ETH zürich

Physically-Based Simulation Project Plan: Trebuchet attack

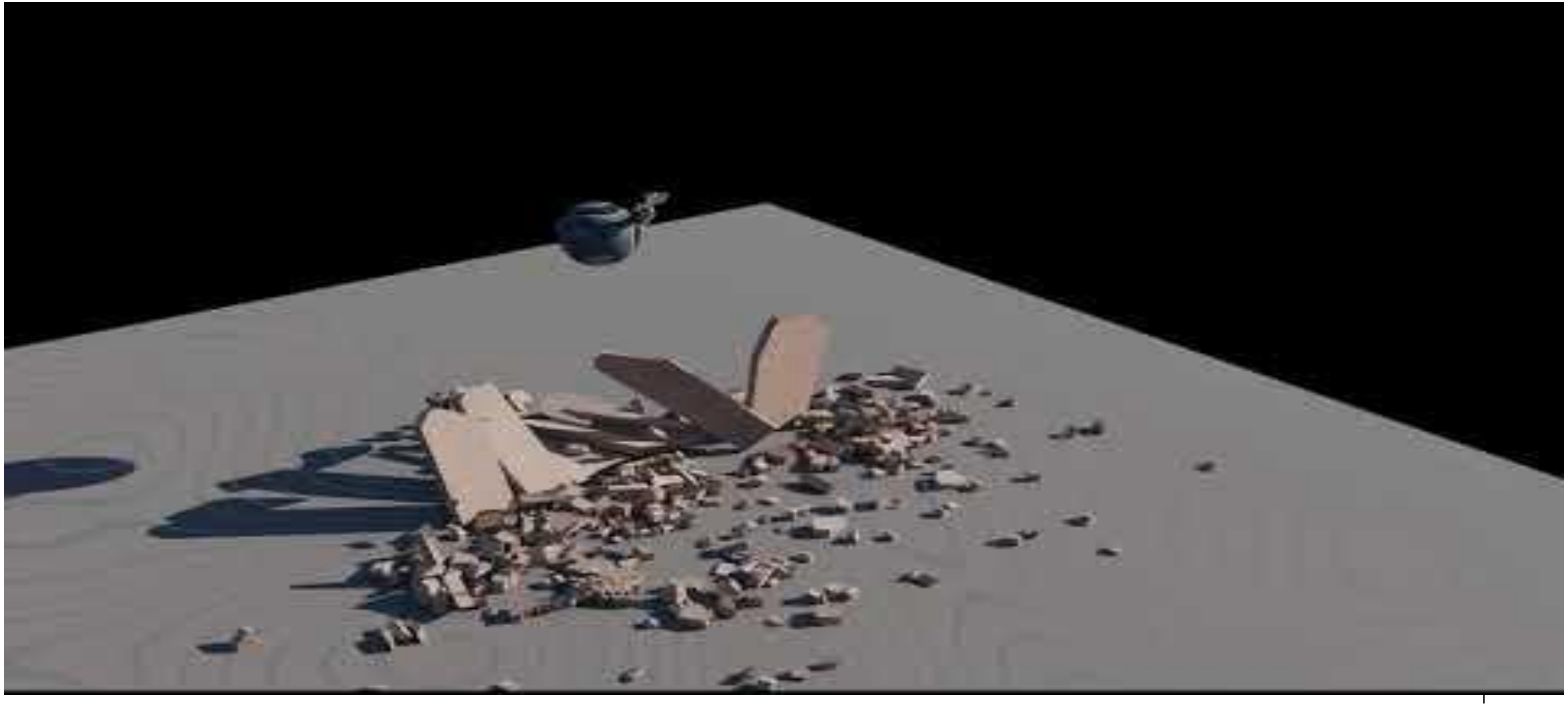
Steve Runser

Computational Biology (CoBi) Group, ETH Zürich

ETH zürich

Simulation Scenario

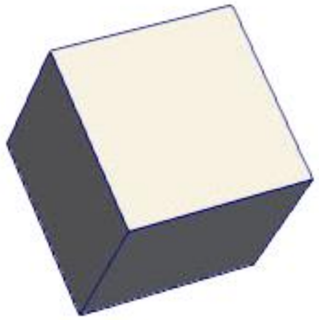
Project goal: Destruction of a wall



Rigid body simulation solver core development

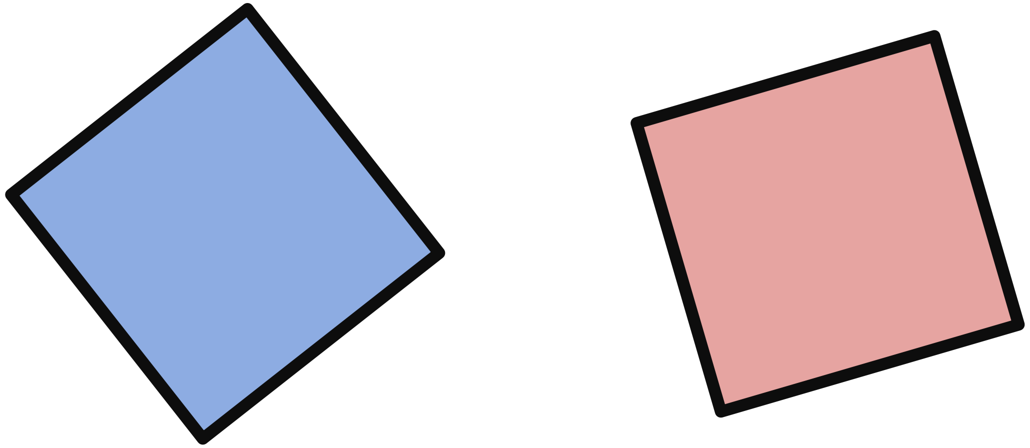
- Development of the mesh representation (nodes, faces, rigid bodies)
- Development of the kinematics:
  - Point coordinate: $\vec{p}(t) = R(t)\hat{r} + \vec{x}(t)$
  - Point velocity: $\frac{d}{dt}\vec{p}(t) = \vec{\omega} \times \hat{r} + \vec{v}$
  - Rotation represented by quaternions: $q_{t+1} = \frac{1}{2}\omega_{t+1}\, q_t$

