Rigid Body Collisions

treuille@cs.cmu.edu

Rigid Body Dynamics

Collision and Contact

David Baraff

PIXAR
ANIMATION STUDIOS

Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces

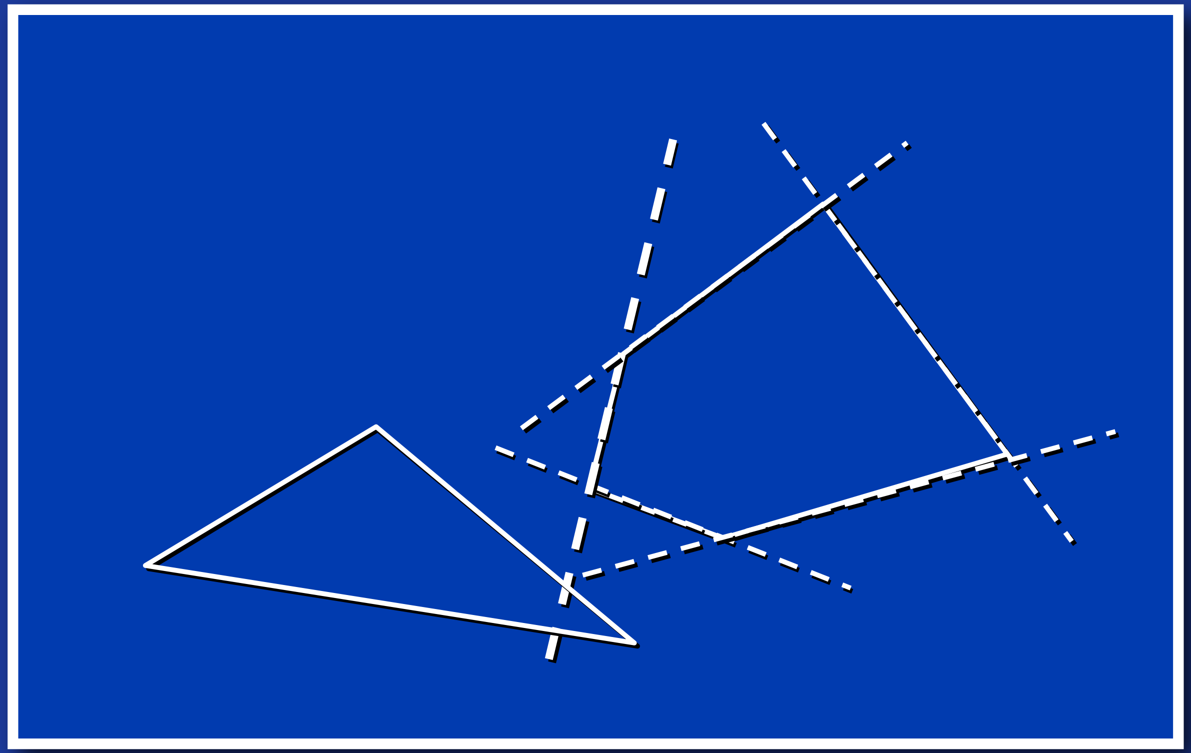
Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces

Problem

- **Positions NOT OK**

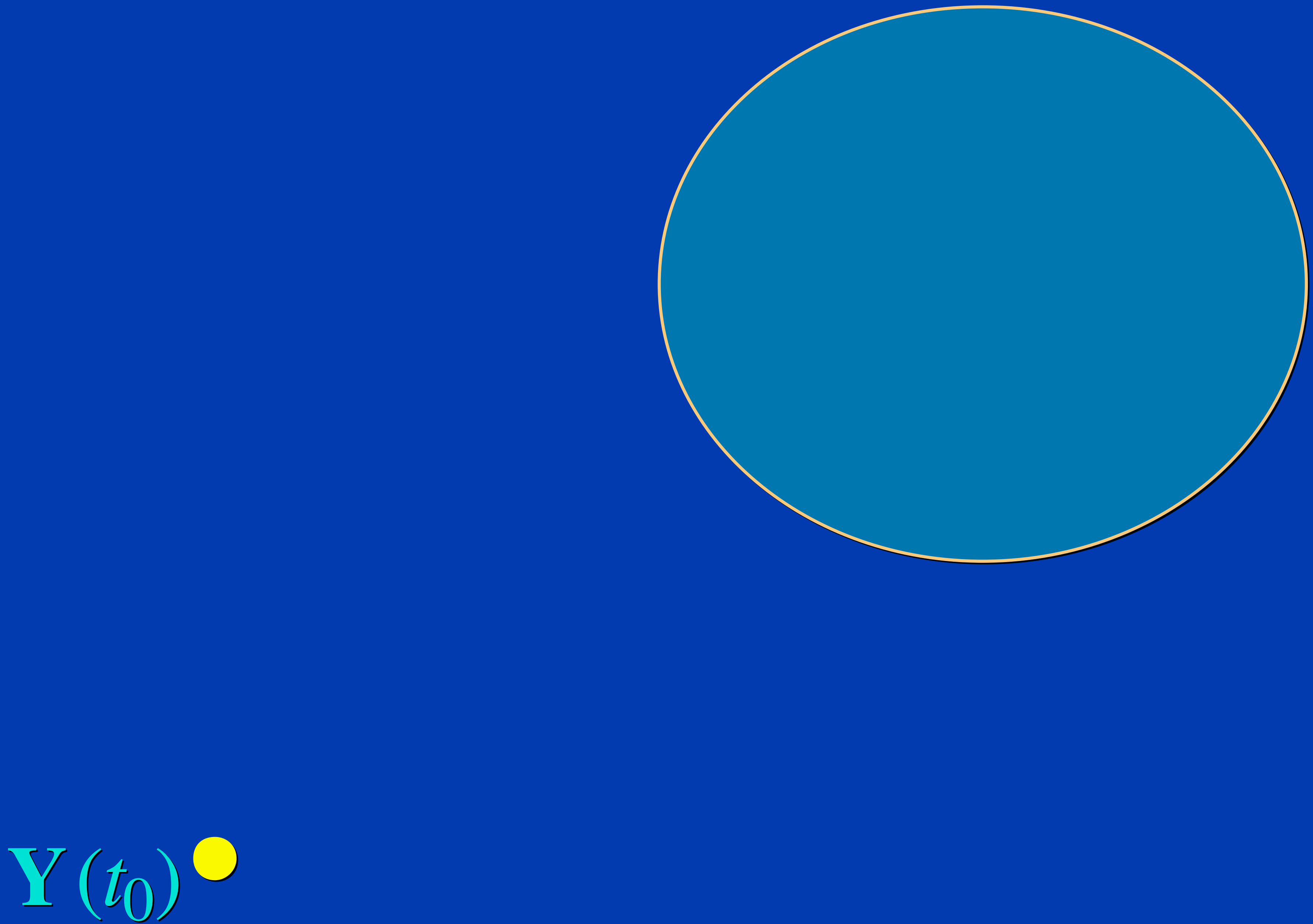
Collision Detection



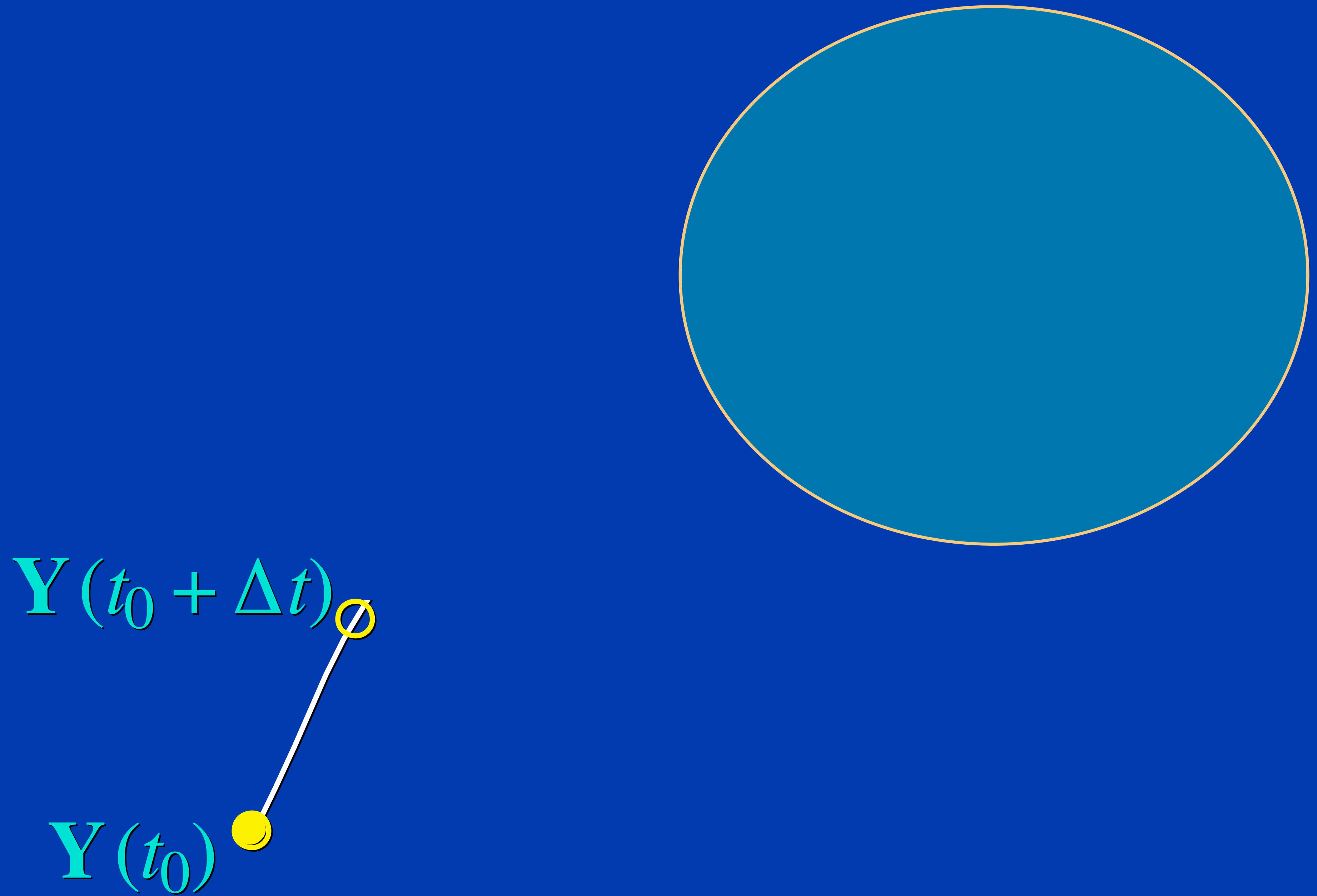
Assume we have some
spatial collision detection algorithm.

(This can be solved in less than $O(n^2)$ time.)

