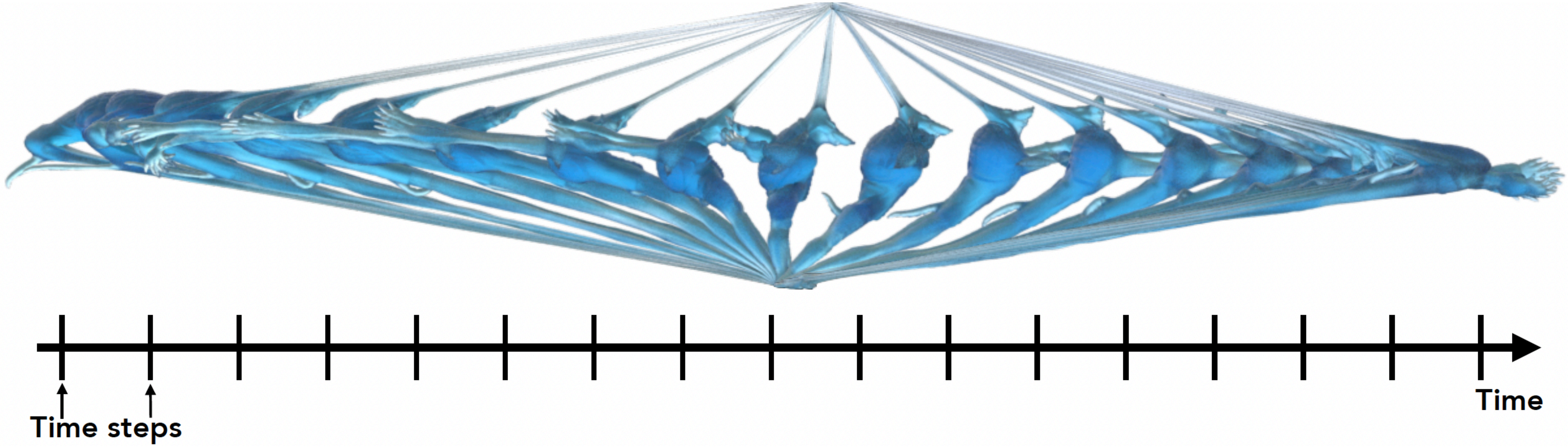
Minchen Li, 08/31/2023



Lec 2: Time Integration

15-769: Physically-based Animation of Solids and Fluids (F23)

Recap on Shape Representations

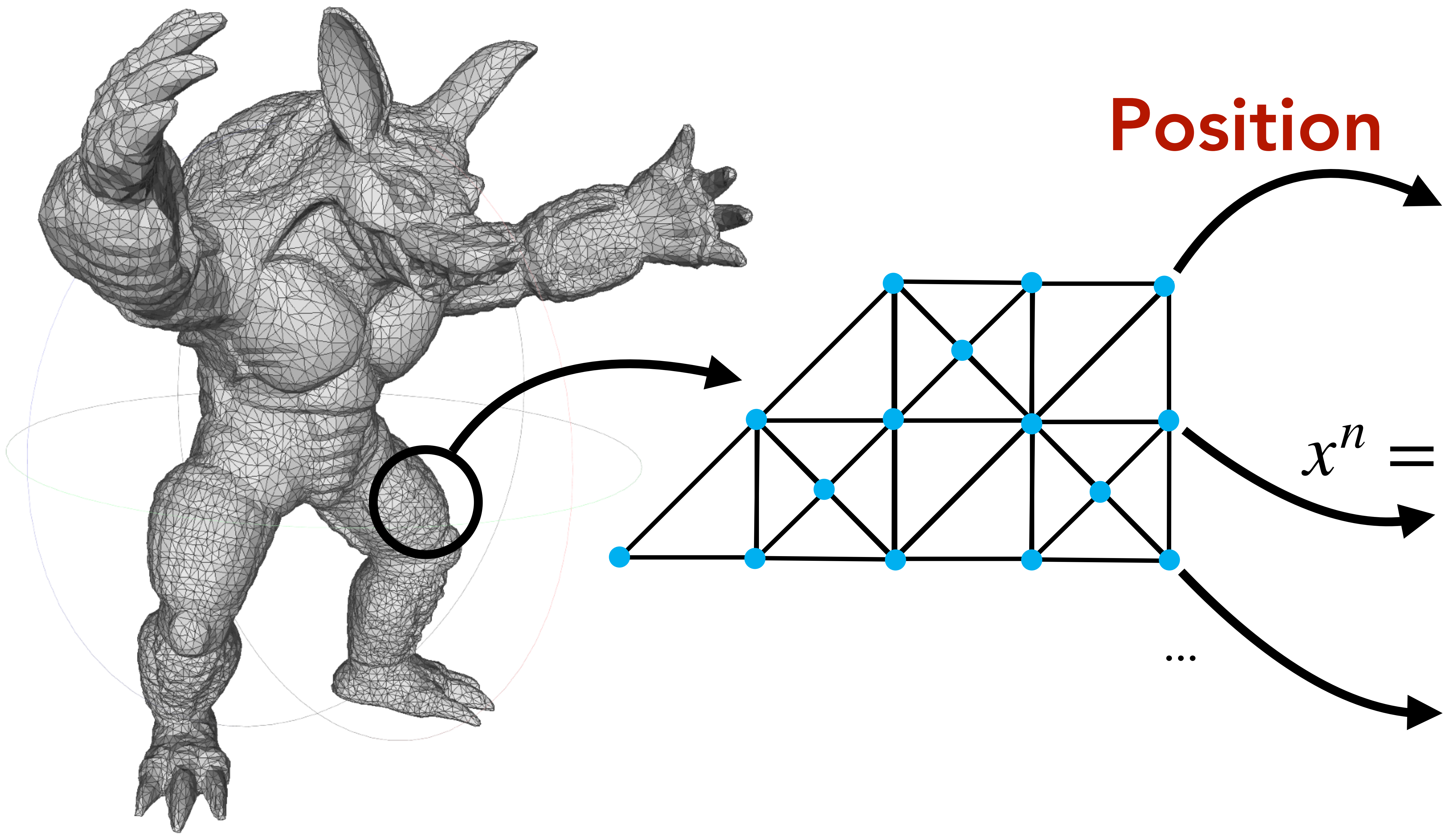
- Signed Distance Field (SDF) **Focus of the lectures:**
  - Particle-In-Cell (PIC) Method and Fluid-Implicit-Particle (FLIP) Method **for fluids**
  - Collision Obstacles
- Particles
  - Smoothed-Particle Hydrodynamics (SPH)
  - Discrete Element Method (DEM)
- Mesh
  - Finite Element Method (FEM) **for solids**
  - Boundary Element Method (BEM)
  - Eulerian Method (grid is a structured mesh) **for fluids**
- Hybrid
  - Material-Point Method (MPM)

