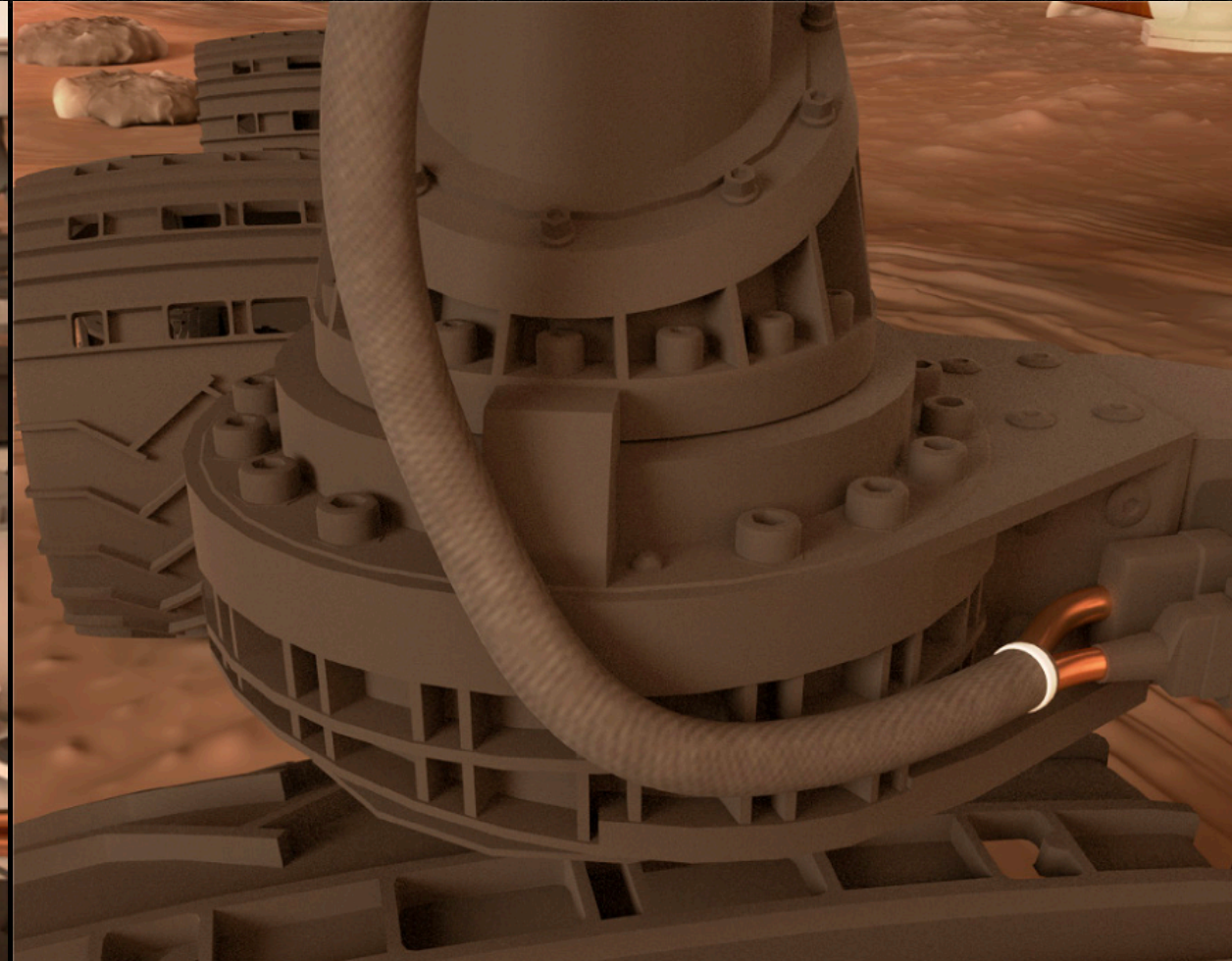
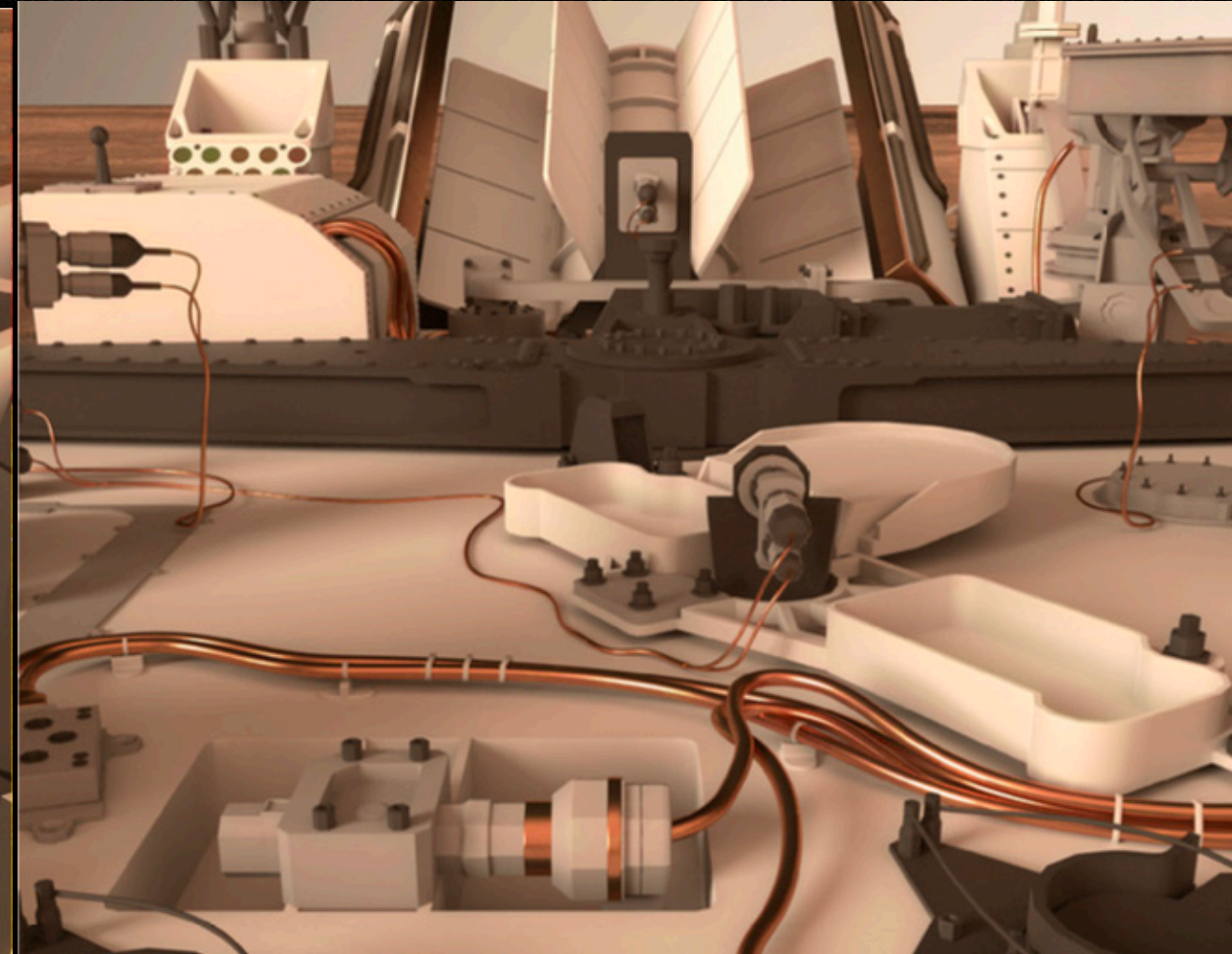
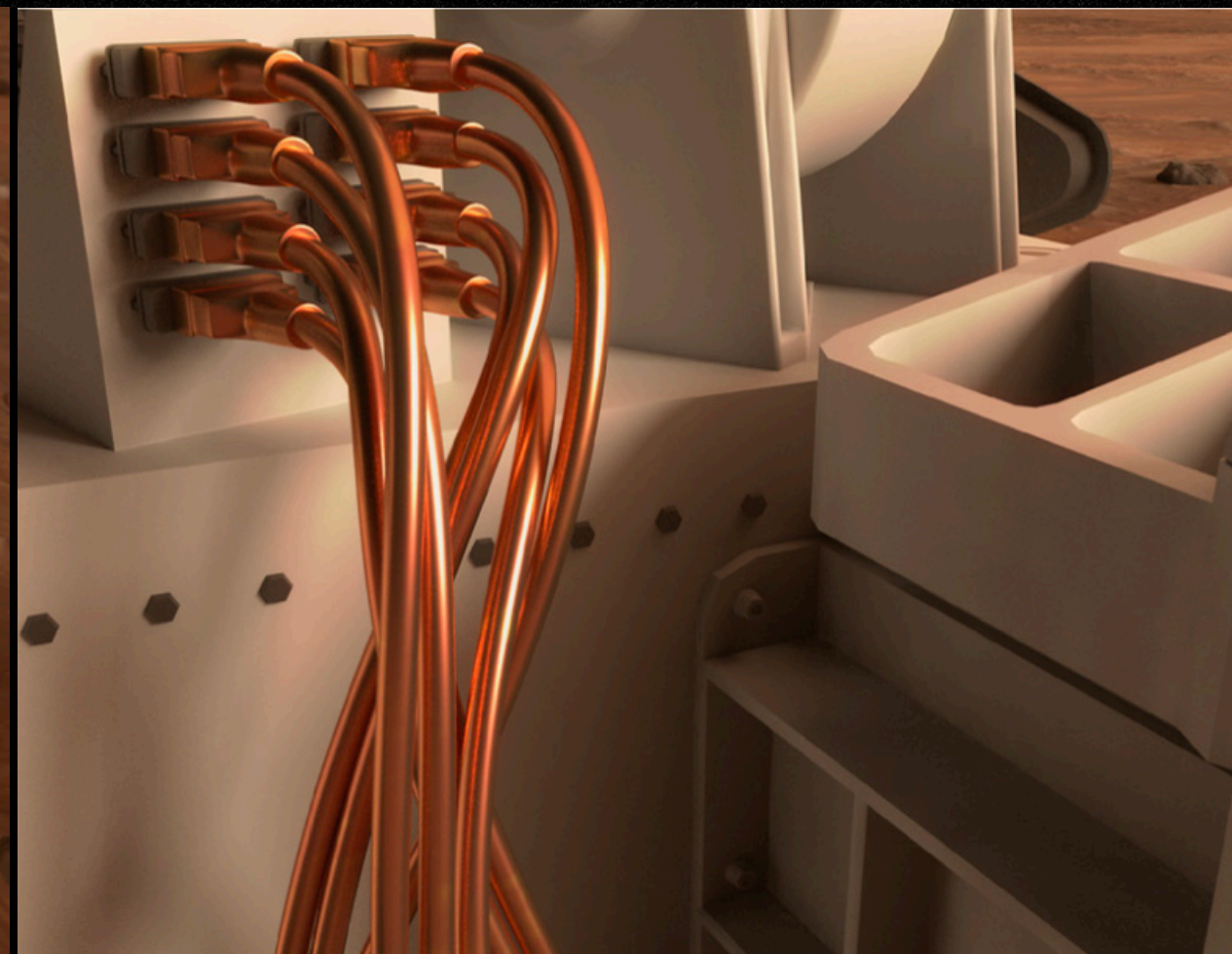
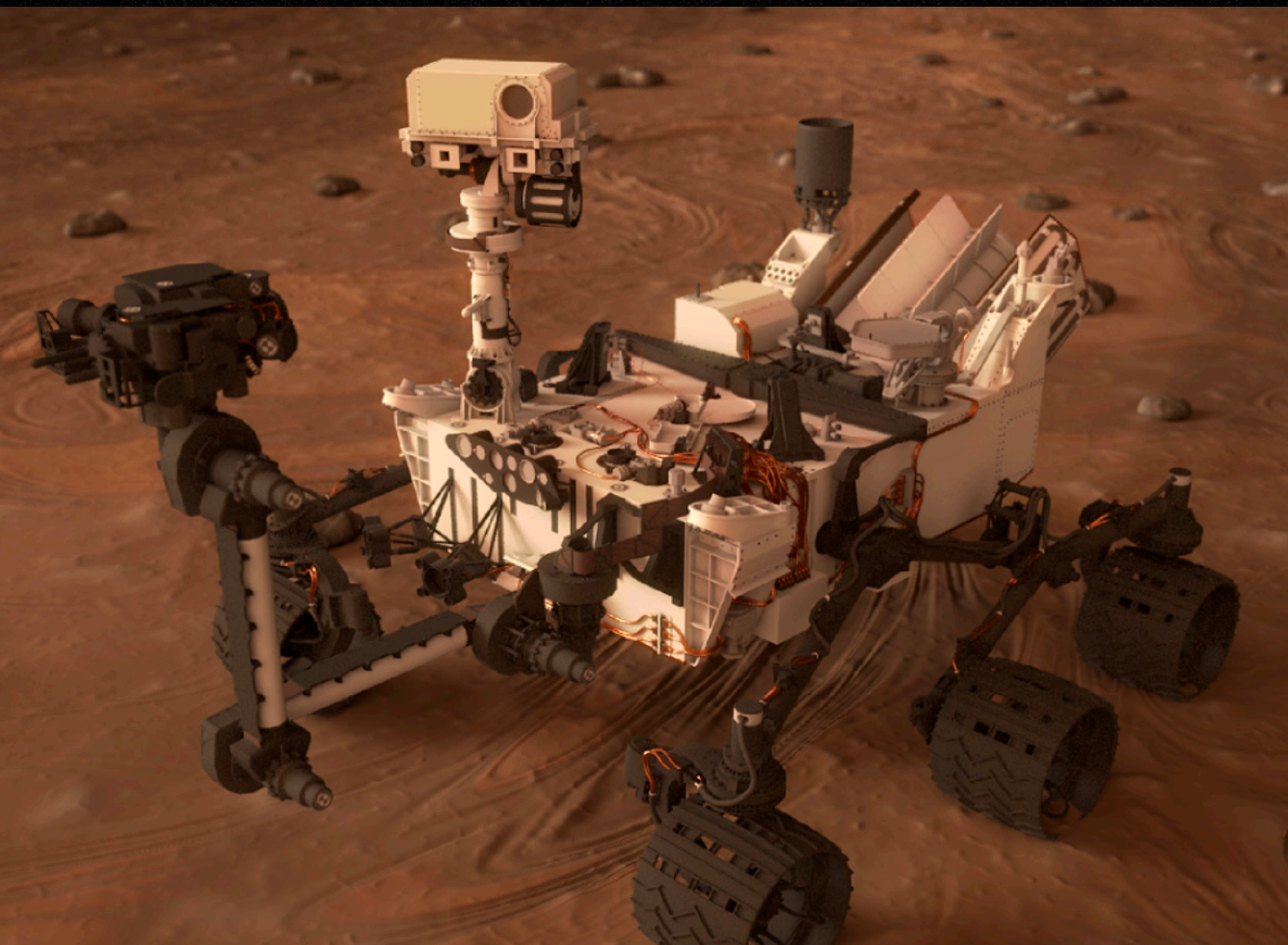
**Dirichlet**  
(absorbing)

$$\mu > 0$$

“how easily does the material absorb heat?”

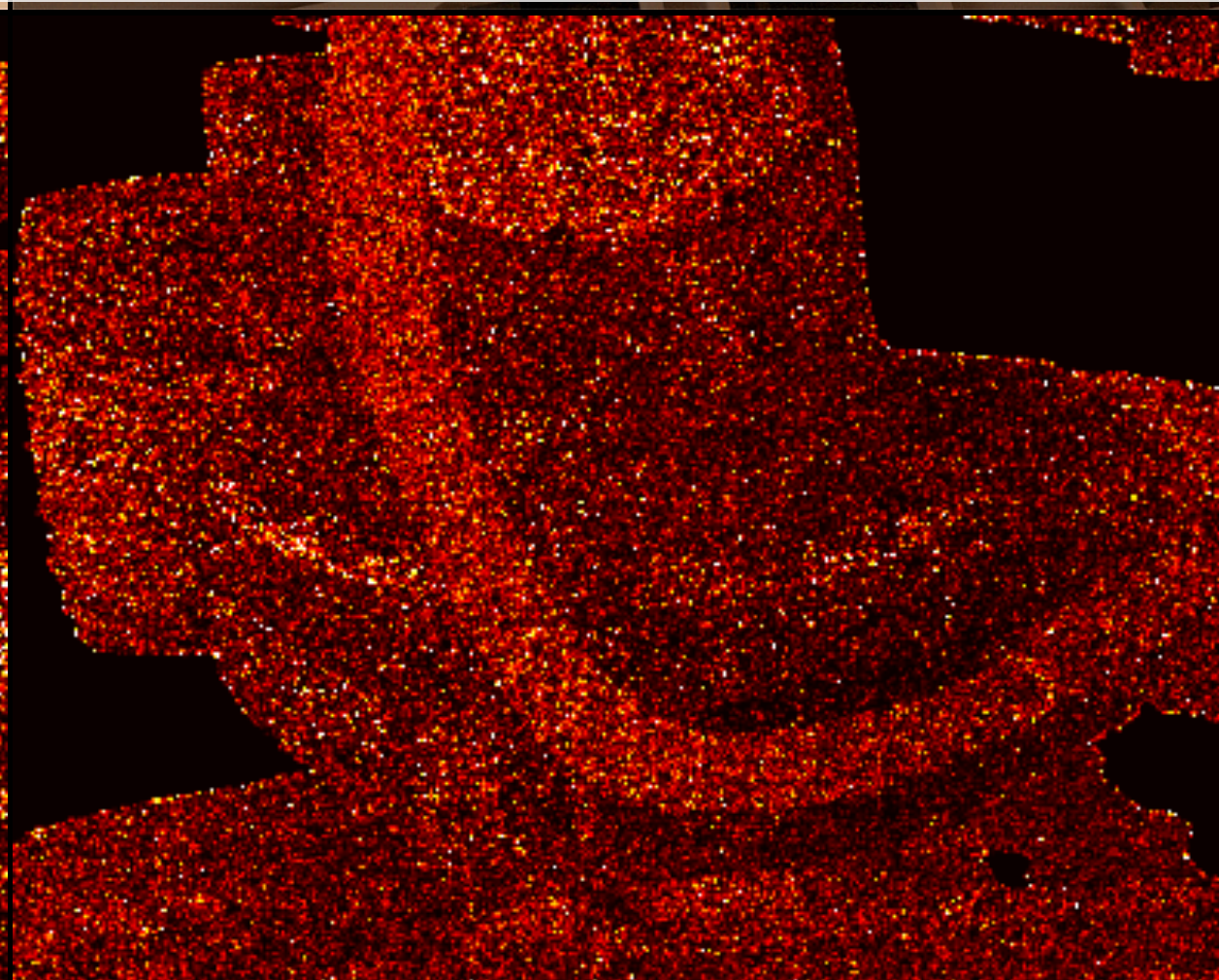
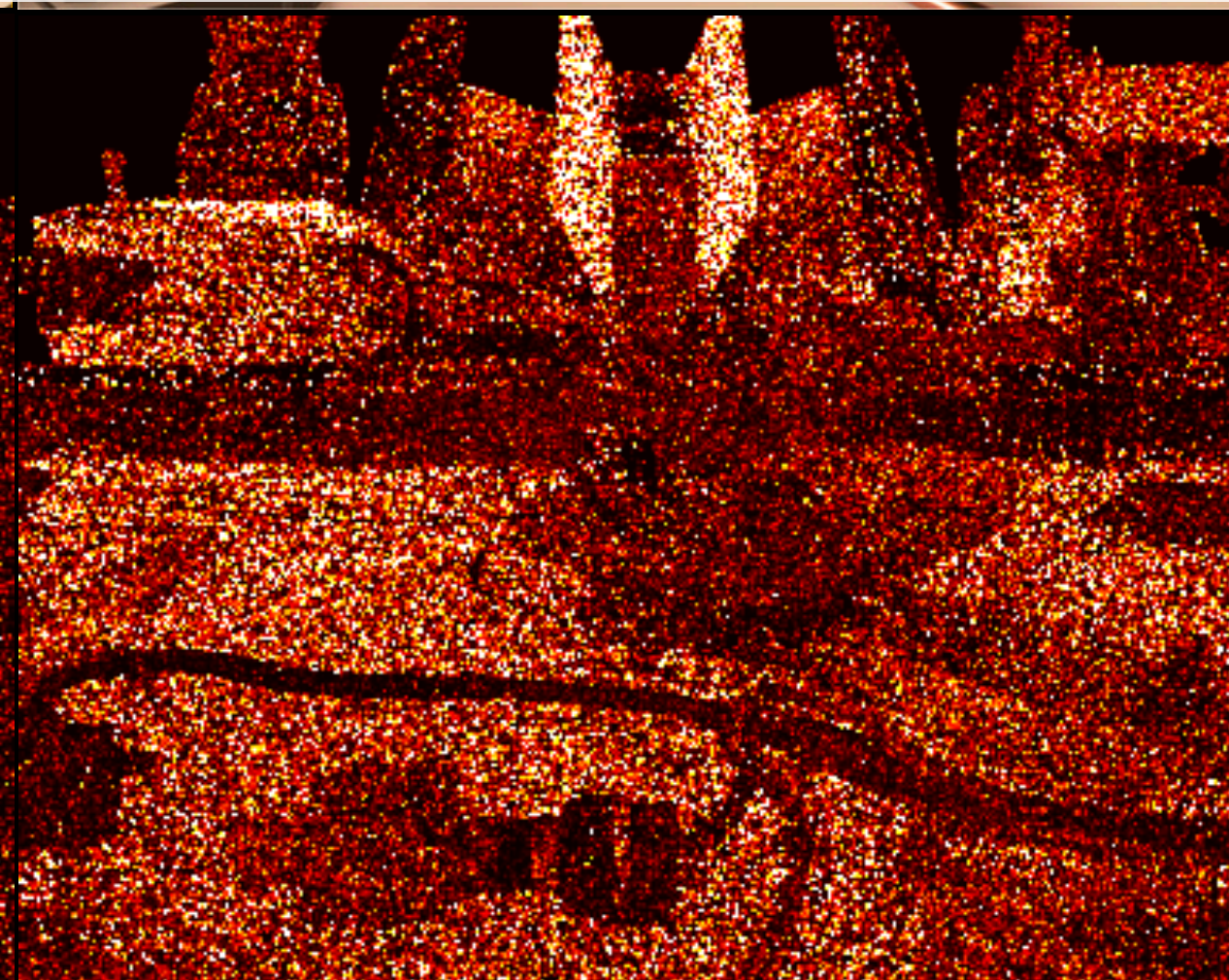
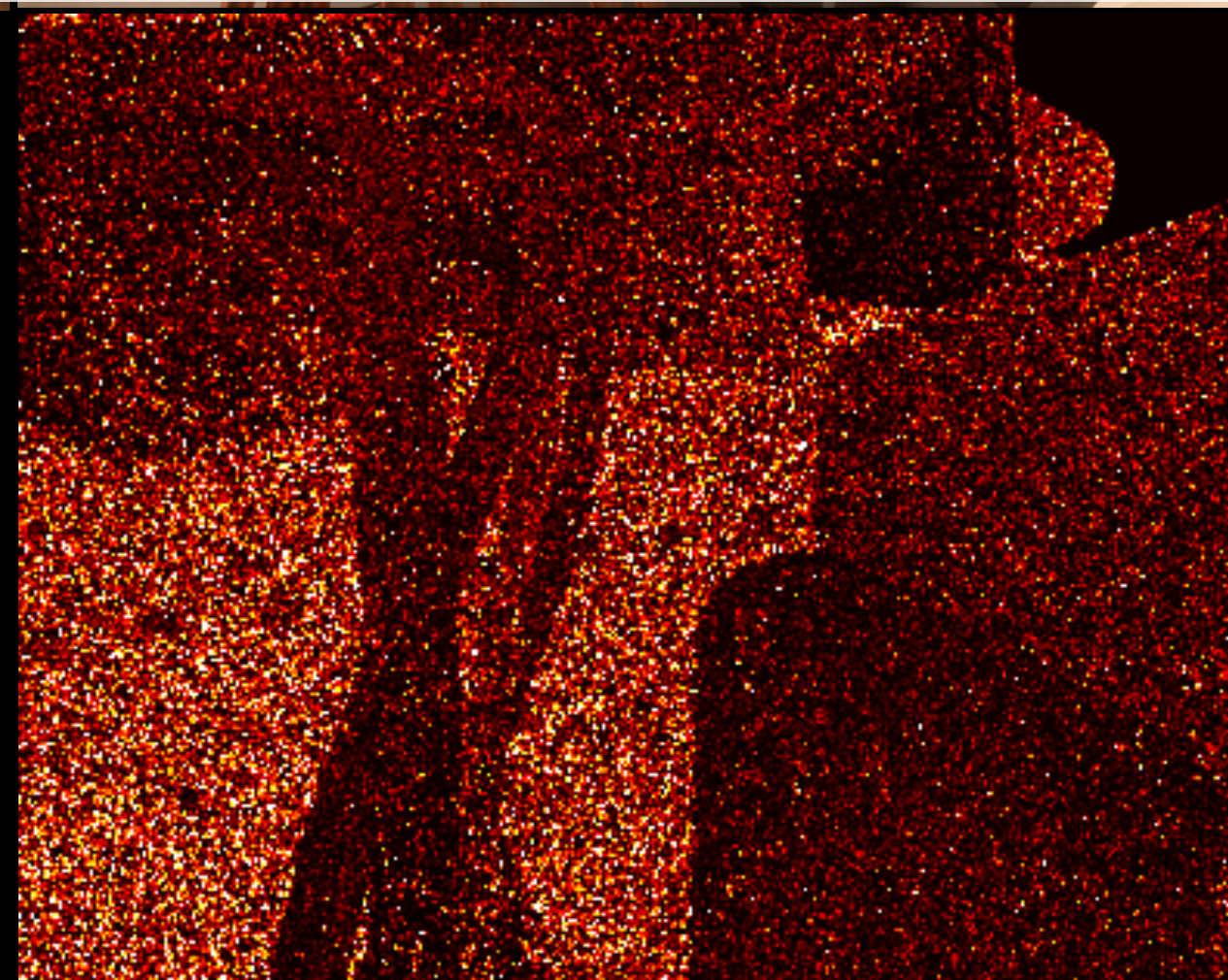


# Robin boundary conditions



visualization (light transport)

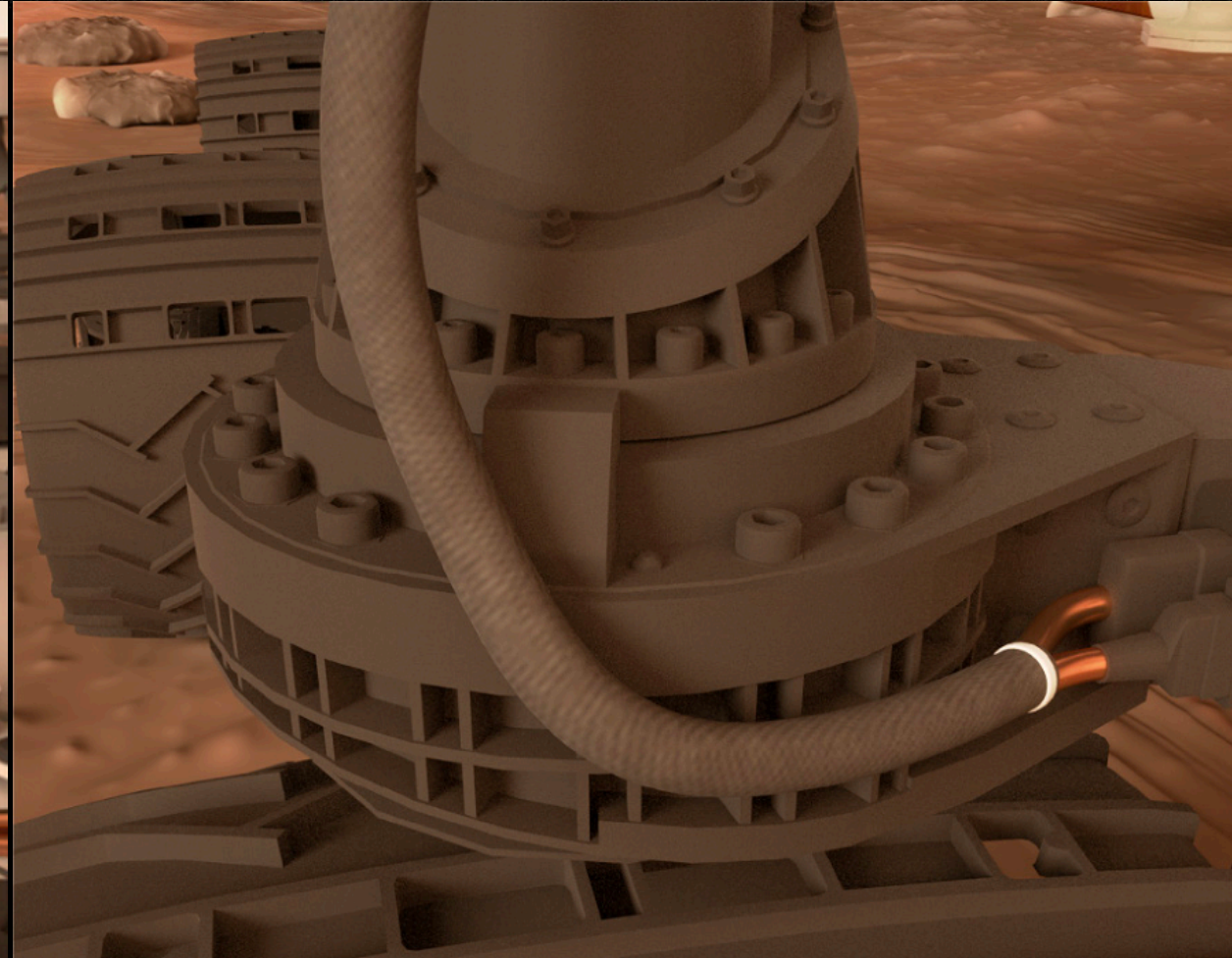
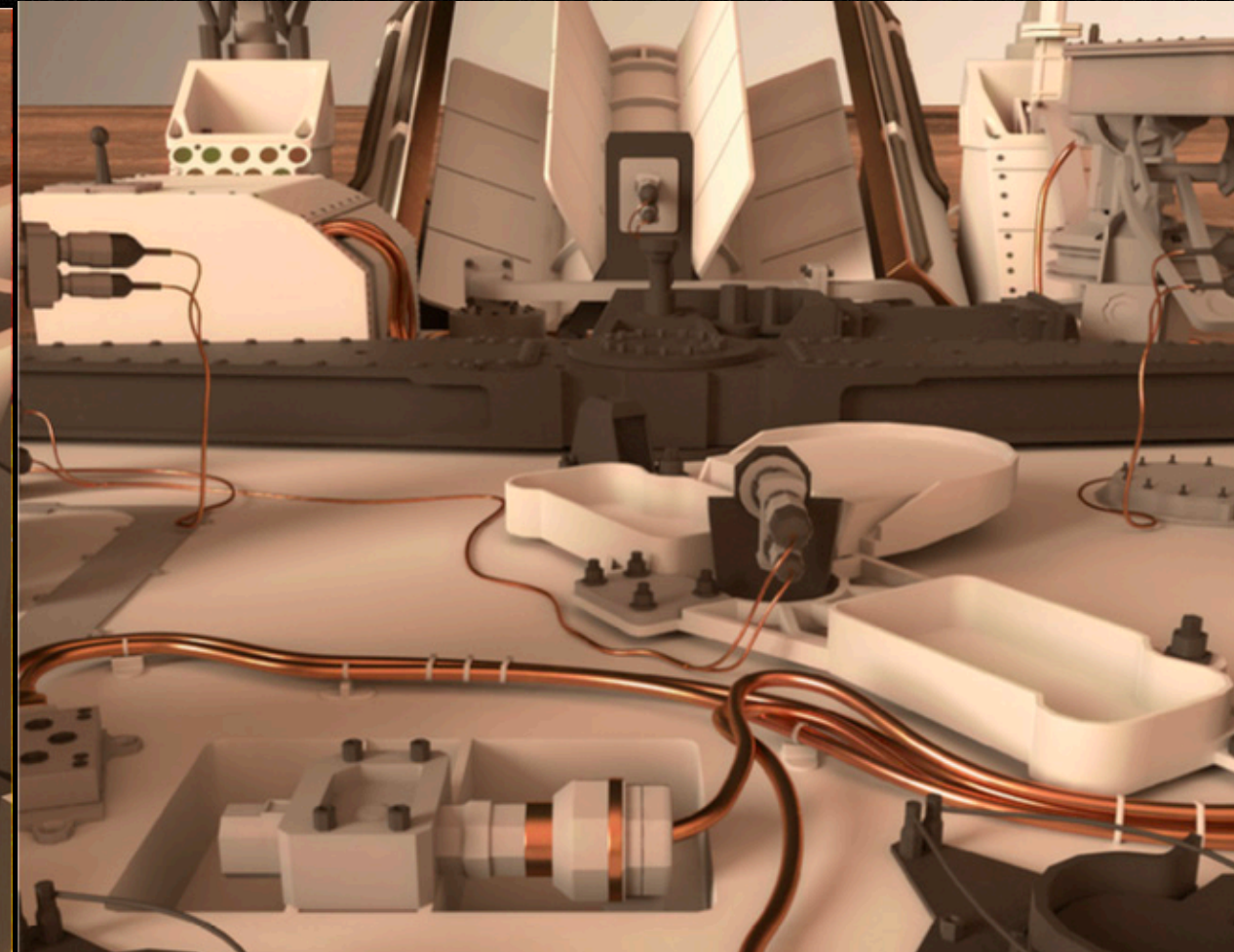
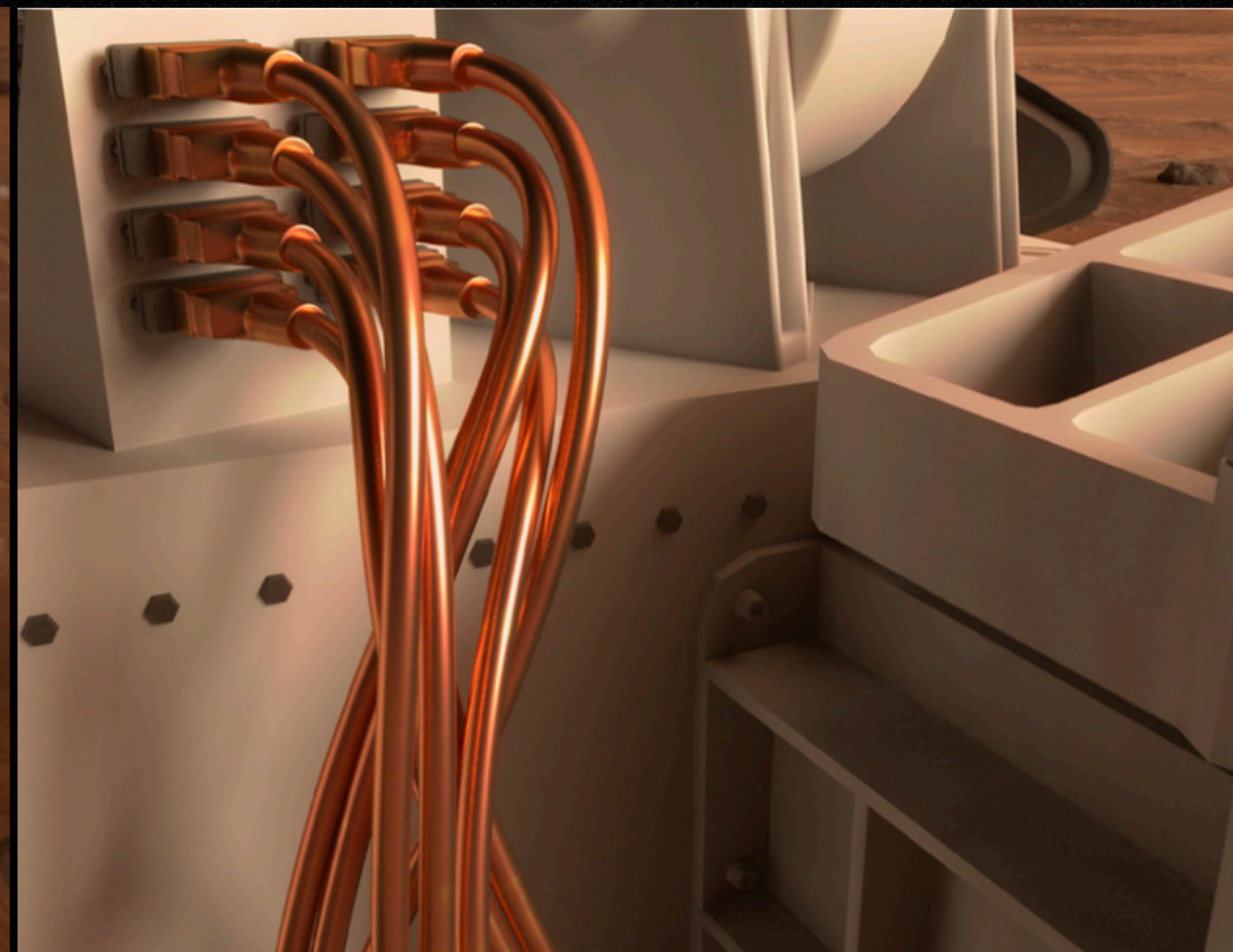
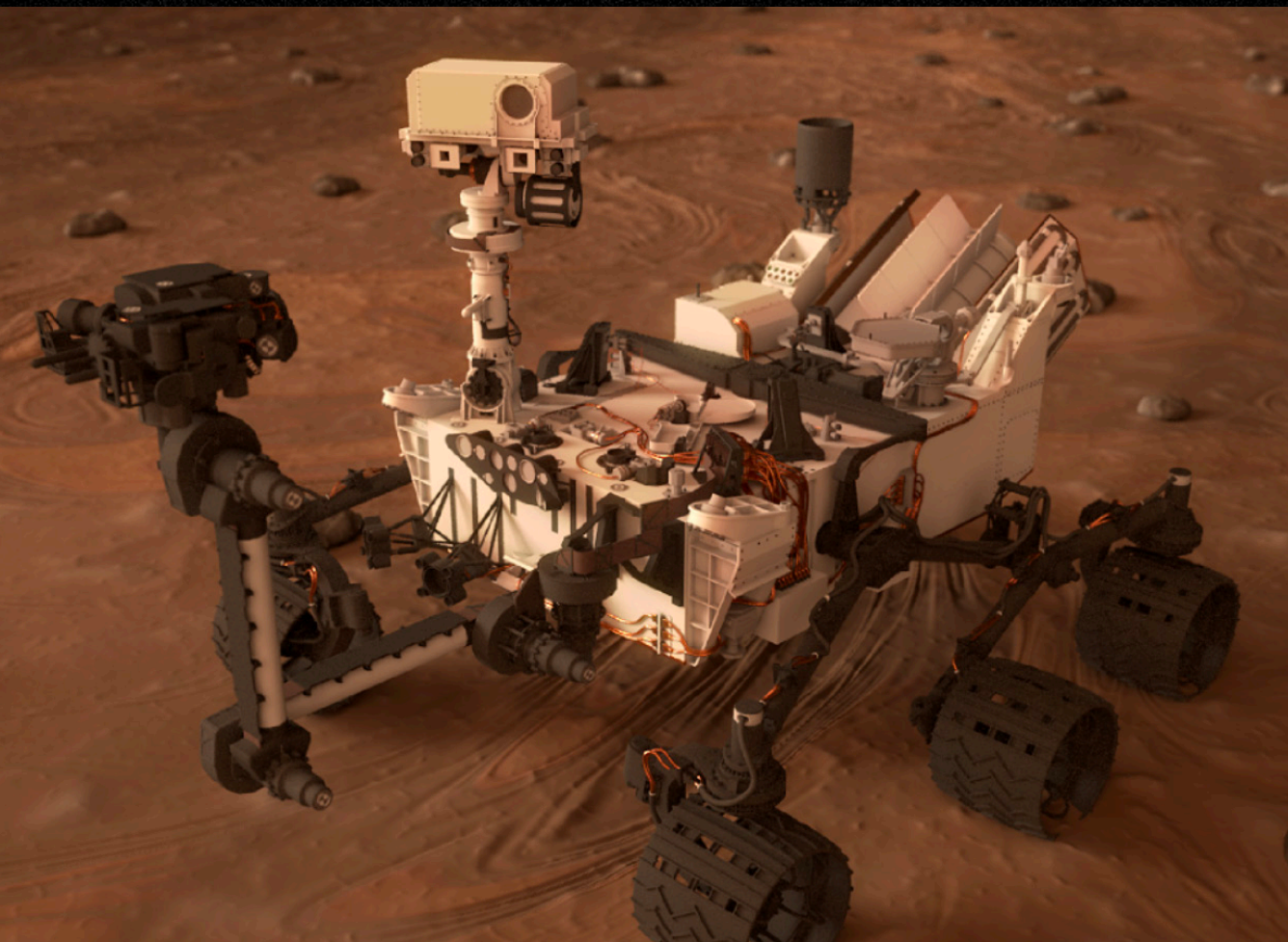
Robin coefficients  $\mu$



simulation (thermal conduction)

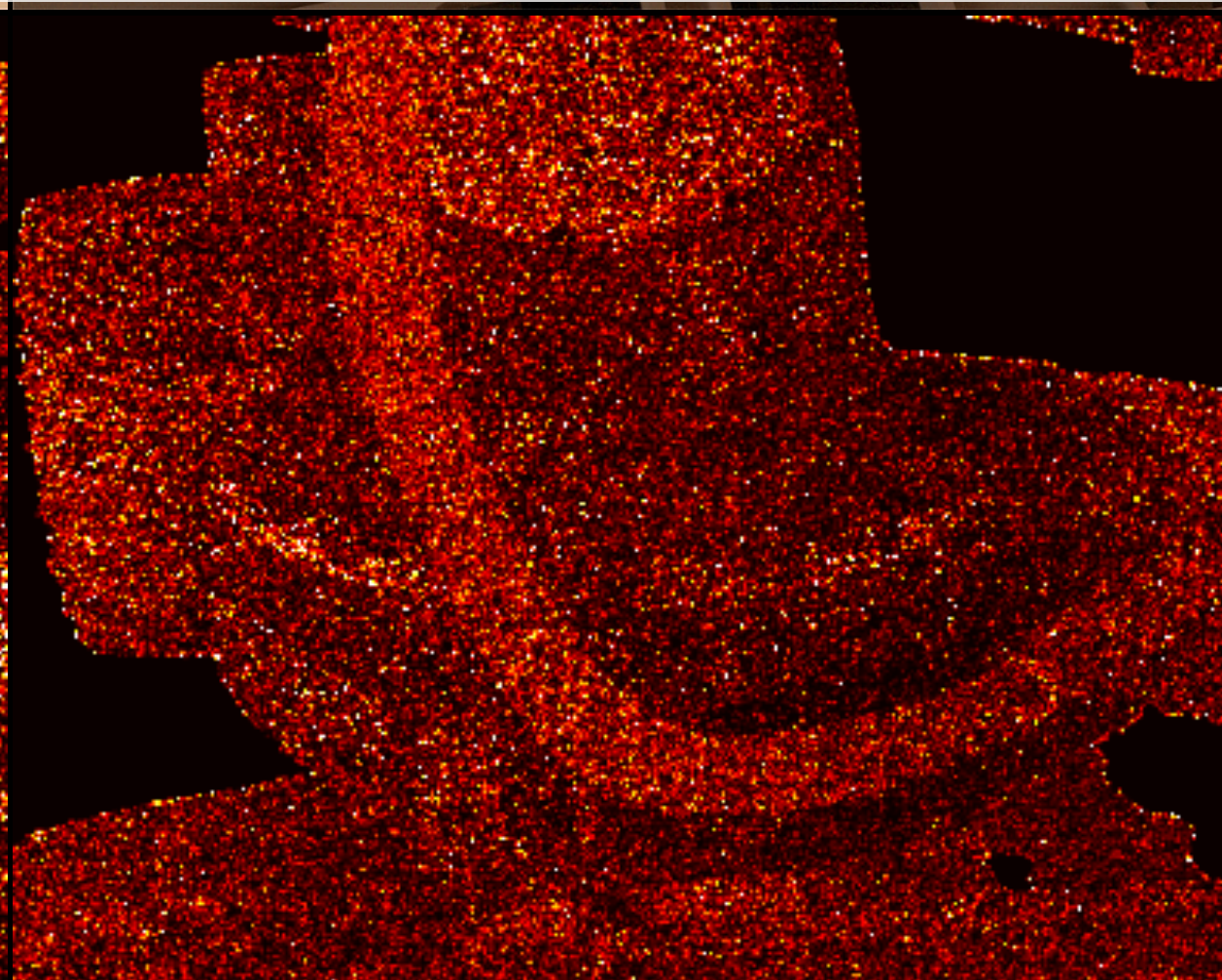
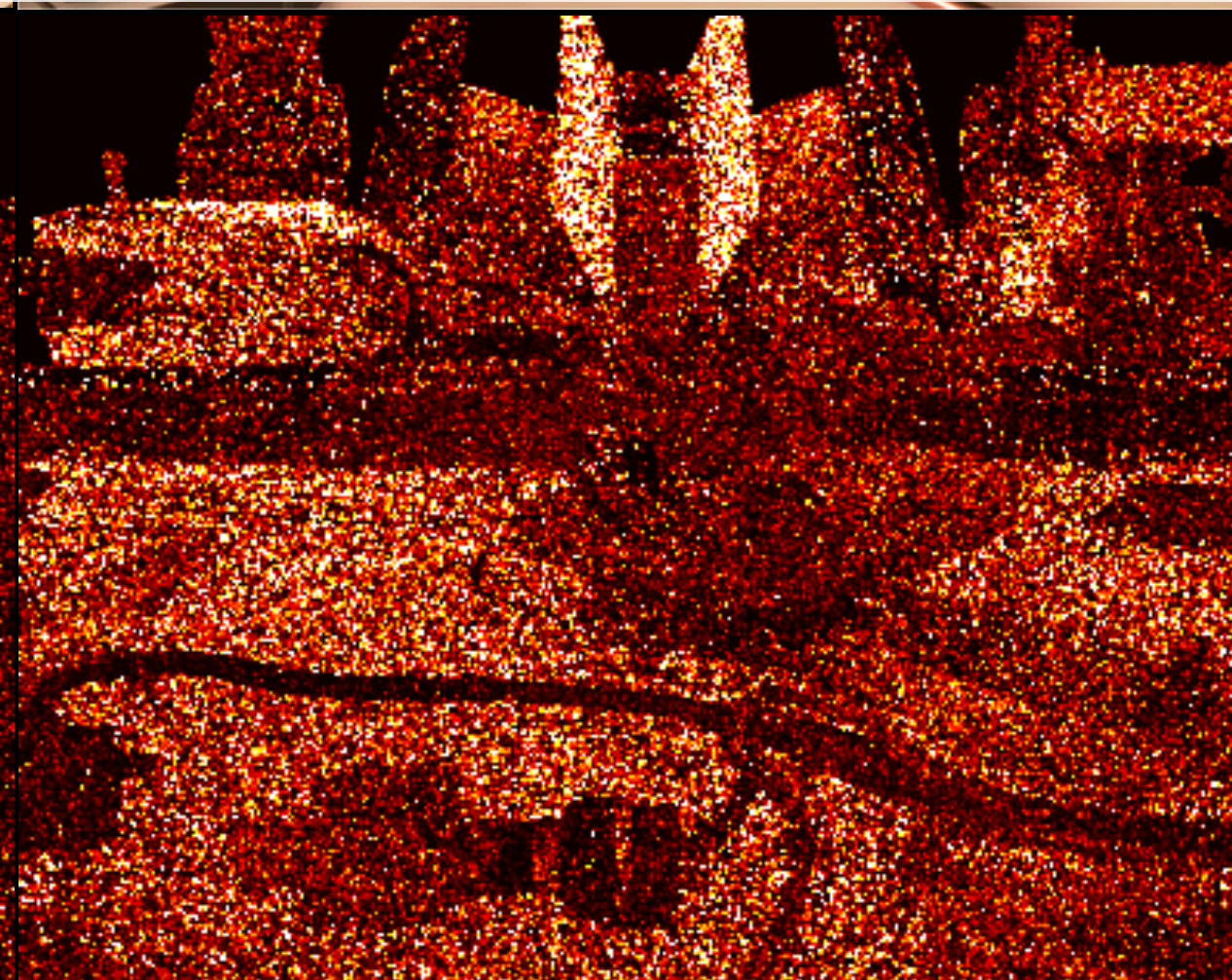
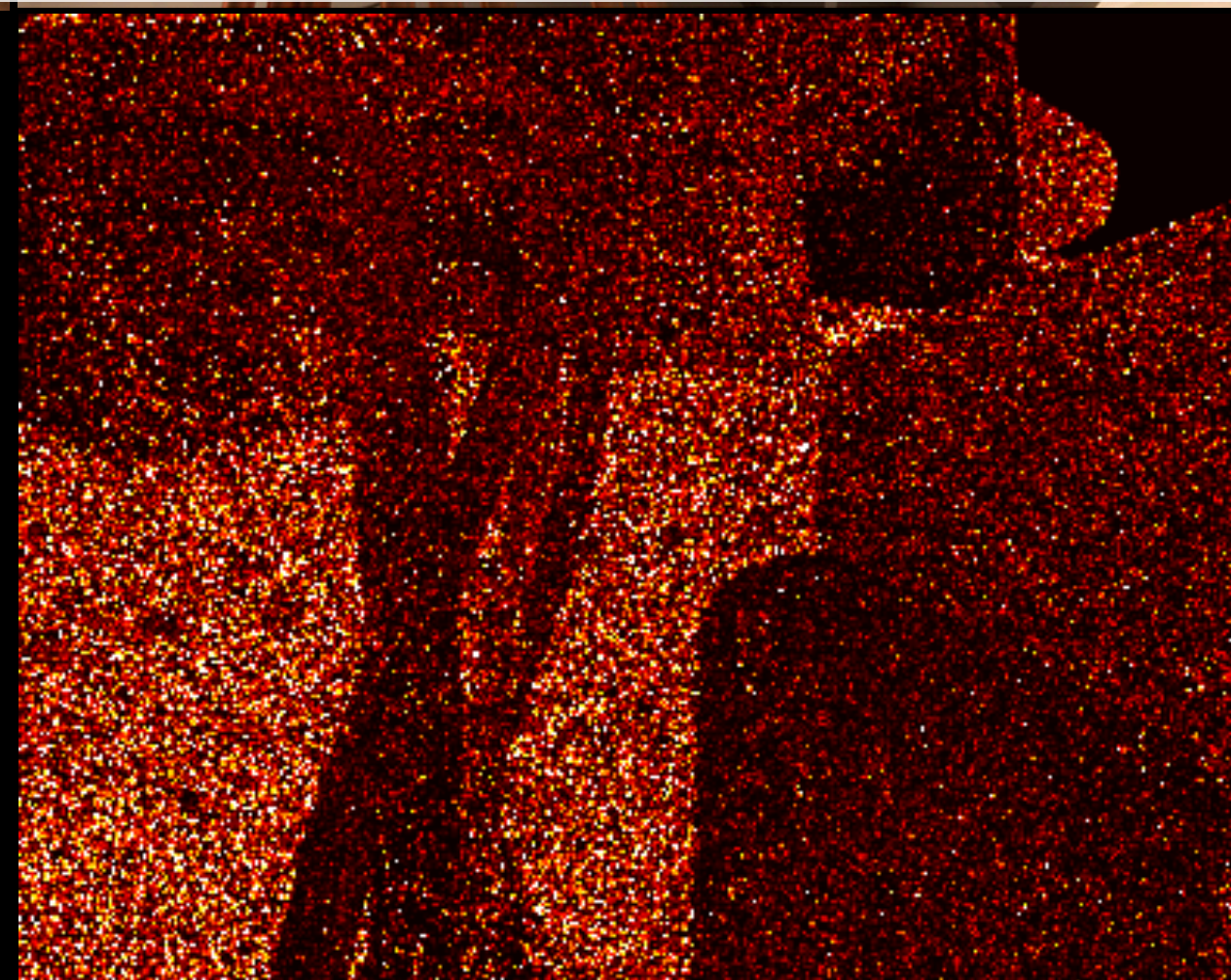
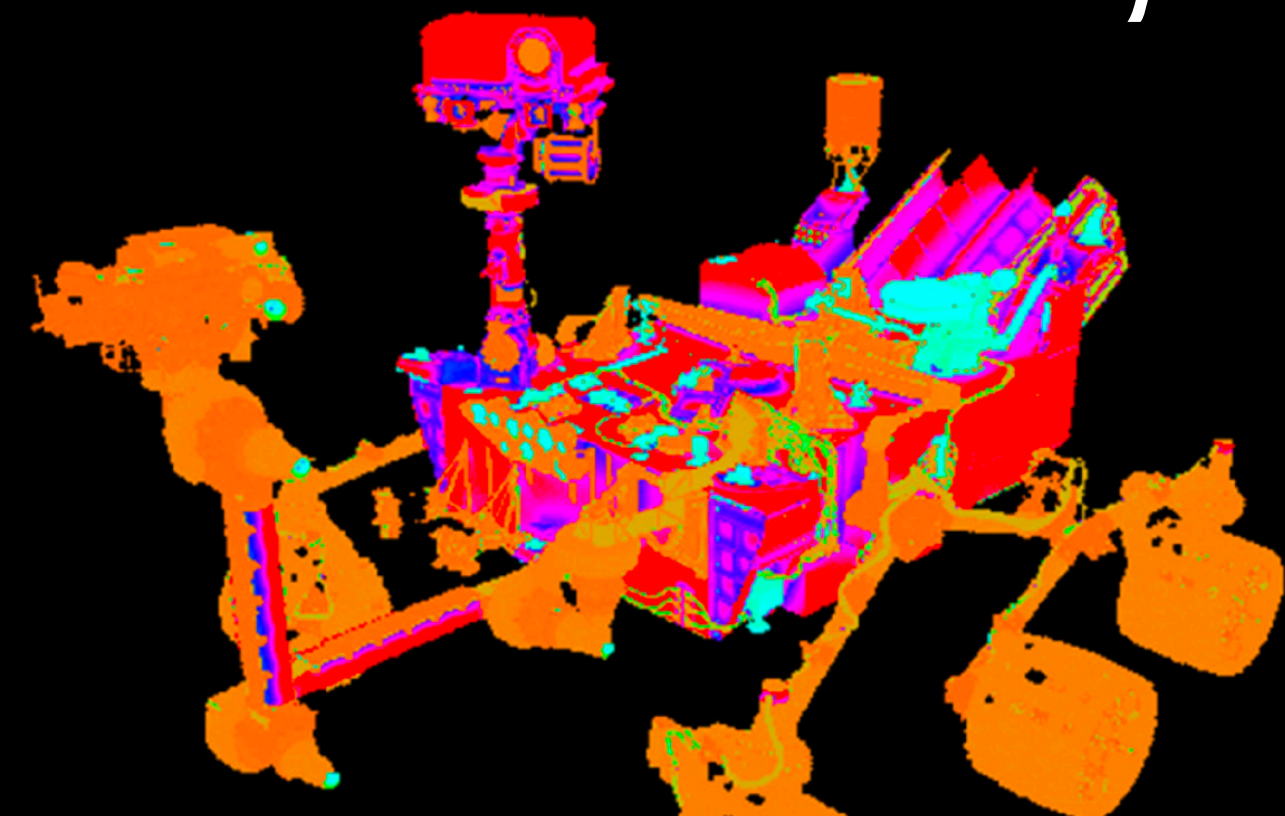