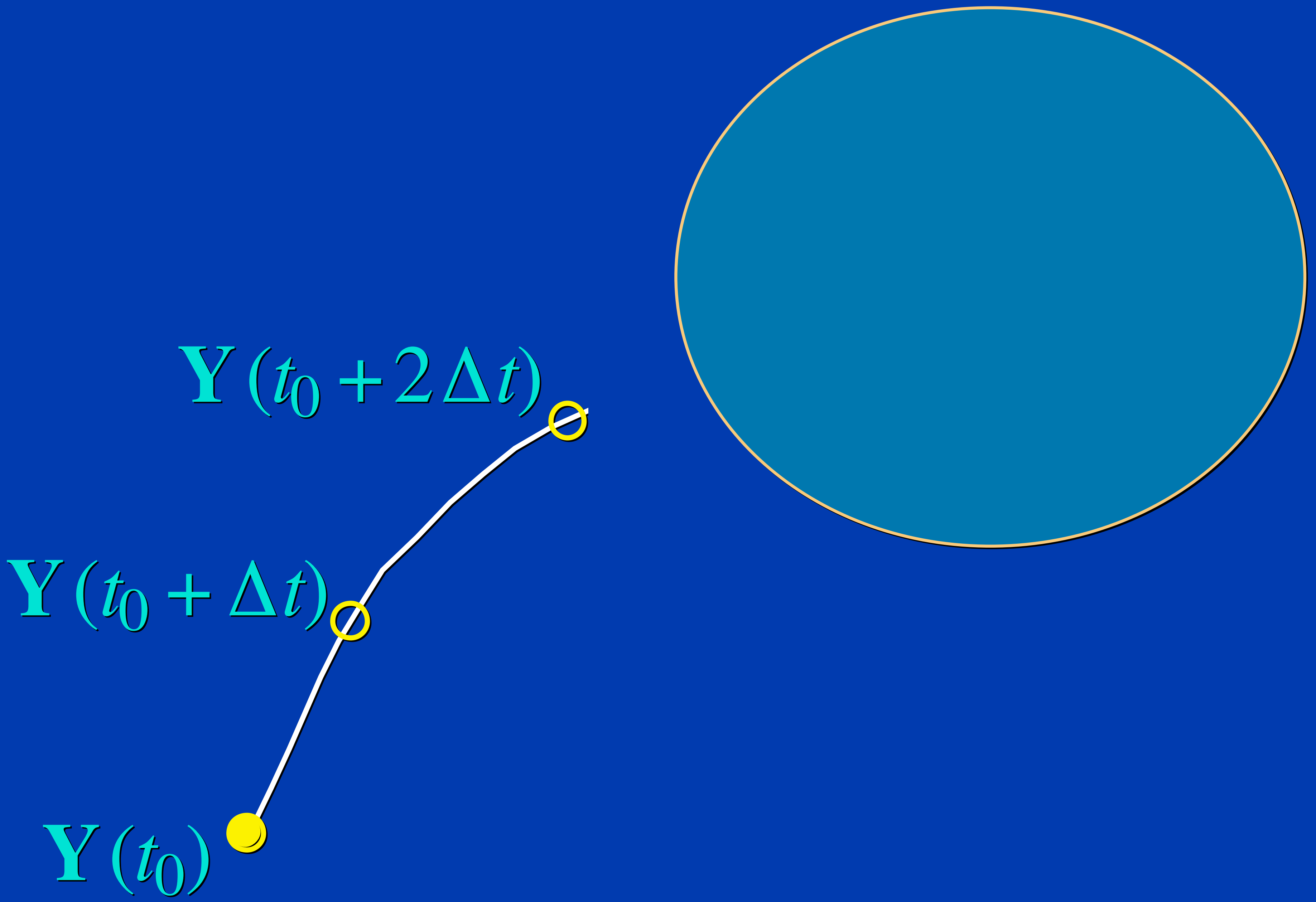
Simulations with Collisions



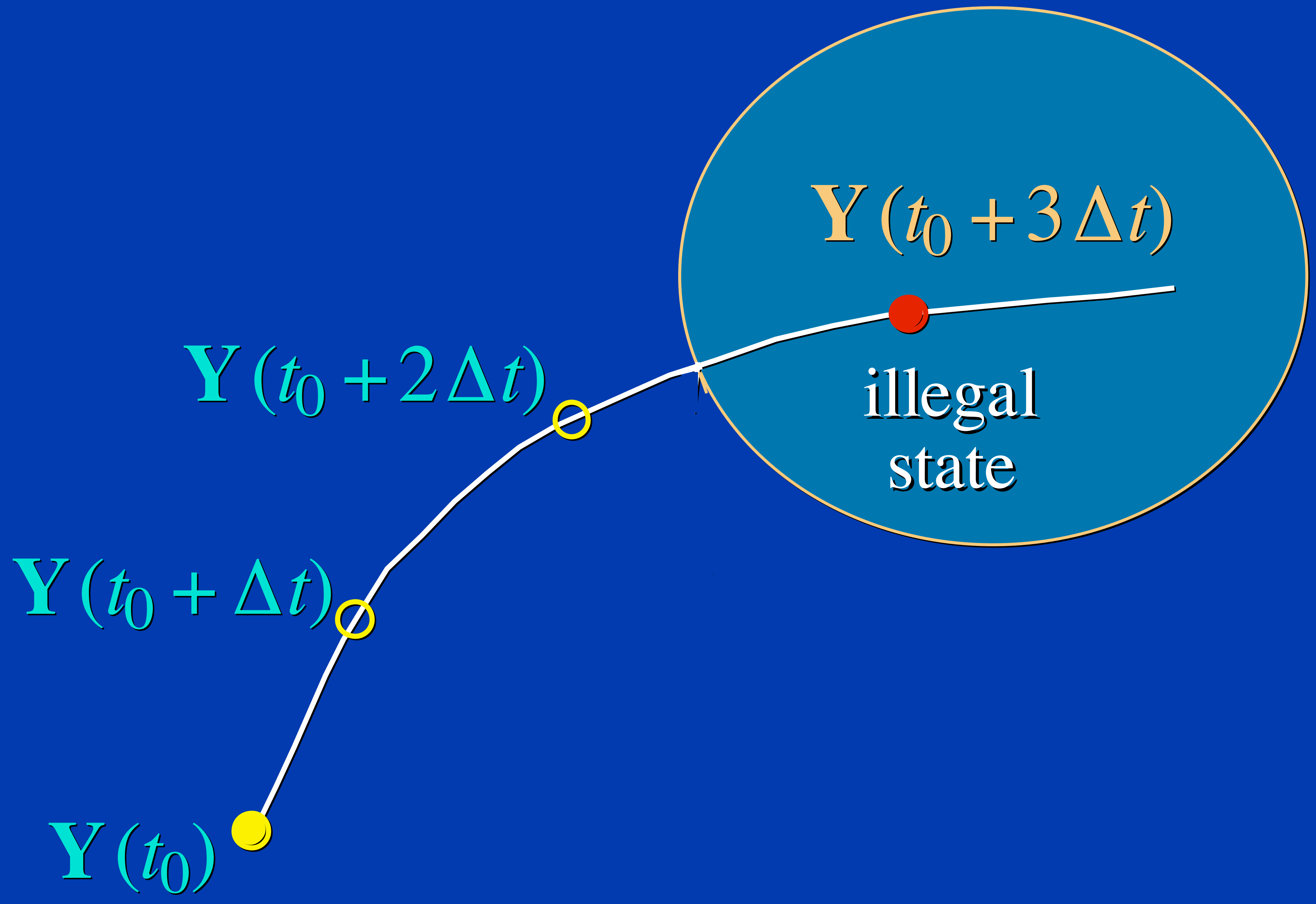
Simulations with Collisions



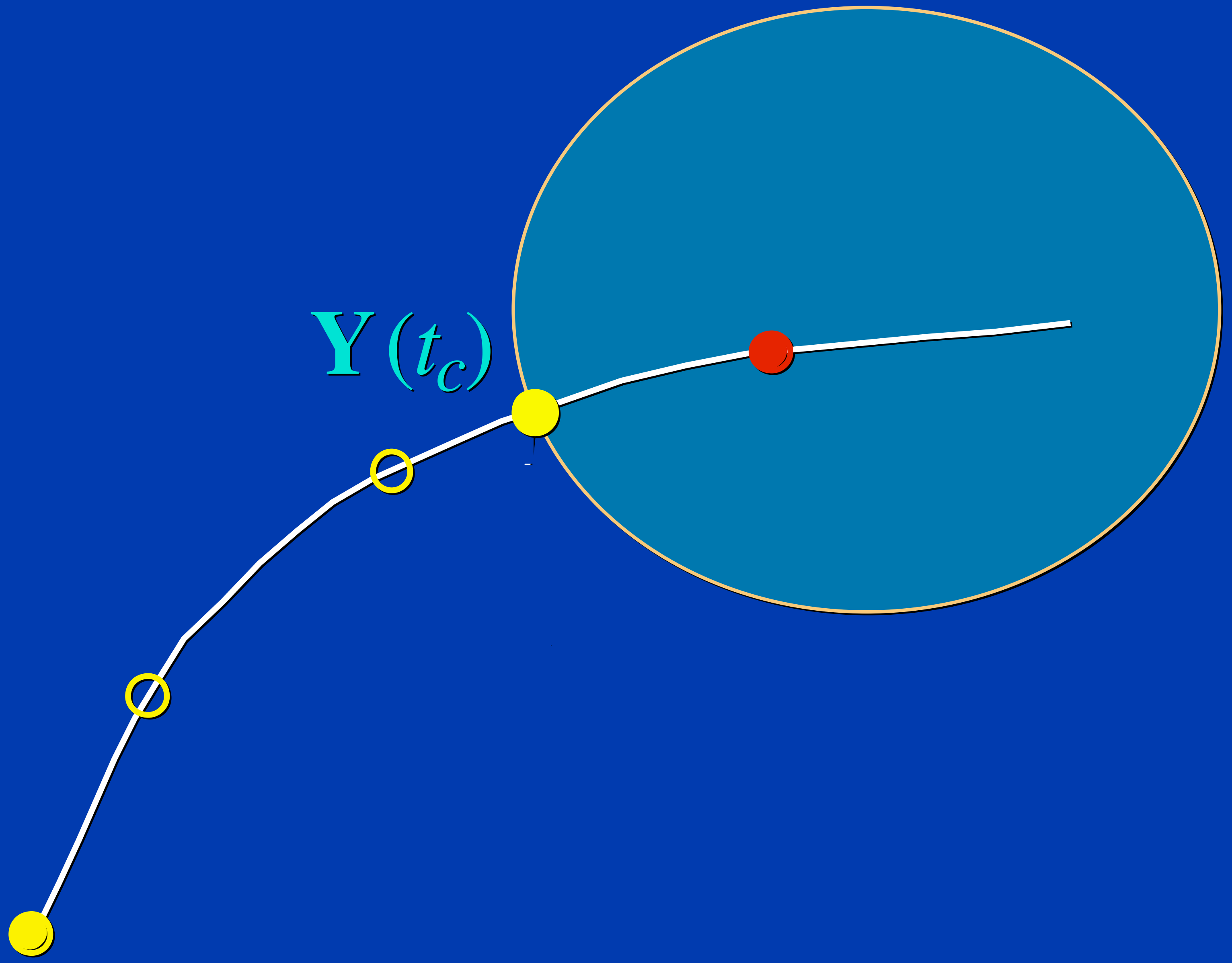
Simulations with Collisions



An Illegal State Y



Backing up to the Collision Time



Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces

Geometric Contact

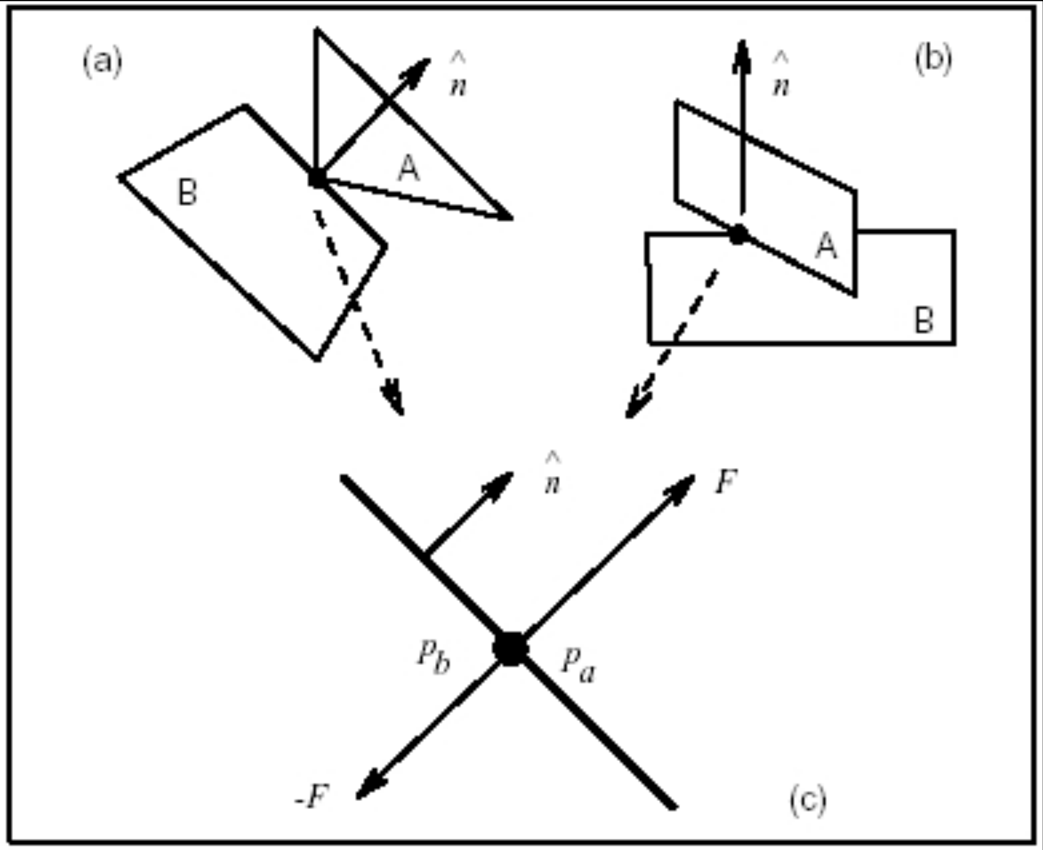
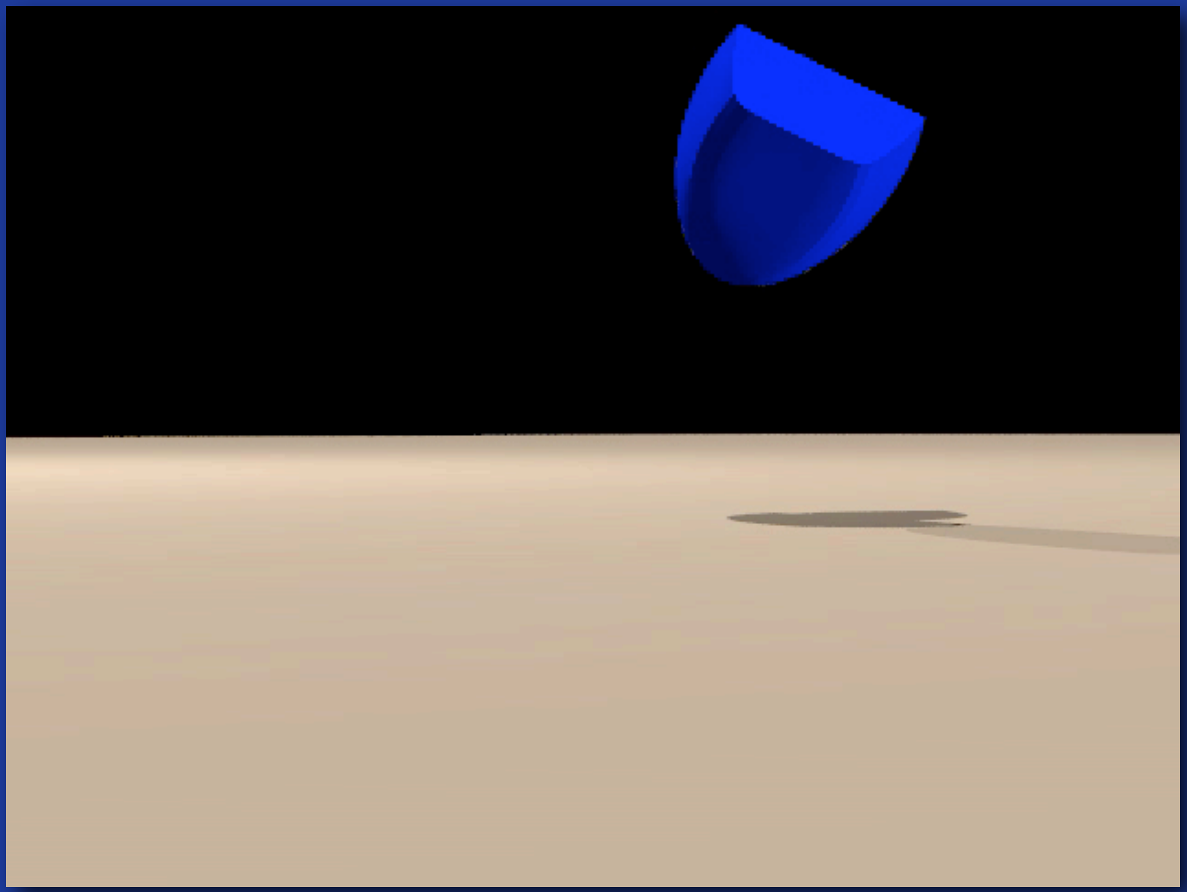


Figure 3. (a) Vertex-plane contact (side view). (b) Edge-edge contact. (c) Contact geometry.

