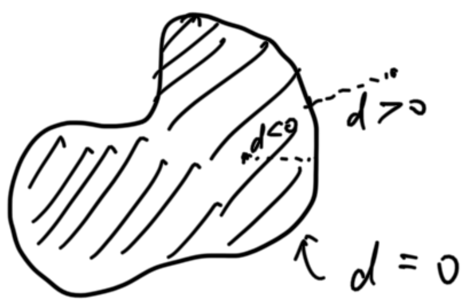
Lec 1 Discrete Representations of Space and Time

1. Shape representation

1.1 Signed Distance Field (SDF)

Given a function $f(x) = d$ ← $d \in \mathbb{R}$

↑ coordinates. $\mathbb{R}^2$ or $\mathbb{R}^3$
2D 3D

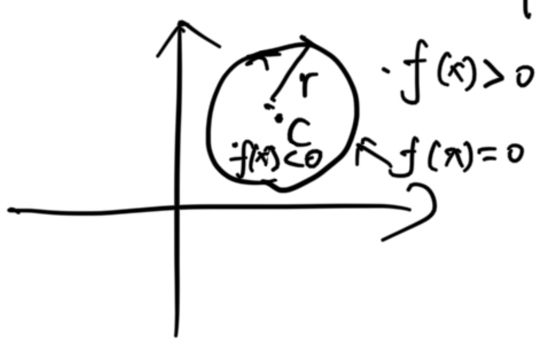


1.1.1 Analytical

For a disk in 2D,

$$f(x) = \|x - c\| - r$$

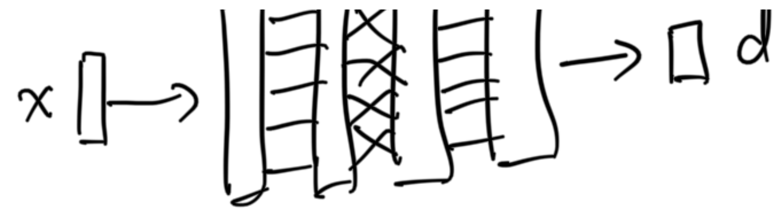
center radius



extends to half-spaces, boxes, ellipsoid ...
but not complex or arbitrary ones

neural representation (DeepSDF [Park et al. 2019]

$$f(x) = d$$



1.1.2 Discrete

e.g. a uniform grid storing distances at nodes

1.1.3 Remark

Analytial ones for primitives:
efficient and exact, but no complex shapes

useful to model obstacles that do not change shape and not very complex

Neural : can represent more complex shapes

- better compatibility with AI tasks
- haven't seen it applied to solids sims

Grid-based: easier to evolve, expensive for complex shapes

- FLIP and Eulerian fluids sims (track fluid volume)
- simulating contact with complex shaped objects

All SDF are not often used for deformable solids
(need both efficient representation of complex shapes, and easy shape change)

and easy shape changes)

1.2 Points (Partiples)

(point clouds)

often seen in surface reconstruction

For deformables

interior points also needed for calulating deformation and forces

based on connectivities of points to their neighbors

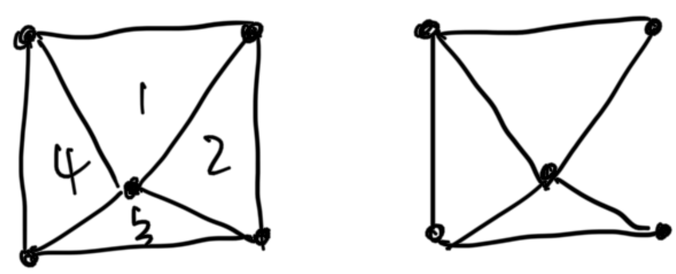
given by radial basis functions (e.g. SPH)

pros: automatic material separation

cons: accuracy requires lots of neighbors in uniform distribution

1.3 Mesh

points connected by elements (e.g. Finite Element Method, FEM)



different shapes

pros: accurate once mesh is constructed with good quality cons: need remeshing for material separation (fracture sims)

