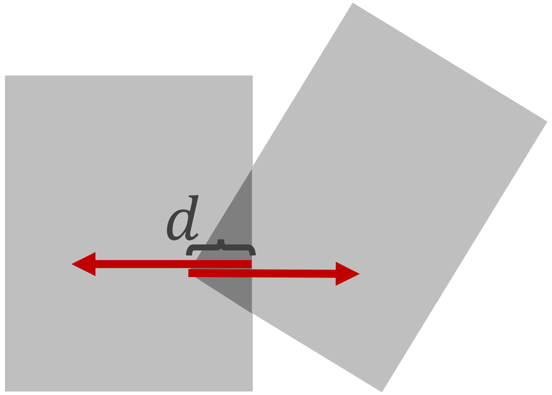
XPBD
Extended Position Based Dynamics

Matthias Müller, Ten Minute Physics

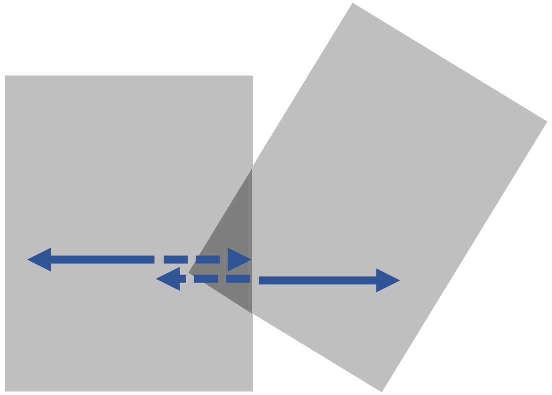
matthiasmueller.info/tenMinutePhysics

Motivation

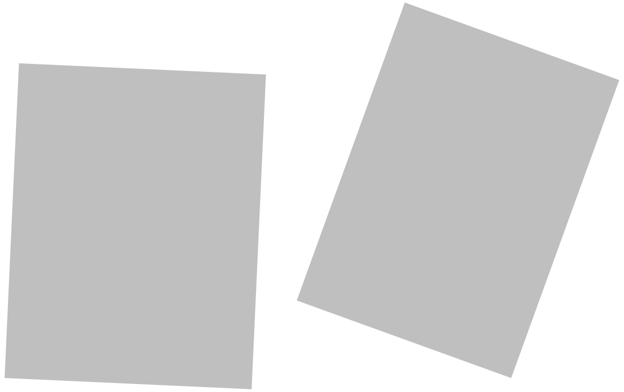
Force Based Simulation



penetration causes force $f = kd$



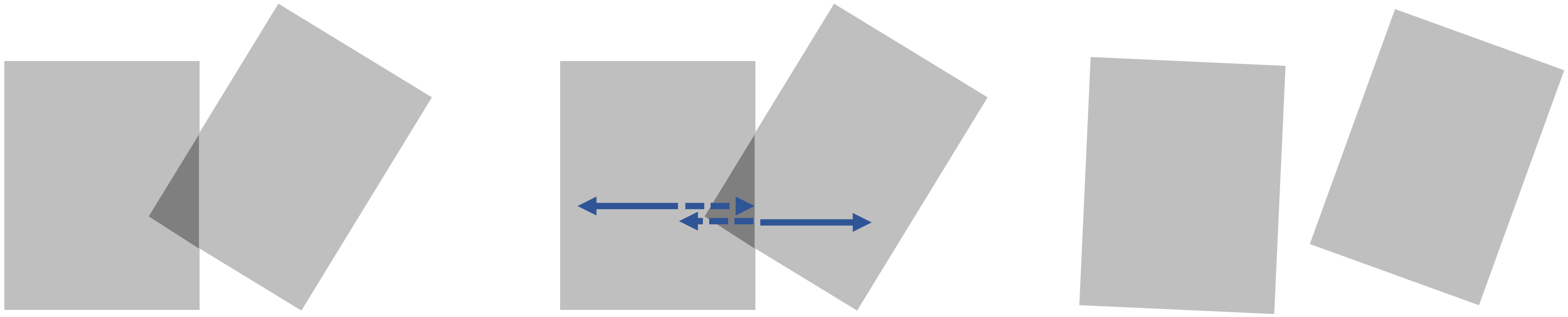
forces change velocities



velocities change positions

- Overlap needed
- Reaction lag
- Large stiffness k → stability problems, overshooting
- Small stiffness → squishy system
- How to choose k for a hard constraint?