Rotation stability test



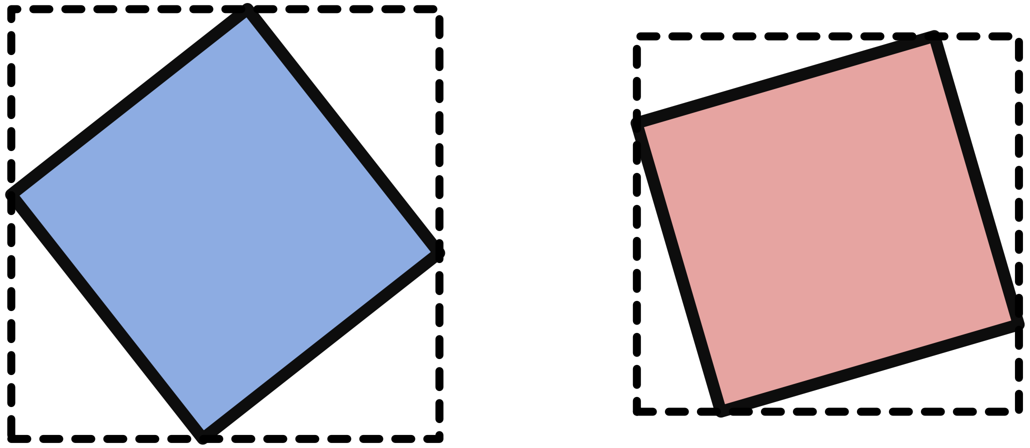
Collision detection: Broad phase

- The space is partitioned with a uniform space partitioning grid



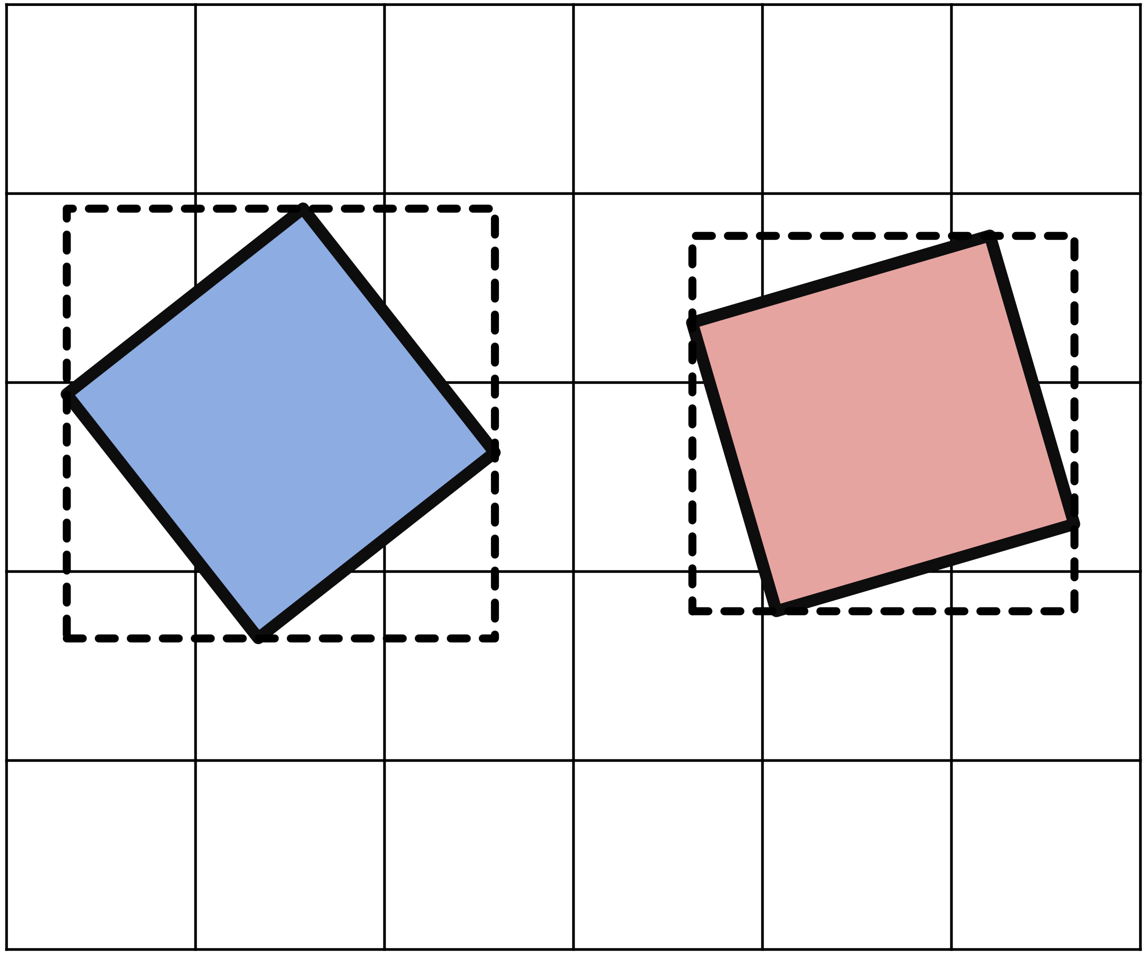
Collision detection: Broad phase

- The space is partitioned with a uniform space partitioning grid



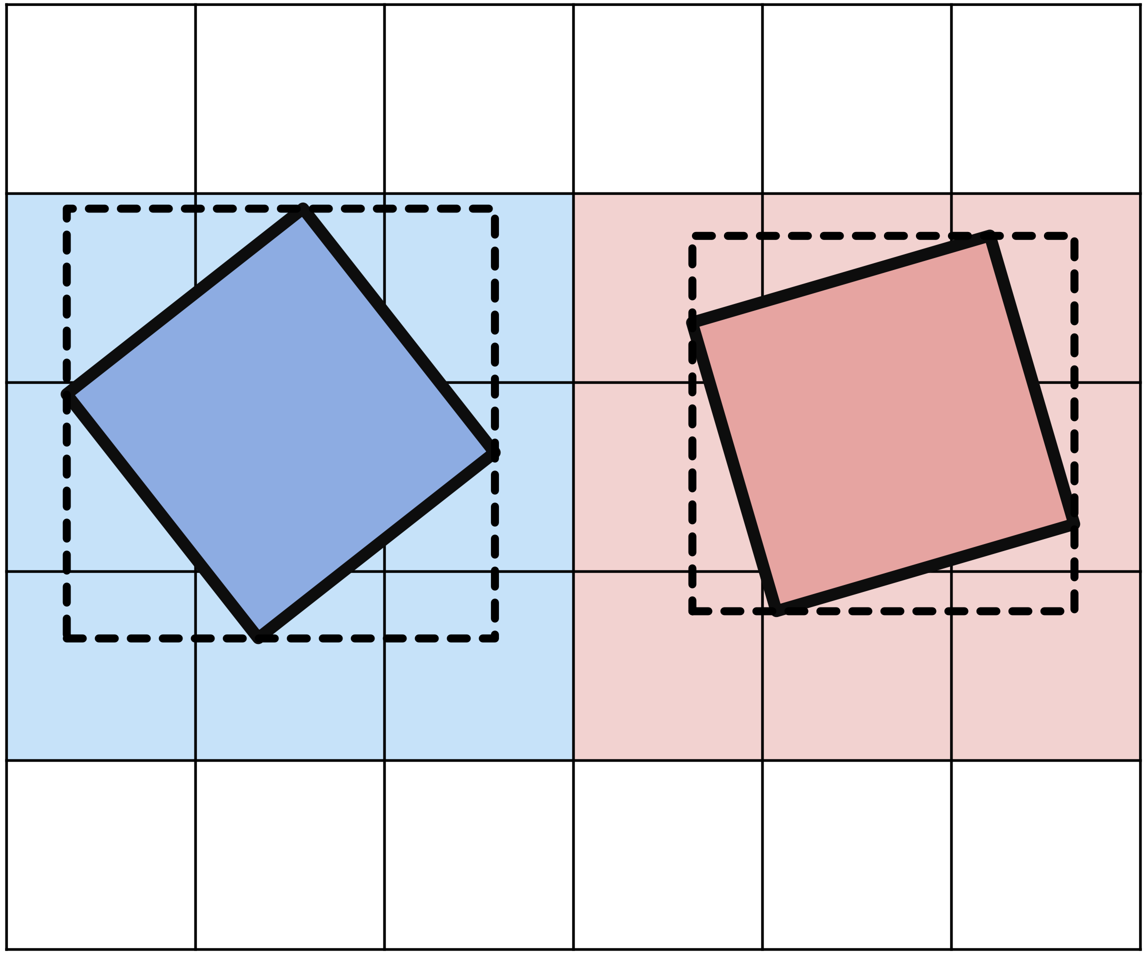