1.4 Hybrid

particles on a background uniform grid (e.g. MPM)

material point method

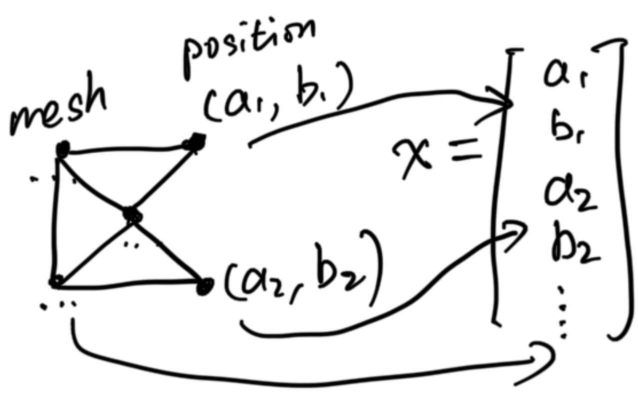
pros: easy separation of materials, more accuracy due to regular grid

cons: still require many neighbors in good distribution

2. Temporal Discretization

2.1 Newton's 2nd Law

$$\underbrace{f}_{\text{force}} = \underbrace{m}_{\text{mass}}\underbrace{a}_{\text{acceleration}}$$



$$\begin{cases} \dfrac{dx}{dt} = v \rightarrow \text{velocity} \\ M\dfrac{dv}{dt} = f \end{cases}$$

x → position

mesh n nodes (vertices)

$x \in \underset{2D}{R^{2n}}$ or $\underset{3D}{R^{3n}}$

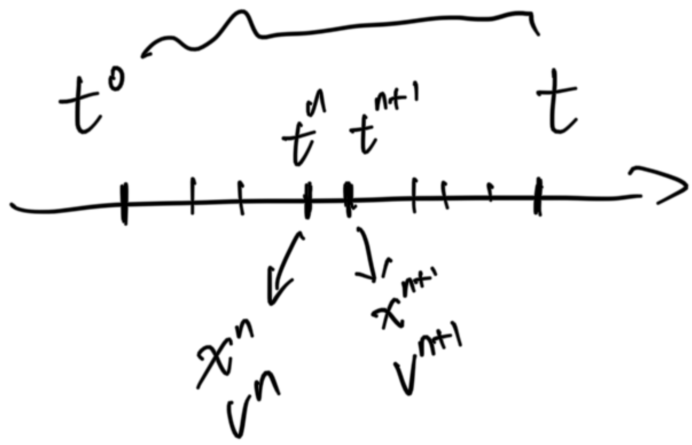
M ↓ matrix $R^{2n\times 2n}$ or $R^{3n\times 3n}$

$\begin{bmatrix} m_1 & & \\ & m_1 & \\ & & \ddots \end{bmatrix}$ m_i is the mass of node i