(Adaptive) Spatial Discretization

- Eulerian solid simulation with c
- A material point method for sn
- Reformulating Hyperelastic Ma
- Hybrid grains: Adaptive coupli
- Fast Corotated Elastic SPH So
- Surface-Only Dynamic Deform
- Multi-Layer Thick Shells (2023
- High-Order Incremental Poten
- Adaptive Anisotropic Remeshi

Papers for all methods available on course website

Recap on Shape Representations



Position

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

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

Velocity

Newton's 2nd Law

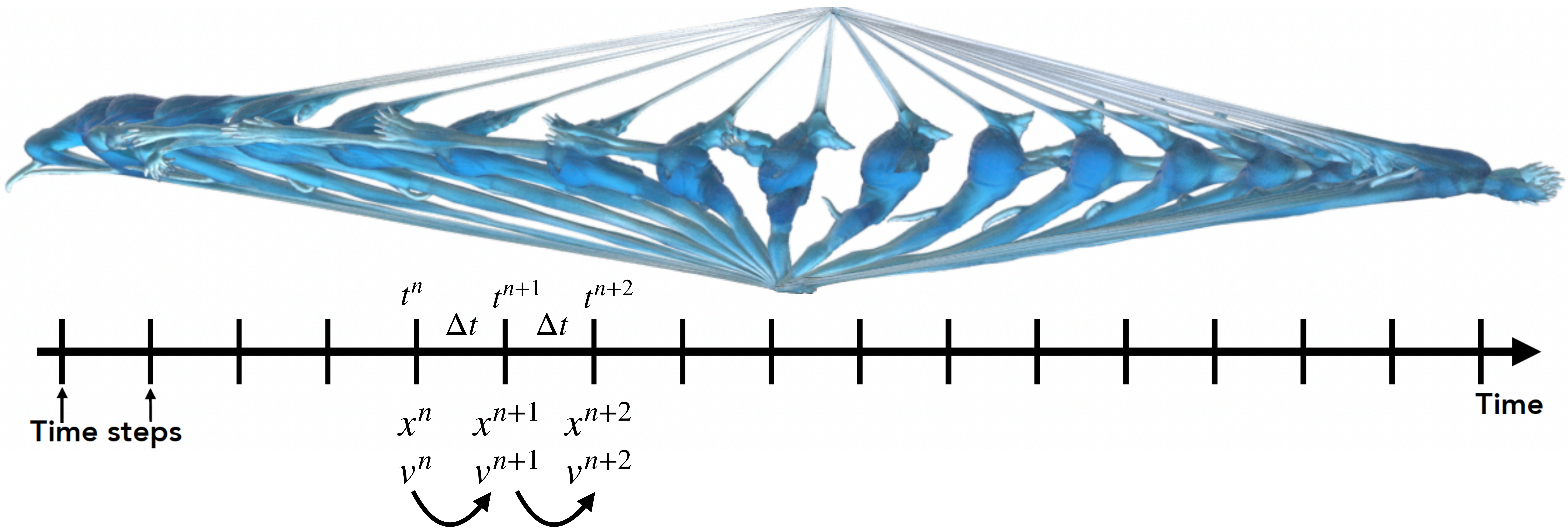
- The spatially discrete, temporally continuous form

