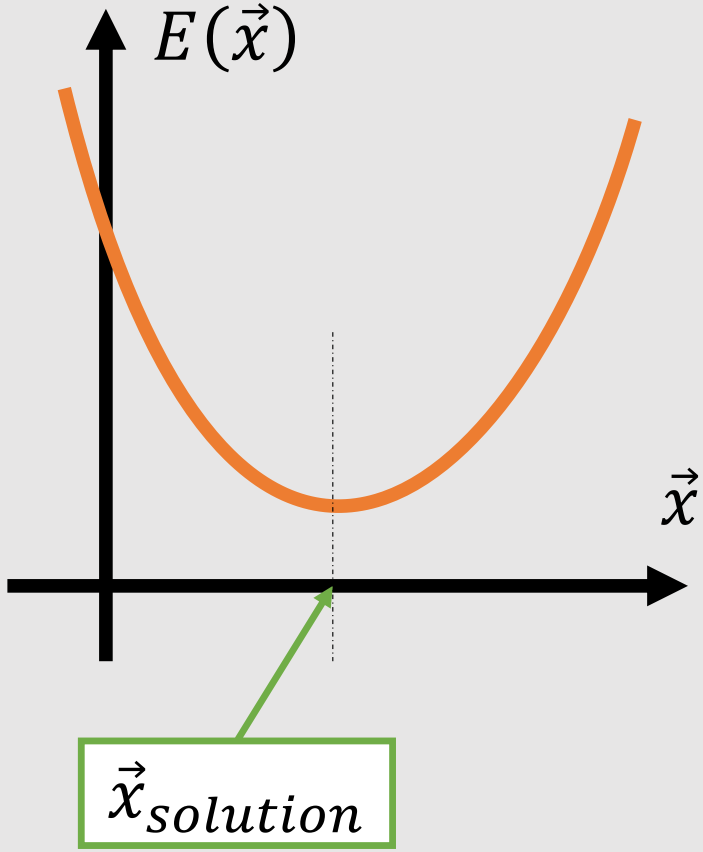
Variational Backward Euler Time Integration

What is Variational Method?

- Solution is expressed by the optimization

$$\vec{X}_{solution} = \underset{\vec{x}}{\operatorname{argmin}} E(\vec{x})$$

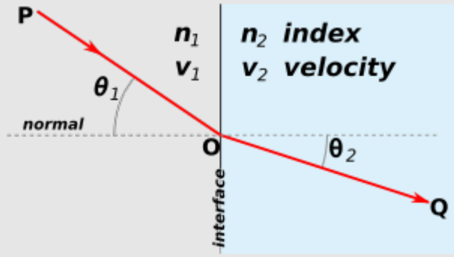


Variational Principles in Physics

- Mechanics

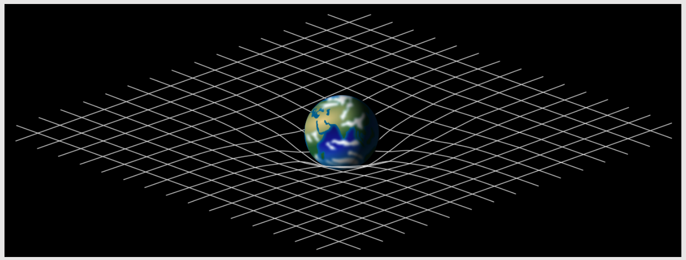
(Wikipedia)

- Optics



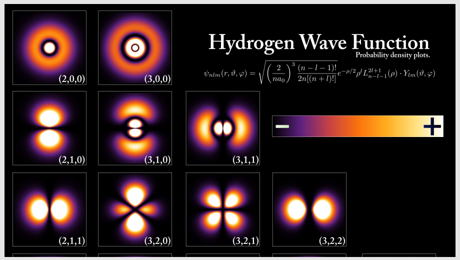
(Wikipedia)

- General relativity



(Wikipedia)

