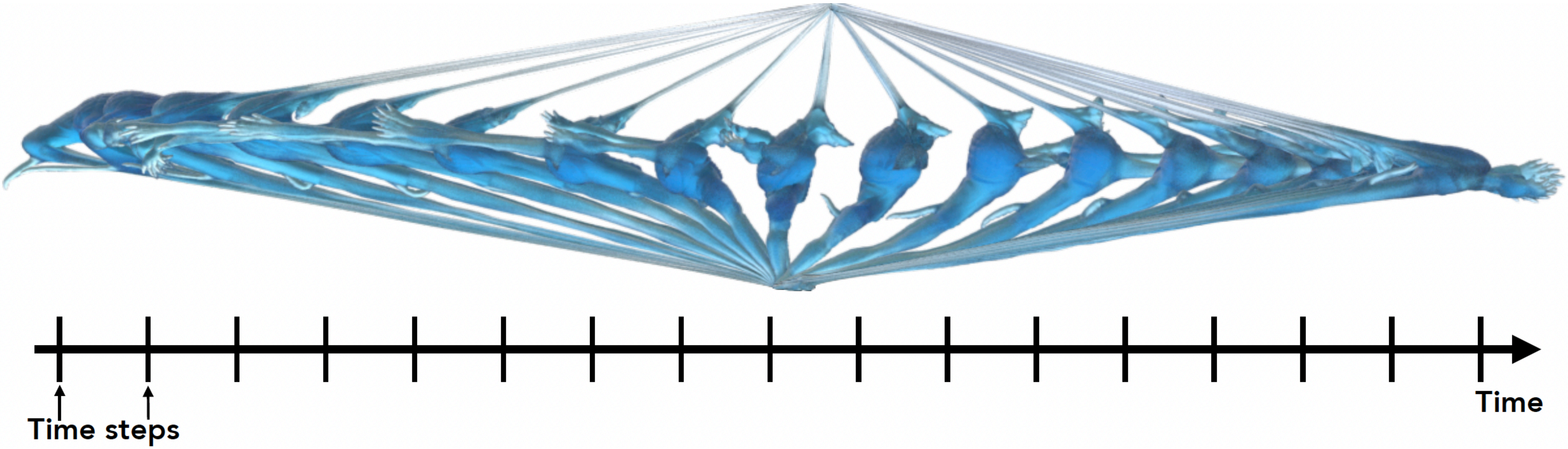
Minchen Li, 01/16/2025



Lec 1: Shape Representation and Time Integration

15-763: Physics-Based Animation of Solids and Fluids (S25)

[Ad] SIGGRAPH 2025

Student Volunteer

- https://sv.siggraph.org/
- Free highest-level registration
- A chance of travel financial aid

Tentative Schedule (S25)

Week 1-2: A quick start

- **Week 1**
  - Jan 14: Lec0: Course Logistics
  - Jan 16: Lec1: Spatial and Temporal Discretization
- **Week 2**
  - Jan 21: Lec2: Mass-Spring Systems

Week 2-4: Boundary treatment

  - Jan 23: Lec3: Dirichlet Boundary Conditions
- **Week 3**
  - Jan 28: Lec4: Normal Contact Force
  - Jan 30: Lec5: Friction
- **Week 4**
  - Feb 4: Lec6: Moving Boundary Conditions

Week 4-6: A bit continuum mechanics

  - Feb 6: Lec7: Strain Energy
- **Week 5**
  - Feb 11: Lec8: Stress and Its Derivative
  - Feb 13: Project Proposal Presentation
- **Week 6**
  - Feb 18: Lec9: Governing Equations
  - Feb 20: Lec10: Finite Element Discretization

Week 7-11: Special topics

- **Week 7**
  - Feb 25: Lec11: Frictional Self-Contact
  - Feb 27: Lec12: Reduced-Order Model
- **Week 8**: *Spring Break, no classes*
- **Week 9**
  - Mar 11: Lec13: Codimensional Solids
  - Mar 13: Lec14: Fluid Simulation Fundamentals, SPH
- **Week 10**
  - Mar 18: Lec15: Hybrid Lagrangian/Eulerian Methods
  - Mar 20: Midterm Progress Presentation
- **Week 11**
  - Mar 25: Lec16: Plasticity
  - Mar 27: *SIGGRAPH committee meeting, no class*

Week 12-15: Paper presentations

- **Week 12**
  - Apr 1: Paper Presentation
  - Apr 3: *Spring Carnival, no class*
- **Week 13**
  - Apr 8: Paper Presentation
  - Apr 10: Paper Presentation
- **Week 14**
  - Apr 15: Paper Presentation
  - Apr 17: Paper Presentation
- **Week 15**
  - Apr 22: Paper Presentation
  - Apr 24: Final Project Presentation
- Final report due May 5 at 23:59 ET

Project checkpoints & presentations:
Week 5, 10, 15

Today:

- Shape Representation

  Options, pros and cons, application

- Time Integration
  - Methods

    Stability, efficiency, accuracy

  - Solver

    Convergence & robustness

Today:

- Shape Representation

  Options, pros and cons, application

- Time Integration
  - Methods

    Stability, efficiency, accuracy

  - Solver

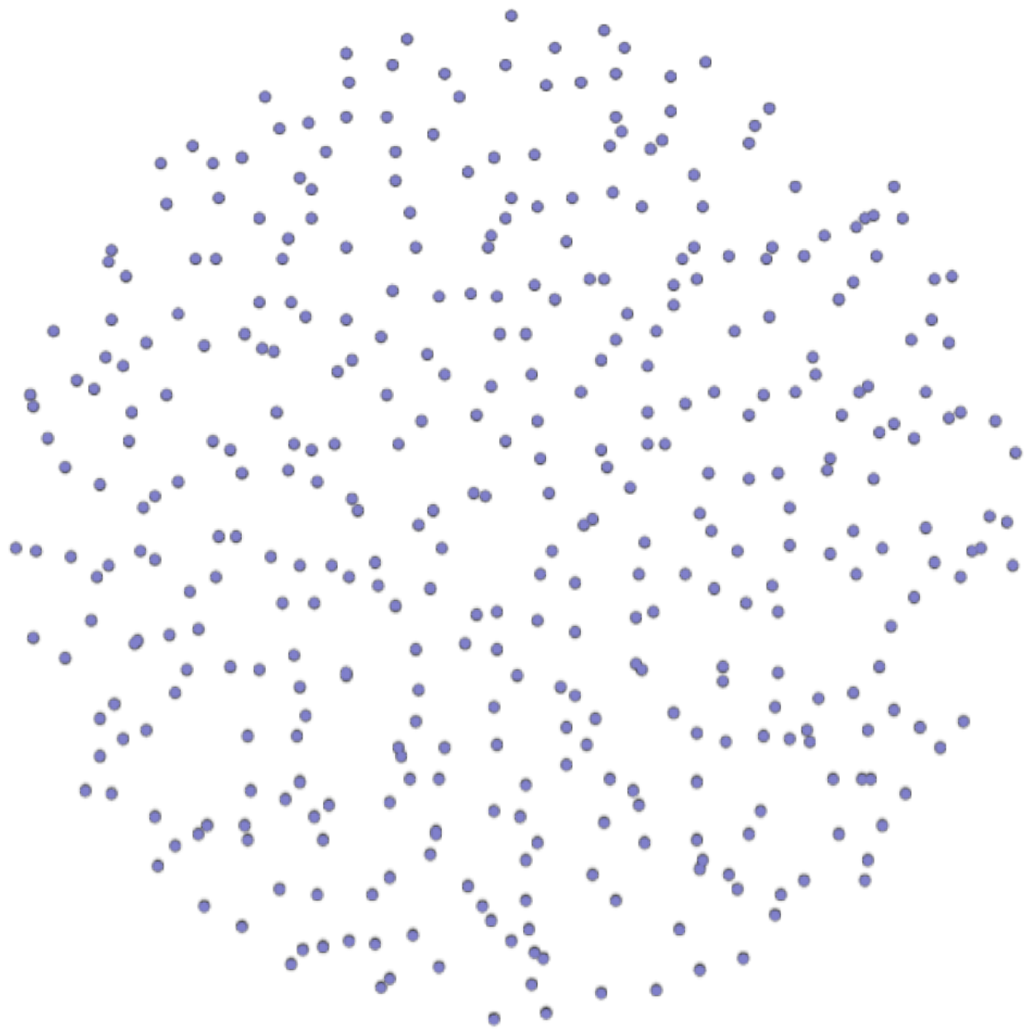
    Convergence & robustness

Shape Representation

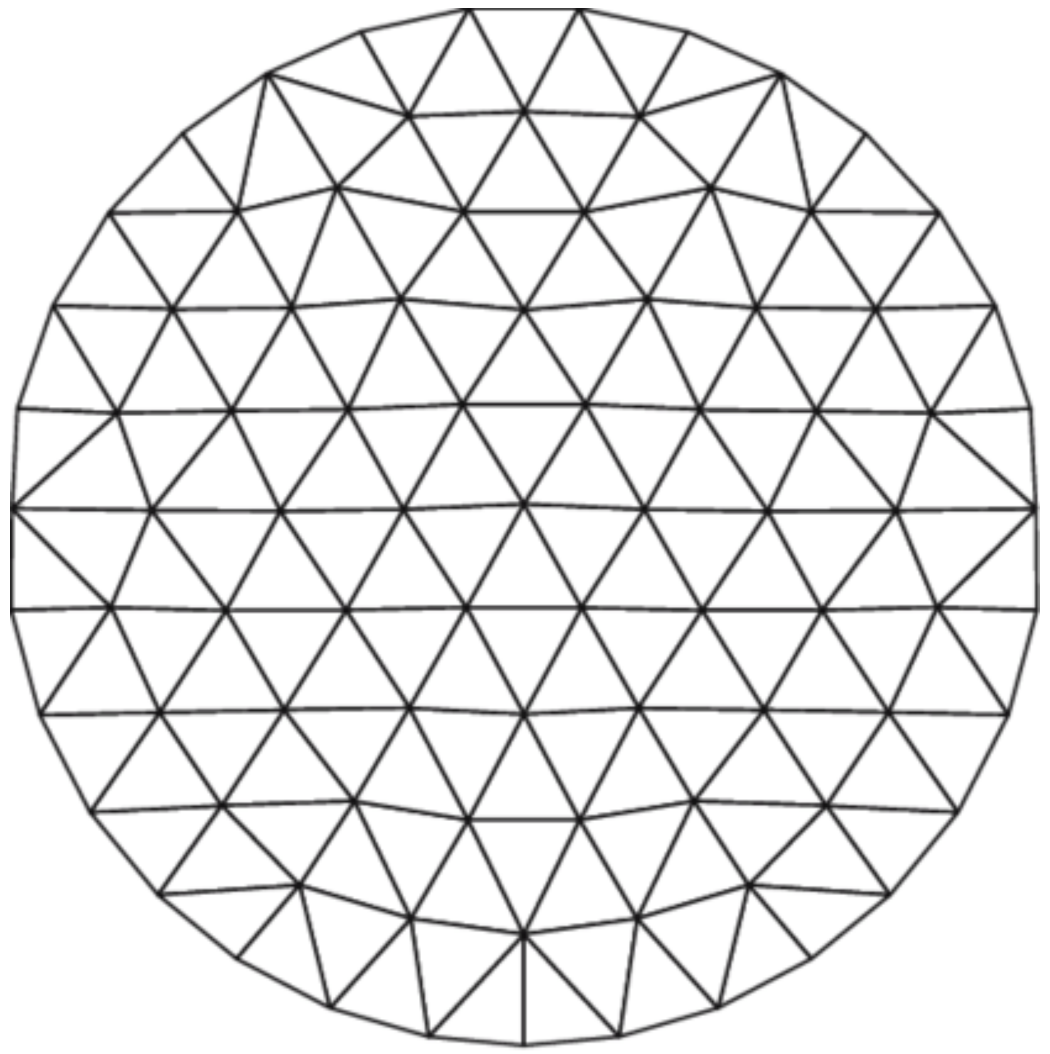
Example: How to represent a disk in 2D?