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

- 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

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

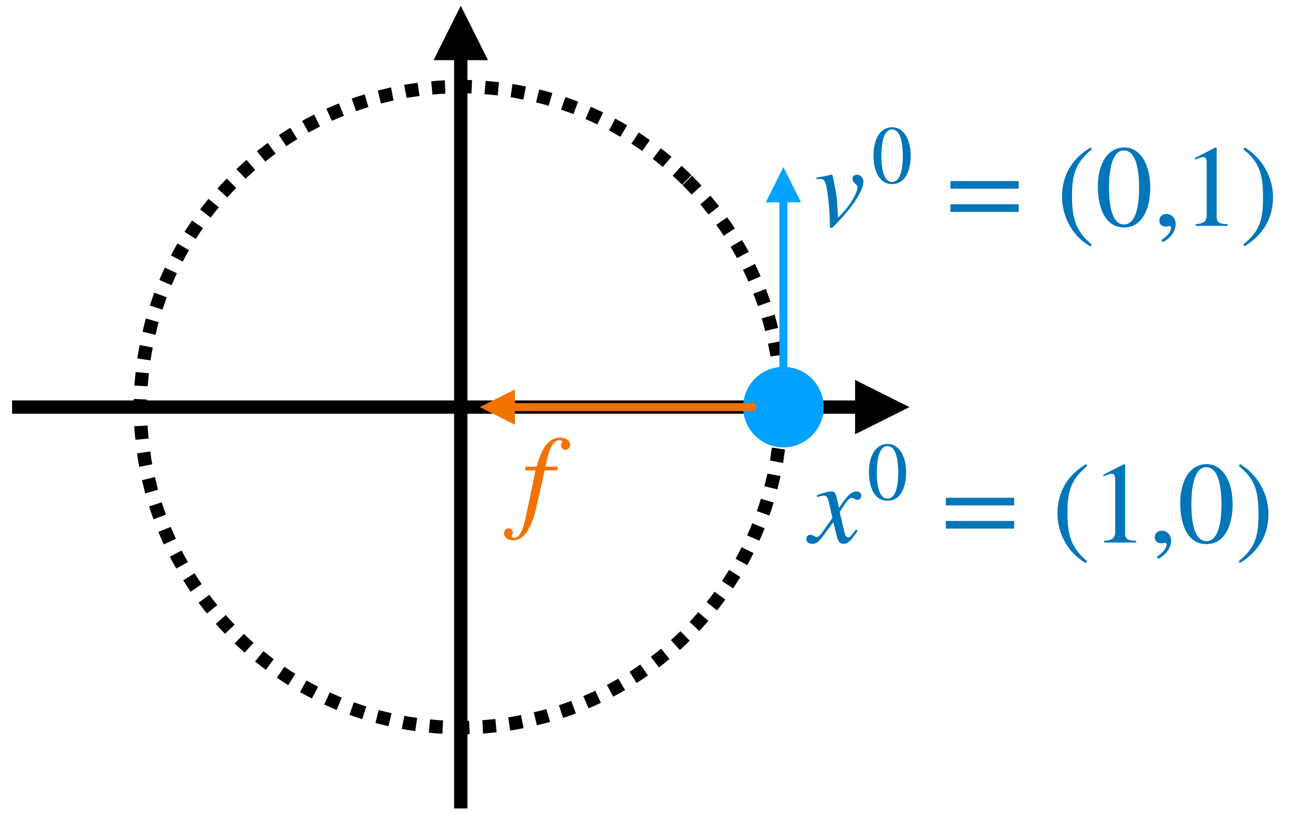
$$f^{n+1} = f(x^{n+1})$$

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

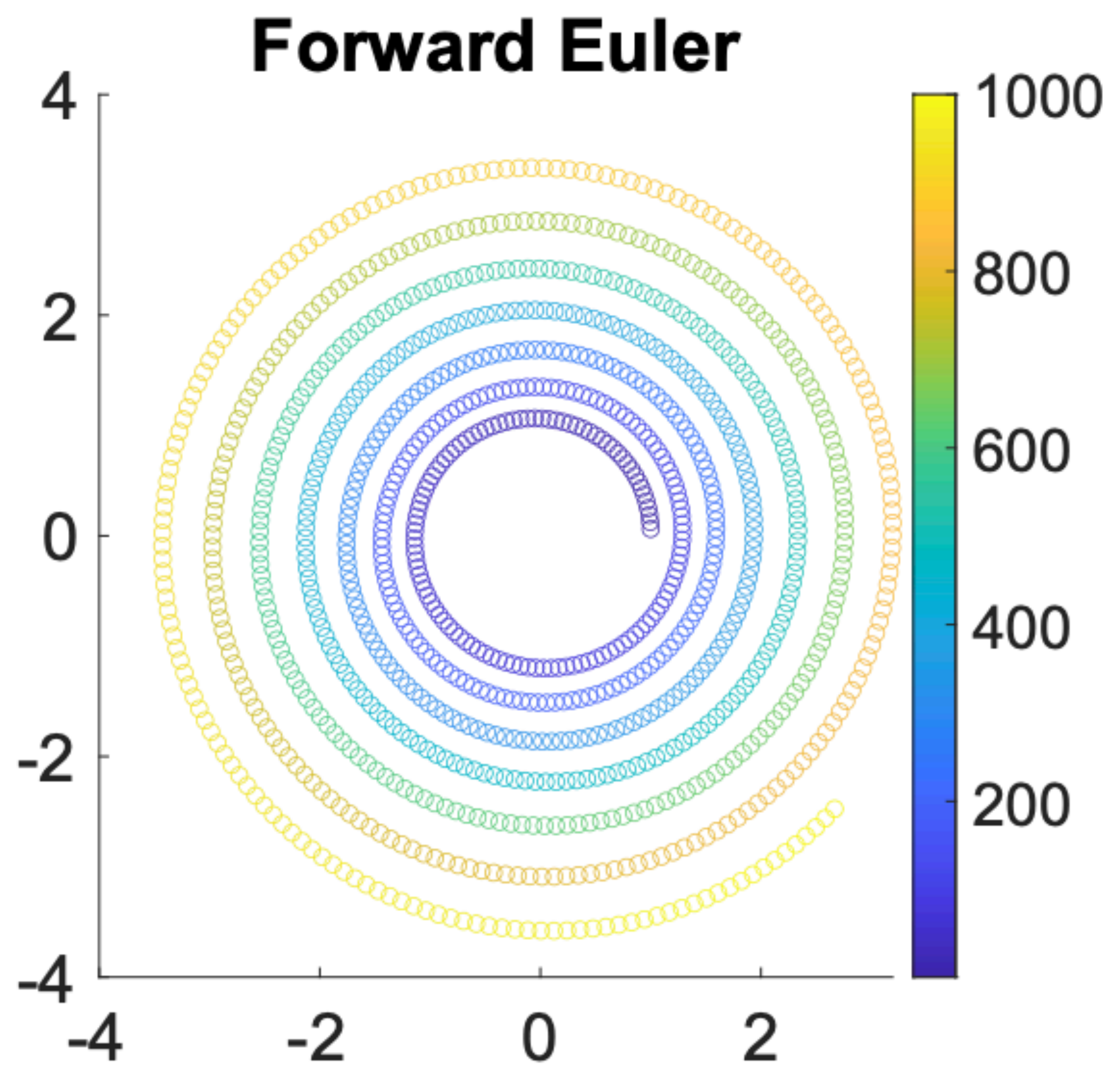