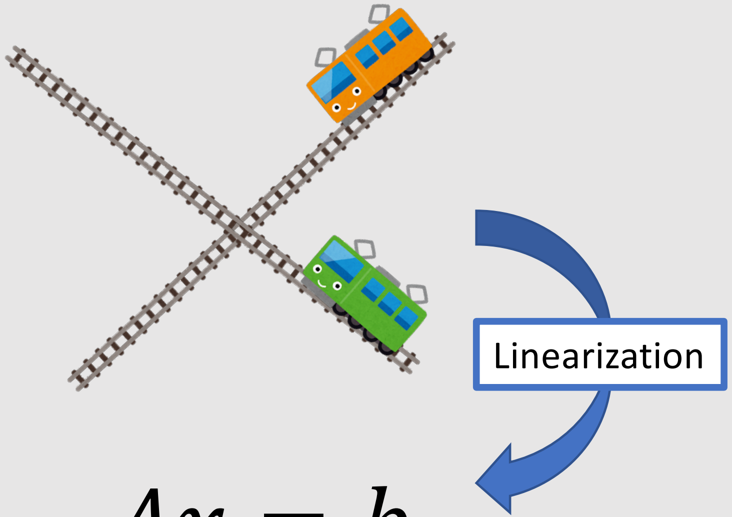
- Quantum physics



(Wikipedia)