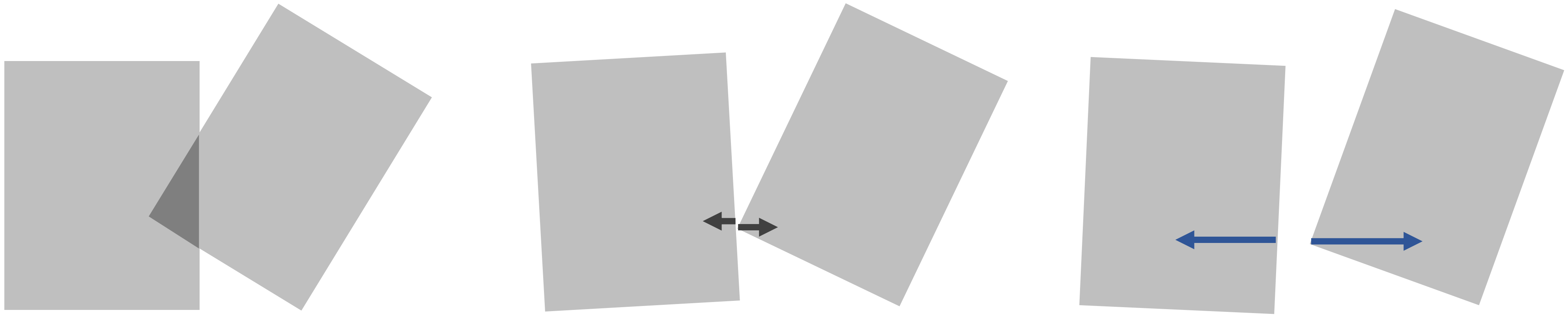
Impulse Based Simulation



penetration detection only | apply impulse to get separating velocities | velocities change positions

- More stable
- Controlled velocity update, no overshooting
- Drift: consistent velocities do not guarantee consistent positions!

Position Based Simulation



penetration detection only | change positions to resolve penetrations | update velocities

- Controlled position change, unconditionally stable
- No drift

Physical? Accurate?

- We found that PBD is very closely related to implicit Euler integration

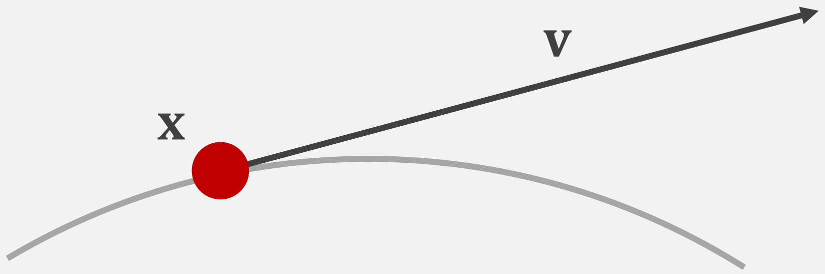
- To be precise, it corresponds to…

… the first iteration of the Newton minimization of a backward Euler integration step in variational position-based form, using the non-linear Gauss-Seidel method, where the Newton solution is initialized with the unconstrained / predicted / inertial position using external forces.