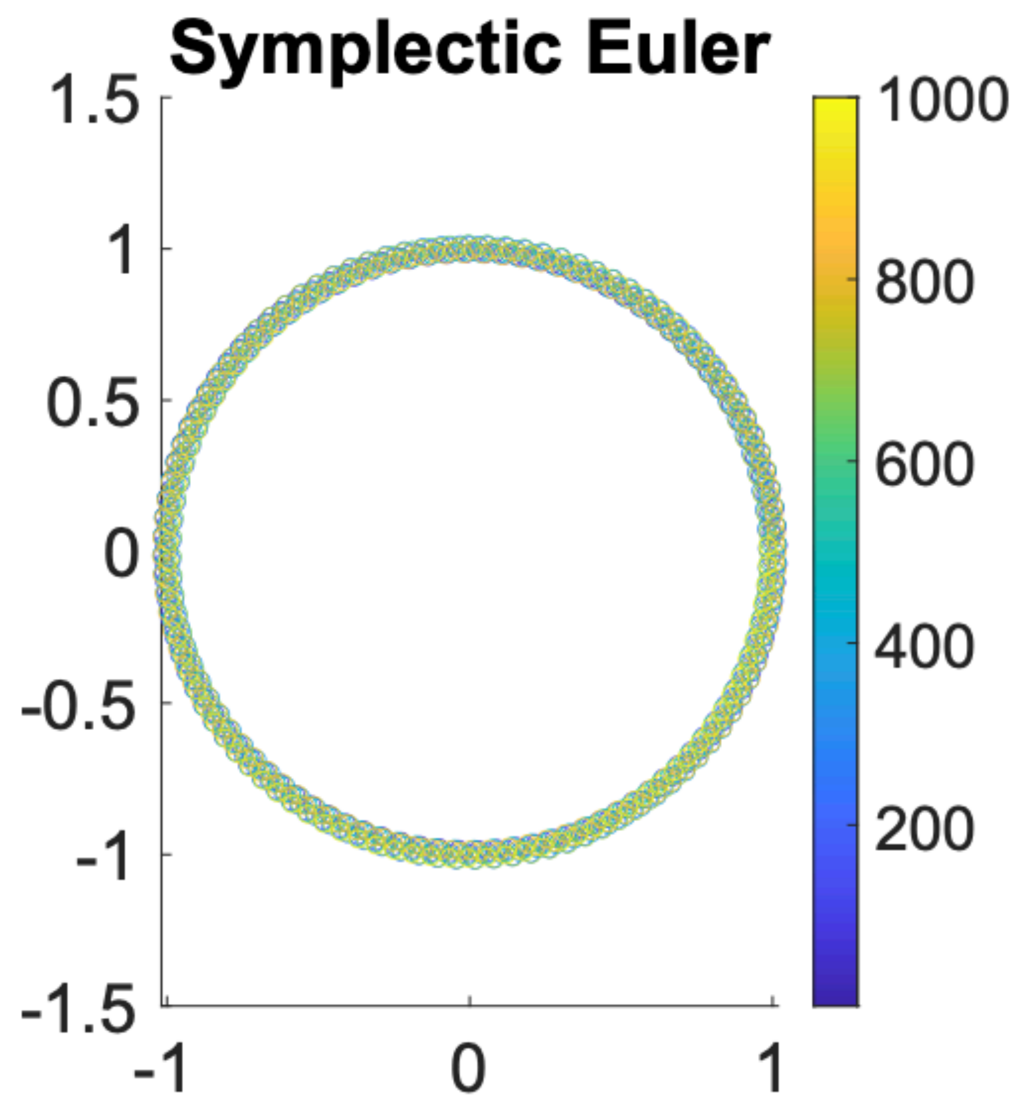
Problem Setup

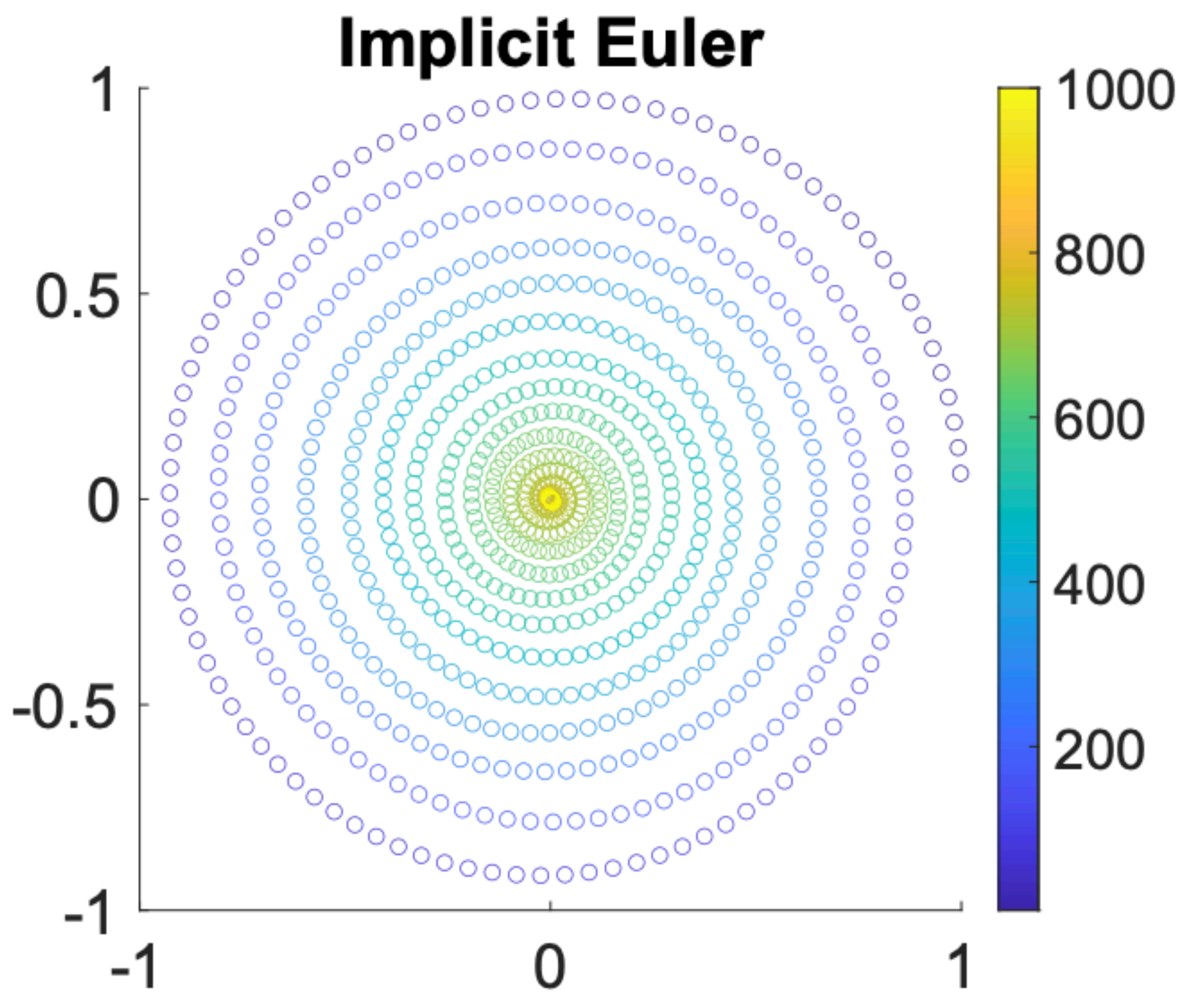
$$x^{n+1} = x^n + \Delta t v^n, \quad v^{n+1} = v^n + \Delta t M^{-1} f^n.$$