Collision detection: Broad phase

- The space is partitioned with a uniform space partitioning grid



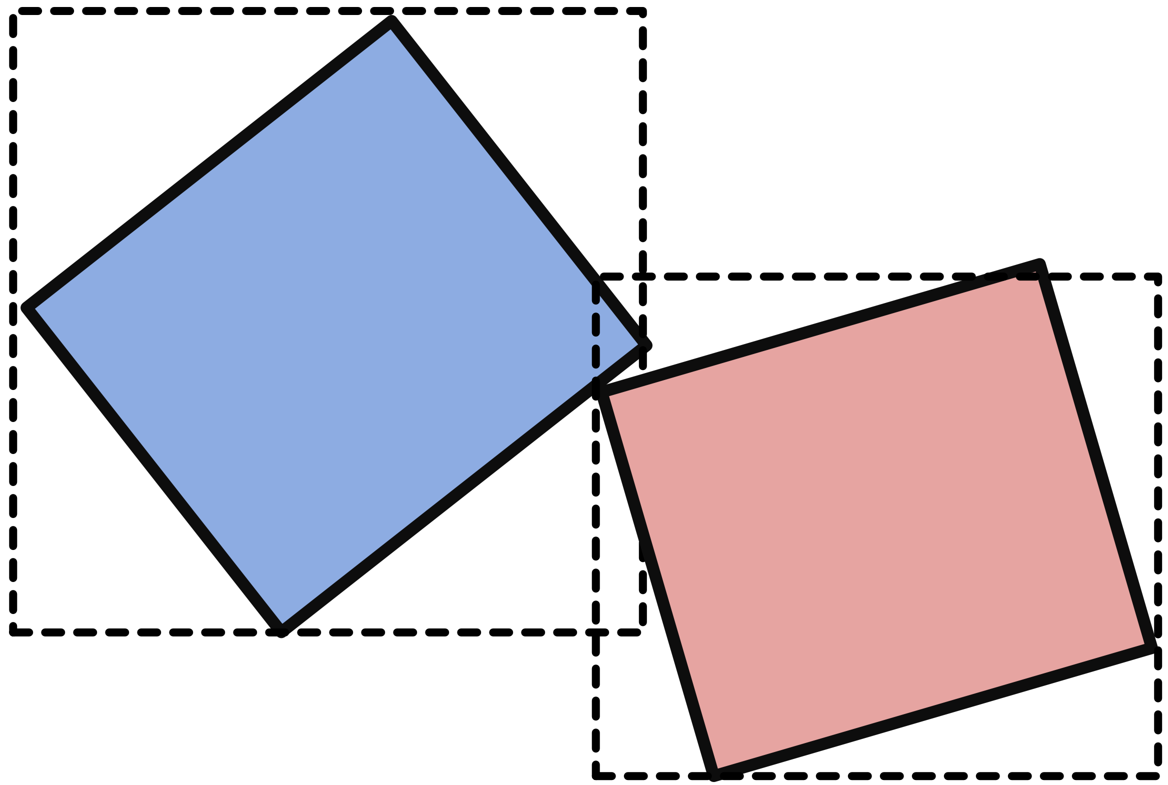
Collision detection: Broad phase

- The space is partitioned with a uniform space partitioning grid



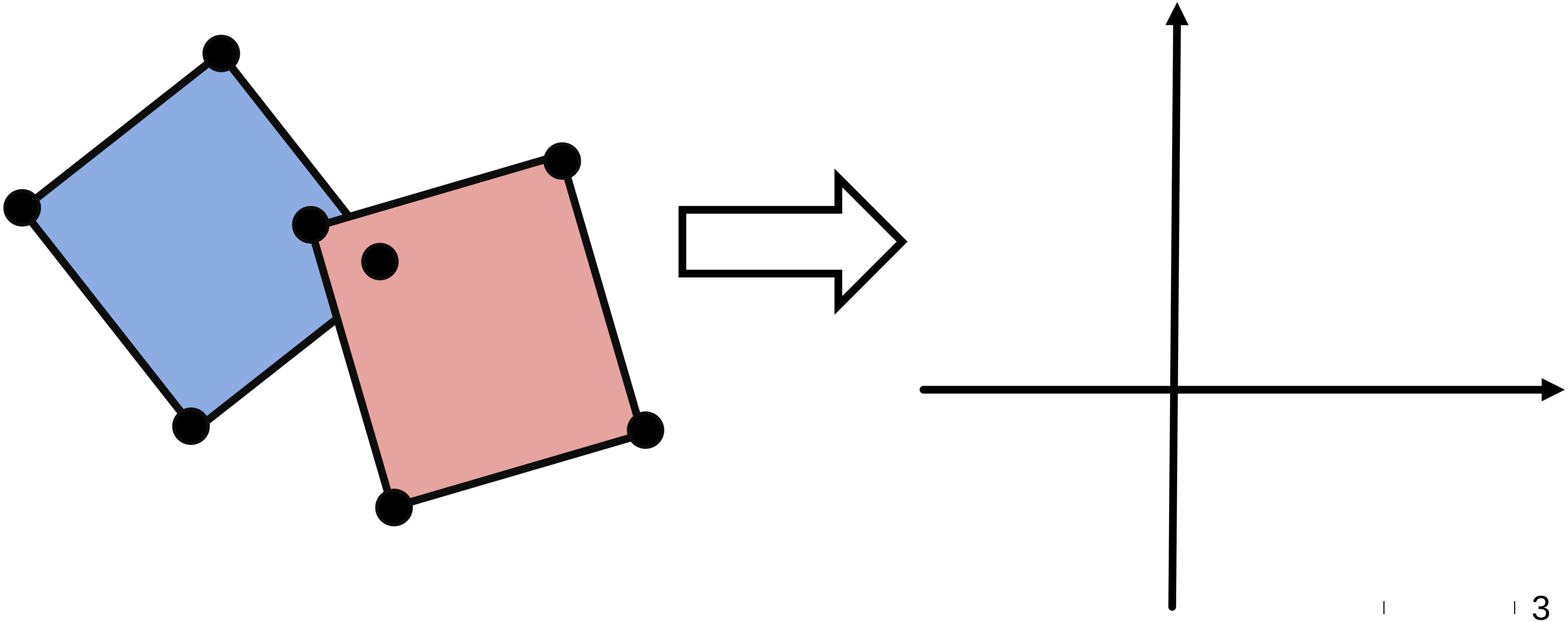
Collision detection: Narrow phase

- Step 1: Axis aligned bounding boxes test