source: http://www.cs.ubc.ca/~van/cpsc526/Vjan2003/projects/gao/index.html

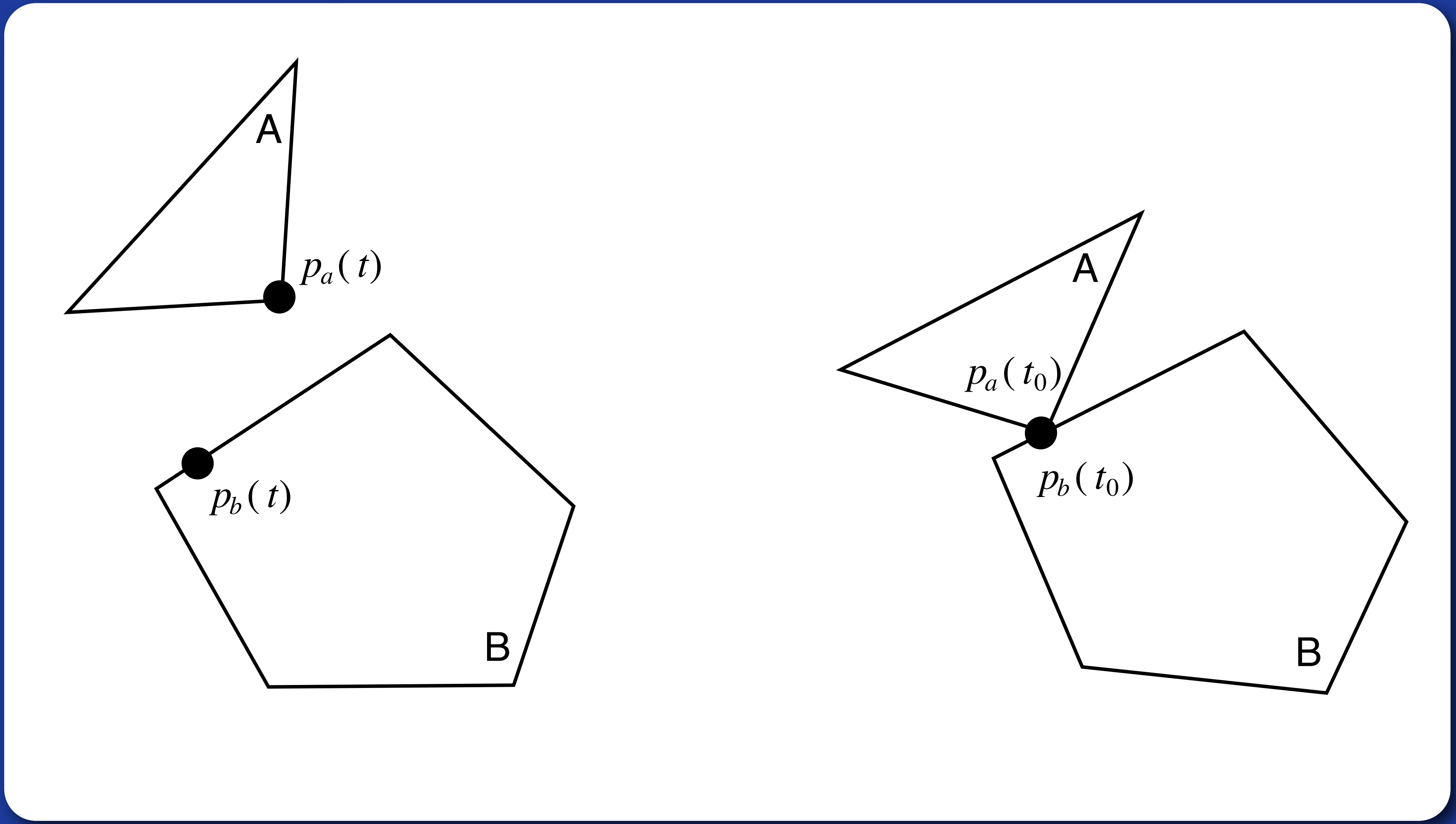
- Vertex-Face
- Edge-Edge

Physical Contact



- Impulse Collision (“bounce”)
- Resting Contact

Physical Contact

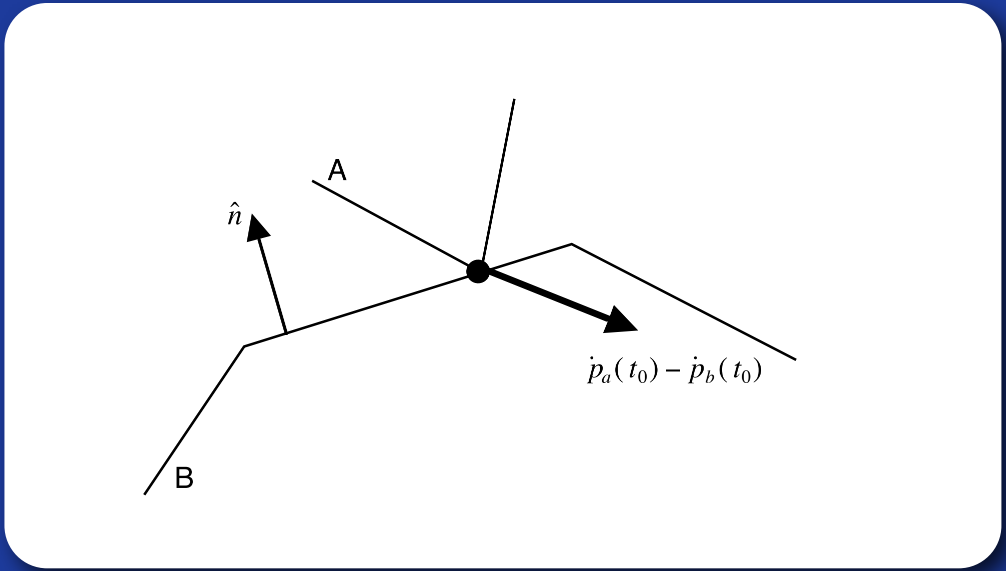


$p_a(t)$ = contact point on body A

$p_b(t)$ = contact point on body B

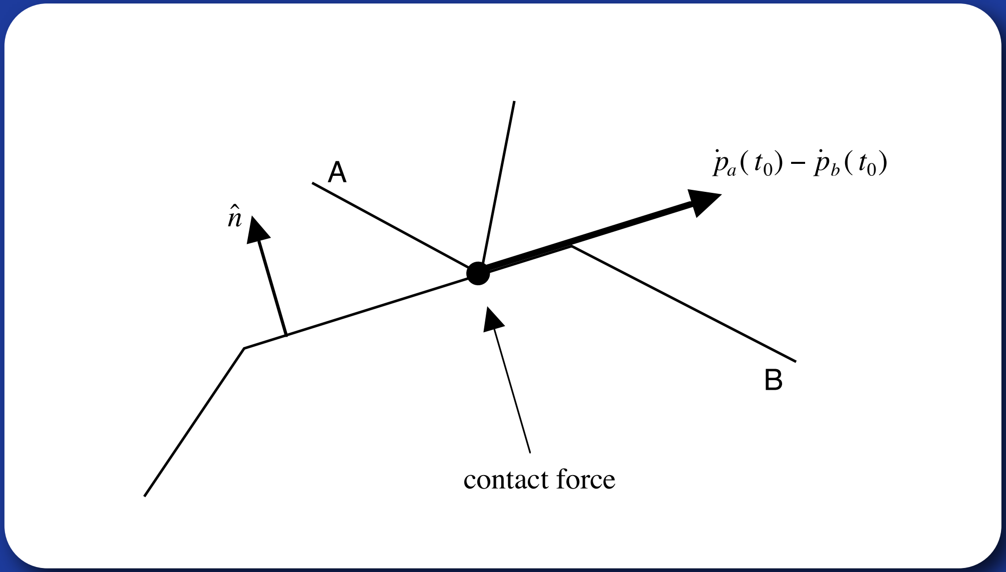
$p_a(t_0) = p_b(t_0)$ but in general $\dot{p}_a(t_0) \neq \dot{p}_b(t_0)$

Physical Contact



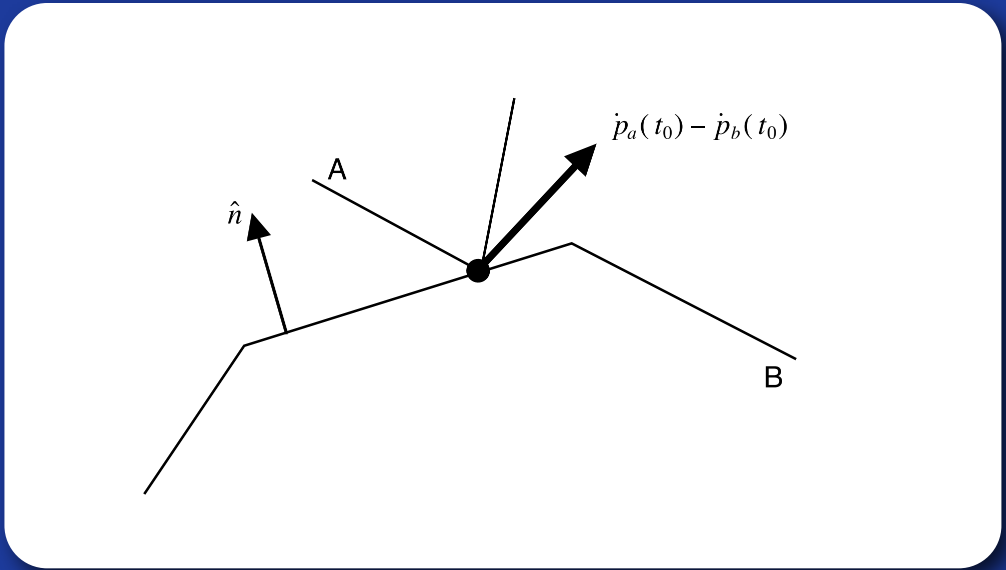
$$(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n < 0$$

Impulse collision.



$$(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n = 0$$

Resting contact.



$$(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n > 0$$

No collision.

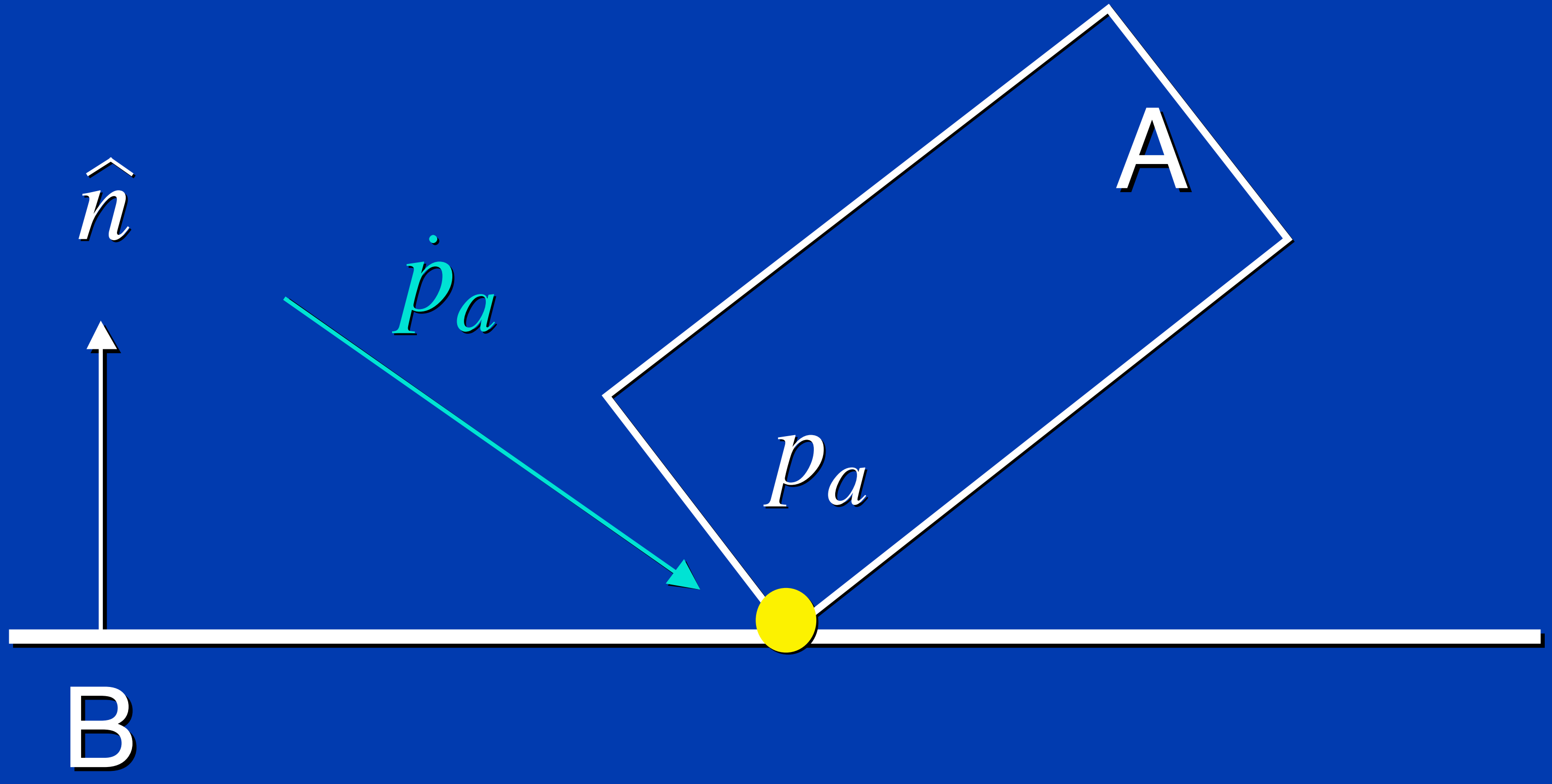
Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
    - Apply Impulse Force
    - Compute Resting Contact Forces

Problem

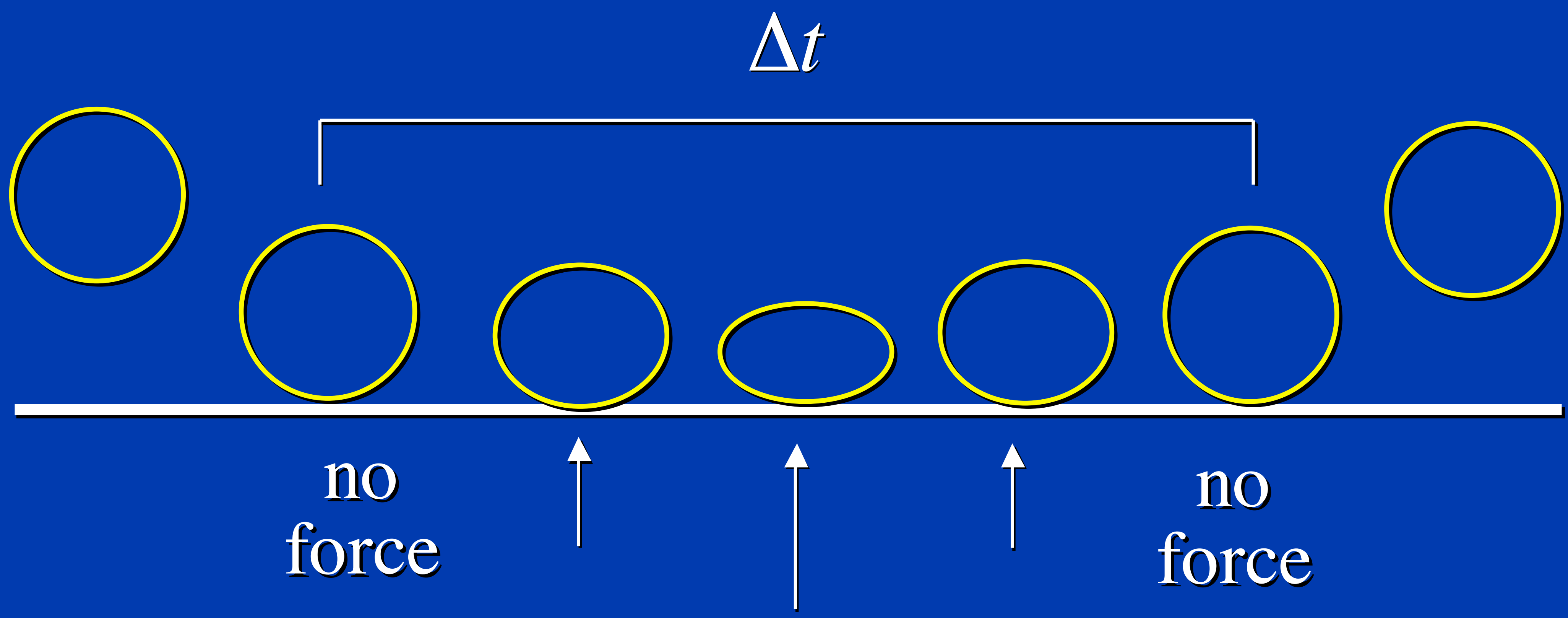
- **Positions OK**
- **Velocities NOT OK**