# Robin boundary conditions



visualization (light transport)

Robin coefficients  $\mu$



simulation (thermal conduction)



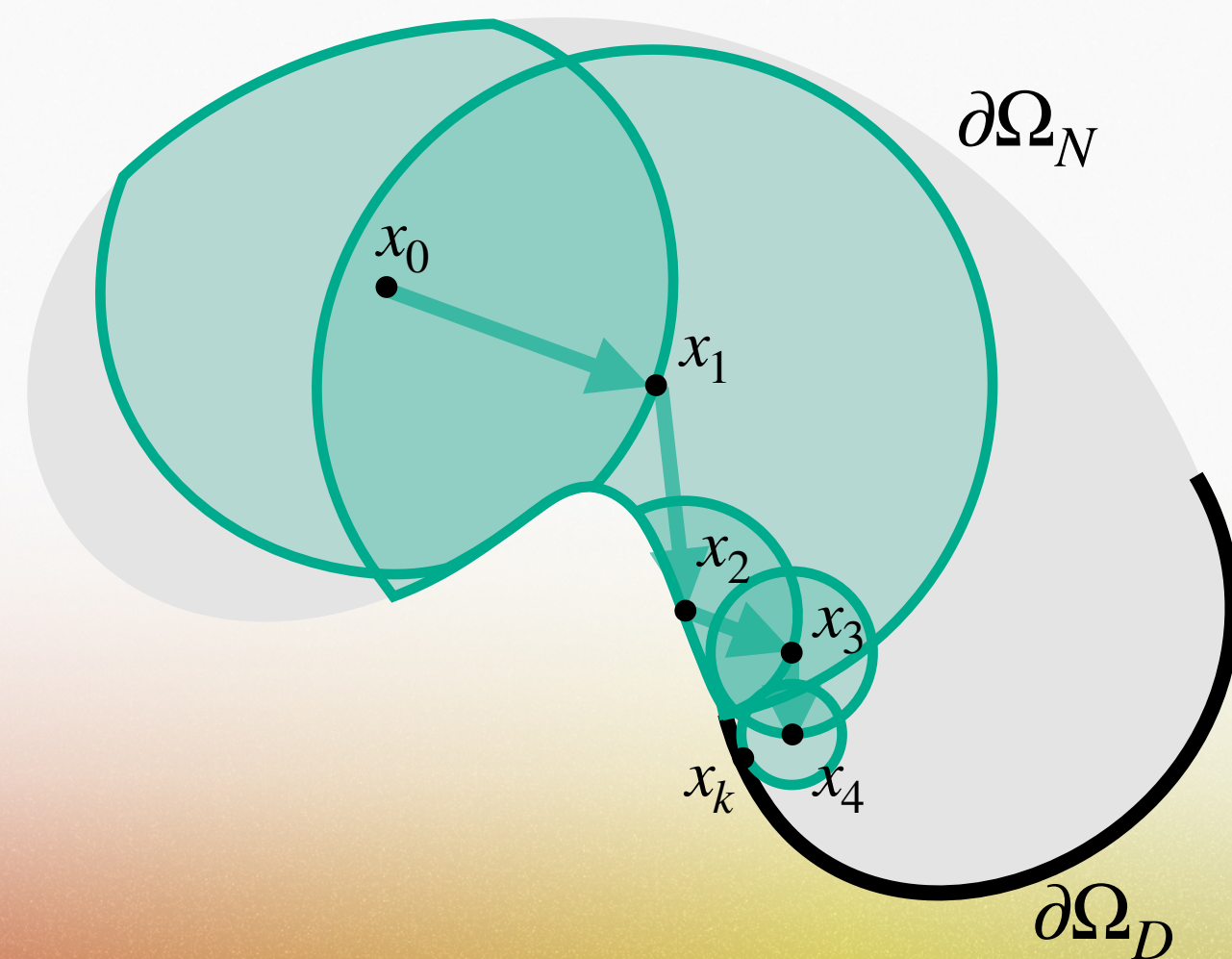
# generalizing walk on stars

[Sawhney et al. 2023]

```
x = x0
until Dirichlet boundary reached:

    S = find_largest_star_shape(x)
    x = sample_point_on_boundary(S)

return g(x)
```



with Robin  
boundary conditions

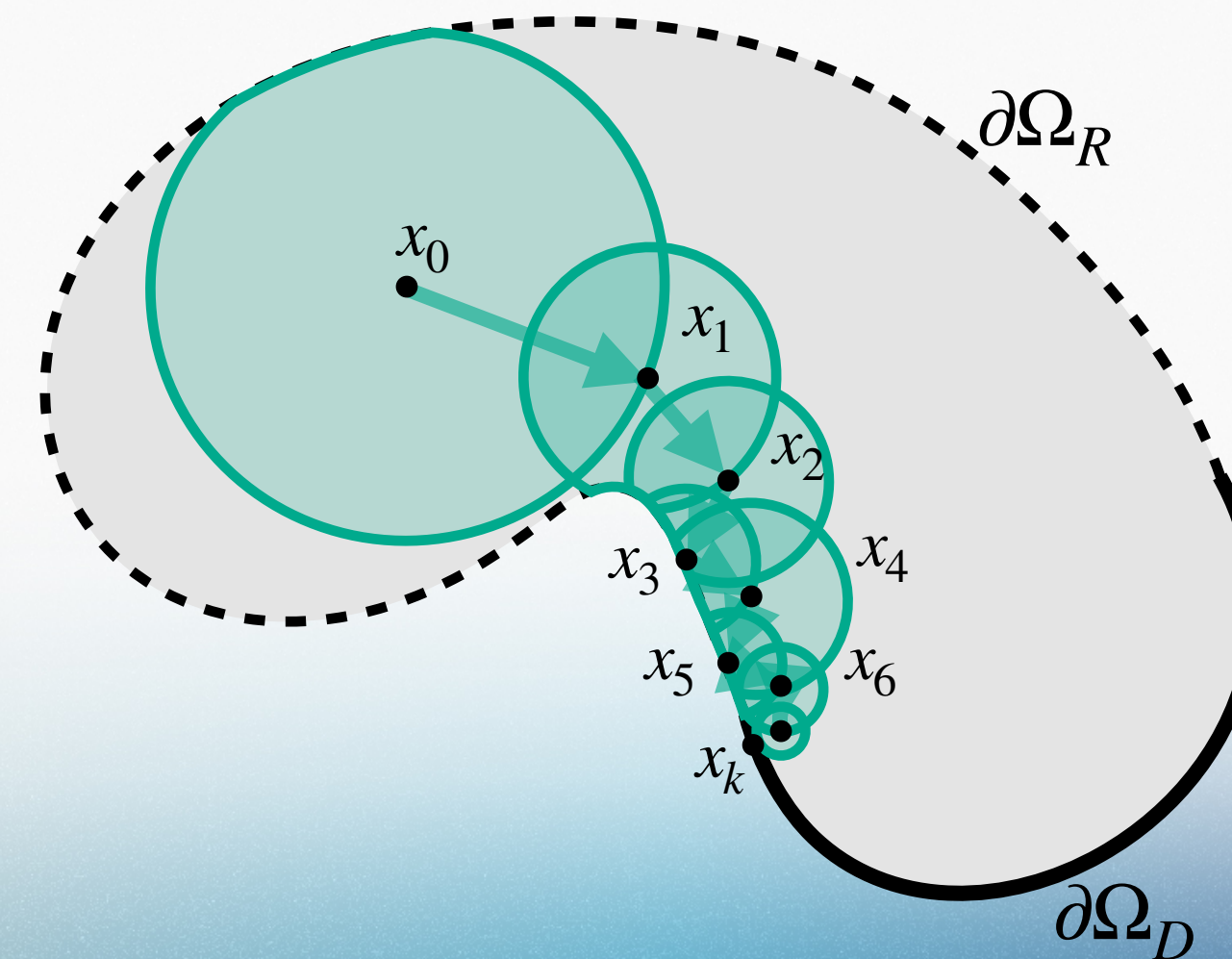


ours

```
r = 1; x = x0
until Dirichlet boundary reached:

    S = find_reflectance_bounded_star_shape(x)
    x = sample_point_on_boundary(S)
    r *= reflectance(S, x)

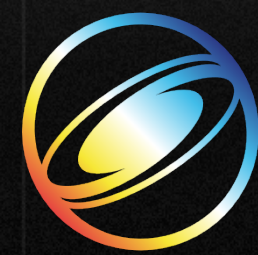
return r * g(x)
```





SIGGRAPH

2024



**SIGGRAPH 2024**

DENVER+ 28 JUL — 1 AUG

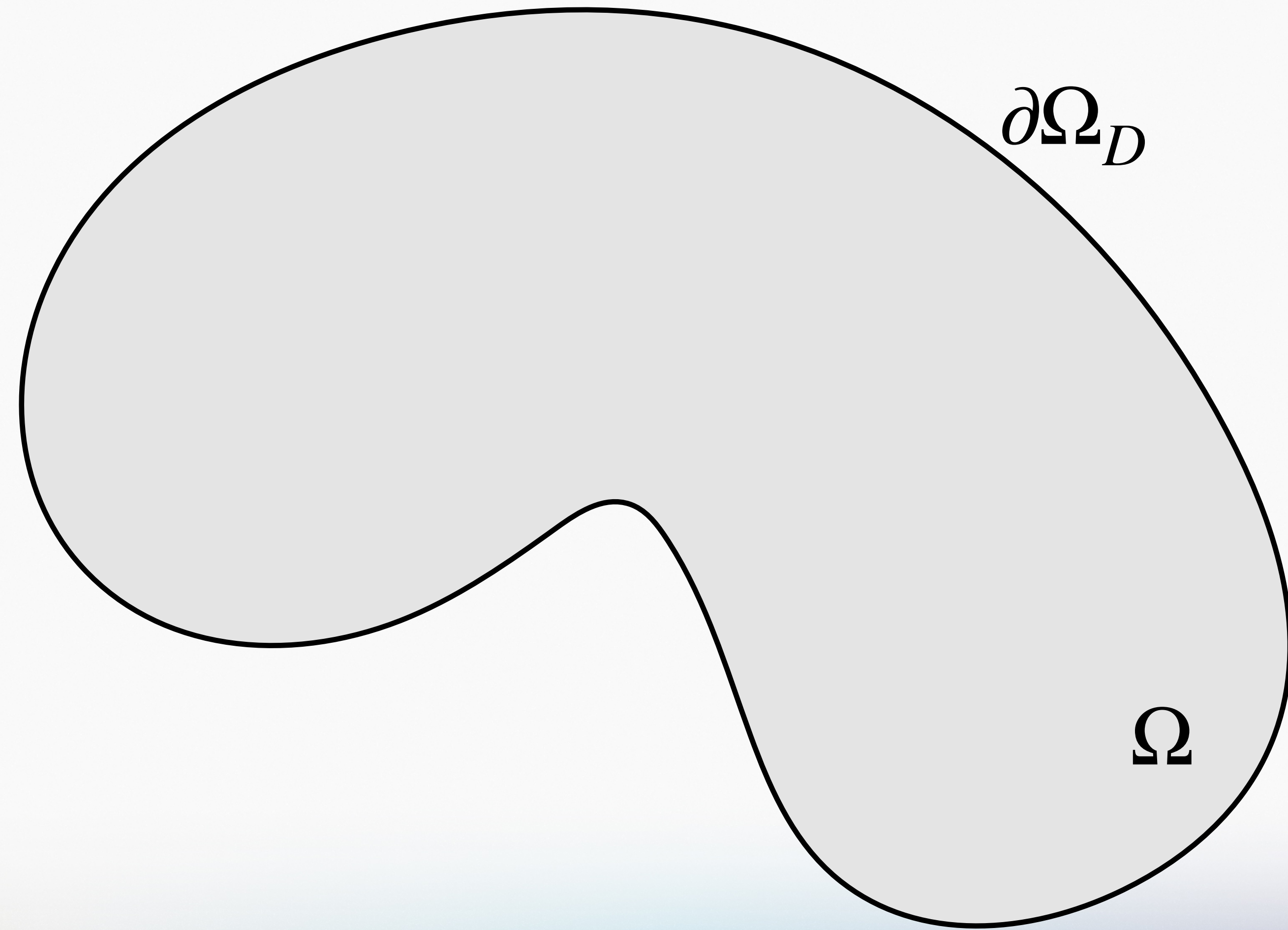
# WALK ON SPHERES TO WALK ON STARS





# Dirichlet problem

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

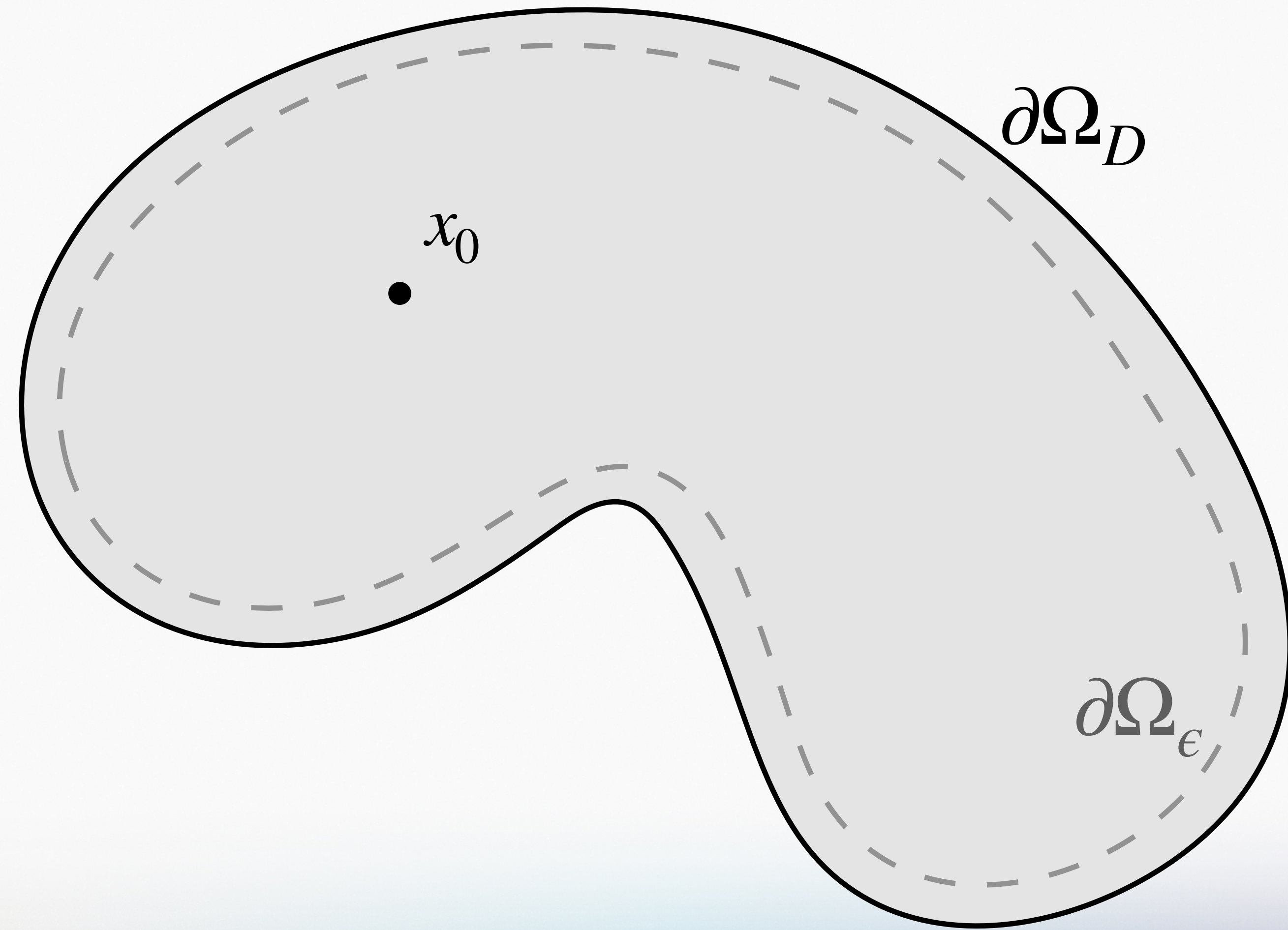




# walk on spheres [Muller 1956, Sawhney and Crane 2020]

mean value integral

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

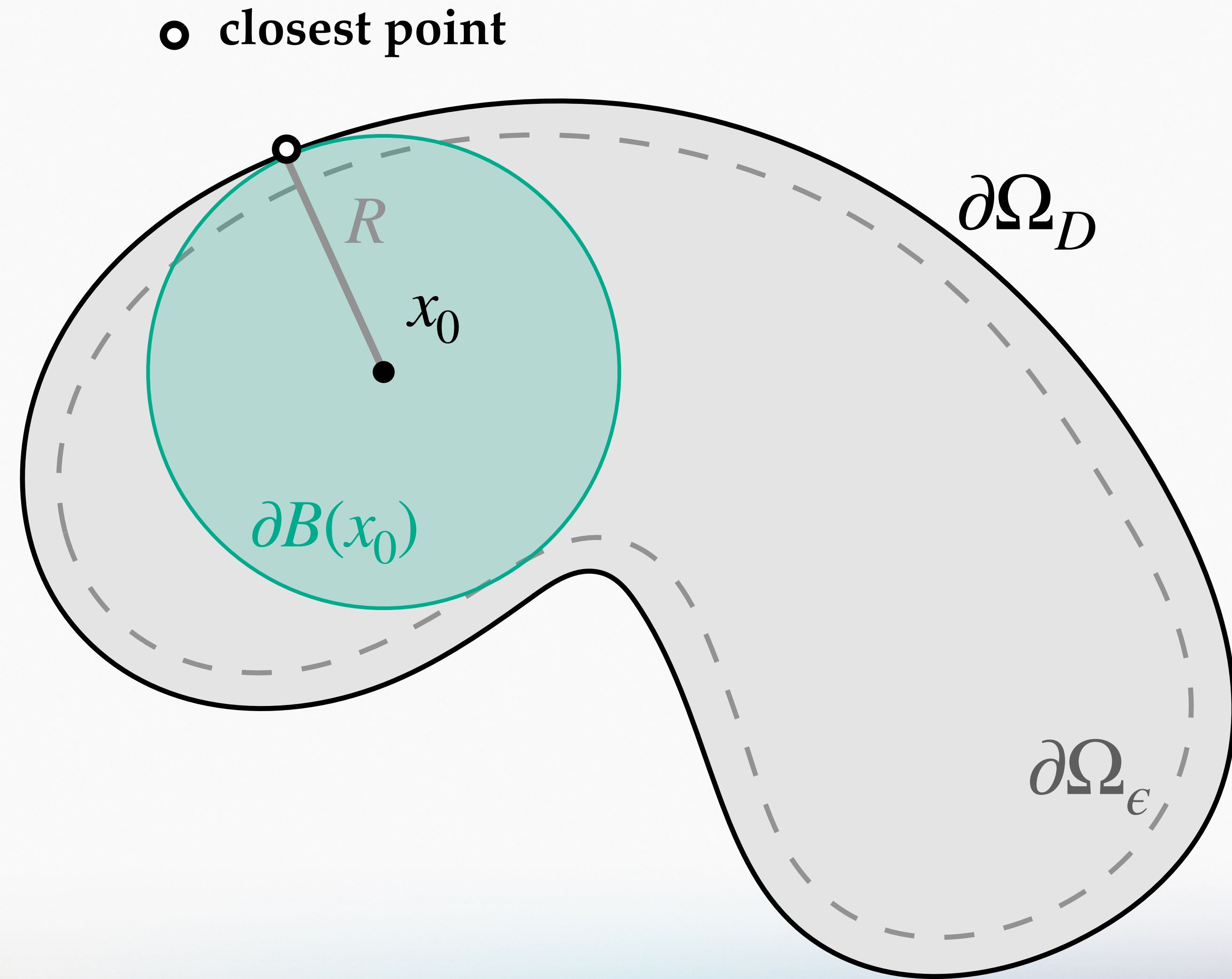




# walk on spheres [Muller 1956, Sawhney and Crane 2020]

mean value integral

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





# walk on spheres [Muller 1956, Sawhney and Crane 2020]

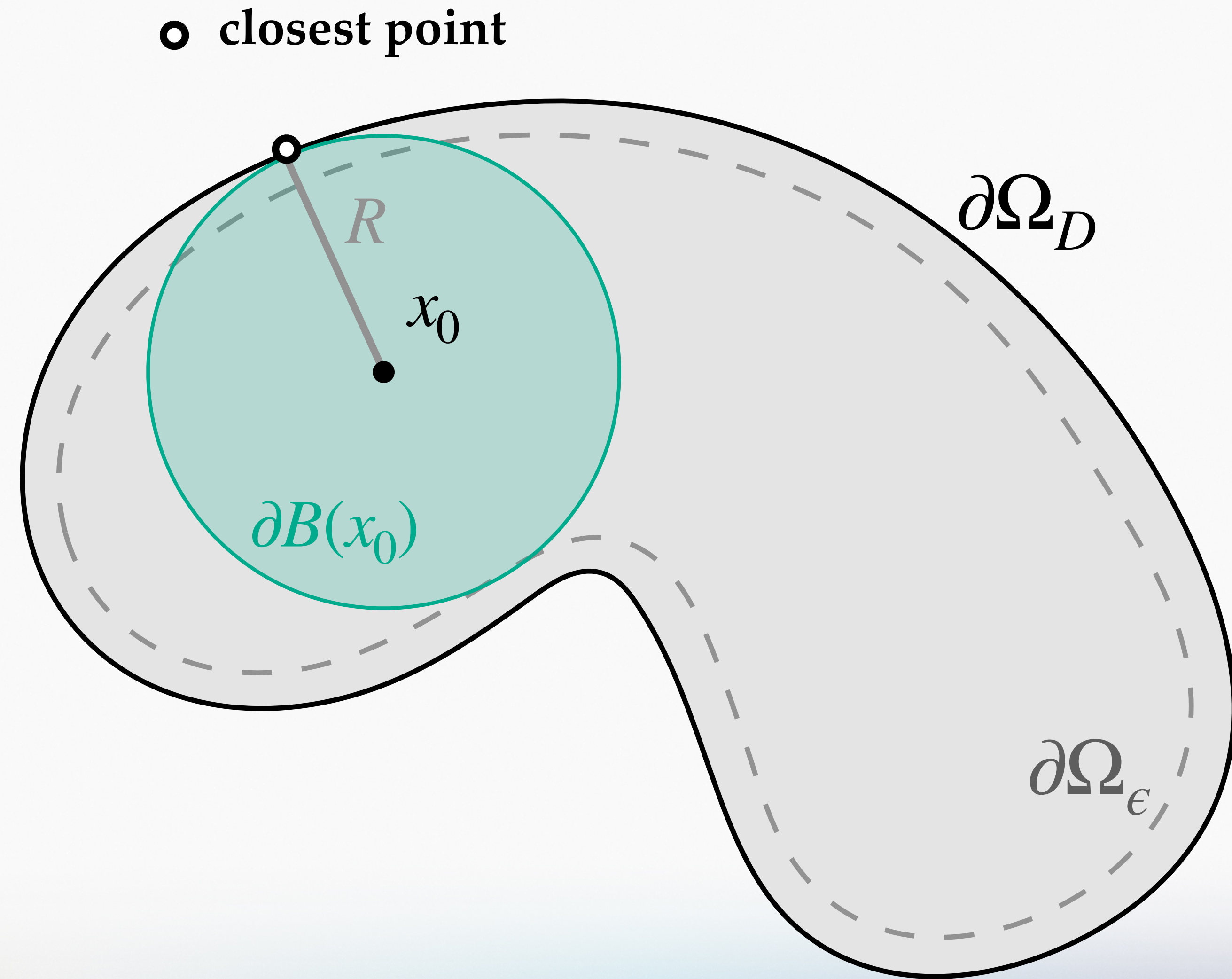
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere  $y \sim U[\partial B(x)]$





# walk on spheres [Muller 1956, Sawhney and Crane 2020]

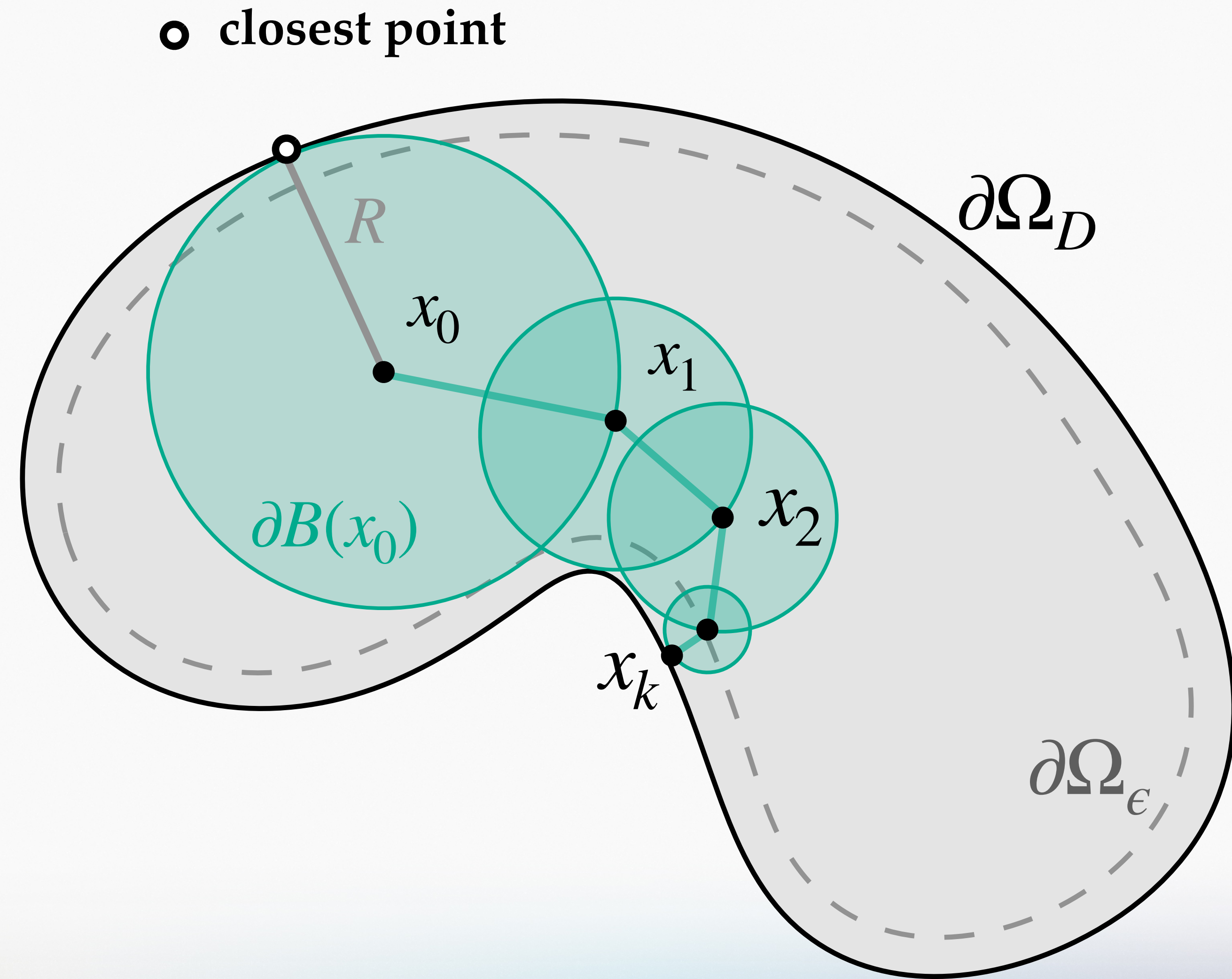
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere  $y \sim U[\partial B(x)]$





# walk on spheres [Muller 1956, Sawhney and Crane 2020]

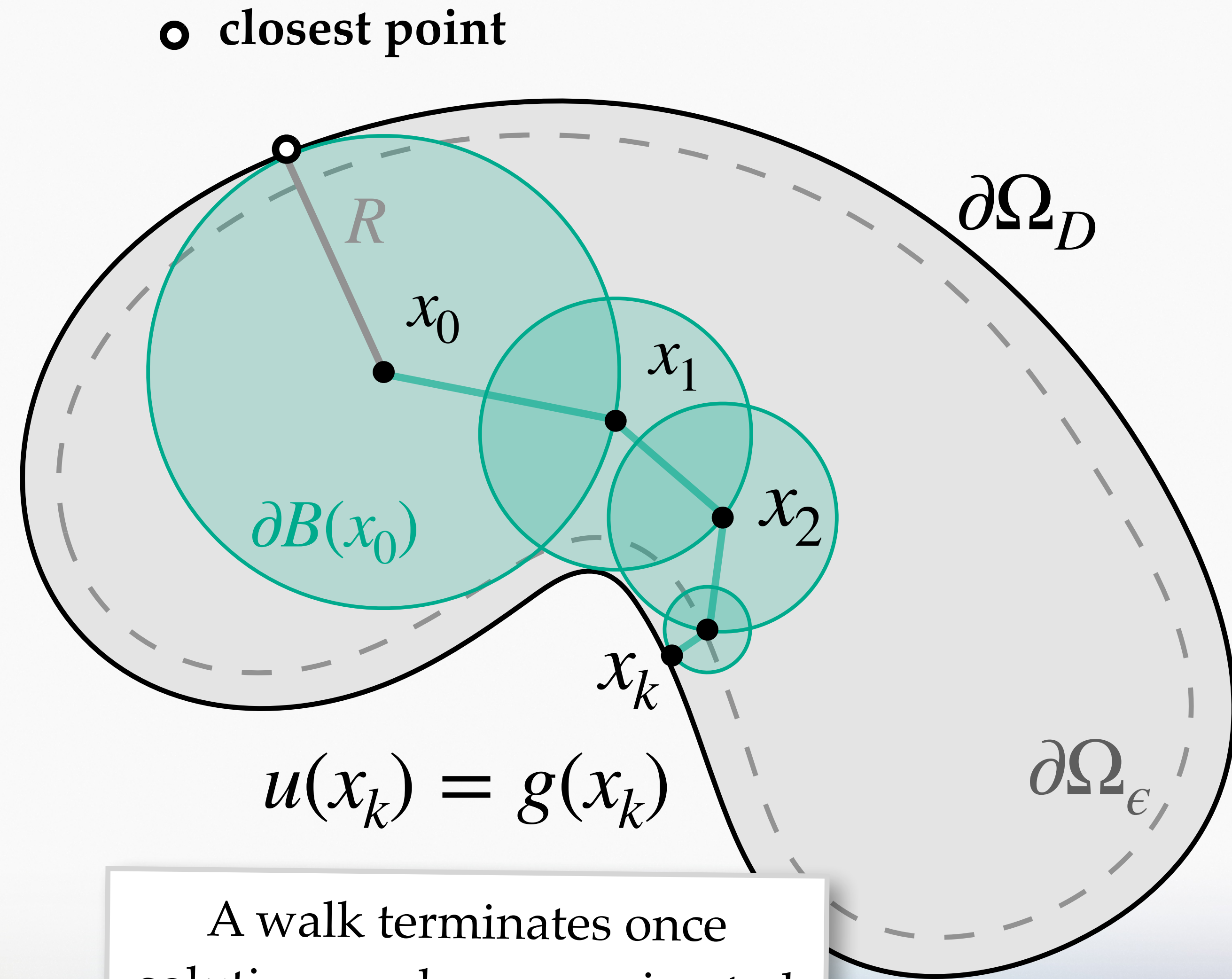
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere  $y \sim U[\partial B(x)]$

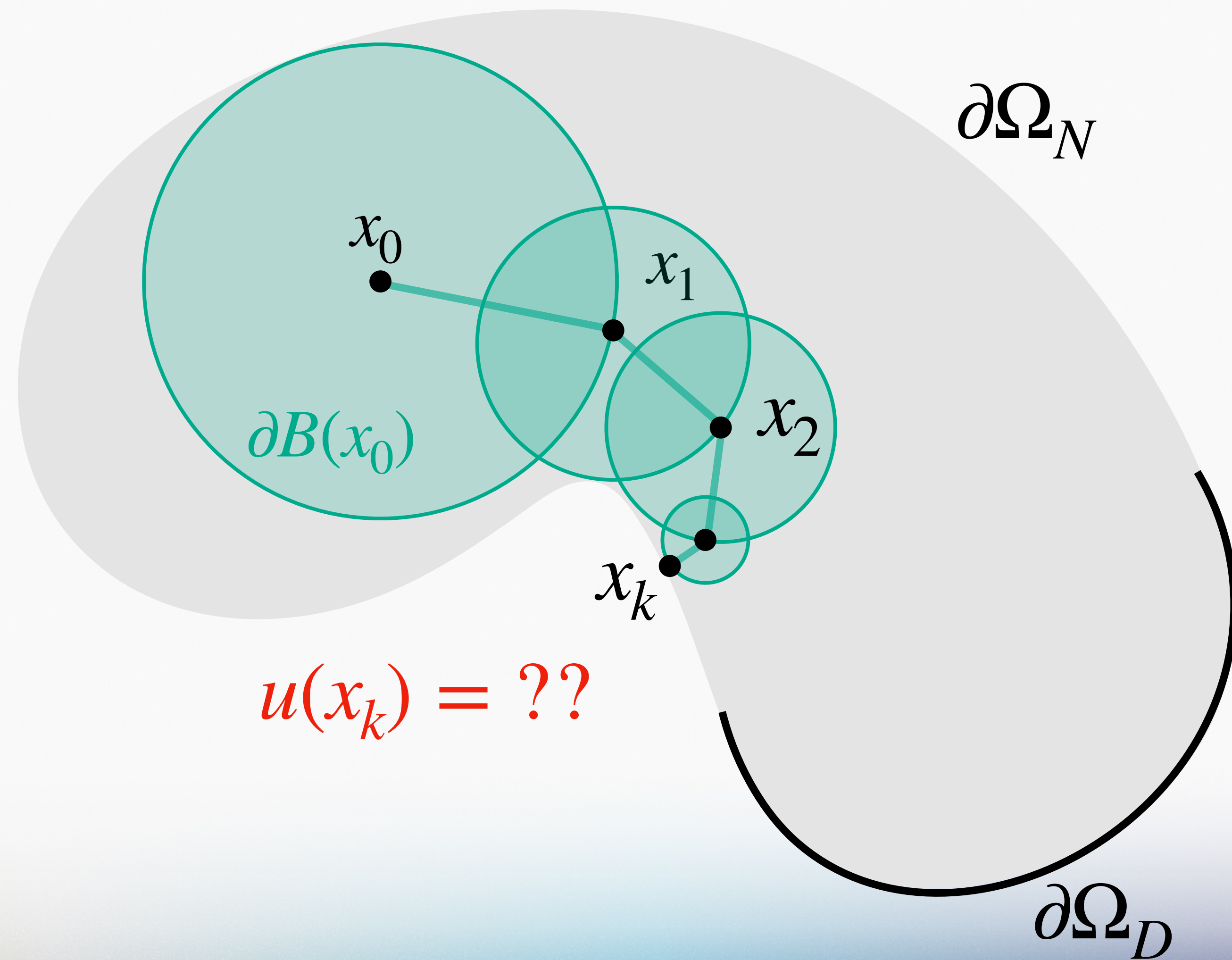


A walk terminates once solution can be approximated with boundary data



# Neumann problem

$$\begin{array}{ll} \Delta u = 0 & \text{on } \Omega \\ u = g & \text{on } \partial\Omega_D \quad \text{---} \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_N \quad \text{= =} \end{array}$$



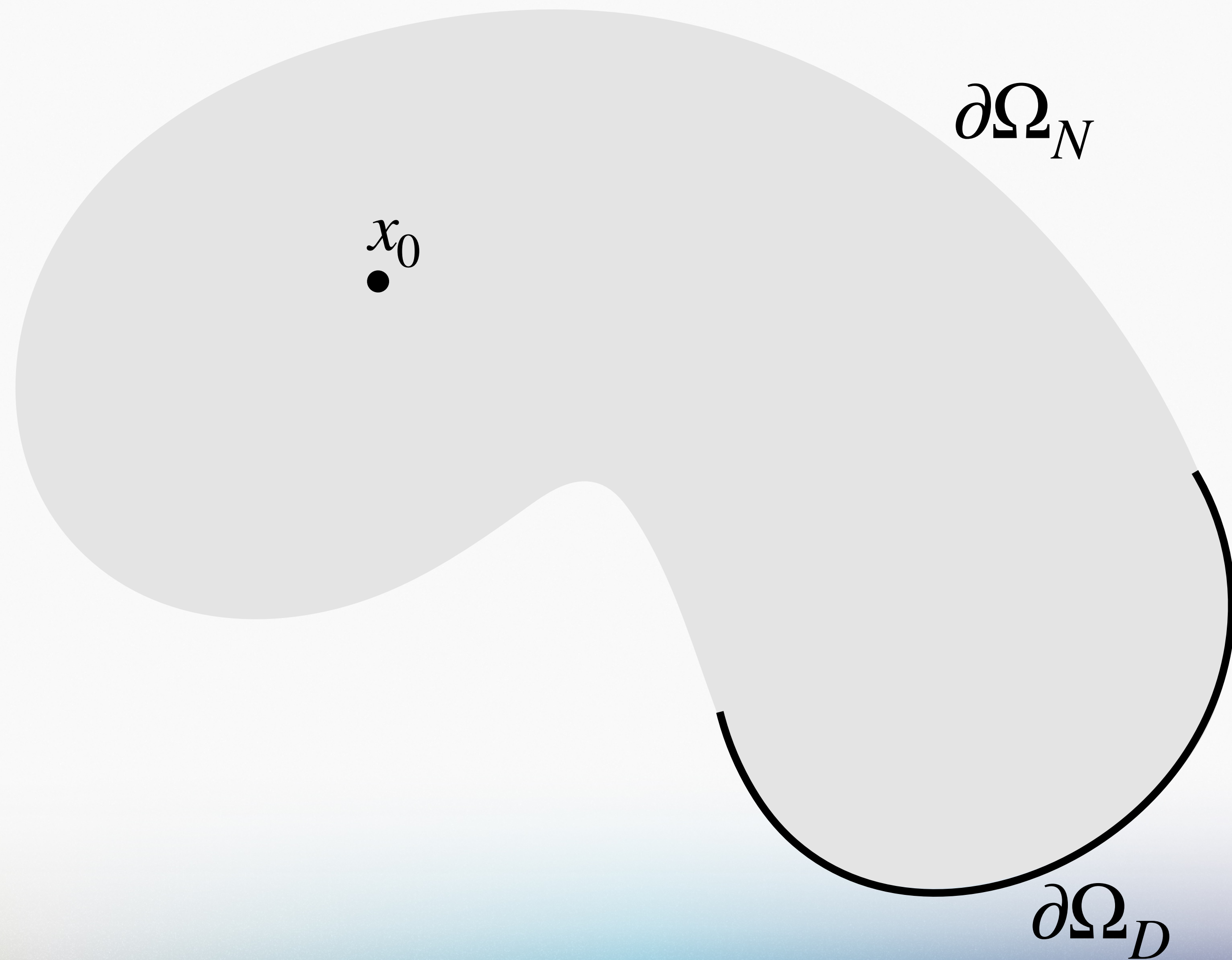


generalized mean value integral

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

**Monte Carlo estimator**

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$



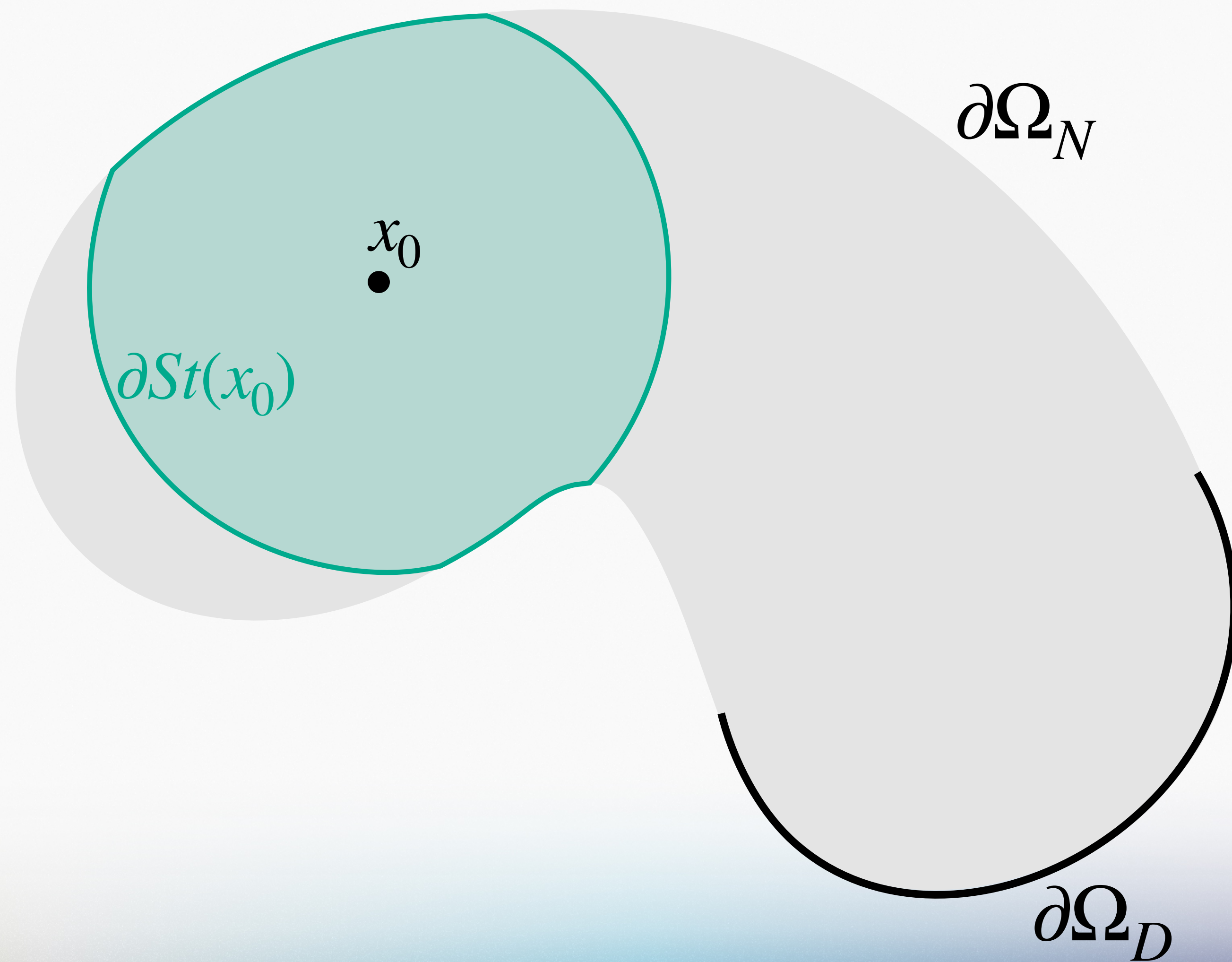


generalized mean value integral

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

**Monte Carlo estimator**

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

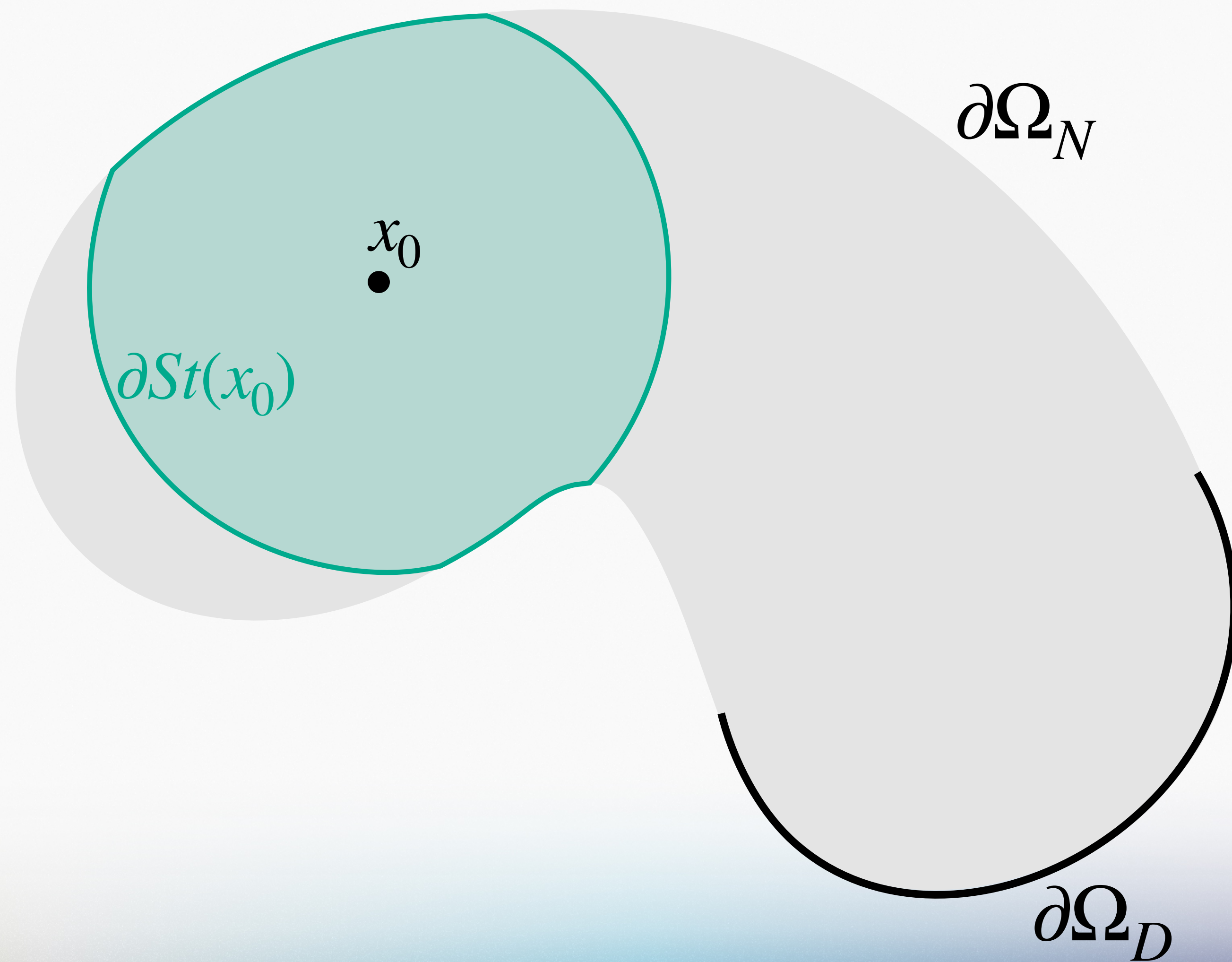
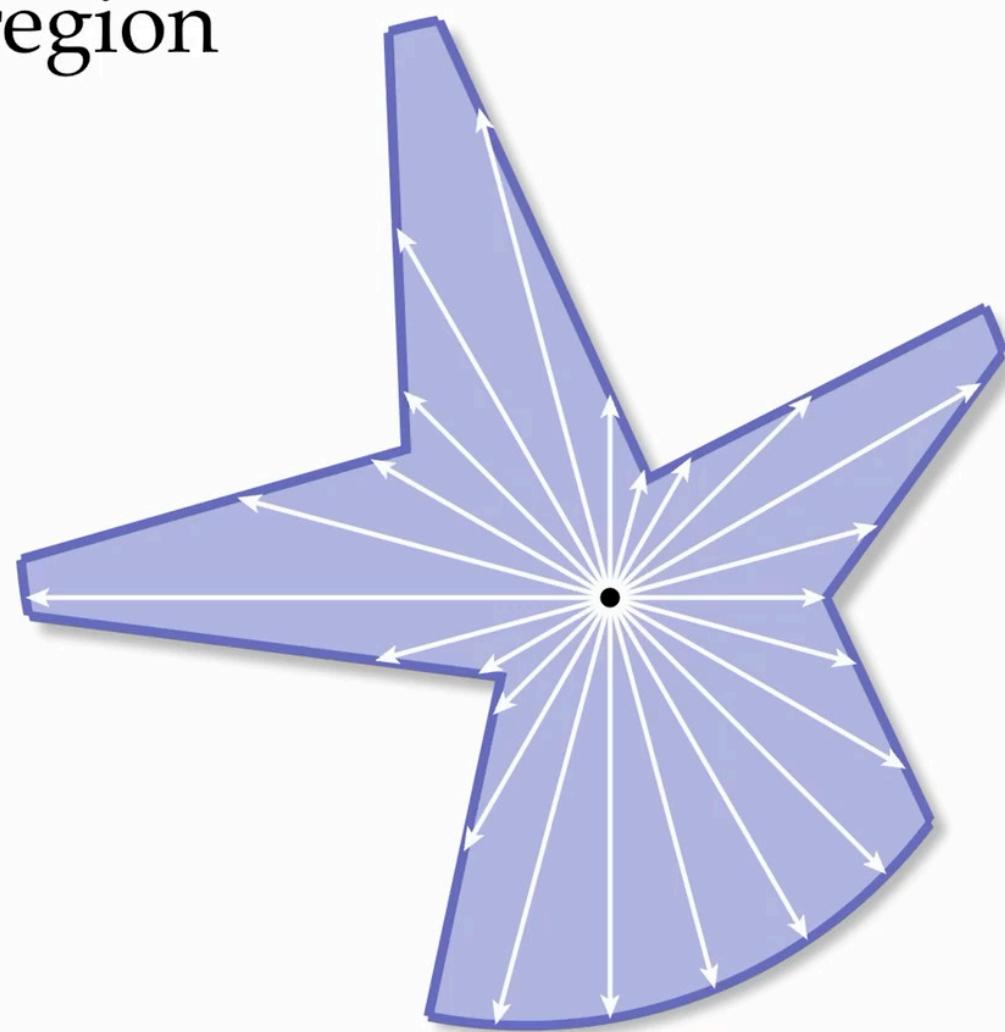




generalized mean value integral

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

star-shaped region



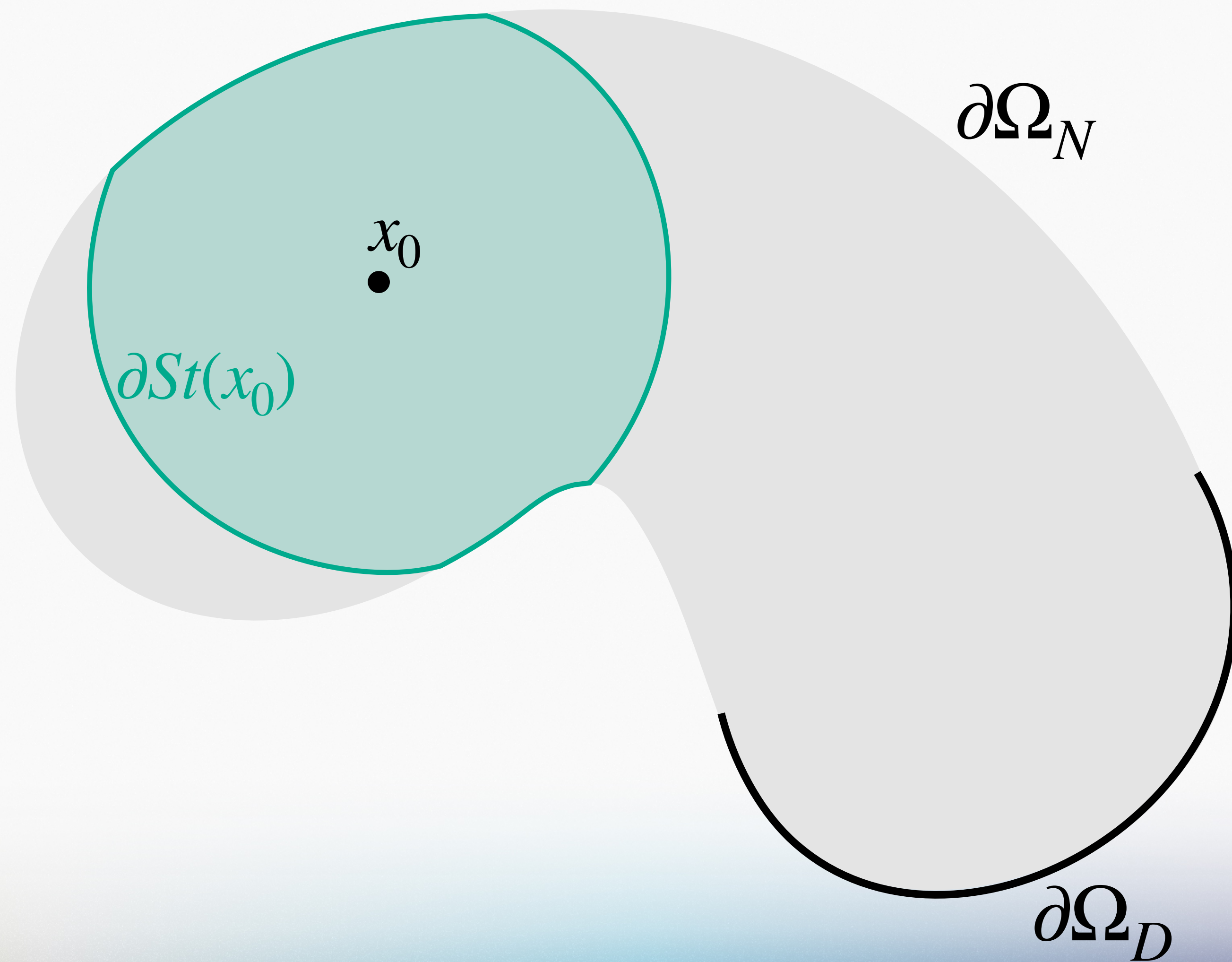


generalized mean value integral

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

**Monte Carlo estimator**

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$





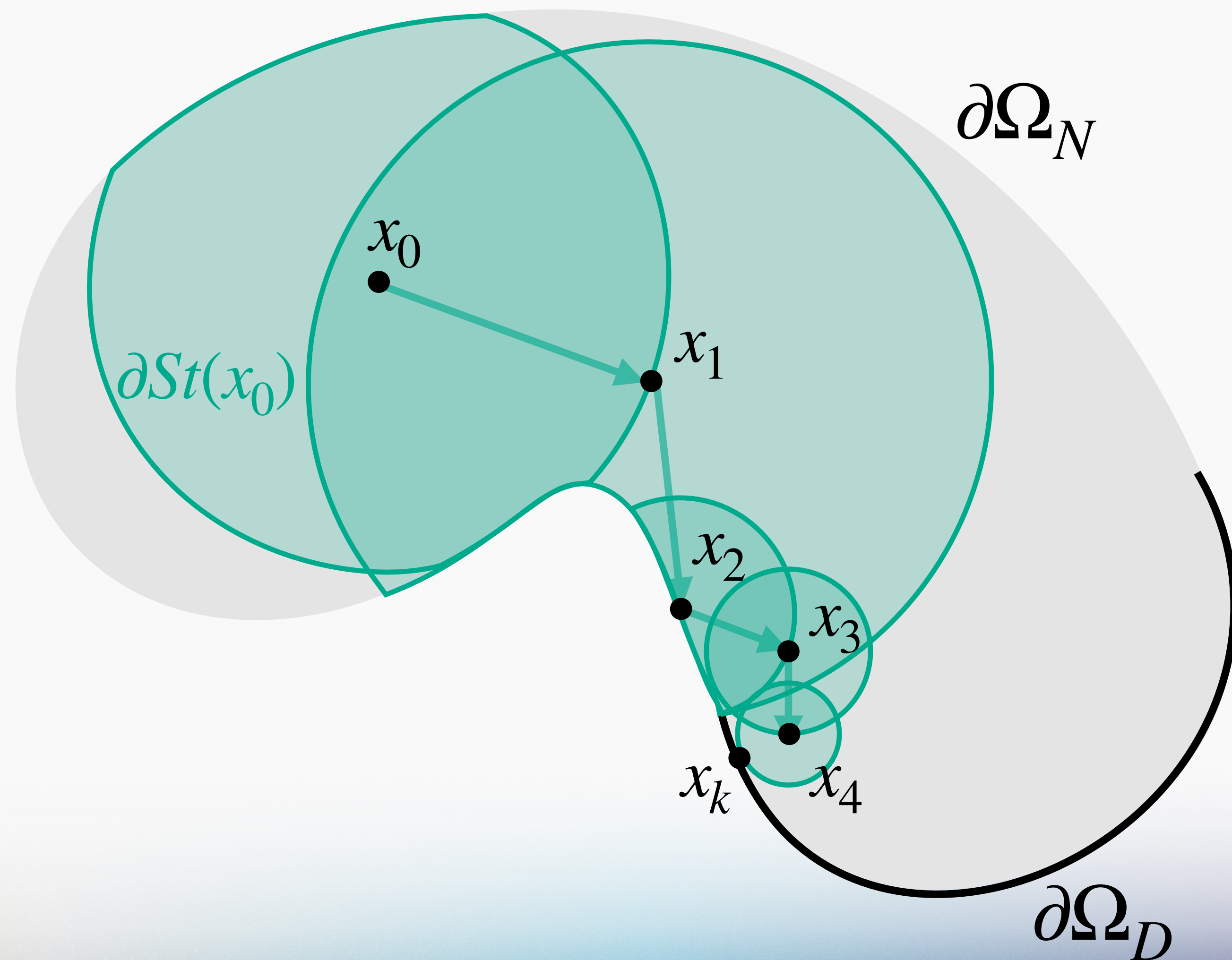
generalized mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform direction sample  $y \sim |P(x, y)|$