$$\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$$

Algebraic equation



Particles



Triangle mesh

3.8	3.1	2.5	2.0	1.7	1.5	1.5	1.7	2.0	2.5	3.1	3.8
3.1	2.4	1.7	1.1	0.7	0.5	0.5	0.7	1.1	1.7	2.4	3.1
2.5	1.7	0.9	0.3	-0.2	-0.5	-0.5	-0.2	0.3	0.9	1.7	2.5
2.0	1.1	0.3	-0.5	-1.1	-1.5	-1.5	-1.1	-0.5	0.3	1.1	2.0
1.7	0.7	-0.2	-1.1	-1.9	-2.4	-2.4	-1.9	-1.1	-0.2	0.7	1.7
1.5	0.5	-0.5	-1.5	-2.4	-3.3	-3.3	-2.4	-1.5	-0.5	0.5	1.5
1.5	0.5	-0.5	-1.5	-2.4	-3.3	-3.3	-2.4	-1.5	-0.5	0.5	1.5
1.7	0.7	-0.2	-1.1	-1.9	-2.4	-2.4	-1.9	-1.1	-0.2	0.7	1.7
2.0	1.1	0.3	-0.5	-1.1	-1.5	-1.5	-1.1	-0.5	0.3	1.1	2.0
2.5	1.7	0.9	0.3	-0.2	-0.5	-0.5	-0.2	0.3	0.9	1.7	2.5
3.1	2.4	1.7	1.1	0.7	0.5	0.5	0.7	1.1	1.7	2.4	3.1
3.8	3.1	2.5	2.0	1.7	1.5	1.5	1.7	2.0	2.5	3.1	3.8

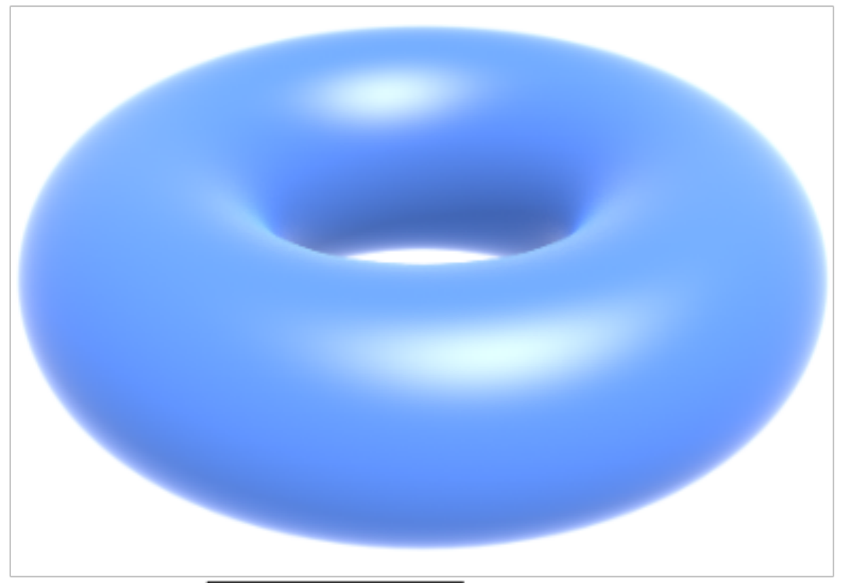
Signed distance field (SDF)

and more…

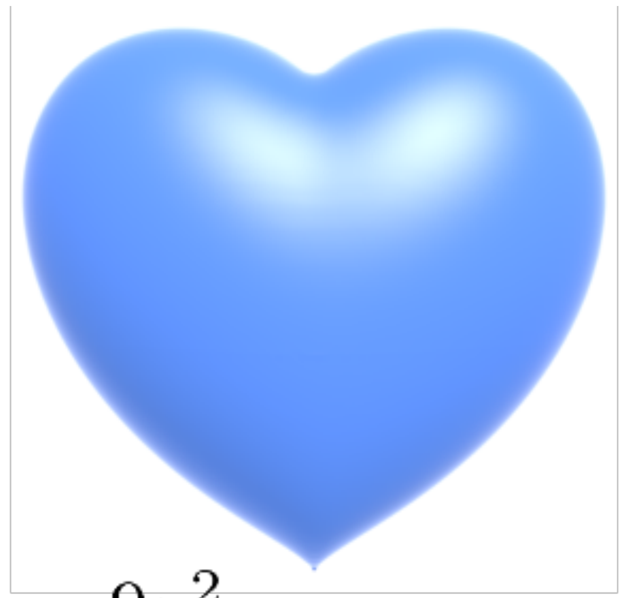
Shape Representation

Implicit — Algebraic Equation

- Pros and cons:
  - [+] Fast to compute distance
  - [+] Little storage needed
  - [-] Hard to represent complex shapes
  - [-] Hard to apply non-uniform deformation
- Often used to represent passive objects in the simulation
  - E.g. ground, spherical collision objects, etc.



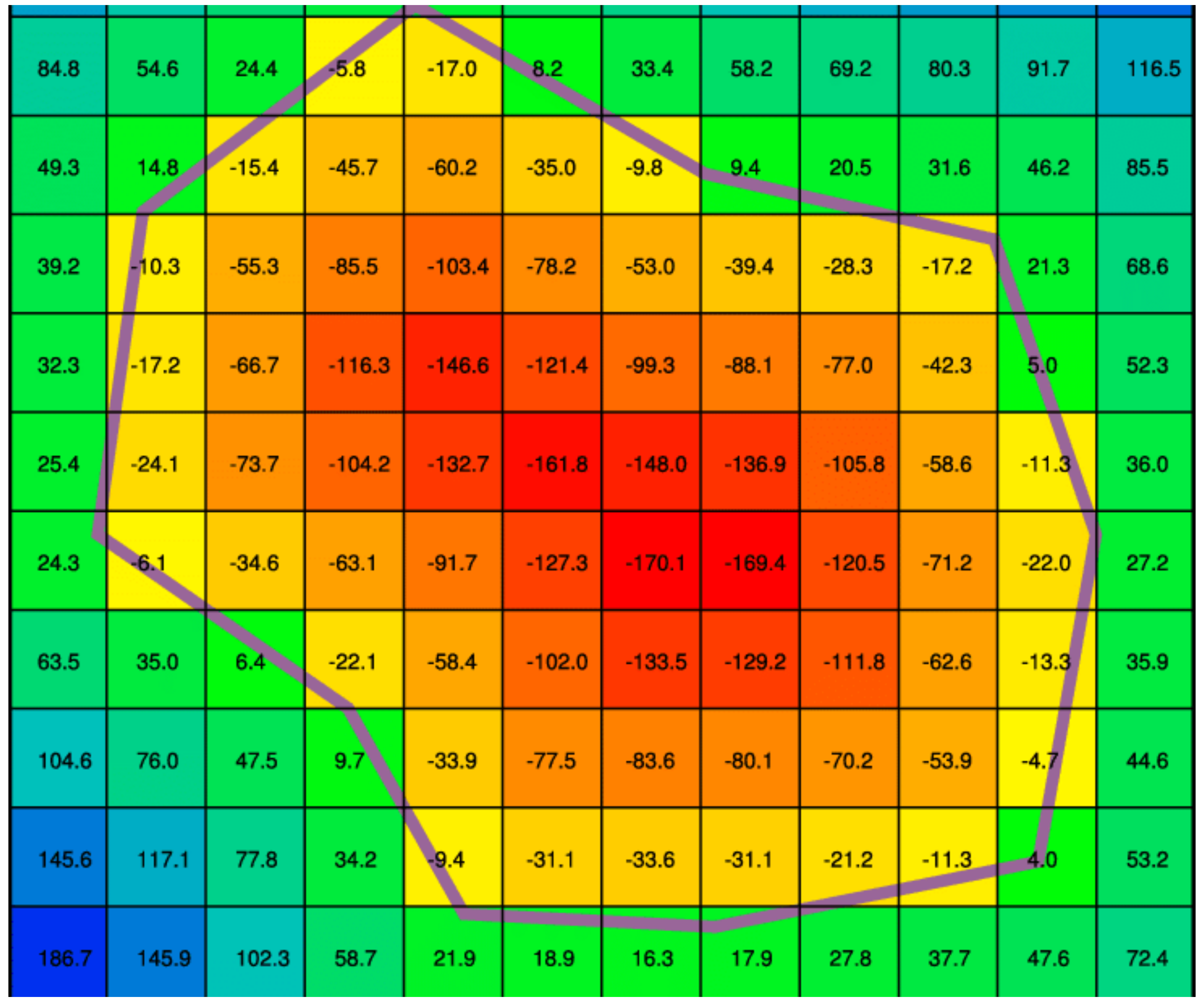
$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$



$$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 = x^2 z^3 + \frac{9y^2 z^3}{80}$$