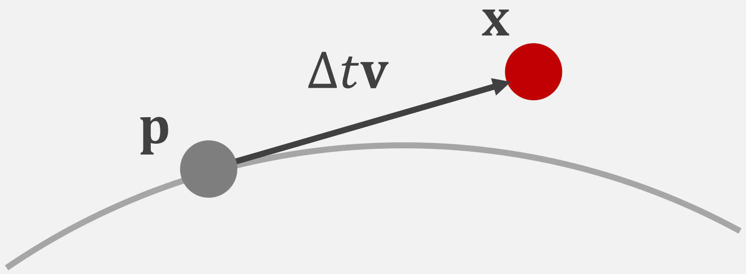
- Original PBD is unphysical only in the way it handles softness
- Problem fixed with XPBD

Bead on a Wire

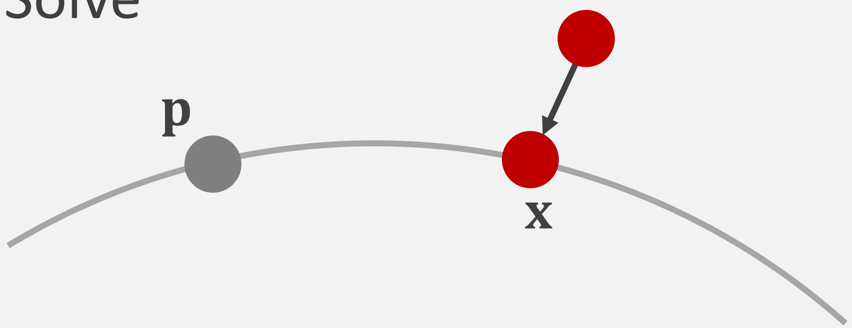
State



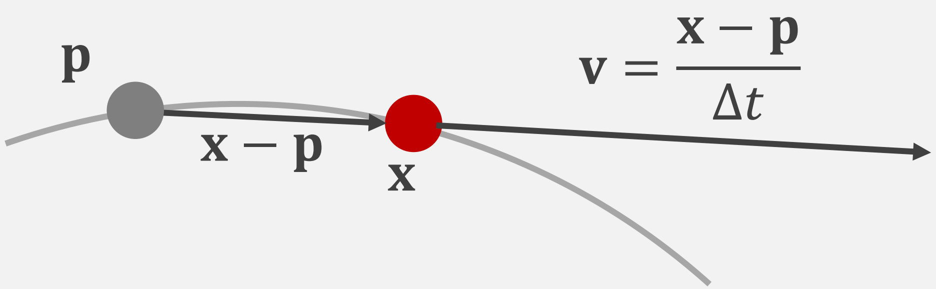
Integration



Solve



Velocity update



PBD = integrator and solver!

PBD Algorithm

while simulating
    for all particles i
        $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\ \mathbf{g}$
        $\mathbf{p}_i \leftarrow \mathbf{x}_i$
        $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\ \mathbf{v}_i$

    for all constraints C
        solve($C, \Delta t$)

    for all particles i
        $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t$

solve($C, \Delta t$):

for all particles i of C
    compute $\Delta\mathbf{x}_i$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta\mathbf{x}_i$

Iterations vs. Sub-steps

while simulating
    for all particles i
        $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\ \mathbf{g}$
        $\mathbf{p}_i \leftarrow \mathbf{x}_i$
        $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\ \mathbf{v}_i$
    for n iterations
        for all constraints C
            solve$(C, \Delta t)$
    for all particles i
        $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t$

$\Delta t_s \leftarrow \Delta t/n$
while simulating
    for n substeps
        for all particles i
            $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t_s \mathbf{g}$
            $\mathbf{p}_i \leftarrow \mathbf{x}_i$
            $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t_s \mathbf{v}_i$
        for all constraints C
            solve$(C, \Delta t_s)$
        for all particles i
            $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t_s$