Colliding Contact



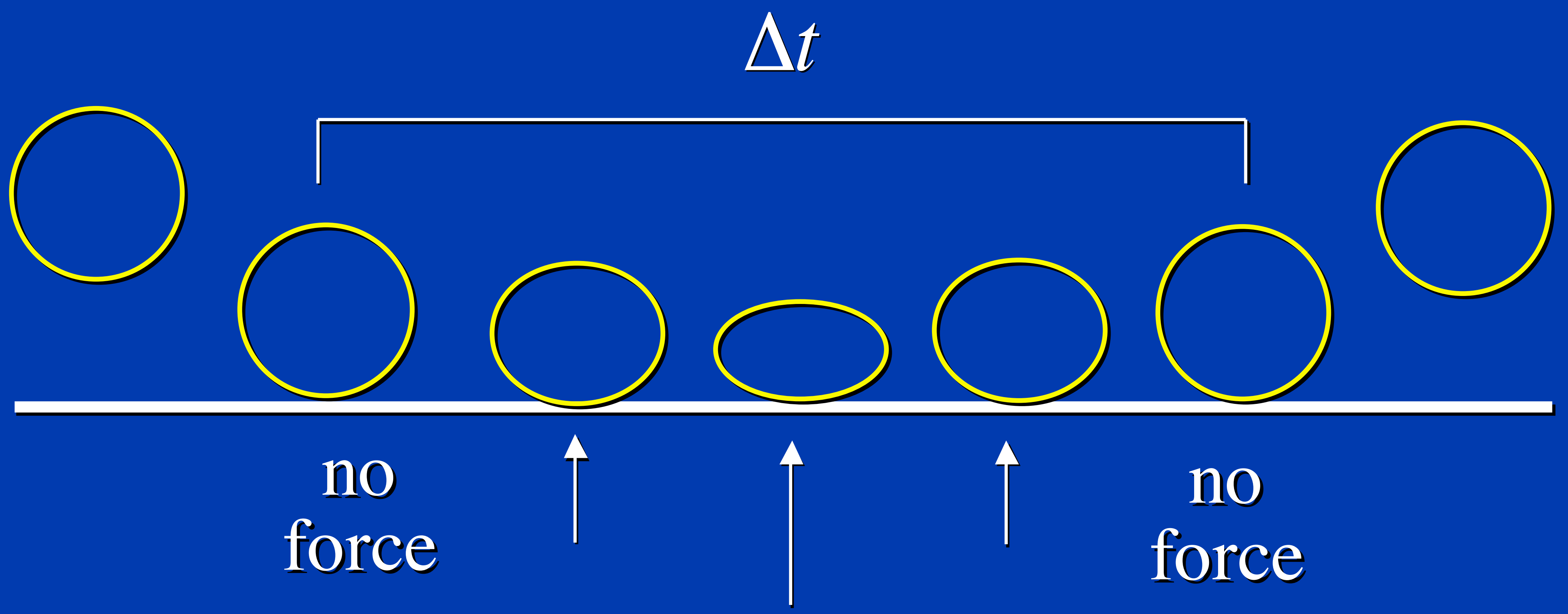
$$\hat{n} \bullet \dot{p}_a < 0$$

Collision Process



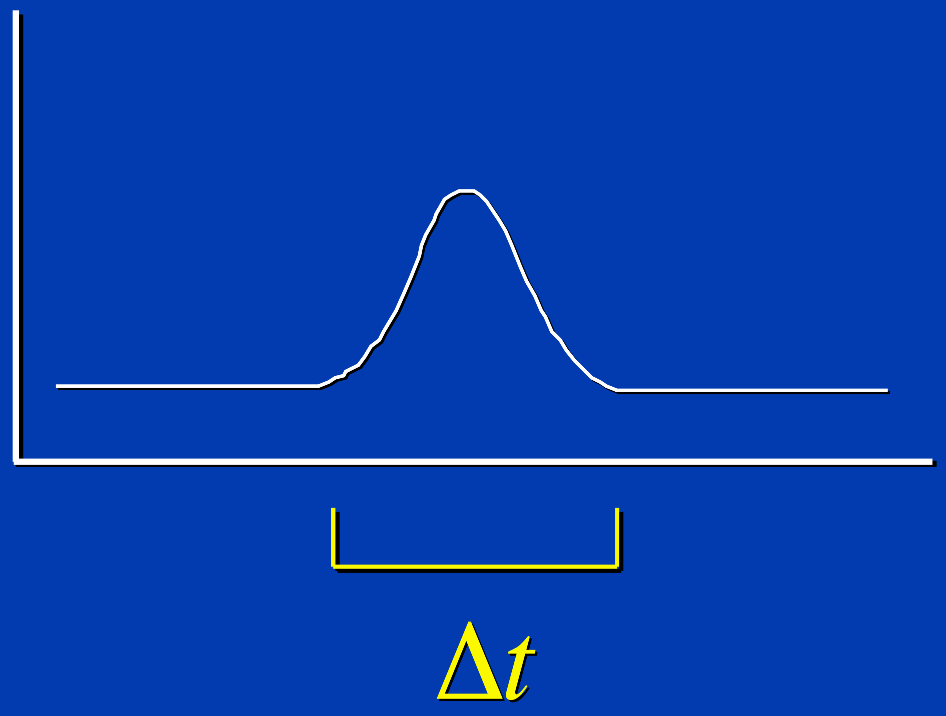
Video

Collision Process

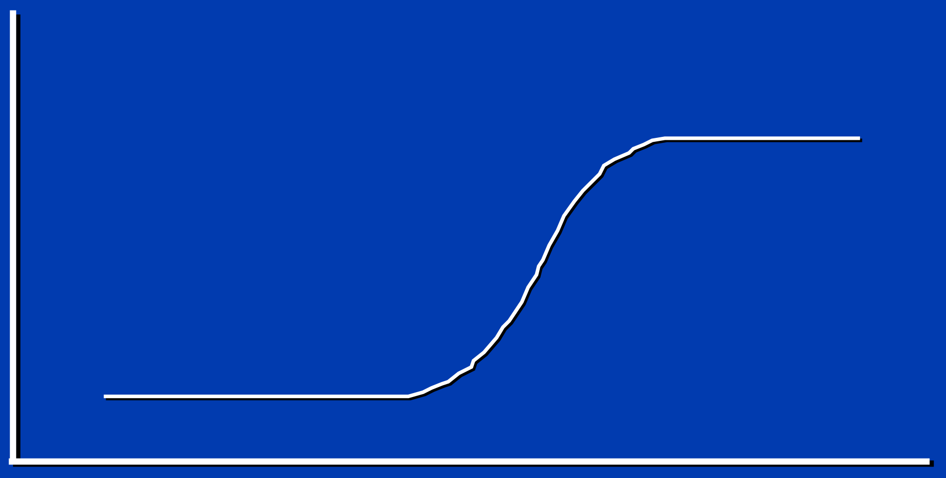


A Soft Collision

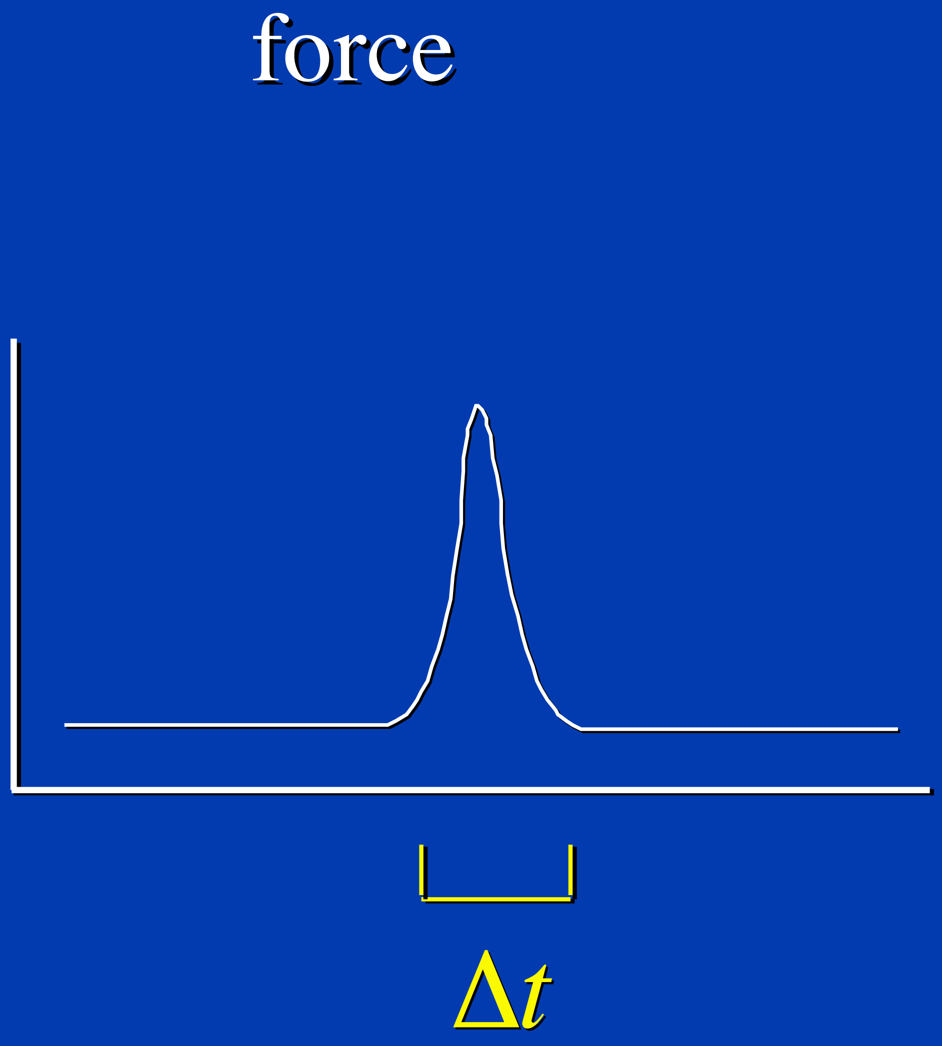
force

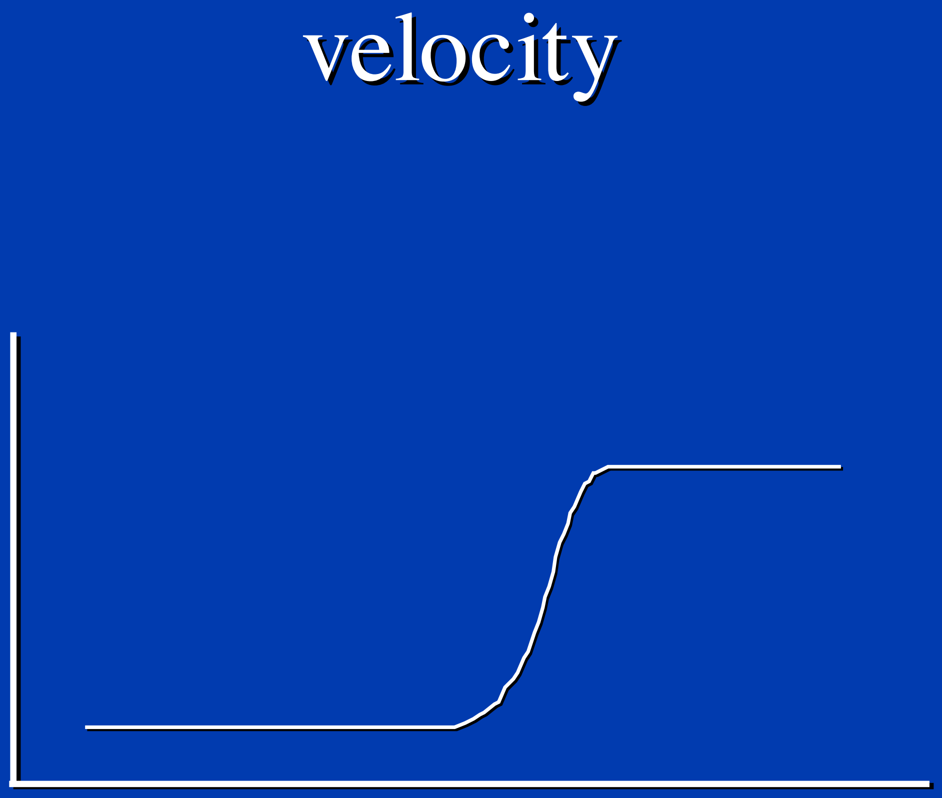


velocity

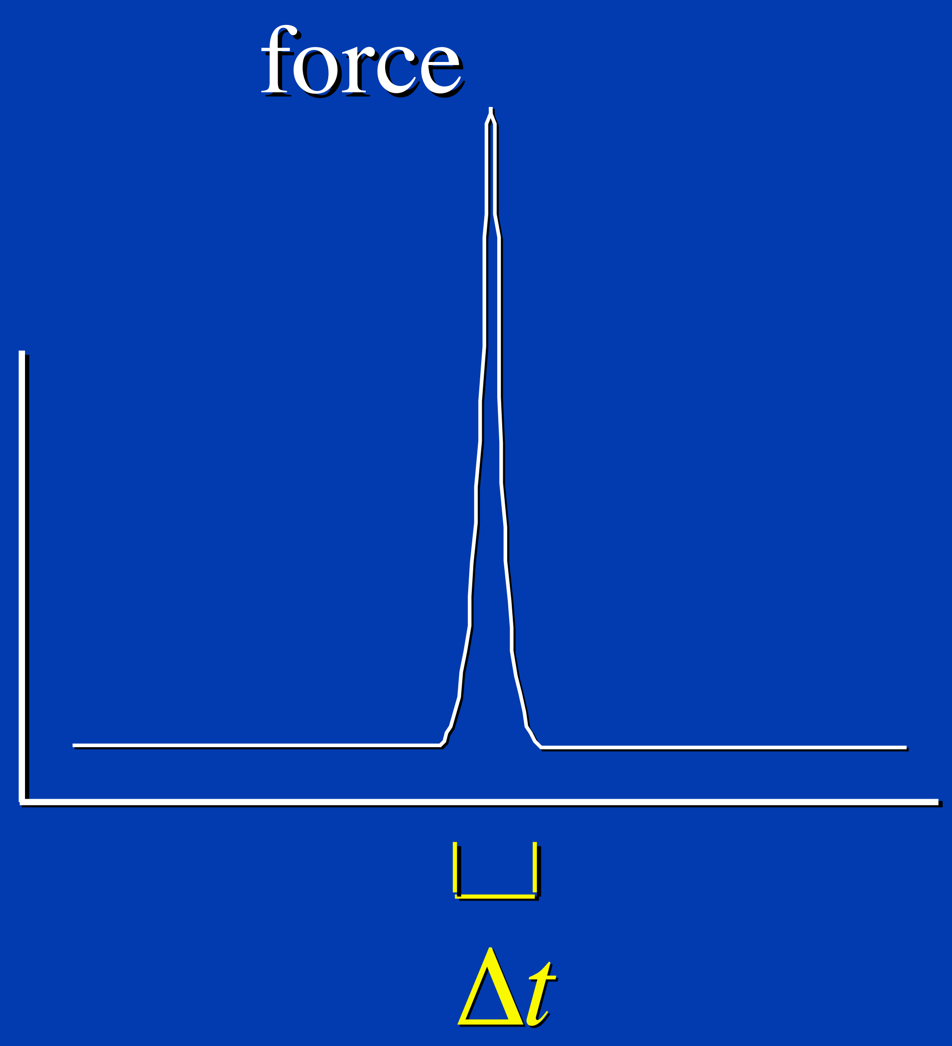