Solving Constraints v.s. Variational Problem

Solution should be on this line

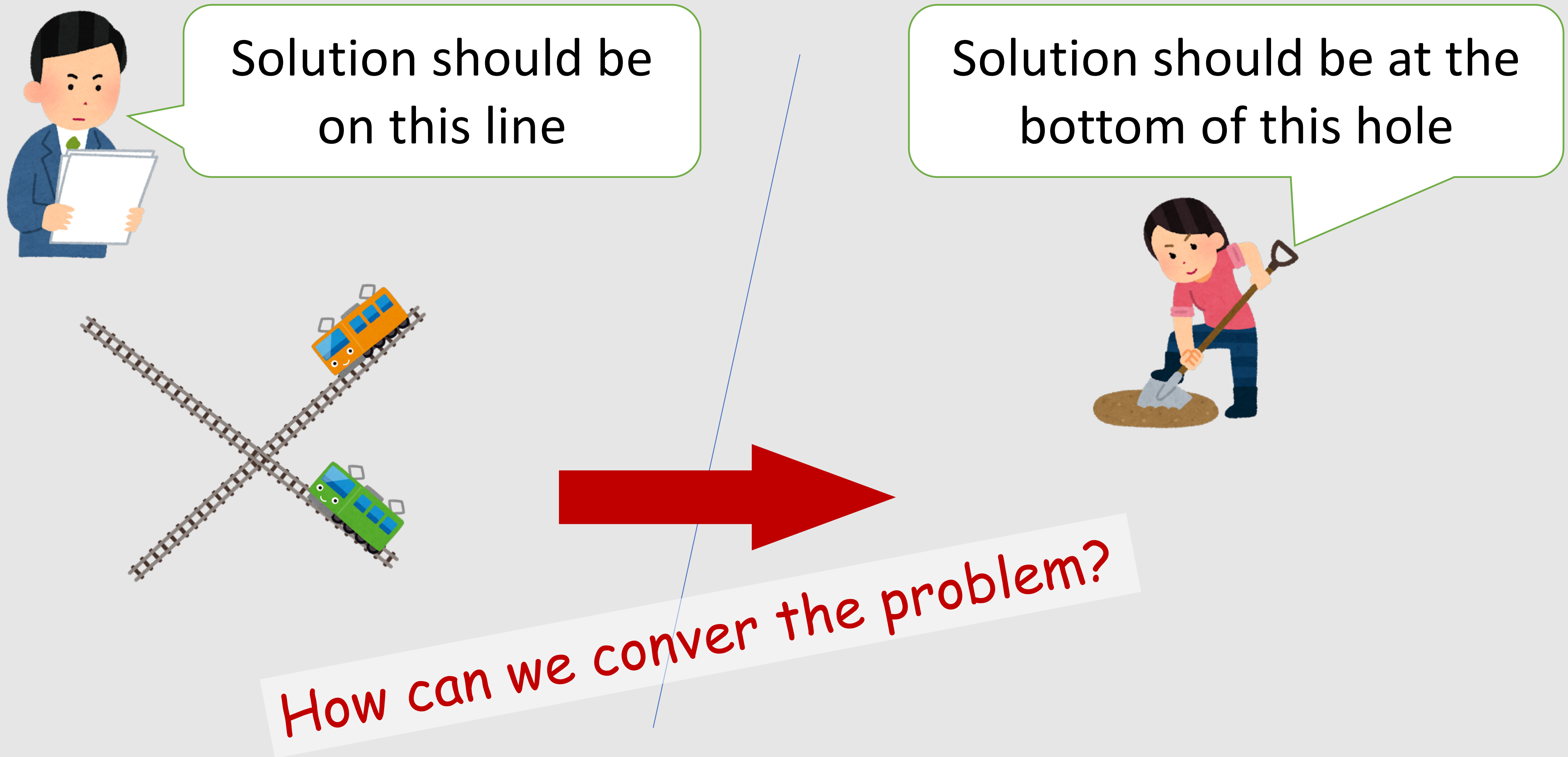


$$Ax = b$$

Solution should be at the bottom of this hole

There are many weapons to fight

Solving Constraints v.s. Variational Problem



Making a Variational Problem

- We only need a single scalar value E to find solution

$$A\vec{x} = \vec{b}$$

L2 norm of residual:

$$E(\vec{x}) = \left\|A\vec{x} - \vec{b}\right\|^2$$

integration:

$$E(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} - \vec{b}^T\vec{x}$$

Making a Variational Problem

- Integration with $\vec{x}$ will make a variational formula

$$\frac{\partial W(\vec{x})}{\partial \vec{x}} = \vec{b} \quad \xrightarrow{\text{integration}} \quad E(\vec{x}) = W(\vec{x}) - \vec{b}^T \vec{x}$$

$$\frac{\partial W(\vec{x})}{\partial \vec{x}} = -M\vec{x} \quad \xrightarrow{\text{integration}} \quad E(\vec{x}) = W(\vec{x}) + \frac{1}{2} \vec{x}^T M \vec{x}$$

Variational Formulation of Backward Euler

- Review of Backward Euler

$$\frac{ds}{dt} = \frac{s_{i+1} - s_i}{dt} = F(s_{i+1})$$

plug in $s_i = \begin{pmatrix} \vec{v}_\mathrm{i} \\ \vec{x}_\mathrm{i} \end{pmatrix}$, $M\dot{\vec{v}} = \frac{\partial W}{\partial \vec{x}}$

$$\begin{cases} \vec{x}_{i+1} = \vec{x}_i + dt \cdot \vec{v}_i + dt^2 \cdot M^{-1} \dfrac{\partial W}{\partial x_{i+1}} \\ \vec{v}_{i+1} = (\vec{x}_{i+1} - \vec{x}_i)/dt \end{cases}$$