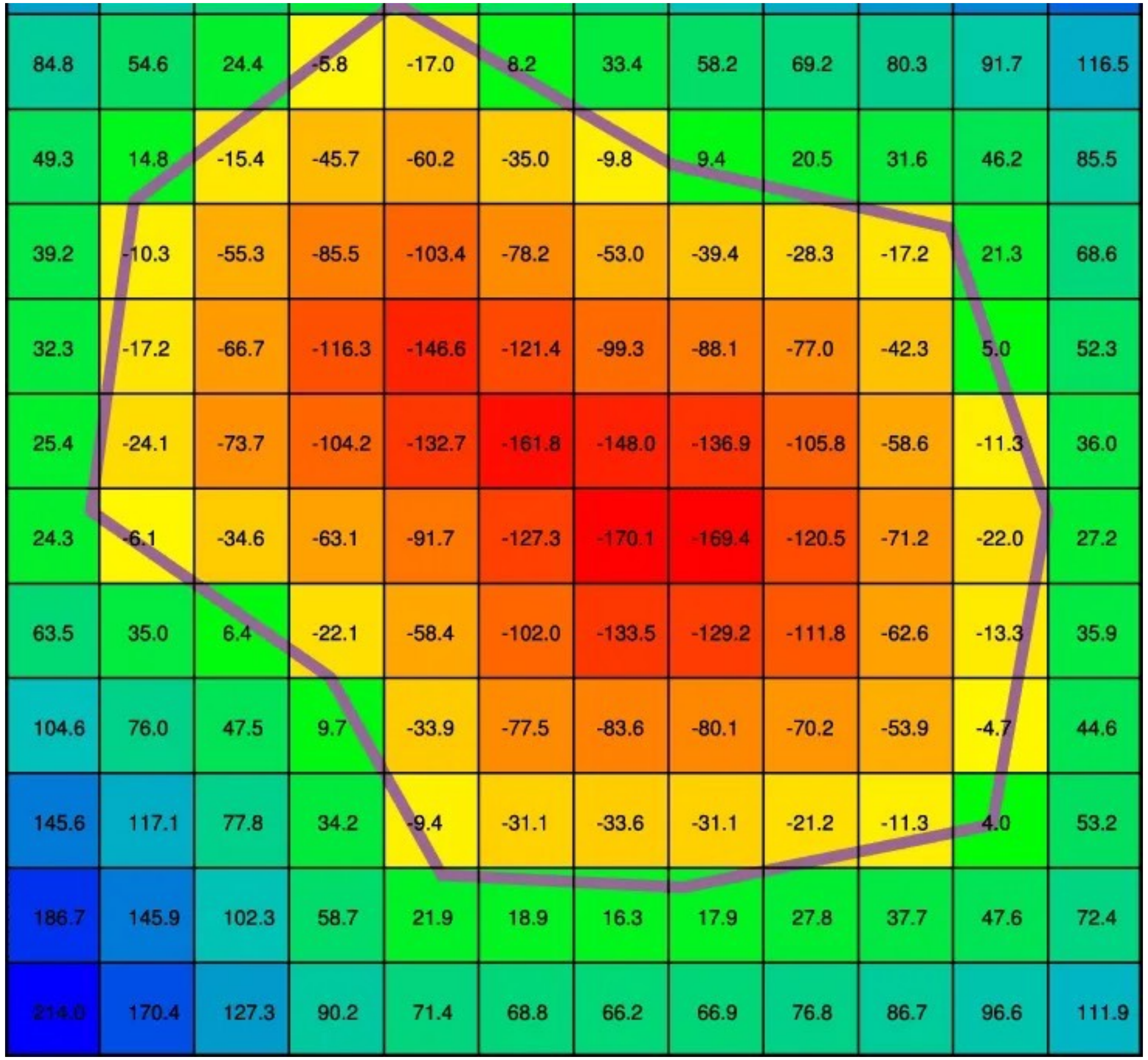
$$M = \begin{bmatrix} \ddots & & & & & \\ & m_2 & & & & \\ & & m_2 & & & \\ & & & \ddots & & \\ & & & & m_n & \\ & & & & & m_n \end{bmatrix}$$