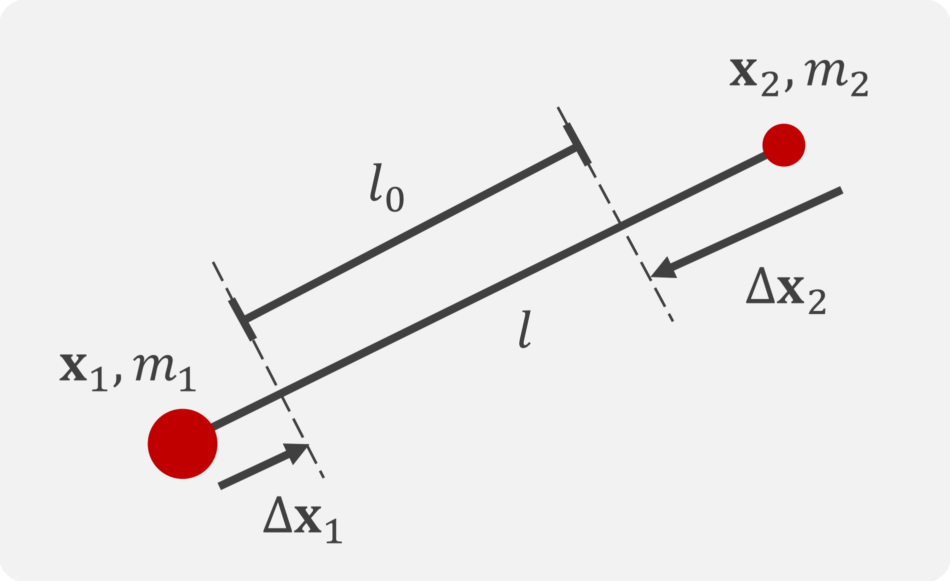
- Spending fixed time budget with sub-steps is much is more effective than with iterations!
- XPBD much simpler (no λ tracking)

Distance Constraint

- Rest distance l_0
- Current distance l
- Masses m_i
- Inverse masses $w_i = 1/m_1$

$$\Delta\mathbf{x}_1 = \frac{w_1}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

$$\Delta\mathbf{x}_2 = -\frac{w_2}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$



General Constraint

$\mathbf{x}_1$

$\mathbf{x}_2$

…

$\mathbf{x}_n$

particles participating in constraint

$C(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \longrightarrow C$

constraint function

constraint error (number)

For the distance constraint:

$$C_{\text{dist}}(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_2 - \mathbf{x}_1| - l_0$$

Constraint Gradient ∇C

$\nabla C(\mathbf{x})$ is a vector

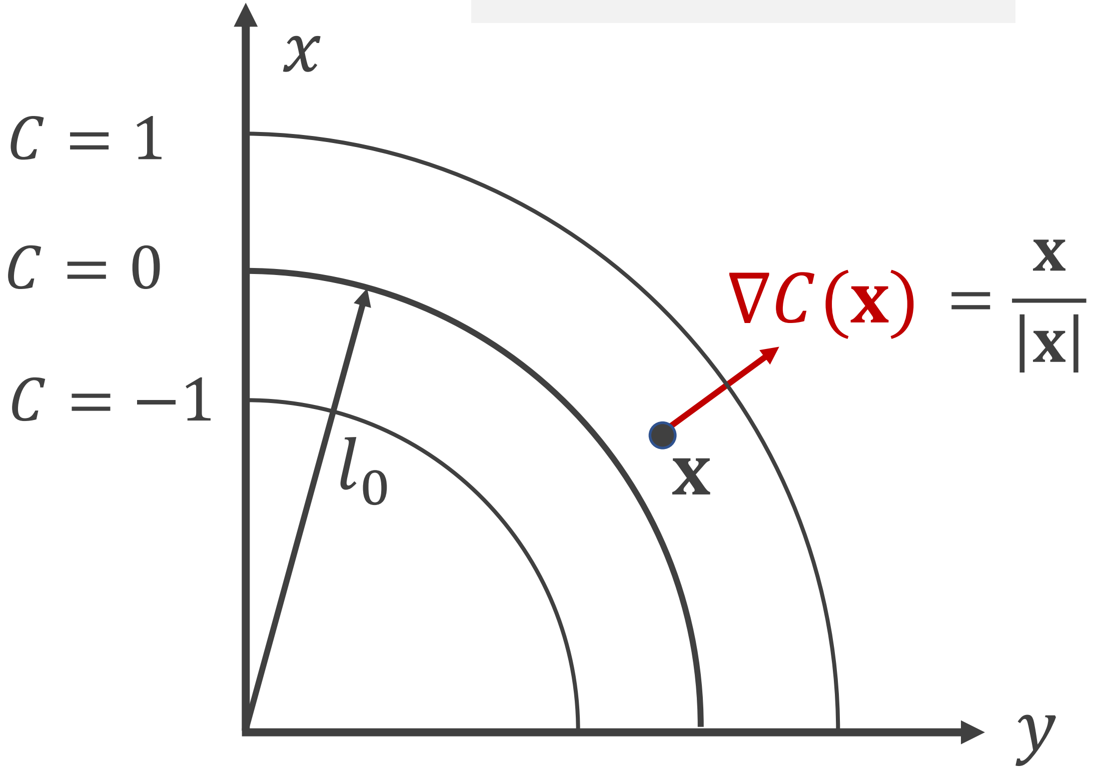
Vector direction:

- direction in which C increases the most

Vector length:

- how much C changes when moving $\mathbf{x}$ by one unit

$$C_{\text{dist}} = |\mathbf{x}| - l_0$$



Solving a General Constraint (PBD)

Compute the scalar value λ (same for all participating particles):

$$\lambda = \frac{-\mathrm{C}}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots + w_n|\nabla C_n|^2}$$

∇C_i: How to move $\mathbf{x}_i$ for a maximal increase of C

$|\nabla C_i|^2$ the squared length of ∇C_i

Compute correction for point $\mathbf{x}_i$ as: $\Delta\mathbf{x}_i = \lambda w_i \nabla C_i$

Making the Constraint Soft

PBD:

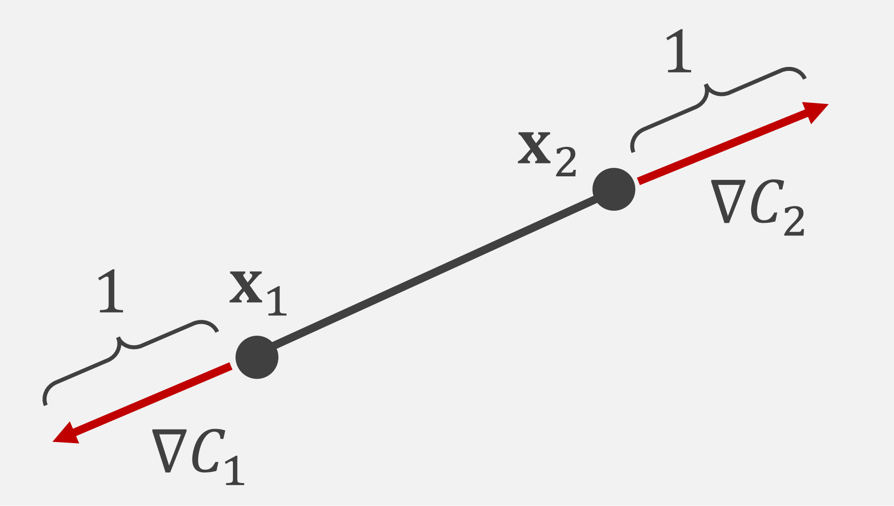
- Scale the correction as $\Delta\mathbf{x}_i = k\,\lambda w_i\,\nabla C_i$
- Stiffness $k \in [0,1]$
- Easy to tune!
- Dependent on time step (stiffer for smaller time steps)

XPBD:

$$\lambda = \frac{-C}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots w_n|\nabla C_n|^2 + \frac{\alpha}{\Delta t^2}}$$

- Compliance α is the inverse of physical stiffness
- Infinitely stiff (hard) when $\alpha = 0$

Example: Distance Constraint



$$\nabla C_1 = \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}$$

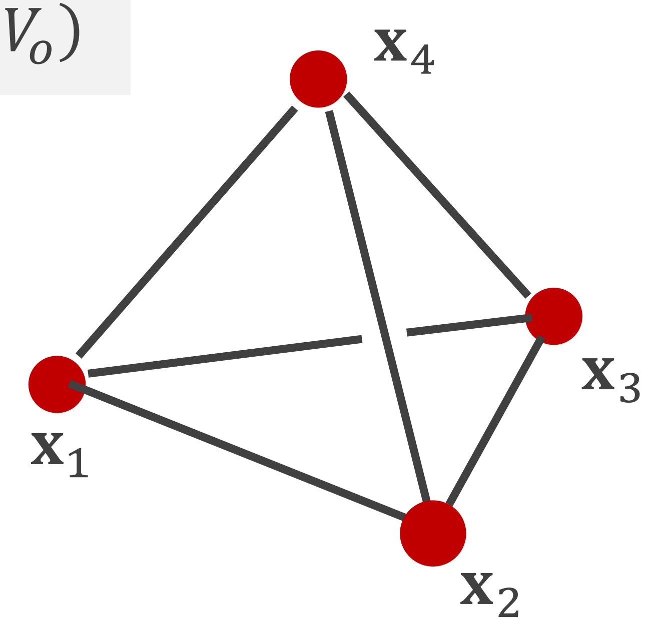
$$\nabla C_2 = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