Variational Formulation of Backward Euler

- Getting next time step by minimization

$$\vec{x}_{i+1} = \underset{\vec{x}}{\operatorname{argmin}}\, E_i(\vec{x})$$

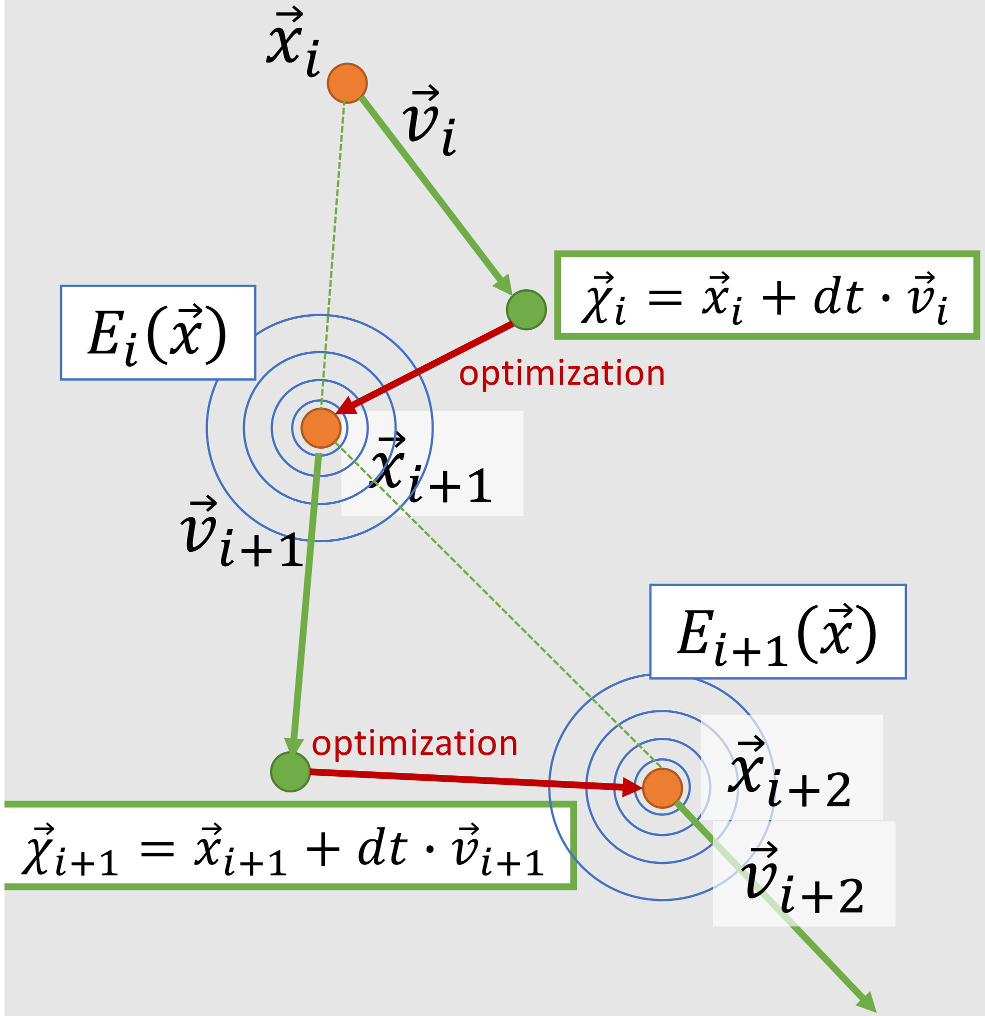
$$E_i(\vec{x}) = \mathrm{W}(\vec{x}) + \frac{1}{2dt^2}(\vec{x} - \vec{\chi}_i)^T M(\vec{x} - \vec{\chi}_i$$

$$\vec{\chi}_i = \vec{x}_i + dt \cdot \vec{v}_i$$

integration

$$\begin{cases} \vec{x}_{i+1} = \vec{x}_i + dt \cdot \vec{v}_i + dt^2 \cdot M^{-1} \dfrac{\partial W}{\partial x_{i+1}} \\ \vec{v}_{i+1} = (\vec{x}_{i+1} - \vec{x}_i)/dt \end{cases}$$