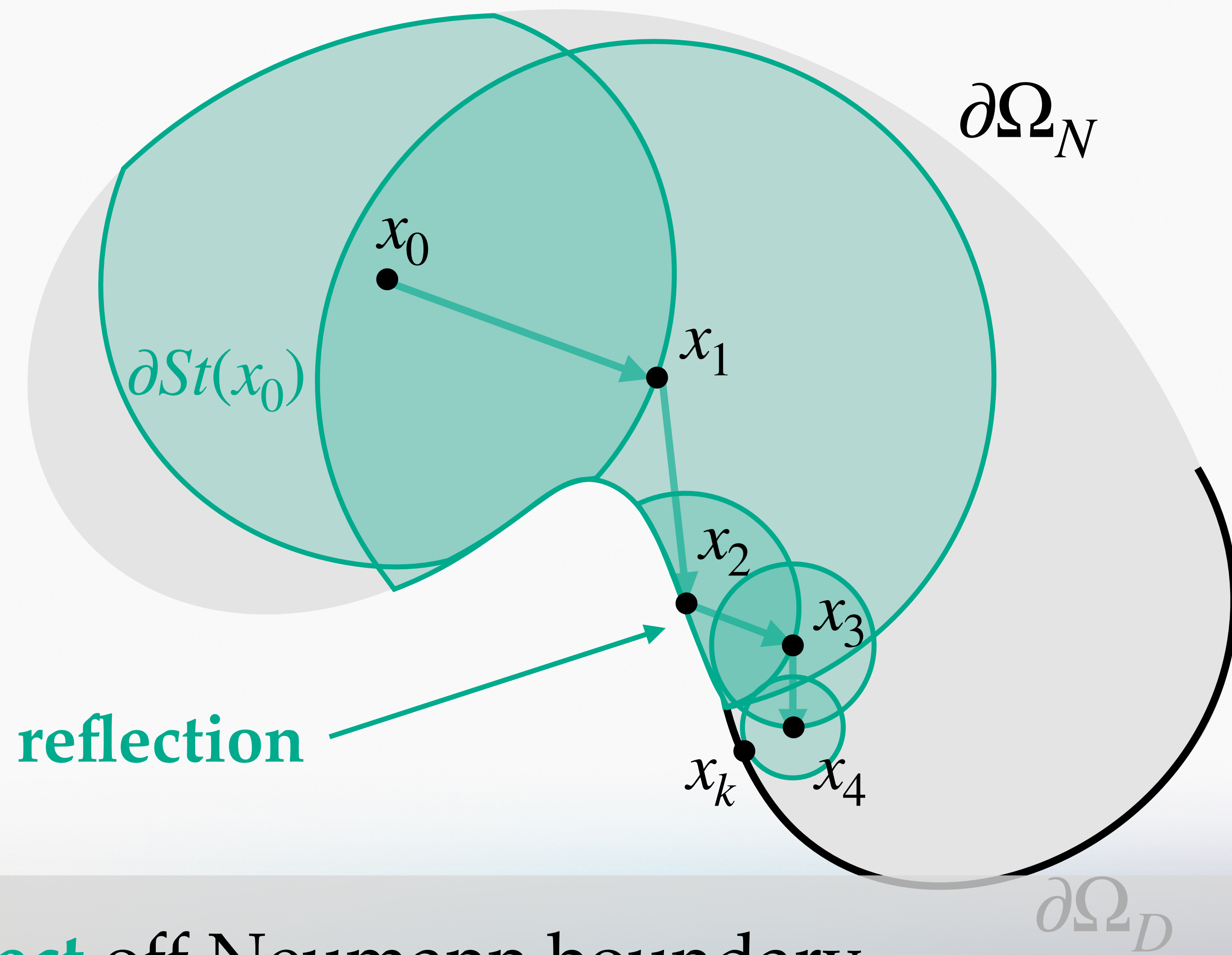
generalized mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform direction sample  $y \sim |P(x, y)|$



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



# avoiding multiple intersections

generalized mean value integral

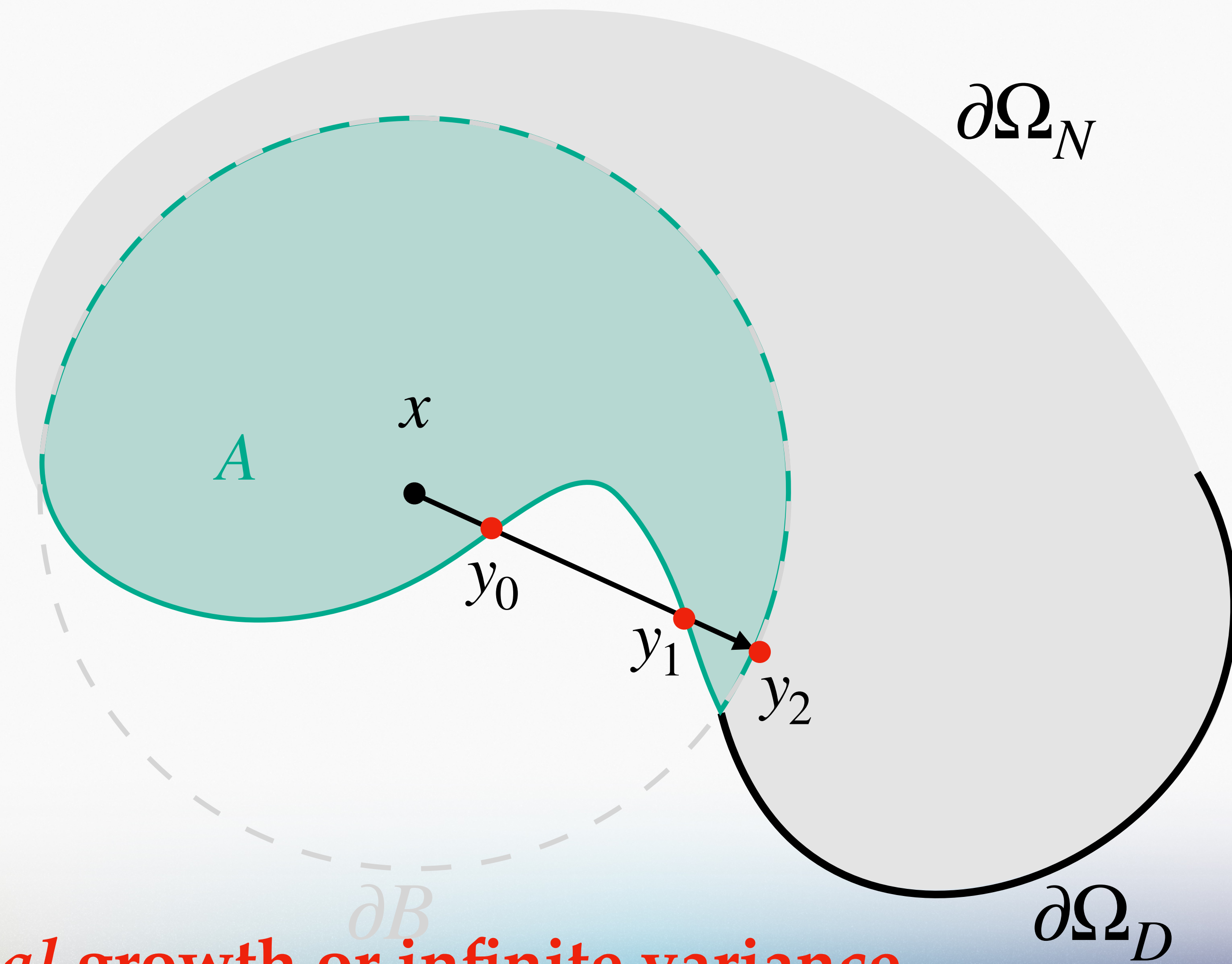
$$u(x) = \int_{\partial A} P^B(x, y) u(y) dy$$

Monte Carlo estimator

$$\hat{u}(x) = \sum_{i=1}^n \frac{P(x, y_i)}{|P(x, y_i)|} \hat{u}(y_i)$$

uniform direction sample  $y \sim |P(x, y)|$

multiple intersections  $\implies$  exponential growth or infinite variance





# walk on stars [Sawhney et al. 2023]

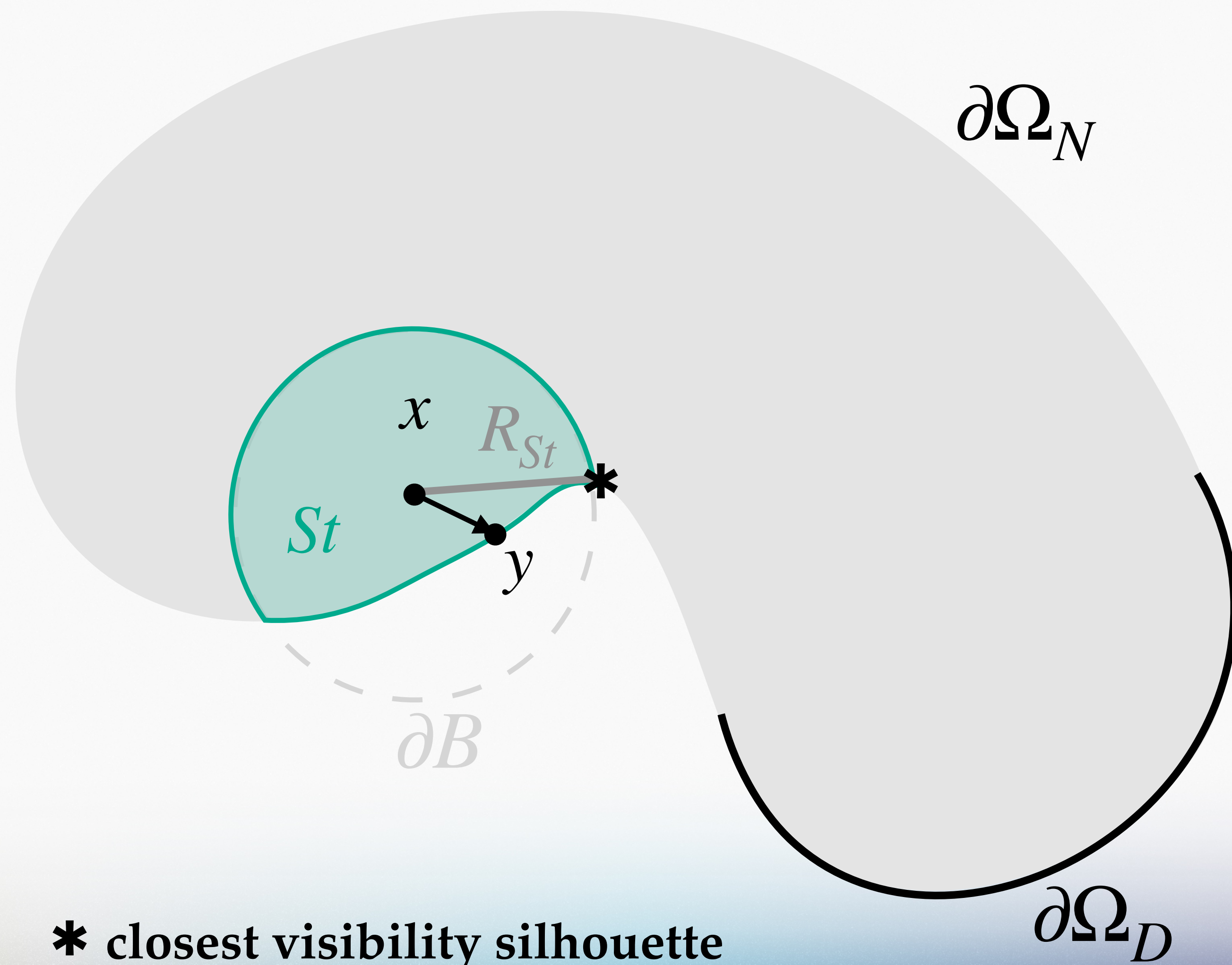
generalized mean value integral

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

Monte Carlo estimator

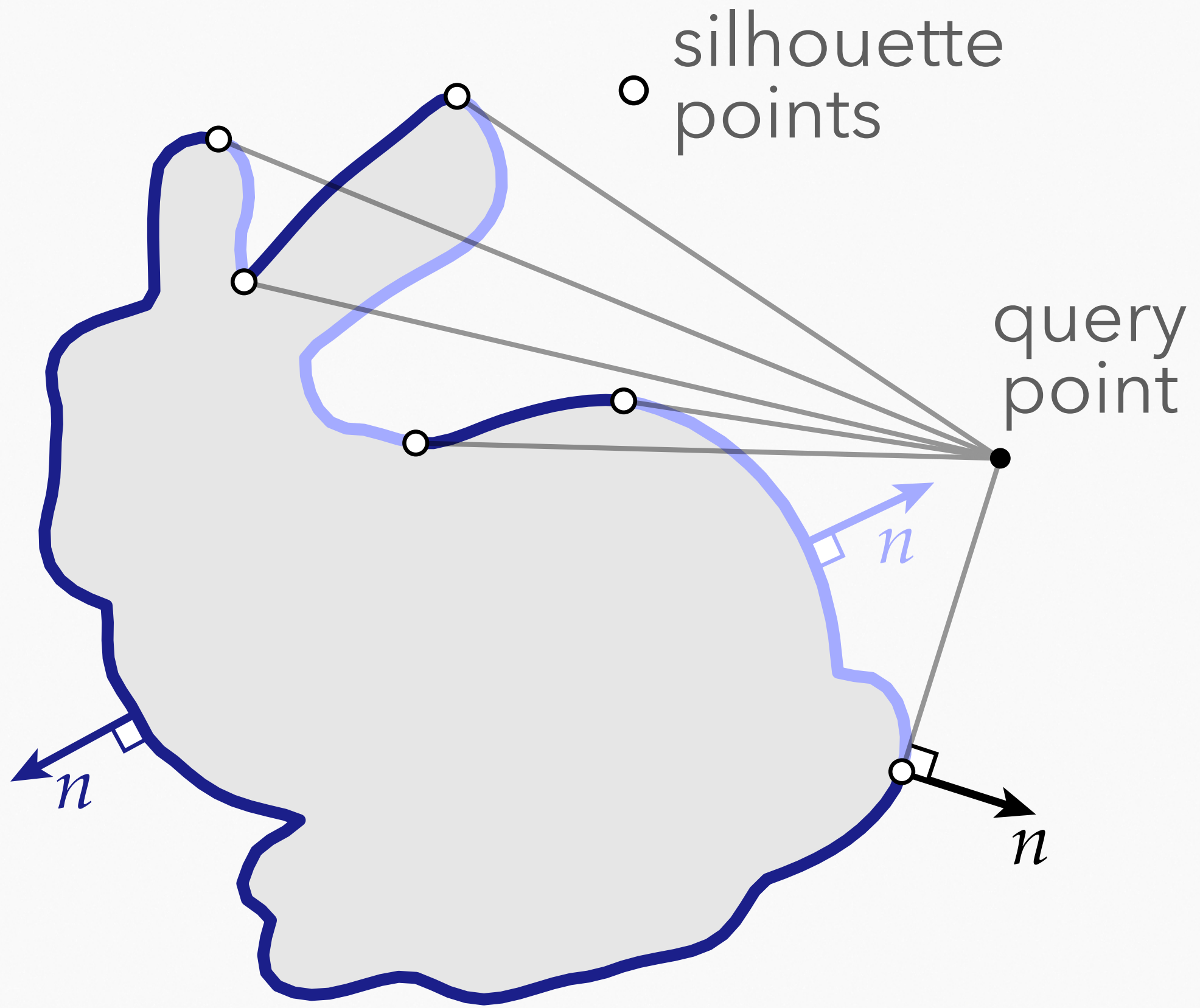
$$\hat{u}(x) = \hat{u}(y)$$

uniform direction sample  $y \sim |P(x, y)|$

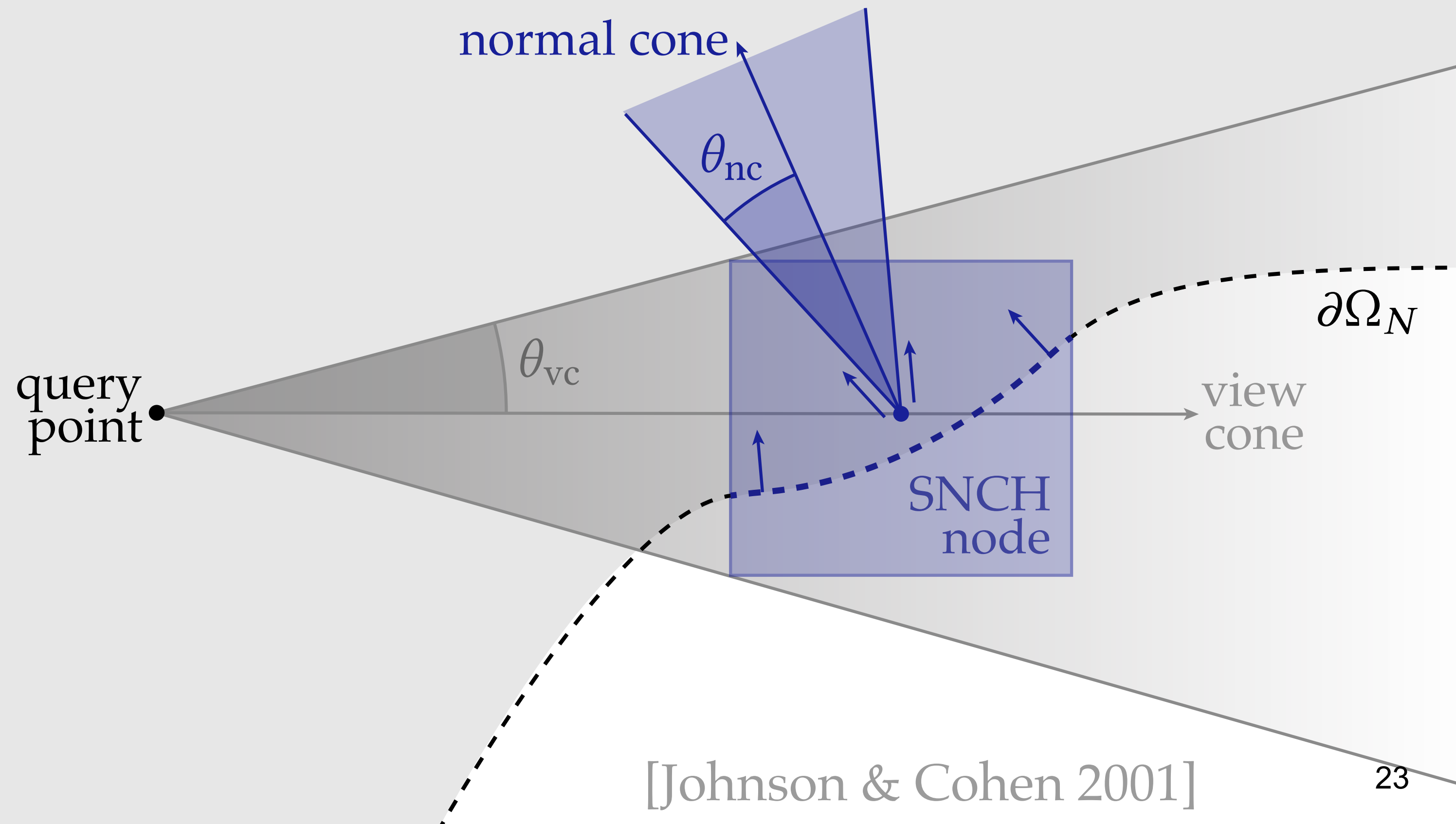




# closest silhouette points



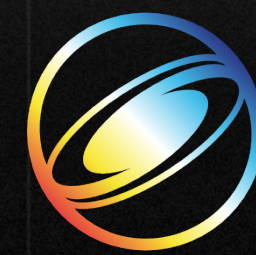
## normal cone hierarchy





SIGGRAPH

2024



**SIGGRAPH 2024**

DENVER+ 28 JUL – 1 AUG

# GENERALIZING WALK ON STARS





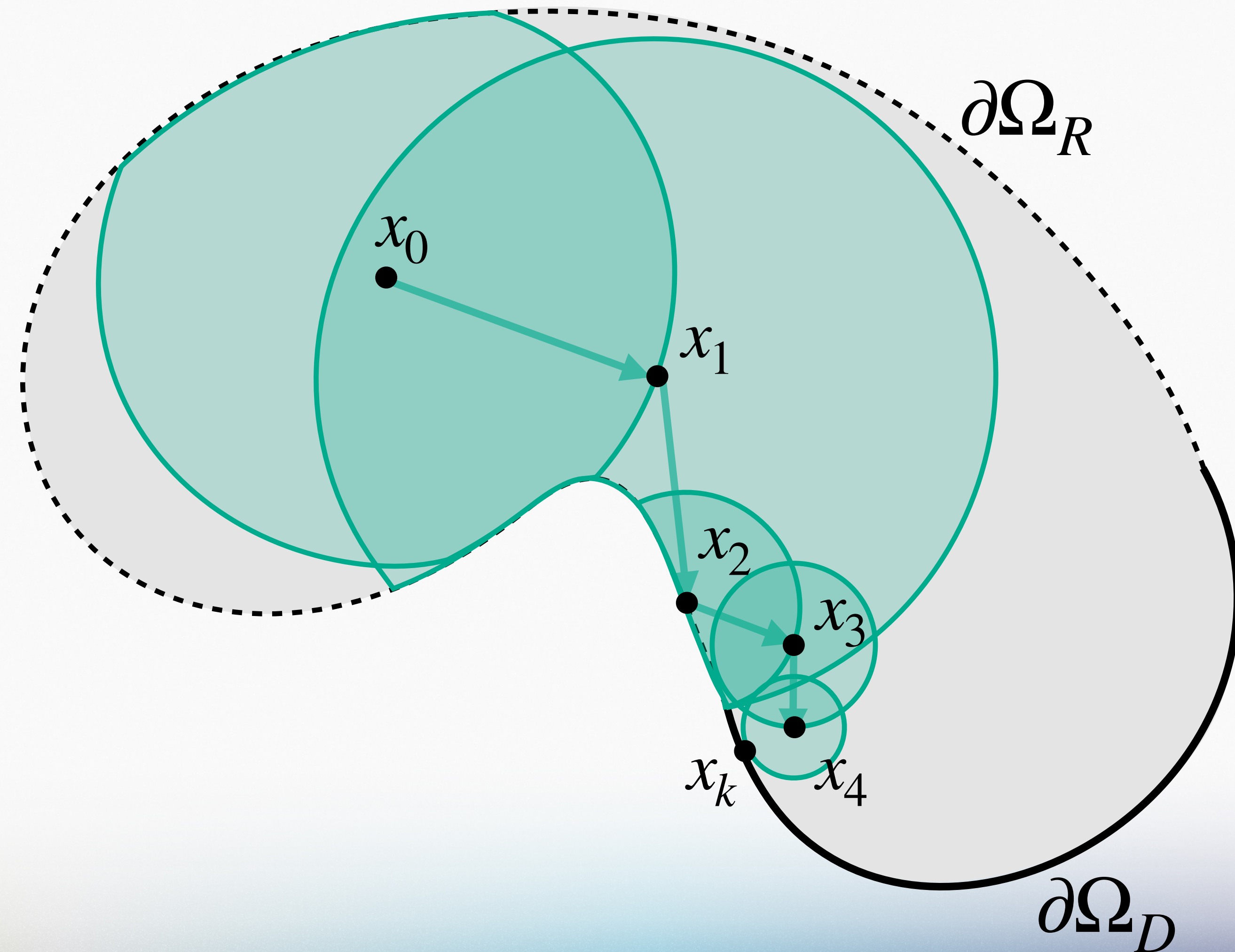
# Robin problem

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

$$u = g \quad \text{on } \partial\Omega_D \quad \text{---}$$

$$\frac{\partial u}{\partial n} - \mu u = 0 \quad \text{on } \partial\Omega_R \quad \text{- - -}$$

$$\mu : \partial\Omega \rightarrow \mathbb{R}_{>0}$$





# generalized mean value with dampening

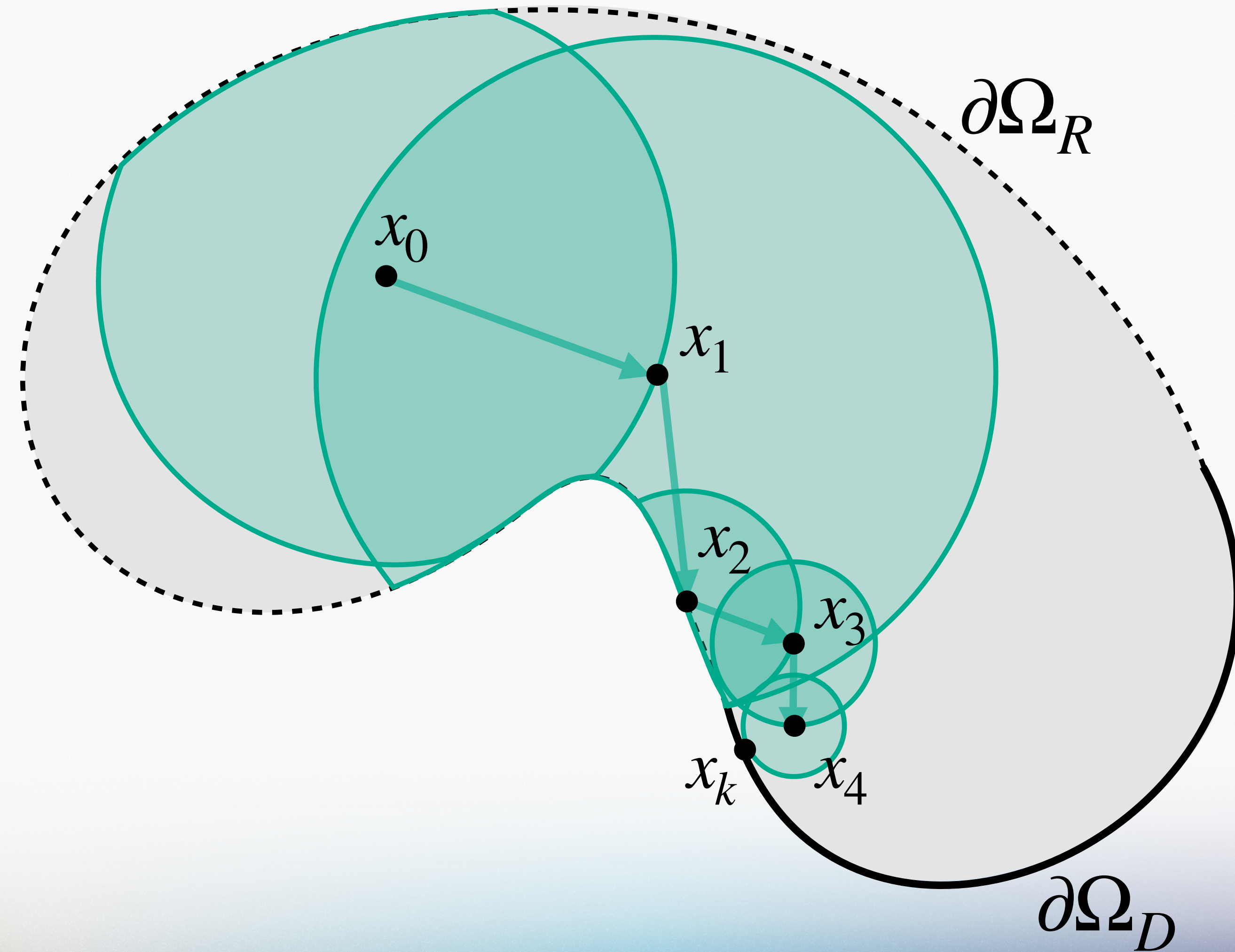
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

no change from walk on stars

**Monte Carlo estimator**

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_{\epsilon} \\ \rho_{\mu}(x, y) \hat{u}(y), & \text{otherwise} \end{cases}$$





# generalized mean value with dampening

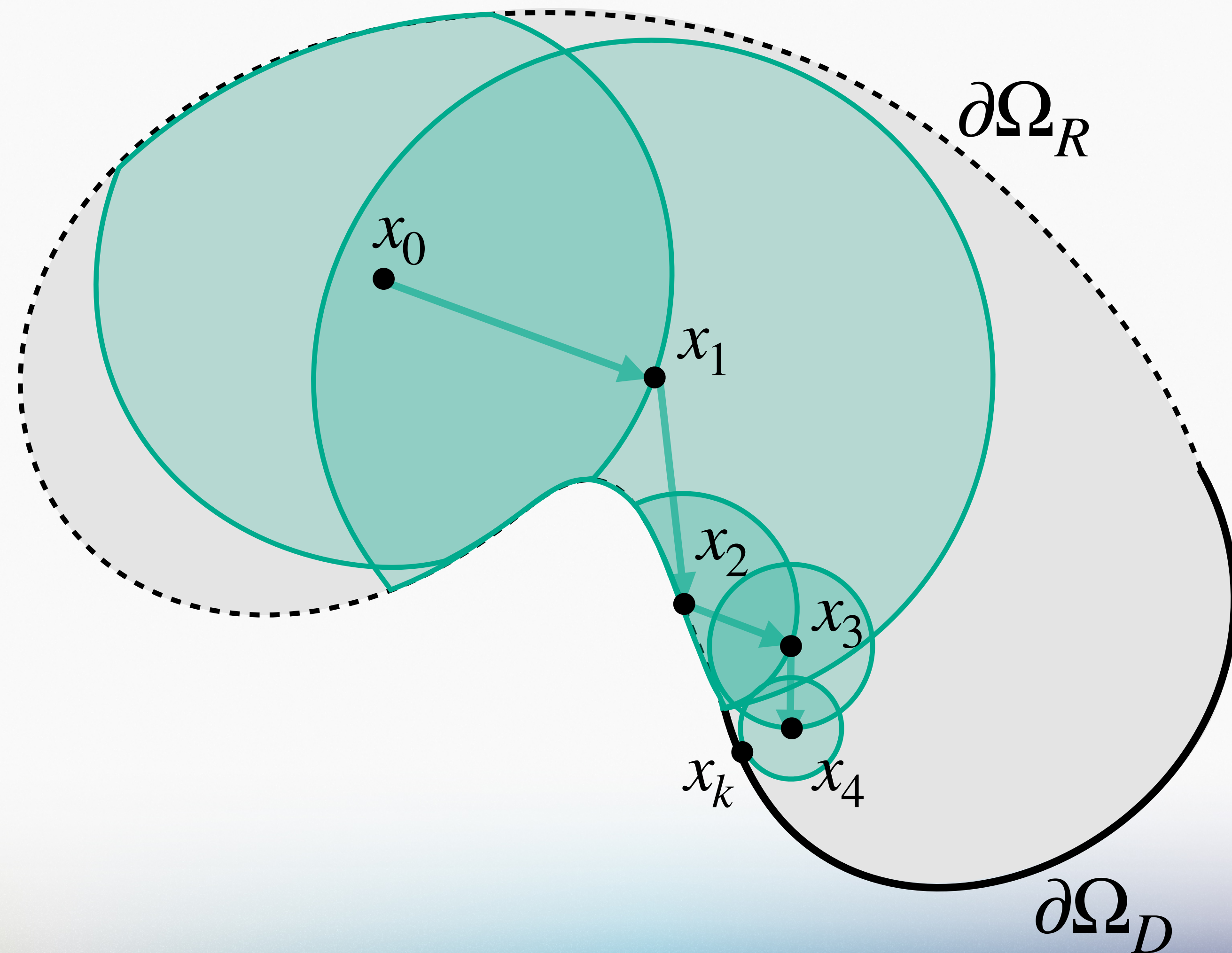
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

reflectance no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_{\epsilon} \\ \rho_{\mu}(x, y) \hat{u}(y), & \text{otherwise} \end{cases}$$





# generalized mean value with dampening

generalized boundary integral

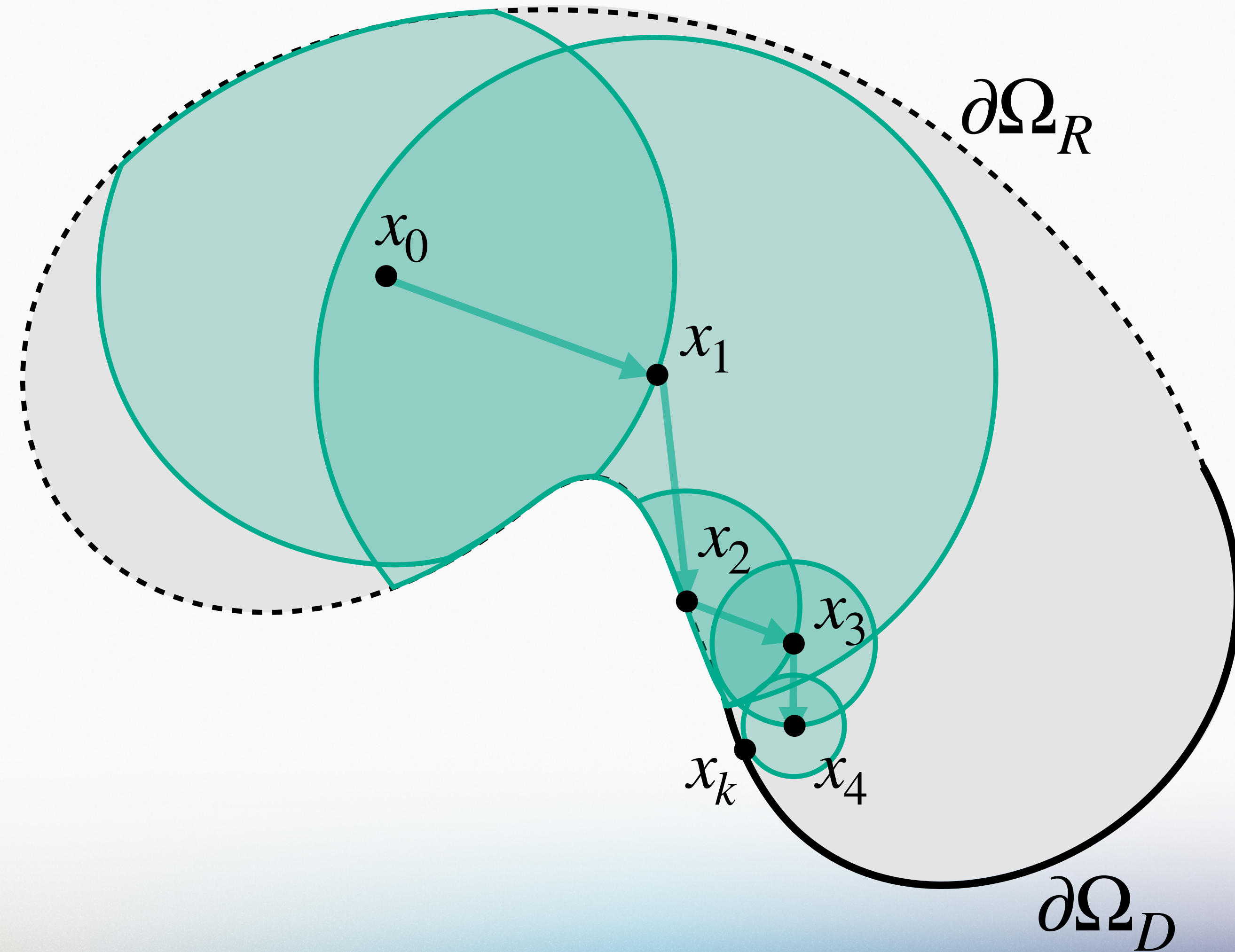
$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

reflectance no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

reflectance modulates path throughput





# generalized mean value with dampening

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

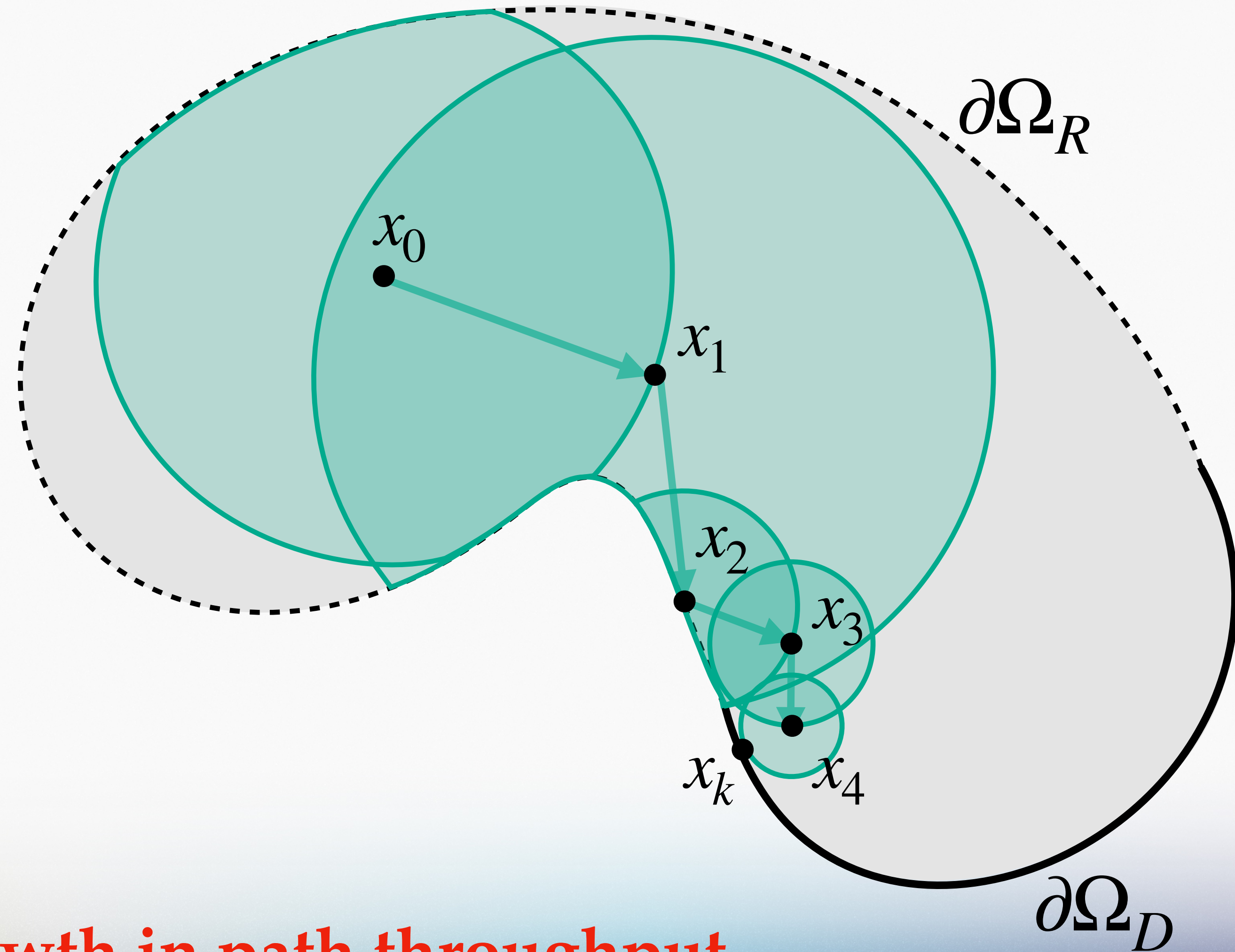
reflectance no change from walk on stars

## Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

reflectance modulates path throughput

$\rho_{\mu}(x, y) \notin [0, 1] \implies$  exponential growth in path throughput





# keeping reflectance bounded

Neumann  $\mu = 0$

Robin

Dirichlet  $\mu = \infty$



\* closest visibility silhouette

$$\rho_{\mu}(x, y) \notin [0, 1]$$

o closest point on boundary

intuition: interpolate between WoSt and WoS



# keeping reflectance bounded

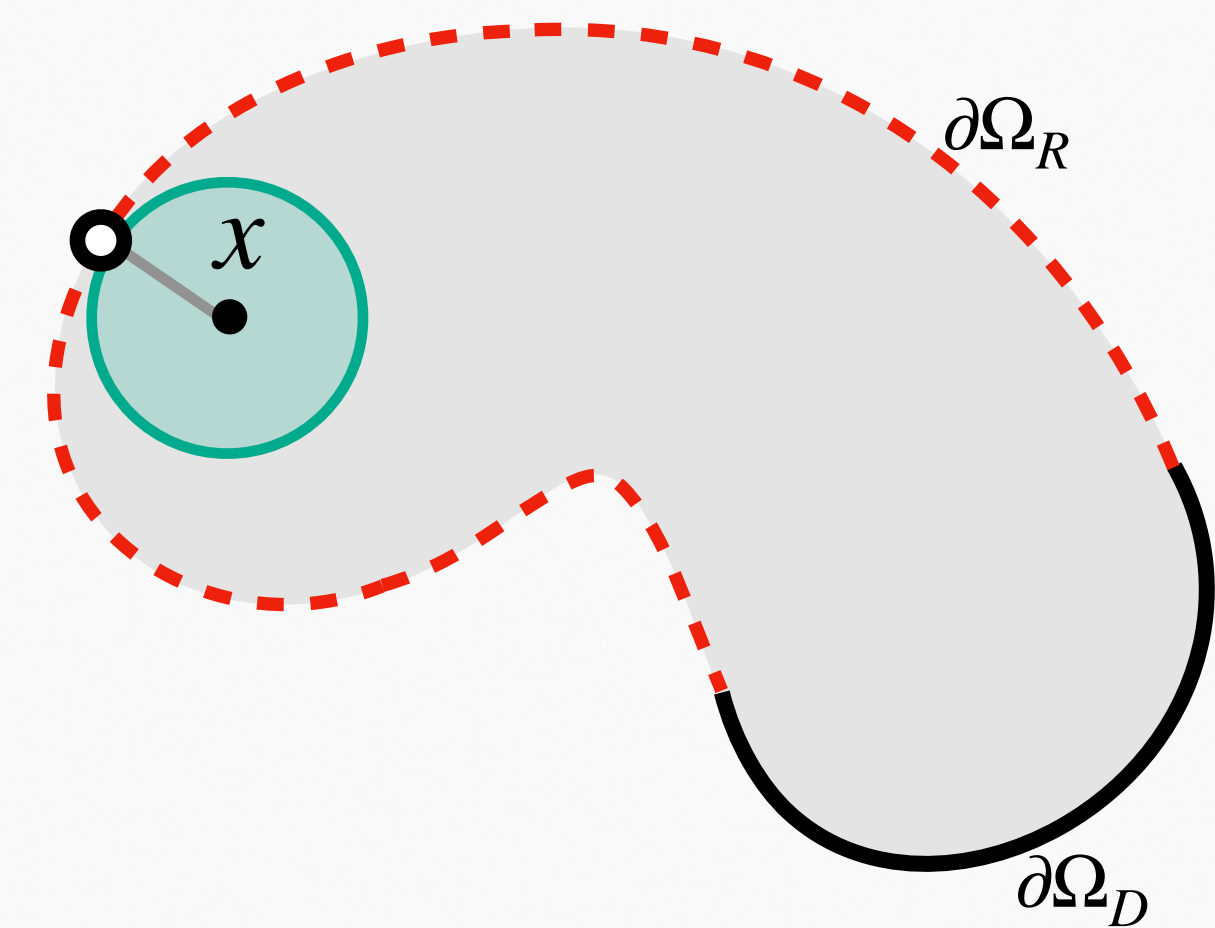
Neumann  $\mu = 0$

Robin

Dirichlet  $\mu = \infty$



Walk on Spheres



\* closest visibility silhouette

$$\rho_\mu(x, y) \notin [0, 1]$$

○ closest point on boundary

intuition: interpolate between WoSt and WoS



# keeping reflectance bounded

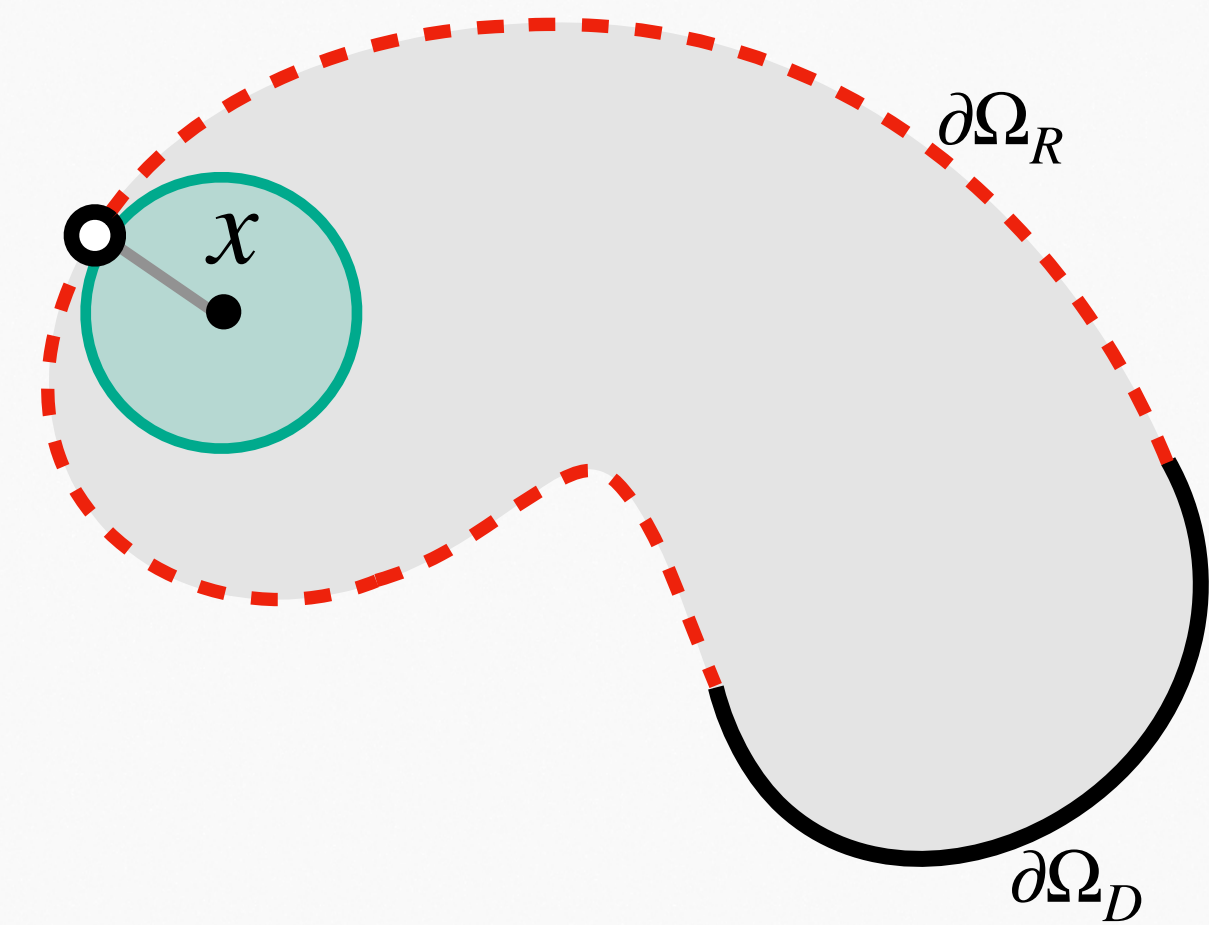
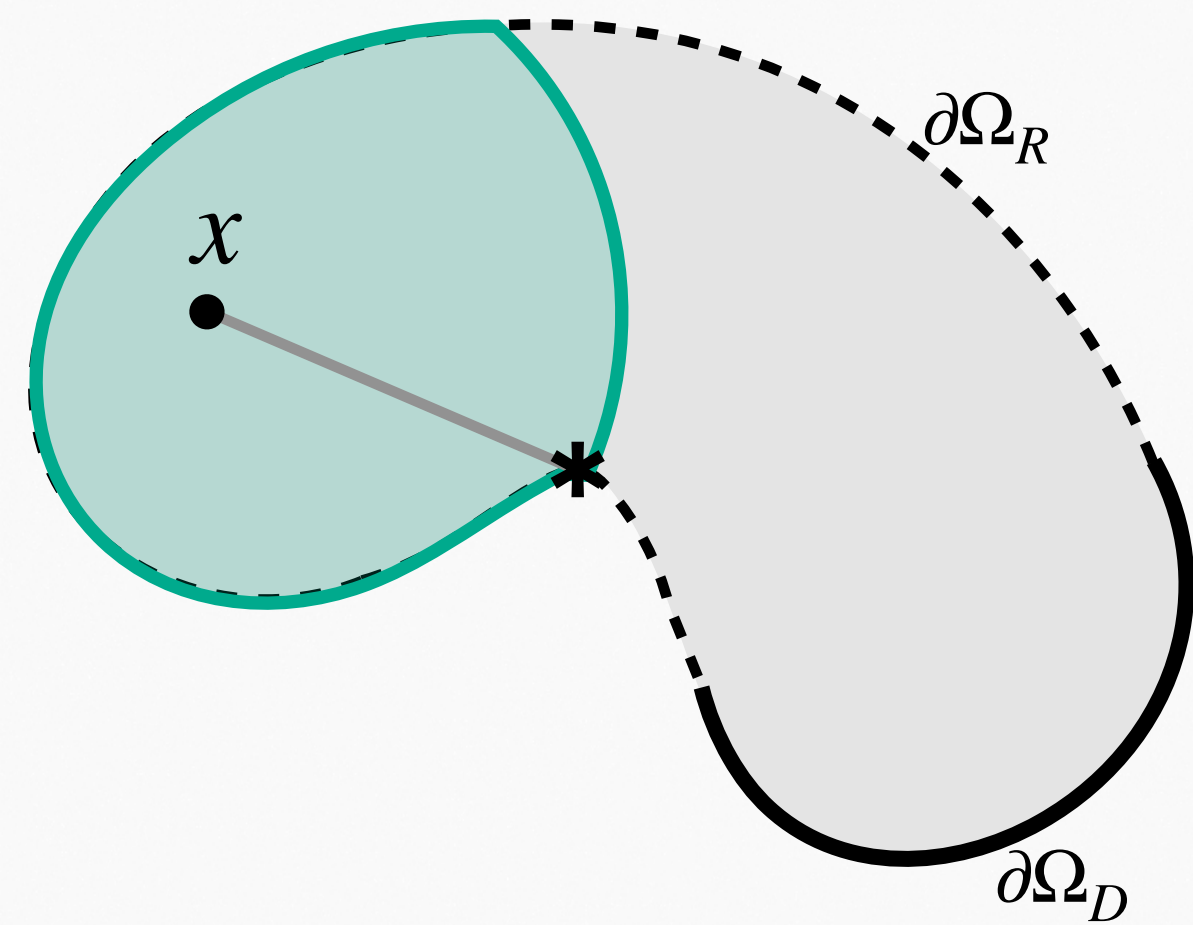
Neumann  $\mu = 0$

