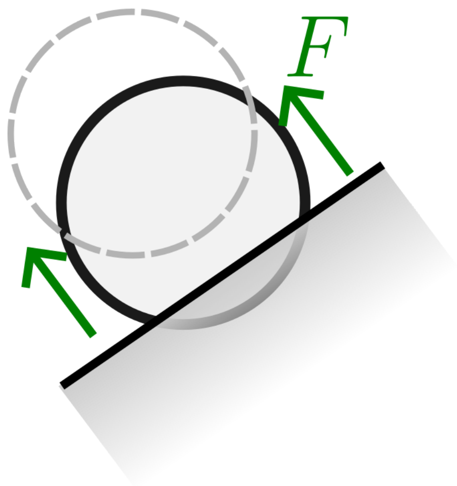
Position Based Dynamics

Constraint handling

Force based



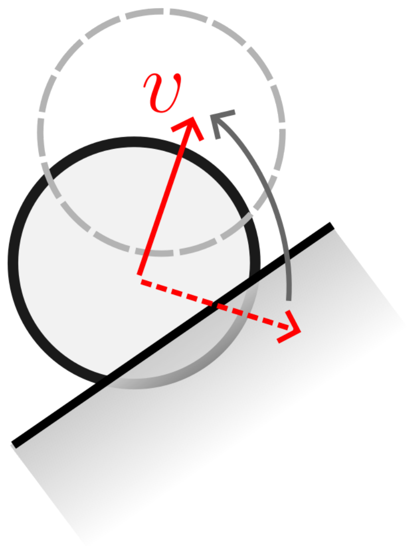
$$v_i^{k+1} = v_i^k + F_i/m_i\,\Delta t$$

(+) Physical interpretation

(+) Arbitrary constraint

(-) Constraint stiffness: Divergence

Impulse based



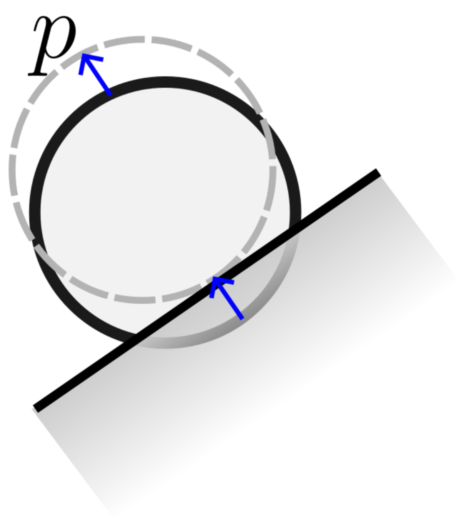
$$v_i^{k+1} = v_i^k + J/m$$

(+) Stability

(-) Collision only

(-) Doesn't correct existing collisions

Position based



$$p_i^{k+1} \to \mathcal{F}(p^k)$$

(+) Stability

(+) Arbitrary constraint

(-) Physical interpretation

PPD: General Idea

- Handle positional constraints by moving vertex positions directly
  - Velocity computed as change of position
  - (+) Very simple, unconditionally stable
    - Formalized in 2005
    - *[Matthias Muler, Bruno Heidelberger, Marcus Hennix, John Racliff. Position based Dynamics. PRIPHYS 2005.]*

PPD simulation step

1) Integrate *standard forces* to predict position

$$p_{\mathrm{prev}} = p$$
$$v = v + \Delta t\, F/m$$
$$p = p + \Delta t\, v$$

2) Project p on constraints

3) Update velocity

$$v = (p - p_{\mathrm{prev}})/\Delta t$$

PPD: Generic Non Linear Constraints

Handling arbitrary non constraint $C(p) > 0$ over $p = (p_1, \cdots, p_N)$

Solve iteratively $C(p + \mathrm{d}p) \simeq C(p) + \nabla C \cdot \mathrm{d}p > 0$ using Lagrange multiplier