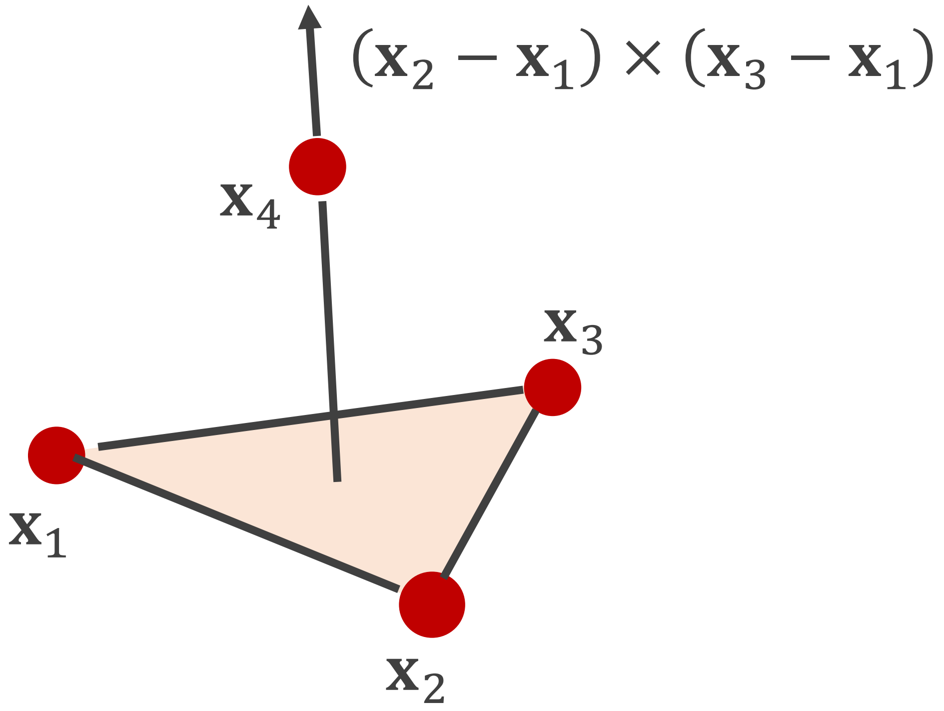
$$\lambda = \frac{-\mathrm{C}}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots + w_n|\nabla C_n|^2} = \frac{-(l - l_0)}{w_1 \cdot 1 + w_2 \cdot 1}$$

$$\Delta \mathbf{x}_1 = \lambda\, w_1\, \nabla C_1 = -\frac{w_1}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