Robin

Dirichlet  $\mu = \infty$

Walk on Stars

Walk on Spheres



\* closest visibility silhouette

$$\rho_\mu(x, y) \notin [0, 1]$$

o closest point on boundary

intuition: interpolate between WoSt and WoS



# keeping reflectance bounded

Neumann  $\mu = 0$

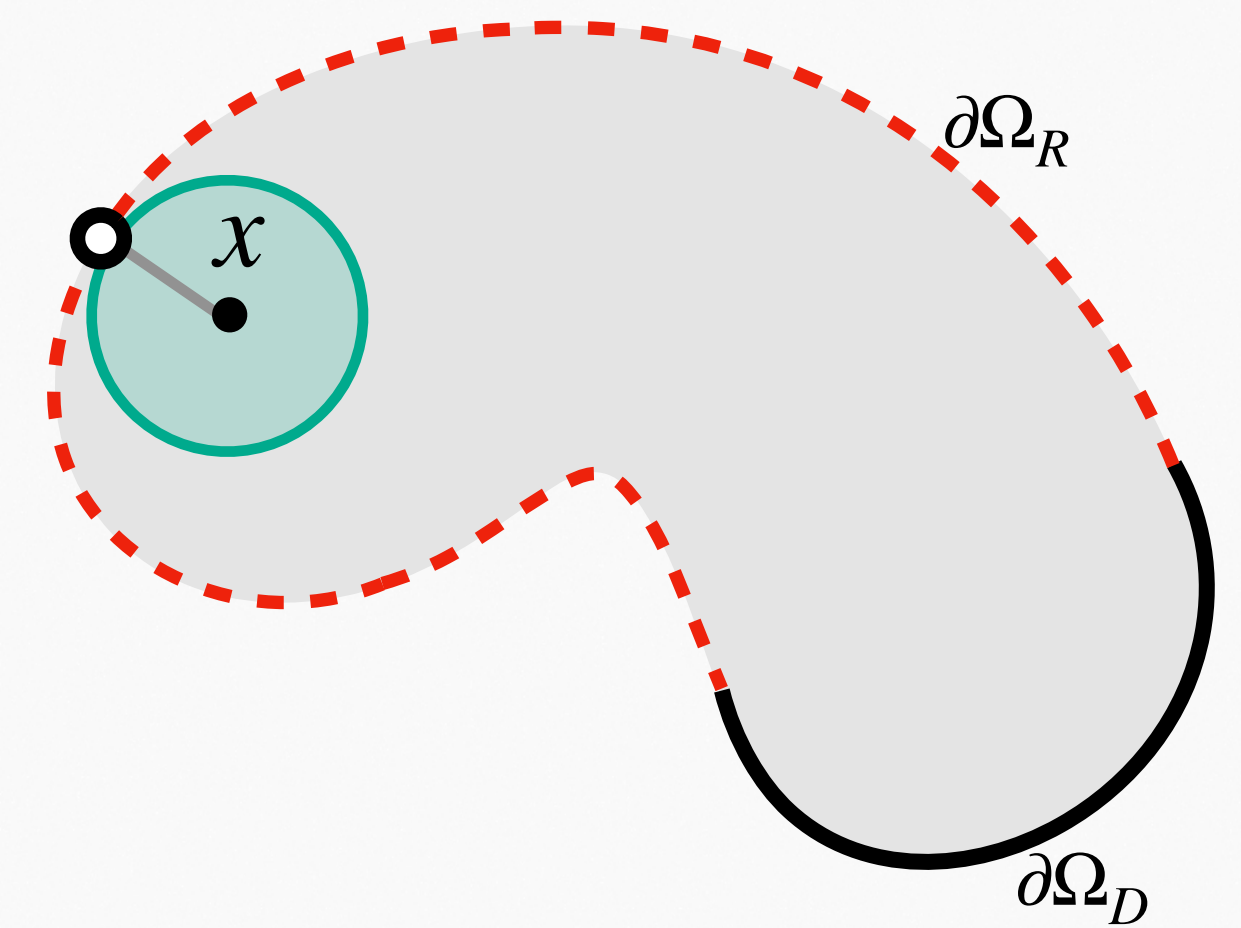
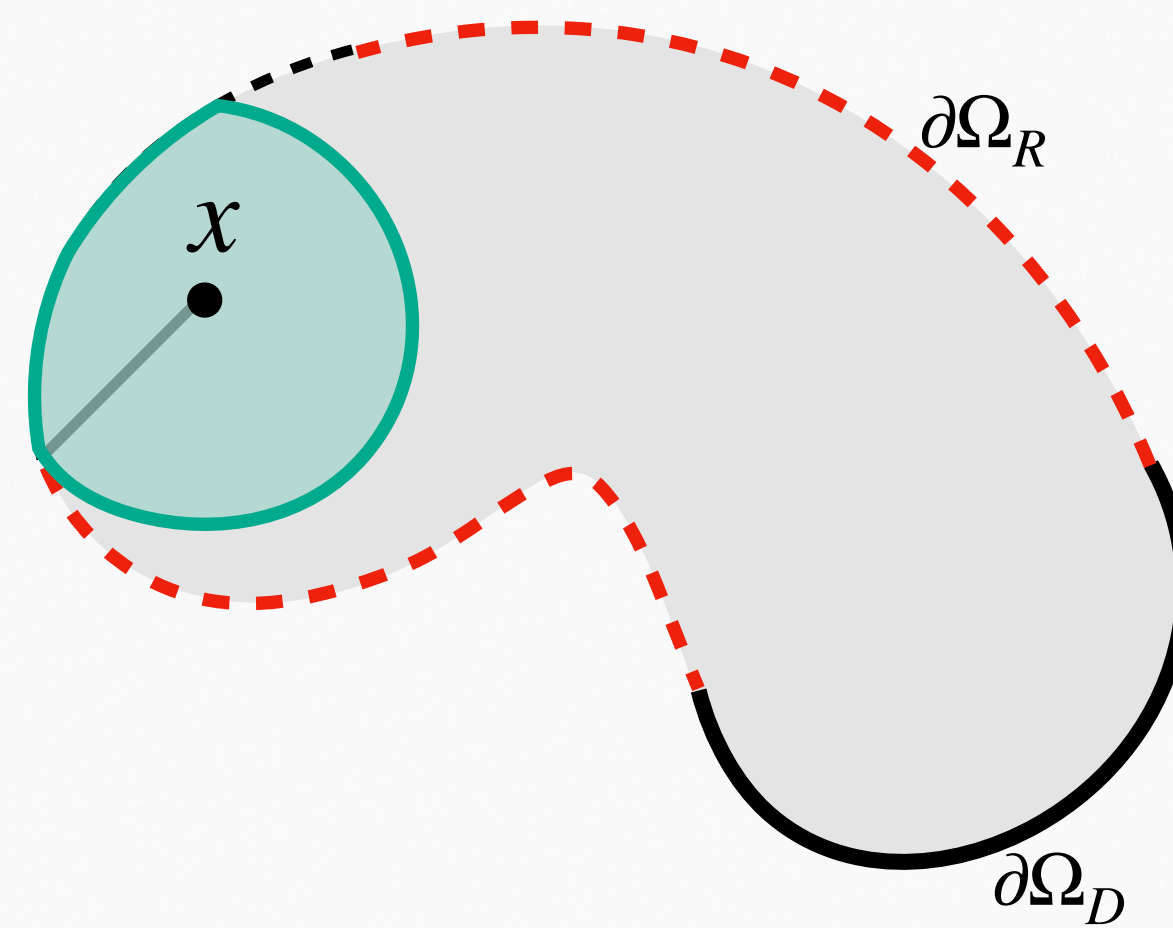
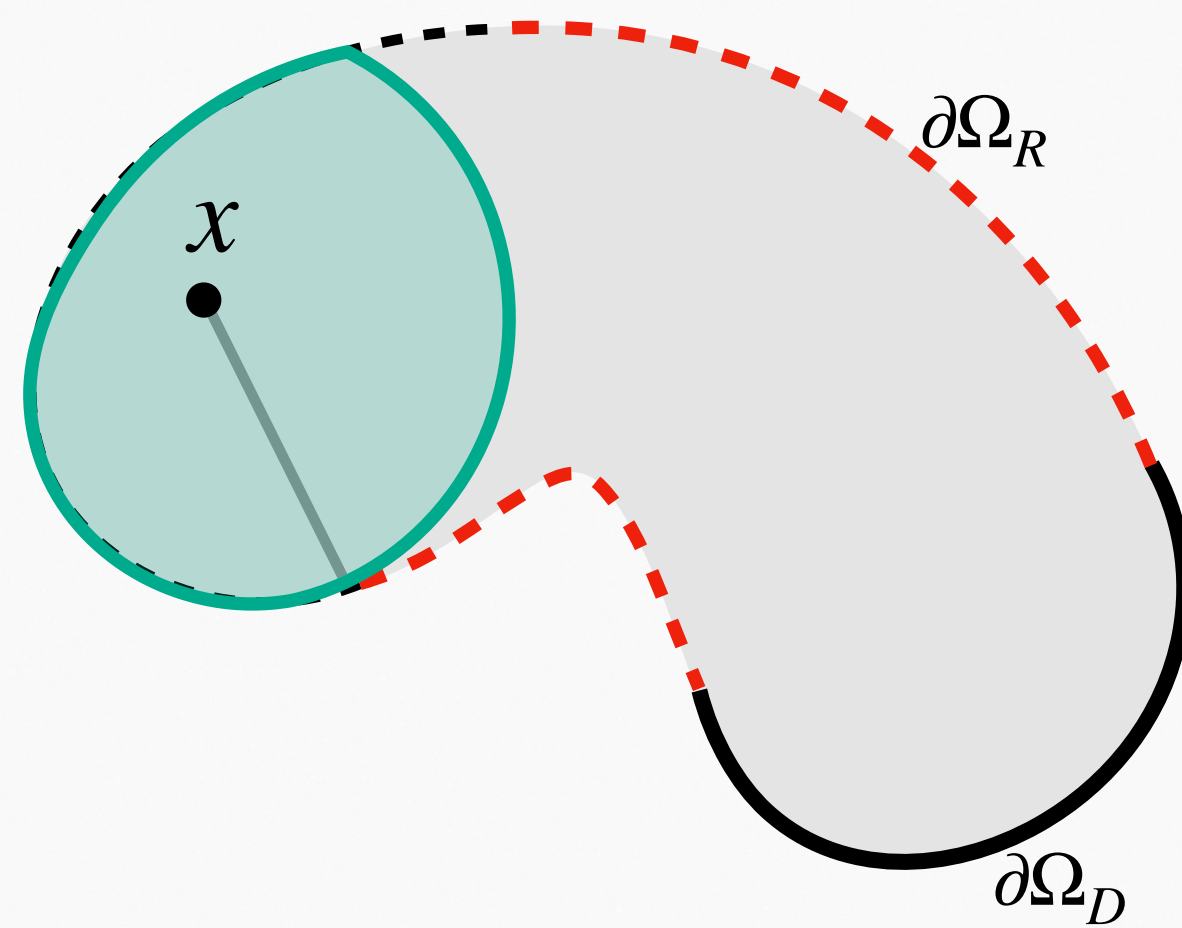
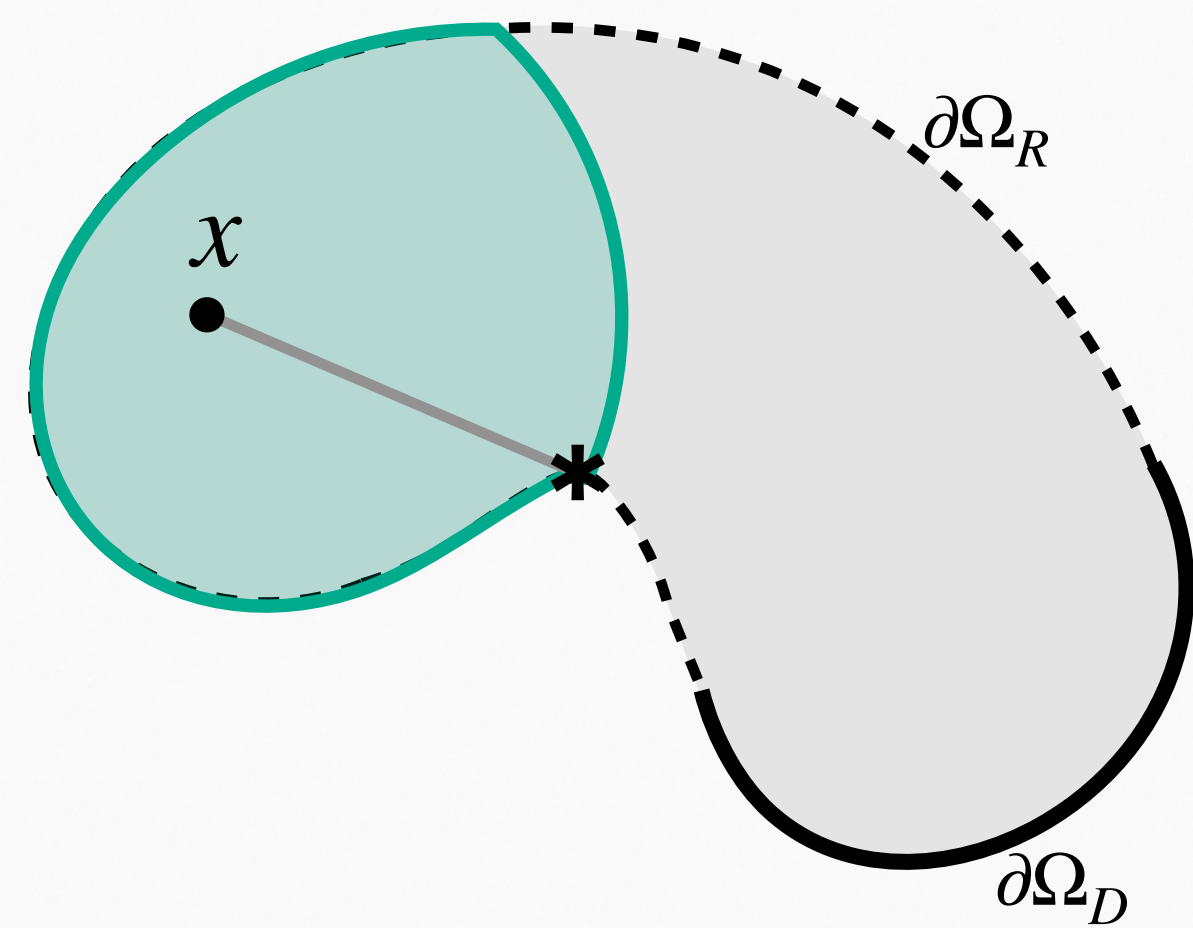
Robin

Dirichlet  $\mu = \infty$



Walk on Stars

Walk on Spheres



\* closest visibility silhouette

$$\rho_\mu(x, y) \notin [0, 1]$$

o closest point on boundary

intuition: interpolate between WoSt and WoS



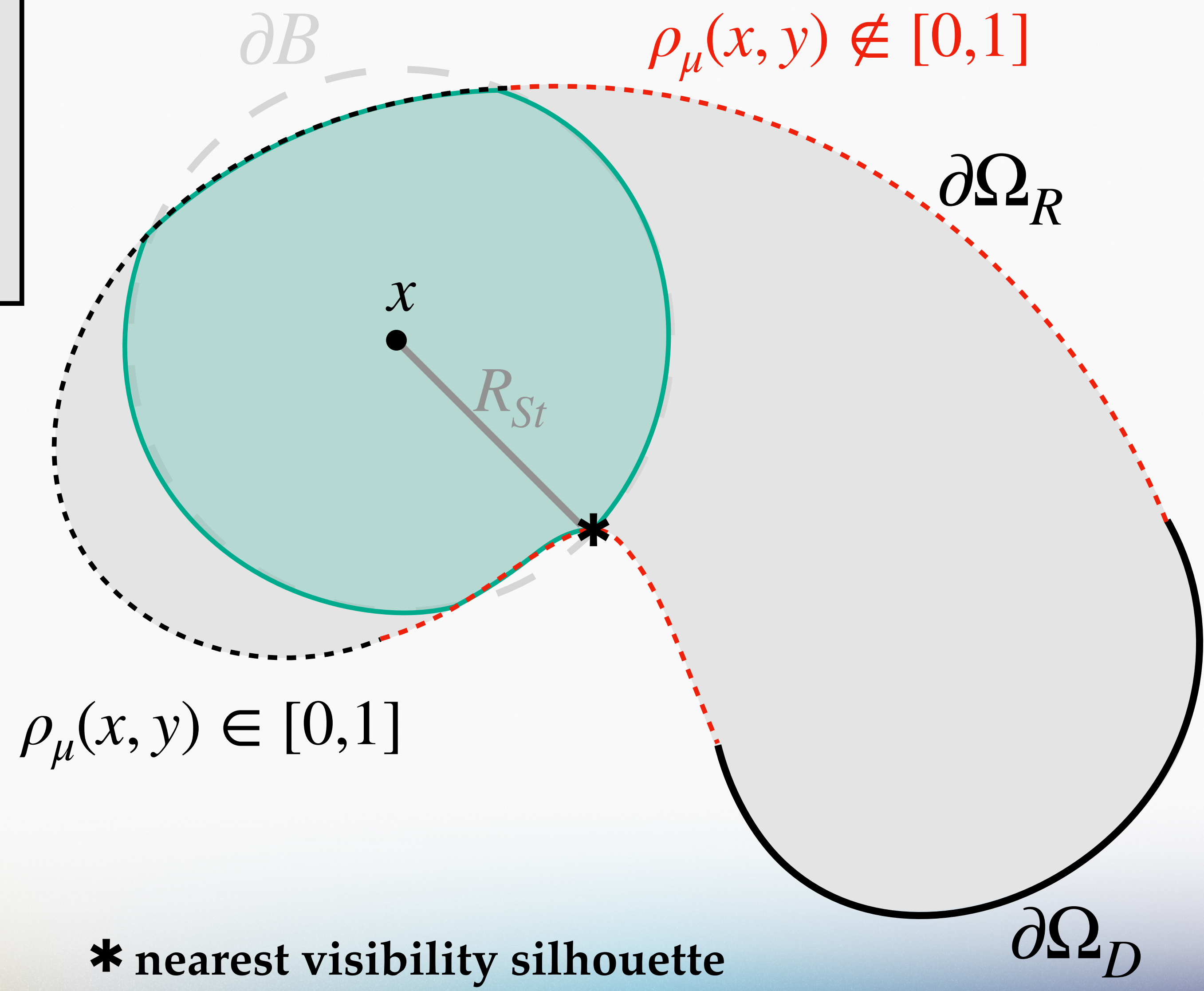
# keeping reflectance bounded

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

## Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$





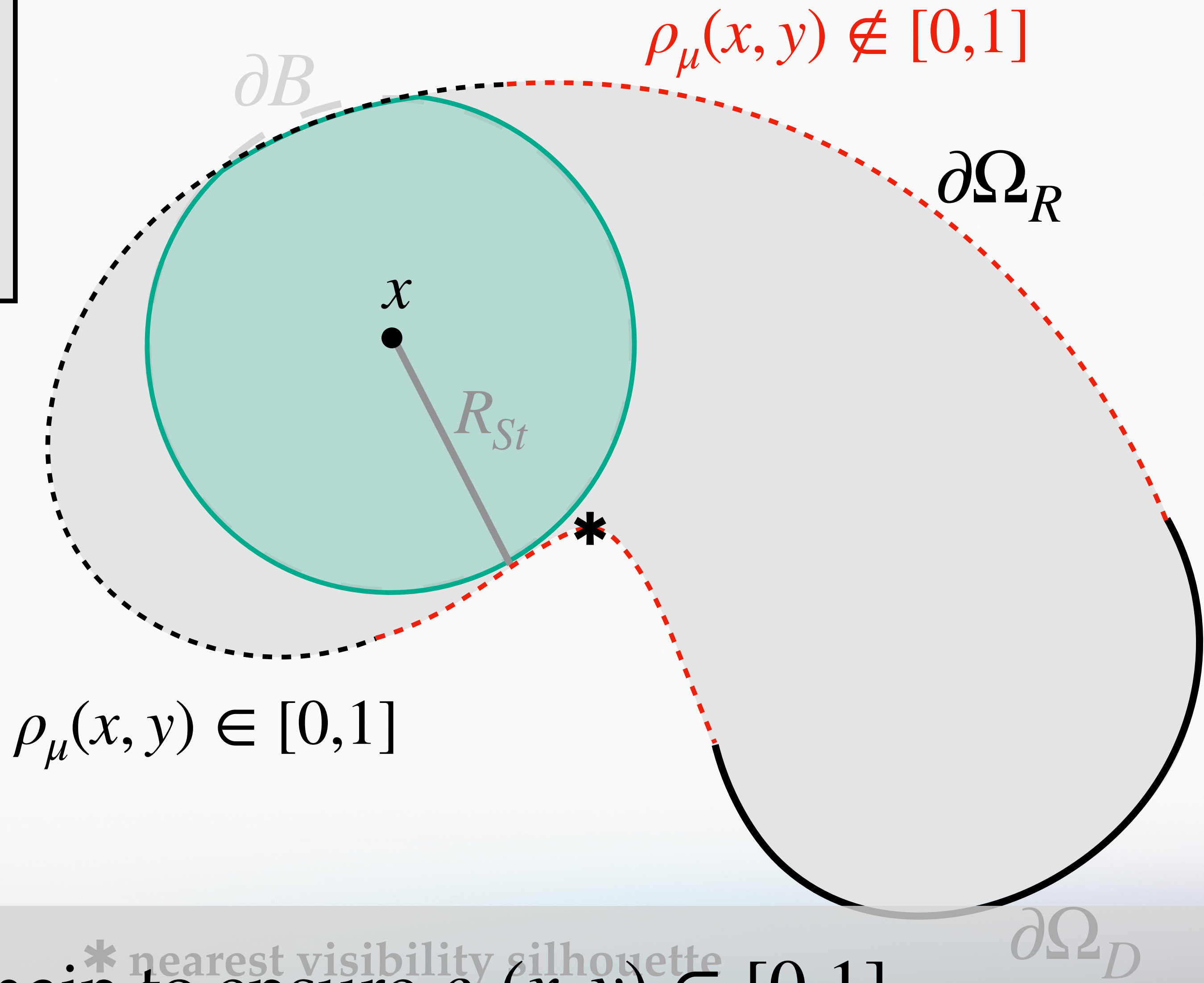
# keeping reflectance bounded

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

## Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$



key idea: shrink star-shaped domain to ensure  $\rho_{\mu}(x, y) \in [0, 1]$



# keeping reflectance bounded

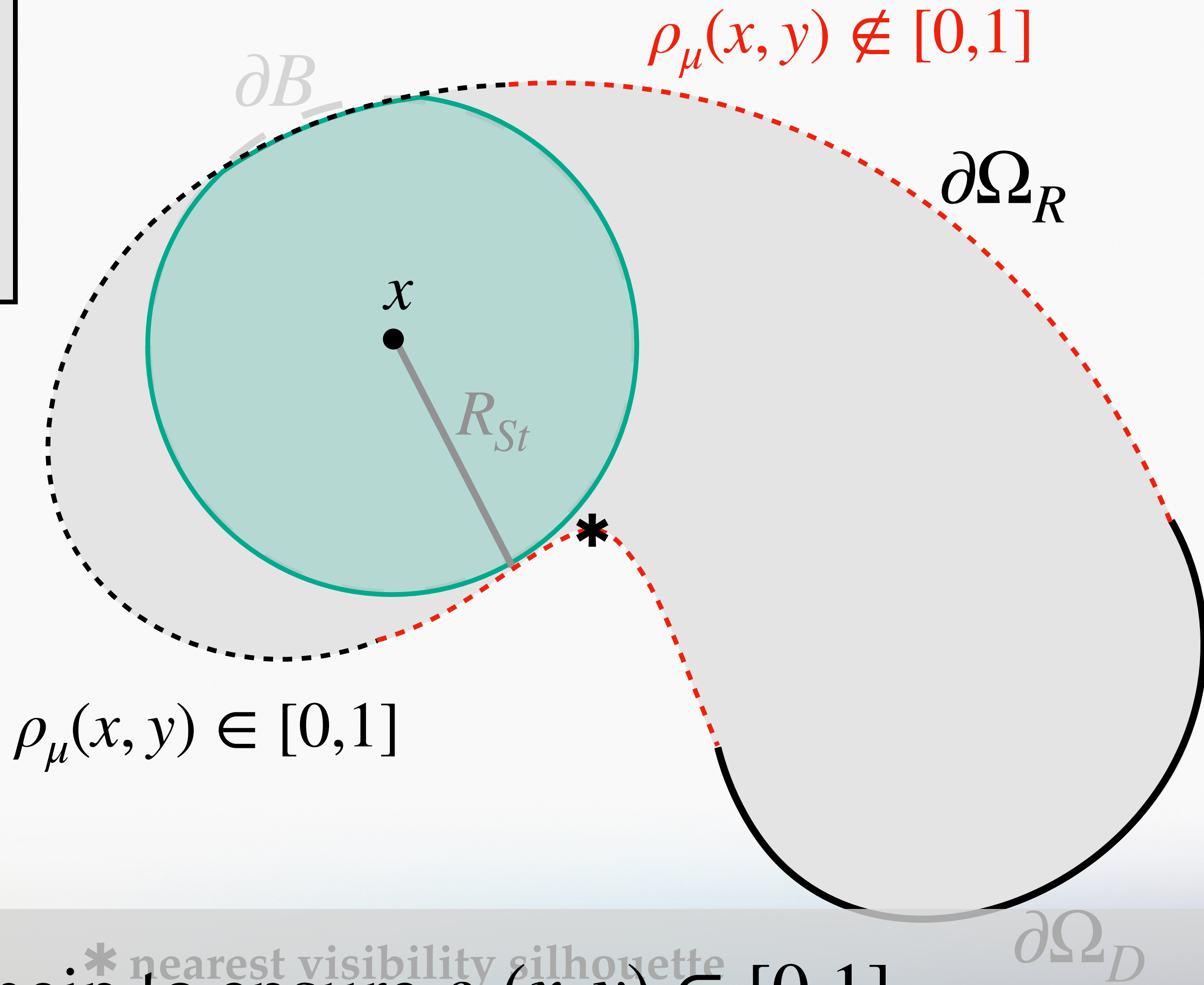
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

## Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

reflectance product bound to [0,1]



key idea: shrink star-shaped domain to ensure  $\rho_{\mu}(x, y) \in [0,1]$



# keeping reflectance bounded

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

## Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

reflectance product bound to [0,1]

$$\rho_{\mu}(x, y) \in [0, 1]$$

WoSt with Robin conditions. Lastly, we consider WoSt with Robin conditions (Section 4). As above, we start by writing the boundary integral in Equation 6 in operator-theoretic form, but we now modify our kernel to include the reflectance term,  $\kappa := \rho_{\mu} P^B$ , and use the same mapping A. Using the corresponding integral operator  $C_{\rho_{\mu} P^B}^{St}$ , we rewrite Equation 6 equivalently in operator form as

$$u = C_{\rho_{\mu} P^B}^{St}[u] + s, \tag{29}$$

As in the previous section, we need to select a radius  $R$  that ensures the unit resolvent  $\mathcal{R}^{St} := (I - C_{\rho_{\mu} P^B}^{St})^{-1}$  of  $C_{\rho_{\mu} P^B}^{St}$  exists and is bounded. This will guarantee that the estimator

$$u = \mathcal{R}^{St}[s] \tag{30}$$

can be estimated using recursive Monte Carlo. Using the radius for a Dirichlet-Neumann problem, we have

$$\int_{\partial St(x,R)} |\rho_{\mu}(x, y) P^B(x, y)| dy \leq \int_{\partial St(x,R)} |\rho_{\mu}(x, y)| dy \int_{\partial St(x,R)} |P^B(x, y)| dy = \int_{\partial St(x,R)} |\rho_{\mu}(x, y)| dy, \tag{31}$$

where we used Hölder's inequality and Equation 27. To ensure that

$$\int_{\partial St(x,R)} |\rho_{\mu}(x, y)| dy \leq 1, \tag{32}$$

we follow Section 4.2 to further restrict  $R$  such that  $|\rho_{\mu}(x, y)| \leq 1$

key idea: shrink star-shaped domain to ensure  $\rho_{\mu}(x, y) \in [0, 1]$  \*nearest visibility silhouette

$\partial\Omega_D$



# walk on stars with probabilistic termination

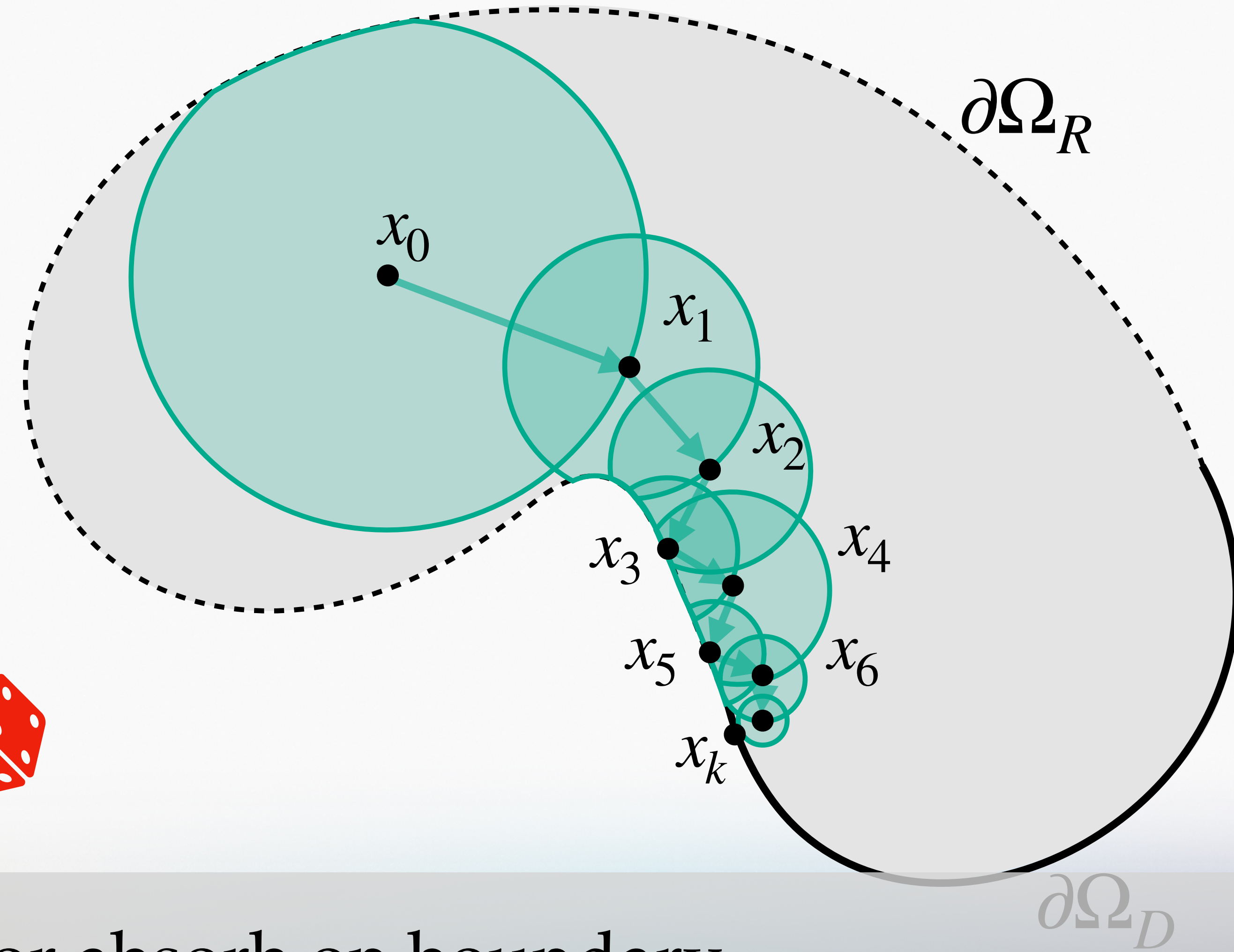
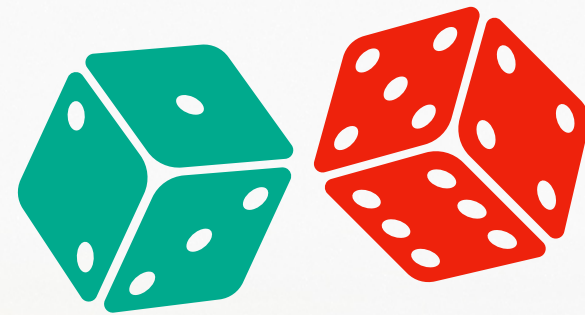
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

apply Russian roulette  
to terminate walks  
with low throughput



key idea: probabilistically **reflect** or absorb on boundary



# walk on stars with probabilistic termination

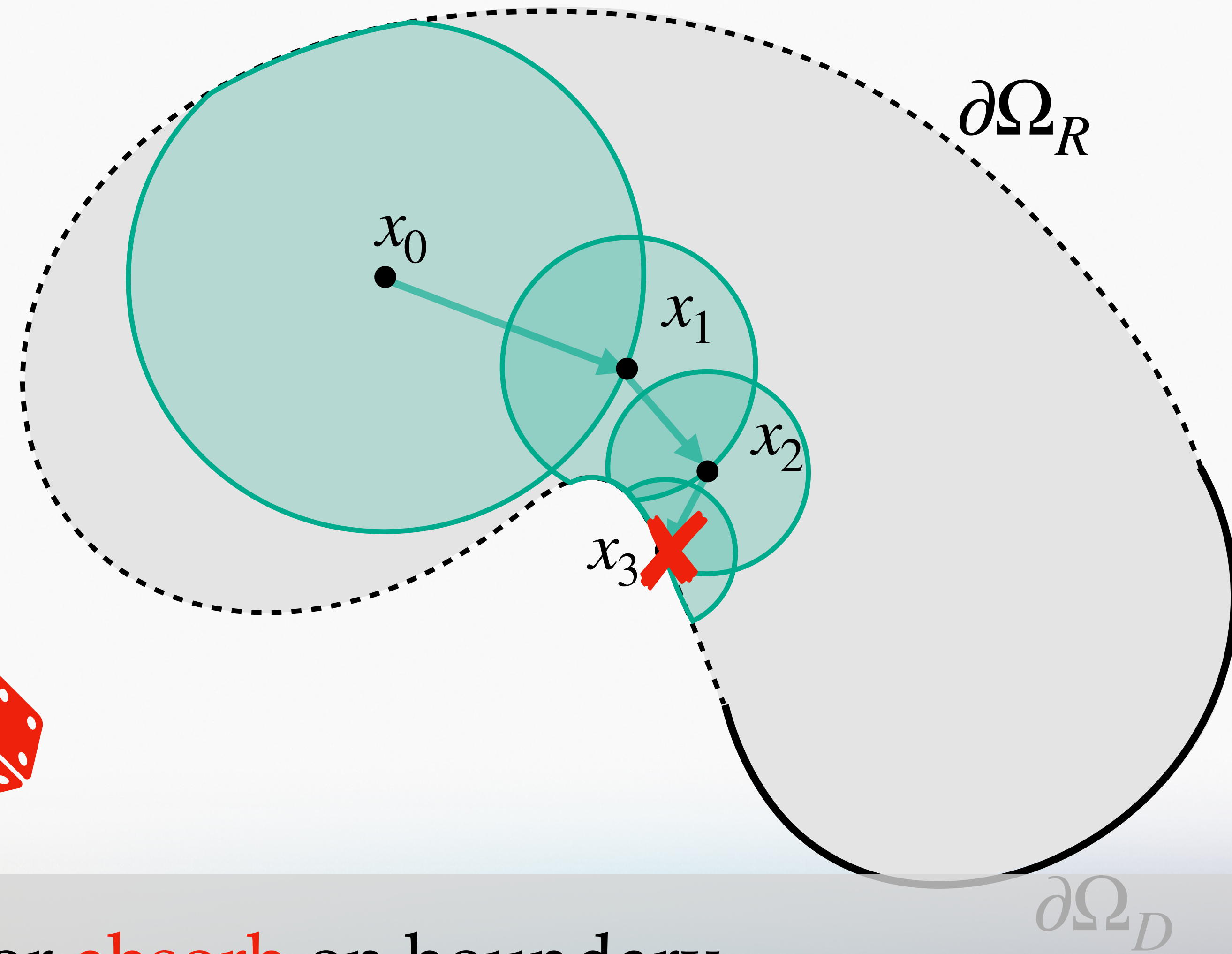
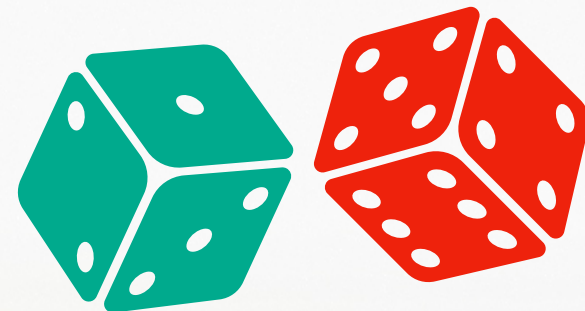
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_{\mu}(x, y) P^B(x, y) u(y) dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

apply Russian roulette  
to terminate walks  
with low throughput



key idea: probabilistically reflect or **absorb** on boundary



# generalized walk on stars overview



# generalized walk on stars overview

**only minor changes to original walk on stars method:**



**only minor changes to original walk on stars method:**

modulate contributions  
by reflectance

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

apply Russian roulette  
to terminate walks  
with low throughput



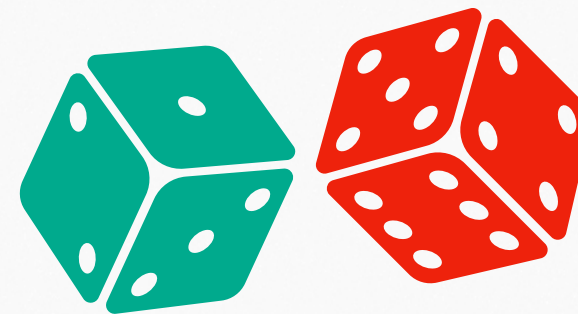
## only minor changes to original walk on stars method:

modulate contributions  
by reflectance

probabilistically terminate  
paths at each boundary hit

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

apply Russian roulette  
to terminate walks  
with low throughput





# generalized walk on stars overview

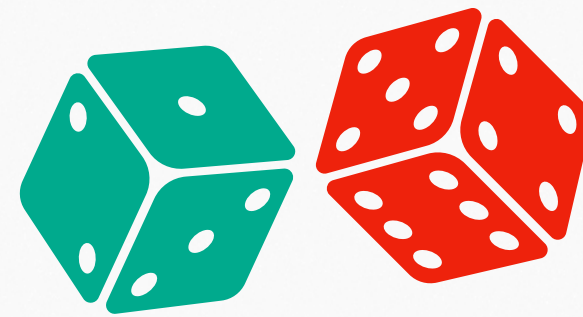
## only minor changes to original walk on stars method:

modulate contributions  
by reflectance

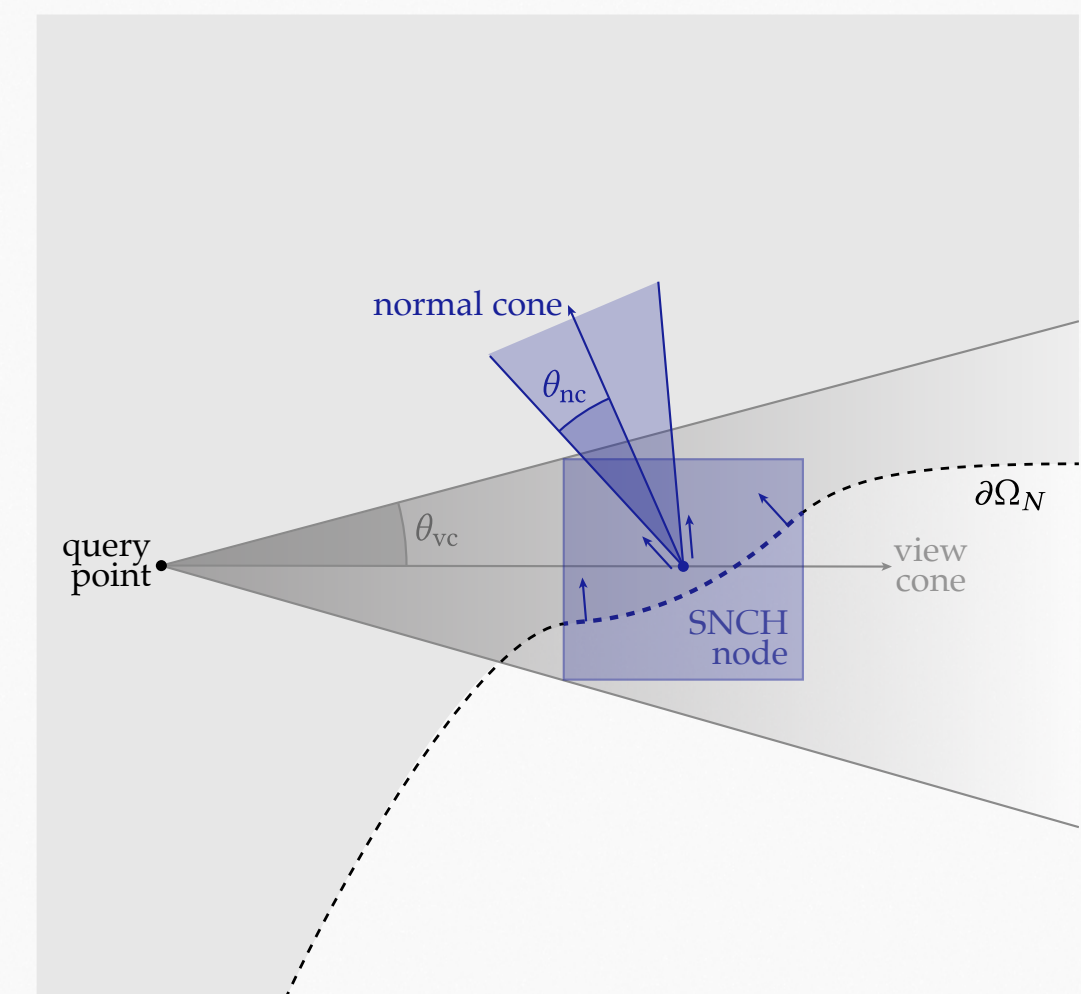
$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_{\mu}(x_i, x_{i+1})$$

apply Russian roulette  
to terminate walks  
with low throughput

probabilistically terminate  
paths at each boundary hit



update SNCH to query  
reflectance bounds





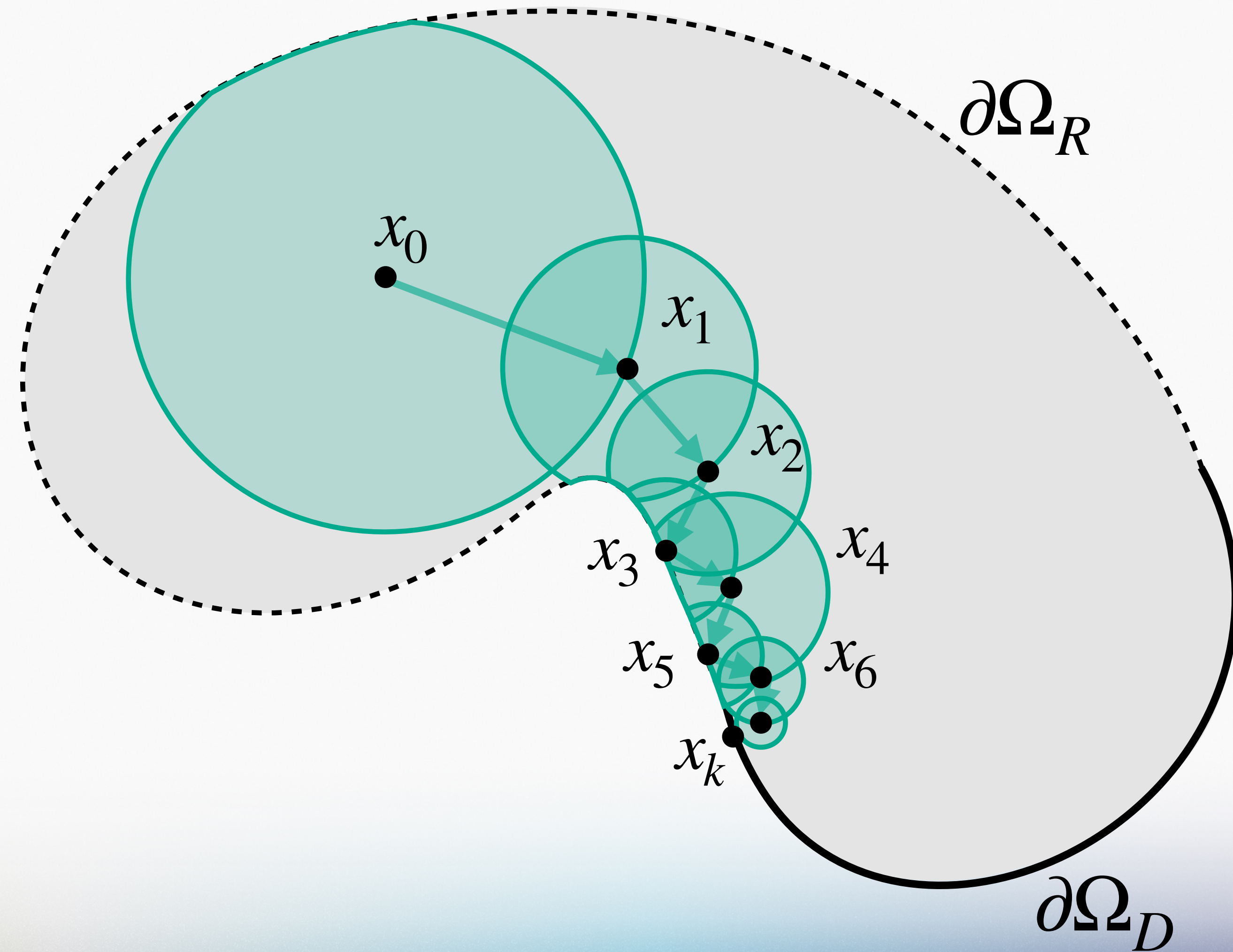
# Poisson equation and non-zero Robin condition

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

$$u = g \quad \text{on } \partial\Omega_D \quad \text{---}$$

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega_R \quad \text{- - -}$$

$$\mu : \partial\Omega \rightarrow \mathbb{R}_{>0}$$

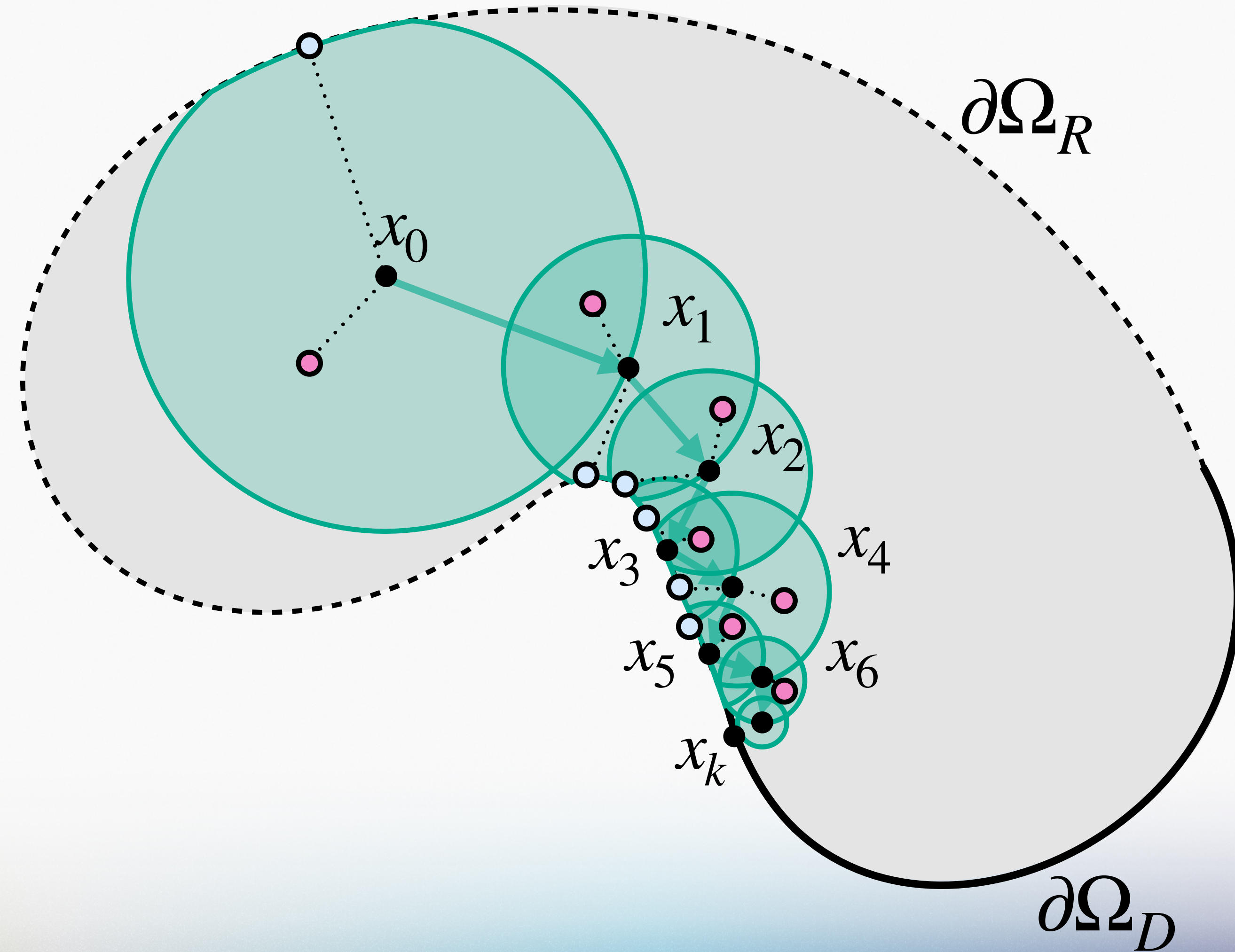




# Poisson equation and non-zero Robin condition

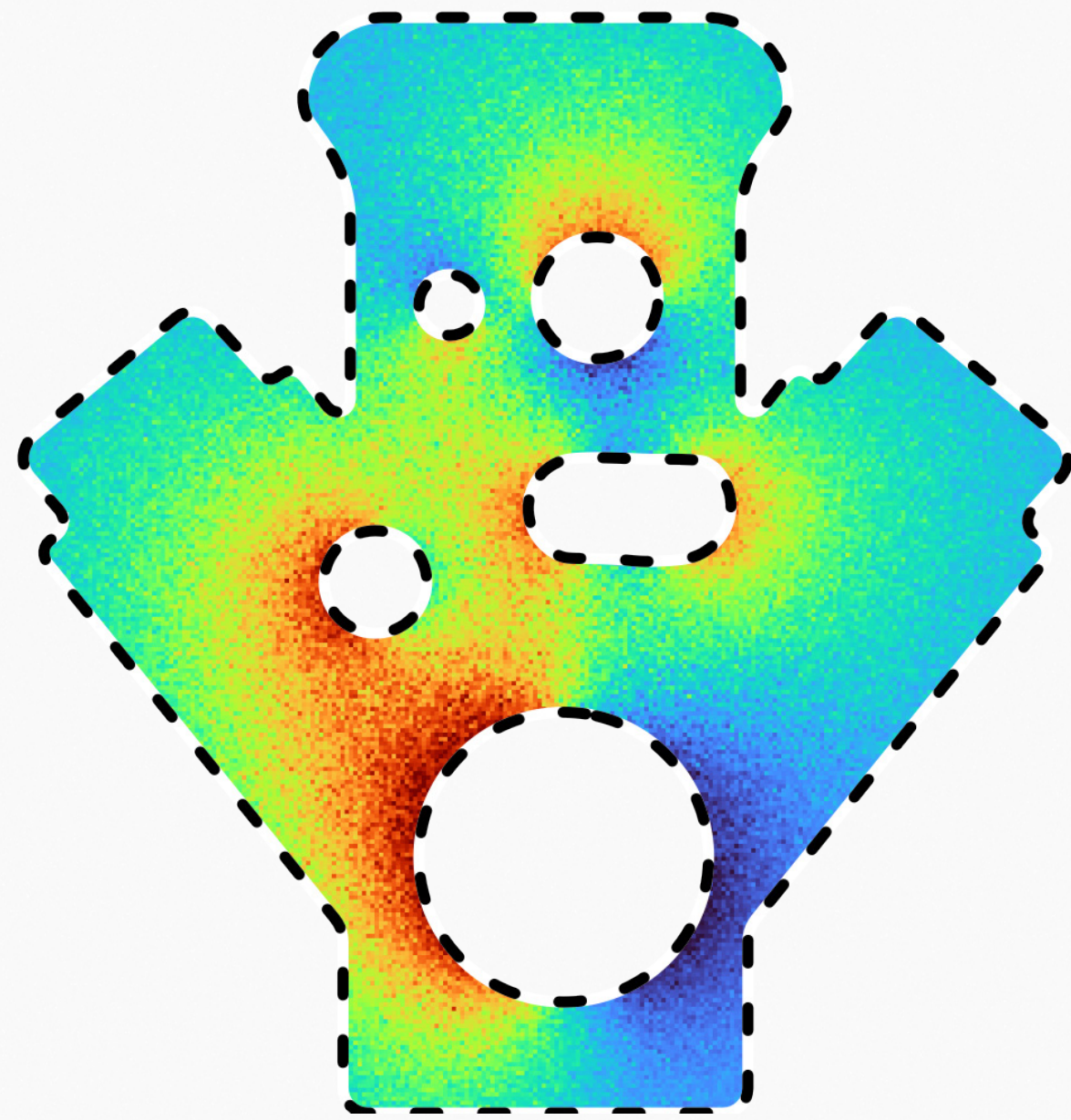
$$\begin{aligned} \Delta u &= f && \text{on } \Omega \\ u &= g && \text{on } \partial\Omega_D \text{ ---} \\ \frac{\partial u}{\partial n} - \mu u &= h && \text{on } \partial\Omega_R \text{ ---} \end{aligned}$$

$$\mu : \partial\Omega \rightarrow \mathbb{R}_{>0}$$



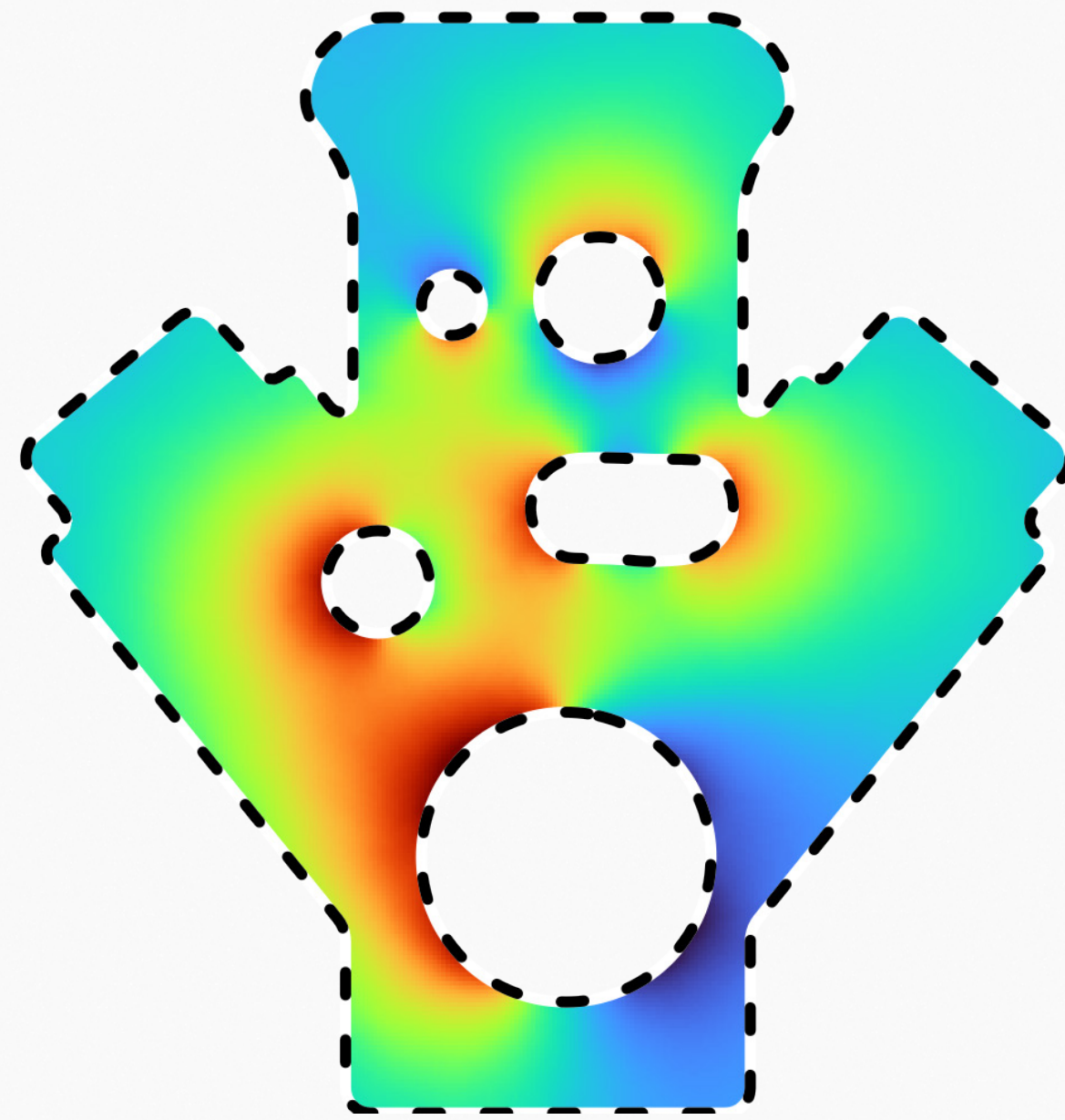


WoSt



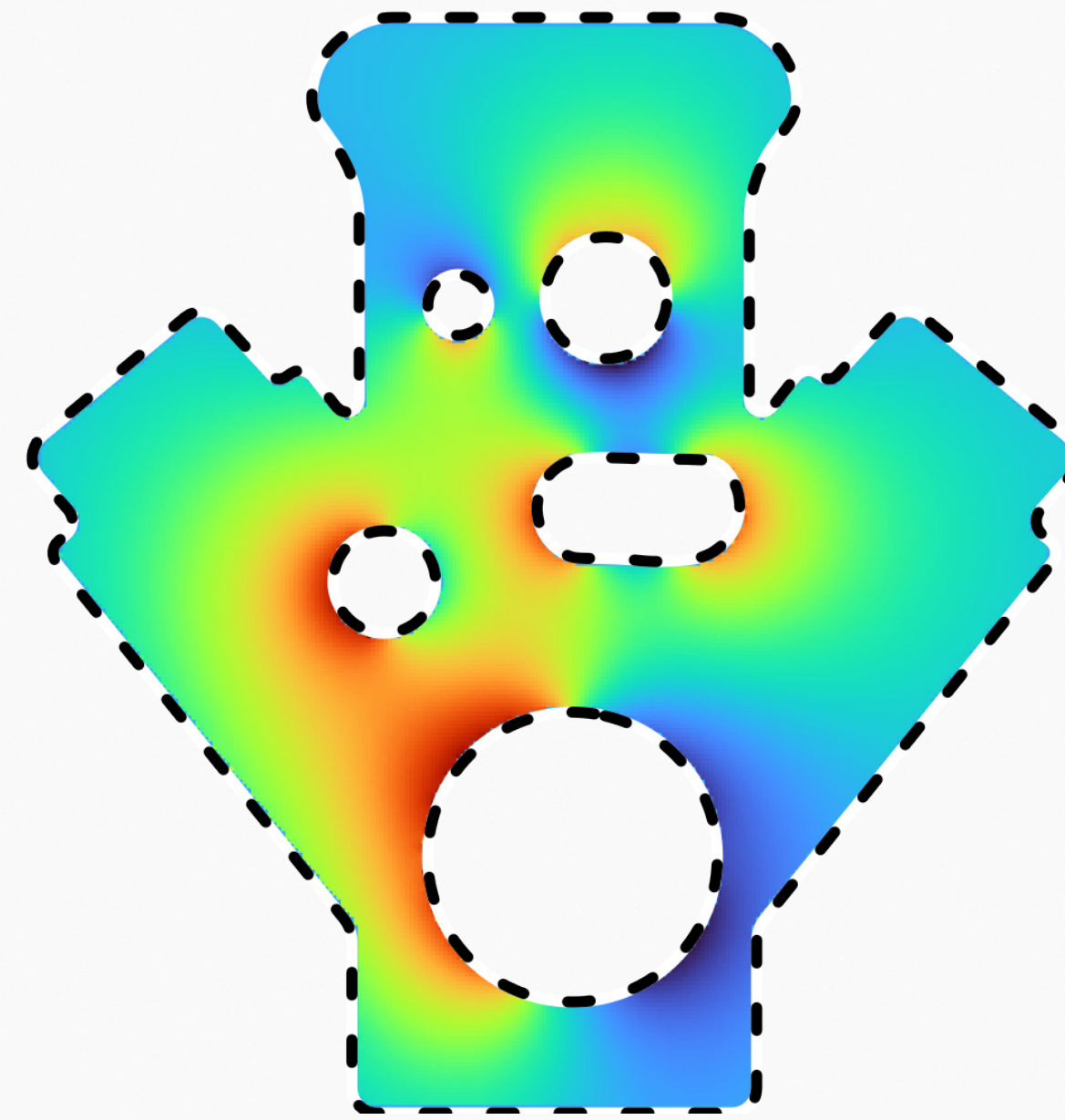
reverse WoSt

[Qi et al. 2022]