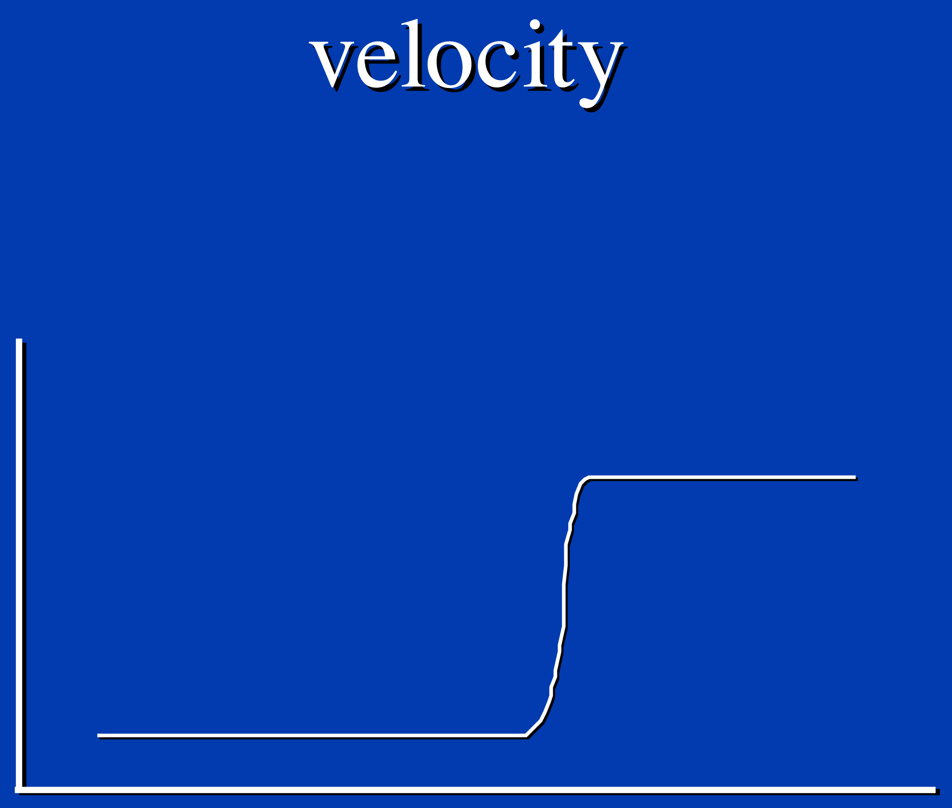
A Harder Collision



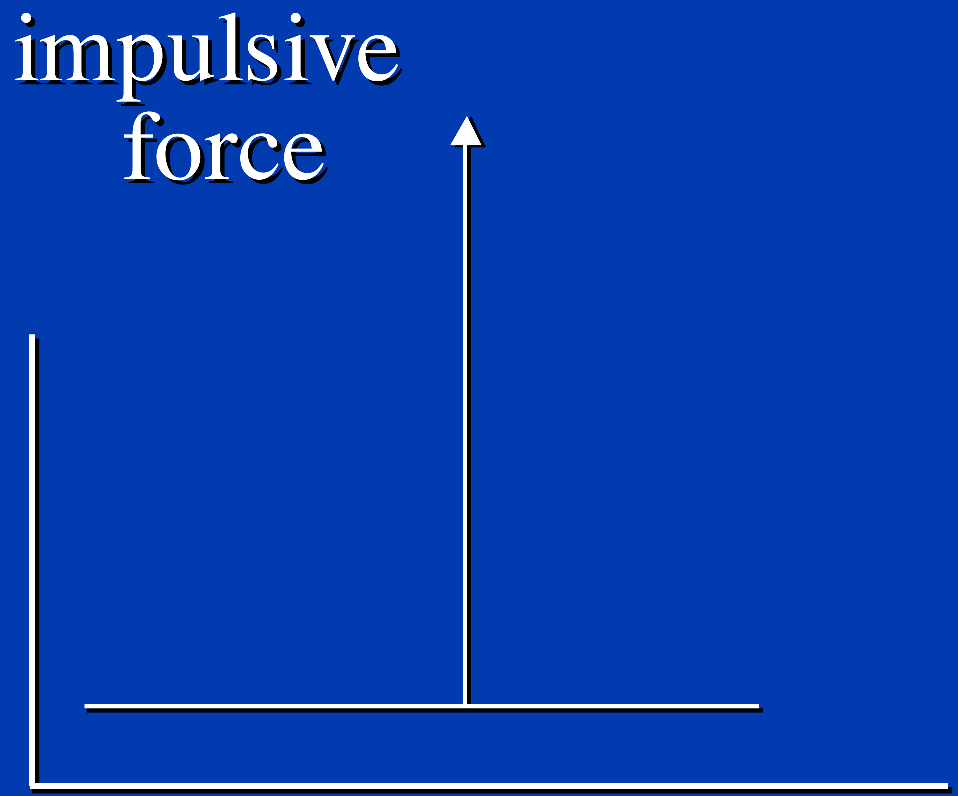


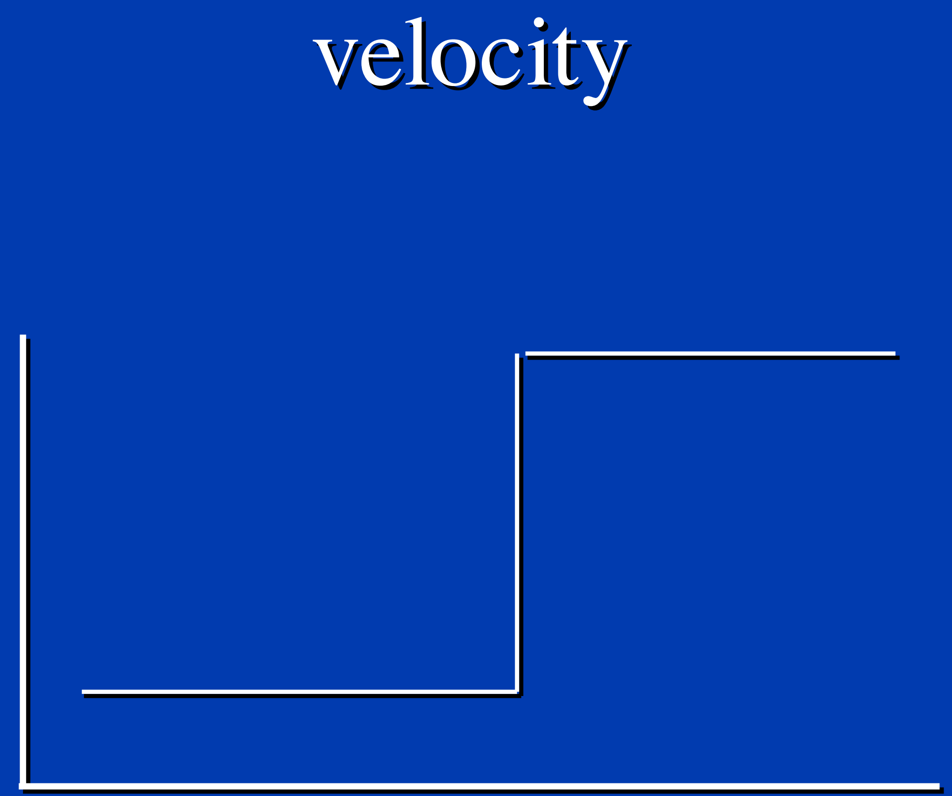
A Very Hard Collision





A Rigid Body Collision

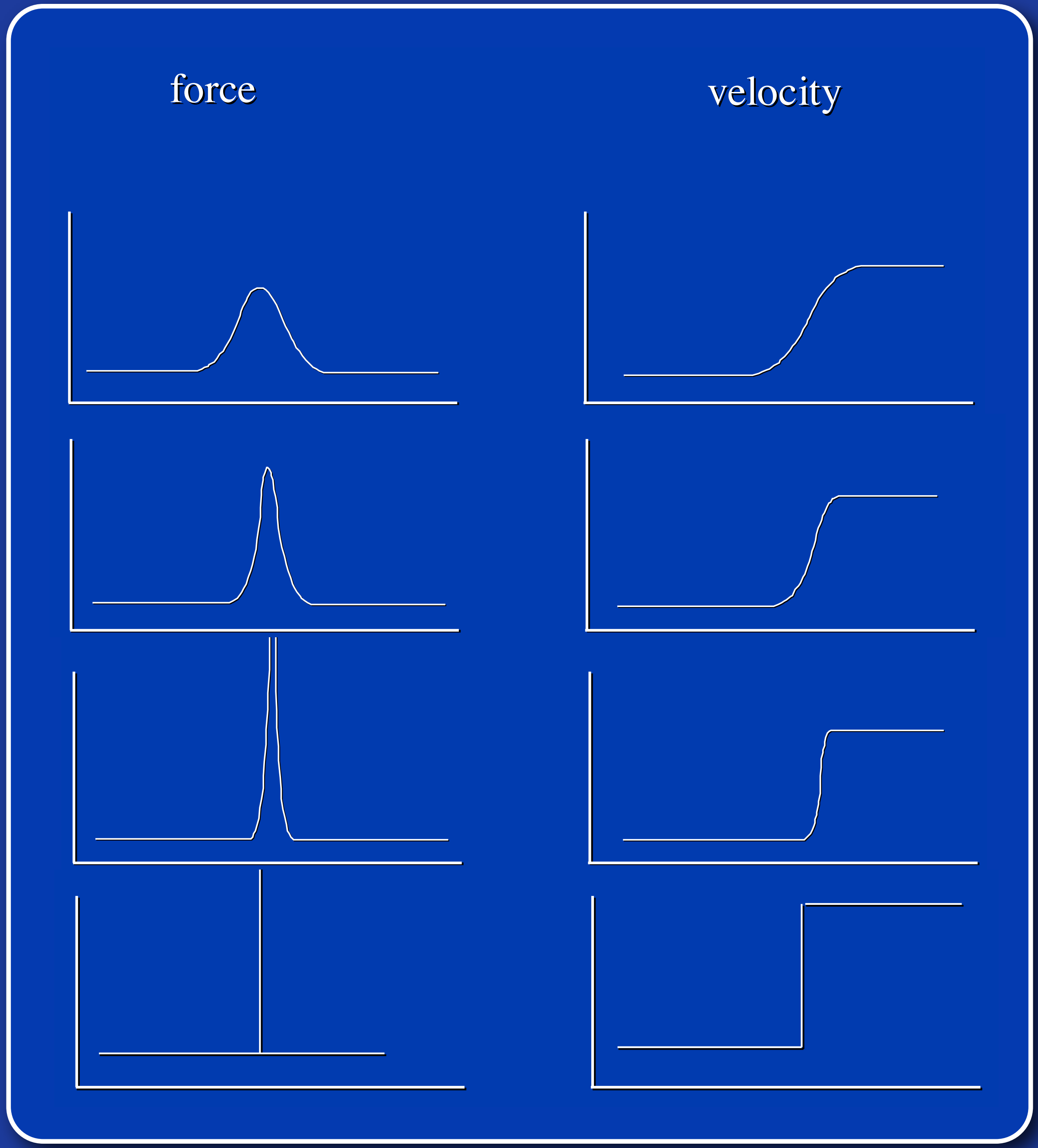




$$f_{imp} = \infty$$

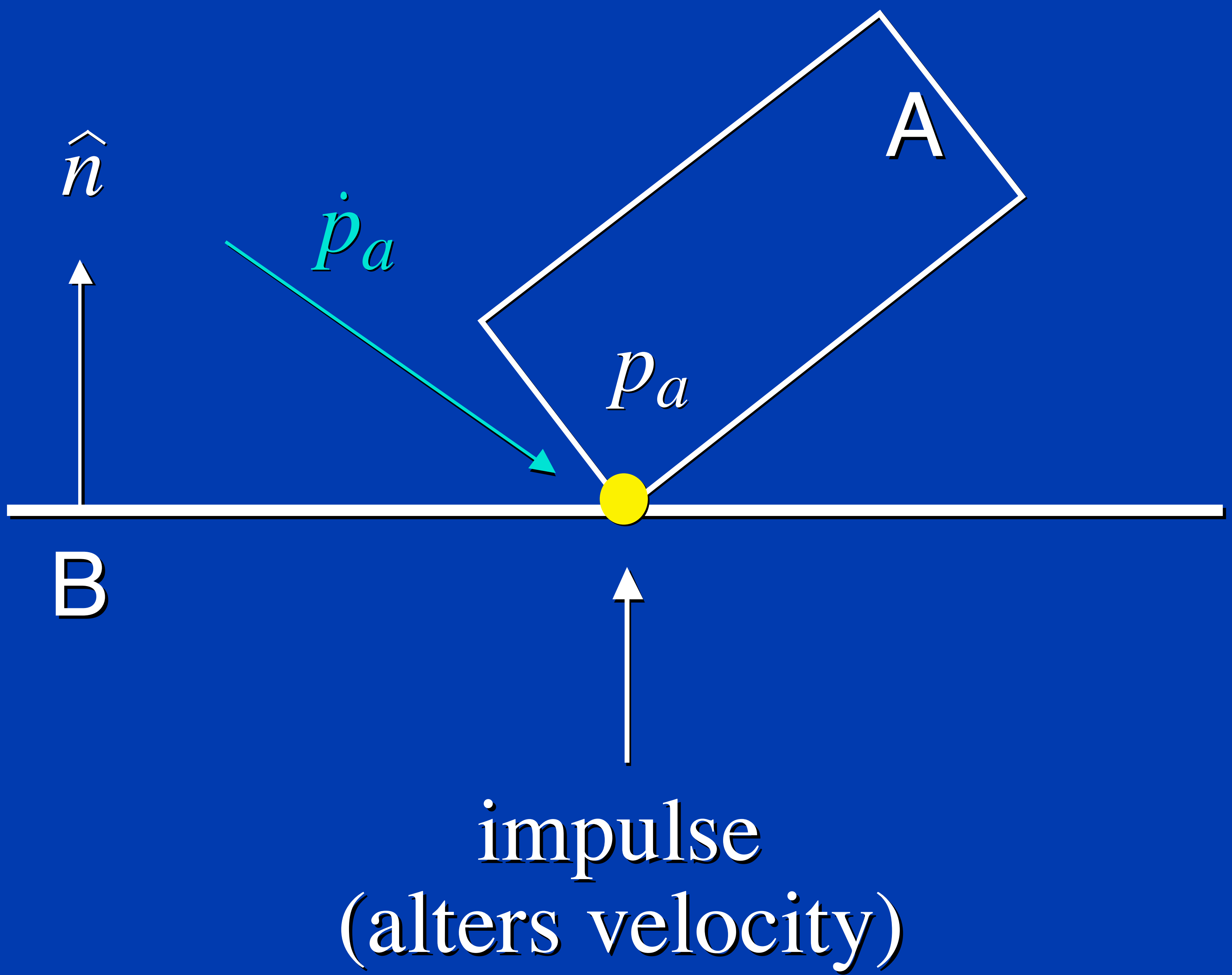
$$\Delta t = 0$$

Notice



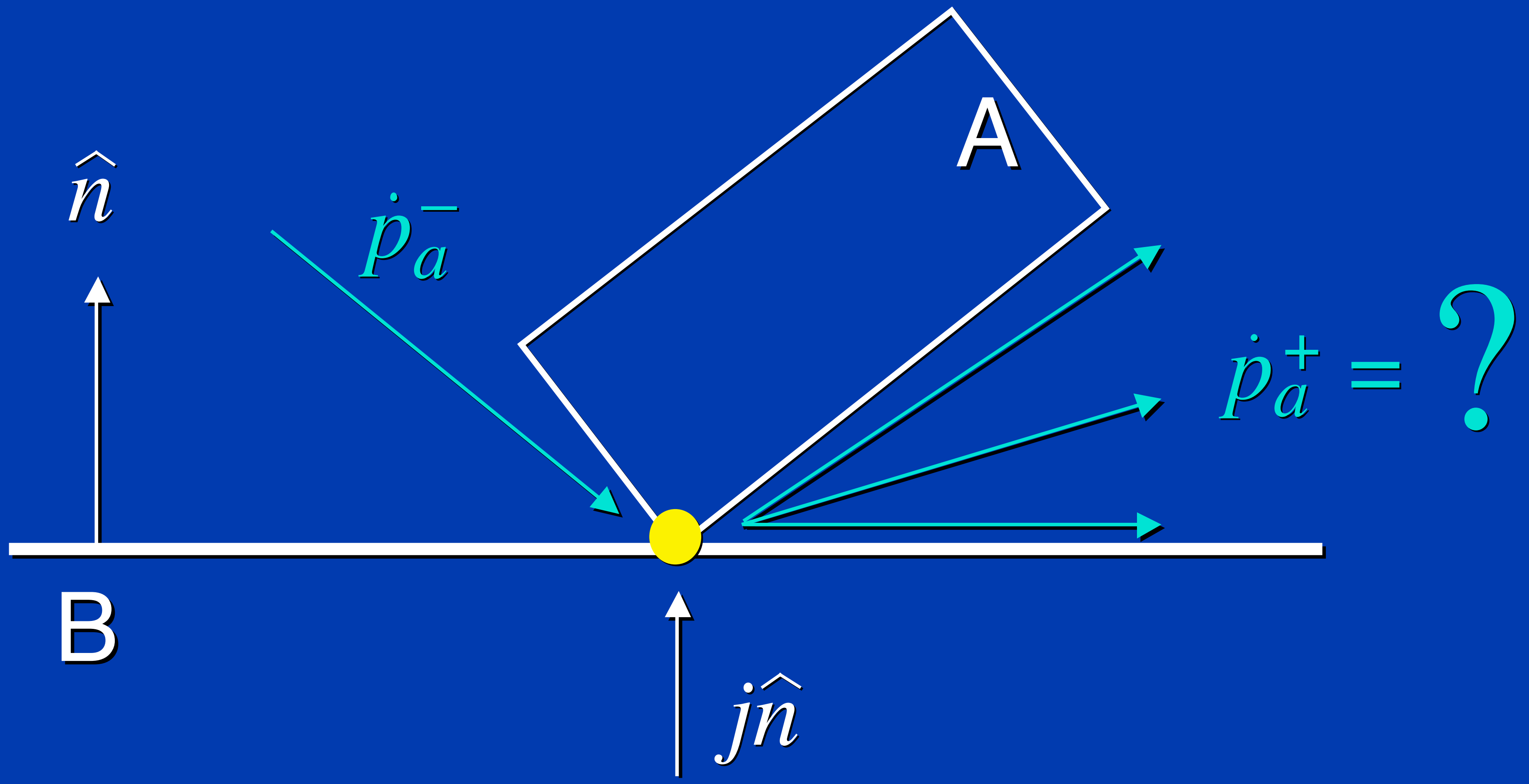
Δv remains constant!