Shape Representation

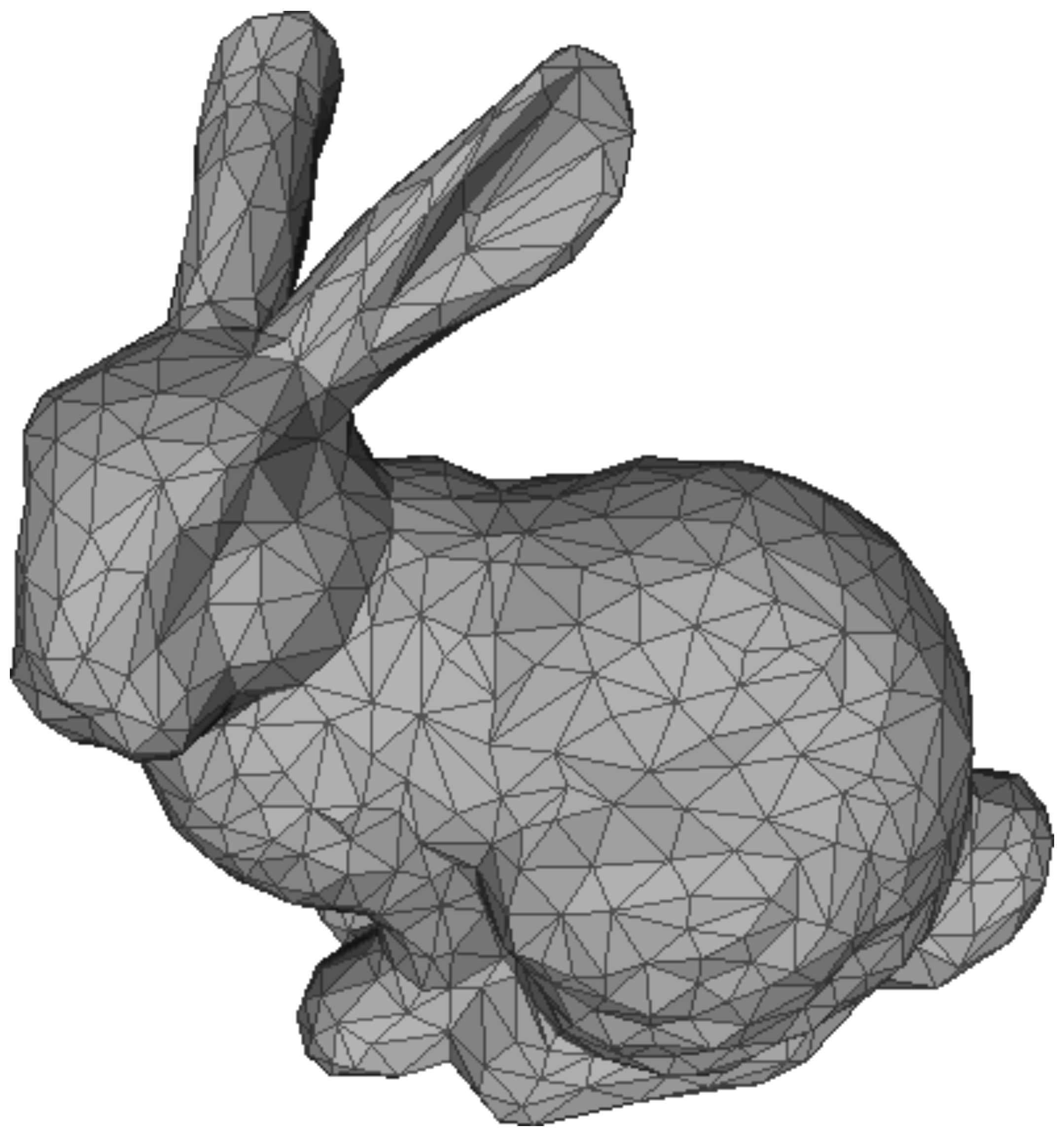
Implicit — Signed Distance Field (SDF)



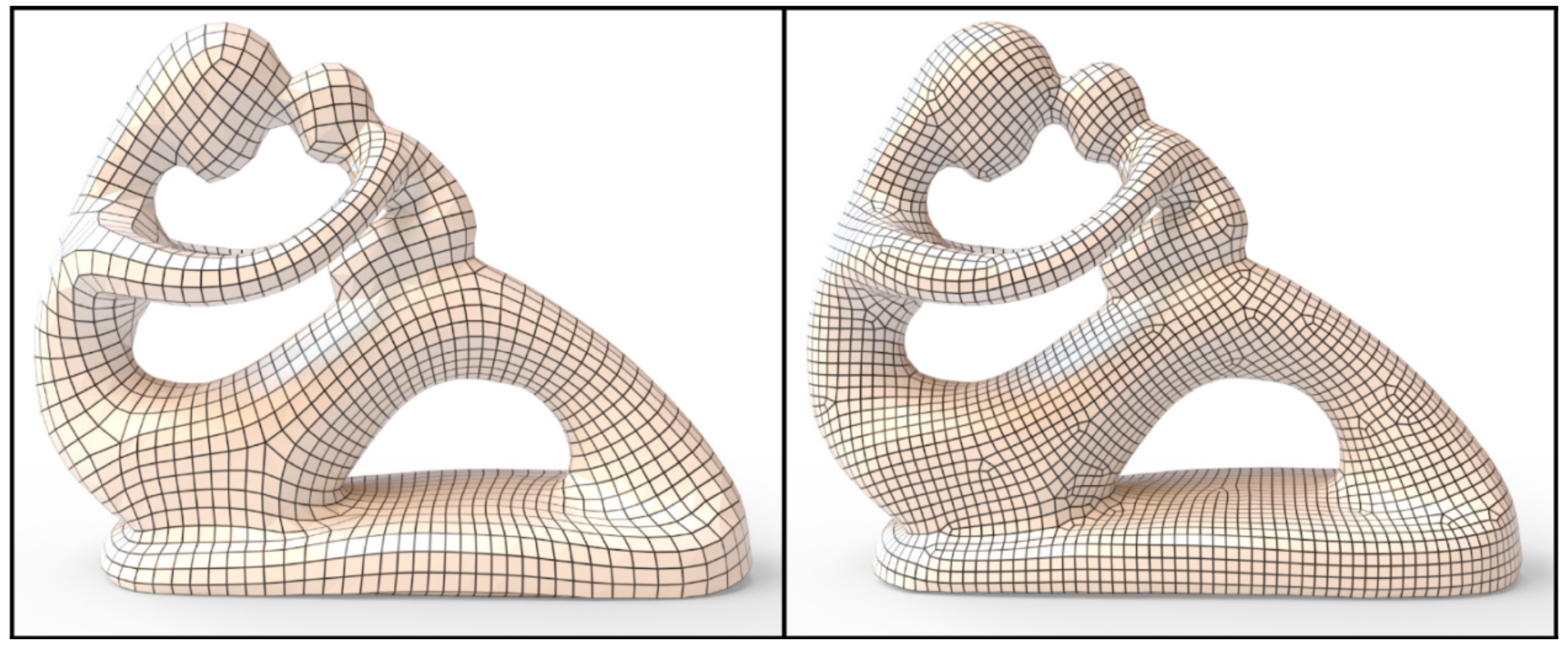
- Pros and cons:
  - [+] Fast to compute distance (interpolation from sampled distances)
  - [+] Structured, easy to
    - exploit memory locality
    - re-topology
  - [+/-] Can represent complex shapes (grid resolution dependent, grid-aligned artifacts)
  - [-] Takes efforts to track deformation
- Often used to
  - Represent passive objects with complex shapes
  - Track liquid surface in fluid sim.
  - Surface repairing, e.g. converting to water-tight surface

Shape Representation

Explicit — Mesh



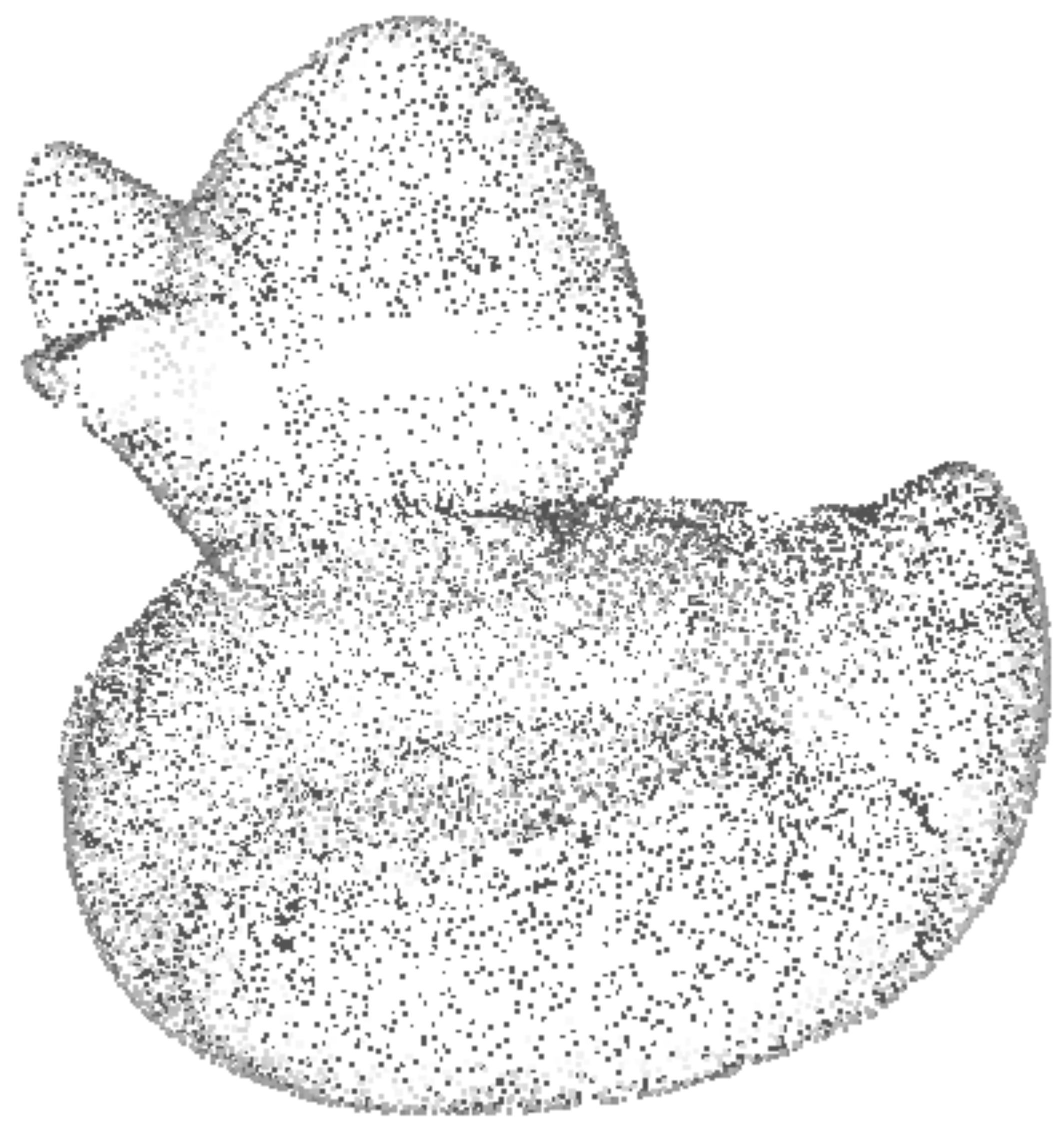
- Pros and cons:
  - [+] Good at representing complex shapes
  - [+] Easy to track deformation
  - [-] Slow to compute distance (needs to first locate the closest element)
  - [-] Unstructured, takes efforts to
    - exploit memory locality
    - re-topology