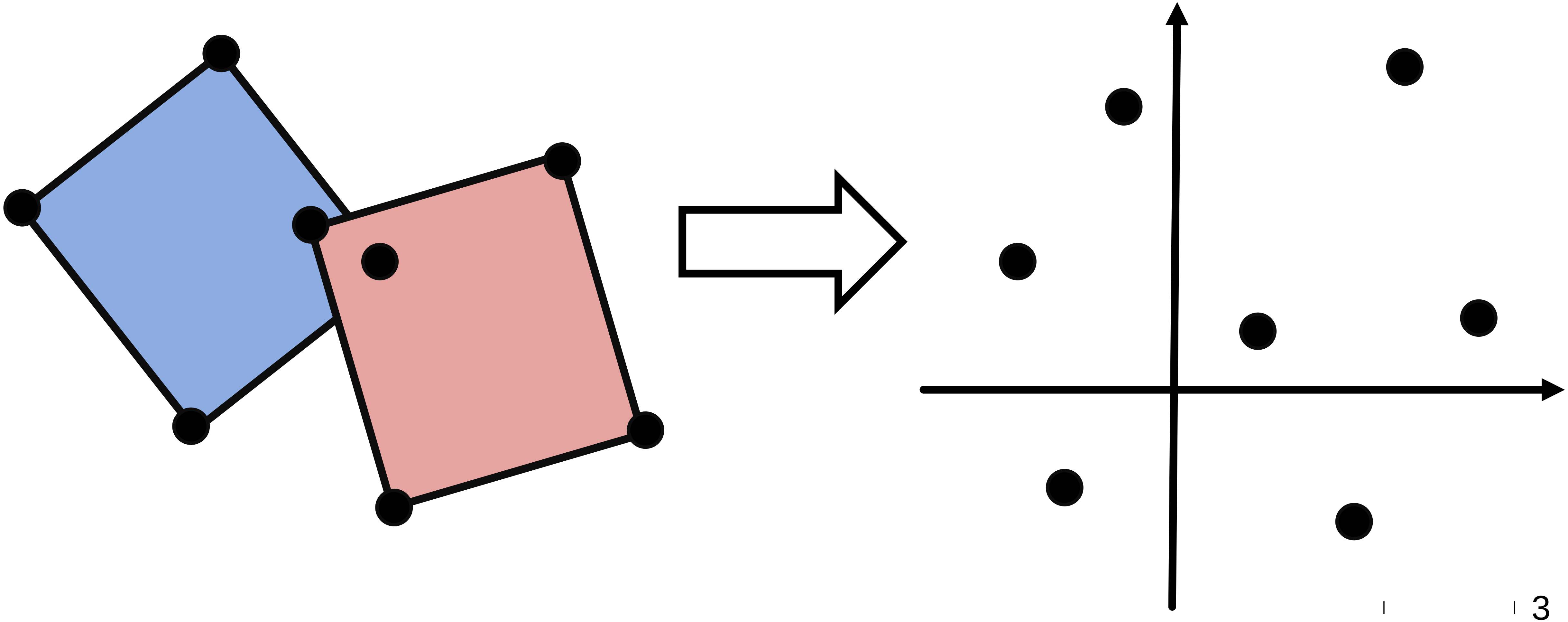
Collision detection: Narrow phase

- Step 2: Minkowski sum calculation



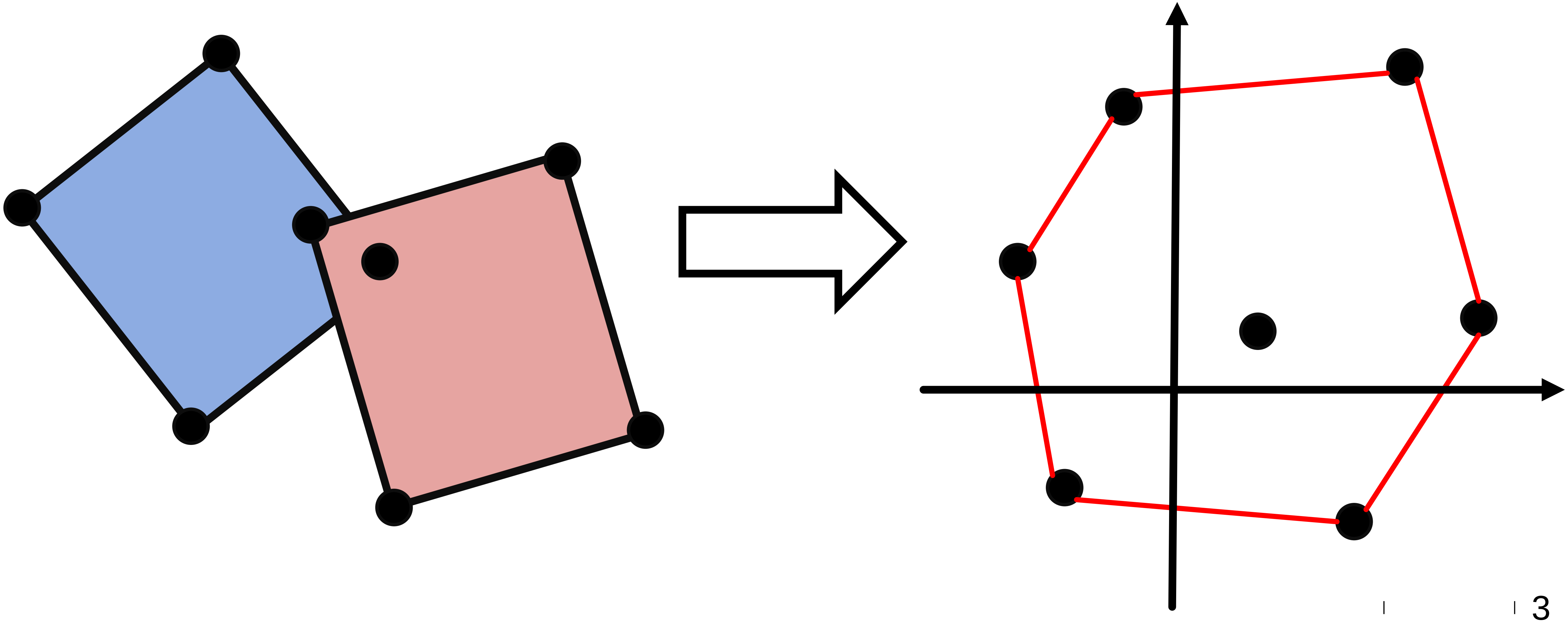
Collision detection: Narrow phase

- Step 2: Minkowski sum calculation



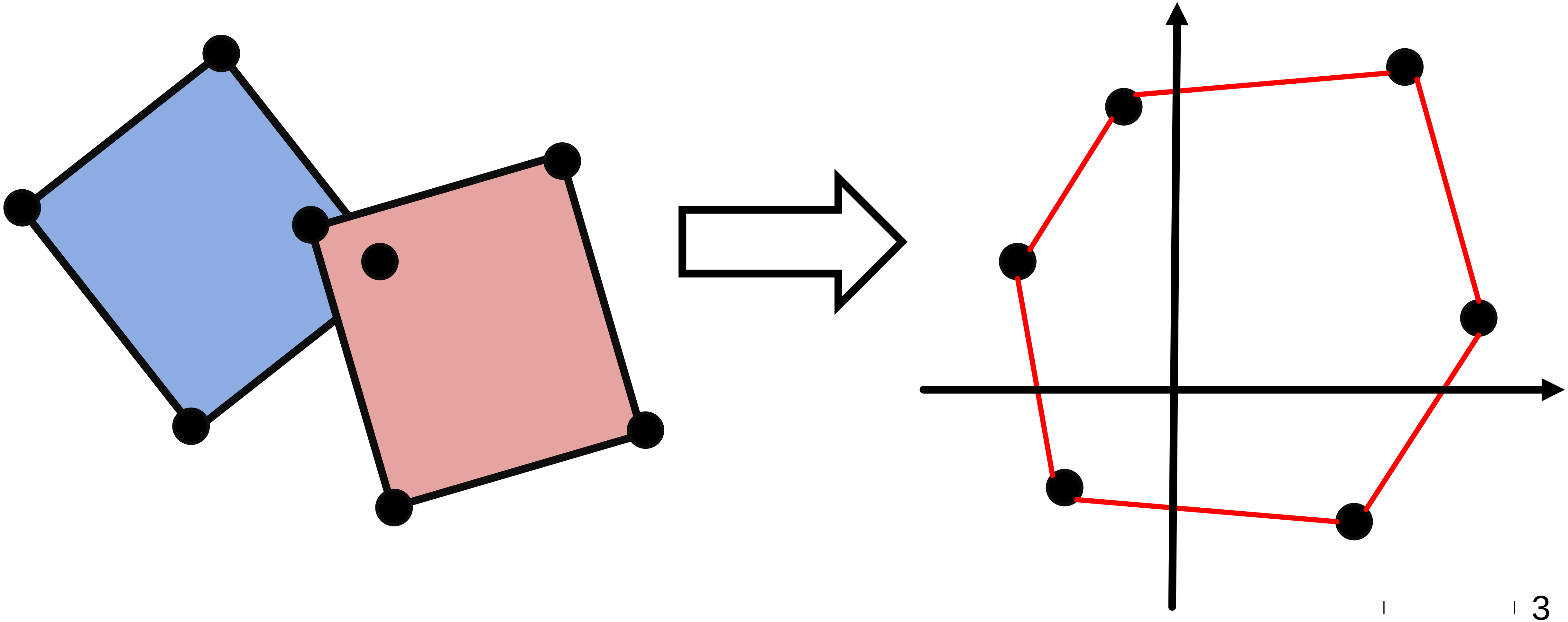
Collision detection: Narrow phase

- Step 2: Minkowski sum calculation



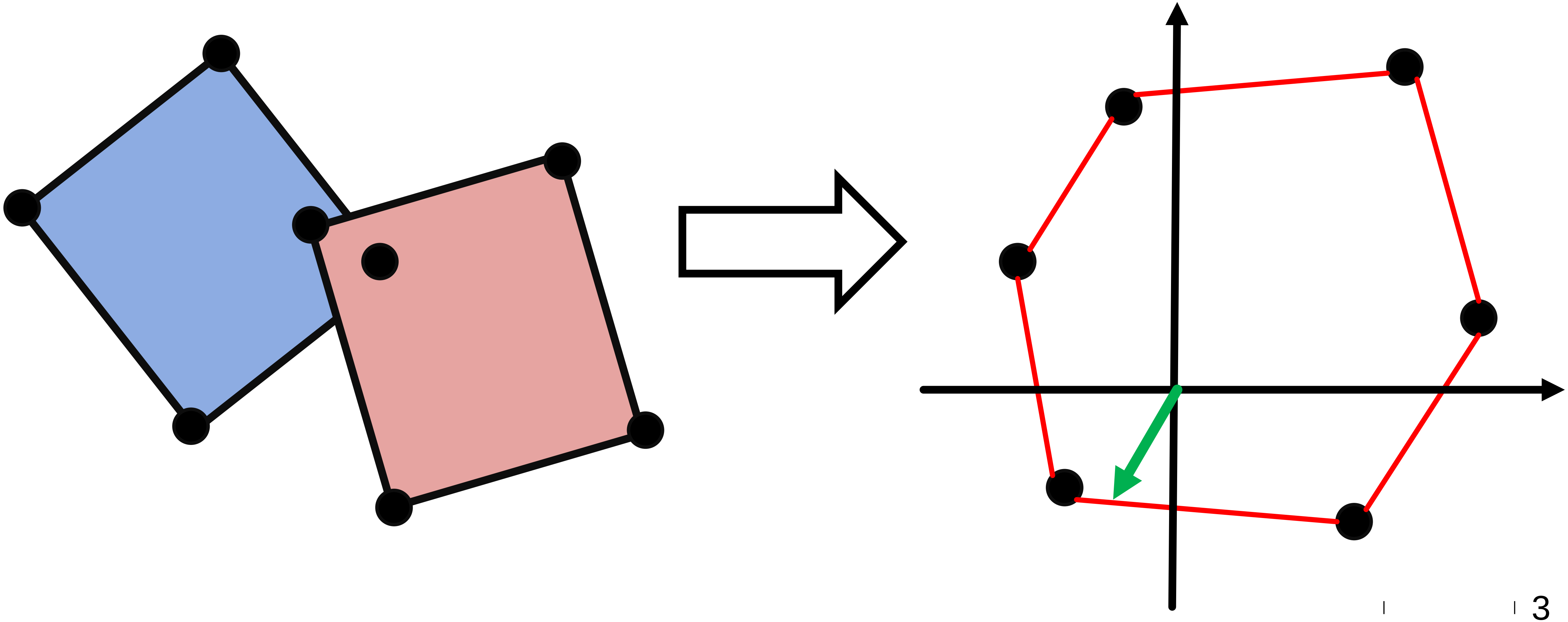