$\mathrm{d}p = \lambda\, M^{-1}\, (\nabla C)^T, M = \mathrm{diag}(m_1, \cdots, m_N)$:masses

$$\Delta p_i = -\lambda / m_i\, \nabla_{p_i} C(p)^T$$

$$\Rightarrow \lambda = \frac{C(p)}{\sum_i 1/m_i\, \|\nabla_{p_i} C(p)\|^2}$$

Note: Converges to infinite stiffness

Proposition of eXtended PDF (XPBD)

$$\Rightarrow \lambda = \frac{C(p)}{\sum_i 1/m_i\, \|\nabla_{p_i} C(p)\|^2 + \frac{\alpha}{(\Delta t)^2}}$$

$\alpha = 1/K$: compliance (inverse stiffness)

[Miles Macklin, Matthias Muller, Nuttapong Chentanez. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. MIG 2016]

[Jan Bender, Matthias Muller, Miles Macklin. A survey on Position Based Dynamics. EUROGRAPHICS STAR 2017]

Note: Formulation may vary a bit depending on the paper.

Rigid bodies dynamic: Shape Matching

General Idea:

1) Consider a reference shape.

2) At time t, deform vertices individually

   $t \to t + \Delta t$

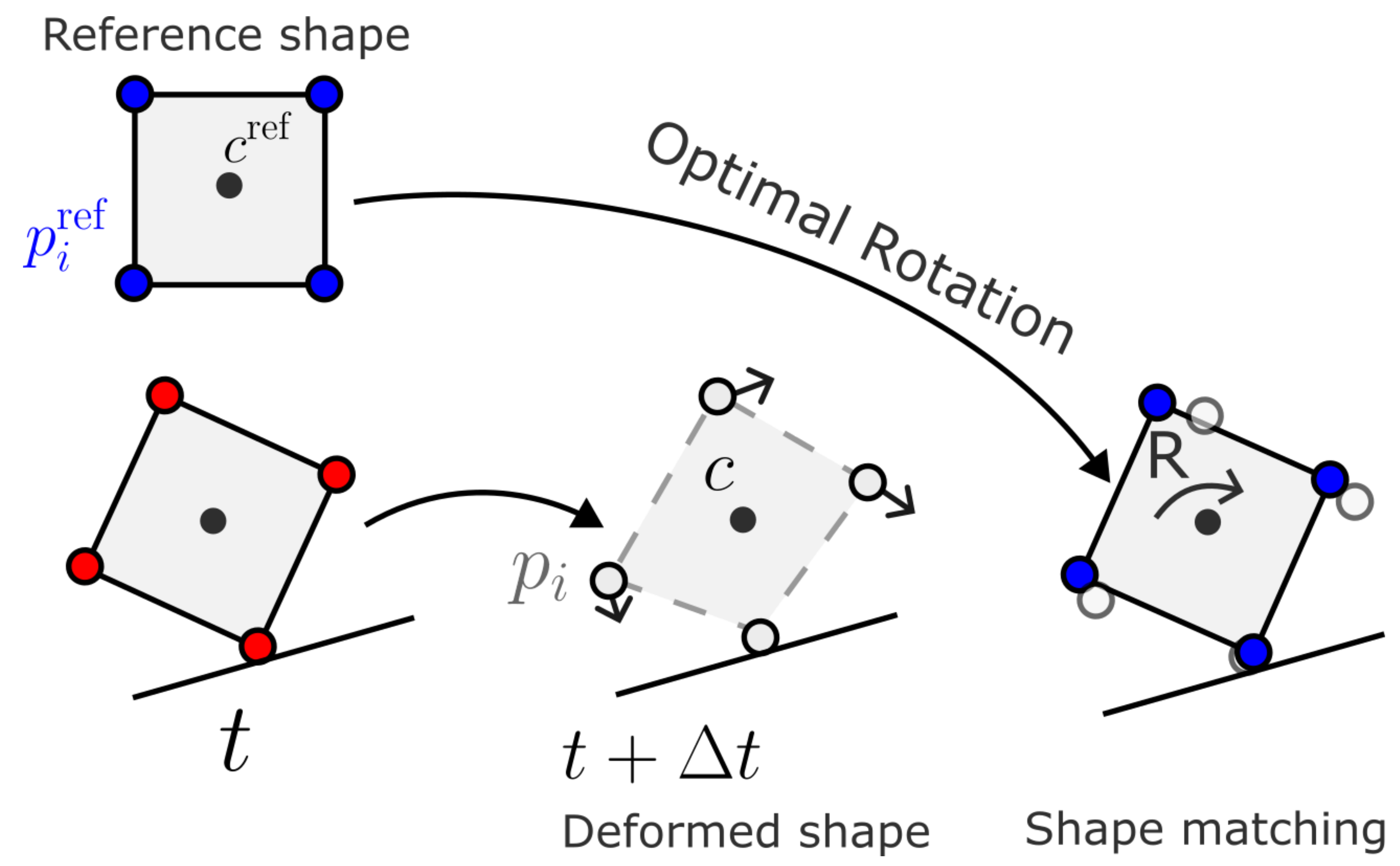
   Solve for the collision constraints on each vertex