- Often used to represent all kinds of solids and simulation domains

Shape Representation

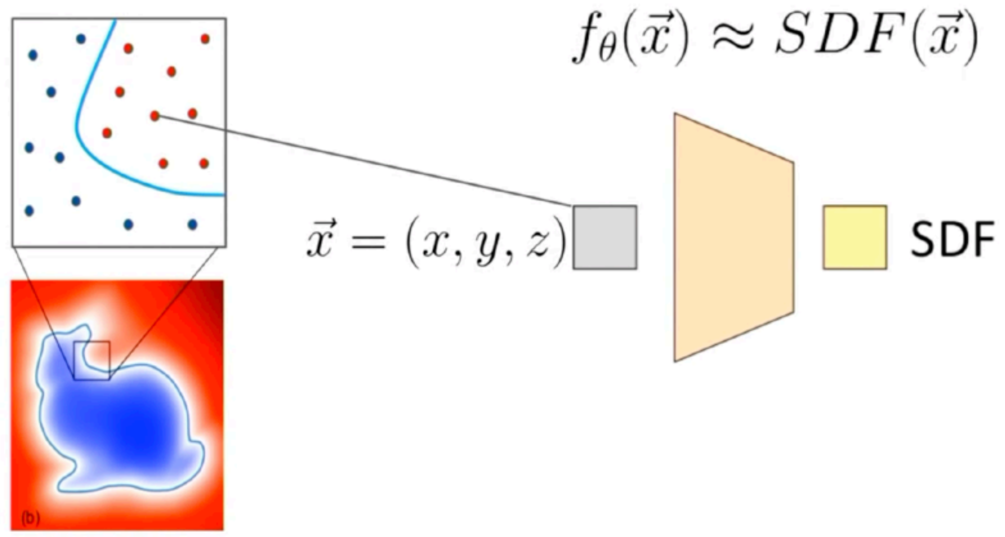
Explicit — Particles

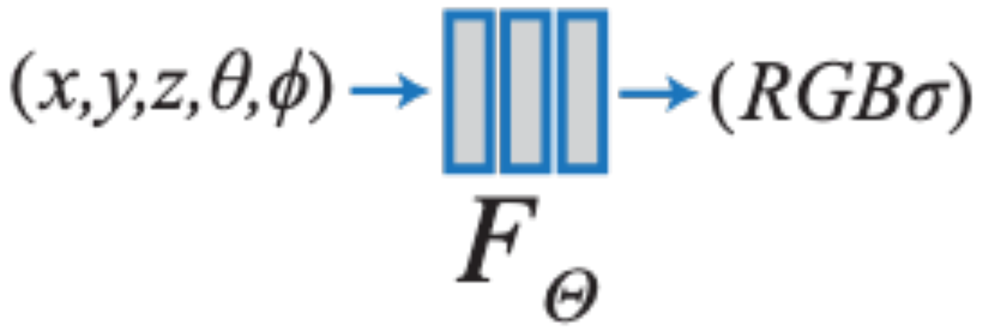


- Pros and cons:
  - [+] no structure, simple primitive: easy to
    - exploit memory locality (after organized based on grids)
    - re-sample
  - [+/-] Easy to track deformation (accuracy relies on large # particles)
  - [+/-] Can represent complex shapes (accuracy relies on large # particles)
  - [-] Slow to compute distance (needs to first locate the closest particle)
- Often used to track solid/fluid volumes under large deformation and topology changes

Shape Representation
Others

- 3D Gaussian Splatting (GS) [Kerbl et al 2023]
  - A set of ellipsoids, each with varying transparency and color
- Neural Implicit Functions, e.g.
  - NeRF [Mildenhall et al 2020]
  - DeepSDF [Park et al 2019]
- Simulations (both using particles):
  - 3D GS: PhysGaussian [Xie et al 2024]
  - NeRF: PIE-NeRF [Feng et al 2024]





Questions?

Today:

- Shape Representation

  Options, pros and cons, application

- Time Integration
  - Methods

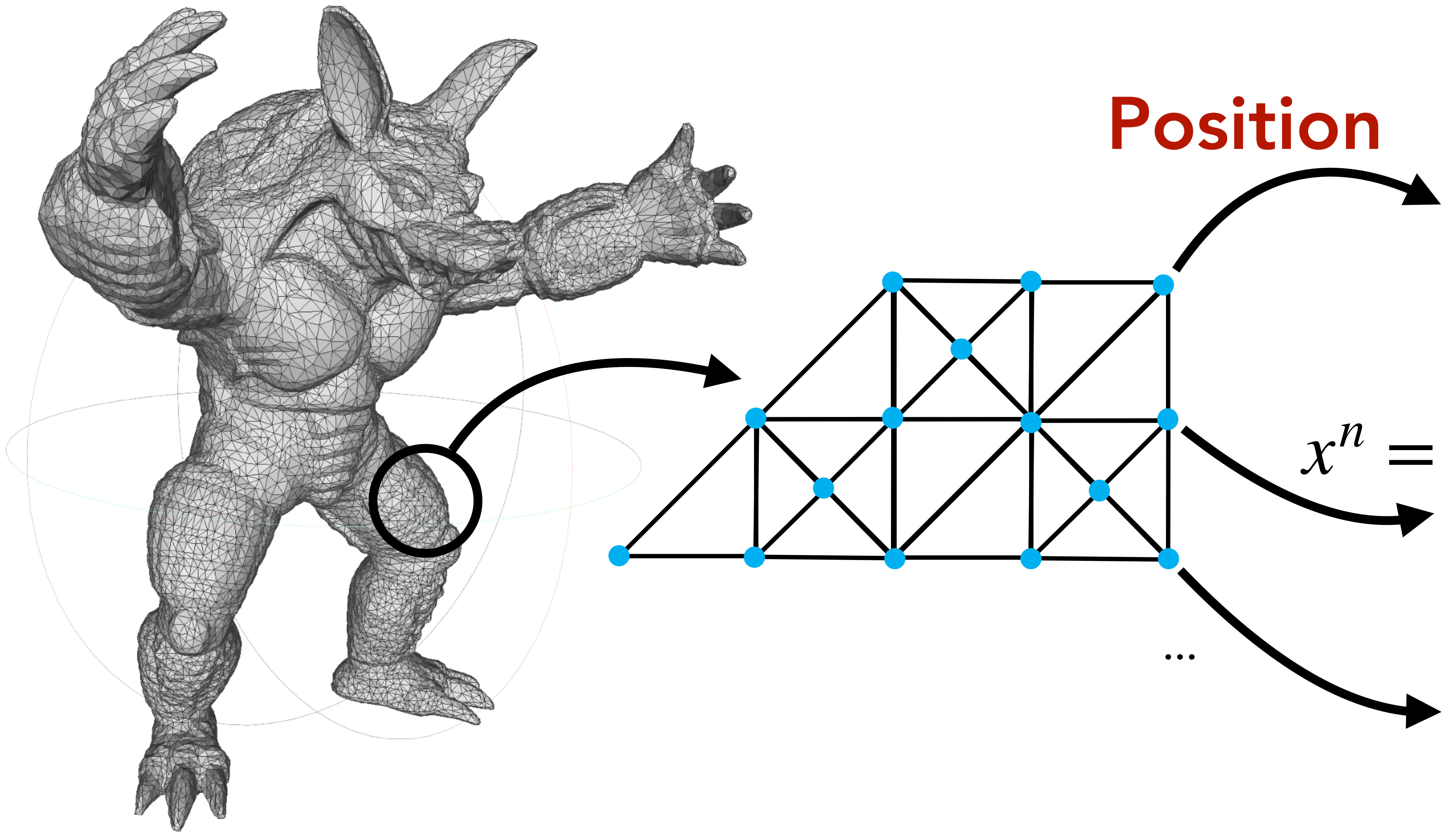
    Stability, efficiency, accuracy

  - Solver

    Convergence & robustness

State Variables

Tetrahedral Mesh as an Example



$$x^n = \begin{bmatrix} x^n_{0x} \\ x^n_{0y} \\ x^n_{0z} \\ x^n_{1x} \\ x^n_{1y} \\ x^n_{1z} \\ \vdots \end{bmatrix} \qquad v^n = \begin{bmatrix} v^n_{0x} \\ v^n_{0y} \\ v^n_{0z} \\ v^n_{1x} \\ v^n_{1y} \\ v^n_{1z} \\ \vdots \end{bmatrix}$$