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

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







Stability of Forward, Symplectic, and Backward Euler

Analysis

More Time Integration Methods

- Backward Difference Formula (BDF)
  - Uses configuration from multiple steps (e.g. $x^n, v^n, x^{n-1}, v^{n-1} \rightarrow x^{n+1}, v^{n+1}$)
  - A Unified Newton Barrier Method for Multibody Dynamics [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]

Potential Course Project Idea:
Comparing efficiency, robustness, and accuracy among different time integration methods on elastodynamics simulation.

Newton’s Method for Backward Euler

Derivation

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

Optimization Time Integration

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

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

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

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

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

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

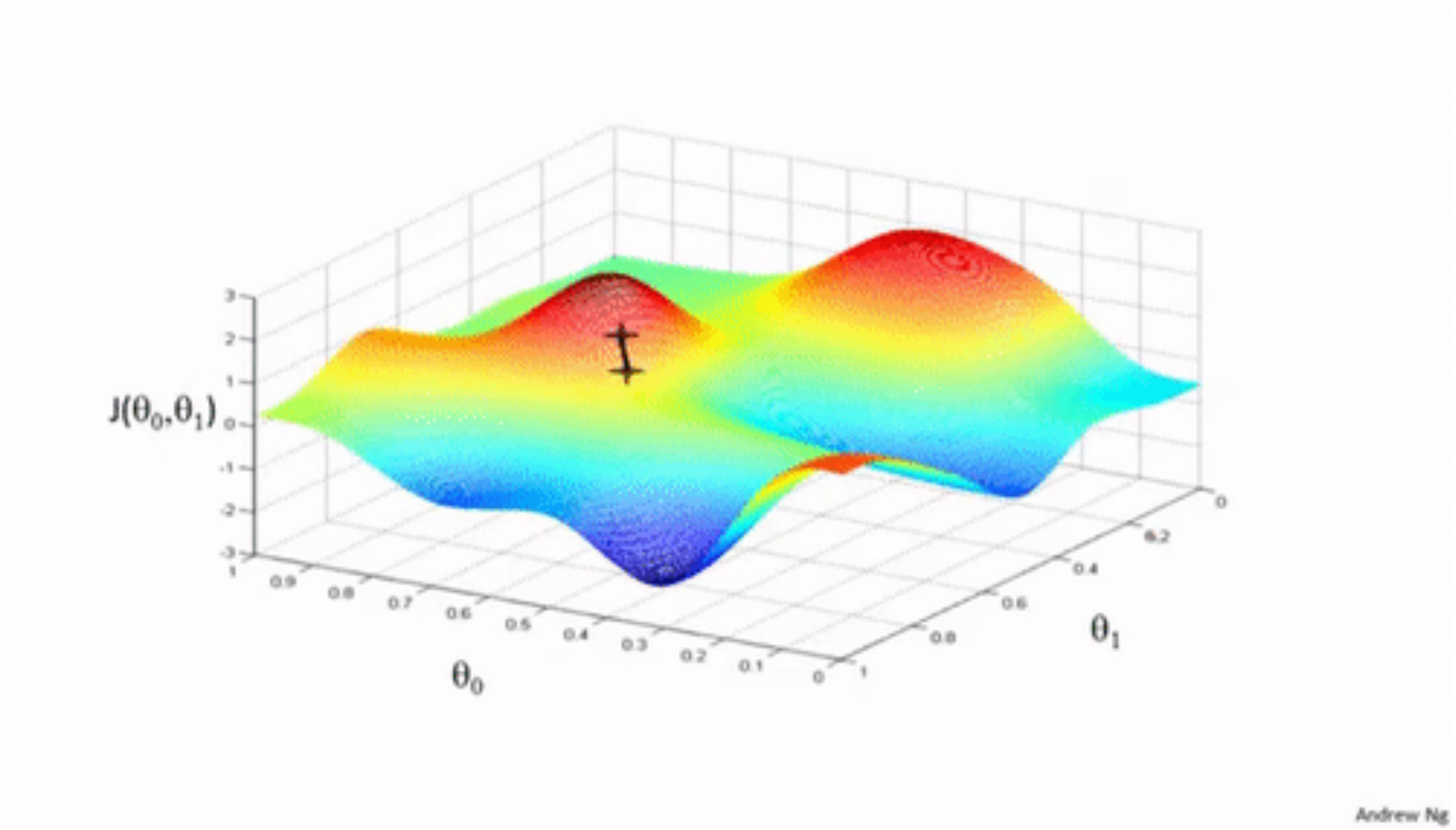
$$M(x^{n+1} - (x^n + \Delta t v^n)) - \Delta t^2 f(x^{n+1}) = 0.$$