Colliding Contact



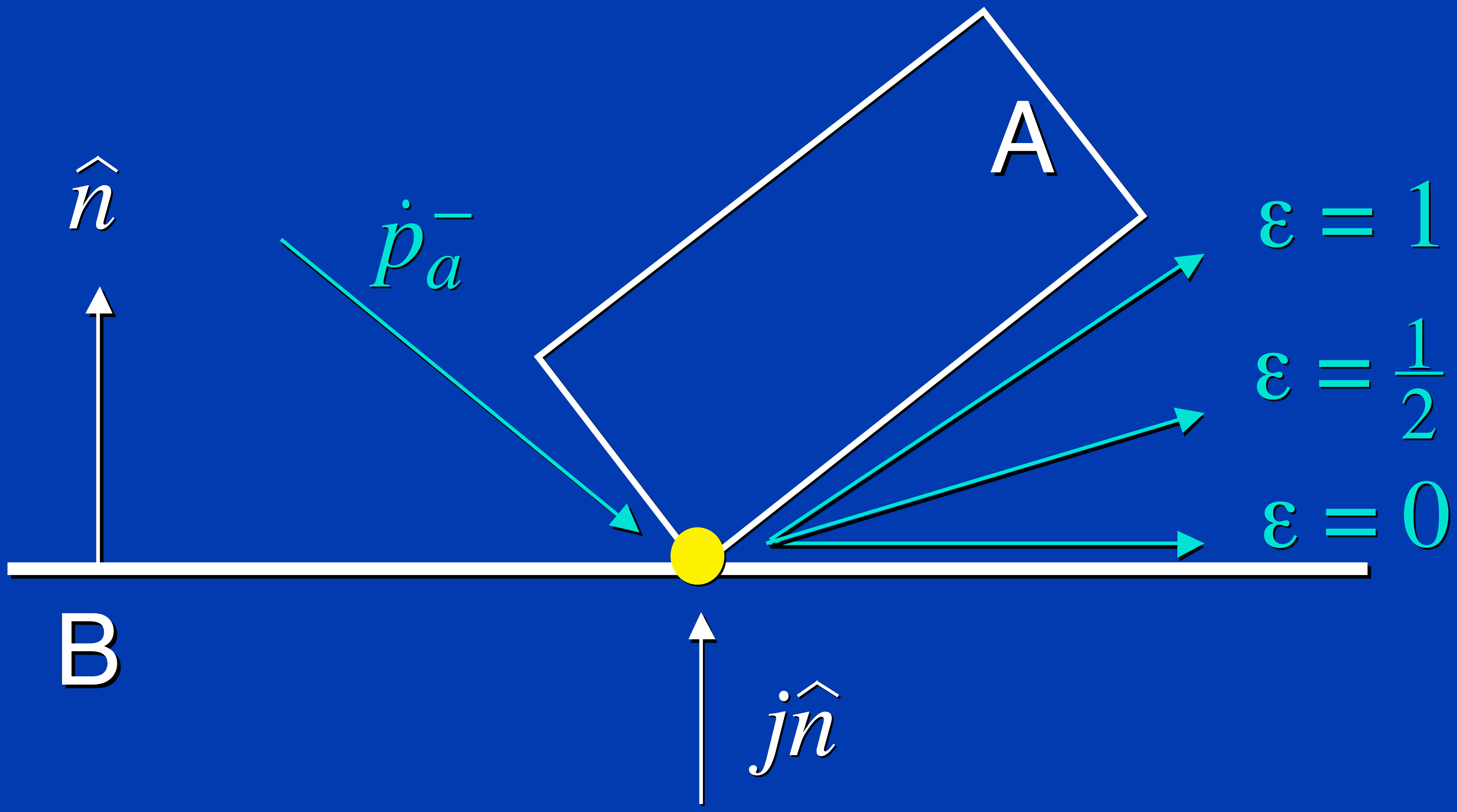
Mathematically...

Computing Impulses



Coefficient of Restitution

$$\hat{n} \bullet \dot{p}_a^+ = -\varepsilon\left(\hat{n} \bullet \dot{p}_a^-\right)$$



Computing j

$$v_a^+(t_0) = v_a^-(t_0) + \frac{j\hat{n}(t_0)}{M_a}$$

$$\omega_a^+(t_0) = \omega_a^-(t_0) + I_a^{-1}\left(r_a \times j\hat{n}(t_0)\right)$$

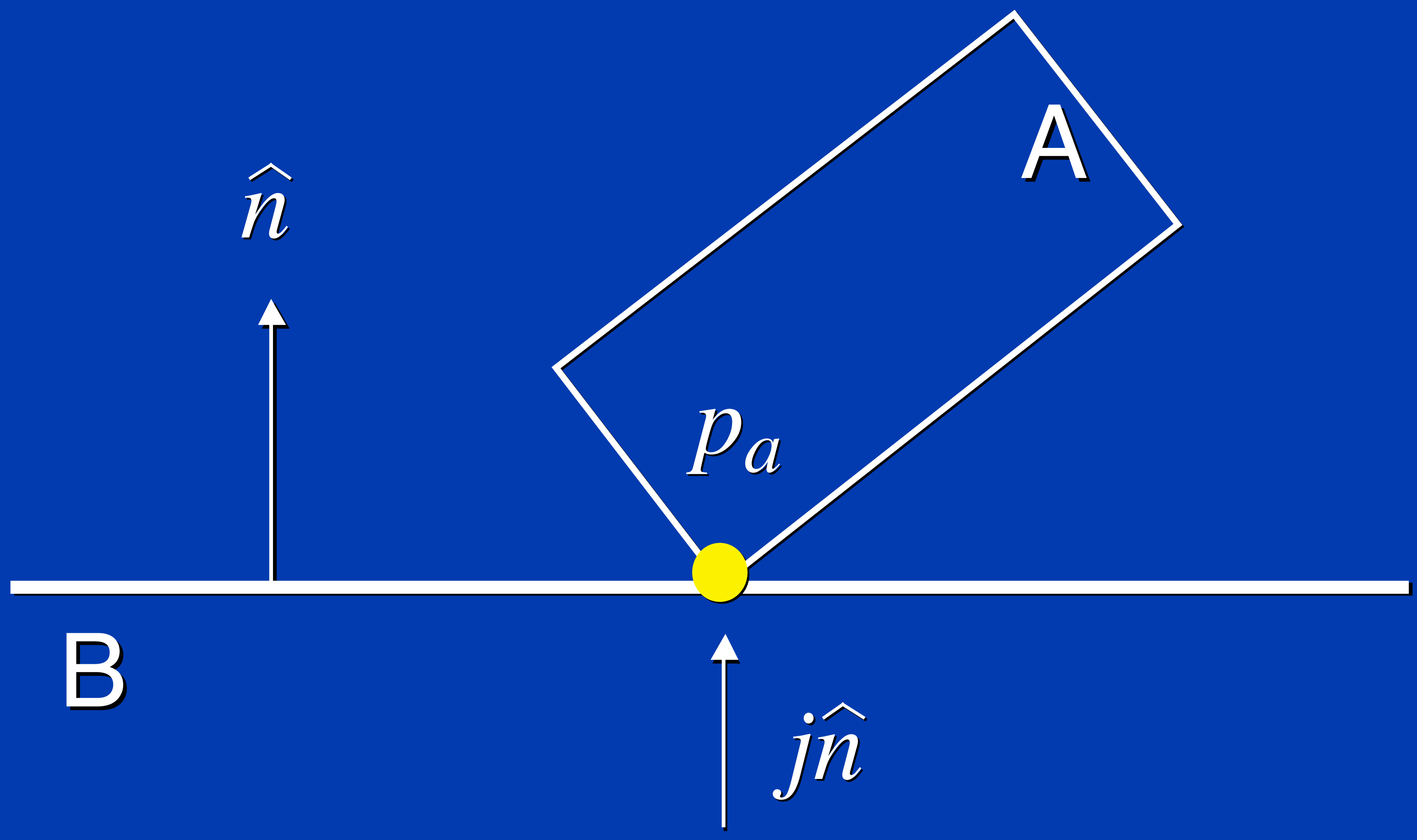
$$\dot{p}_a^+(t_0) = v_a^+(t_0) + \omega_a^+(t_0) \times r_a$$

$$\Downarrow$$

$$\dot{p}_a^+(t_0) = aj + b$$

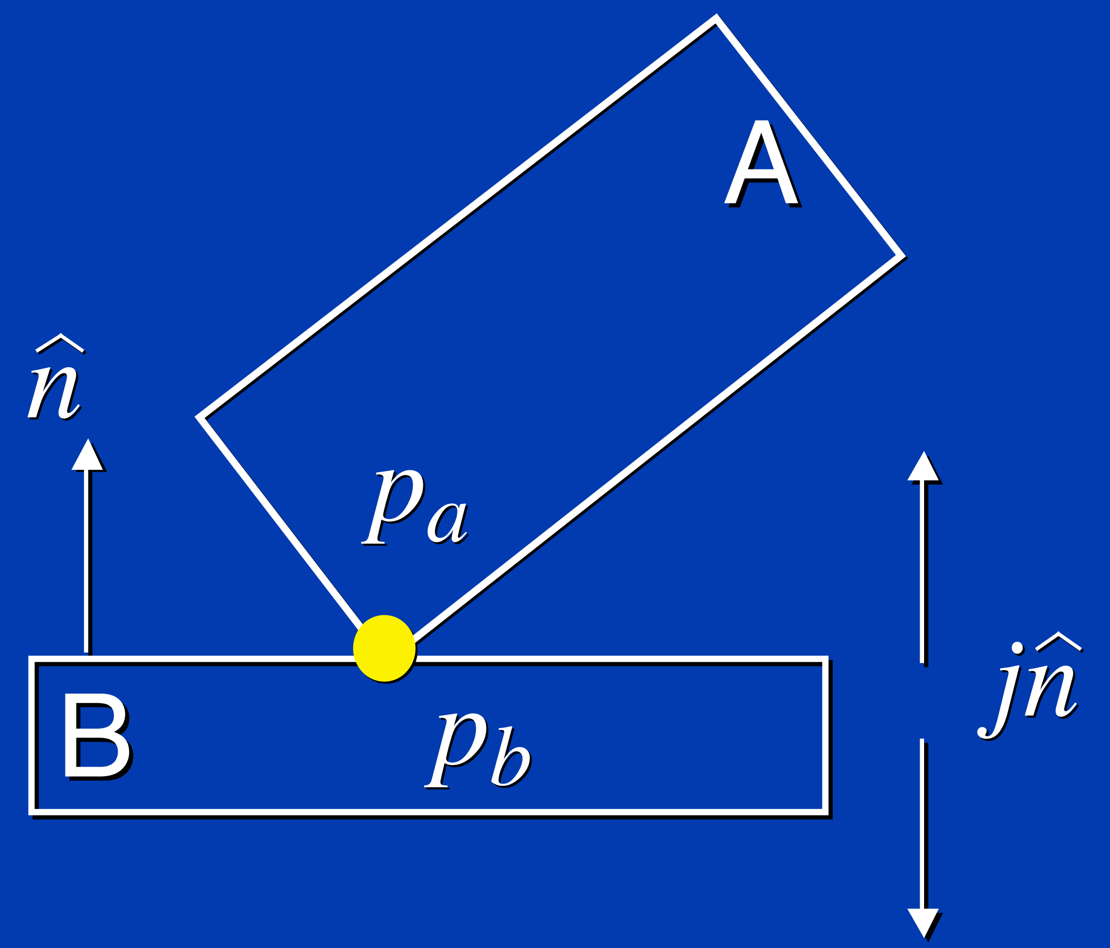
Computing j

$$\hat{n} \bullet \dot{p}_a^{+} = -\varepsilon\left(\hat{n} \bullet \dot{p}_a^{-}\right) \longrightarrow cj + b = d$$



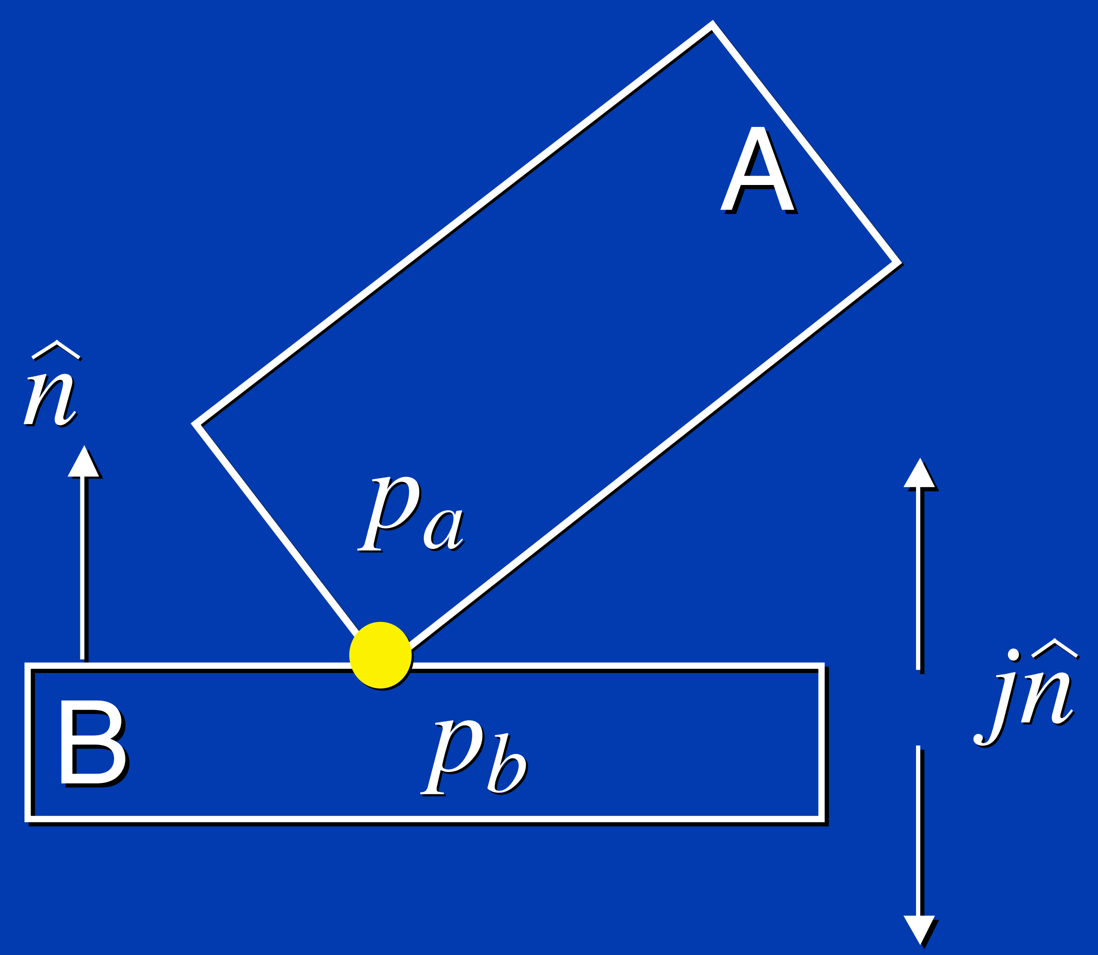
Computing j

$$\hat{n} \bullet (\dot{p}_a^+ - \dot{p}_b^+) = -\varepsilon\left(\hat{n} \bullet (\dot{p}_a^- - \dot{p}_b^-)\right)$$



Computing j

$$\hat{n} \bullet (\dot{p}_a^+ - \dot{p}_b^+) = -\varepsilon\left(\hat{n} \bullet (\dot{p}_a^- - \dot{p}_b^-)\right) \longrightarrow cj + b = d$$