Velocity

Newton’s 2nd Law

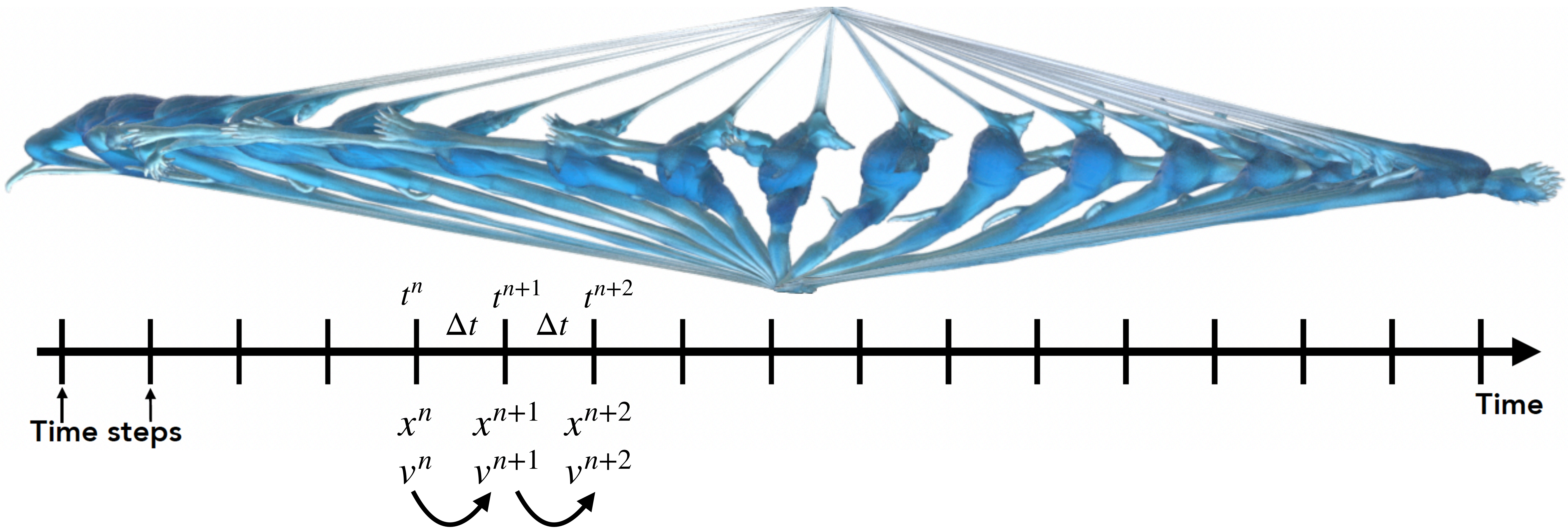
- The spatially discrete, temporally continuous form

$$\frac{dx}{dt} = v,$$
$$M\frac{dv}{dt} = f.$$

- Mass matrix (for now)

$$M = \begin{pmatrix} m_1 & & & \\ & m_1 & & \\ & & m_2 & \\ & & & m_2 \end{pmatrix}$$

Time Stepping (Time Integration)



Newton's 2nd Law (Temporally Discrete)

Forward Difference — Forward Euler

- Forward difference approximation on velocity and acceleration

$$\left(\frac{\mathbf{d}x}{\mathbf{d}t}\right)^n \approx \frac{x^{n+1}-x^n}{\Delta t} \qquad \left(\frac{\mathbf{d}v}{\mathbf{d}t}\right)^n \approx \frac{v^{n+1}-v^n}{\Delta t}$$

$$\left(f(t^n + \Delta t) = f(t^n) + \frac{\mathbf{d}f}{\mathbf{d}t}(t^n)\Delta t + O(\Delta t^2) \right)$$

Taylor's expansion

$$\frac{x^{n+1}-x^n}{\Delta t} = v^n, \qquad M\frac{v^{n+1}-v^n}{\Delta t} = f^n.$$

$$\Rightarrow$$

$$x^{n+1} = x^n + \Delta t v^n, \qquad v^{n+1} = v^n + \Delta t M^{-1} f^n.$$

Newton’s 2nd Law (Temporally Discrete)

Forward and Backward Difference — Symplectic Euler

- Forward difference on acceleration, backward difference on velocity

$$x^{n+1} = x^n + \Delta t v^{n+1}$$
$$v^{n+1} = v^n + \Delta t M^{-1} f^n$$

Newton's 2nd Law (Temporally Discrete)

Backward Difference — Backward Euler (or Implicit Euler)

- Backward difference approximation on velocity and acceleration

$$x^{n+1} = x^n + \Delta t v^{n+1},$$
$$v^{n+1} = v^n + \Delta t M^{-1} f^{n+1}$$

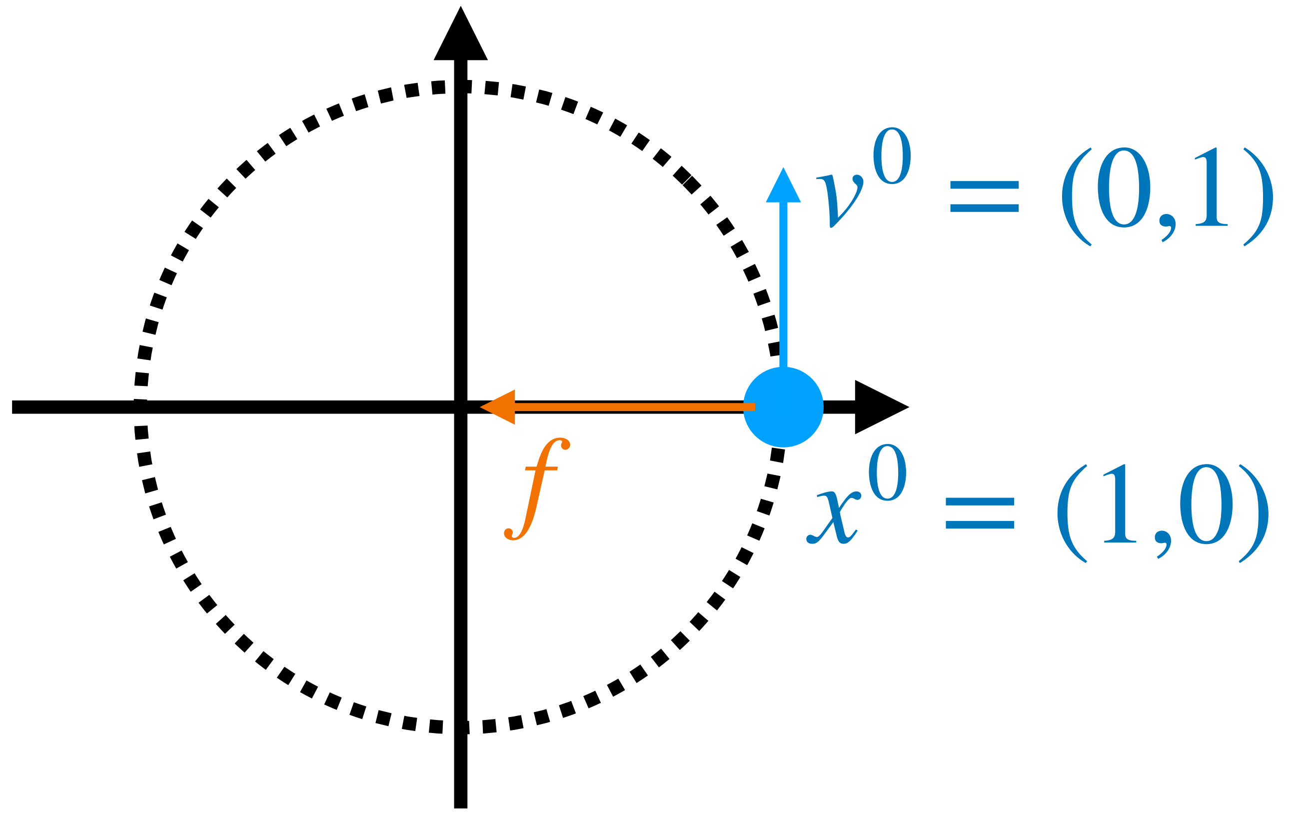
$$f^{n+1} = f(x^{n+1})$$

Needs to solve a system of equations:

$$M(x^{n+1} - (x^n + \Delta t v^n)) - \Delta t^2 f(x^{n+1}) = 0.$$

Stability of Forward, Symplectic, and Backward Euler

Example on a Uniform Circular Motion

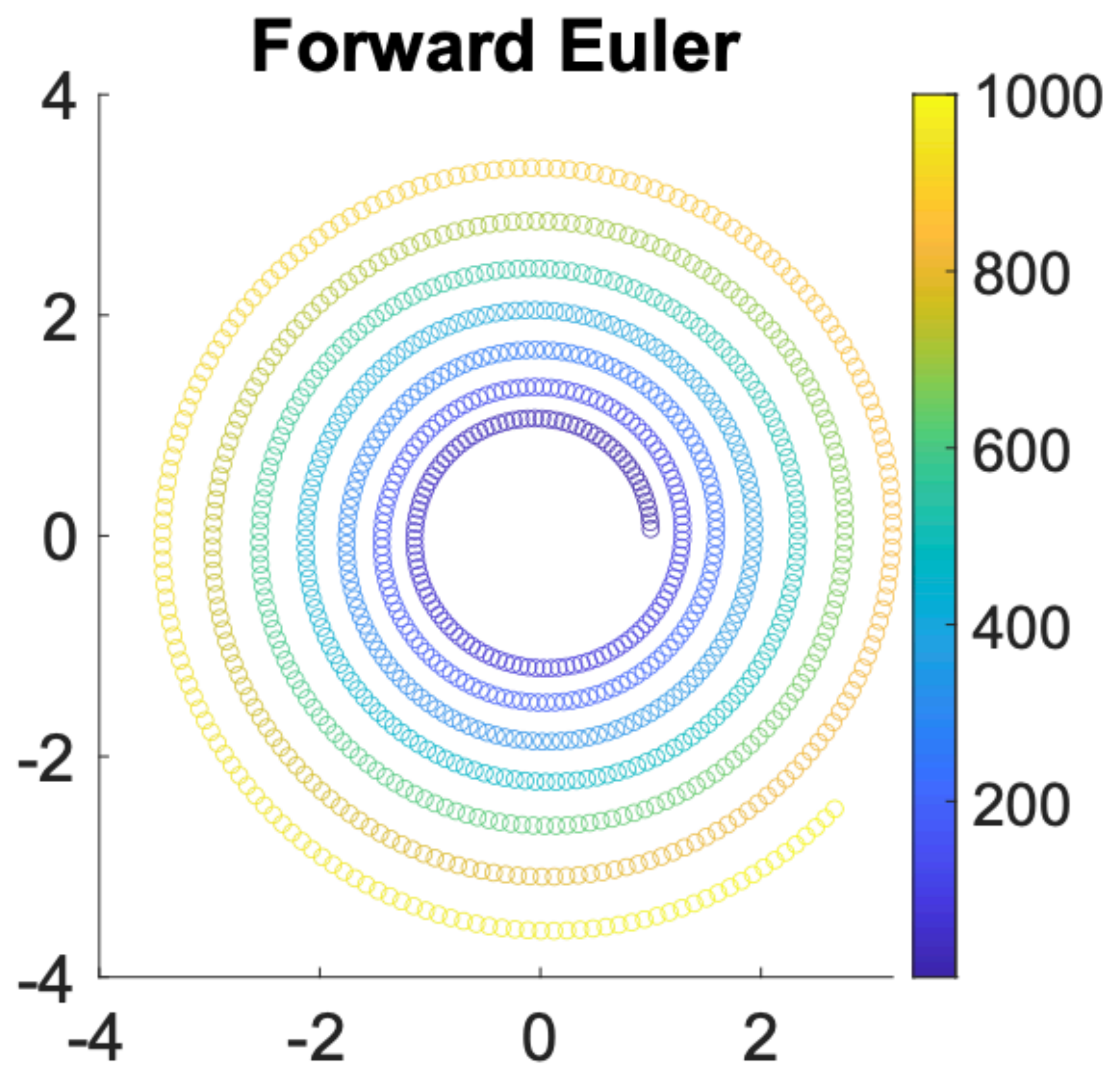


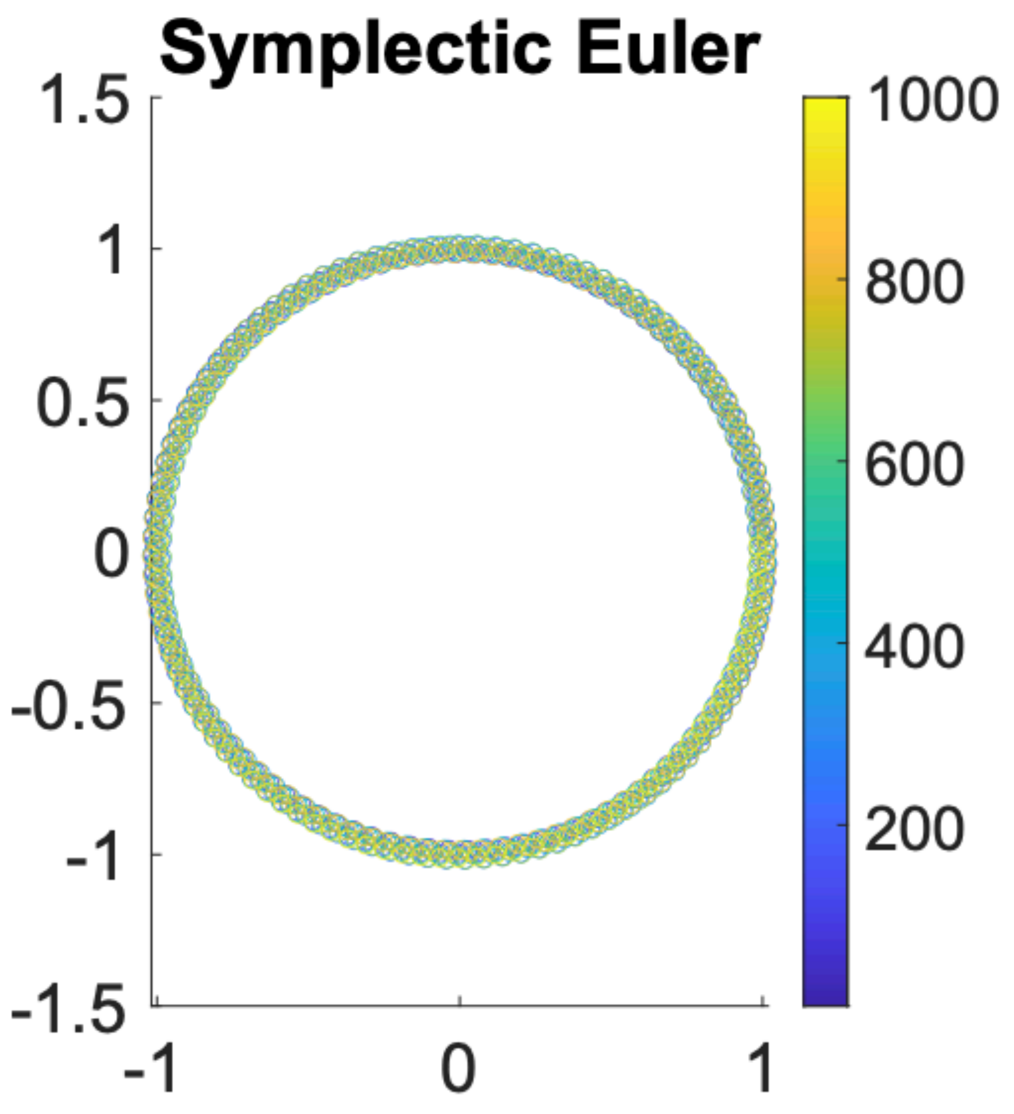
Problem Setup

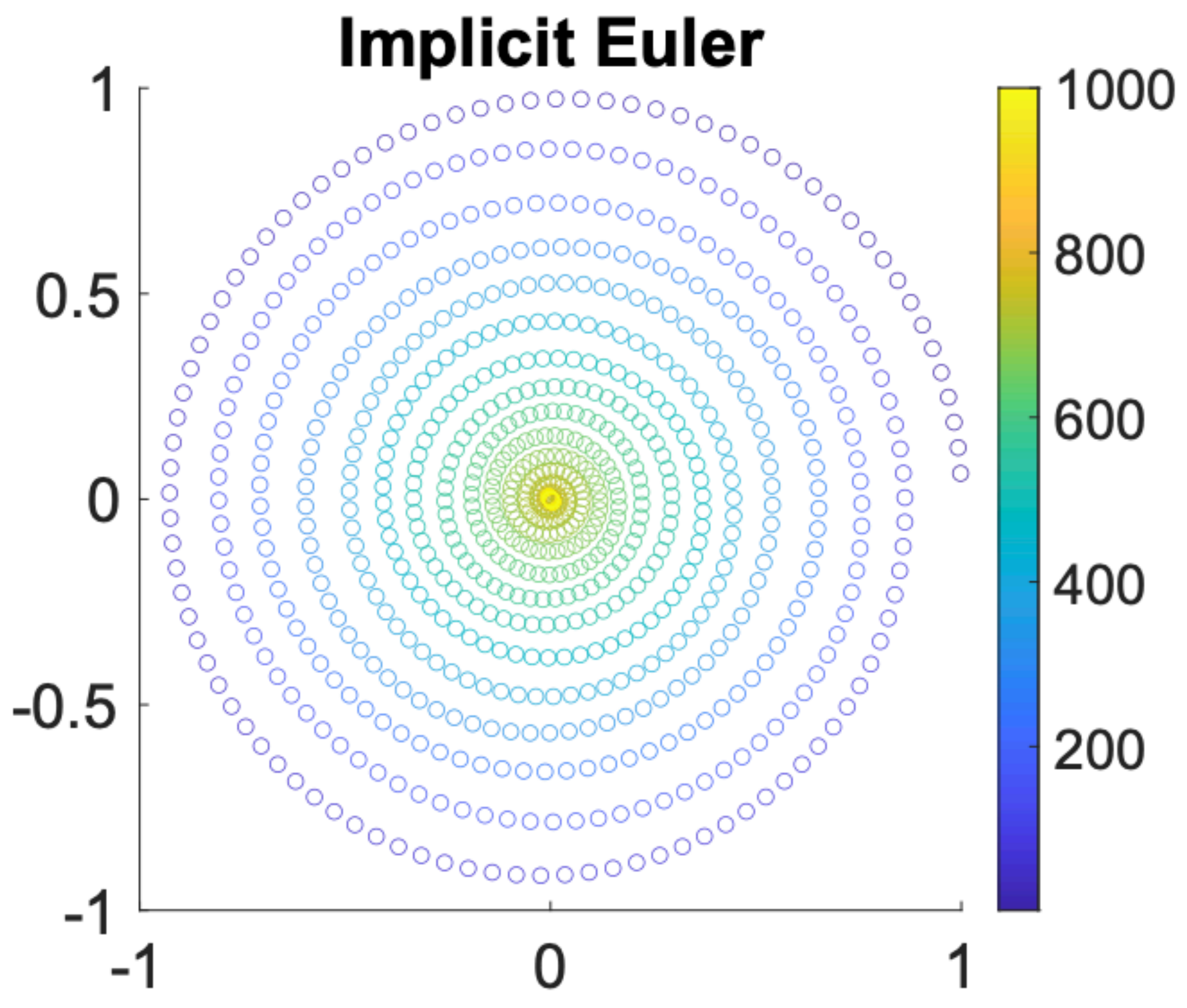
$$x^{n+1} = x^n + \Delta t v^n,$$
$$v^{n+1} = v^n + \Delta t M^{-1} f^n.$$

$$x^{n+1} = x^n + \Delta t v^{n+1}$$
$$v^{n+1} = v^n + \Delta t M^{-1} f^n$$

$$x^{n+1} = x^n + \Delta t v^{n+1},$$
$$v^{n+1} = v^n + \Delta t M^{-1} f^{n+1}$$







More Time Integration Methods

- Backward Difference Formula (BDF)
  - Uses configuration from multiple steps (e.g. $x^n, v^n, x^{n-1}, v^{n-1} \rightarrow x^{n+1}, v^{n+1}$)
  - BDF-2 is similarly stable as IE and conserves energy better, see experiments in Chen et al. [2022]
- Leapfrog
  - Uses staggered configurations (e.g. $x^n, v^{n+1/2} \rightarrow x^{n+1}, v^{n+3/2}$)
- Runge-Kutta Methods
- Exponential
  - Exponential integrators for stiff elastodynamic problems [Michels et al. 2014]
- *Comparison of high-order time integrators for deformable solids [Löschner et al 2020]*

Questions?