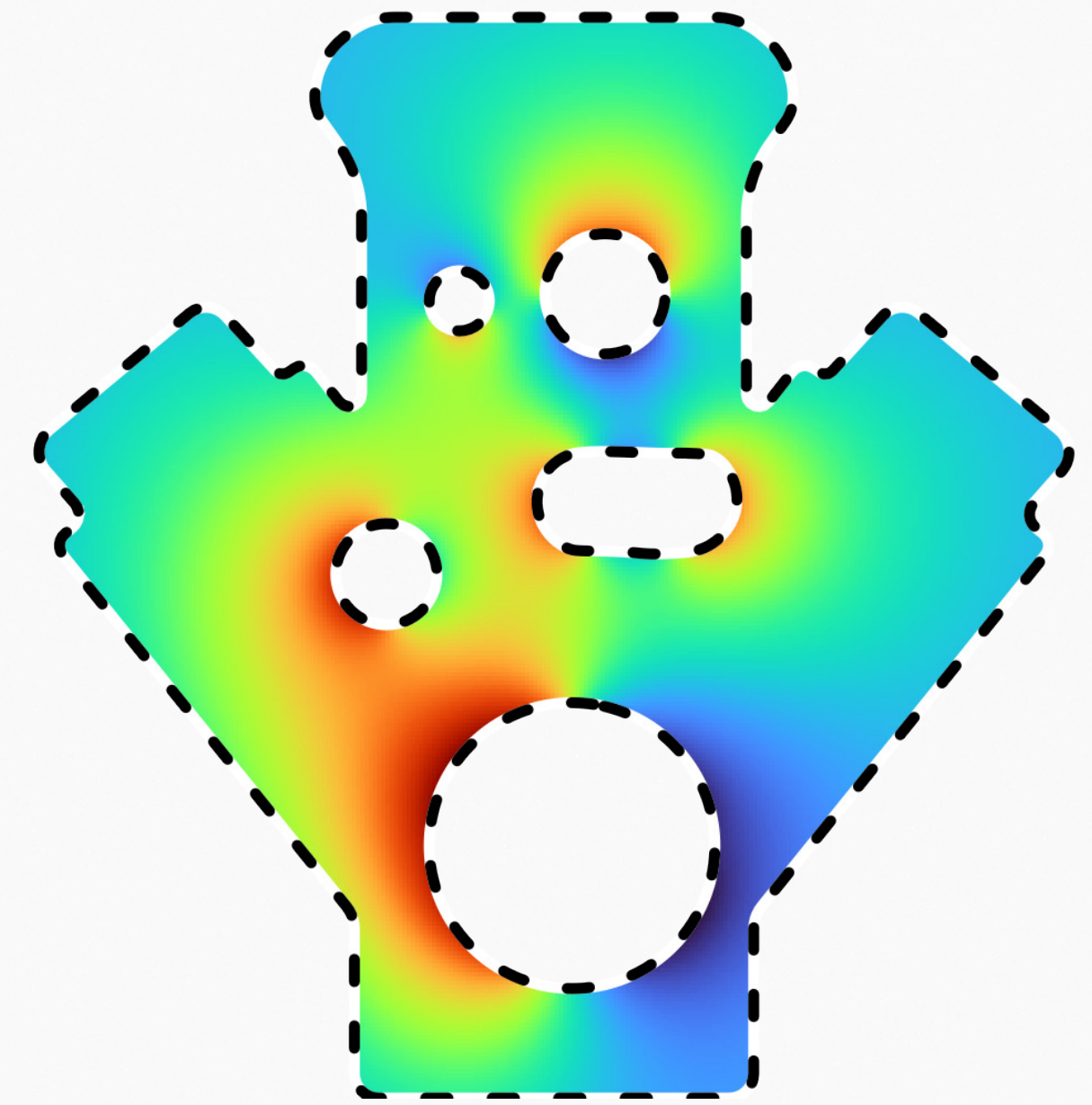


boundary value caching

[Miller et al. 2023]



reference



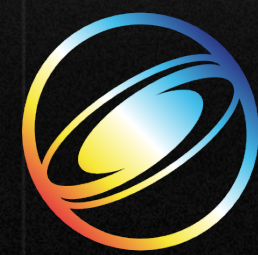
min value  max value

■ ■ ■ Robin boundary condition

**key idea:** can directly apply existing variance reduction techniques



SIGGRAPH  
2024



**SIGGRAPH 2024**  
DENVER+ 28 JUL — 1 AUG

# EVALUATION





# walk on boundary vs walk on stars

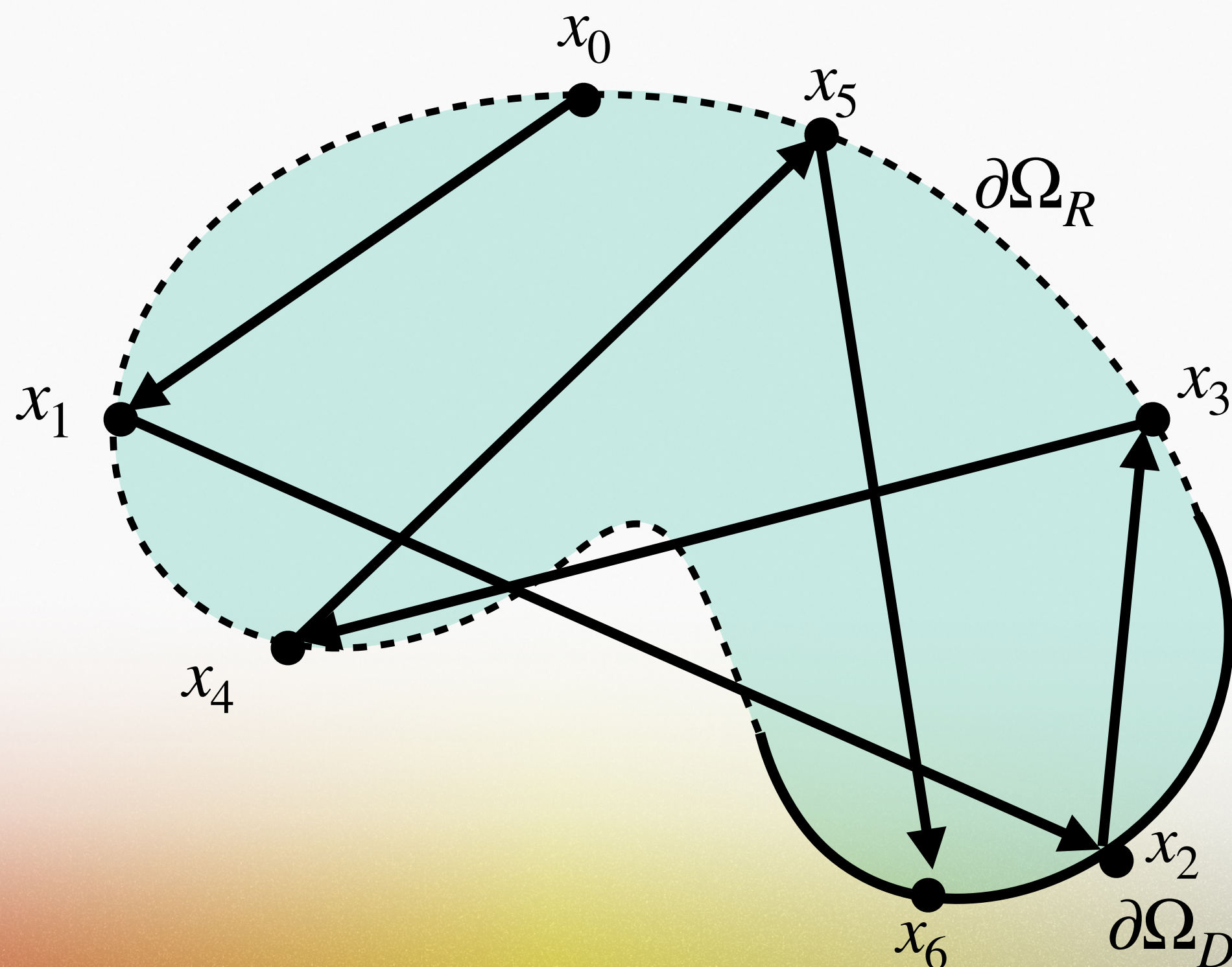
## walk on boundary

[Sabelfeld and Simonov 2013, Sugimoto et al. 2023]

**branching** paths (multiple intersections)

**unbounded** path throughput

ray trace on entire domain



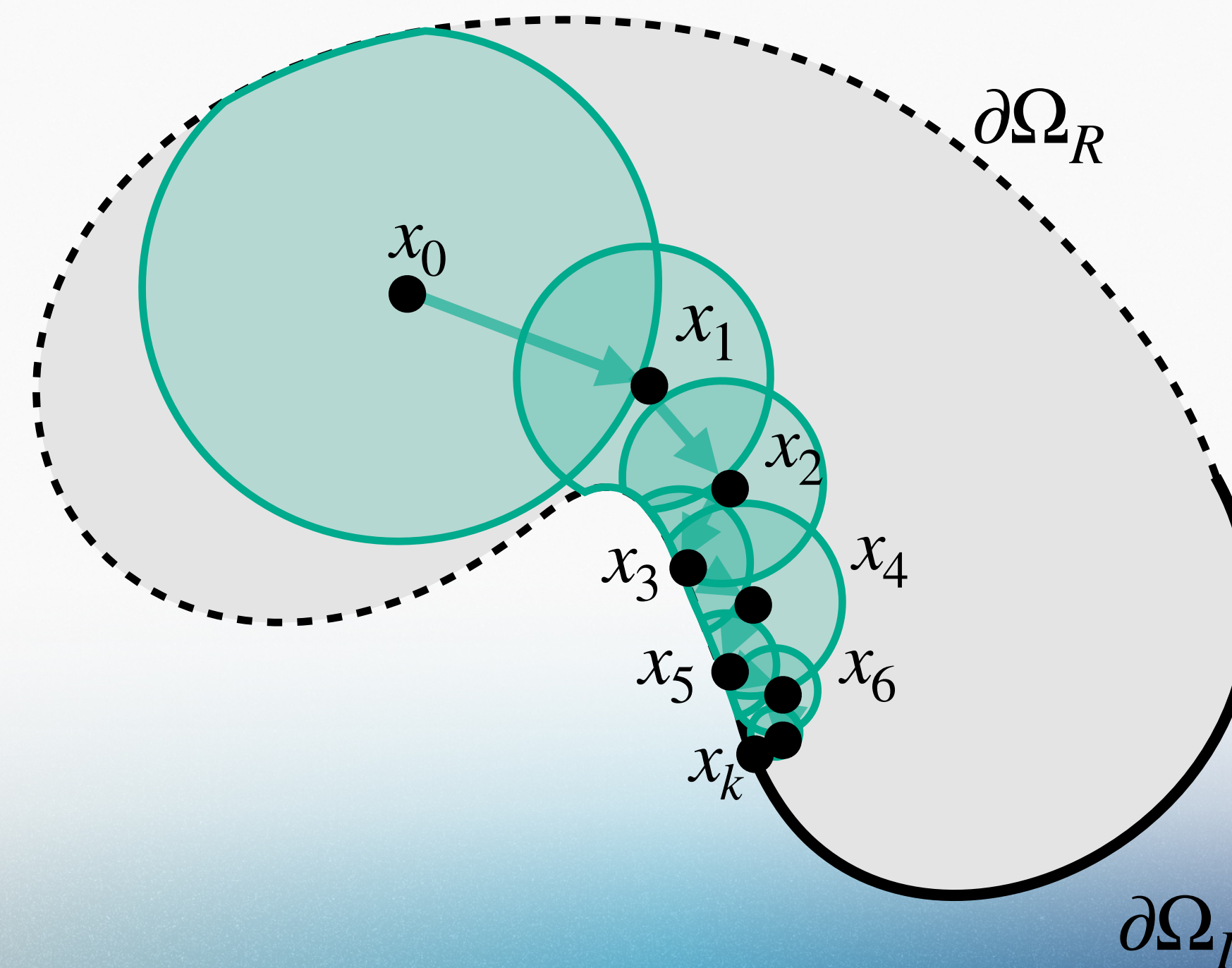
## walk on stars

[Sawhney et al. 2023, ours]

**no branching** (single-intersections)

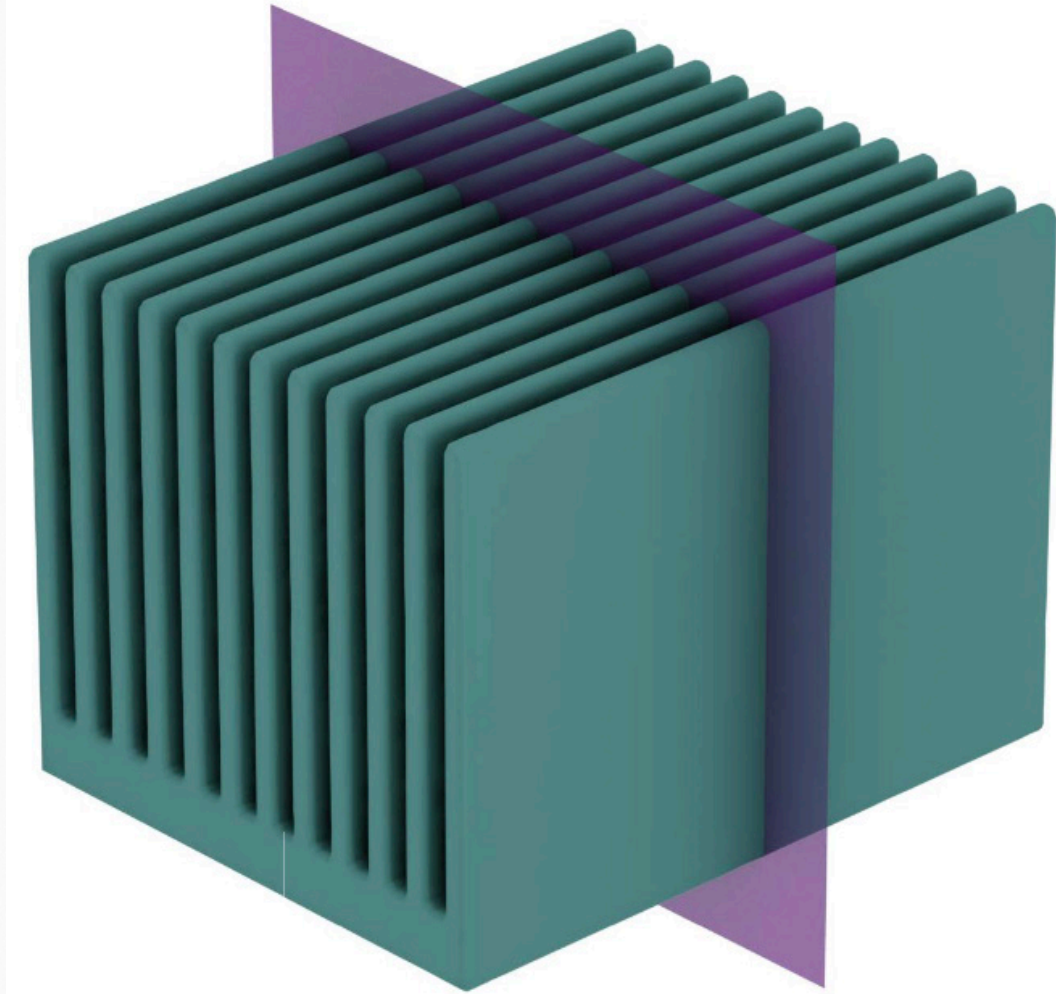
**bounded** path throughput  $[0,1]$

construct star-shaped subdomains





# walk on boundary vs walk on stars

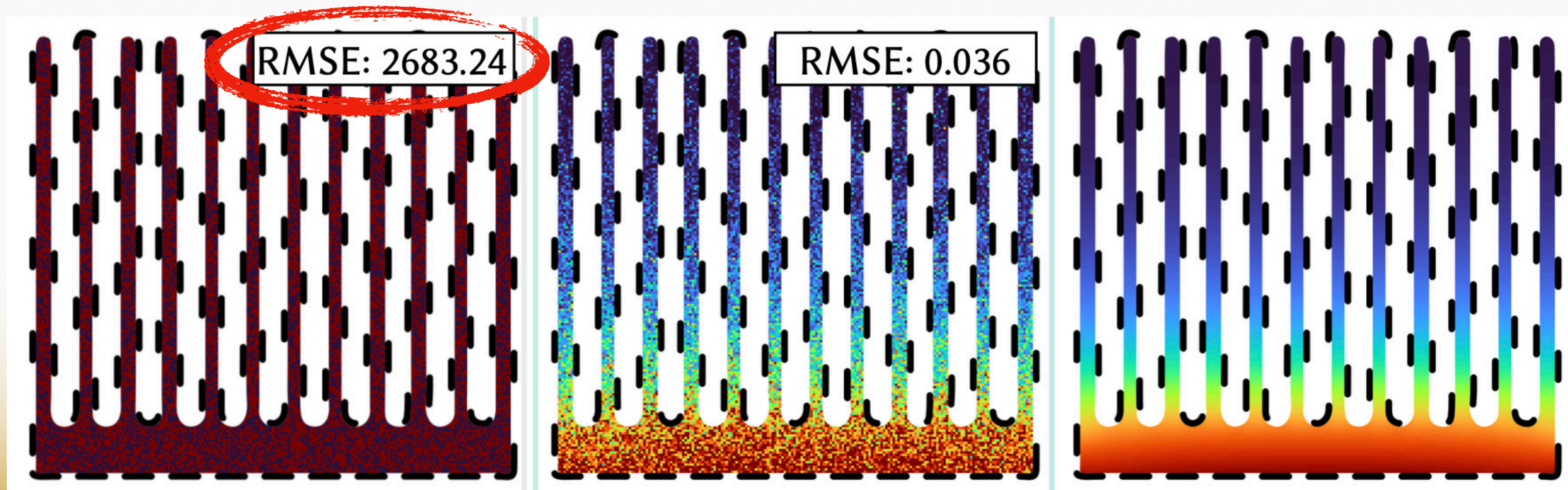


WoB

WoSt (ours)

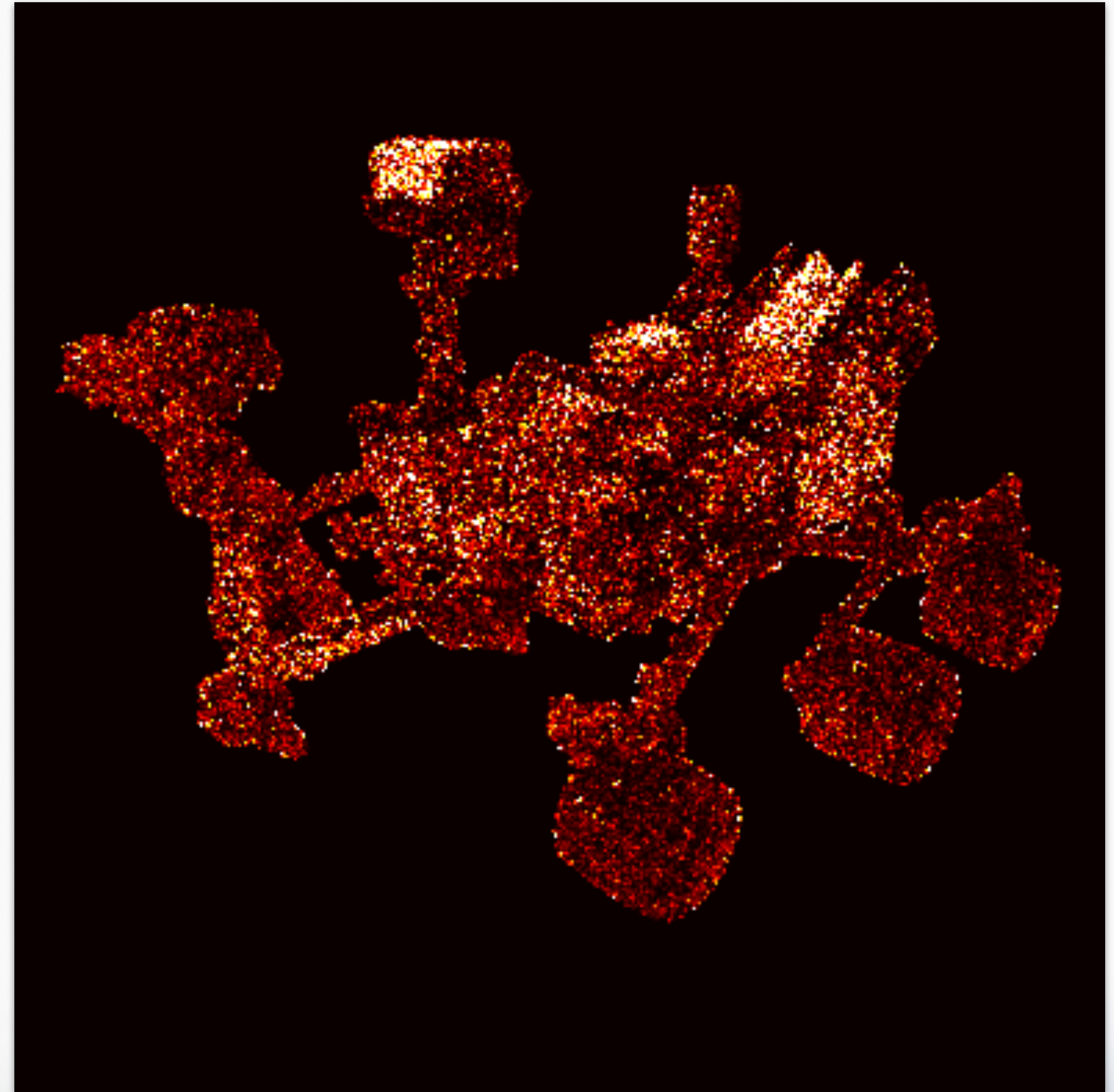
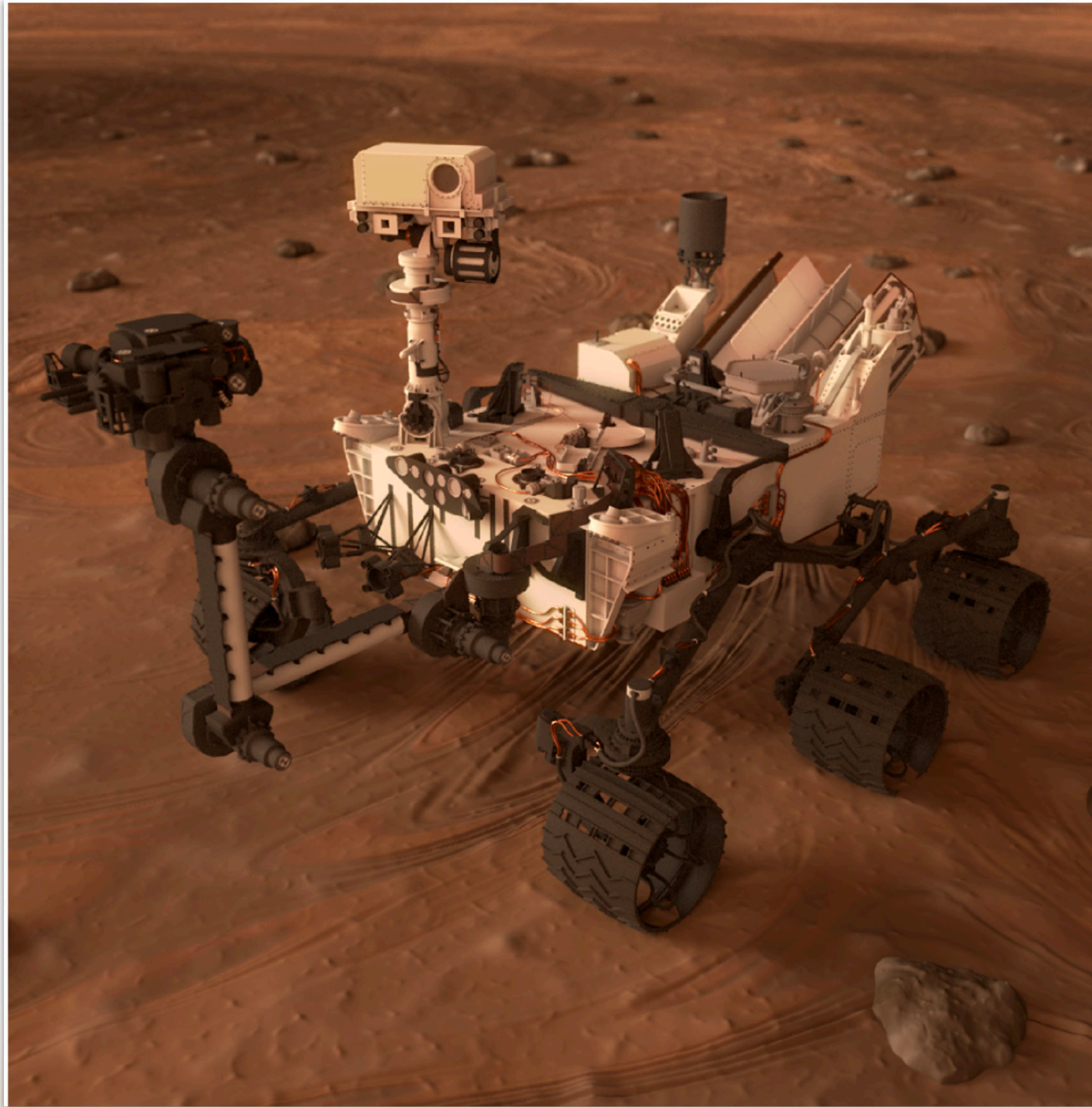
reference

*Robin*



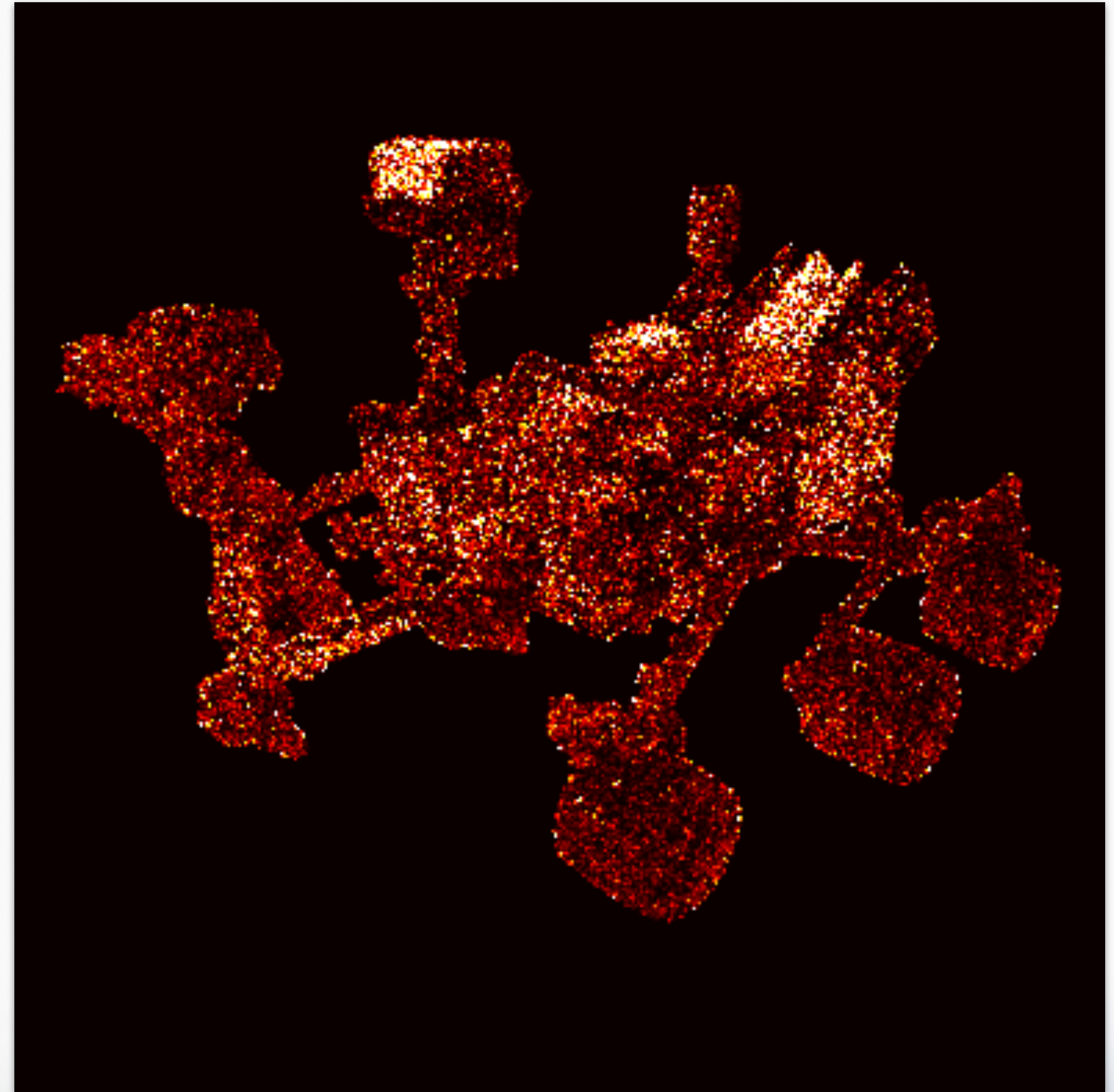
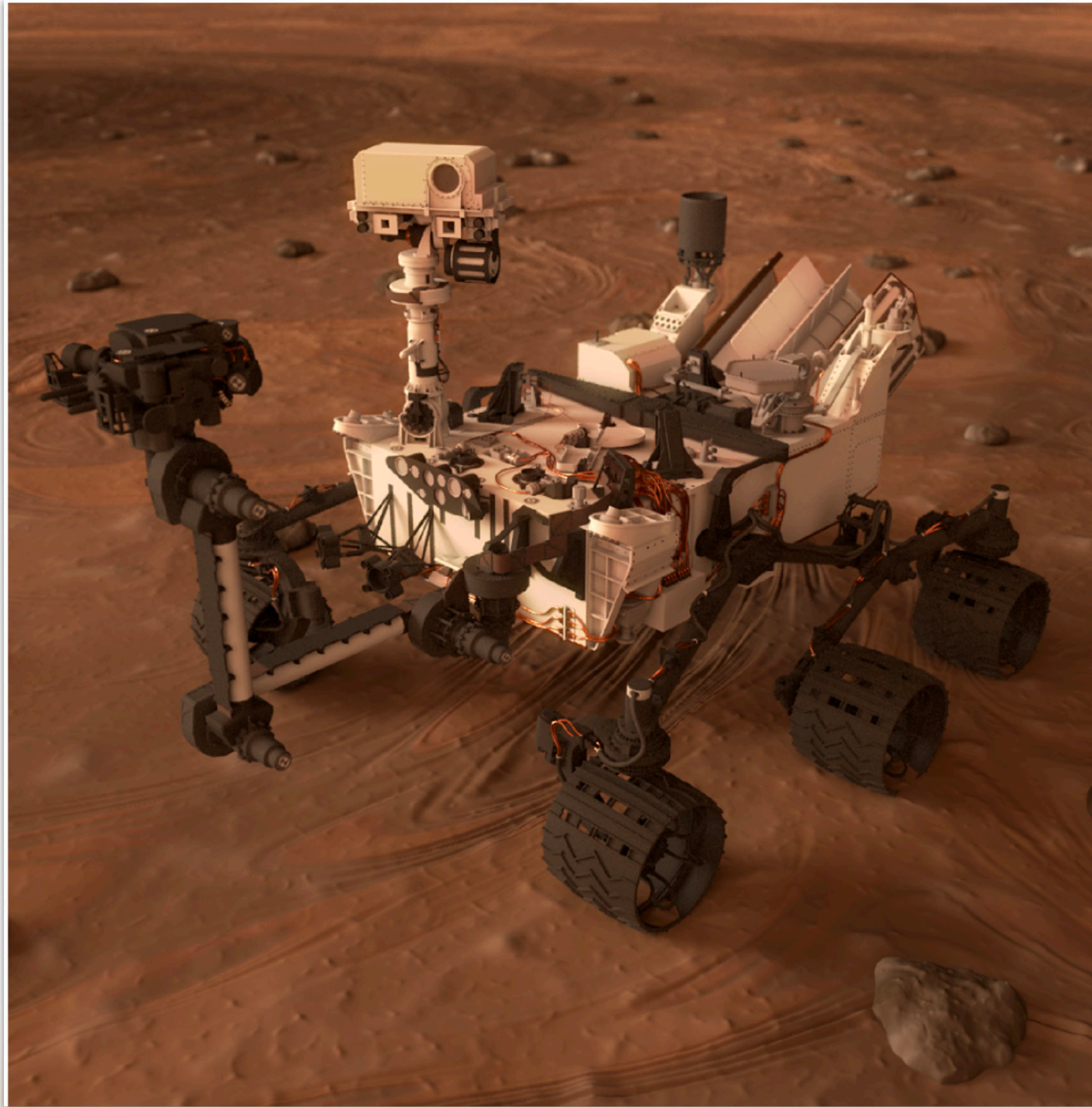


# Monte Carlo thermal simulation



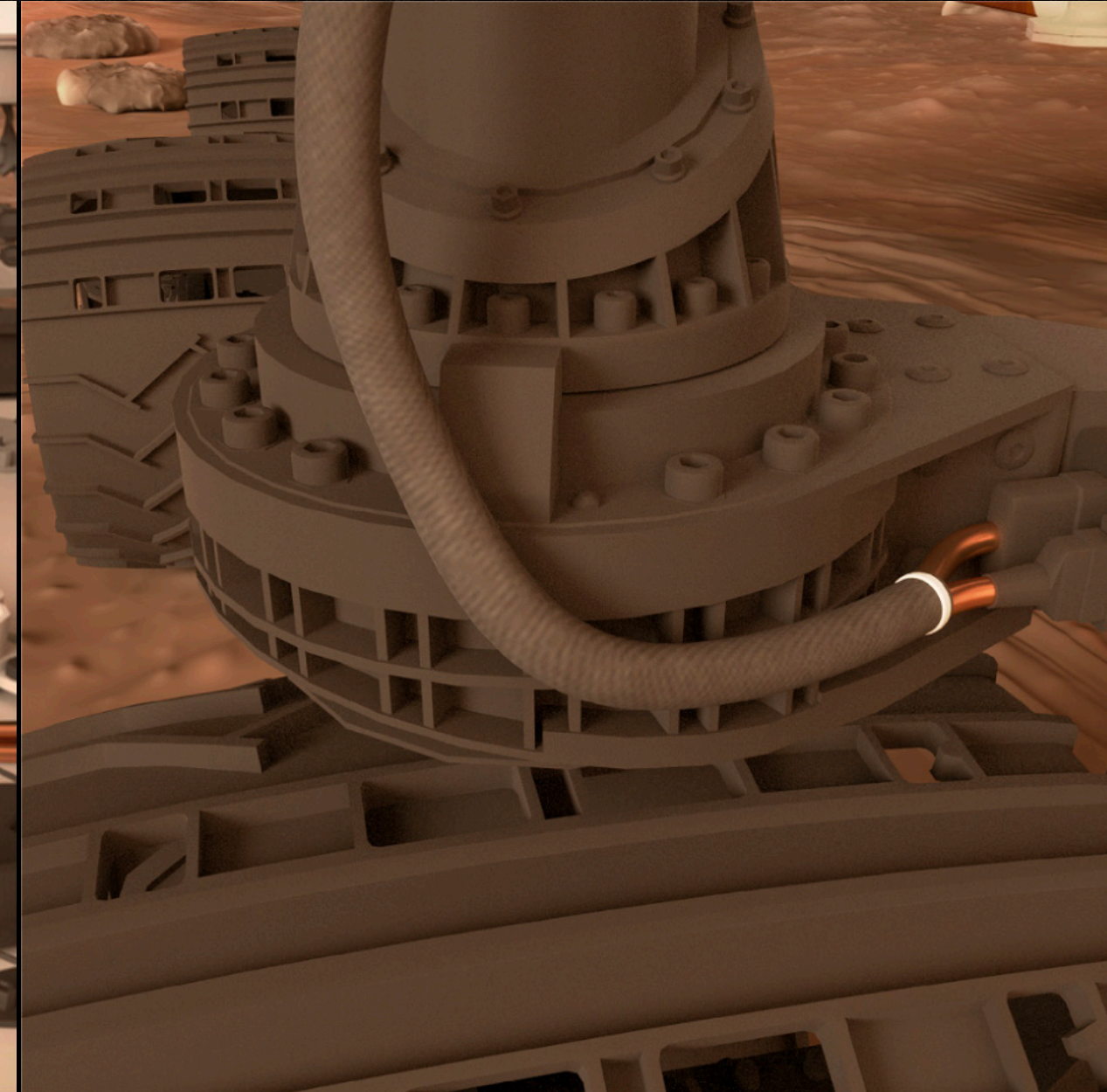
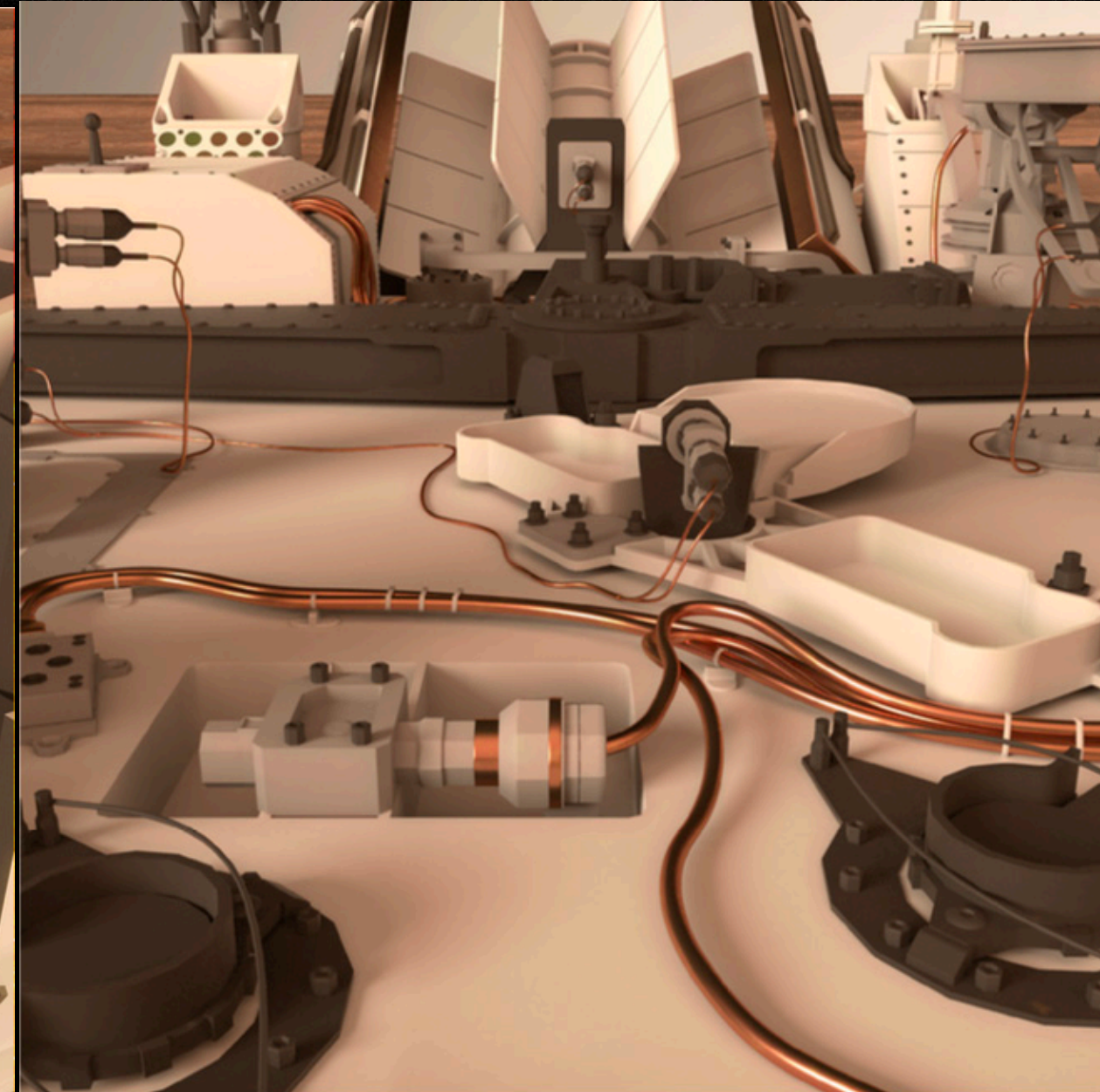
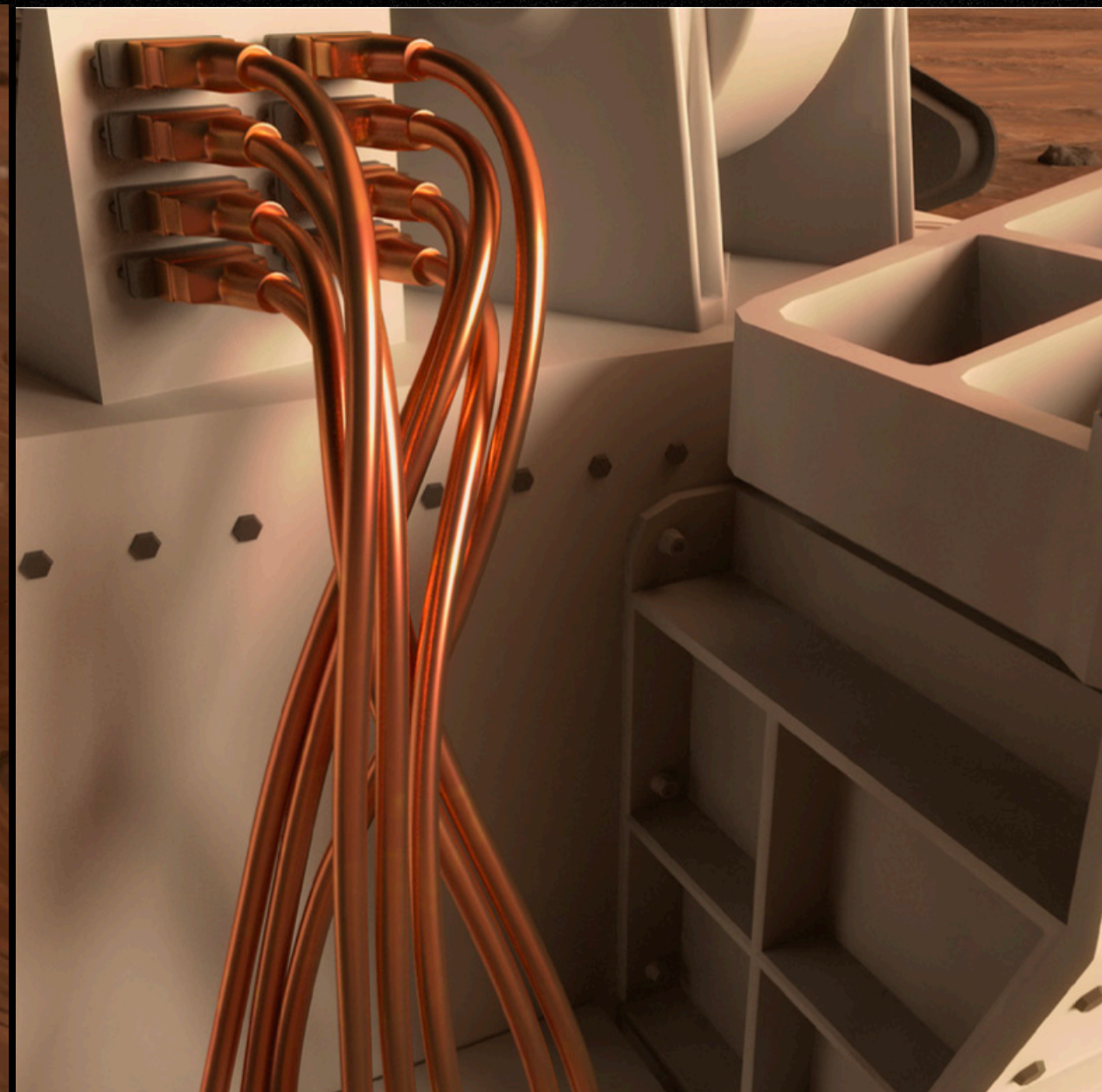
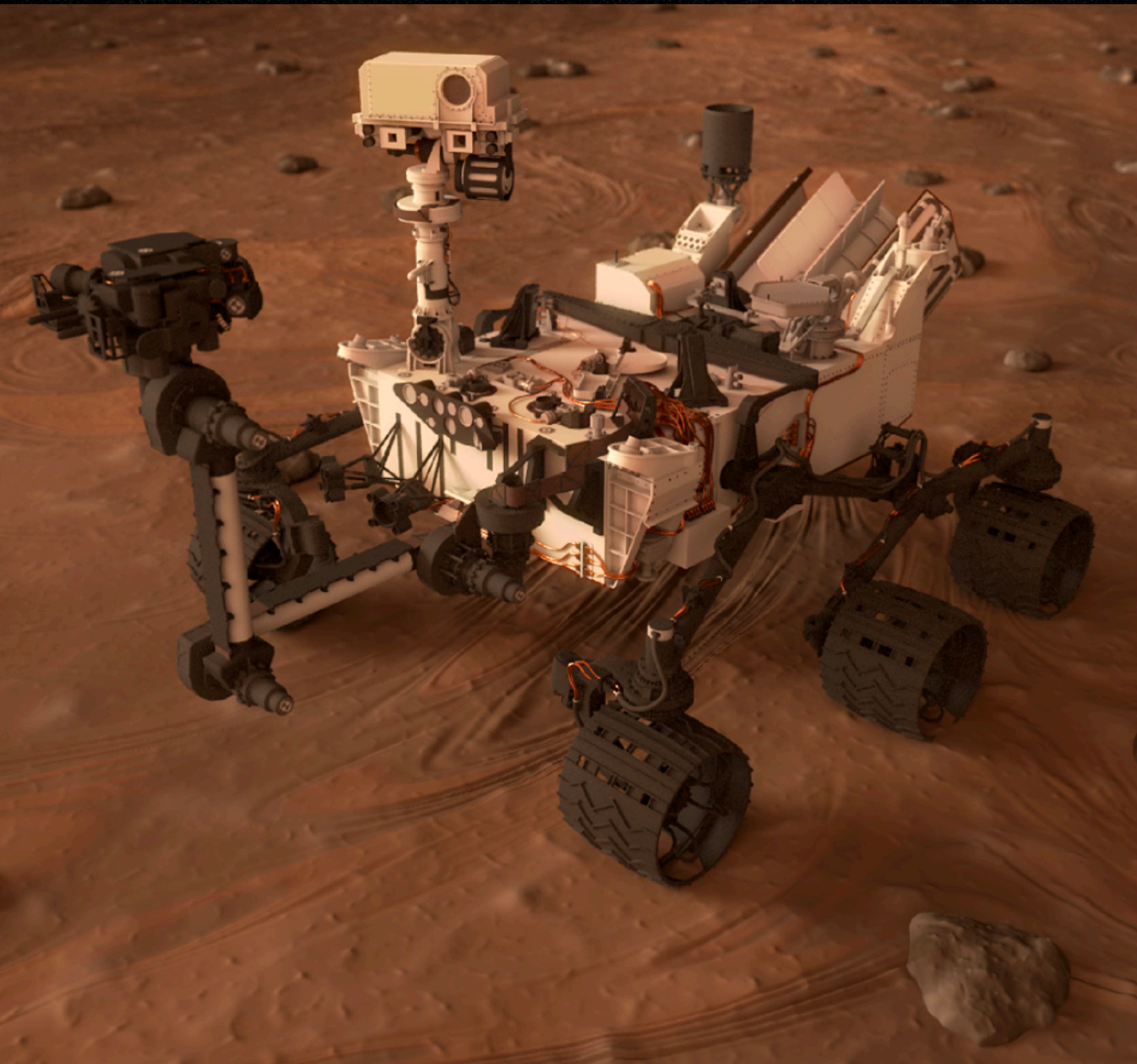