m_i is the mass of node i

2.2 Time Stepping



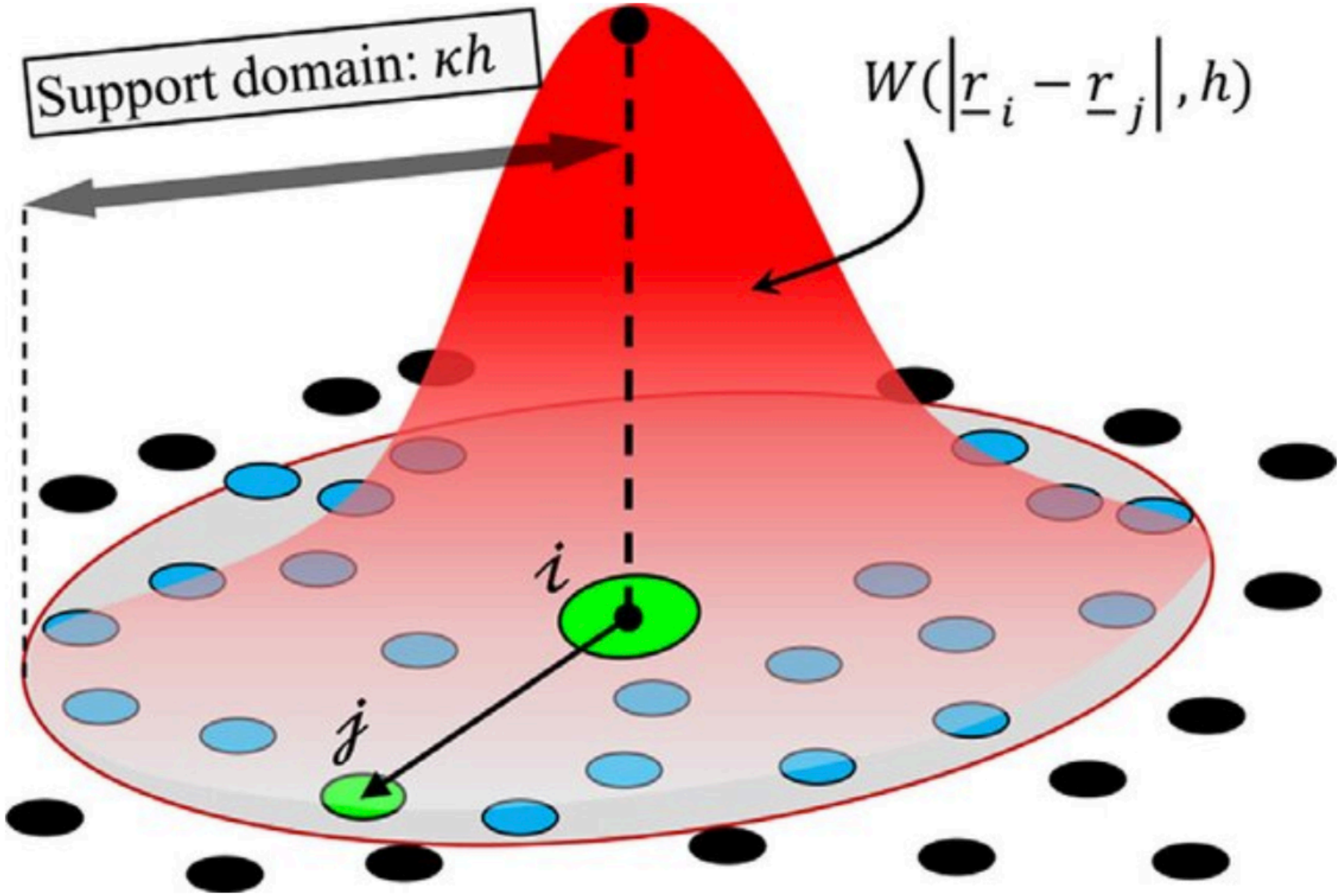
1.1.2 Discrete Version of SDF



https://medium.com/@pushkarevmm/signed-distance-field-simple-example-with-raymarched-soft-shadows-in-unity-2b3fdf20218