Today:

- Shape Representation

  Options, pros and cons, application

- Time Integration
  - Methods

    Stability, efficiency, accuracy

  - Solver

    Convergence & robustness

Newton's Method for Backward Euler

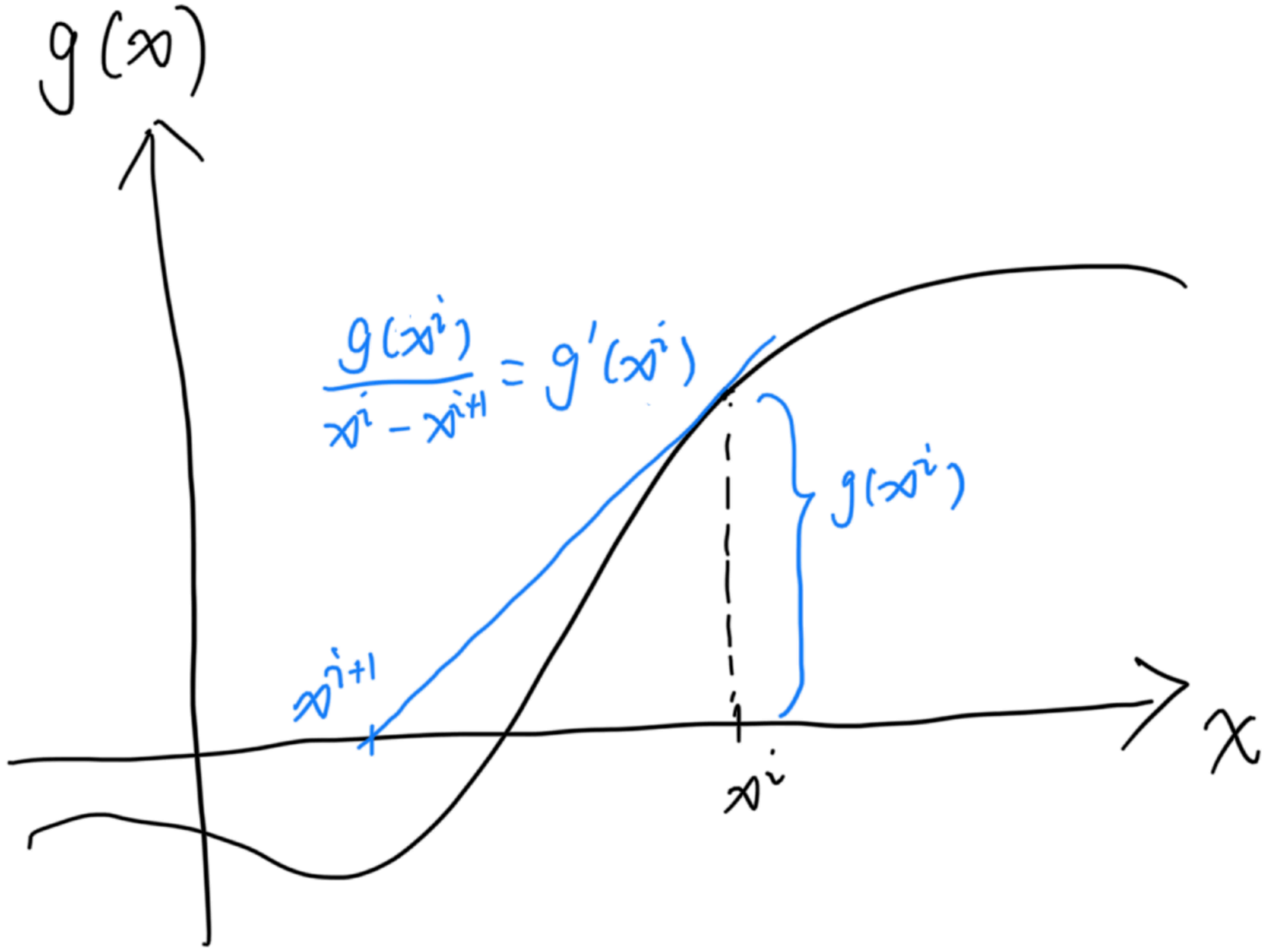
Formulation

Let $g(x) = M(x - (x^n + \Delta t v^n)) - \Delta t^2 f(x)$

We want to solve $g(x) = 0$

Newton's method in 1D:

- **Start from initial guess** x^0
- **For each iteration (until convergence)**
  - $x^{i+1} \leftarrow x^i - g(x^i)/g'(x^i)$



Newton's Method for Backward Euler

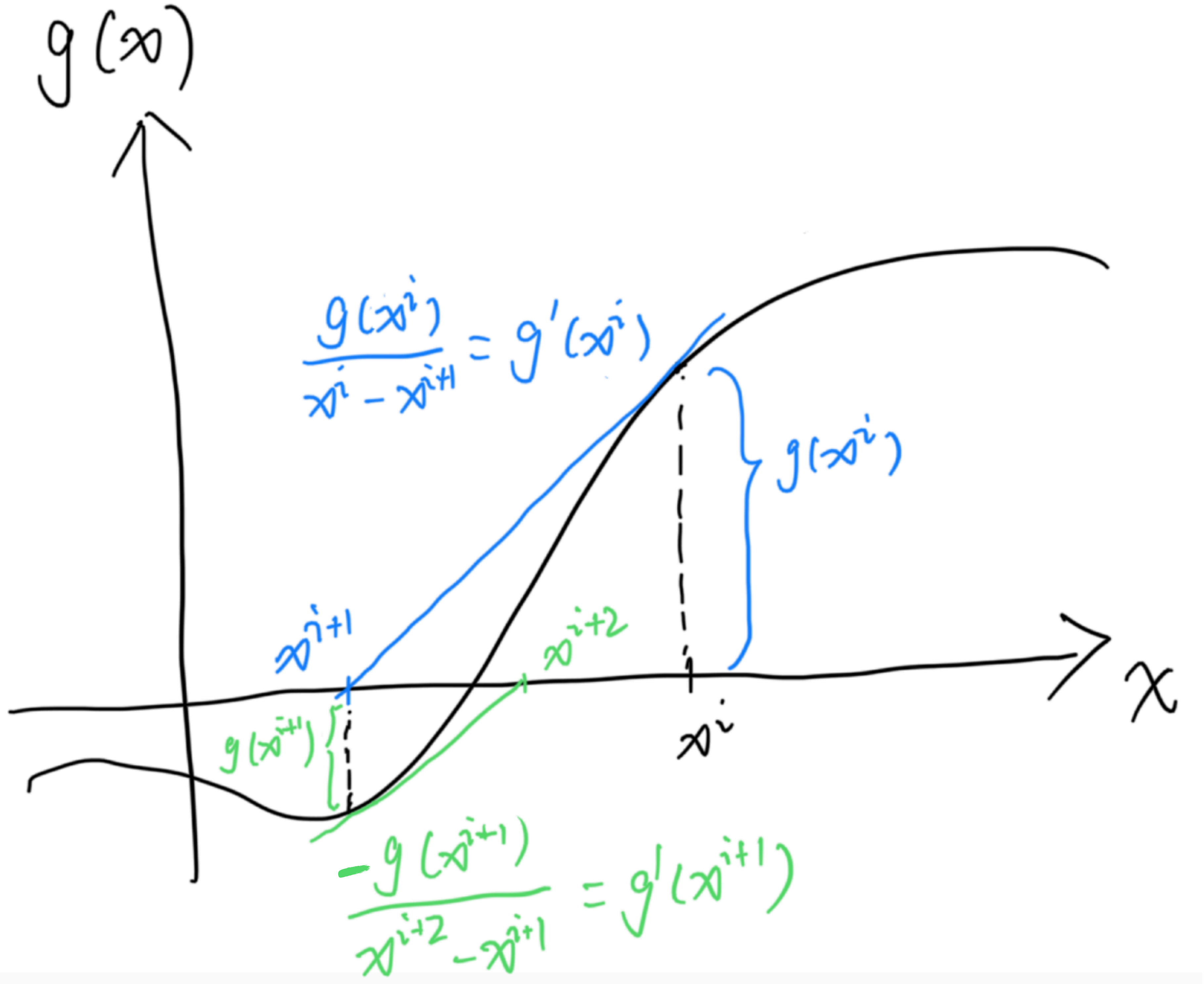
Formulation

Let $g(x) = M(x - (x^n + \Delta t v^n)) - \Delta t^2 f(x)$

We want to solve $g(x) = 0$

Newton's method in 1D:

- **Start from initial guess** x^0
- **For each iteration (until convergence)**
  - $x^{i+1} \leftarrow x^i - g(x^i)/g'(x^i)$



In higher dimensions:

$x^{i+1} \leftarrow x^i - (\nabla g(x^i))^{-1} g(x^i)$

Derivation:

Linearly approximate $g(x) = 0$ **at** x^i :