Collision detection: Narrow phase

- Step 2: Minkowski sum calculation



Collision detection: Narrow phase

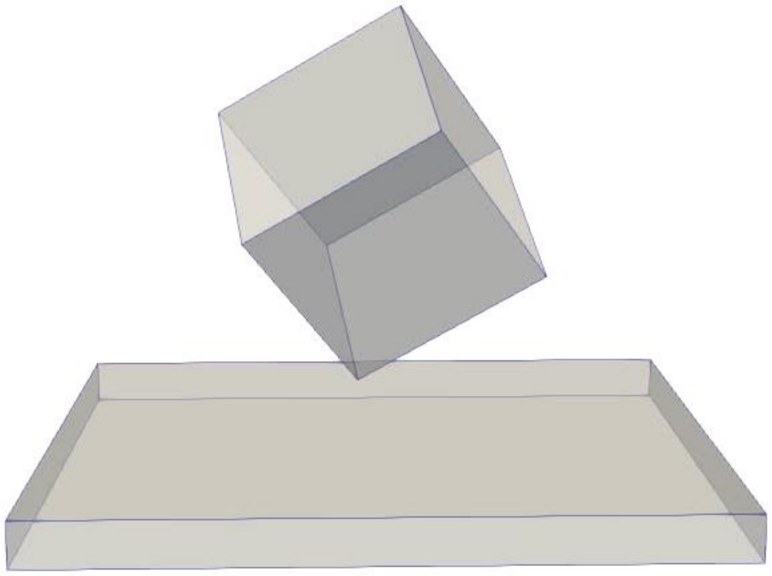
- Step 2: Minkowski sum calculation



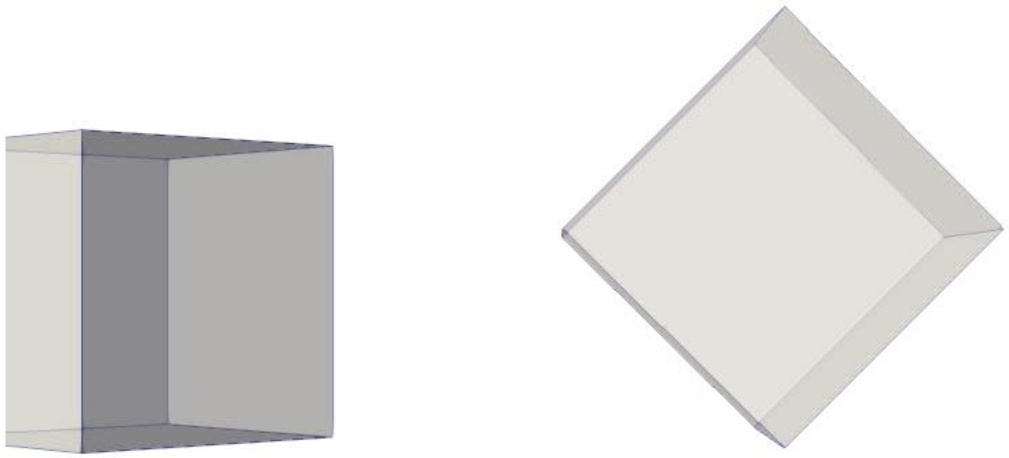
Collision detection: Narrow phase

- Step 3: Distance calculations between simplexes

Face / Node contact



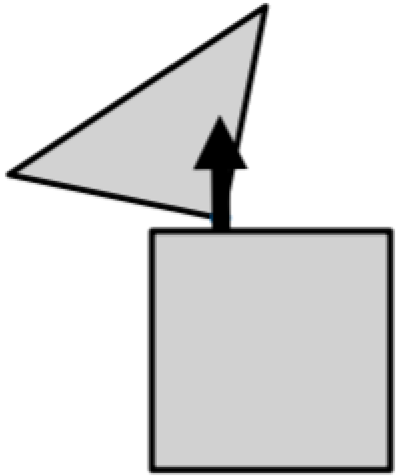