1.2 Particle Representations

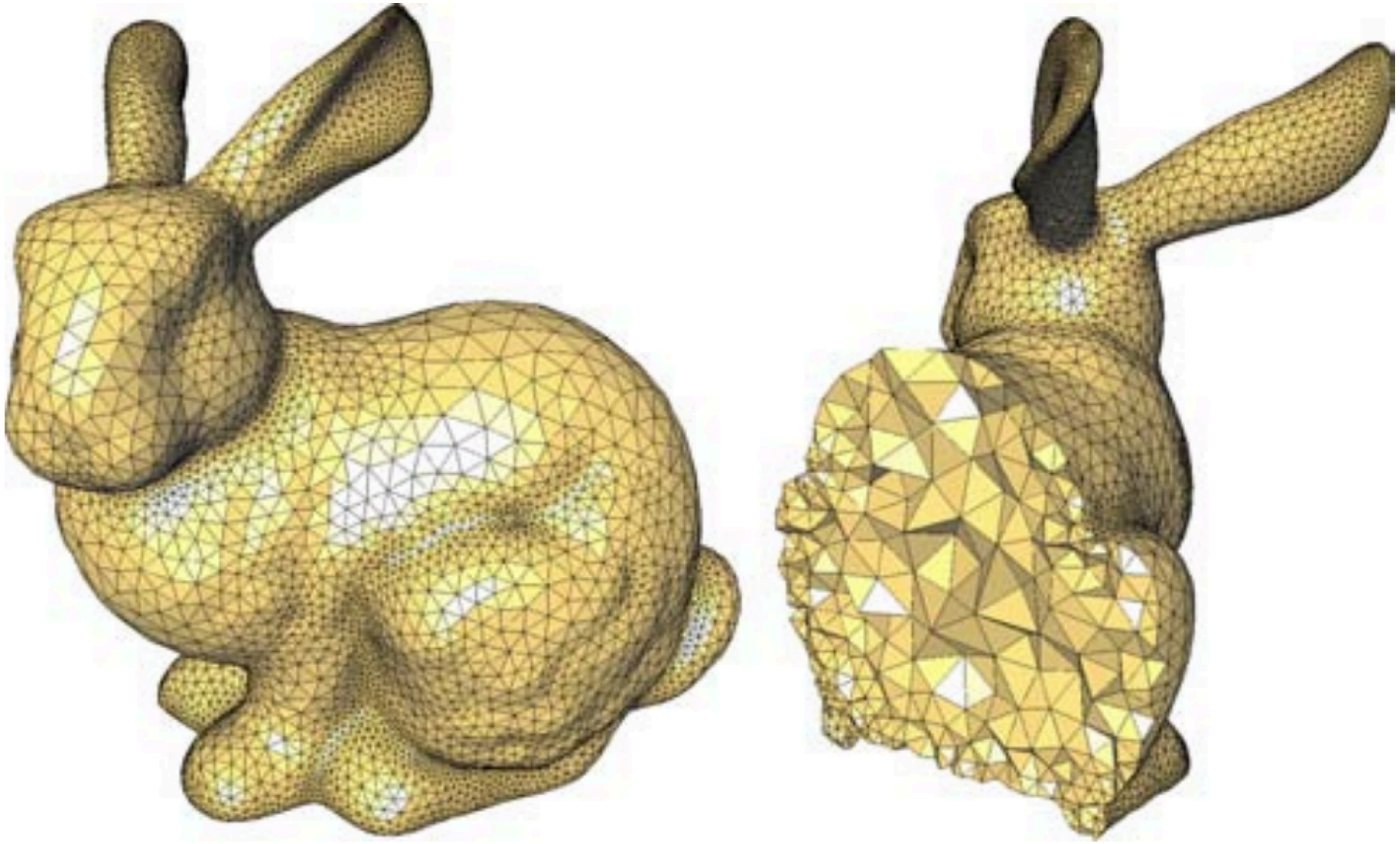
Radial Basis Functions



https://www.researchgate.net/publication/354002743_An_improved_thermal_model_for_SPH_metal_cutting_simulations_on_GPU

1.3 Mesh

Tetrahedral Mesh



http://geometry.caltech.edu/pubs/ACYD05.pdf

2.2 Time Stepping

Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, Danny M. Kaufman. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. ACM ToG (SIGGRAPH), 2019