# Monte Carlo thermal simulation

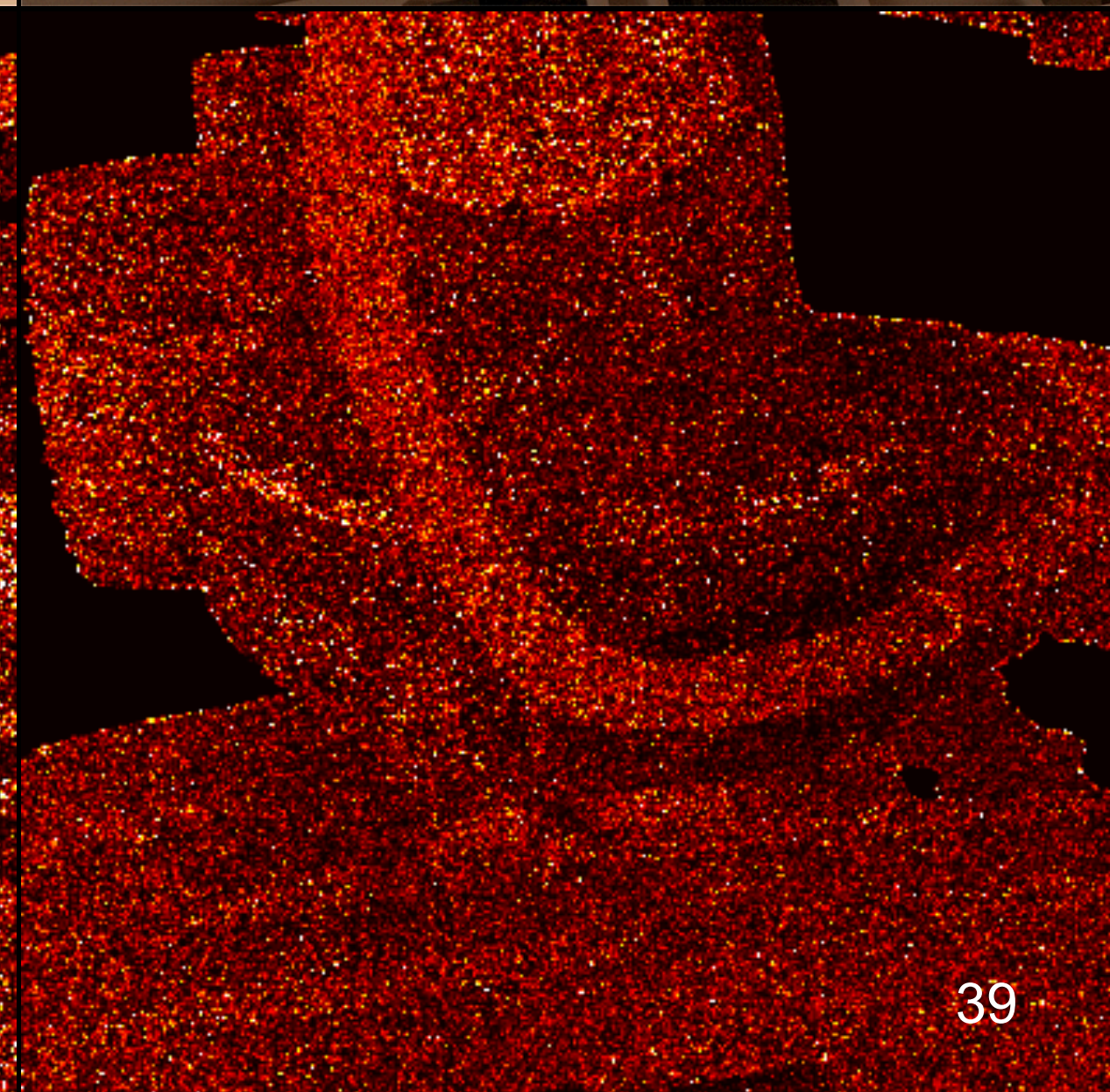
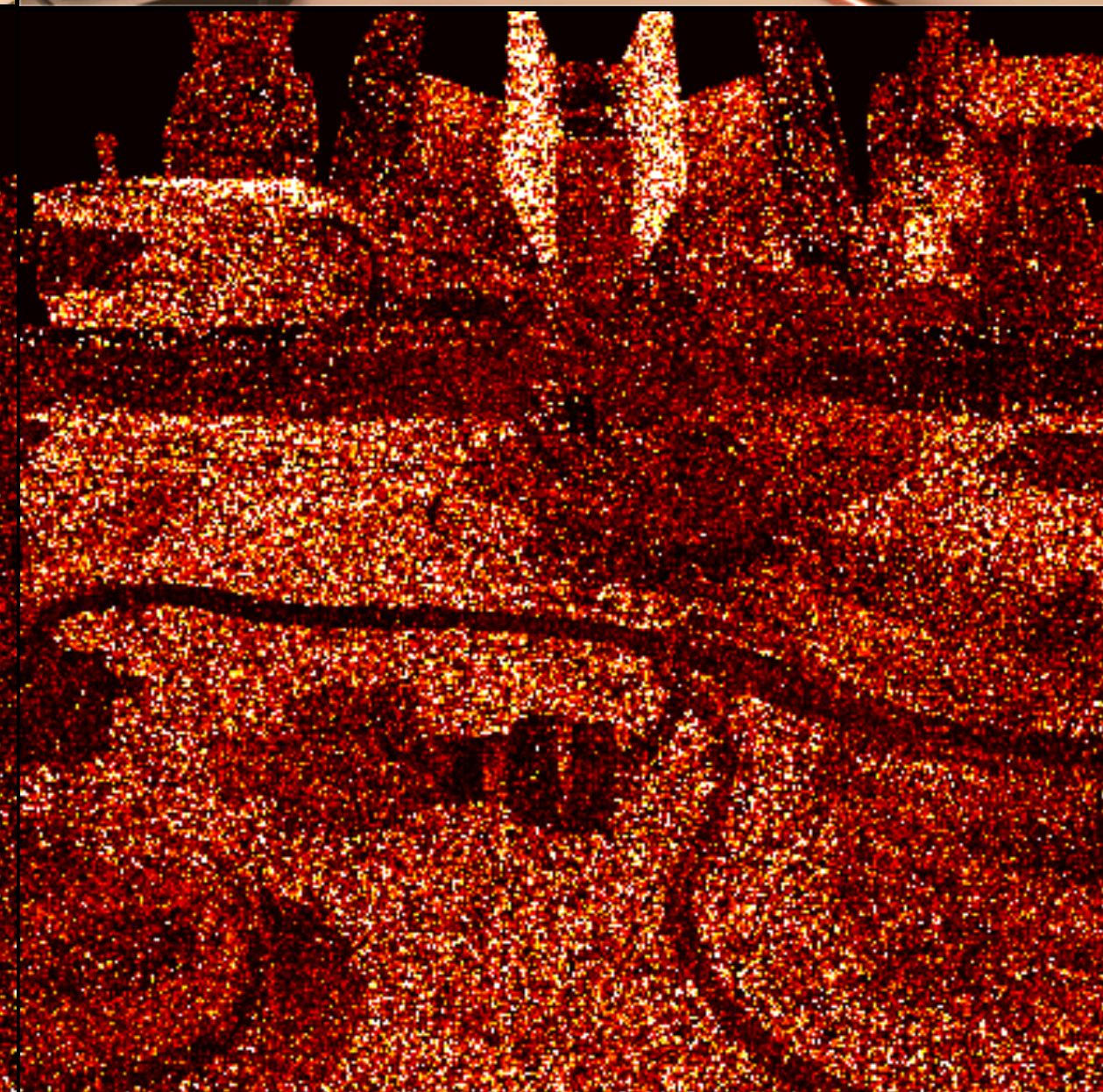
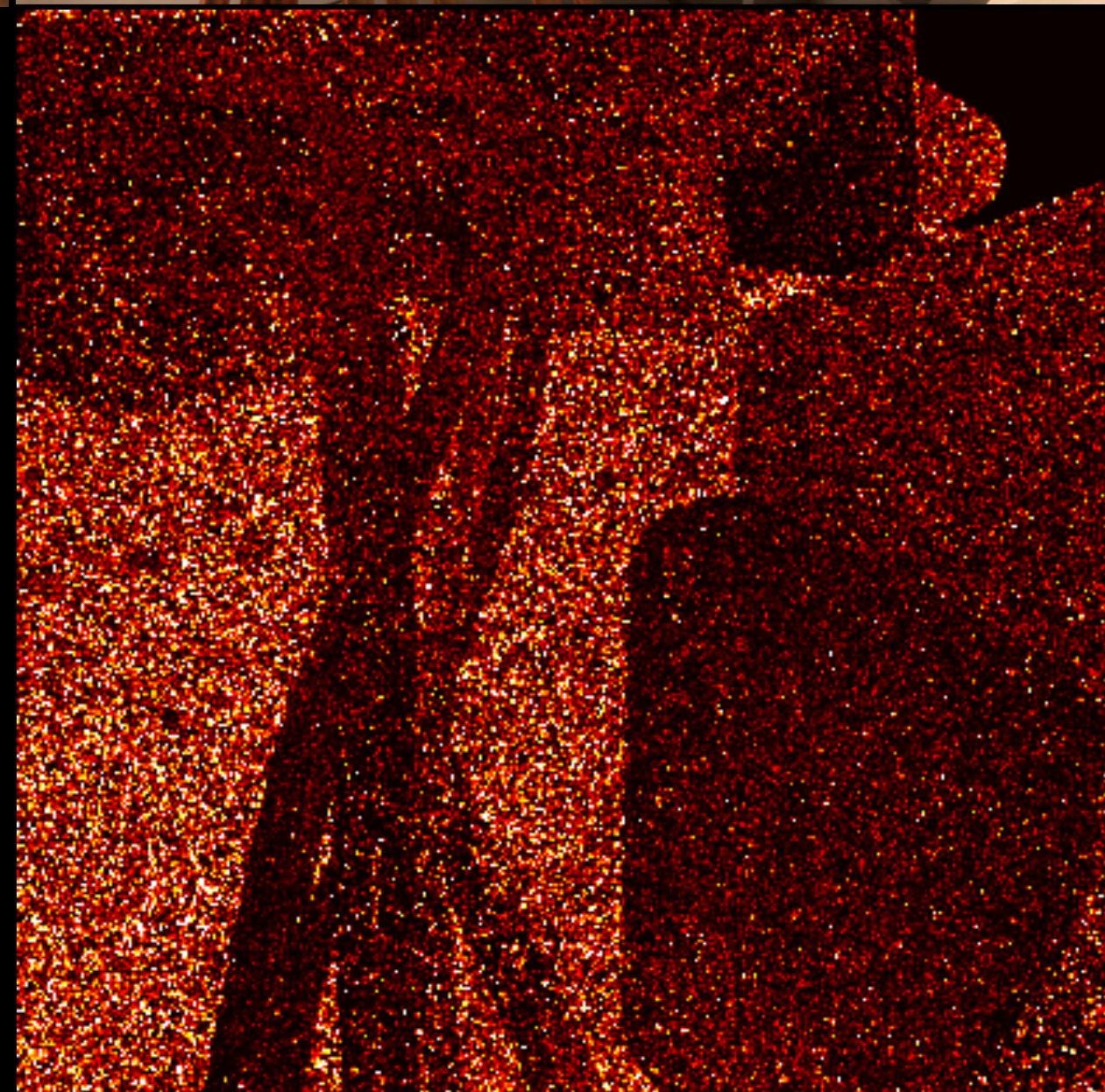




# Monte Carlo thermal simulation

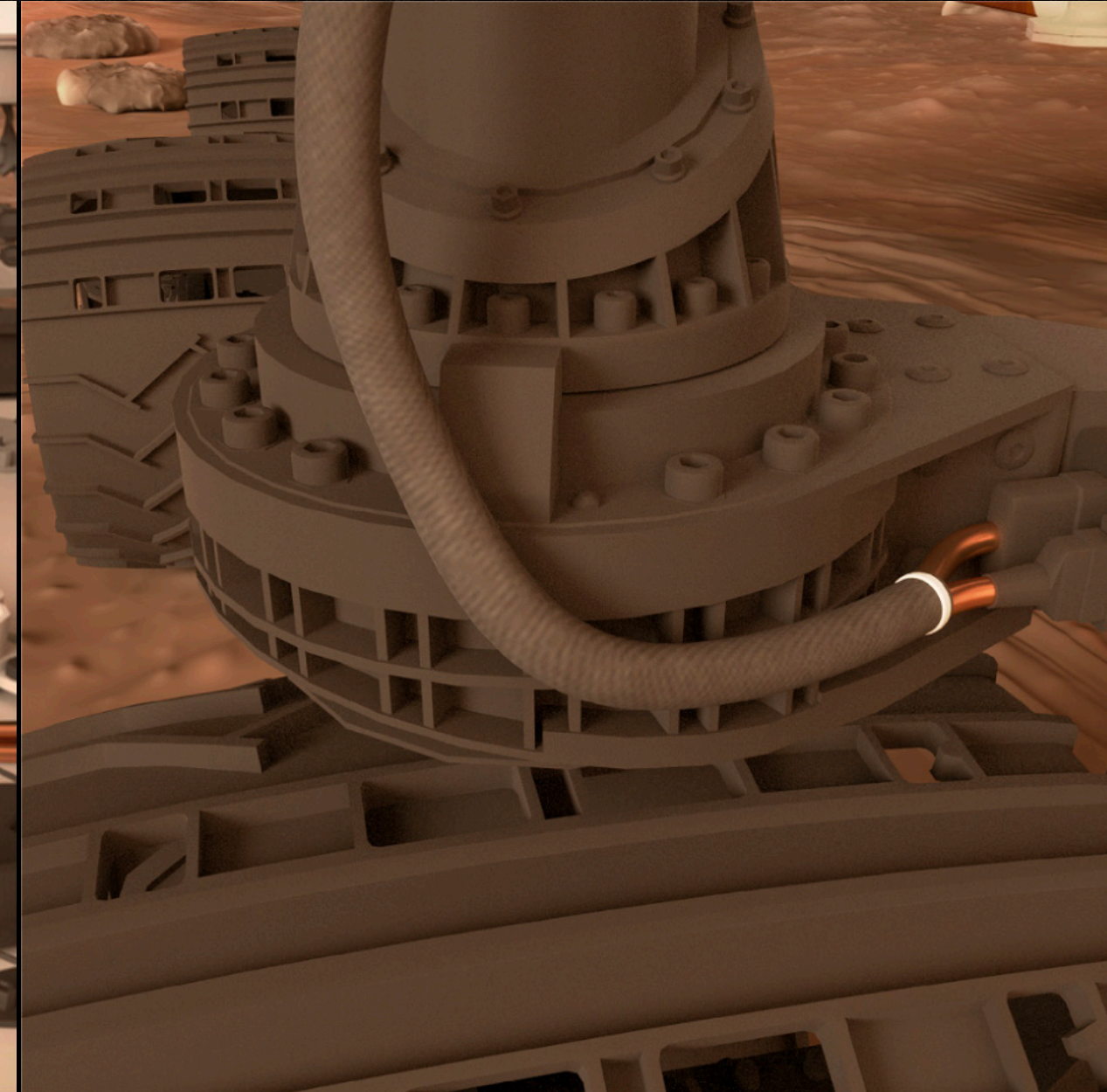
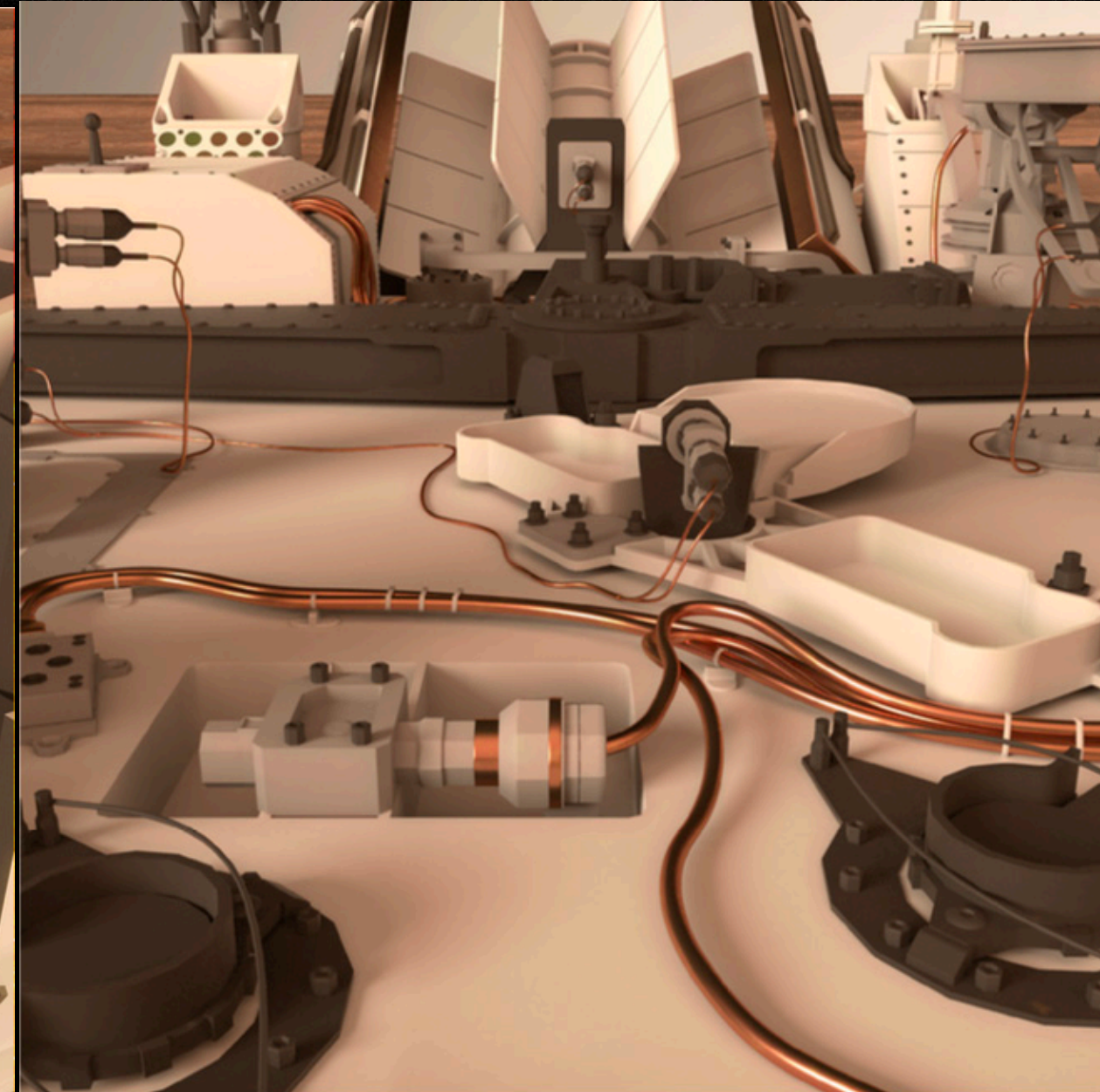
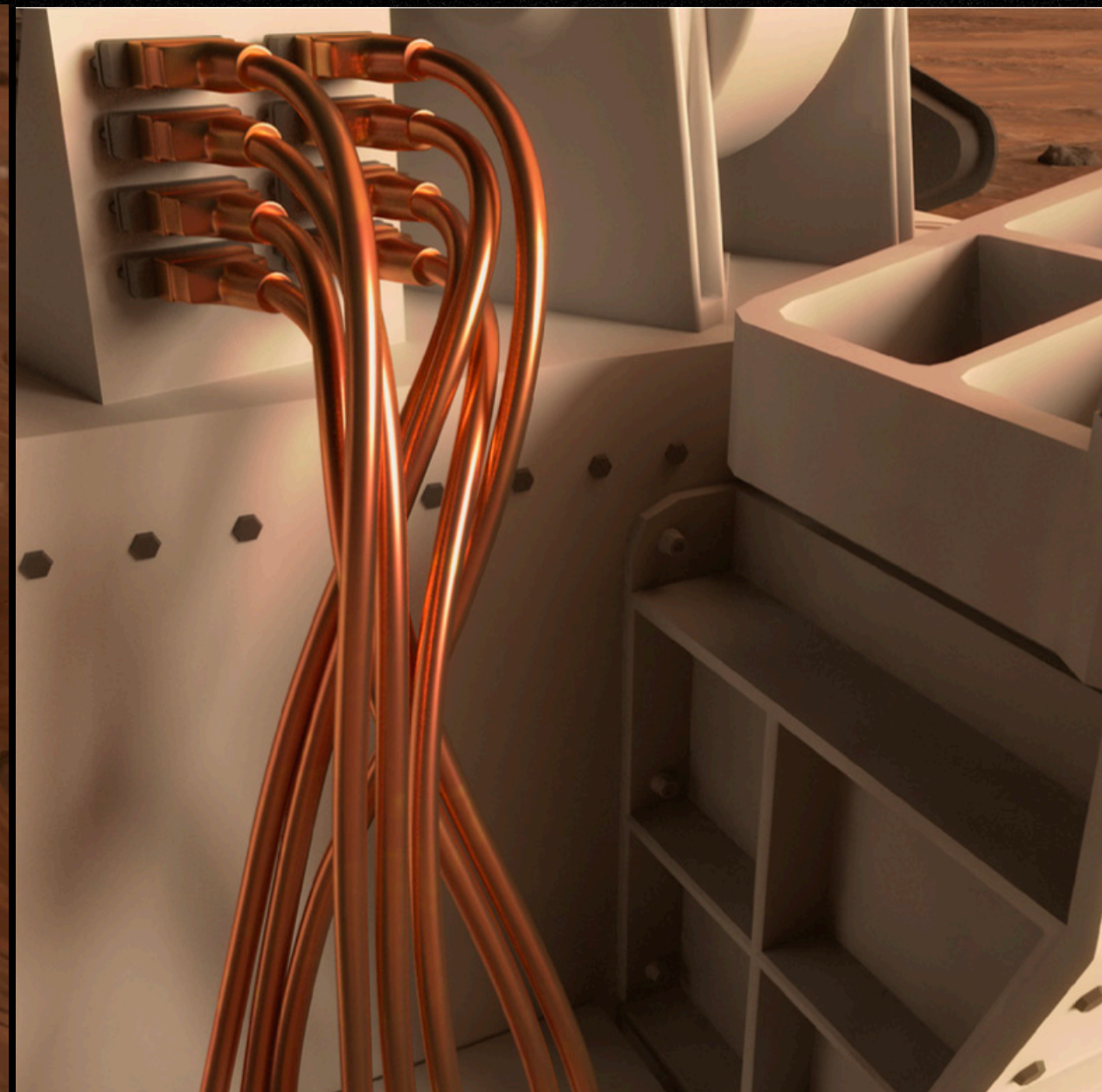
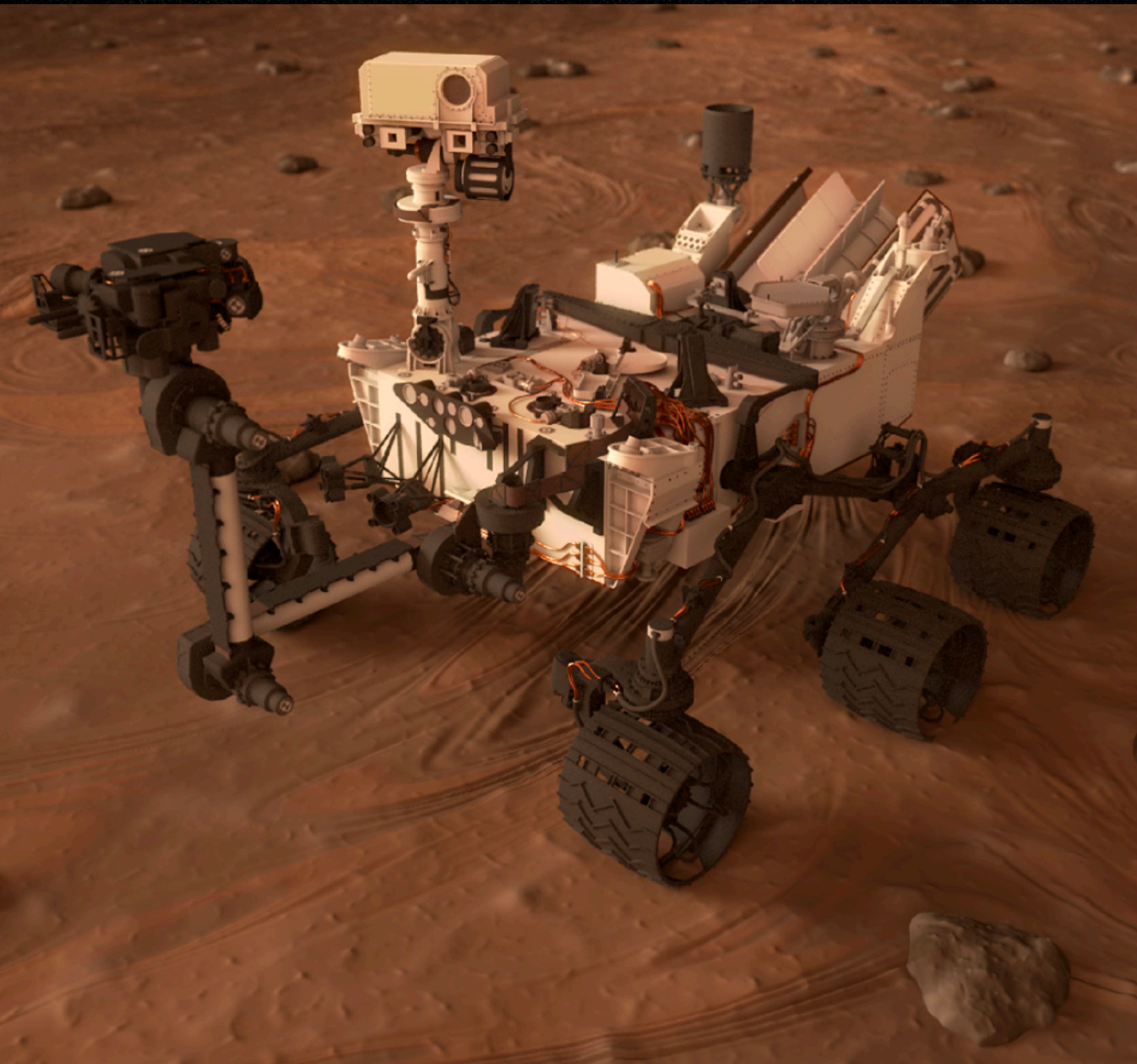


analyze *locally*  
in region of  
interest.

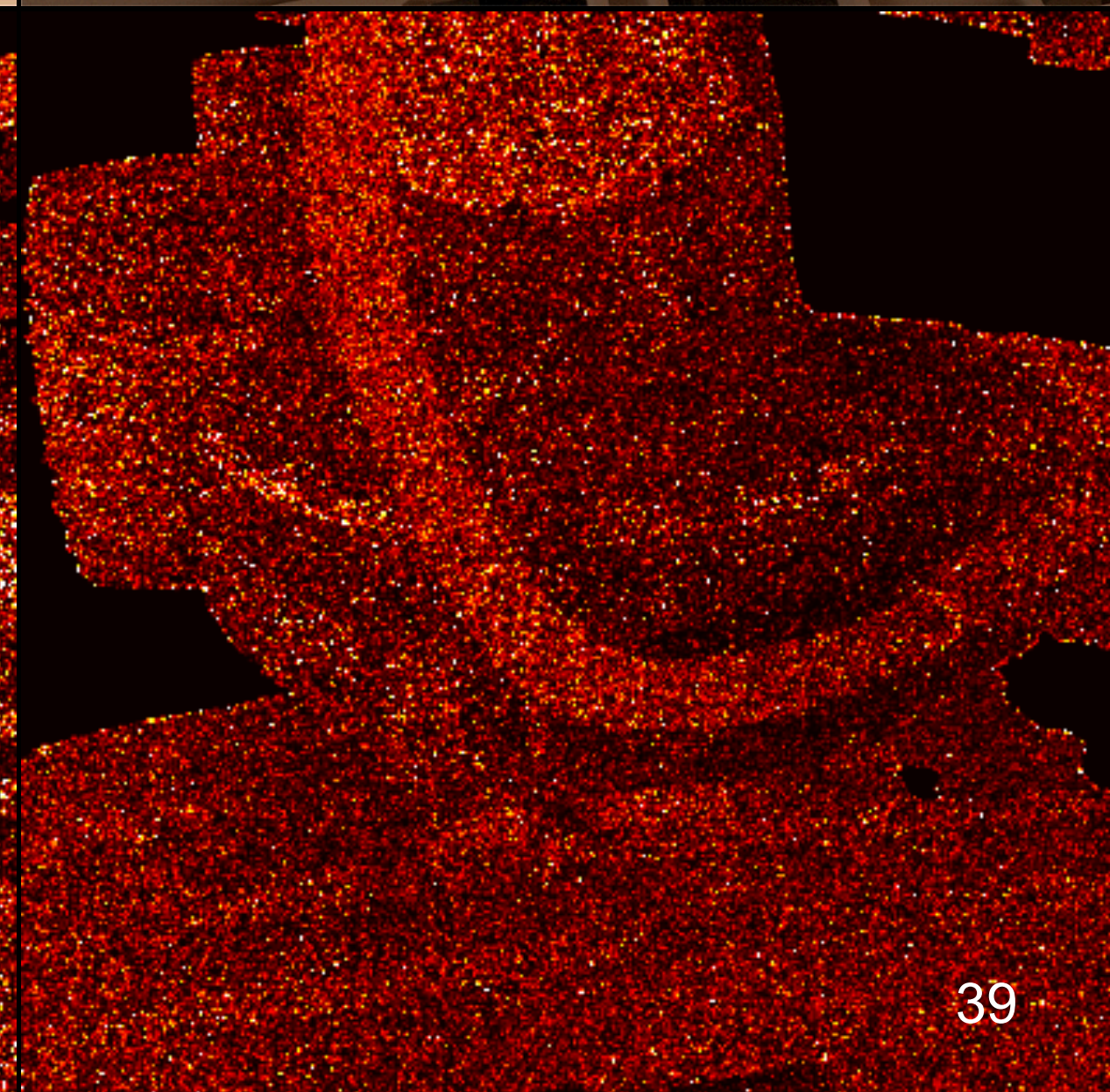
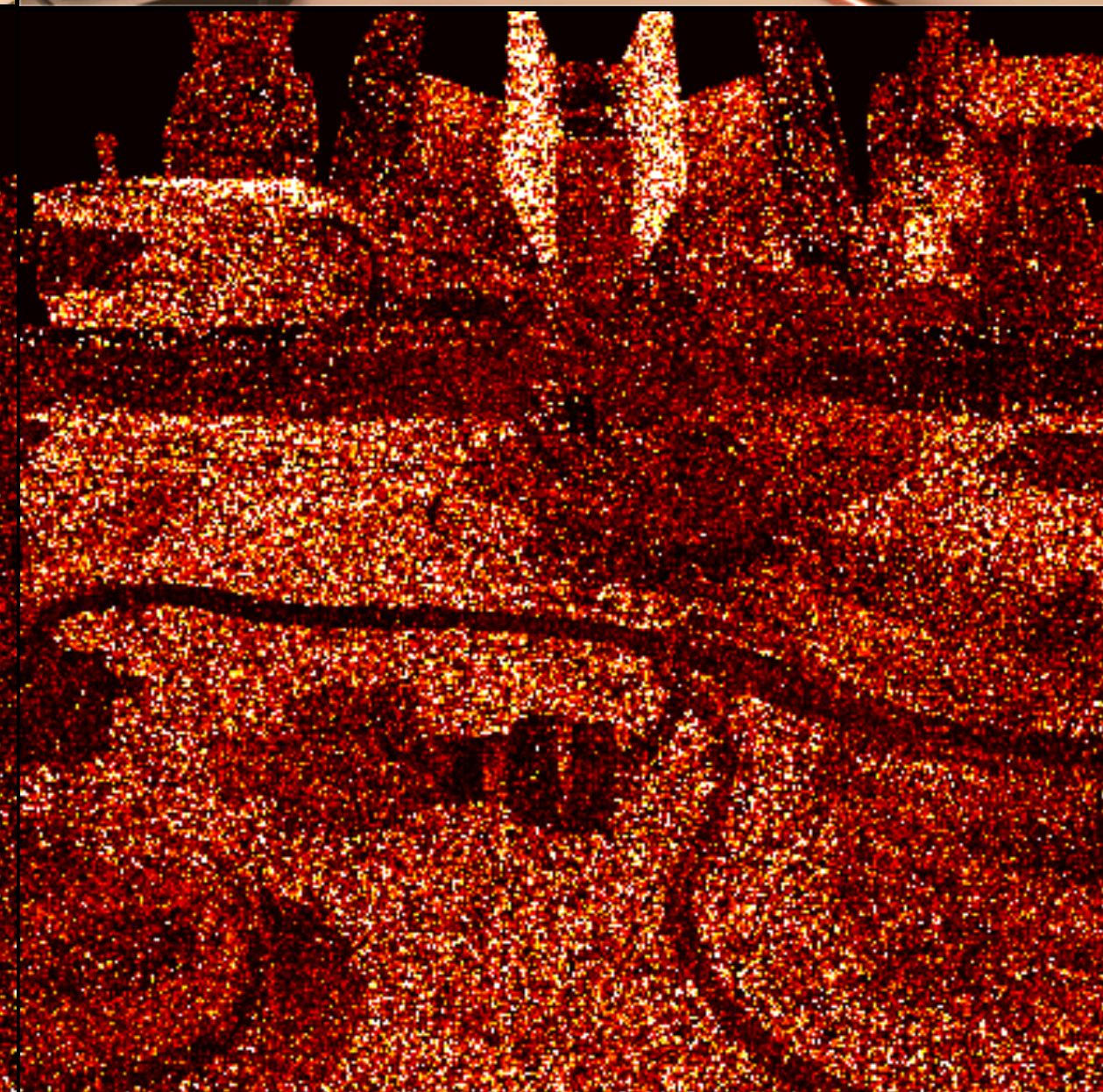
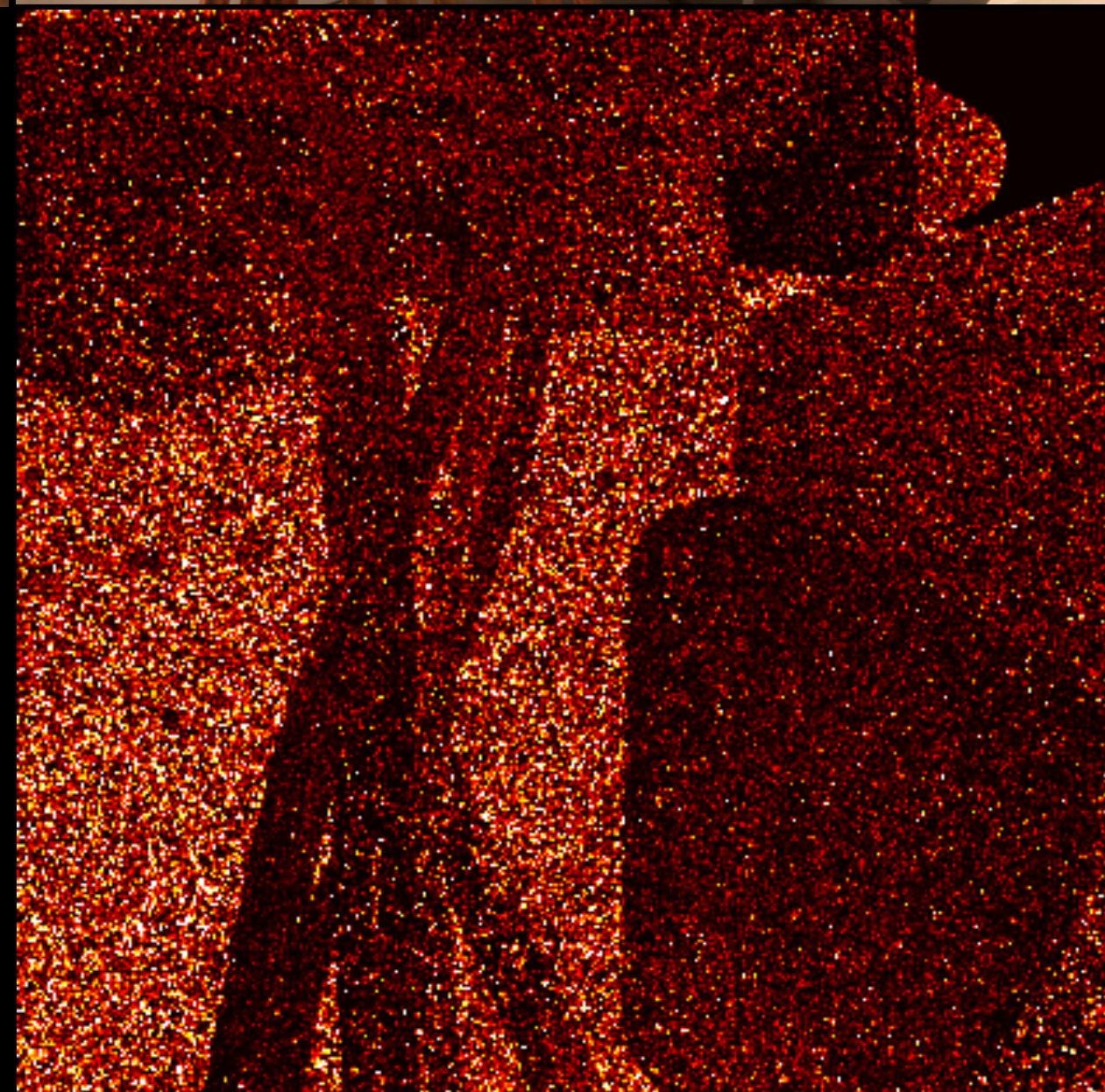




# Monte Carlo thermal simulation

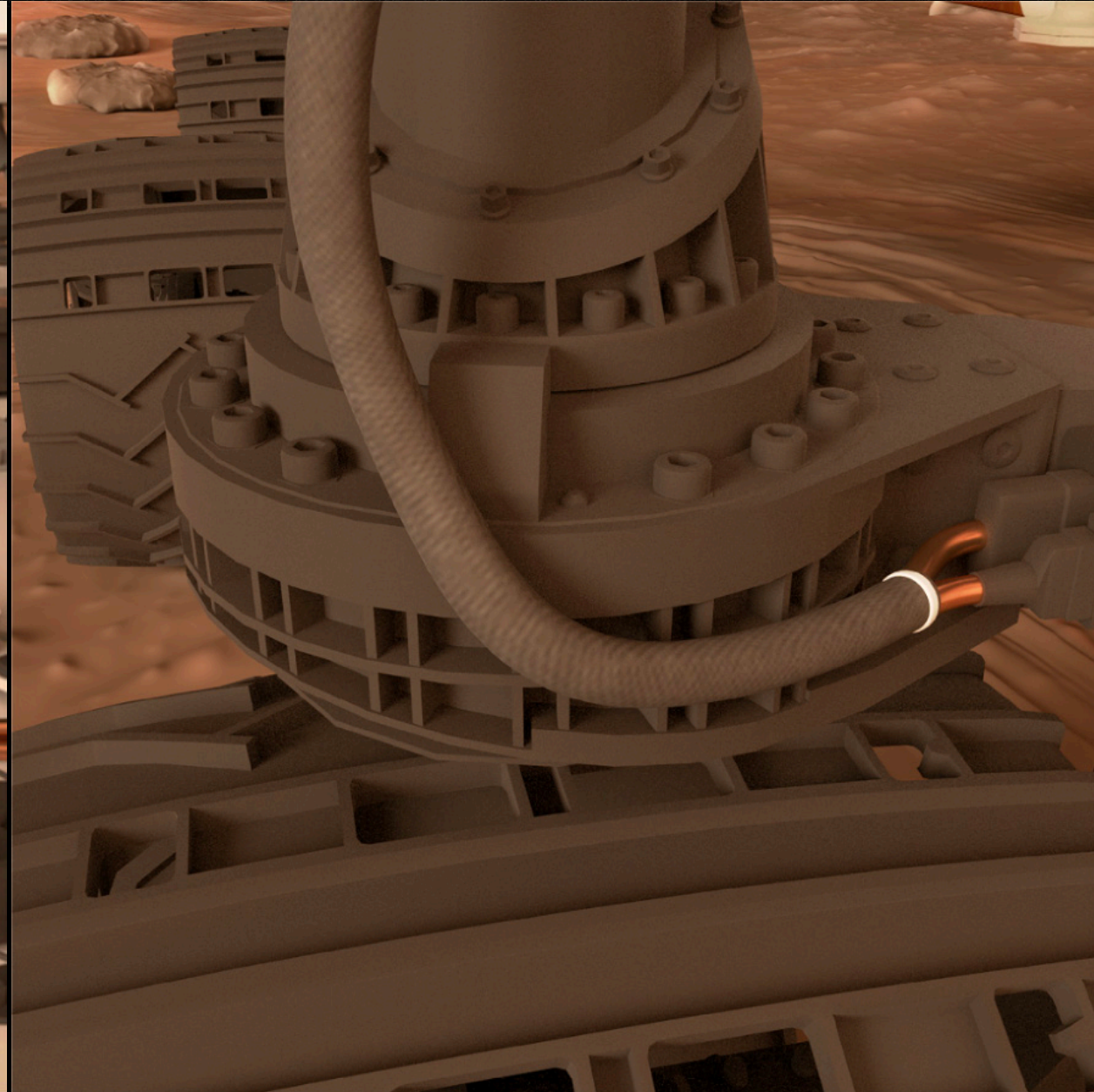
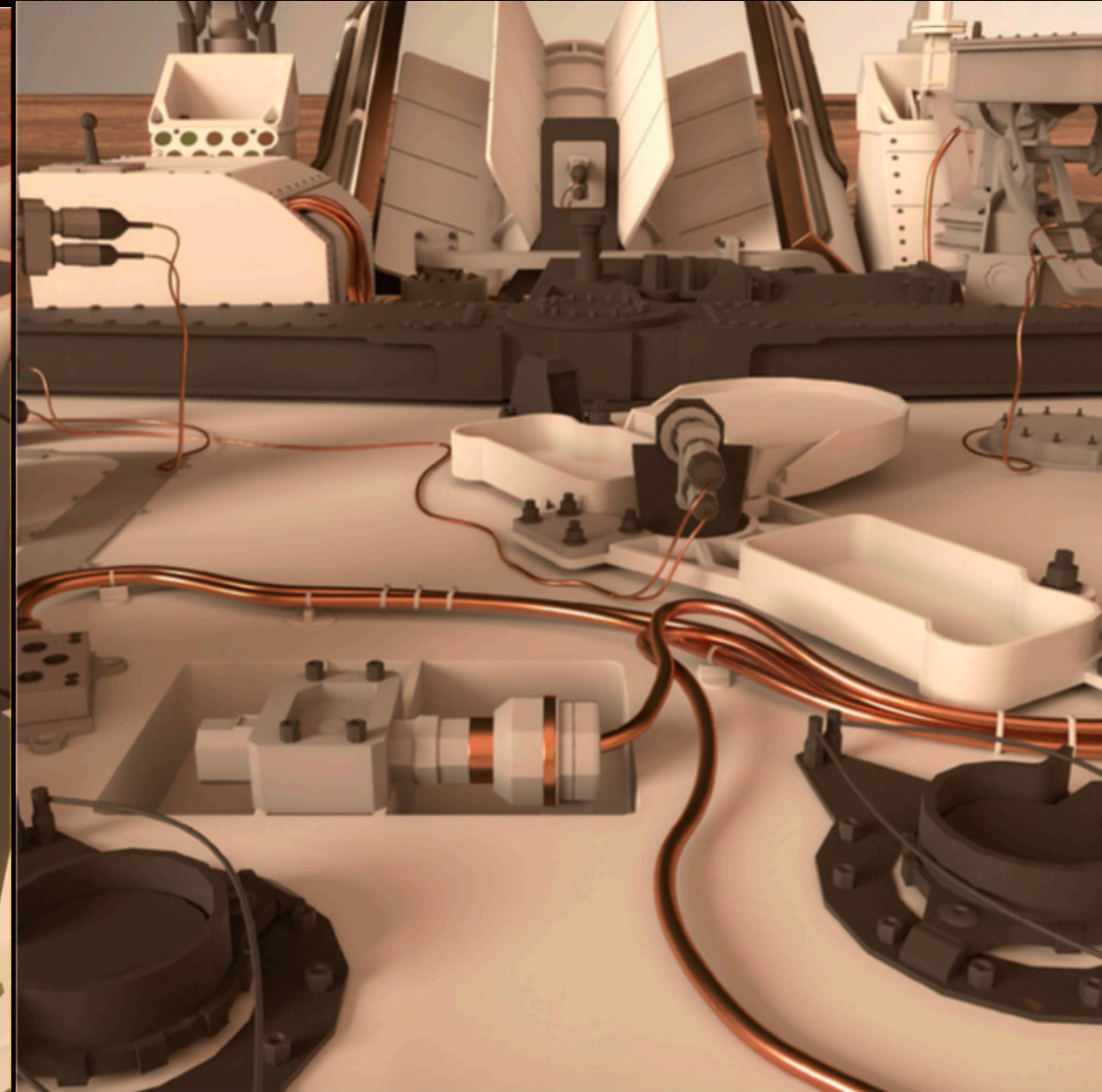
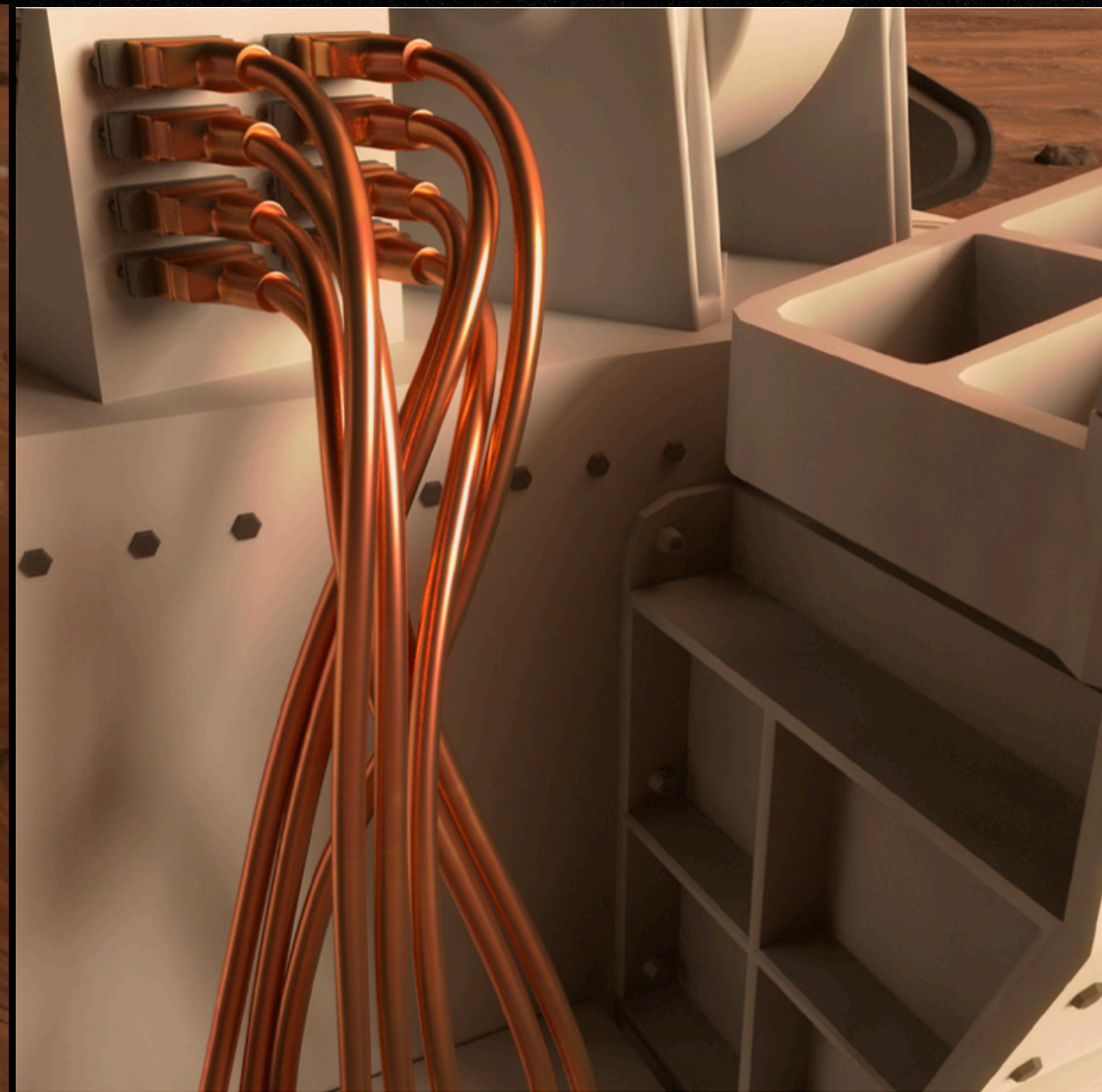
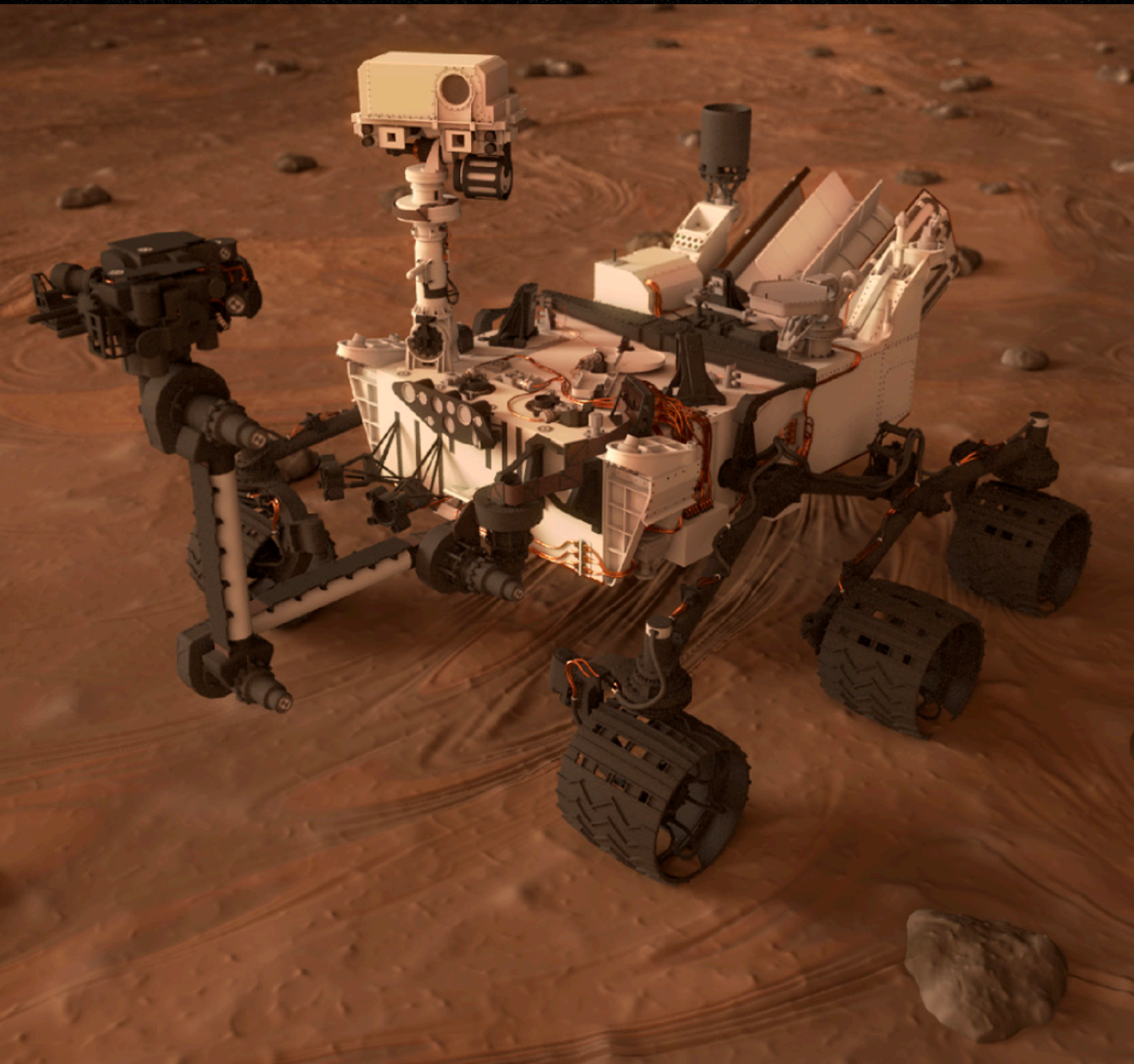


analyze *locally*  
in region of  
interest.





# Monte Carlo thermal simulation



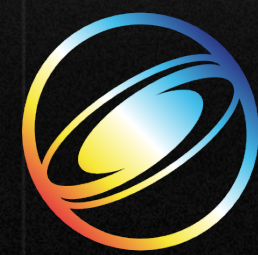
analyze *locally*  
in region of  
interest.