Volume Conservation Constraint

$$C = 6(V - V_o)$$

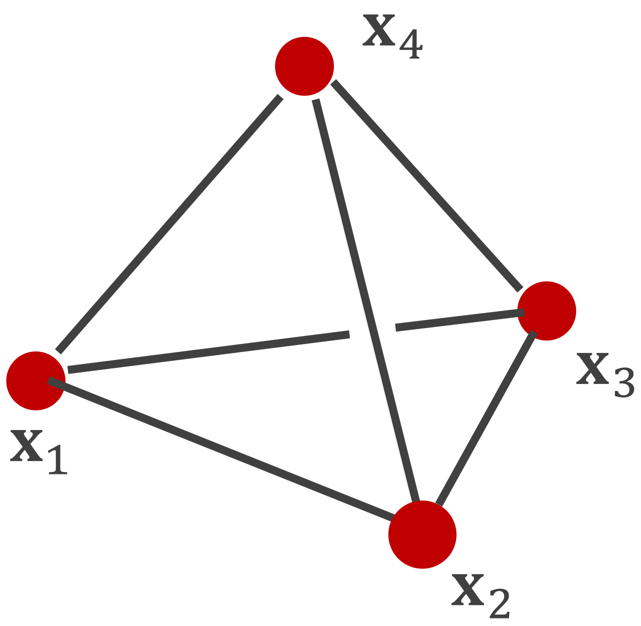




$$C = 6(V - V_o) = [(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)] \cdot (\mathbf{x}_4 - \mathbf{x}_1) - 6V_o$$

$$\nabla_4 C = (\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)$$

Solve



Right hand rule

$$\nabla_1 C = (\mathbf{x}_4 - \mathbf{x}_2) \times (\mathbf{x}_3 - \mathbf{x}_2)$$

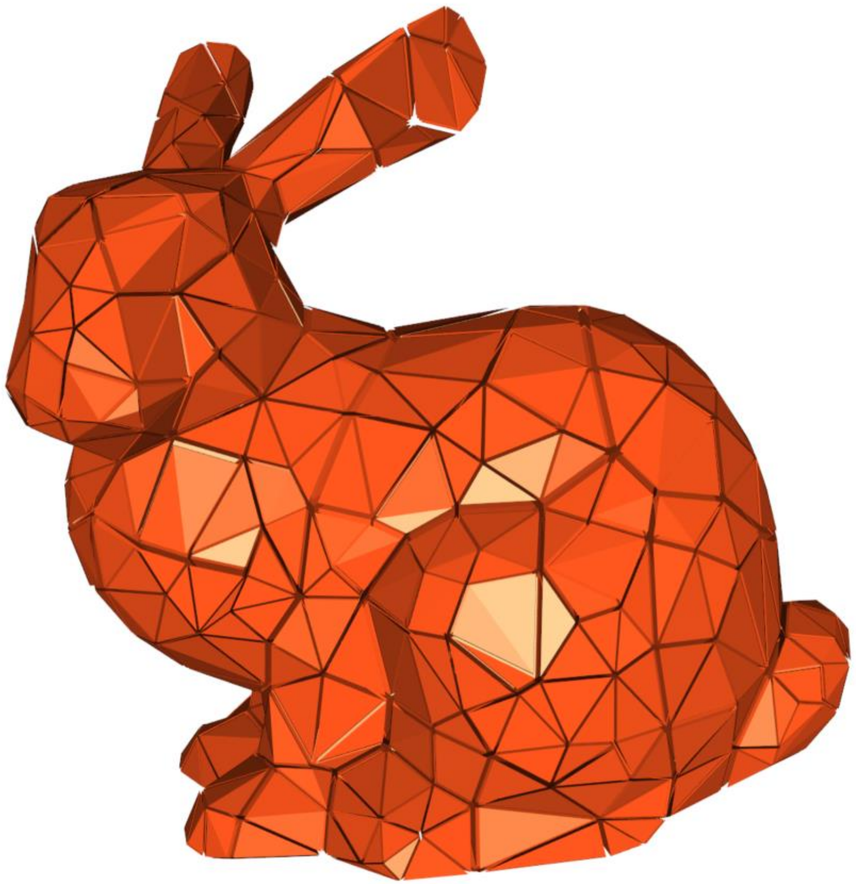
$$\nabla_2 C = (\mathbf{x}_3 - \mathbf{x}_1) \times (\mathbf{x}_4 - \mathbf{x}_1)$$

$$\nabla_3 C = (\mathbf{x}_4 - \mathbf{x}_1) \times (\mathbf{x}_2 - \mathbf{x}_1)$$

$$\nabla_4 C = (\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)$$

$$\lambda = \frac{-6(V - V_o)}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + w_3|\nabla C_3|^2 + w_4|\nabla C_4|^2}$$

$$\Delta \mathbf{x}_i = \lambda\, w_i\, \nabla C_i$$



→ See soft body tutorial for action!