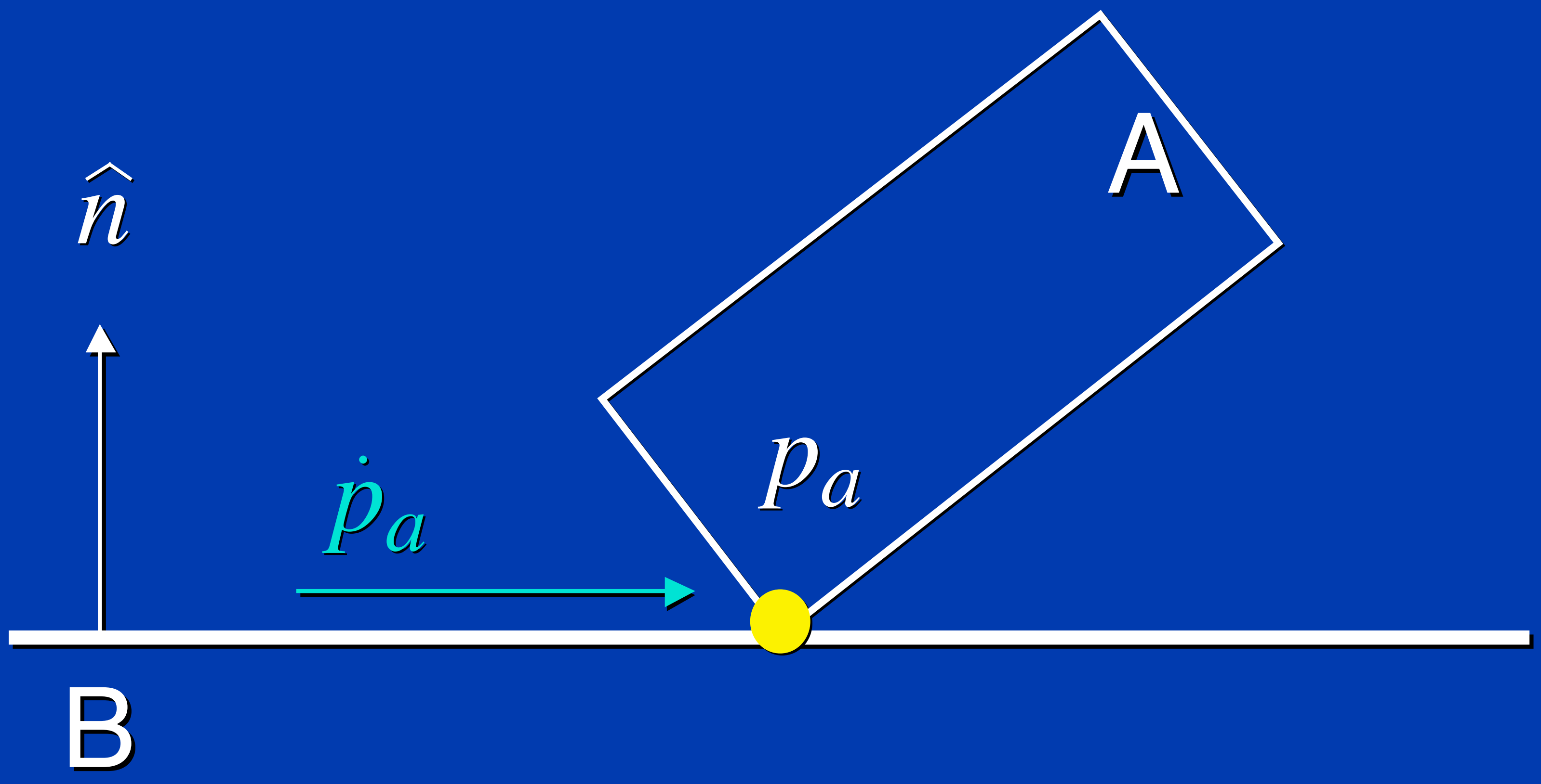
Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces

Problem

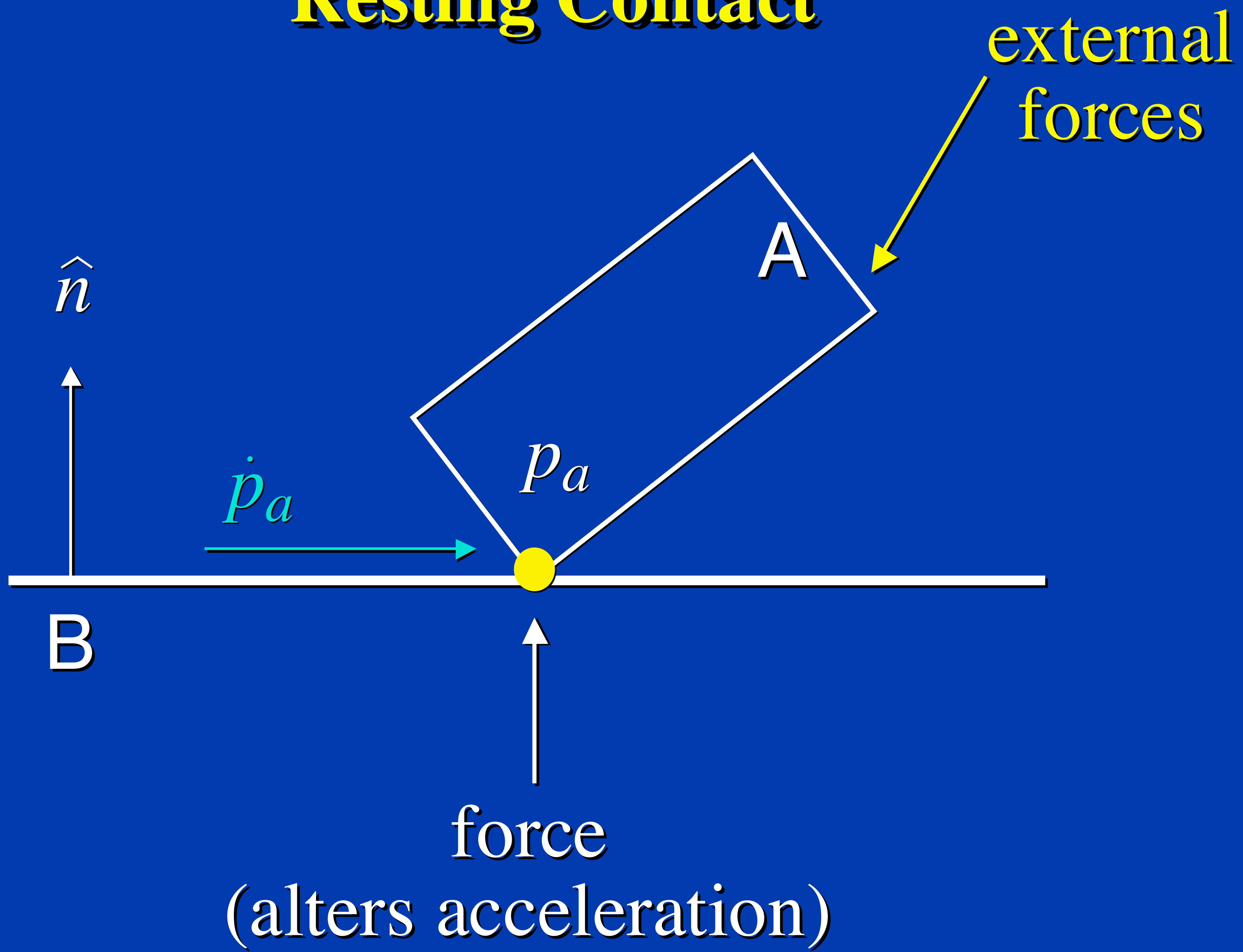
- Positions OK
- Velocities OK
- Accelerations NOT OK

Resting Contact

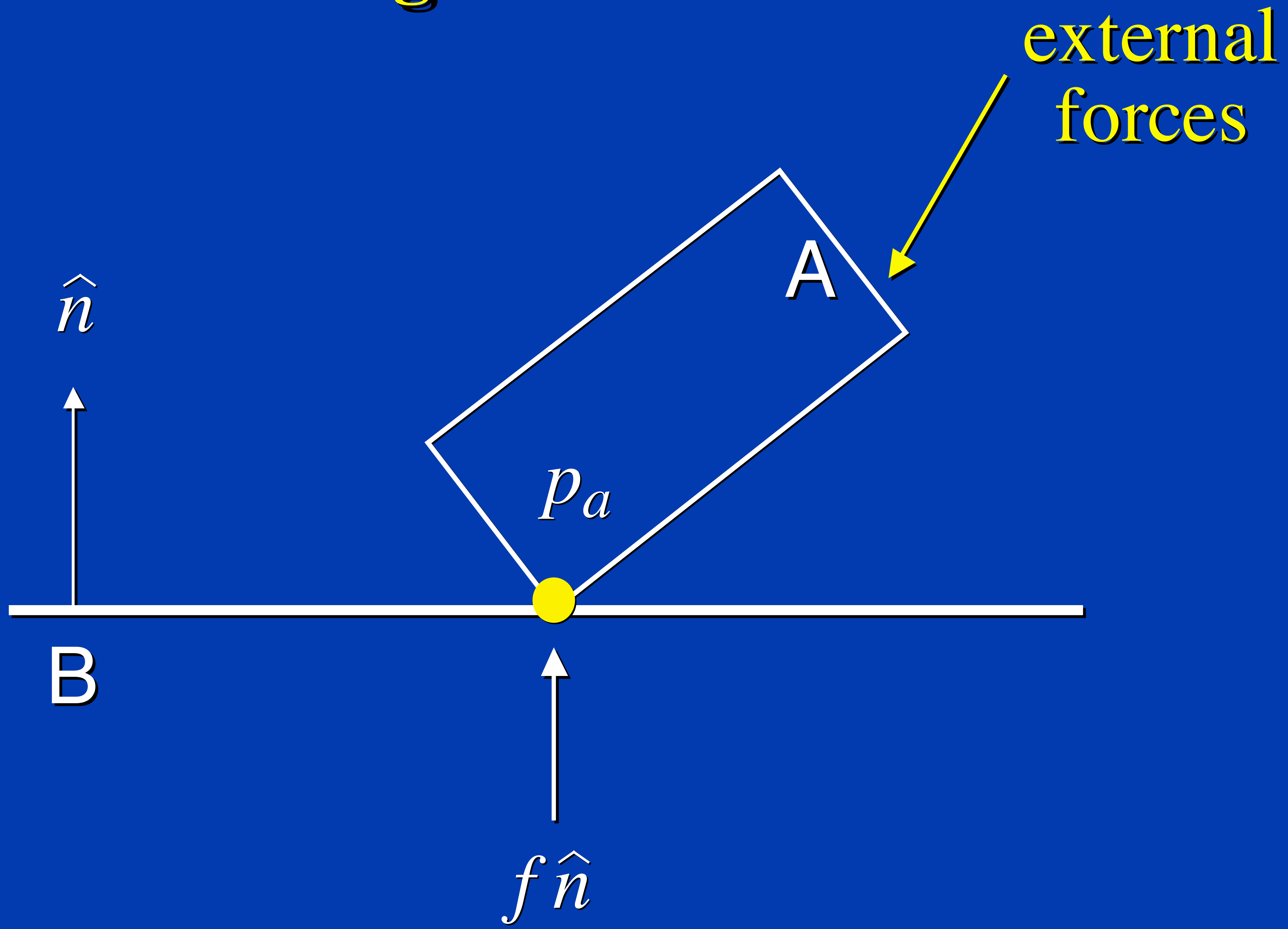


$$\hat{n} \bullet \dot{p}_a = 0$$

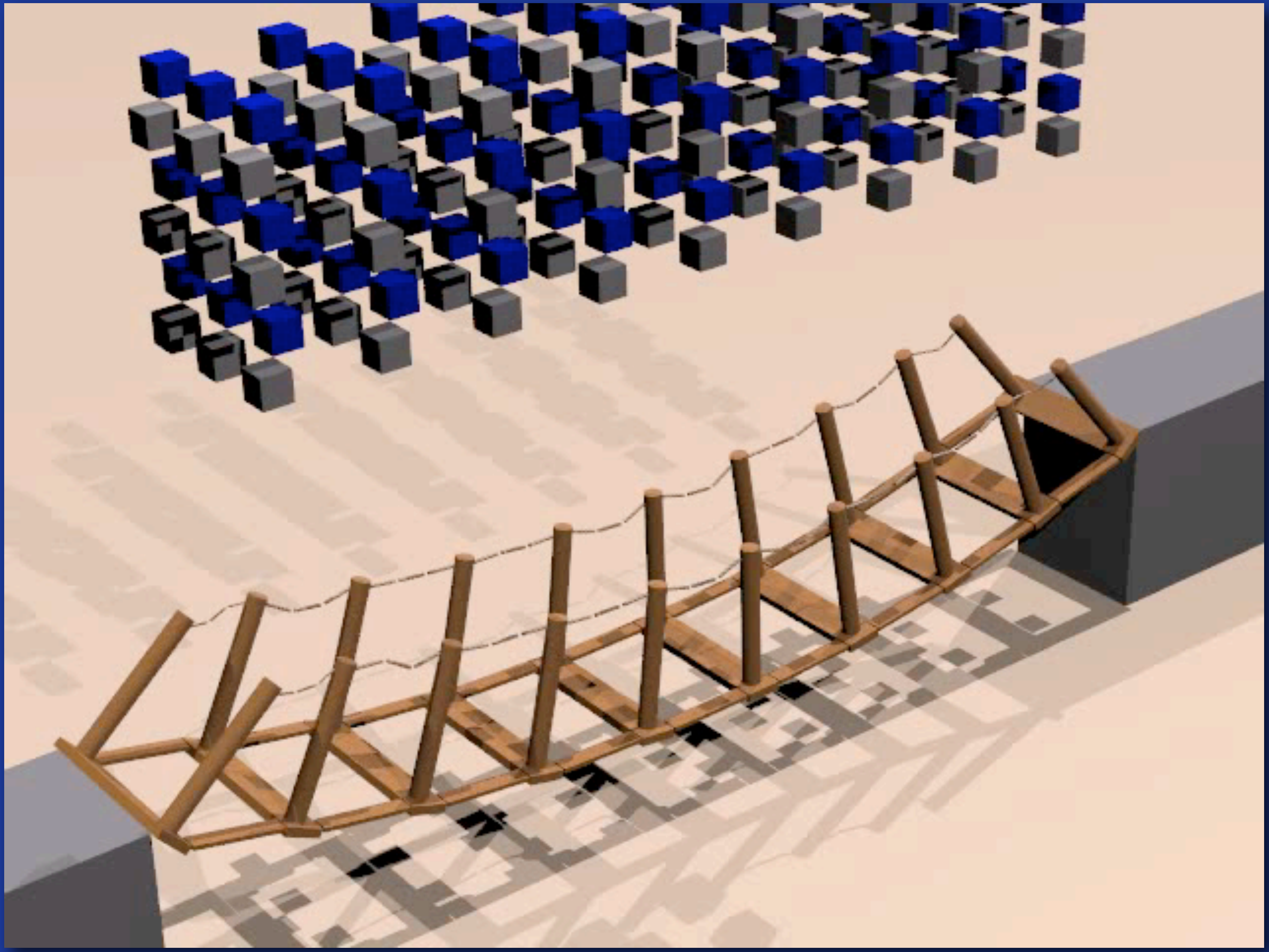
Resting Contact



Resting Contact Forces

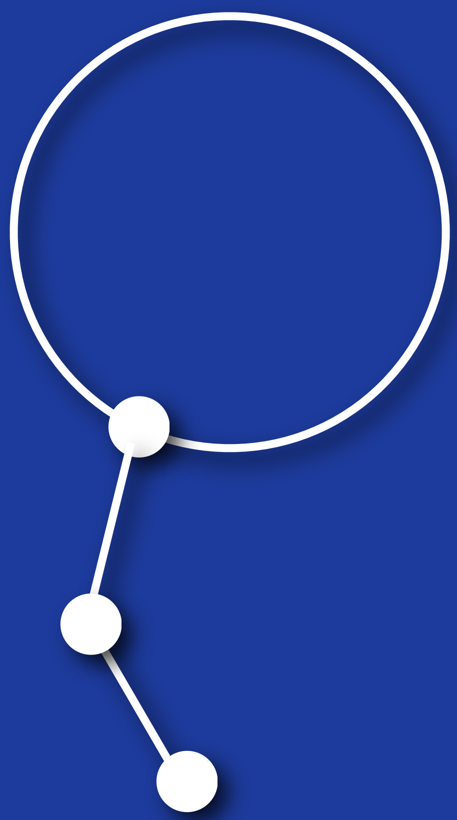


Example



Solution Outline

- Similar to constraints before, we will compute **constraint forces**.
- Except...
  - There will be **inequalities**.
  - There will be **quadratic terms**.