Edge / Edge contact



Collision response

- Rigid Body Simulation II— Nonpenetration Constraints, David Baraff
- Impulse based collision

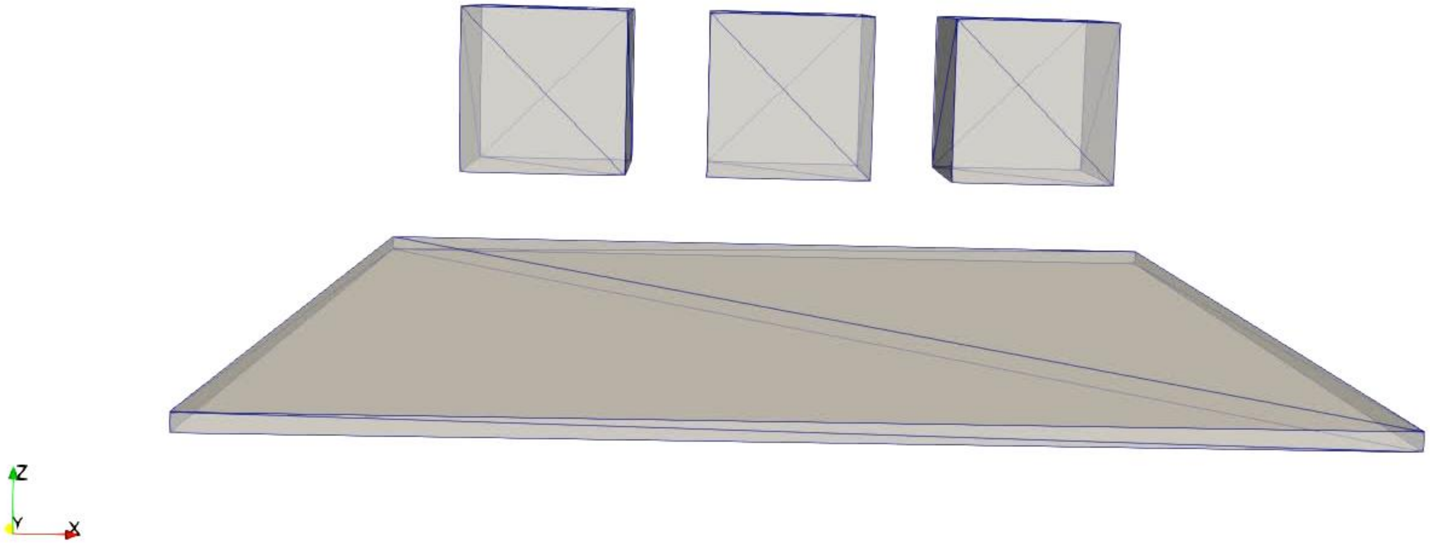
$$j = -\frac{(1+\epsilon)\vec{v}_{rel}^{-}}{M_a^{-1} + M_b^{-1} + \vec{n}\cdot(\boldsymbol{I}^{-1}(\vec{r}_a \times \vec{n})) \times \vec{r}_a + \vec{n}\cdot(\boldsymbol{I}^{-1}(\vec{r}_b \times \vec{n})) \times \vec{r}_b}$$



Source: lecture course Rigid Bodies II slides

Collision response: impulse based collision

- Rigid Body Simulation II— Nonpenetration Constraints, David Baraff



Collision response: Resting contact forces

- Rigid Body Simulation II— Nonpenetration Constraints, David Baraff
- Solving a linear complementarity problem

