Lec 2: Time Integration
15-769: Physically-based Animation of Solids and Fluids (F23)
Recap on Shape Representations

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

**Focus of the lectures:**

- For fluids
  - Particle-In-Cell (PIC) Method and Fluid-Implicit-Particle (FLIP) Method
  - Collision Obstacles

- For solids
  - Finite Element Method (FEM)

(Adaptive) Spatial Discretization

- Eulerian solid simulation with deformations
- A material point method for solids
- Reformulating Hyperelastic Material
- Hybrid grains: Adaptive coupling
- Fast Corotated Elastic SPH Solids
- Surface-Only Dynamic Deformations
- Multi-Layer Thick Shells (2023)
- High-Order Incremental Potentials
- Adaptive Anisotropic Remeshing

Papers for all methods available on course website
Recap on Shape Representations

\[ x^n = \begin{bmatrix} x_{0x}^n \\ x_{0y}^n \\ x_{0z}^n \\ x_{1x}^n \\ x_{1y}^n \\ x_{1z}^n \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \]

\[ v^n = \begin{bmatrix} v_{0x}^n \\ v_{0y}^n \\ v_{0z}^n \\ v_{1x}^n \\ v_{1y}^n \\ v_{1z}^n \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \]
Newton’s 2nd Law

• The spatially discrete, temporally continuous form

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

• Mass matrix (for now)

\[
M = \begin{pmatrix}
m_1 & m_1 & m_2 \\
m_1 & m_2 \\
m_2 & m_2
\end{pmatrix}
\]
Time Stepping (Time Integration)

\[ x^n + \Delta t \frac{\partial x}{\partial t} = x^{n+1} \]

\[ v^n + \Delta t \frac{\partial v}{\partial t} = v^{n+1} \]

Time steps

\[ t^n, t^{n+1}, t^{n+2} \]

\[ x^n, x^{n+1}, x^{n+2} \]

\[ v^n, v^{n+1}, v^{n+2} \]
Newton’s 2nd Law (Temporally Discrete)
Forward Difference, Forward Euler

- Forward difference approximation on velocity and acceleration

\[
\left( \frac{dx}{dt} \right)^n \approx \frac{x^{n+1} - x^n}{\Delta t} \quad \left( \frac{dv}{dt} \right)^n \approx \frac{v^{n+1} - v^n}{\Delta t}
\]

\[
(f(t^n + \Delta t) = f(t^n) + \frac{df}{dt}(t^n)\Delta t + O(\Delta t^2))
\]

Taylor’s expansion

\[
\begin{align*}
\frac{x^{n+1} - x^n}{\Delta t} &= v^n, \\
M \frac{v^{n+1} - v^n}{\Delta t} &= f^n. \\
x^{n+1} &= x^n + \Delta tv^n, \\
v^{n+1} &= v^n + \Delta tM^{-1}f^n.
\end{align*}
\]
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

\[ x^{n+1} = x^n + \Delta t v^n, \]
\[ v^{n+1} = v^n + \Delta tM^{-1} f^n. \]

\[ x^{n+1} = x^n + \Delta t v^{n+1}, \]
\[ v^{n+1} = v^n + \Delta tM^{-1} f^{n+1}. \]

\[ x^{n+1} = x^n + \Delta t v^{n+1}, \]
\[ v^{n+1} = v^n + \Delta tM^{-1} f^{n+1}. \]
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: $x^{n+1}, v^{n+1}$

1. $x^i \leftarrow x^n;
2. \textbf{while } \|M(x^i - (x^n + \Delta tv^n)) - \Delta t^2 f(x^i)\| > \epsilon \textbf{ do}
3. \quad \text{solve } M(x - (x^n + \Delta tv^n)) - \Delta t^2 (f(x^i) + \nabla f(x^i)(x - x^i)) = 0 \textbf{ for } x;
4. \quad x^i \leftarrow x;
5. \quad x^{n+1} \leftarrow x^i;
6. v^{n+1} \leftarrow (x^{n+1} - x^n)/\Delta t;
Convergence Issue of Newton’s Method