Optimization Time Integration

Applying Newton’s Method

Optimization Time Integration

Applying Newton's Method, 2D Illustration



Global Convergence with Line Search

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**
  $\alpha \leftarrow \alpha / 2$;

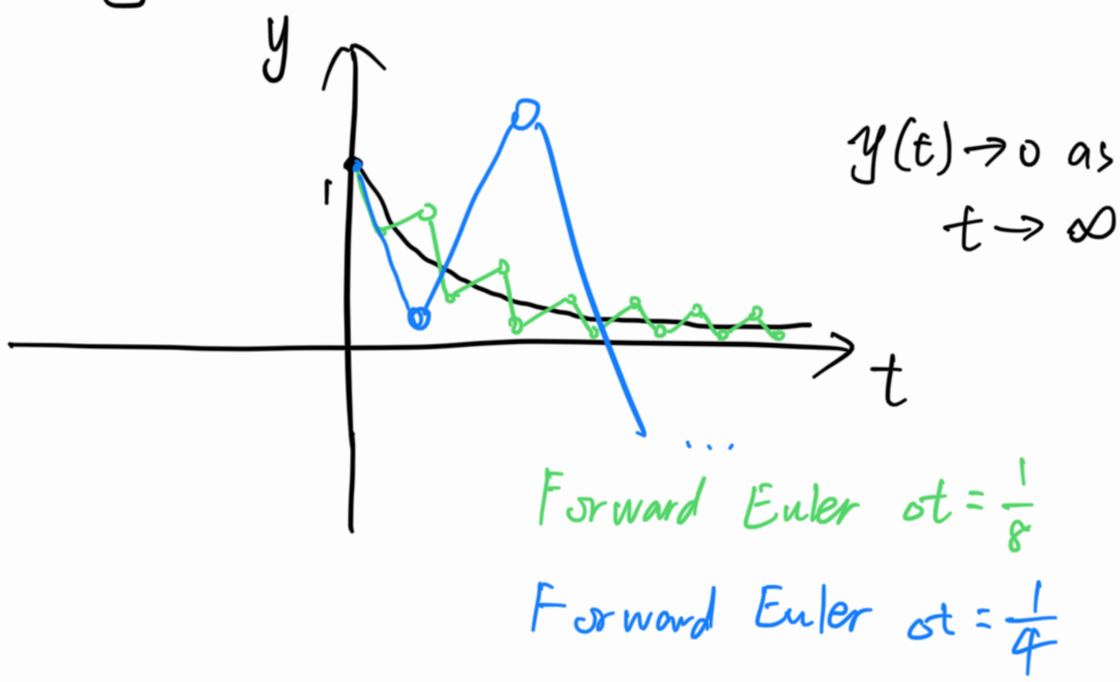
Descent Direction

Stability of Forward, Symplectic, and Backward Euler

Consider IVP

$$y'(t) = -15\,y(t), \quad t \geq 0, \quad y(0) = 1$$

Solution: $y(t) = e^{-15t}$



Forward Euler:

$$\frac{y^{n+1} - y^n}{\Delta t} = -a\,y^n \qquad (a > 0)$$

$$y^{n+1} = (1 - a\Delta t)\,y^n$$

$$y^{n+1} = (1 - a\Delta t)^n\,y^0$$

to remain stable:

$$|1 - a\Delta t| \leq 1 \implies 0 \leq \Delta t \leq \frac{2}{a}$$

Backward Euler

$$\frac{y^{n+1} - y^n}{\Delta t} = -a\,\Delta t\,y^{n+1}$$

or

$$y^{n+1} = \frac{y^n}{(1 + a\Delta t)}$$

For the Forward/Backward Euler time integration schemes,

they can seen as applying the [Forward/Backward Euler] on

$$\frac{dq}{dt} = g(q)$$

$$q = \begin{bmatrix} x \\ v \end{bmatrix}, \quad g(q) = \begin{bmatrix} v \\ M^{-1} f(x) \end{bmatrix}$$

But symplectic Euler is special

it is explicit (no systems to solve)

condionally stable (like Forward Euler),

but it's symplectic — preserves structures like system energy with small deviations (proper dt)

Newton's Method for Backward Euler

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

Newton's method:

initial guess x^0

in iteration i

$$f(x^{n+1}) \approx f(x^i) + \frac{df}{dx}(x^i)(x^{n+1} - x^i)$$

f(x) ≈ [illegible]

solve

$$M(x - (x^n + \Delta t v^n)) - \Delta t^2 \left(f(x^i) + \frac{df}{dt}(x^i)(x - x^i) \right) = 0$$

— linear system

for x

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

. . . .