3) Shape Matching:

   Compute optimal rotation R

   between reference shape and current one.

   Apply R on the reference shape (+ translate it).



(+) Garantee to preserve reference shape appearance

(-) May not preserve individual constraints

Shape Matching Formulation

Reference Shape:

Vertex position p_i^{ref}

Center of mass c^{ref}

Local vertex vector $r_i^{\text{ref}} = p_i^{\text{ref}} - c^{\text{ref}}$

Deformed shape at $t + \Delta t$

Vertex position p_i

Center of mass c

Local vertex vector $r_i = p_i - c$

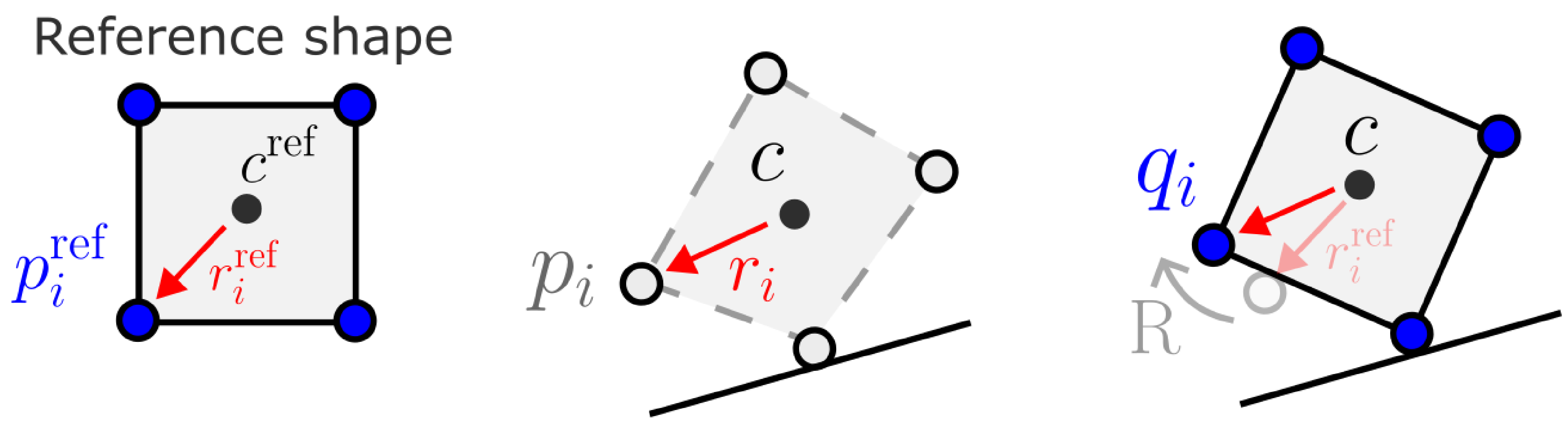
Averaged transformation $\mathrm{T} = \sum_i r_i \, (r_i^{\text{ref}})^T$