Over-shooting
Optimization Time Integration

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

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

\[ \tilde{x}^n = x^n + \Delta tv^n \]

\[ \frac{1}{2} \| x - \tilde{x}^n \|_M^2 = \frac{1}{2} (x - \tilde{x}^n)^T M (x - \tilde{x}^n) \]

\[ \frac{\partial P}{\partial x}(x) = -f(x) \]

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

\[ M (x^{n+1} - (x^n + \Delta tv^n)) - \Delta t^2 f(x^{n+1}) = 0. \]
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:** $x^{n+1}, v^{n+1}$

1. $x^i \leftarrow x^n$
2. do
3. \[ P \leftarrow \text{SPDProjection}(\nabla^2 E(x^i)); \]
4. \[ p \leftarrow -P^{-1} \nabla E(x^i); \]
5. $\alpha \leftarrow \text{BackTrackingLineSearch}(x^i, p);$ \hspace{1cm} \[ x^i \leftarrow x^i + \alpha p; \]
6. while $\|p\|_\infty/h > \epsilon;$
7. \[ x^{n+1} \leftarrow x^i; \]
8. \[ v^{n+1} \leftarrow (x^{n+1} - x^n)/\Delta t; \]

**Algorithm 2:** Backtracking Line Search

**Result:** $\alpha$

1. $\alpha \leftarrow 1$
2. while $E(x^i + \alpha p) > E(x^i)$ do
3. \[ \alpha \leftarrow \alpha/2; \]
Descent Direction
Stability of Forward, Symplectic, and Backward Euler

Consider IVP
\[ y'(t) = -15y(t), \quad t \geq 0, \quad y(0) = 1 \]

Solution: \[ y(t) = e^{-15t} \]

Forward Euler:
\[ y^{n+1} - y^n \over \Delta t = -\alpha y^n \quad (\alpha > 0) \]
\[ y^{n+1} = (1 - \alpha \Delta t) y^n \]
\[ y^{n+1} = (1 - \alpha \Delta t)^n y^0 \]
To remain stable:
\[ |1 - \alpha \Delta t| < 1 \quad \Rightarrow \quad 0 \leq \Delta t \leq \frac{2}{\alpha} \]

Backward Euler
\[ y^{n+1} - y^n \over \Delta t = -\alpha \Delta t y^{n+1} \]
\[ y^{n+1} = \frac{y^n}{1 + \alpha t} \]

For the Forward/Backward Euler time integration schemes,
they can be seen as applying the 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)
conditionally stable (like Forward Euler),
but it's symplectic — presence structures
like system energy with small deviations
\[
\text{Newton's Method for Backward Euler}
\]
\[
M(x^{n+1} - (x^n + st v^n)) - st^2 f(x^{n+1}) = 0
\]

Newton's method:
initial guess \( x^0 \)
in iteration \( i \)
\[
\begin{align*}
(x^{n+1}) &= \left[ f(x_i) + \frac{df}{dx}(x_i)(x_i - x_i) \right] \\
\end{align*}
\]
\[ M(x - (x^n + \alpha_t v^n)) - \alpha_t^2 \left( f(x^i) + \frac{df}{dt}(x^i)(x - x^i) \right) = 0 \]

for \( x \)

\[ x^{i+1} \leftarrow x \]

get \( x^{n+1} \)

**Convergence Issue of Newton's Method**

In 1D,

\[ g(x) = 0 \]

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

\[ \tan \theta = \frac{g(x)}{g'(x)} \implies l = \frac{g(x)}{g'(x)} \]

**Applying Newton's Method on optimization**

\[ \min_x E(x) \]

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

\[ \min_x E(x^i) + \alpha x^T \nabla E(x^i) + \frac{1}{2} \alpha^2 \nabla^2 E(x^i) x \]
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:

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

or

$$P^T \nabla E(x^i) < 0 \quad \text{e.g. } -\nabla E(x^i)$$

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 \( \nabla E(x^i) \)