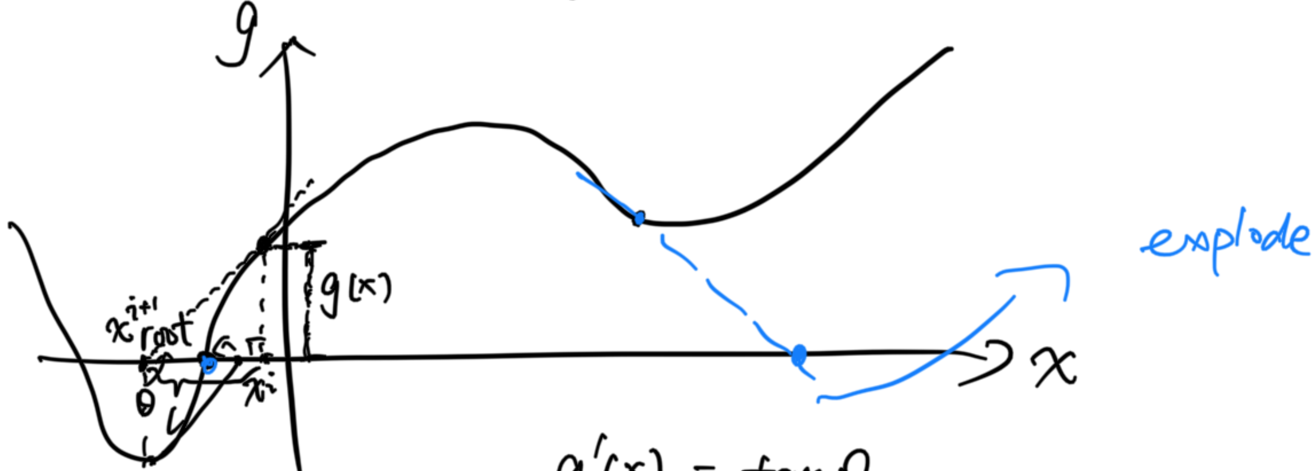
get x^{n+1}

Convergence Issue of Newton's Method

In 1D,

$$g(x) = 0$$

$$x^{i+1} = x^i - \frac{g(x)}{g'(x)}$$



$$g'(x) = \tan\theta$$

$$\underbrace{\tan\theta}_{g'(x)} = \frac{g(x)}{l} \Rightarrow l = \frac{g(x)}{g'(x)}$$

Applying Newton's Methon on optimization

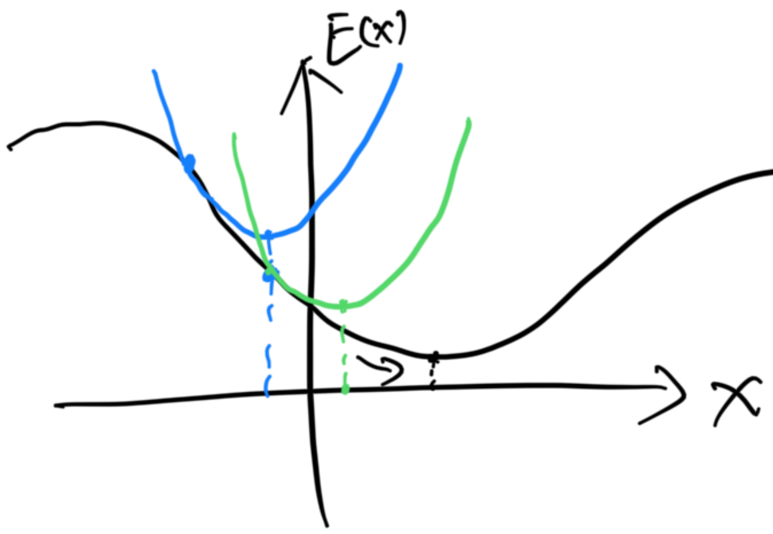
$$\min_x E(x)$$

Newton's method start x^0, in iteration i:

$$\min E(x^i) + \Delta x^T \nabla E(x^i) + \frac{1}{2} \Delta x^T \nabla^2 E(x^i) \Delta x$$

Δx

$$x^{i+1} = x^i + \Delta x$$



Descent Directions

For a smooth objective function $E(x)$, at x^i where $\nabla E(x^i) \neq 0$, there exists a descent direction p

Definition:

$$\exists \beta > 0 \quad \text{s.t.} \quad \forall \alpha \in [0, \beta]$$
$$E(x^i + \alpha p) \leq E(x^i)$$

or

$$p^T \nabla E(x^i) < 0$$

e.g. $-\nabla E(x^i)$

Newton's search direction:

$$p^i = -\left(\nabla^2 E(x^i)\right)^{-1} \nabla E(x^i)$$

is descent if $\nabla^2 E(x^i)$ is SPD

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

where P is SPD and P is close

$\nabla E(x^i)$