Scheme of Variational Backward Euler



1. compute temporary position

$$\vec{\chi}_i = \vec{x}_i + dt \cdot \vec{v}_i$$

2. optimize $E_i(\vec{x})$ to get $\vec{x}_{i+1}$

3. Set velocity

$$\vec{v}_{i+1} = \frac{(\vec{x}_{i+1} - \vec{x}_i)}{dt}$$

4. Goto 1

Variational Formula Explained

- Solving tradeoff between elasticity & inertia

$$\vec{\chi}_i = \vec{x}_i + dt \cdot \vec{v}_i$$

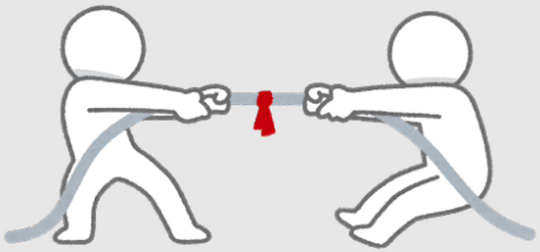
$$E_i(\vec{x}) = \underbrace{\mathrm{W}(\vec{x})}_{\text{elasticity}} + \underbrace{\frac{1}{2dt^2}(\vec{x} - \vec{\chi}_i)^T M(\vec{x} - \vec{\chi}_i)}_{\text{inertia}}$$

elasticity

inertia

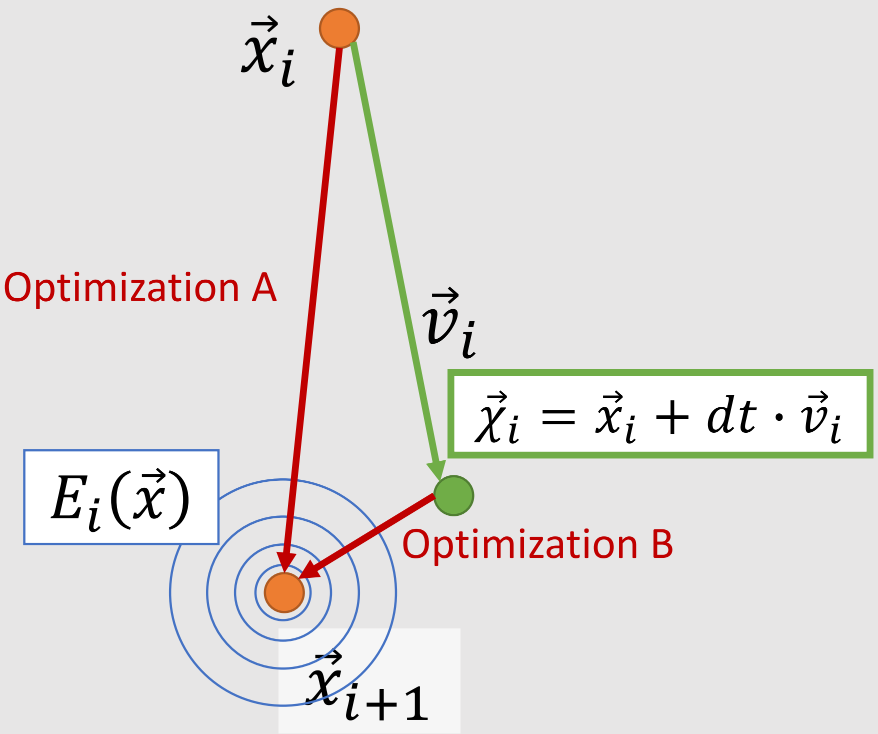
Trying to “undeform” shape

Trying to move shape with velocity $\vec{v}_i$



Optimization with Newton Method

- optimize $E_i(\vec{x}) = \mathrm{W}(\vec{x}) + 1/2dt^2\,(\vec{x} - \vec{\chi}_i)^T M(\vec{x} - \vec{\chi}_i)$ to get $\vec{x}_{i+1}$



Optimization A (bad☹)

$$\vec{x}_{i+1} = \vec{x}_i - \left[\frac{\partial^2 W(\vec{x}_i)}{\partial^2 \vec{x}}\right]^{-1} \left(\frac{\partial W(\vec{x}_i)}{\partial \vec{x}}\right)$$