**only simulate what you see!**

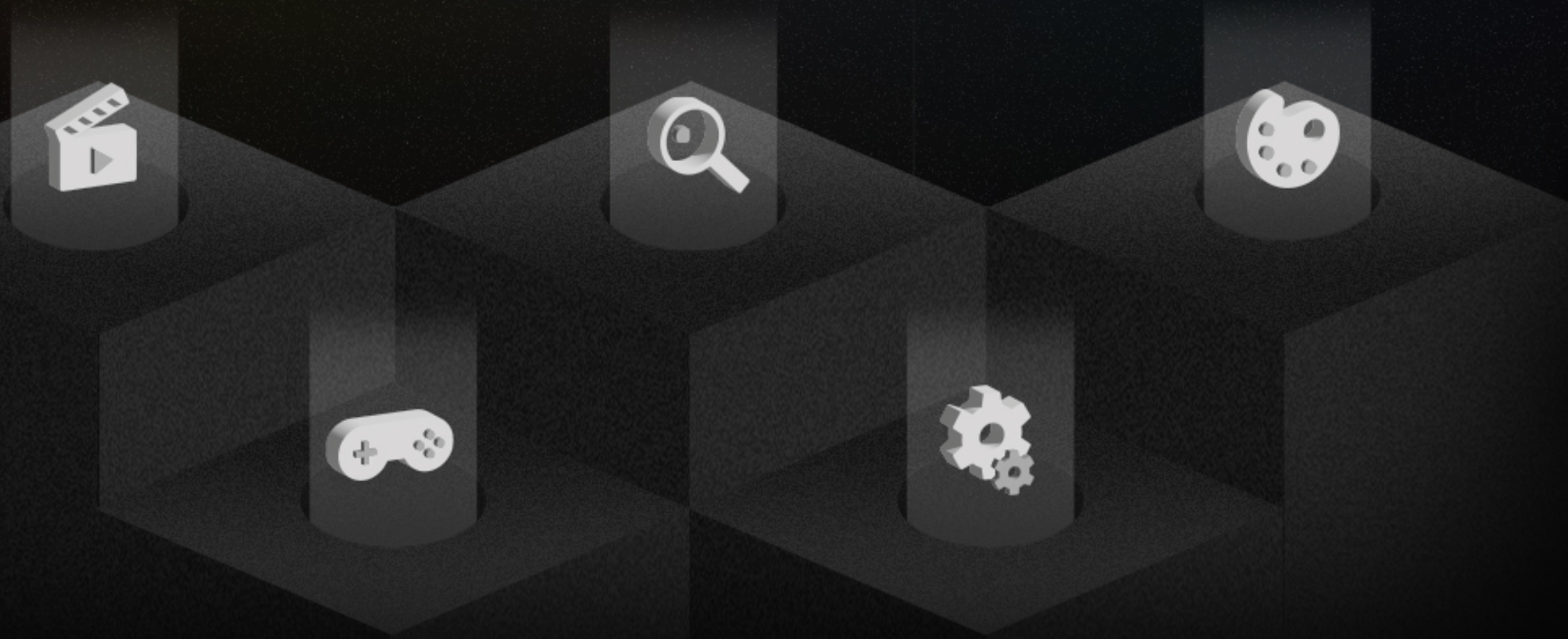


SIGGRAPH  
2024



**SIGGRAPH 2024**  
DENVER+ 28 JUL — 1 AUG

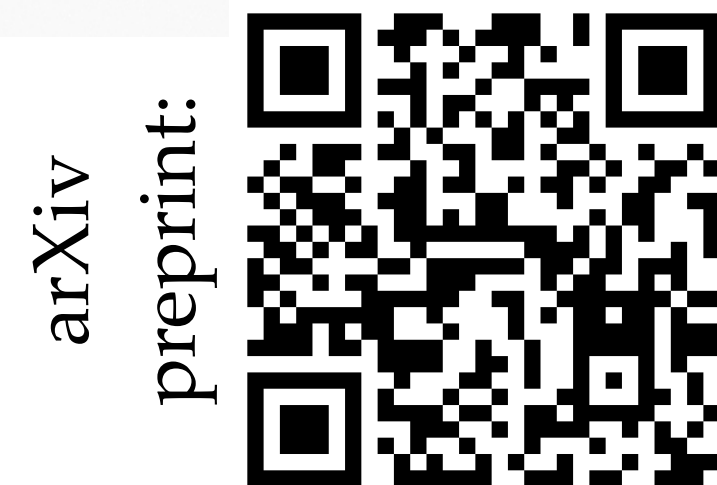
# WHAT'S NEXT?





# differentiable solvers

Goal: recover shape given measurements, e.g., temperature

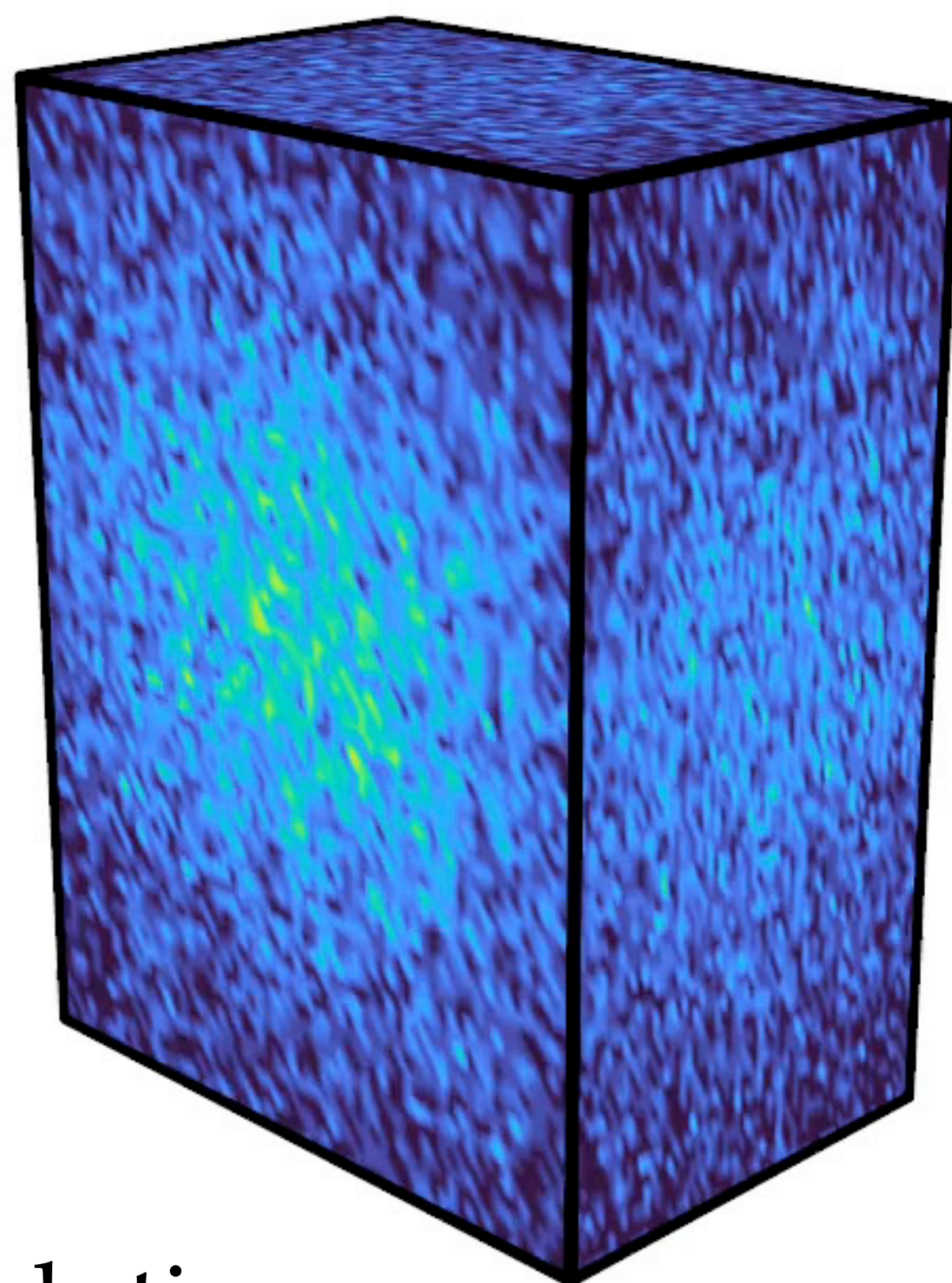
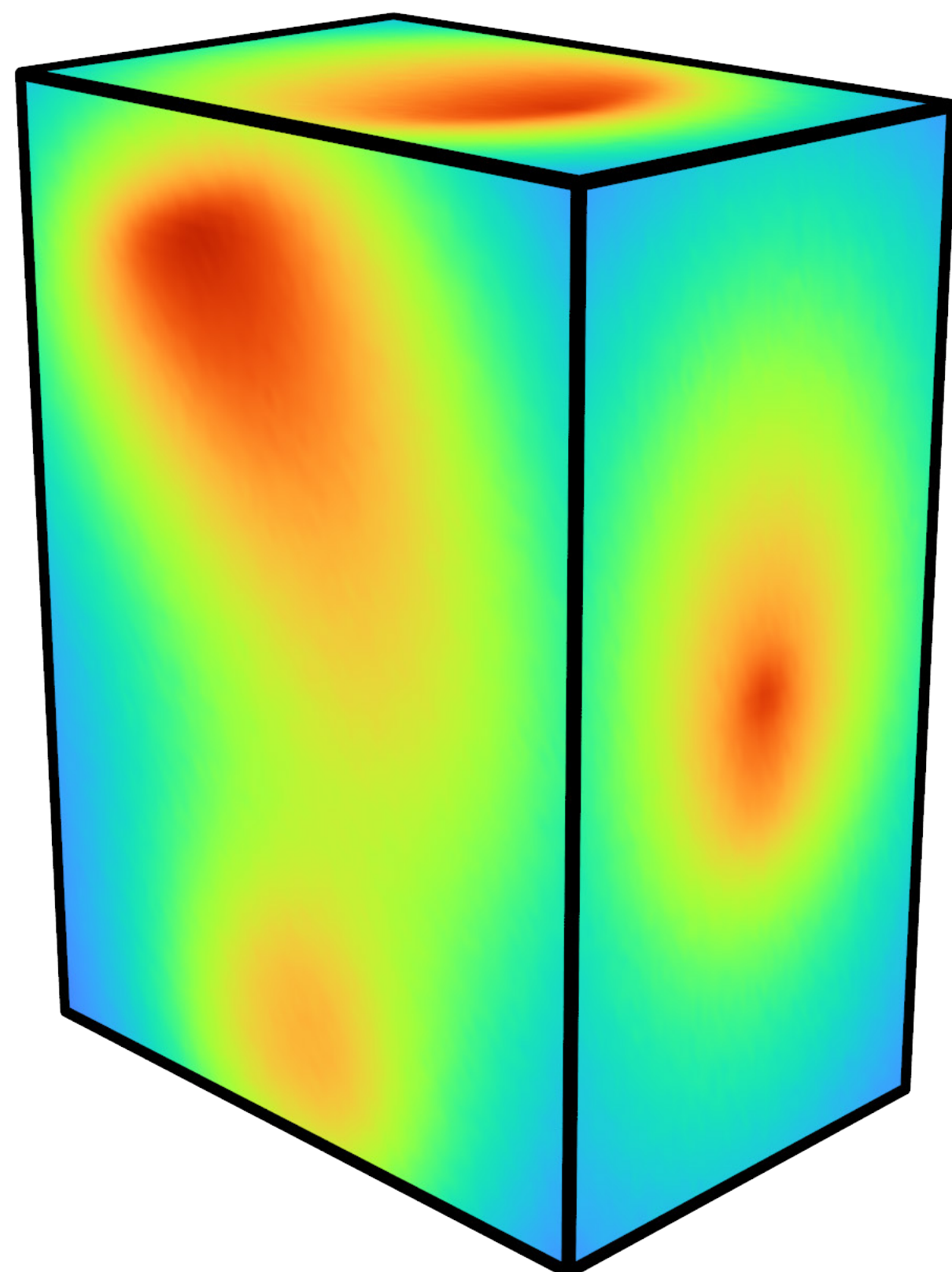


reference

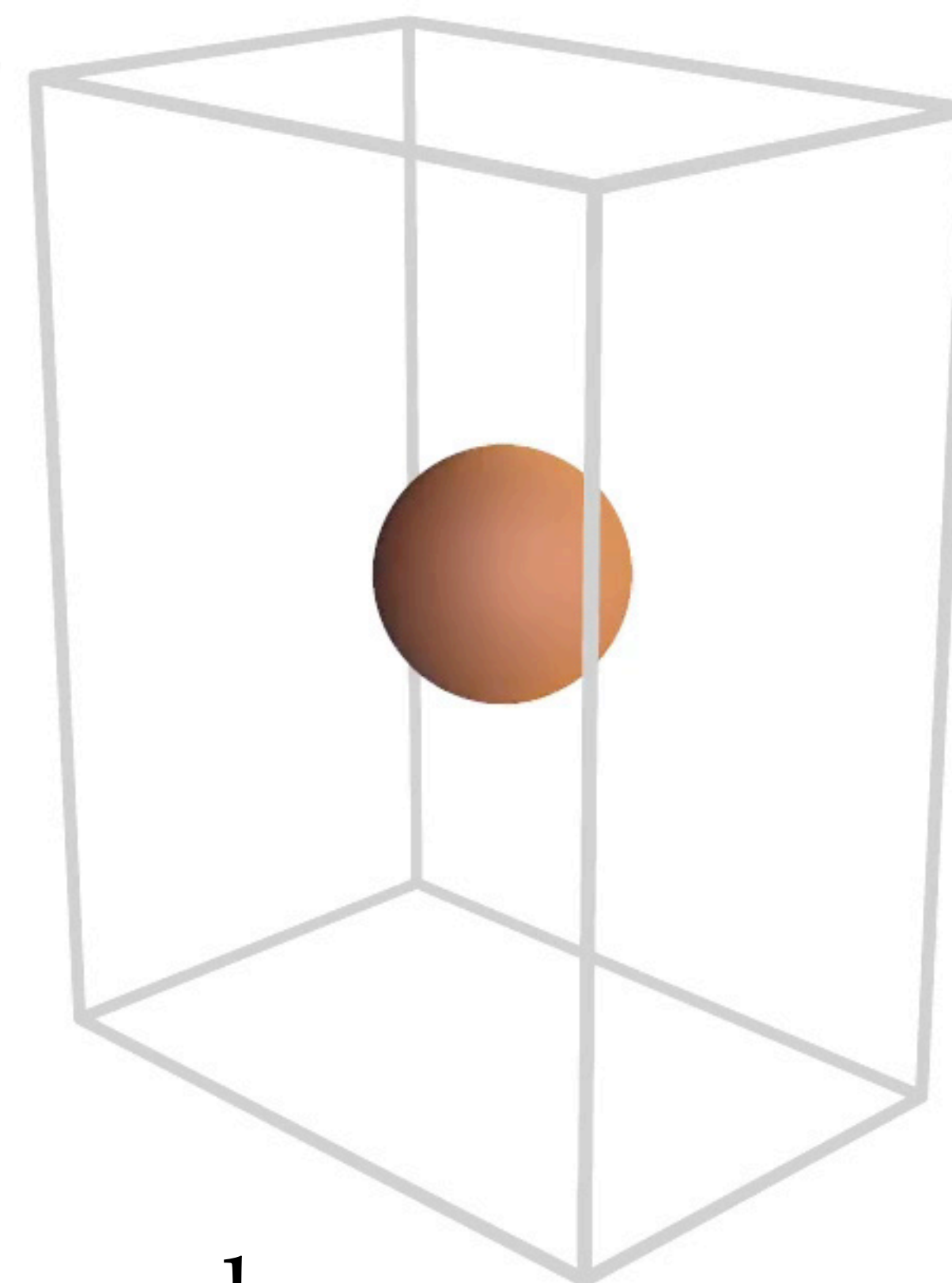
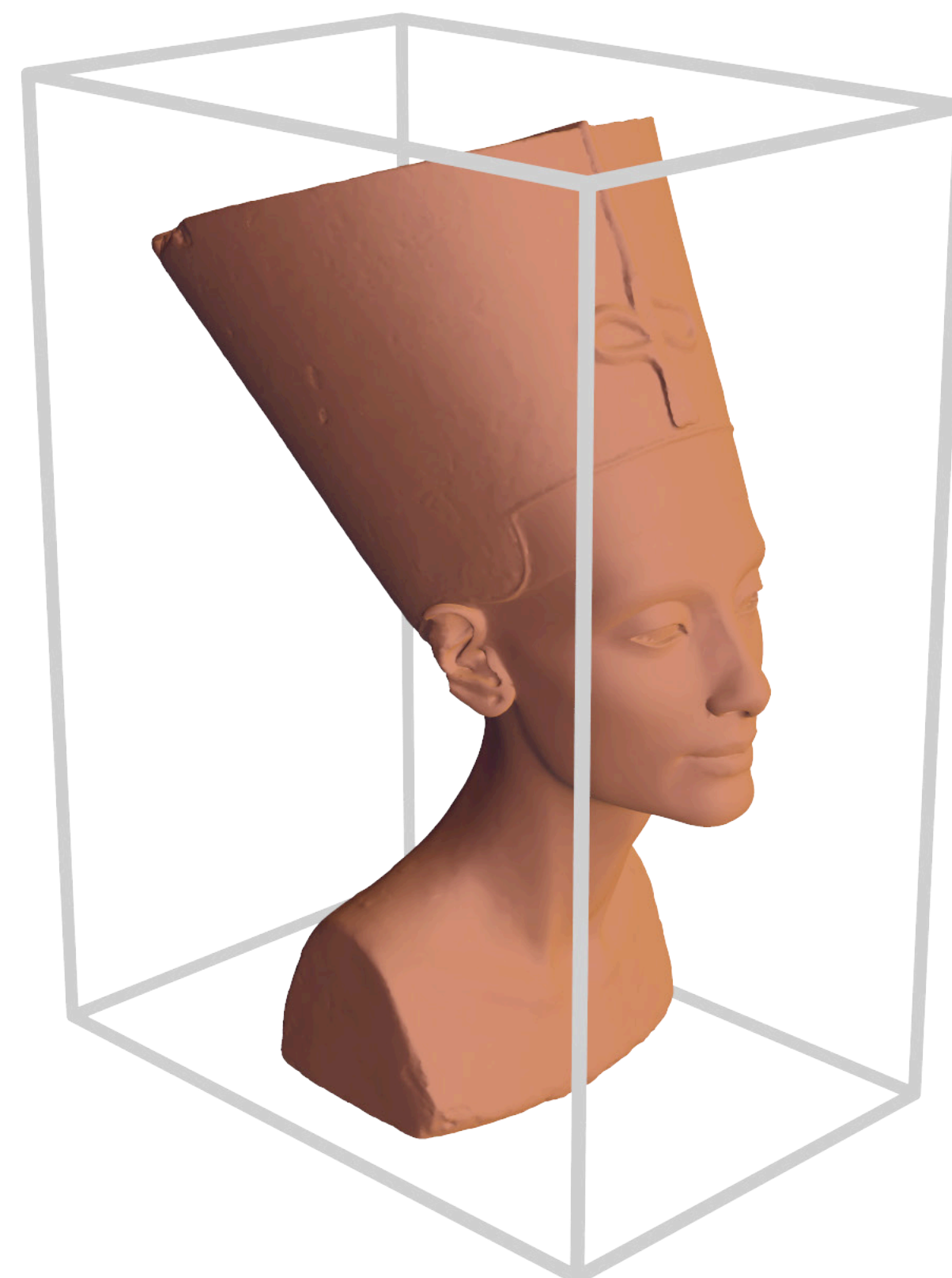
optimized

reference

optimized



PDE solution

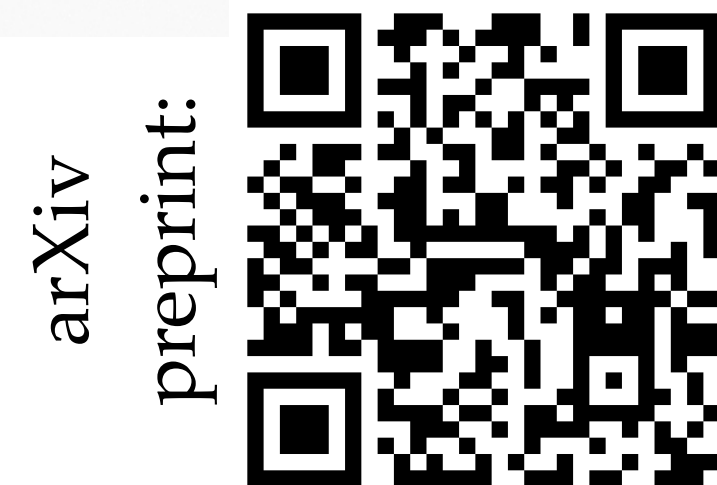


triangle mesh



# differentiable solvers

**Goal:** recover shape given measurements, e.g., temperature

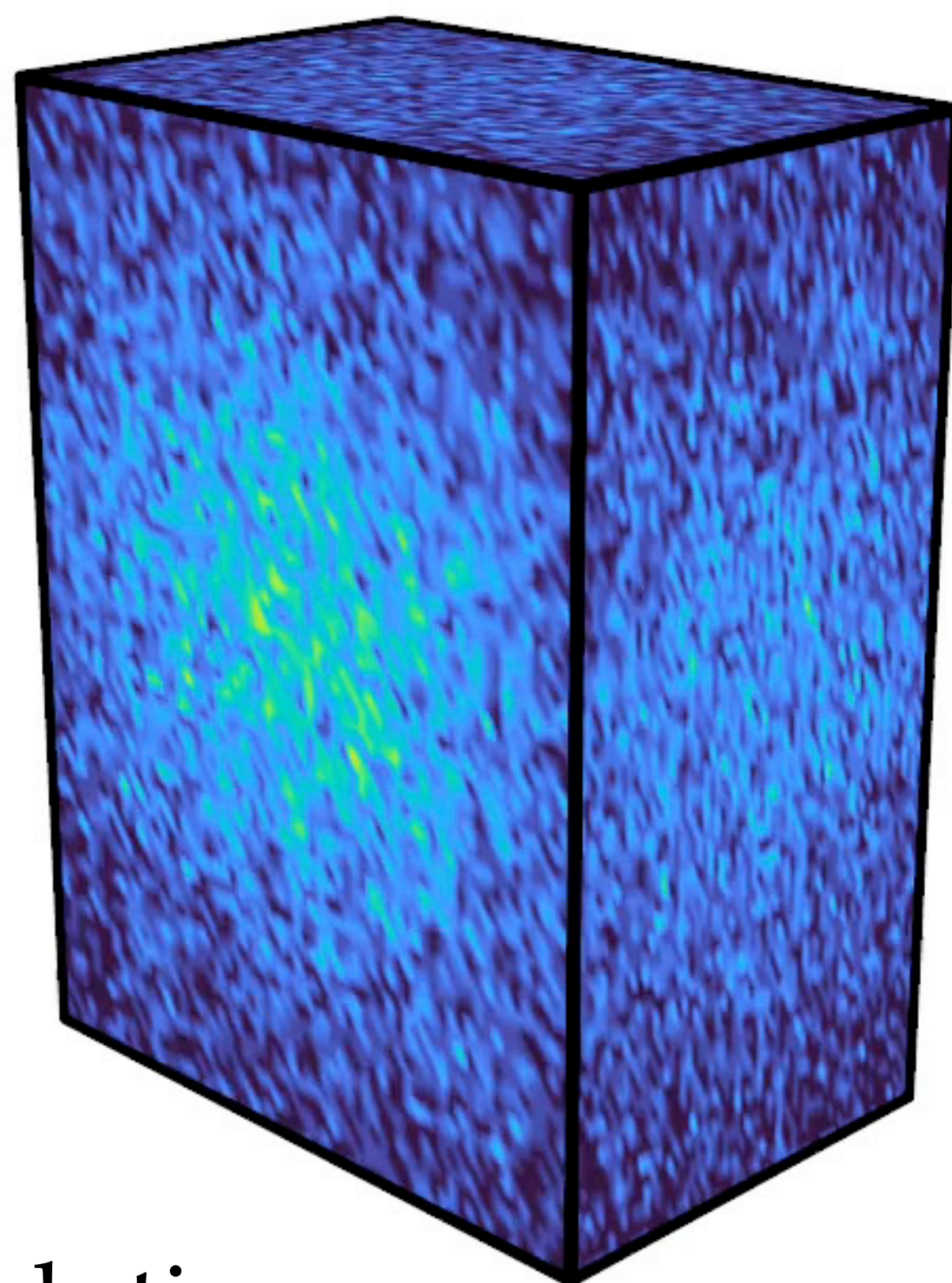
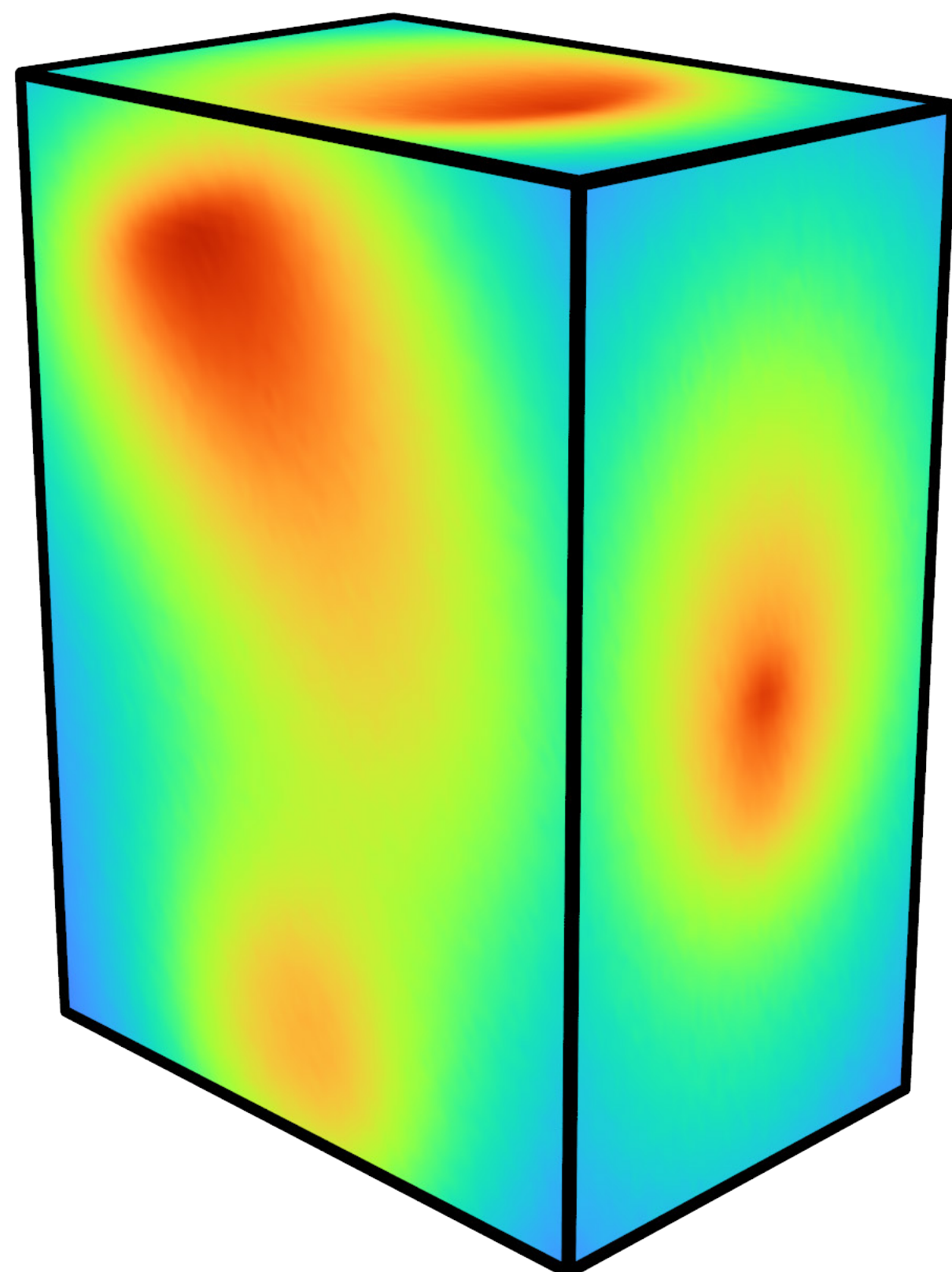


reference

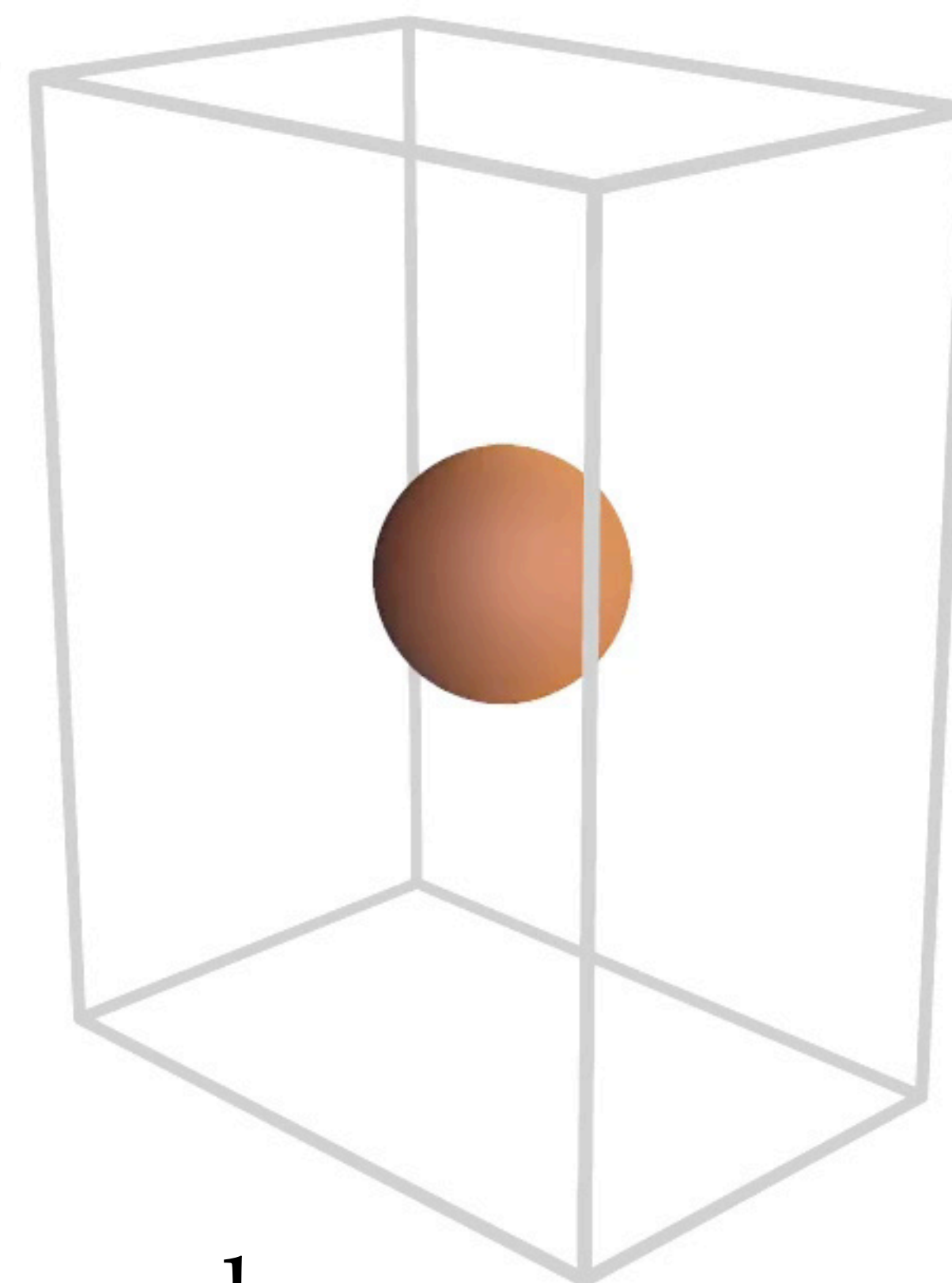
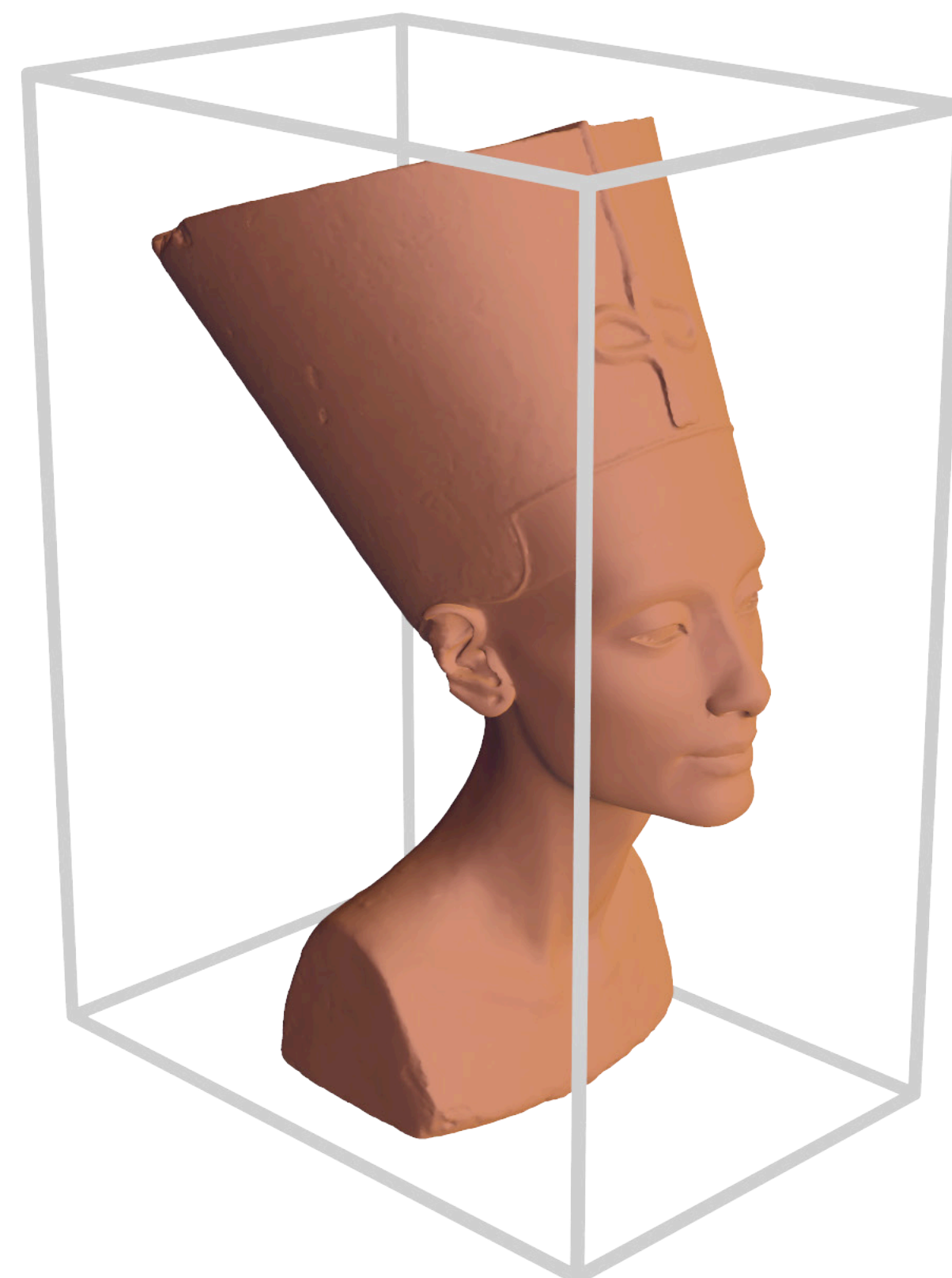
optimized

reference

optimized



PDE solution

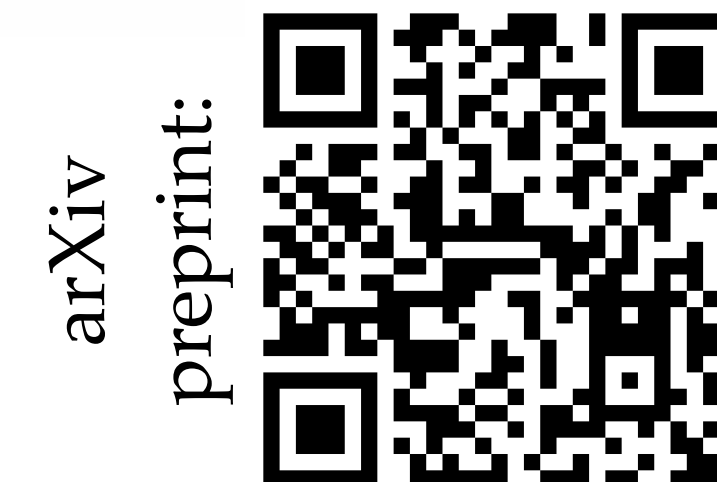


triangle mesh



## boundary and source terms [Yilmazer et al. 2022]

## shape derivatives [Yu et al. 2024, Miller et al. 2024]



arXiv  
preprint:

### Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators

EKREM FATIH YILMAZER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland  
DELIO VICINI, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland  
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Modeling physical phenomena like heat transport and diffusion is crucially dependent on the numerical solution of partial differential equations (PDEs). A PDE solver finds the solution given coefficients and a boundary condition, whereas an inverse PDE solver goes the opposite way and reconstructs these inputs from an existing solution. In this article, we investigate techniques for solving inverse PDE problems using a gradient-based methodology.

Conventional PDE solvers based on the finite element method require a domain meshing step that can be fragile and costly. Grid-free Monte Carlo methods instead stochastically sample paths using variations of the walk on spheres algorithm to construct an unbiased estimator of the solution. The uncanny similarity of these methods to physically-based rendering algorithms has been observed by several recent works.

In the area of rendering, recent progress has led to the development of efficient unbiased derivative estimators. They solve an adjoint form of the problem and exploit arithmetic invertibility to compute gradients using a constant amount of memory and linear time complexity.

Could these two lines of work be combined to compute cheap parametric derivatives of a grid-free PDE solver? We investigate this question and present preliminary results.

CCS Concepts: • Mathematics of computing → Partial differential equations; • Computing methodologies → Rendering.

Additional Key Words and Phrases: walk on spheres, Monte Carlo, differentiable simulation, path replay backpropagation

#### 1 INTRODUCTION

Many physical phenomena are naturally described using partial differential equations (PDEs). For example, the heat equation models the spread of thermal energy in a potentially heterogeneous material. Solvers that numerically approximate solutions of such PDEs are in widespread use. We pursue the opposite direction in this article, which is known as an *inverse PDE problem*: estimating unknown parameters from observations of the solution. This set of unknown parameters could include various PDE coefficients, boundary conditions, and even the shape of the domain.

Such problems arise in diverse scientific and engineering contexts, for example to determine the physical parameters of a thermal conductor from measurements [Cannon 1964]. Electrical impedance tomography [Cheney et al. 1999] seeks to reconstruct the interior of a living organism. Electrodes provide measurements of the electric field, which is influenced by the tissue's conductivity, impedance, and dielectric permittivity.

Our approach entails differentiating the solver and recovering the unknown parameters using gradient descent. However, one issue with conventional PDE solvers based on the finite element method (FEM) is that they require a meshing step that can be fragile

and computationally costly. An alternative are Monte Carlo PDE solvers based on the *walk on spheres* (WoS) [Muller 1956]. These *grid-free* methods sample random paths in the domain to compute unbiased estimates of the solution. Grid-free solvers have recently attracted significant attention in the computer graphics community, partly owing to the remarkable similarities to Monte Carlo rendering methods [Sawhney and Crane 2020] and the algorithmic synergies that this creates [Sawhney et al. 2022; Qi et al. 2022].

A common issue with gradient-based optimization is that the standard approach for reverse-mode differentiation (known as *backpropagation*) reverses all data dependencies of an underlying computation. When applied to the WoS algorithm, this means that intermediate results of a large number of iterations would need to be stored to enable the subsequent differentiation.

In the field of rendering, recent progress has led to the development of *differentiable rendering* methods [Gkioulekas et al. 2013; Li et al. 2018; Nimier-David et al. 2019] that estimate parametric derivatives of complete light transport simulations. A similar issue arises here as well: light paths can potentially be very long, particularly in highly-scattering media, which makes naïve reverse-mode differentiation prohibitively memory-intensive. Methods like *radiative backpropagation* [Nimier-David et al. 2020] and *path replay backpropagation* [Vicini et al. 2021] cast the differentiation step into an independent simulation of “derivative light” to address this issue. The latter project solves an adjoint version of the underlying equation and furthermore exploits arithmetic invertibility in the computation to differentiate using a constant amount of memory and a runtime cost that is linear in the number of path vertices.

Given these striking similarities, could a similar approach be useful to compute reverse-mode derivatives of grid-free Monte Carlo solvers? We show that this is indeed the case and that this combination yields an unbiased derivative estimator in the same complexity class. The paper presents preliminary results on synthetic inverse problems. We make no claims about the utility of such an approach for solving concrete inverse-PDE problems but find it a promising direction for future work.

2 METHOD  
2.1 Background  
Inverse PDE problems. We seek to solve an inverse PDE problem of the form:

$$\hat{\pi} = \arg \min_{\pi} \ell(u(\pi)), \quad (1)$$

where  $u(\pi)$  is the solution of a PDE parameterized by the vector  $\pi$  containing the boundary values, source terms, etc. The function  $\ell$  is a differentiable objective function. In the simplest case, this could be the  $L_2$  difference between the solution of the PDE and a reference solution evaluated at a set of locations spread throughout

### A Differential Monte Carlo Solver For the Poisson Equation

Zihan Yu  
zihay19@uci.edu  
University of California, Irvine & NVIDIA  
USA

Lifan Wu  
lifanw@nvidia.com  
NVIDIA  
USA

Zhiqian Zhou  
zhiqiaz8@uci.edu  
University of California, Irvine  
USA

Shuang Zhao  
shz@ics.uci.edu  
University of California, Irvine & NVIDIA  
USA

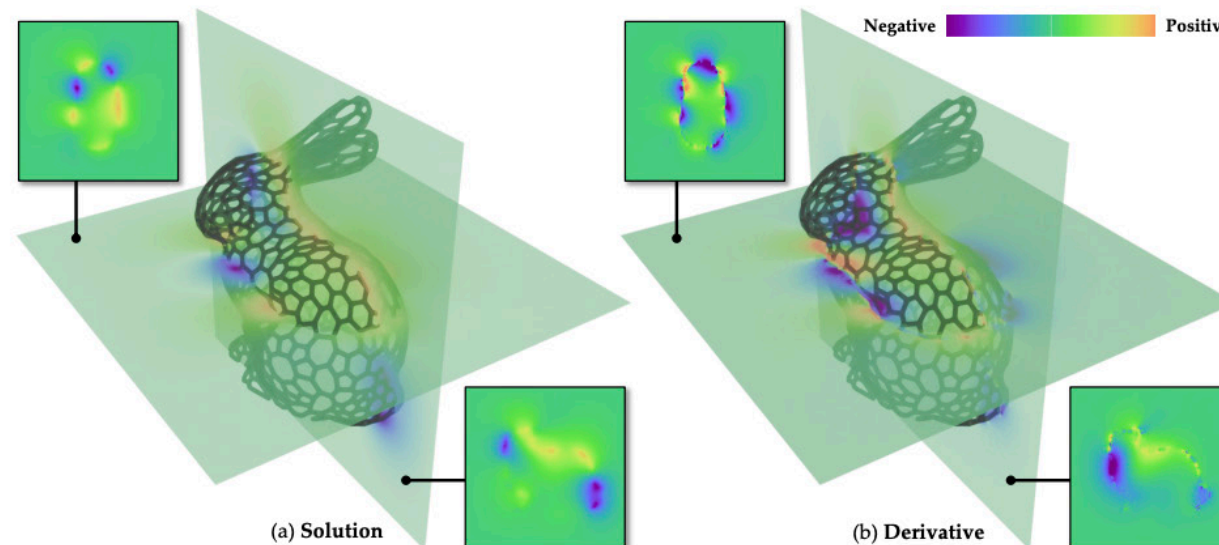


Figure 1: We introduce a new grid-free technique to estimate derivatives of solutions to the Poisson equation with respect to arbitrary parameters including domain shapes. This example includes a 3D Laplace problem with Dirichlet boundary conditions on a wired bunny shape. We visualize the solution to this problem in two cross-sectional planes in (a) and the derivative of this solution (with respect to the translation of the bunny) estimated with our method in (b).

#### ABSTRACT

The Poisson equation is an important partial differential equation (PDE) with numerous applications in physics, engineering, and computer graphics. Conventional solutions to the Poisson equation require discretizing the domain or its boundary, which can be very expensive for domains with detailed geometries. To overcome this challenge, a family of grid-free Monte Carlo solutions has recently been developed. By utilizing walk-on-sphere (WoS) processes, these

techniques are capable of efficiently solving the Poisson equation over complex domains.

In this paper, we introduce a general technique that differentiates solutions to the Poisson equation with Dirichlet boundary conditions. Specifically, we devise a new boundary-integral formulation for the derivatives with respect to arbitrary parameters including shapes of the domain. Further, we develop an efficient walk-on-spheres technique based on our new formulation—including a new approach to estimate normal derivatives of the solution field. We demonstrate the effectiveness of our technique over baseline meth-

### Differential Walk on Spheres

BAILEY MILLER, Carnegie Mellon University, USA  
ROHAN SAWHNEY, NVIDIA, USA  
KEENAN CRANE, Carnegie Mellon University, USA  
IOANNIS GKIIOULEKAS, Carnegie Mellon University, USA

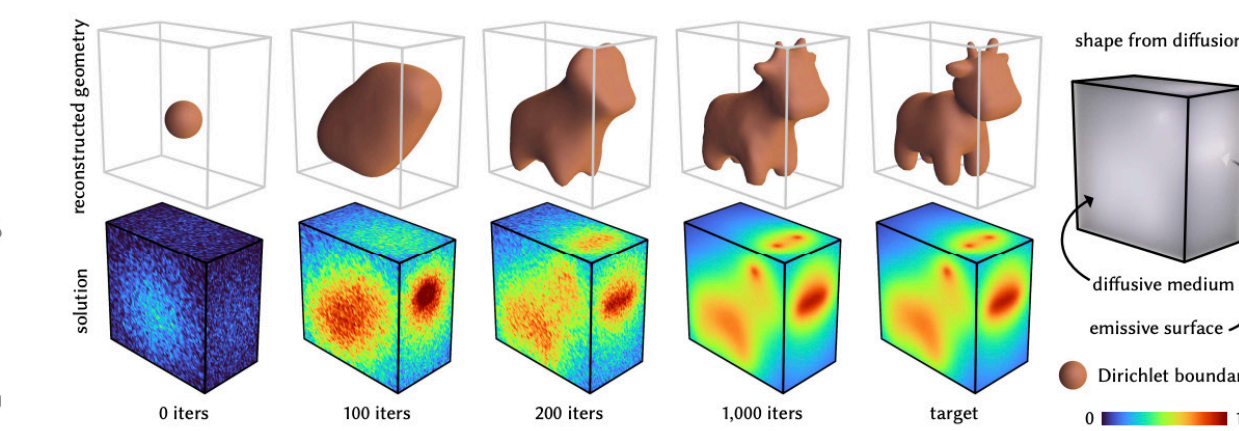


Figure 1. For a given PDE, our differential walk on spheres algorithm makes it possible to differentiate solution values with respect to problem parameters. Here we consider an inverse problem where we recover the shape of an emissive object from its observed diffusion profile on the boundary of a box, via gradient-based optimization. Unlike conventional mesh- or grid-based approaches, we can evaluate derivatives at points of interest without needing to compute a global solution (here, only at the observed points).

We introduce a Monte Carlo method for evaluating derivatives of the solution to a partial differential equation (PDE) with respect to problem parameters (such as domain geometry or boundary conditions). Derivatives can be evaluated at arbitrary points without performing a global solve, or constructing a volumetric grid or mesh. The method is hence well-suited to inverse problems with complex geometry, such as PDE-constrained shape optimization. Like other *walk on spheres* (WoS) algorithms, our method is trivial to parallelize, and is agnostic to boundary representation (meshes, splines, implicit surfaces etc.), supporting large topological changes. We focus in particular on screened Poisson equations, which model diverse problems from scientific and geometric computing. As in differentiable rendering, we jointly estimate derivatives with respect to all parameters—hence, cost does not grow significantly with parameter count. In practice, even noisy derivative estimates exhibit fast, stable convergence for stochastic gradient-based optimization—as we show via examples from thermal design, shape from diffusion, and computer graphics.

CCS Concepts: • Computing methodologies → Physical simulation; Rendering.

Additional Key Words and Phrases: Walk on spheres, differentiable simulation

Authors' addresses: Bailey Miller, bmmiller@andrew.cmu.edu, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA; Rohan Sawhney, rsawhney@nvidia.com, NVIDIA, 2788 San Tomas Expy, Santa Clara, CA, 95051, USA; Keenan Crane, kcrane@cs.cmu.edu, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA; Ioannis Gkioulekas, gkioule@cs.cmu.edu, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA.