$g(x) = g(x^i) + \nabla g(x^i)(x - x^i)$

$g(x^{i+1}) \approx g(x^i) + \nabla g(x^i)(x^{i+1} - x^i) = 0$

Newton's Method for Backward Euler

Pseudo-code

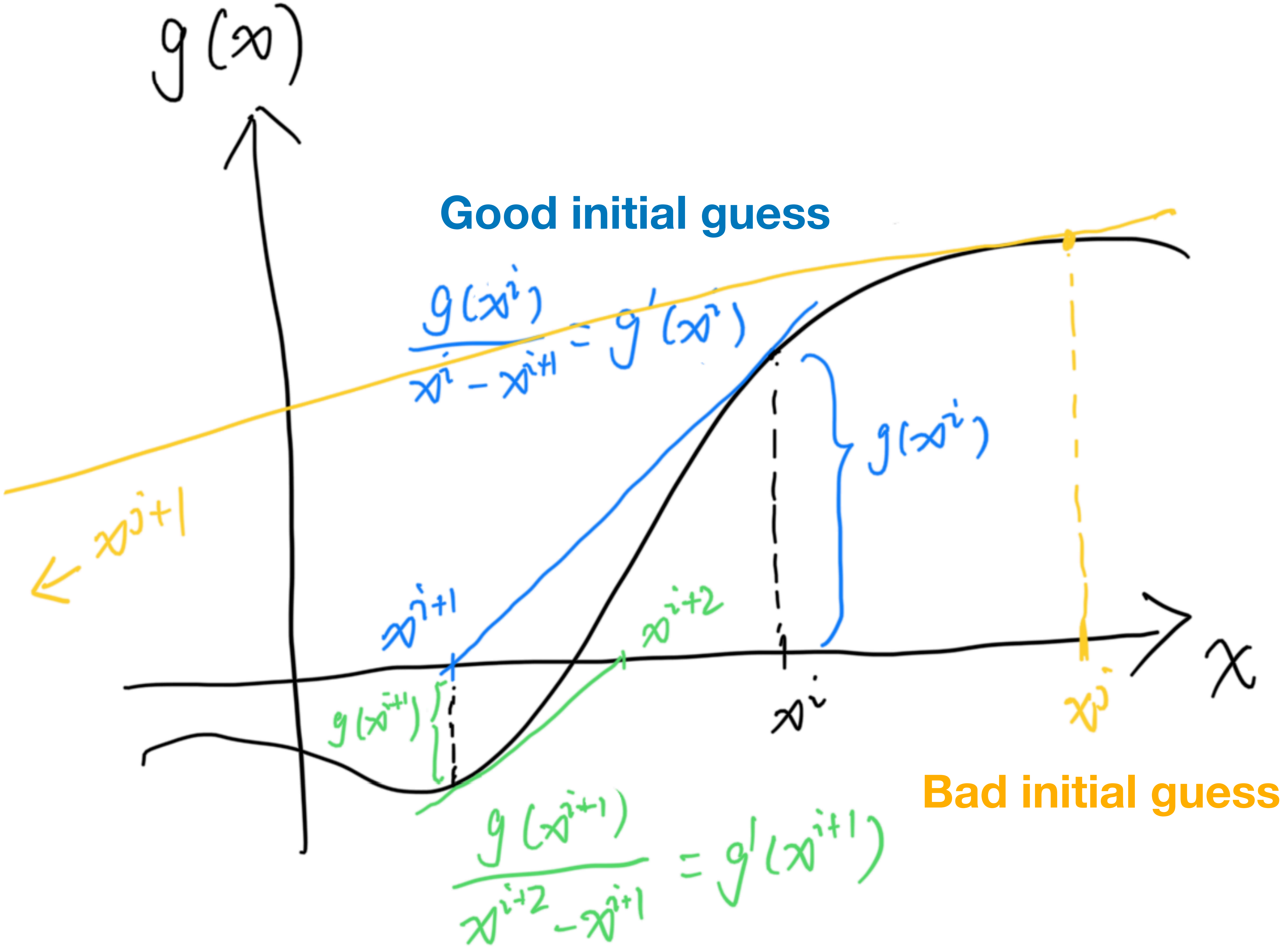
Algorithm 1: Newton's Method for Backward Euler Time Integration

Result: x^{n+1}, v^{n+1}

$x^i \leftarrow x^n$;
while $\|M(x^i - (x^n + \Delta t v^n)) - \Delta t^2 f(x^i)\| > \epsilon$ **do**
    solve $M(x - (x^n + \Delta t v^n)) - \Delta t^2 (f(x^i) + \nabla f(x^i)(x - x^i)) = 0$ for x;
    $x^i \leftarrow x$;
$x^{n+1} \leftarrow x^i$;
$v^{n+1} \leftarrow (x^{n+1} - x^n)/\Delta t$;

Convergence Issue of Newton's Method

Over-Shooting



Bad initial guess → **Simulation explodes!**

Optimization Time Integration

A Reformulation

$$x^{n+1} = \arg\min_x E(x)$$

$$\text{where } E(x) = \frac{1}{2}\|x - \tilde{x}^n\|_M^2 + \Delta t^2 P(x).$$

$$\tilde{x}^n = x^n + \Delta t v^n$$

$$\tfrac{1}{2}\|x - \tilde{x}^n\|_M^2 = \tfrac{1}{2}(x - \tilde{x}^n)^T M (x - \tilde{x}^n)$$

$$\tfrac{\partial P}{\partial x}(x) = -f(x)$$

At the local minimum of $E(x)$, $\frac{\partial E}{\partial x}(x^{n+1}) = 0$

$$M(x^{n+1} - (x^n + \Delta t v^n)) - \Delta t^2 f(x^{n+1}) = 0.$$

Optimization Time Integration

Newton's Method with Line Search

We want to solve $\nabla E(x) = 0$

Newton's method:

- **Start from initial guess** x^0
- **For each iteration (until convergence)**
  - $x^{i+1} \leftarrow x^i - (\nabla E(x^i))^{-1} \nabla E(x^i)$

Let $p = -(\nabla E(x^i))^{-1} \nabla E(x^i)$

Line Search along direction p**:**

$$\min_{\alpha} E(x^i + \alpha p)$$

$$x^{i+1} \leftarrow x^i + \alpha p$$

Theory:

If p **is a descent direction at** $x = x^i$ **(like** $-\nabla E(x^i)$**),**

$\exists \alpha > 0,\ s.t.\ E(x^i + \alpha p) < E(x^i)$

— need $\nabla^2 E(x)$ **to be symmetric positive-definite**

Idea:

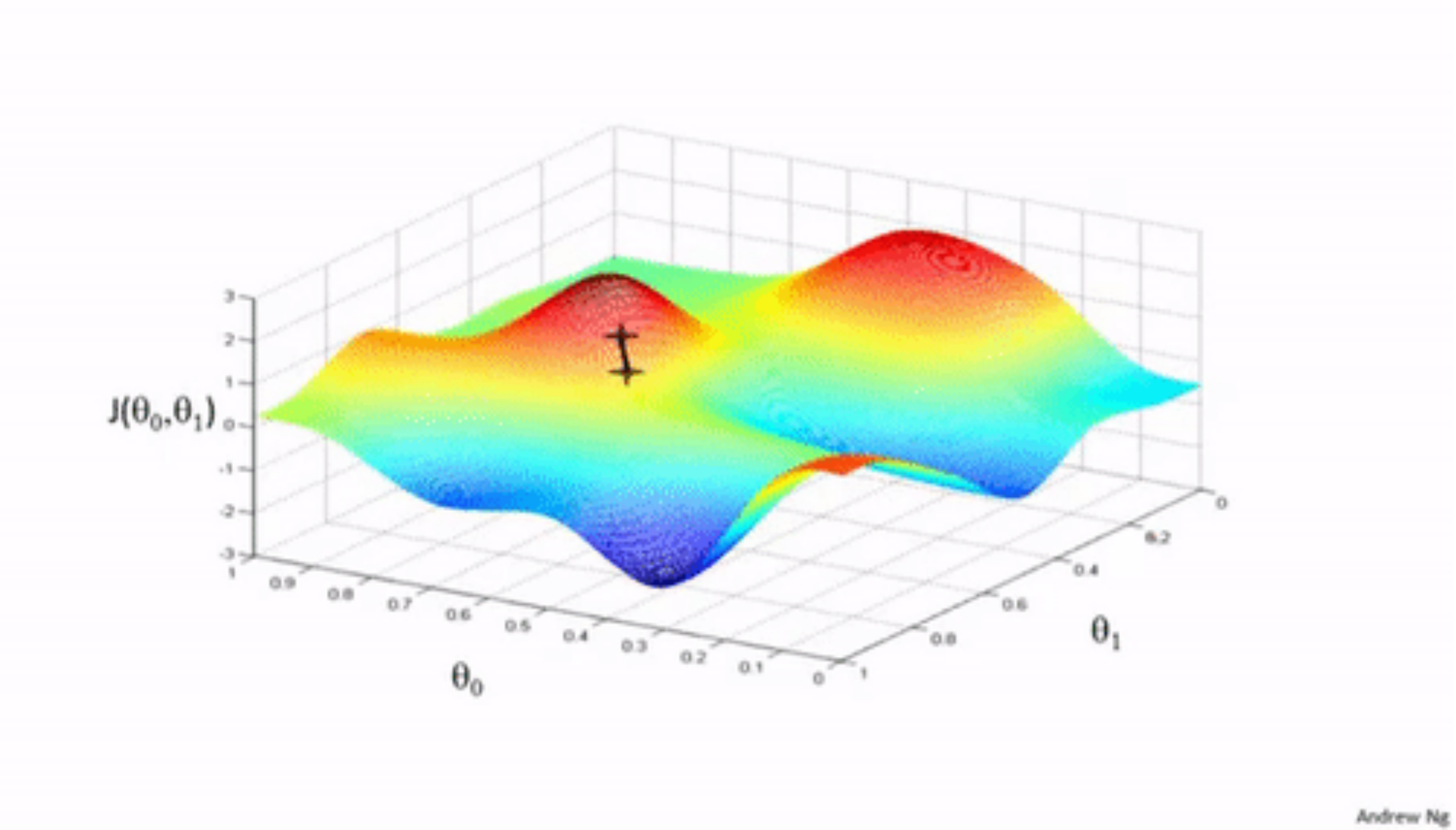
We can project $\nabla^2 E(x)$ **to a nearby SPD matrix for computing** p

Then we can ensure $E(x^{i+1}) < E(x^i)\ \forall i$

— no explosion!

Optimization Time Integration

Optimization Methods, 2D Illustration



Optimization Time Integration

Pseudo-code

Algorithm 3: Projected Newton Method for Backward Euler Time Integration

Result: x^{n+1}, v^{n+1}

$x^i \leftarrow x^n$;

do

  $P \leftarrow \text{SPDProjection}(\nabla^2 E(x^i))$;

  $p \leftarrow -P^{-1} \nabla E(x^i)$;

  $\alpha \leftarrow \text{BackTrackingLineSearch}(x^i, p)$; //

  $x^i \leftarrow x^i + \alpha p$;

while $\|p\|_\infty / h > \epsilon$;

$x^{n+1} \leftarrow x^i$;

$v^{n+1} \leftarrow (x^{n+1} - x^n)/\Delta t$;

Algorithm 2: Backtracking Line Search

Result: α

$\alpha \leftarrow 1$;

while $E(x^i + \alpha p) > E(x^i)$ **do**

  $\alpha \leftarrow \alpha/2$;

Questions?

Today:

- Shape Representation

  Options, pros and cons, application

- Time Integration
  - Methods

    Stability, efficiency, accuracy

  - Solver

    Convergence & robustness

Image Sources

- https://shaderfun.com/2018/03/25/signed-distance-fields-part-2-solid-geometry/
- https://pbr-book.org/3ed-2018/Monte_Carlo_Integration/2D_Sampling_with_Multidimensional_Transformations
- https://stackoverflow.com/questions/53406534/procedural-circle-mesh-with-uniform-faces
- https://www.matthewtancik.com/nerf
- https://www.youtube.com/watch?v=0ILnHe0xbZE&t=1404s
- https://xpandora.github.io/PhysGaussian/
- https://medium.com/@pushkarevmm/signed-distance-field-simple-example-with-raymarched-soft-shadows-in-unity-2b3fdf20218
- https://elmoatazbill.users.greyc.fr/point_cloud/index.html
- https://www.lix.polytechnique.fr/~maks/Verona_MPAM/TD/TD2/
- https://geometryfactory.com/products/igm-quad-meshing/