T minimizes $\sum_i \|\mathrm{T}\, r_i^{\text{ref}} - r_i\|^2$

Extract optimal rotation R

via polar decomposition from T

New position $q_i = \mathrm{R}\, r_i^{\text{ref}} + c$

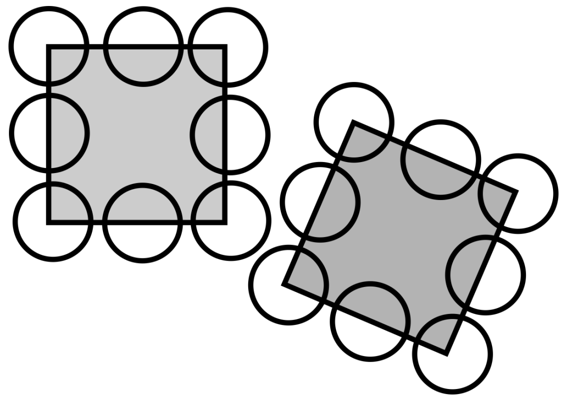


Reminder Polar Decomposition: $\mathrm{R} = W V^T$, for $svd(T) = W\,\Sigma\,V^T$

Shape Matching Usage

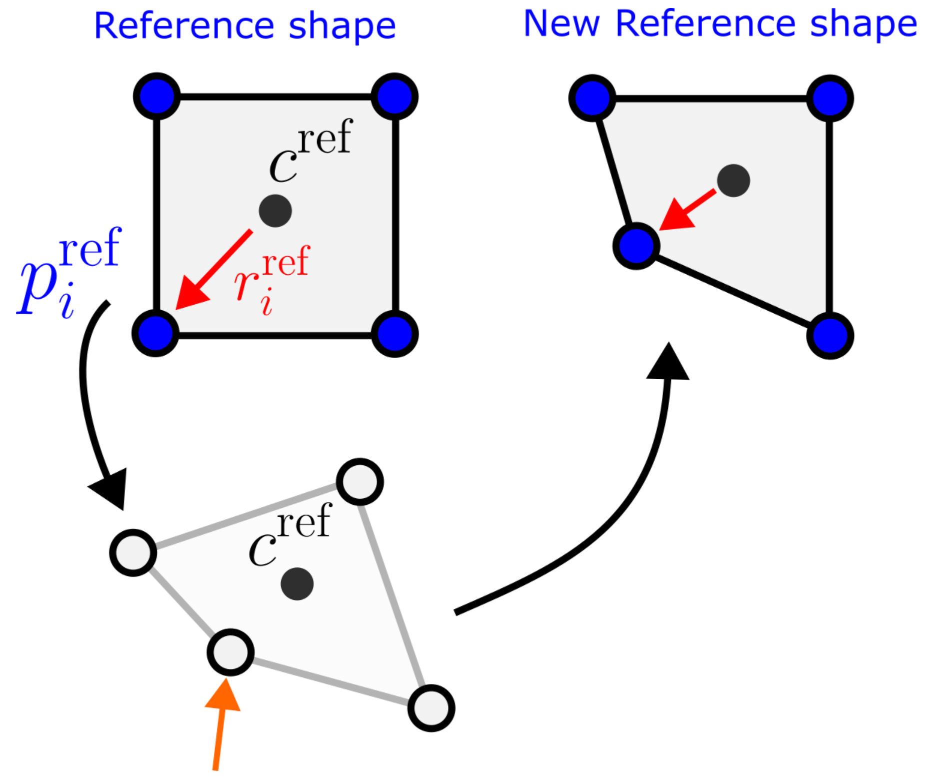
Rigid bodies

Collision b/w bodies using colliders around vertices



Plastic deformation

Update the reference shape for large deformation



[*Muller et al. Meshless Deformations Based on Shape Matching. SIGGRAPH 2005*]

[*Macklin et al. Unified Particle Physics for Real-Time Applications. SIGGRAPH 2014*]