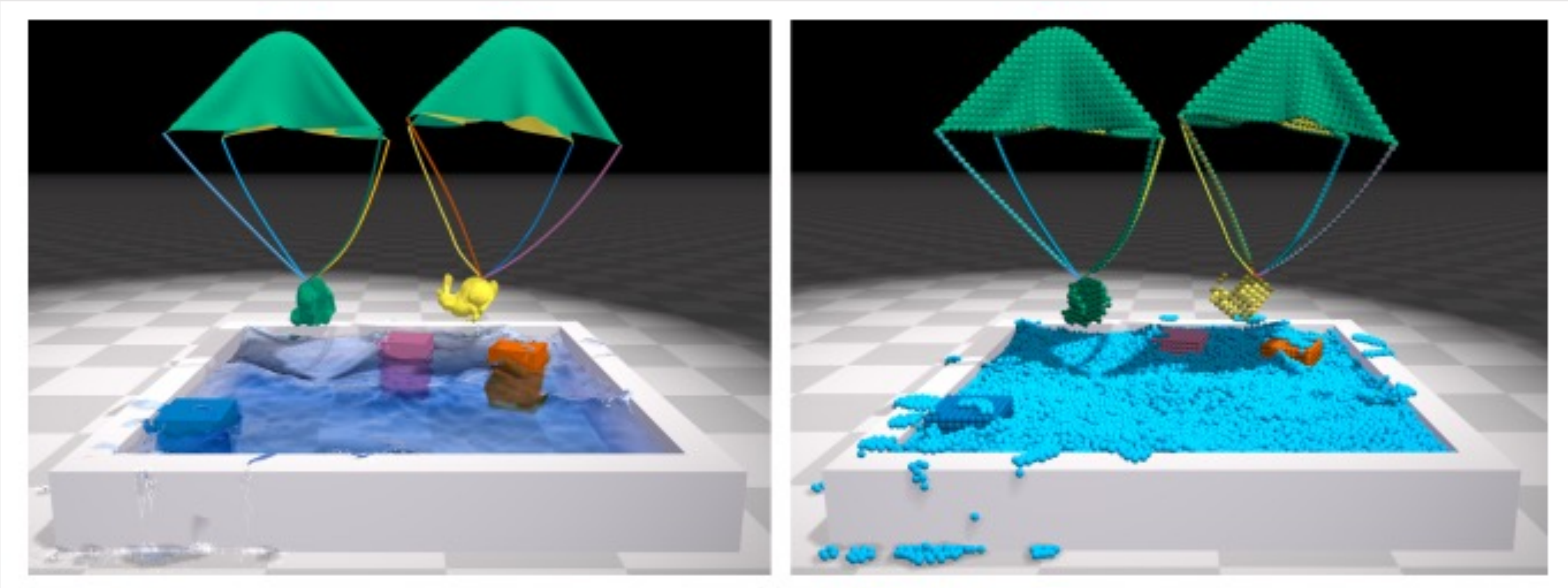
Optimization B (good ☺)

$$\vec{x}_{i+1} = \vec{\chi}_i - \left[\frac{\partial^2 W(\vec{\chi}_i)}{\partial^2 \vec{x}}\right]^{-1} \left(\frac{\partial W(\vec{\chi}_i)}{\partial \vec{x}}\right)$$

Position-based Dynamics

Position-based Dynamics (PBD) [Müller et al.,2006]

Employed in many real-time game engine



[Macklin et al. 14, "Unified Particle Physics for Real-Time Applications]

Variational Backward Euler and PBD

$$E(\vec{x}) = \mathrm{W}(\vec{x}) + \frac{1}{2dt^2}(\vec{x} - \vec{\chi})^T M(\vec{x} - \vec{\chi})$$