Reference Format:  
Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024. Differential Walk on Spheres. *Technical Report 1*, 1 (May 2024), 17 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

#### 1 INTRODUCTION

Which shape best explains observed physical behavior? How can one design shapes that maximize (or minimize) a target physical quantity? Such *inverse problems* are fundamental to numerous challenges in science and engineering. For instance, one might need to assess damage to an airplane wing using indirect thermal measurements [Zalameda and Parker 2014], or infer the shape of a tumor through deep layers of tissue [Arridge 1999]. Likewise, one might seek to design circuit geometry that maximizes dissipation of heat [Zhan et al. 2008], airfoils that generate prescribed lift [Hicks and Henne 1977], or lightweight structures that withstand significant load [Allaire et al. 2014]. To solve such problems, one must be able to efficiently and accurately differentiate solutions to partial differential equations (PDEs) with respect to the shape of the domain, or its boundary conditions [Hadamard 1908; Cea et al. 1973]. However, for problems with complex geometry, even just solving such PDEs can be a non-trivial task. The *walk on spheres* (WoS) method [Muller 1956; Sawhney and Crane 2020] and its recent extensions [Sawhney et al. 2022, 2023; Miller et al. 2024] provide *grid-free* alternatives to traditional PDE

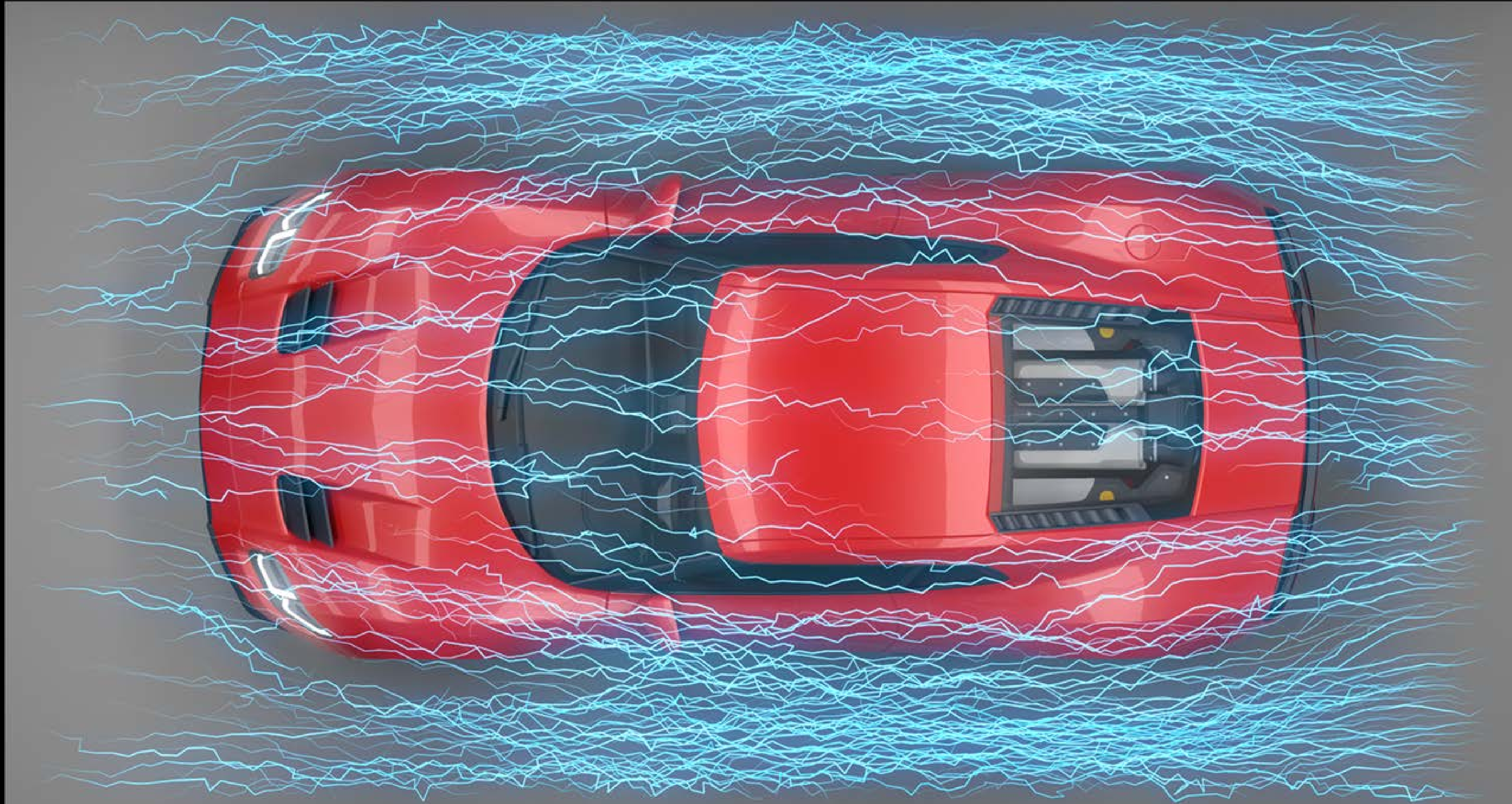
arXiv:2208.02114v1 [cs.GR] 3 Aug 2022

arXiv:2405.12964v2 [cs.GR] 27 May 2024





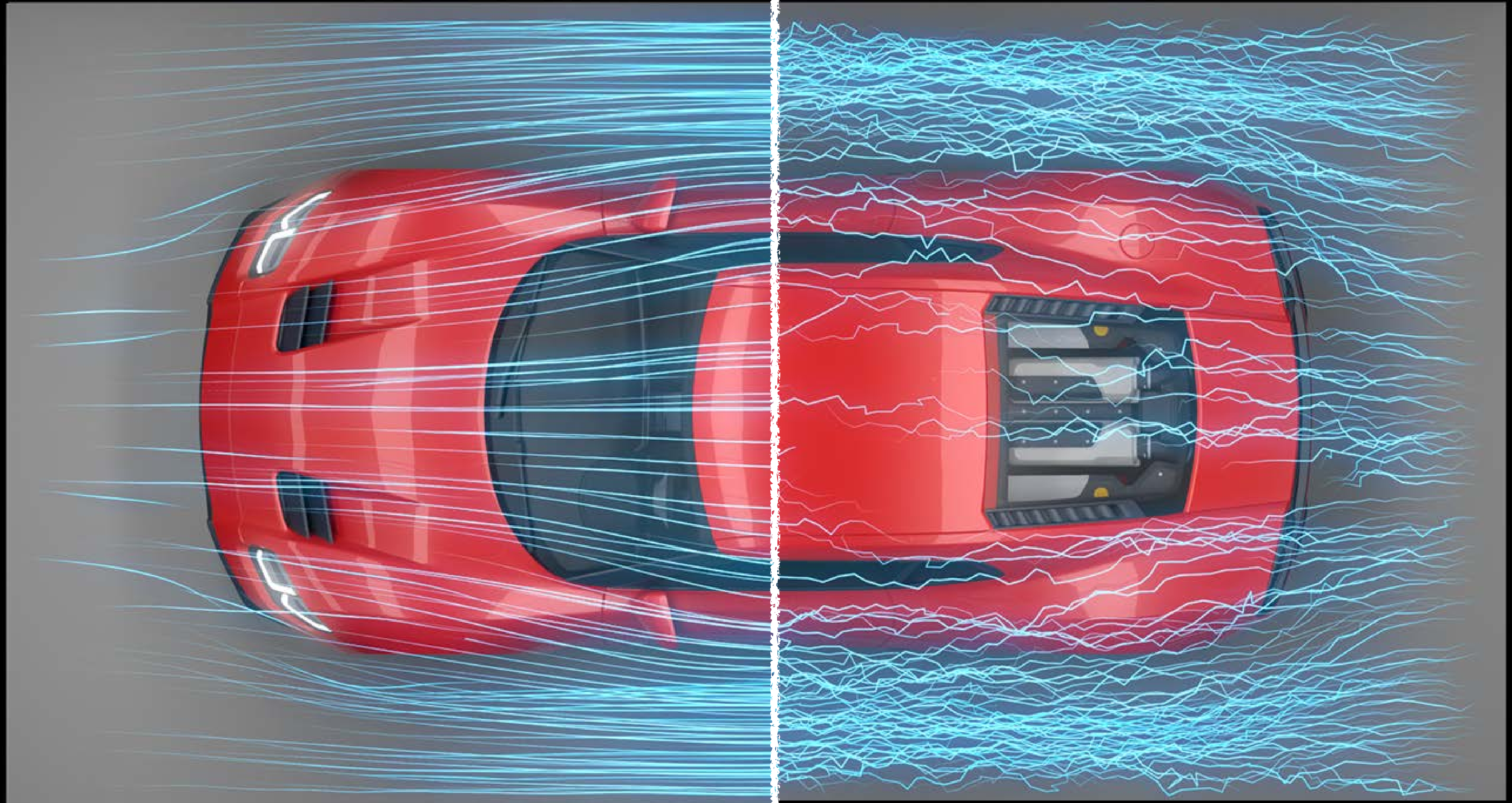
variance reduction, variance reduction, var..



pointwise estimator



variance reduction, variance reduction, var..



boundary value caching [Miller et al. 2023]

pointwise estimator



# variance reduction, variance reduction, var..

## caching methods

## sampling methods

## control variates

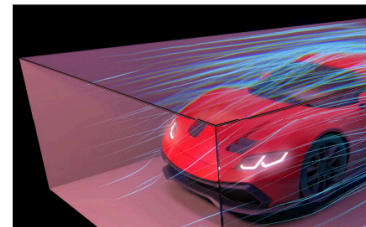
[Miller et al. 2023, Bakbouk and Peers 2023, Li et al. 2023]

[Qi et al. 2022]

[Sawhney and Crane 2020, Li et al. 2024]

### Boundary Value Caching for Walk on Spheres

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



### Neural Caches for Monte Carlo Partial Differential Equation Solver

Zilu Li\* Cornell University zli327@cornell.edu  
Guandao Yang\* Cornell University gy46@cornell.edu  
Xi Deng Cornell University xd93@cornell.edu  
Christopher De Sa Cornell University cdesa@cs.cornell.edu  
Bharath Hariharan bharathh@cs.cornell.edu  
Steve Marschner sm@cs.cornell.edu

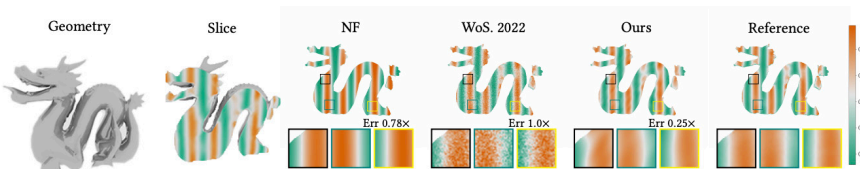


Figure 1: We visualize a slice of the solution to an elliptic PDE within a dragon-shaped boundary. Our hybrid solver can reduce the error of the neural field baseline, while achieving lower variance compared to the Walk-on-Spheres [Sawhney et al. 2022] method when working within the constraints of a limited computing budget

**ABSTRACT**  
This paper presents a method that uses neural networks as a caching mechanism to reduce the variance of Monte Carlo Partial Differential Equation solvers, such as the Walk-on-Spheres algorithm [Sawhney and Crane 2020]. While these Monte Carlo PDE solvers have the merits of being unbiased and discretization-free, their high variance often hinders real-time applications. On the other hand, neural networks can approximate the PDE solution, and evaluating these networks at inference time can be very fast. However, neural-network-based solutions may suffer from convergence difficulties and high bias. Our hybrid system aims to combine these two potentially complementary solutions by training a neural field to approximate the PDE solution using supervision from a WoS solver. This neural field is then used as a cache in the WoS solver to reduce variance during inference. We demonstrate that our neural field training procedure is better than the commonly used self-supervised objectives in the literature. We also show that our hybrid solver exhibits lower variance than WoS with the same computational budget: it is significantly better for small compute budgets and provides smaller improvements for larger budgets, reaching the same performance as WoS in the limit.

**KEYWORDS**  
PDE Solver, Monte Carlo, Neural Fields, Geometry Processing  
**ACM Reference Format:**  
Zilu Li, Guandao Yang, Xi Deng, Christopher De Sa, Bharath Hariharan, and Steve Marschner. 2023. Neural Caches for Monte Carlo Partial Differential Equation Solver. In SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3610548.3618141>

**1 INTRODUCTION**  
Solving elliptic PDEs is critical for various computer graphics applications, including 3D reconstruction, animation, and physics simulation. Conventional PDE solvers, however, typically involve time-consuming and error-prone discretization of space with finite elements or meshes. Monte Carlo PDE solvers based on the Walk on Spheres (WoS) algorithm [Sawhney and Crane 2020; Sawhney et al. 2022, 2022] offer a way to circumvent these issues by estimating solution values without discretization. These solvers, however, suffer from high variance, making them slow as they require numerous samples to reduce the variance. This prevents their use in many applications with limited computing budgets.

An alternative to both discretized and Monte Carlo solvers is to use neural fields to approximate the solution to a PDE. Neural fields are a class of neural networks that take spatial coordinates as input and output values of a continuous field [Paiasi et al. 2019; Xie et al. 2022]. Prior works have developed self-supervised losses that can be used to optimize a neural field so that it satisfies a given PDE

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

### A bidirectional formulation for Walk on Spheres

Yang Qi<sup>1</sup> Dario Seyb<sup>1</sup> Benedikt Bitterli<sup>1,2</sup> Wojciech Jarosz<sup>2</sup>  
<sup>1</sup>Dartmouth College <sup>2</sup>NVIDIA

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

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

**1. Introduction**  
Monte Carlo methods have been very successful in rendering. They provide accurate solution estimates to the rendering equation [Kaji86] in very complex scenes with an often simple implementation compared to mesh based methods such as radiosity [CW93; GT08B4]. While initially Monte Carlo methods were comparatively slow and of mostly academic interest, they now form the predominant rendering methodology in movie production [C16; FH17] and increasingly in interactive applications such as games.

Catalyzed by the recent introduction of the walk-on-spheres (WoS) algorithm [Mul56] to graphics [SC20; SSJ22], a similar development is now happening for numerical solvers of partial differential equations (PDEs). Analogous to path tracing in rendering, WoS uses random walks to compute point estimates of the solution of harmonic PDEs [Eva10]. These equations are of great importance in many areas of science since they can model natural phenomena such as heat dissipation, diffusion of electrostatic charges, and distribution of water in soil. Additionally, diffusion equations are often used in rendering and related fields to approximate the behaviour of light in highly scattering media [JML10]; Sta95].

Analogous to path tracing, existing WoS algorithms start "paths" (i.e. walks) at "sensor points" and end them at "lights" (i.e. boundary points). This works well when the probability of finding source points is high, but produces high variance for sparse sources.

Because of the apparent parallels between the Monte Carlo algorithms used for solving rendering problems and those solving PDEs,

we hope to leverage the decades of rendering research and apply them to PDEs to develop new robust and efficient Monte Carlo PDE solvers. A major step in rendering was the transition from unidirectional "forward" methods like path tracing to "backward" methods such as light tracing, photon mapping [Jen90] and virtual point lights (VPLs) [Ker97]. This precipitated the comprehensive framework of bidirectional rendering methods [Vee97] and, ultimately, the wealth of transport methods available today [PJH16]. In this paper, we mirror this development and propose extensions of WoS in both forward and backward directions.

The standard "forward" WoS algorithm leverages the mean value theorem to form recursive estimators for the solution. We instead formulate a mean value theorem for the Green's function. This allows us to derive a new class of WoS algorithm that start paths "backwards" from source and boundary points, distributing energy more evenly throughout the domain, before connecting either to sensor points directly, or to short sensor subpaths, mimicking a form of "final gather" [Re92] to reduce structured sampling artifacts.

We demonstrate the effectiveness and correctness of our method both by solving the diffusion equation in highly scattering media, and by rendering diffusion curve images [OBW08]. Much like in light transport, we hope that the development of these algorithms opens the path to comprehensive bidirectional methods in the future for fully robust Monte Carlo solution of PDEs.

For simplicity, we limit ourselves to a particular class of elliptic PDEs called the Poisson equation with Dirichlet boundary conditions

© 2022 The Author(s)  
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

### Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-Based Methods on Volumetric Domains

ROHAN SAWHNEY and KEENAN CRANE, Carnegie Mellon University



Fig. 1. Real-world geometry has not only rich surface algorithms struggle to mesh, setup, and solve PDEs—in Carlo solver uses about 1GB of memory and takes less [Boundary mesh of Fijian strumigenys F1]3 used courtesy

This paper explores how core problems in PDE-based MC can be efficiently and reliably solved via grid-free MC. Modern geometric algorithms often need to solve Poisson geometrically intricate domains. Conventional methods free Monte Carlo methods avoid mesh generation entirely evaluate closest point queries. They hence do not do not even function spaces, and provide the exact solid even on extremely challenging models. More broadly benefits with Monte Carlo methods from photo-realistic scaling, trivial parallel implementation, view-dependent ability to work with any kind of geometry (including descriptions). We develop a complete "black box" solve integration, variance reduction, and visualization, and use for various geometry processing tasks. In particular fundamental linear elliptic PDEs with constant coefficients. Overall we find that Monte Carlo methods signi horizons of geometry processing, since they easily handle and complexity that are essentially hopeless for conve

**CCS Concepts**  
• Computing methodologies → Sha  
Additional Key Words and Phrases: numerical method  
Authors' address: Rohan Sawhney, Keenan Crane, Carnegie Mellon University, Pittsburgh, PA, 15213

Permission to make digital or hard copies of part or all of this classroom use is granted without fee provided that copies are for profit or commercial advantage and that copies bear this notice on the first page. Copyrights for third-party components of this work for all other uses, contact the owner(s)/author(s).  
© 2020 Copyright held by the owner(s).  
<https://doi.org/10.1145/368669.3392734>

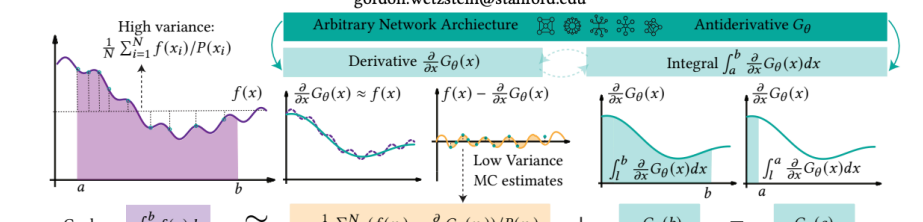


Figure 1: We propose a novel method to use arbitrary neural network architectures as control variates (CV). Instead of using the network to approximate the integrand, we deploy it to approximate the antiderivative of the integrand. This allows us to construct pairs of networks where one is the analytical integral of the other, tackling a main challenge of neural CV methods.

**ABSTRACT**  
This paper presents a method to leverage arbitrary neural network architecture for control variates. Control variates are crucial in reducing the variance of Monte Carlo integration, but they hinge on finding a function that both correlates with the integrand and has a known analytical integral. Traditional approaches rely on heuristics to choose this function, which might not be expressive enough to correlate well with the integrand. Recent research alleviates this issue by modeling the integrands with a learnable parametric model, such as a neural network. However, the challenge remains in creating an expressive parametric model with a known analytical integral. This paper proposes a novel approach to construct learnable parametric control variates functions from

**CCS CONCEPTS**  
• Computing methodologies → Computer graphics; Modeling and simulation; Neural networks.

Eurographics Symposium on Rendering 2022  
T. Ritschel and A. Witschold (Editors)

### Mean Value Caching for Walk on Spher

Ghada Bakhouk\* Pieter Peers\*  
College of William & Mary

**Abstract**  
Walk on Spheres (WoS) is a grid-free Monte Carlo method for numerically estimating solutions for elliptic equations (PDE) such as the Laplace and Poisson PDEs. While WoS is efficient for computing an evaluation point, it becomes less efficient when the solution is required over a whole domain or a computes a solution for each evaluation point separately, possibly recomputing similar sub-walks multiple evaluation points. In this paper, we introduce a novel filtering and caching strategy that leverages properties (in contrast to the boundary mean value property that forms the core of WoS). In addition, a sparse cache regimes, we describe a weighted mean as well as a non-uniform sampling method. Final reduce the variance within the cache by recursively applying the volume mean value property on the CC

**1. Introduction**  
Partial differential equations (PDEs) form the basis of many fundamental computer graphics problems. The recently introduced Monte Carlo Geometry Processing (MC-GP) framework [SC20] offers an exciting new strategy for solving PDEs defined over volumes without the need to discretize or create a grid. At its core, MC-GP builds on the Walk on Spheres (WoS) algorithm for solving PDEs. Due to its conceptual similarity to path tracing, many Monte Carlo innovation and solution strategies from rendering have been applied to MC-GP, such as importance sampling [SC20] and reverse and bidirectional algorithms [QBB22].

Monte Carlo techniques are very effective for computing a solution estimate at a single evaluation point. However, in many practical cases, the solution over the whole volume or region of interest in the volume is desired. Because the solution estimate is computed for each evaluation point separately and independently, many similar sub-walks are recomputed multiple times. To reduce recomputation, prior work looked at interpolation with Moving Least Squares [Neu04] of forward walks, and reverse reverse walks [QBB22]. While the former is biased, the latter still requires a large number of reverse walks to ensure a sufficiently dense overlap with each evaluation point.

In this paper we present a novel method for reusing forward walks that is easy to implement in existing WoS frameworks. At the core of our method is the volume mean value property, hence we call our method (volume) mean value caching. The volume mean value property is the volumetric counterpart of the (boundary) mean value property that enables walking on spheres. We show that sim-

ply performing the first step in a WoS instead of its boundary leads to reduces Monte Carlo noise over m covering a volume. We also show the very leads to an efficient caching scheme distributed cache samples. It between evaluation points, we describe sampling strategy. Furthermore, we mean value property to improve it when using a low number of cache by recursively applying the volume also reduce the variance in the cache

We validate our mean value caching and show that we can reduce several orders of magnitude for our contributions are:

- A hybrid volume and boundary
- An unbiased post-processing Monte Carlo network from permission@cmg.org
- A mean value caching and gate walks that is unbiased for uniform and consistent for a non-uniform
- A weighted volume mean value
- A recursive algorithm for reducing

**2. Related Work**  
Walk on spheres [Mul56] was introduced to computer graphics in the context of grid-free Monte Carlo geometry process-

© 2023 The Authors.  
Proceedings published by Eurographics - The Eurographics Association for Computer Graphics.  
This work is licensed under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

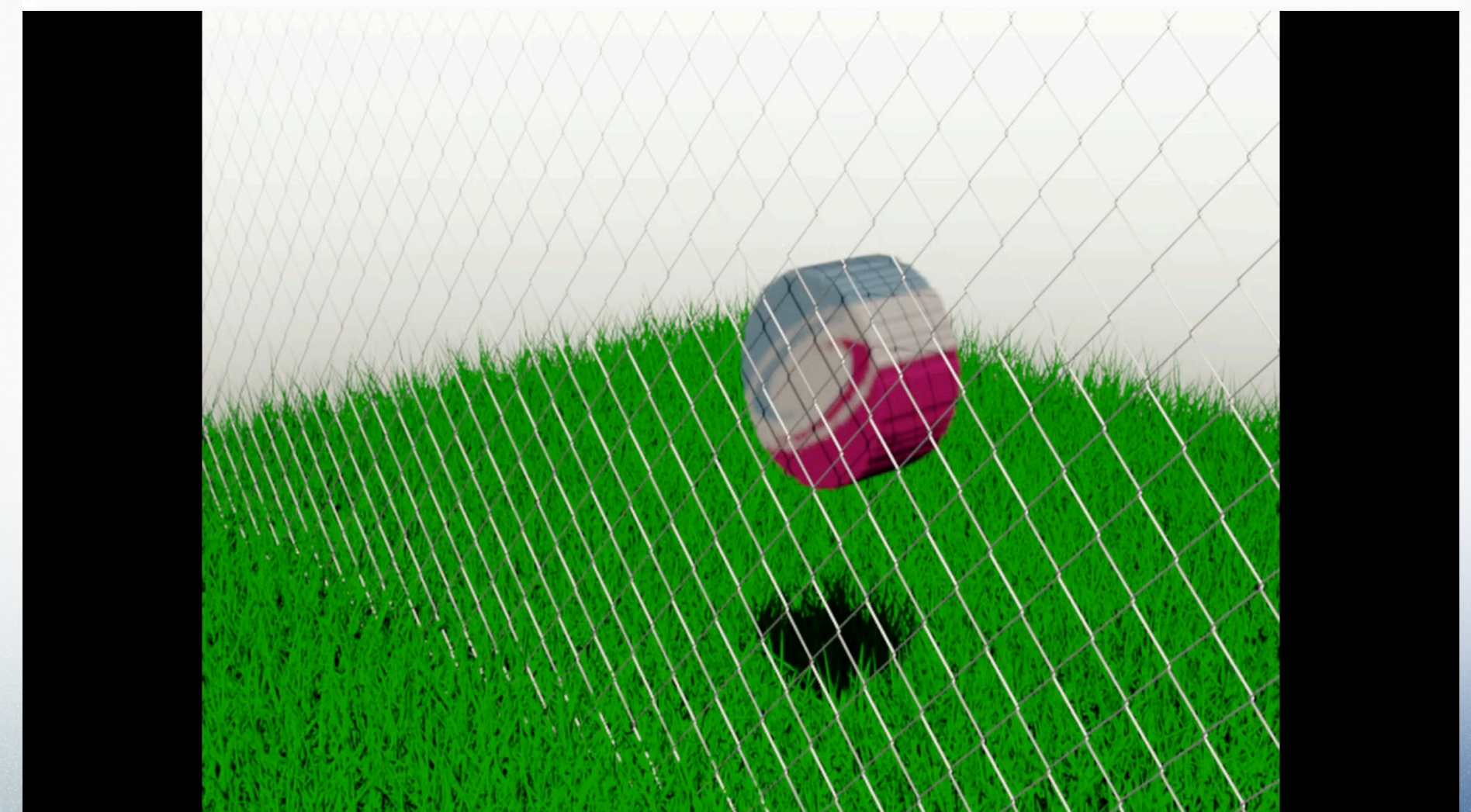
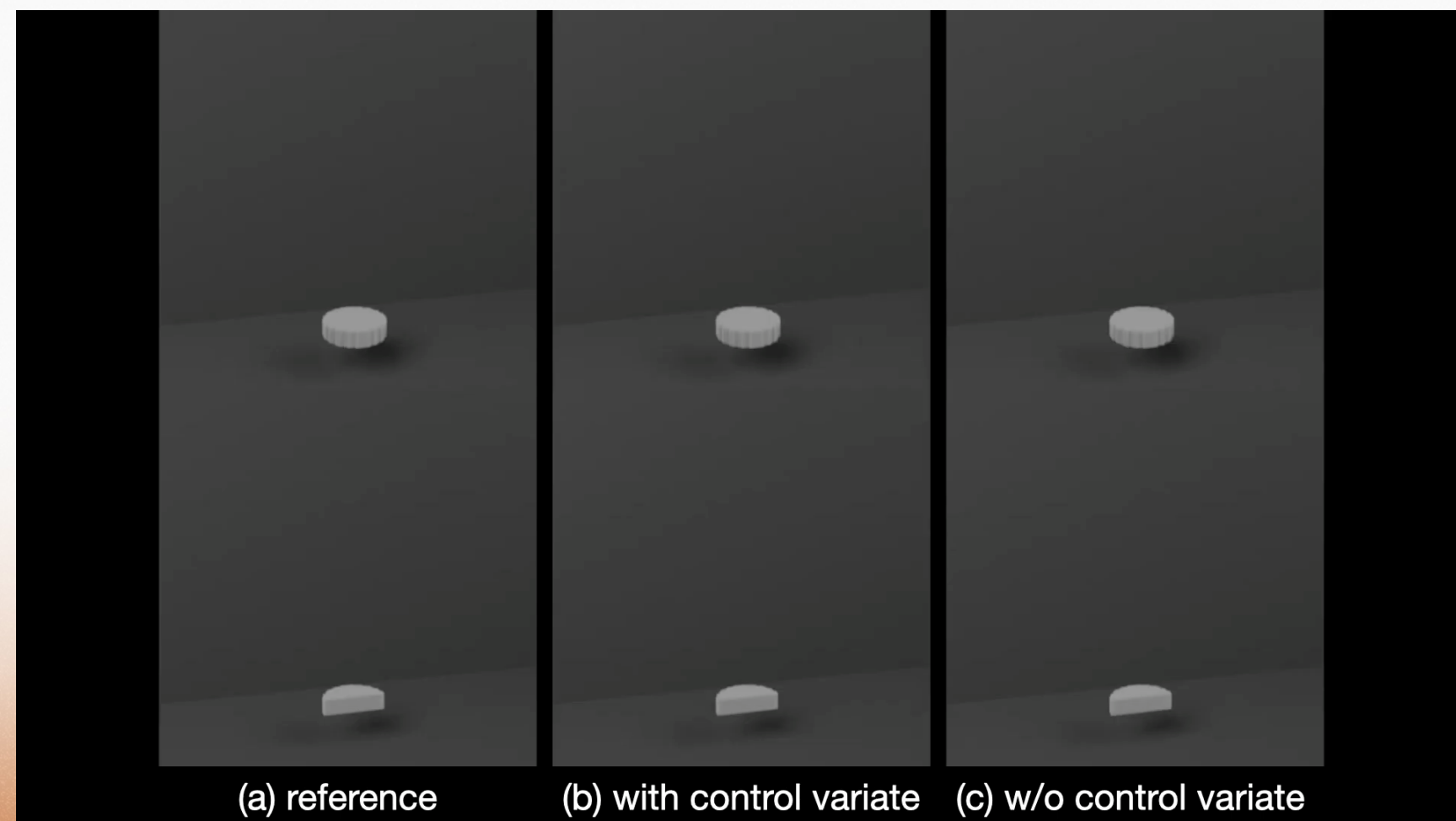
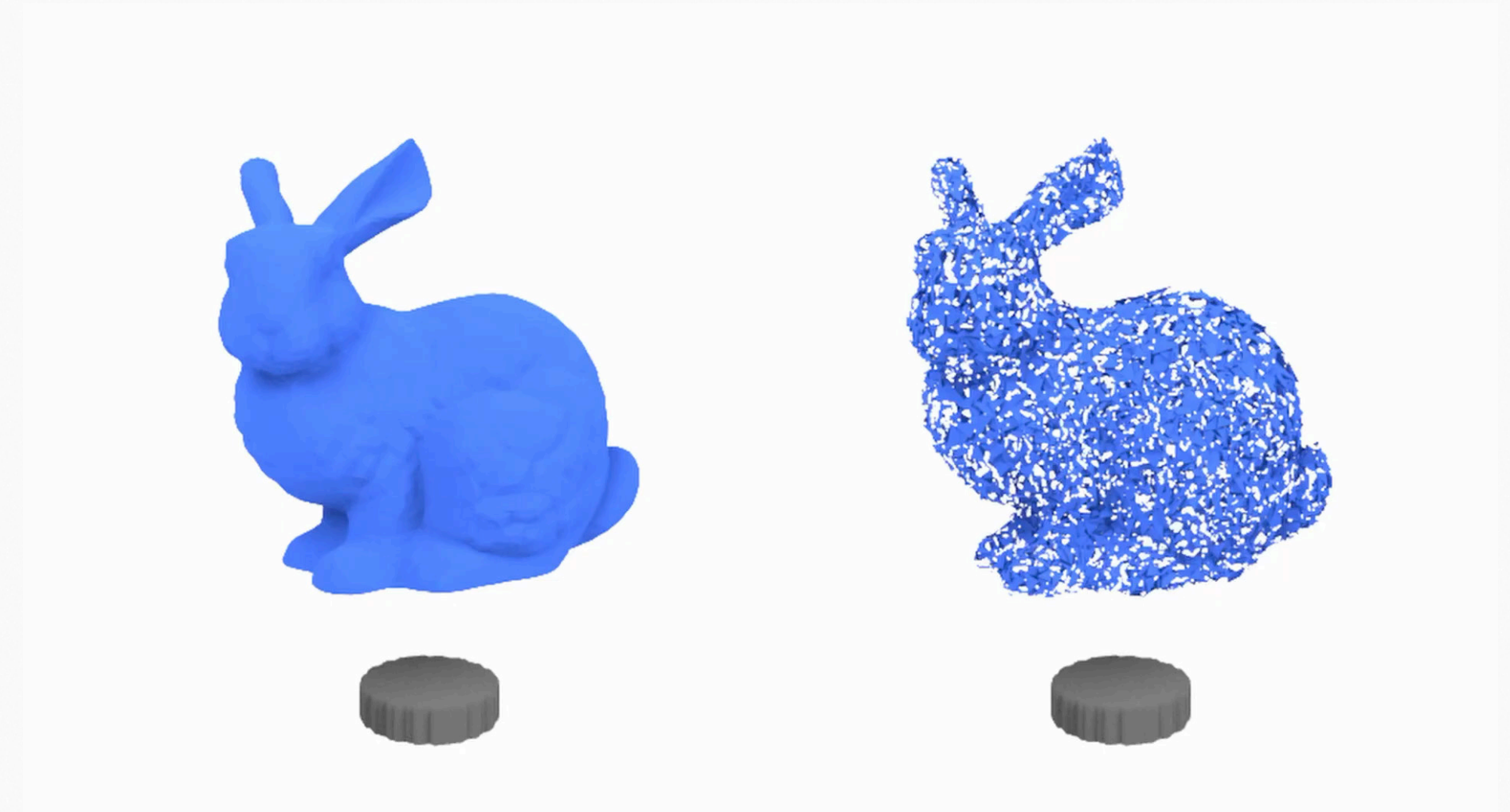


DOI: 10.2312/eur.2023.1120



# broader range of physics

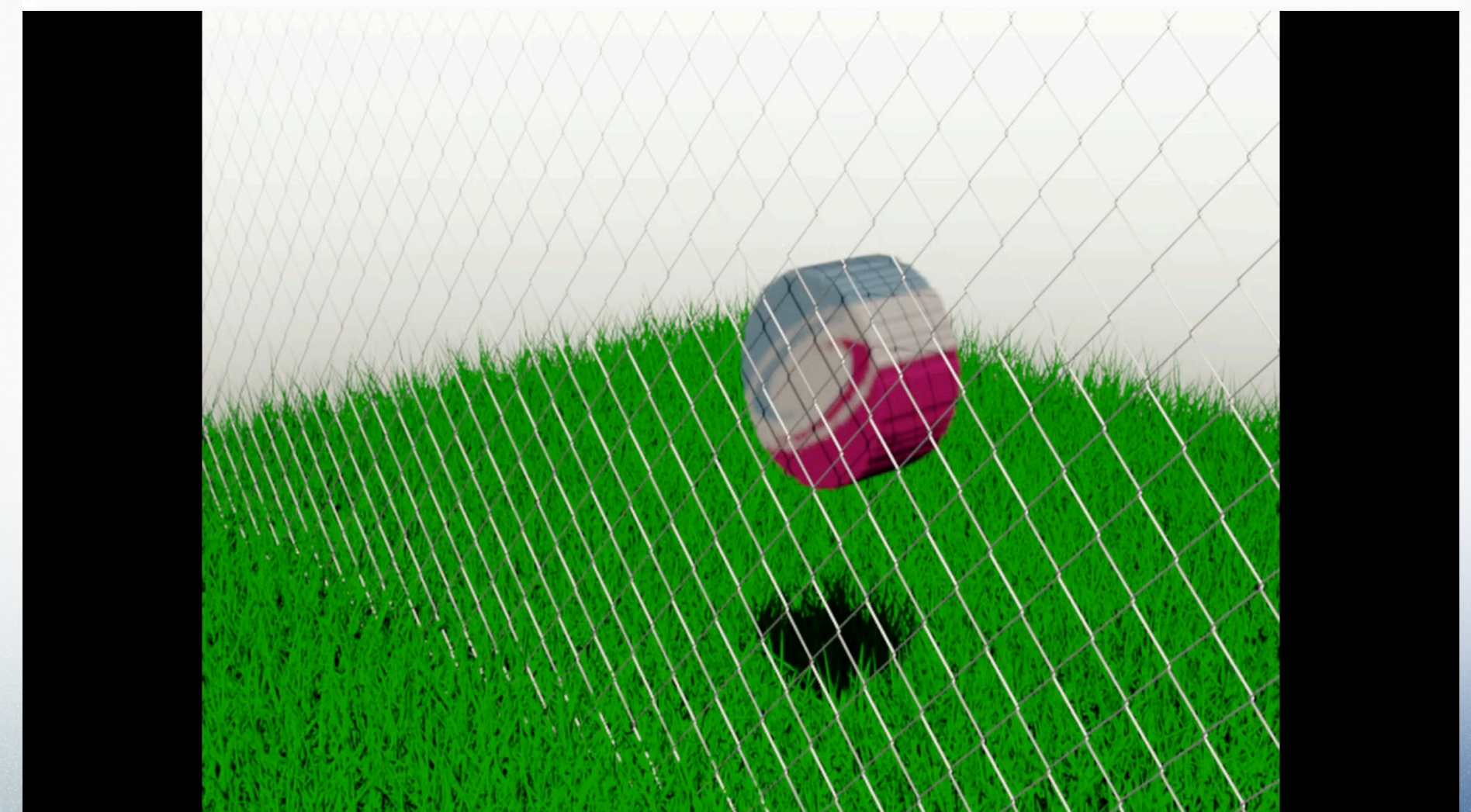
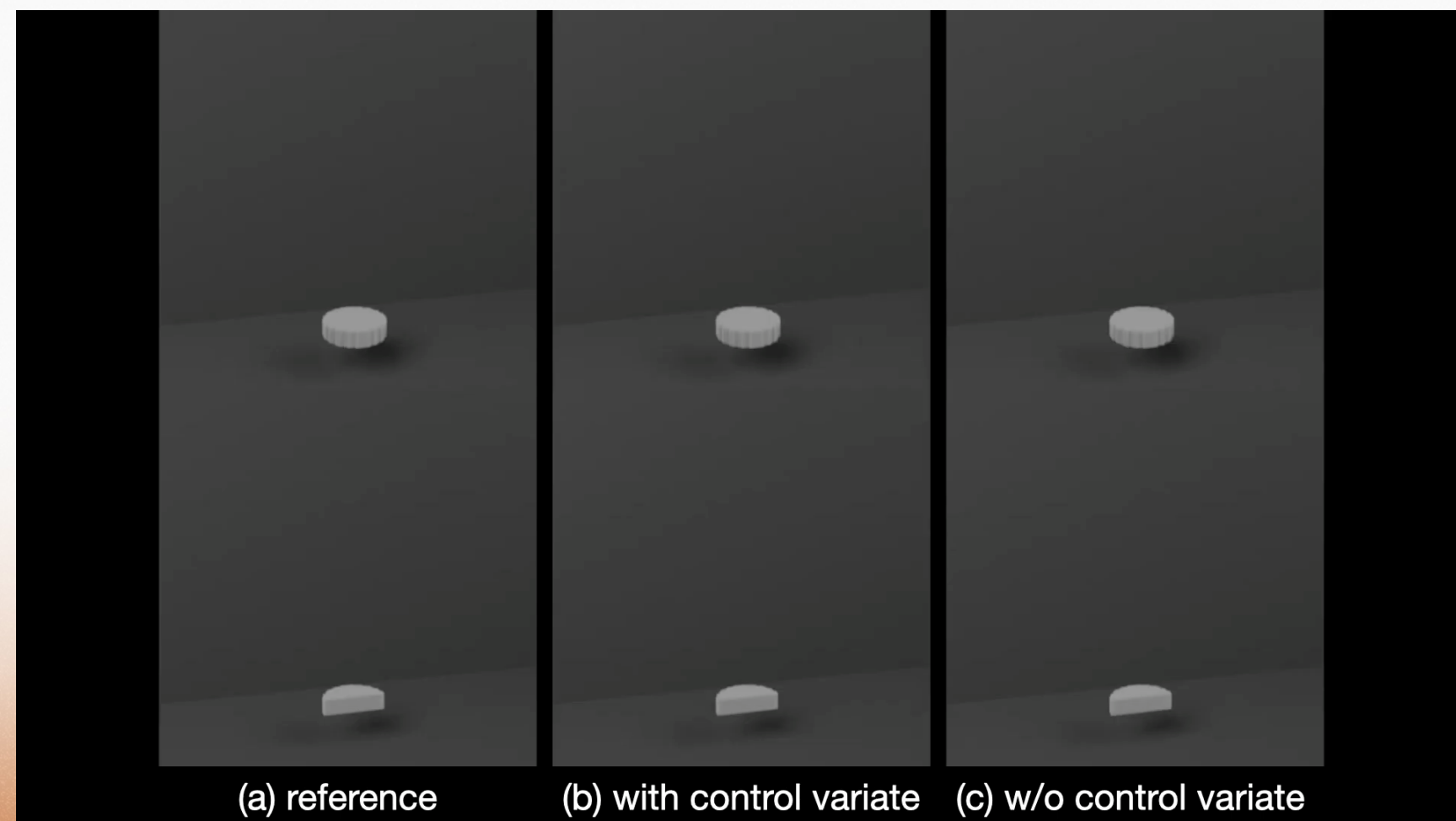
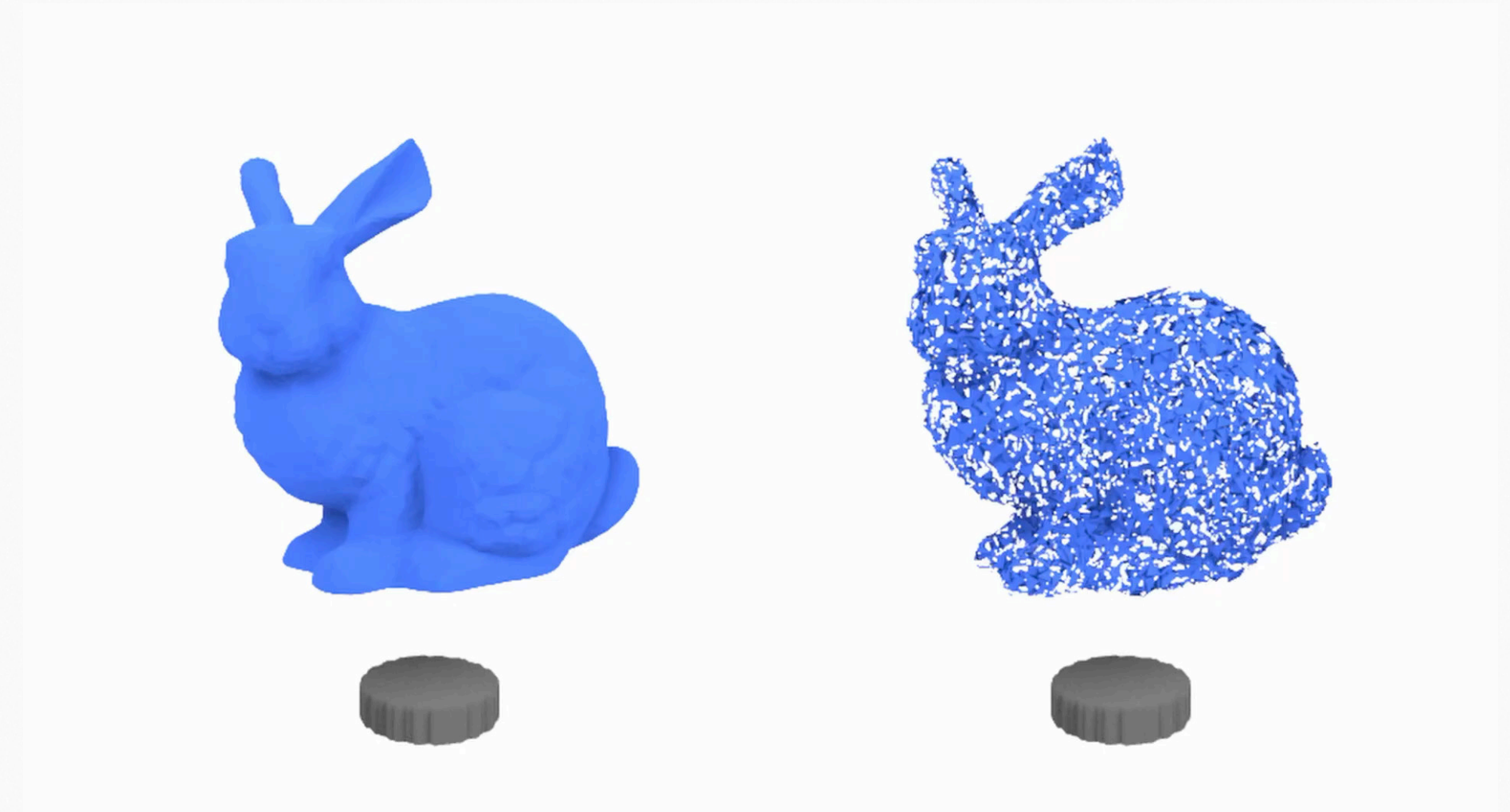
Rioux-Lavoie et al. 2022, "A Monte Carlo Method for Fluid Simulation"





# broader range of physics

Rioux-Lavoie et al. 2022, "A Monte Carlo Method for Fluid Simulation"





# broader range of physics

## MC solver as subroutine

[Rioux-Lavoix et al. 2022, Sugimoto et al. 2024, Jain et al. 2024]

## coupling MC physics solvers

[Bati et al. 2023]

## new MC physics solvers

### A Monte Carlo Method for Fluid Simulation

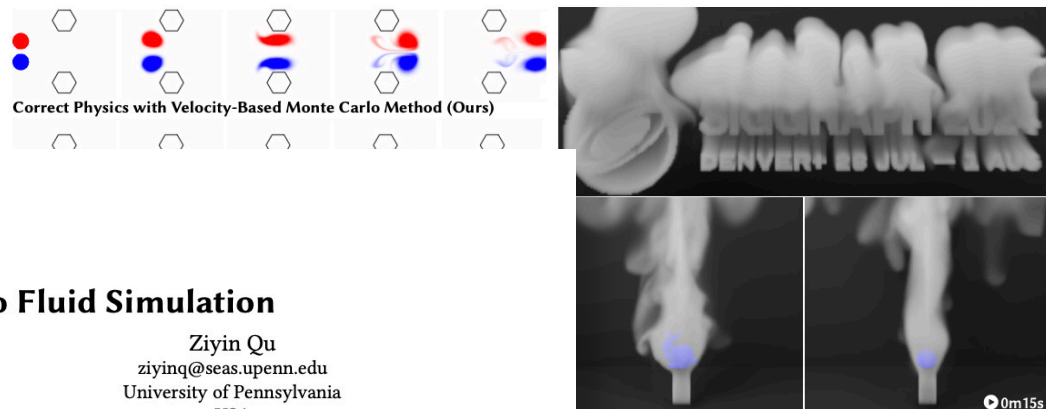
DAMIEN RIOUX-LAVOIX\*, McGill University and Ubisoft Montreal, Canada  
 RYUSUKE SUGIMOTO\*, University of Waterloo, Canada  
 TÜMAY ÖZDEMİR, University of Waterloo, Canada  
 NAOHARU H. SHIMADA, Osaka University, Japan  
 CHRISTOPHER BATTY, University of Waterloo, Canada  
 DEREK NOWROUZEZAHRAI, McGill University, Canada  
 TOSHIYA HACHISUKA, University of Waterloo, Canada

### Velocity-Based Monte Carlo Fluids

Ryusuke Sugimoto  
 University of Waterloo  
 Waterloo, Ontario, Canada  
 rsugimot@uwaterloo.ca

Christopher Batty  
 University of Waterloo  
 Waterloo, Ontario, Canada  
 christopher.batty@uwaterloo.ca

Toshiya Hachisuka  
 University of Waterloo  
 Waterloo, Ontario, Canada  
 toshiya.hachisuka@uwaterloo.ca



### Neural Monte Carlo Fluid Simulation

Pranav Jain  
 pranavj@usc.edu  
 University of Southern California  
 USA

Ziyin Qu  
 ziyinqu@seas.upenn.edu  
 University of Pennsylvania  
 USA

Peter Yichen Chen  
 pyc@csail.mit.edu  
 MIT CSAIL  
 USA

Oded Stein  
 ostein@usc.edu  
 University of Southern California  
 USA

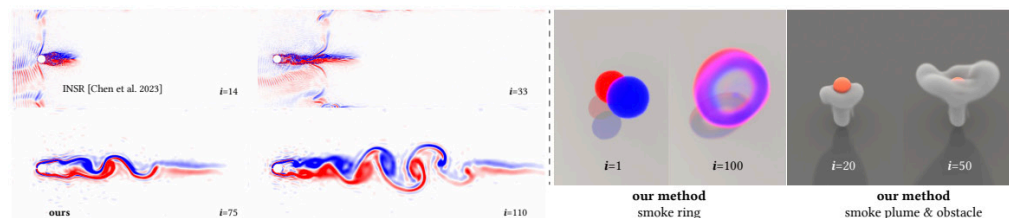


Figure 1: Our method simulates fluids in the presence of obstacles with a combined neural network and Monte Carlo approach to operator splitting for the Navier Stokes equations. With our method, we can simulate important qualitative vorticity-based phenomena, such as vortex shedding in the von Kármán vortex street experiment, various neural spatial representation papers [Chen et al. 2023b] cannot (left).

**ABSTRACT**  
 The idea of using a neural network to represent continuous vector fields (i.e., neural fields) has become popular for solving PDEs arising from physics simulations. Here, the classical spatial discretization (e.g., finite difference) of PDE solvers is replaced with a neural network that models a differentiable function, so the spatial gradients of the PDEs can be readily computed via autodifferentiation. When used in fluid simulation, however, neural fields fail to capture many important phenomena, such as the vortex shedding experienced in the von Kármán vortex street experiment. We present a novel neural network representation for fluid simulation that augments neural fields with explicitly enforced boundary conditions as well as a Monte Carlo pressure solver to get rid of all weakly-enforced boundary conditions. Our method, the Neural

Fig. 1. Our novel Monte Carlo fluid of the vorticity transport equation. The adoption of a Monte Carlo method

We present a novel Monte Carlo-based pointwise and stochastic estimator Feynman-Kac representation of the recursive Monte Carlo estimator - a stream function formulation that conditions using a Walk-on-Sphere literature in rendering, we design a suited to a fluid simulation context: complex boundary settings, and det with temporal grid caching. We via quantitative and qualitative evaluation domain geometries - and thorough. Finally, we provide an in-depth of building on this new numerical skin

\*Authors with equal contribution.

Authors' addresses: Damien Rioux-Lavoix, damien.rioux-lavoix@gmail.com; Ryusuke Sugimoto, rsugimot@uwaterloo.ca; Tümay Özdemir, tuzdemir@uwaterloo.ca; Naoharu H. Shimada, hshimada@uwaterloo.ca; Christopher Batty, cbatty@uwaterloo.ca; Derek Nowrouzezahrai, dnowrouz@uwaterloo.ca; Toshiya Hachisuka, thachisuka@uwaterloo.ca

Permission to make digital or hard copy classroom use is granted without fee for profit or commercial advantage and to be published on servers or to redistribute and/or a fee. Request permissions from the ACM Copyright held by the owner/author(s). 0730-0301/2022/12-ART240 \$15.00 https://doi.org/10.1145/3592121

handle scenes for which the existing vorticity-based Monte Carlo methods produce incorrect results. Our solver allows the red and blue smoke plumes (left, top), similar to the conventional non-Monte Carlo vorticity-based method produces an incorrect result, in (right, top). Our solver also readily supports commonly available buoyancy effects: we simulate a smoke plume rising over a bunny-shaped obstacle (right, bottom), where the

### CCS CONCEPTS

Computing methodologies → Physical simulation; Ray tracing; Mathematics of computing → Partial differential equations; Integral equations.

### KEYWORDS

Monte Carlo methods, fluid simulation, walk-on-boundary

### Coupling Conduction, Convection and Radiative Transfer in a Single Path-Space: Application to Infrared Rendering

MÉGANE BATI, IRTIT, Université de Toulouse, UPS, CNRS, France  
 STÉPHANE BLANCO, LAPLACE, Université de Toulouse, CNRS, INPT, UPS, France  
 CHRISTOPHE COUSTET, Meso-Star, France  
 VINCENT EYMET, Meso-Star, France  
 VINCENT FOREST, Meso-Star, France  
 RICHARD FOURNIER, LAPLACE, Université de Toulouse, CNRS, INPT, UPS, France  
 JACQUES GAUTRAIS, CRCA, CBI, Université de Toulouse, CNRS, UPS, France  
 NICOLAS MELLADO, CNRS, IRTIT, Université de Toulouse, UPS, France  
 MATHIAS PAULIN, IRTIT, Université de Toulouse, UPS, CNRS, France  
 BENJAMIN PIAUD, Meso-Star, France

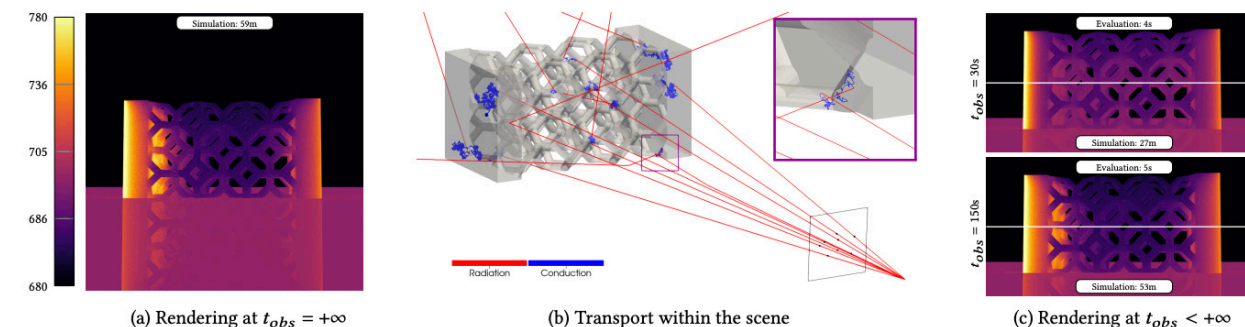


Fig. 1. We propose a Monte Carlo approach to tackle multiple physics with a single algorithm, translating their coupling into a single path-space composed of randomly chained sub-paths for each physics. Application is exemplified with heat transfer. (a) An infrared image of a steady state thermal exchanger, with temperature imposed on the left and right walls. (b) Monte Carlo paths alternate between heat-transfer modes (here conduction and radiation). (c) A huge benefit is the fast production of transient simulations, at any time, using the information gathered in (a), i.e. from only one Monte Carlo run at steady state.

In the past decades, Monte Carlo methods have shown their ability to solve PDEs, independently of the dimensionality of the integration domain and for different use-cases (e.g. light transport, geometry processing, physics simulation). Specifically, the path-space formulation of transport equations is a key ingredient to define tractable and scalable solvers, and we observe nowadays a strong interest in the definition of simulation systems based on Monte Carlo algorithms. We also observe that, when simulating combined physics (e.g. thermal rendering from a heat transfer simulation), there is a

lack of coupled Monte Carlo algorithms allowing to solve all the physics at once, in the same path space, rather than combining several independent MC estimators, a combination that would make the global solver critically sensitive to the complexity of each simulation space. This brings to our proposal: a coupled, single path-space, Monte Carlo algorithm for efficient multi-physics problems solving.

In this work, we combine our understanding and knowledge of Physics and Computer Graphics to demonstrate how to formulate and arrange different simulation spaces into a single path space. We define a tractable formalism for coupled heat transfer simulation using Monte Carlo, and we leverage the path-space construction to interactively compute multiple simulations with different conditions in the same scene, in terms of boundary conditions and observation time. We validate our proposal in the context of infrared rendering with different thermal simulation scenarios: e.g., room temperature simulation, visualization of heat paths within materials (detection of thermal bridges), heat diffusion capacity of thermal exchanger. We expect that our theoretical framework will foster collaboration and multidisciplinary studies. The perspectives this framework opens are detailed and we suggest a research agenda towards the resolution of coupled PDEs at the interface of Physics and Computer Graphics.

Authors' addresses: Mégane Bati, IRTIT, Université de Toulouse, UPS, CNRS, Toulouse, France, megane.bati@irit.fr; Stéphane Blanco, LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France, stephane.blanco@laplace.univ-tlse.fr; Christophe Coustet, Meso-Star, Toulouse, France, christophe.coustet@meso-star.com; Vincent Eymet, Meso-Star, Toulouse, France, vincent.eymet@meso-star.com; Vincent Forest, Meso-Star, Toulouse, France, vincent.forest@meso-star.com; Richard Fournier, LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France, richard.fournier@laplace.univ-tlse.fr; Jacques Gautrais, CRCA, CBI, Université de Toulouse, CNRS, UPS, Toulouse, France, jacques.gautrais@univ-tlse3.fr; Nicolas Mellado, CNRS, IRTIT, Université de Toulouse, UPS, Toulouse, France, nicolas.mellado@irit.fr; Mathias Paulin, IRTIT, Université de Toulouse, UPS, CNRS, Toulouse, France, mathias.paulin@irit.fr; Benjamin Piaud, Meso-Star, Toulouse, France, benjamin.piaud@meso-star.com.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in , https://doi.org/10.1145/3592121.

Vol. 42, No. 4, Article . Publication date: August 2023.

lots to explore here!

wave equations,  
 linear elasticity,  
 Etc.



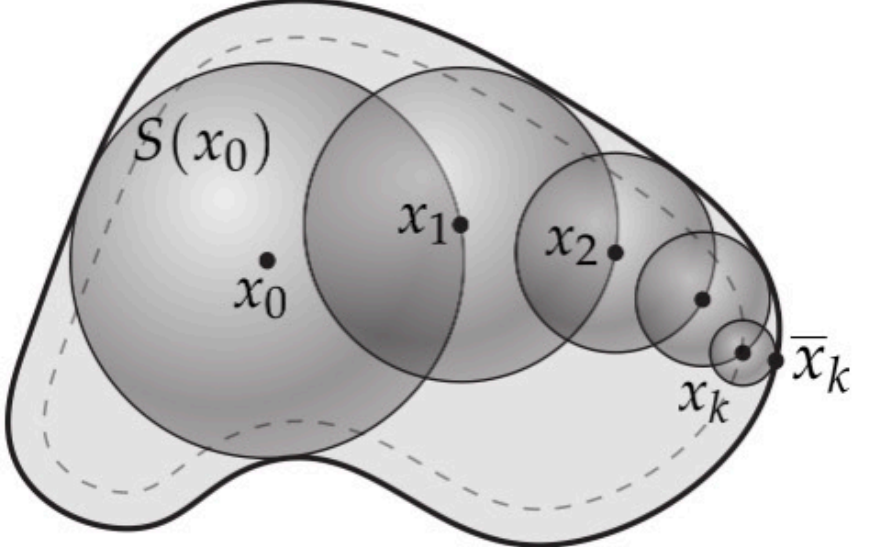
# open-source walk on stars solver

📖 README    📄 MIT license

# ZOMBIE

Zombie is a C++ header-only library for solving fundamental partial differential equations (PDEs) like the Poisson equation using the [walk on spheres \(WoS\)](#) method and its [extensions](#). Unlike finite element, boundary element, or finite difference methods, WoS does not require a volumetric grid or mesh, nor a high-quality boundary mesh. Instead, it uses random walks and the Monte Carlo method to solve the problem directly on the original boundary representation. It can also provide accurate solution values at a single query point, rather than needing to solve the problem over the entire domain. This [talk](#) provides an overview of WoS, while the following papers discuss its present capabilities in greater detail:

**WALK ON SPHERES**

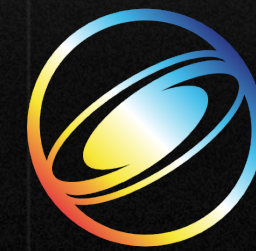


[github.com/rohan-sawhney/zombie](https://github.com/rohan-sawhney/zombie)





SIGGRAPH  
2024



**SIGGRAPH 2024**  
DENVER+ 28 JUL — 1 AUG

**THANK YOU FOR YOUR ATTENTION!**

**COME CHAT DURING THE POSTER SESSION**