$$\mathbf{w} = \mathbf{A}\mathbf{f} + \mathbf{b} \geq 0$$

$$\mathbf{f} \geq 0$$

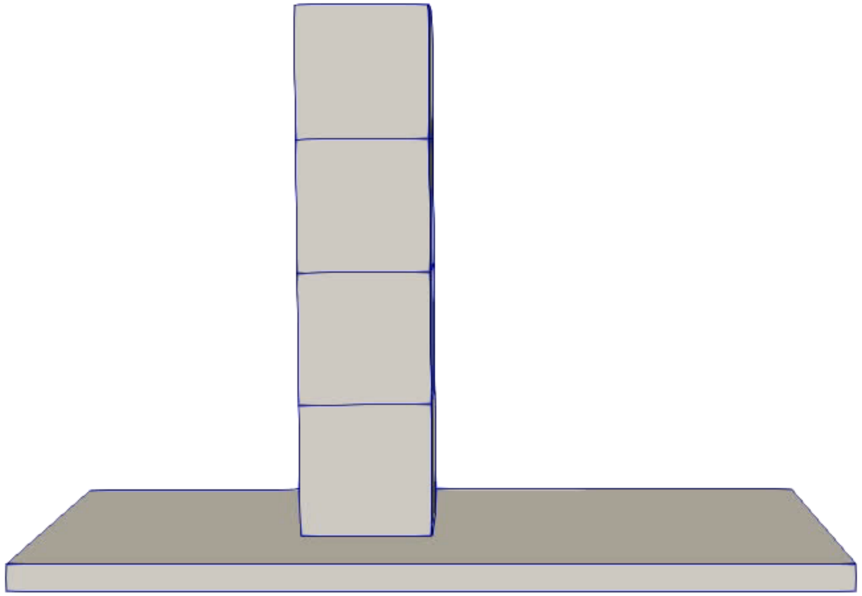
$$\mathbf{f}^T(\mathbf{w}) = 0$$

$$A_{ij} = \vec{n}_i \cdot \left(\left(\frac{\vec{n}_j}{M_{A_i}} + \lambda_{A_i}(r^*_{A_j}\vec{n}_j) \right) - \left(\frac{-\vec{n}_j}{M_{B_i}} + \lambda_{B_i}(r^*_{B_j}(-\vec{n}_j)) \right) \right)$$

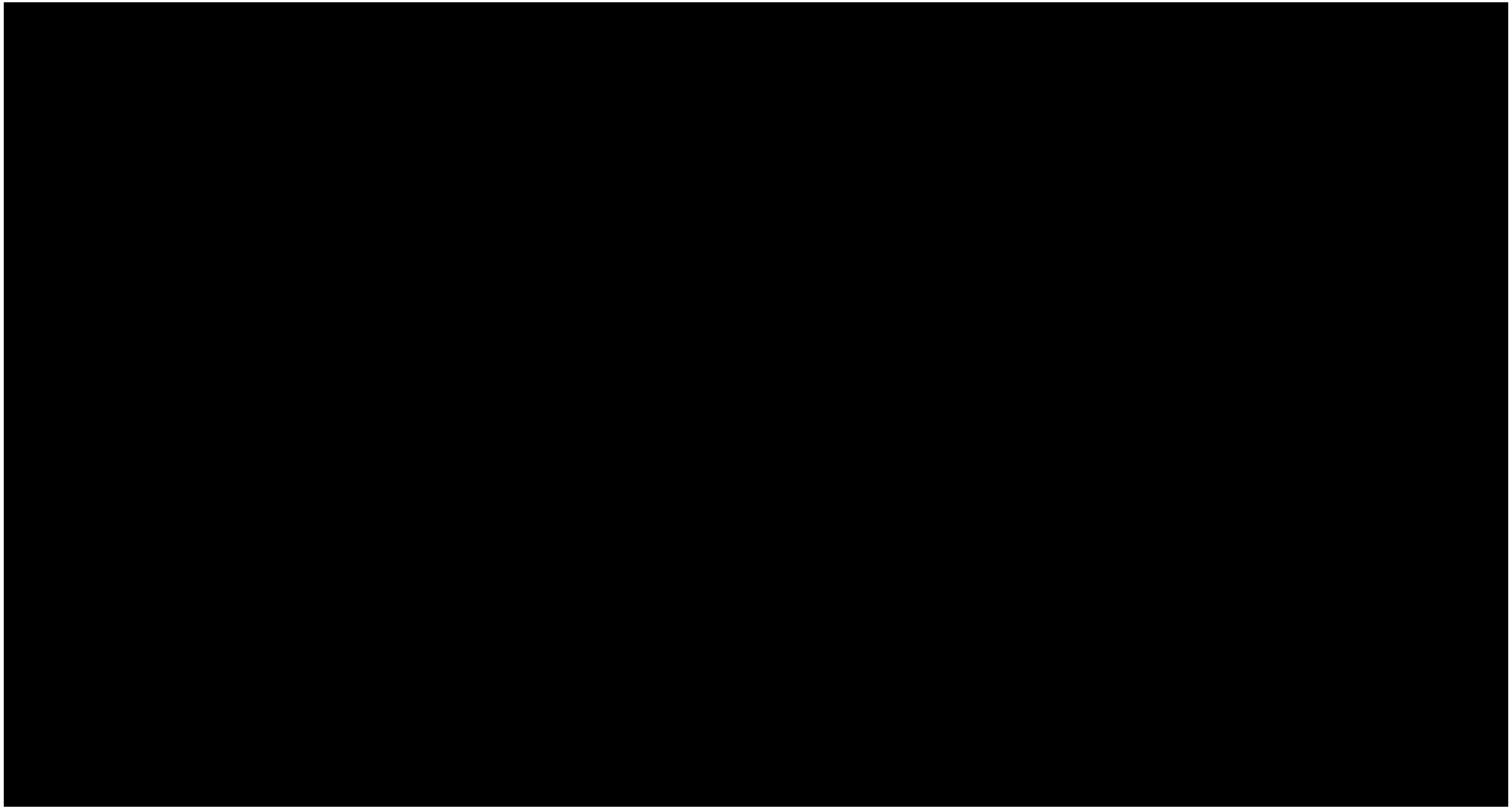
$$b_i = v^-_{rel_i}(1 + \varepsilon_i)$$

Collision response: Resting contact forces

- Rigid Body Simulation II— Nonpenetration Constraints, David Baraff
- Solving a linear complementarity problem



Simulations



Conclusion

1. Generate the initial geometry of a castle / wall
2. Implement collision detection
   1. Broad phase: Space partitioning grid and oriented bounding boxes
   2. Narrow phase: Gilbert–Johnson–Keerthi distance algorithm
3. Implement collision response
   1. Impulse based response
   2. Multiple resting contacts
4. Rendering effects

Conclusion

Thank you for your attention!