Energy is based on the constraint

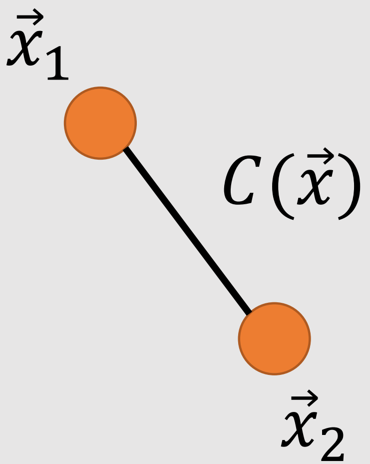
Point mass approximation

$$\mathrm{W}(\vec{x}) = \sum_{j \in constraints} C_j^2(\vec{x})$$

$$\sum_{k \in points} (\vec{x}_k - \vec{\chi}_k)^T m_k (\vec{x}_k - \vec{\chi}_k)$$

Variational Backward Euler and PBD

- Energy minimization by finding root of constraints: $C(\vec{x}) = 0$



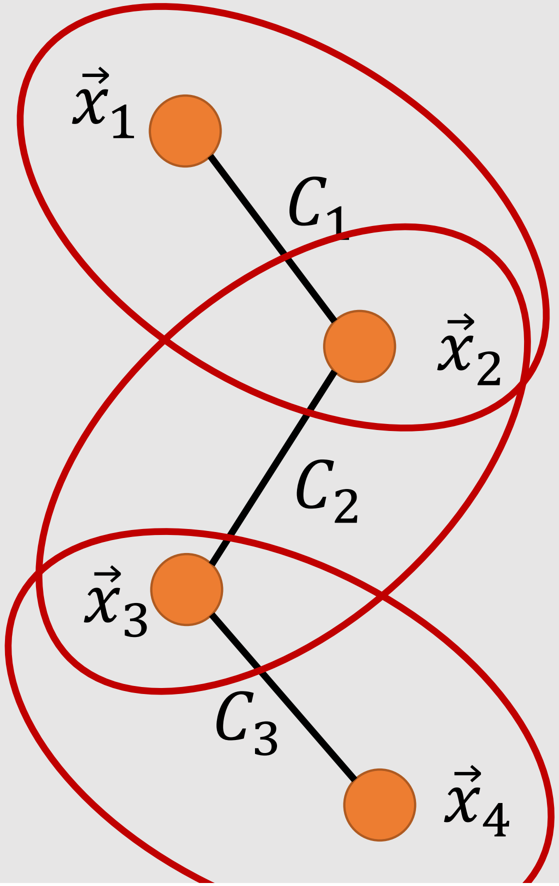
Update while preserving momentum

$$M\Delta\vec{x} = -\alpha\nabla\left\{C_j^2(\vec{x})\right\}$$

Chose α s.t. $C(\vec{x} + \Delta\vec{x}) = 0$

$$\Delta\vec{x} = \frac{-C_j(\vec{x})\nabla C_j(\vec{x})}{\nabla C_j^T(\vec{x})M^{-1}\nabla C_j(\vec{x})}$$

Solving Constraints with Gauss-Seidel



1. Satisfy C_1 by changing $\vec{x}_1, \vec{x}_2$
2. Satisfy C_2 by changing $\vec{p}_2, \vec{p}_3$
3. Satisfy C_3 by changing $\vec{p}_3, \vec{p}_4$

Comparison: PBD & Newton Method

$$E_i(\vec{x}) = \underbrace{W(\vec{x})}_{\text{elasticity}} + \underbrace{\frac{1}{2dt^2}(\vec{x}-\vec{\chi}_i)^T M(\vec{x}-\vec{\chi}_i)}_{\text{inertia}}$$

elasticity << ***inertia***

- The matrix is easy
- PBD solves optimization well

elasticity >> ***inertia***

- The matrix is difficult
  (large condition number, stiff equation)
- PBD cannot optimize

References

- Müller, Matthias, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. "**Position based dynamics.**" Journal of Visual Communication and Image Representation 2, no. 18 (2007): 109-118.

- Jan Bender, Matthias Müller and Miles Macklin, **A Survey on Position Based Dynamics, 2017**, In *Tutorial Proceedings of Eurographics*, 2017