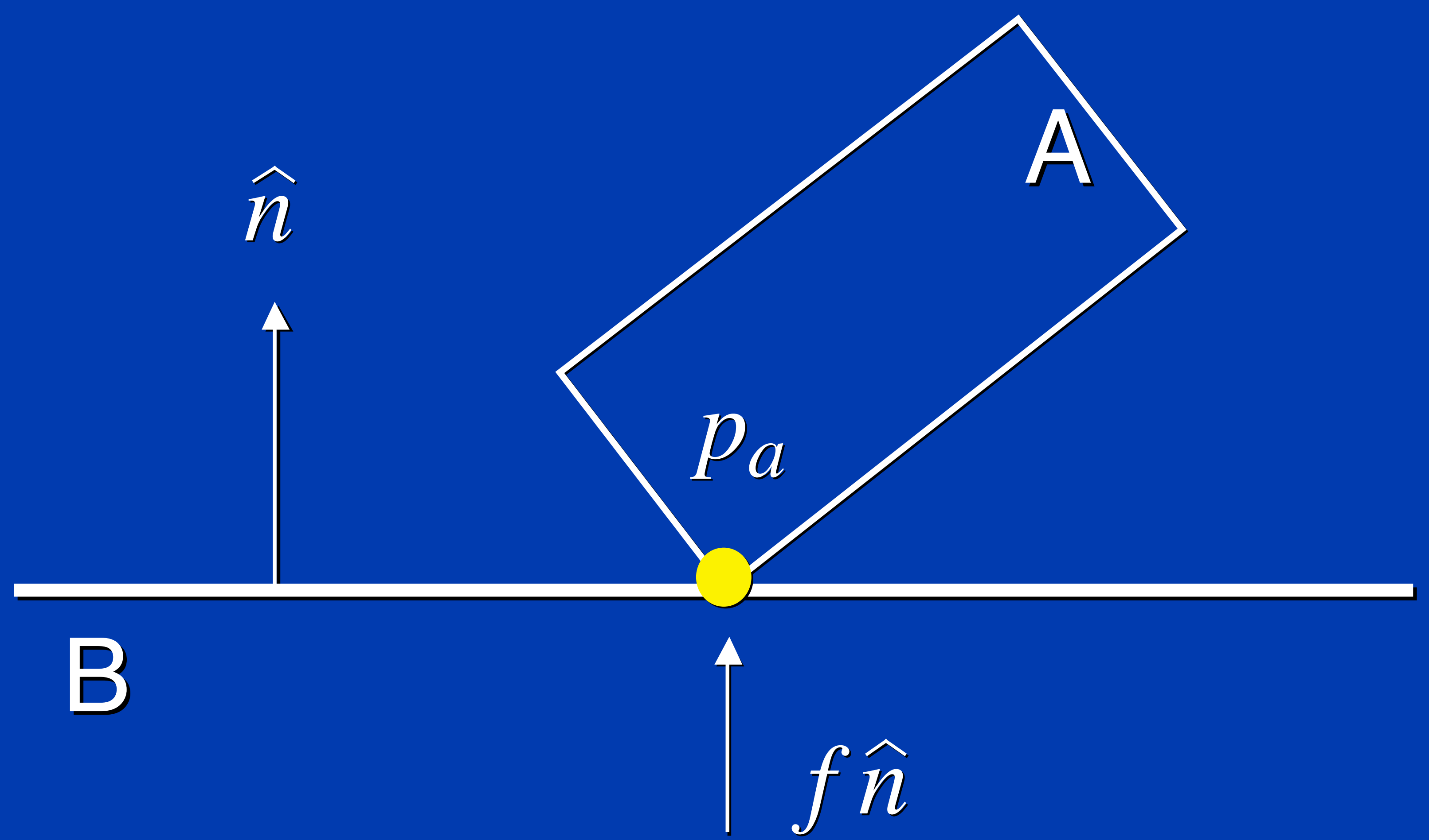
Conditions on the Constraint Force

To avoid inter-penetration, the force strength f must prevent the vertex p_a from accelerating downwards. If B is fixed, this is written as

$$\hat{n} \bullet \ddot{p}_a \geq 0$$

Computing f

$$\hat{n} \bullet \ddot{p}_a \geq 0 \longrightarrow af + b \geq 0$$



Conditions on the Constraint Force

To prevent the constraint force from holding bodies together, the force must be repulsive:

$$f \geq 0$$

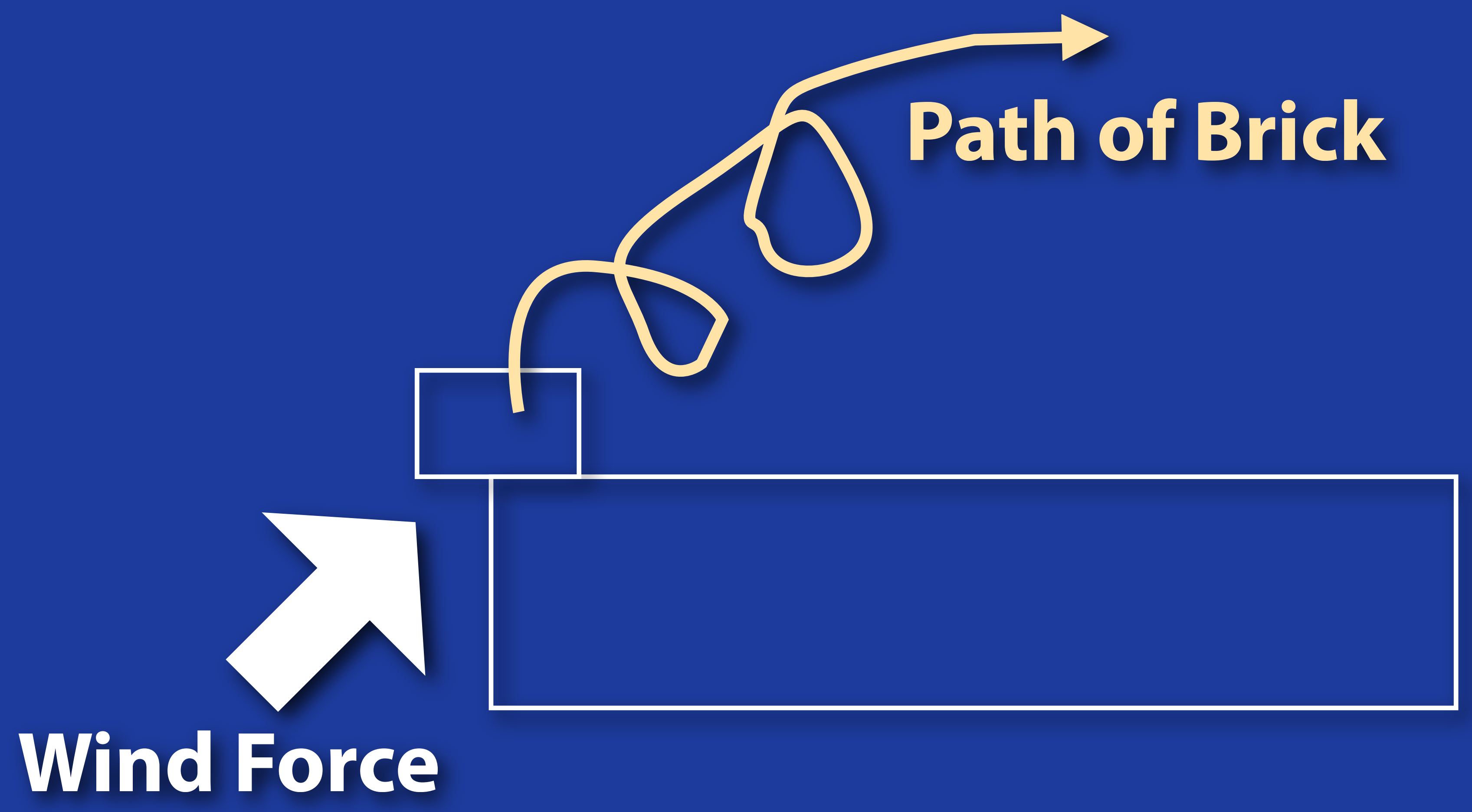
Does the above, along with

$$\hat{n} \bullet \ddot{p}_a \geq 0 \longrightarrow af + b \geq 0$$

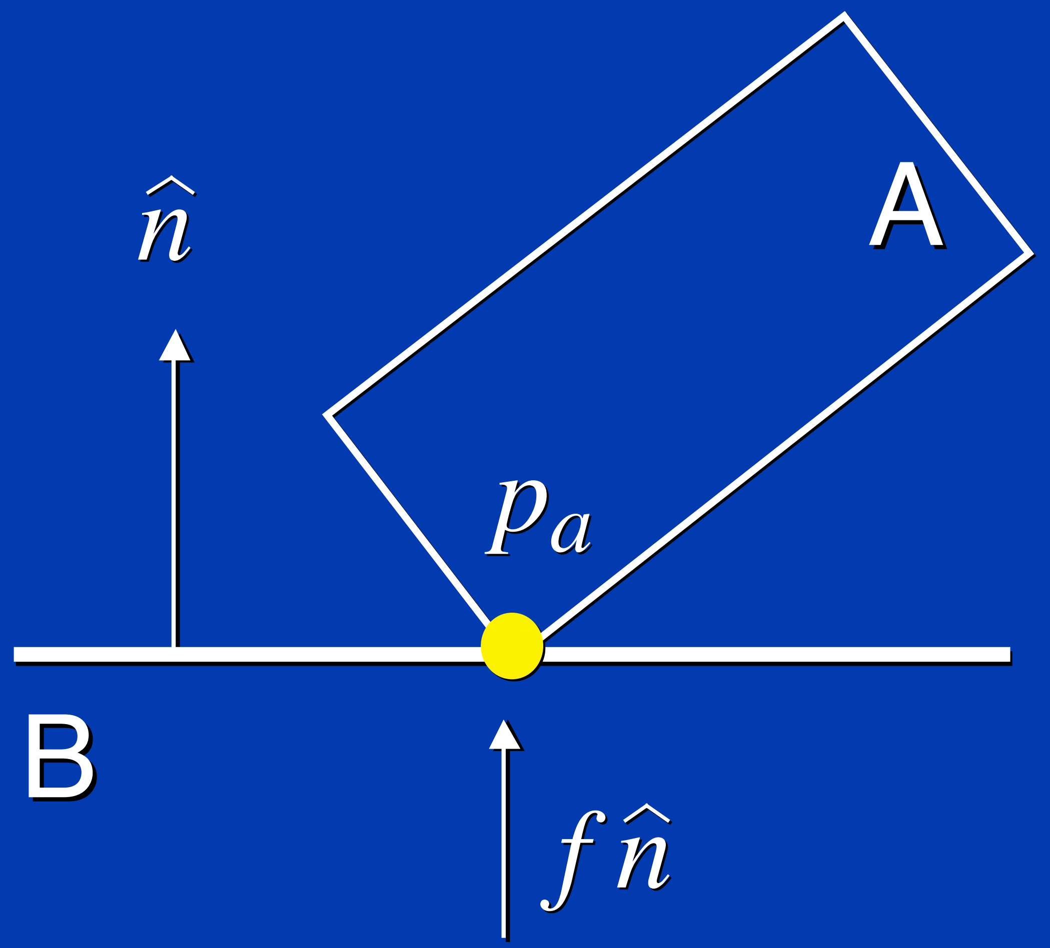
sufficiently constrain f?

3rd Constraint

- We require that the force at a contact point become zero if the bodies begin to separate.



Workless Constraint Force



Either

$$af + b = 0$$
$$f \geq 0$$

or

$$af + b > 0$$
$$f = 0$$

Conditions on the Constraint Force

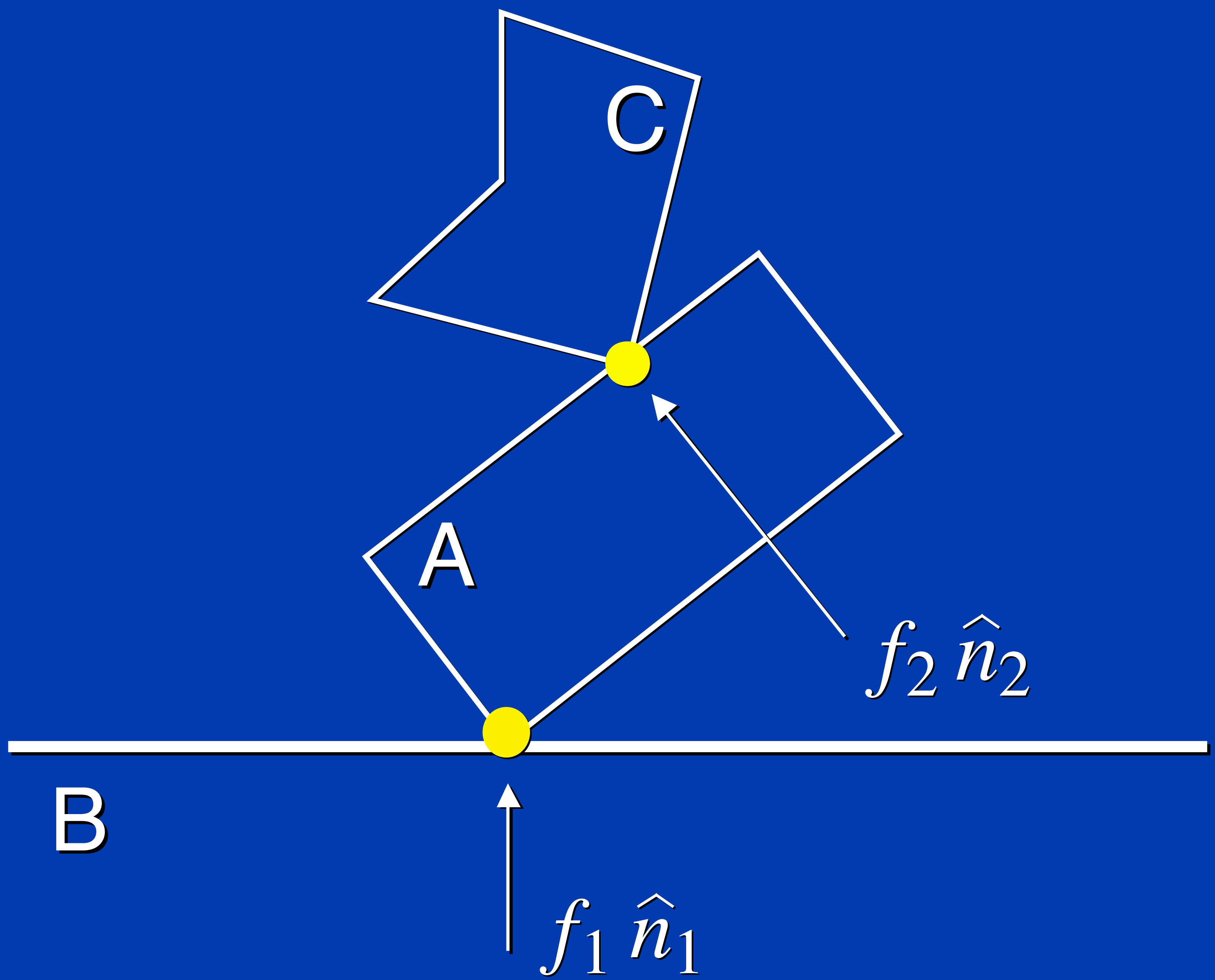
To make f be workless, we use the condition

$$f \cdot (af + b) = 0$$

The full set of conditions is

$$\begin{aligned} af + b &\geq 0 \\ f &\geq 0 \\ f \cdot (af + b) &= 0 \end{aligned}$$

Multiple Contact Points



Conditions on f_1

Non-penetration:

$$a_{11} f_1 + a_{12} f_2 + b_1 \geq 0$$

Repulsive:

$$f_1 \geq 0$$

Workless:

$$f_1 \cdot (a_{11} f_1 + a_{12} f_2 + b_1) = 0$$

Quadratic Program for f_1 and f_2

Non-penetration:

$$a_{11}f_1 + a_{12}f_2 + b_1 \geq 0$$
$$a_{21}f_1 + a_{22}f_2 + b_2 \geq 0$$

Repulsive:

$$f_1 \geq 0$$
$$f_2 \geq 0$$

Workless:

$$f_1 \cdot (a_{11}f_1 + a_{12}f_2 + b_1) = 0$$
$$f_2 \cdot (a_{21}f_1 + a_{22}f_2 + b_2) = 0$$

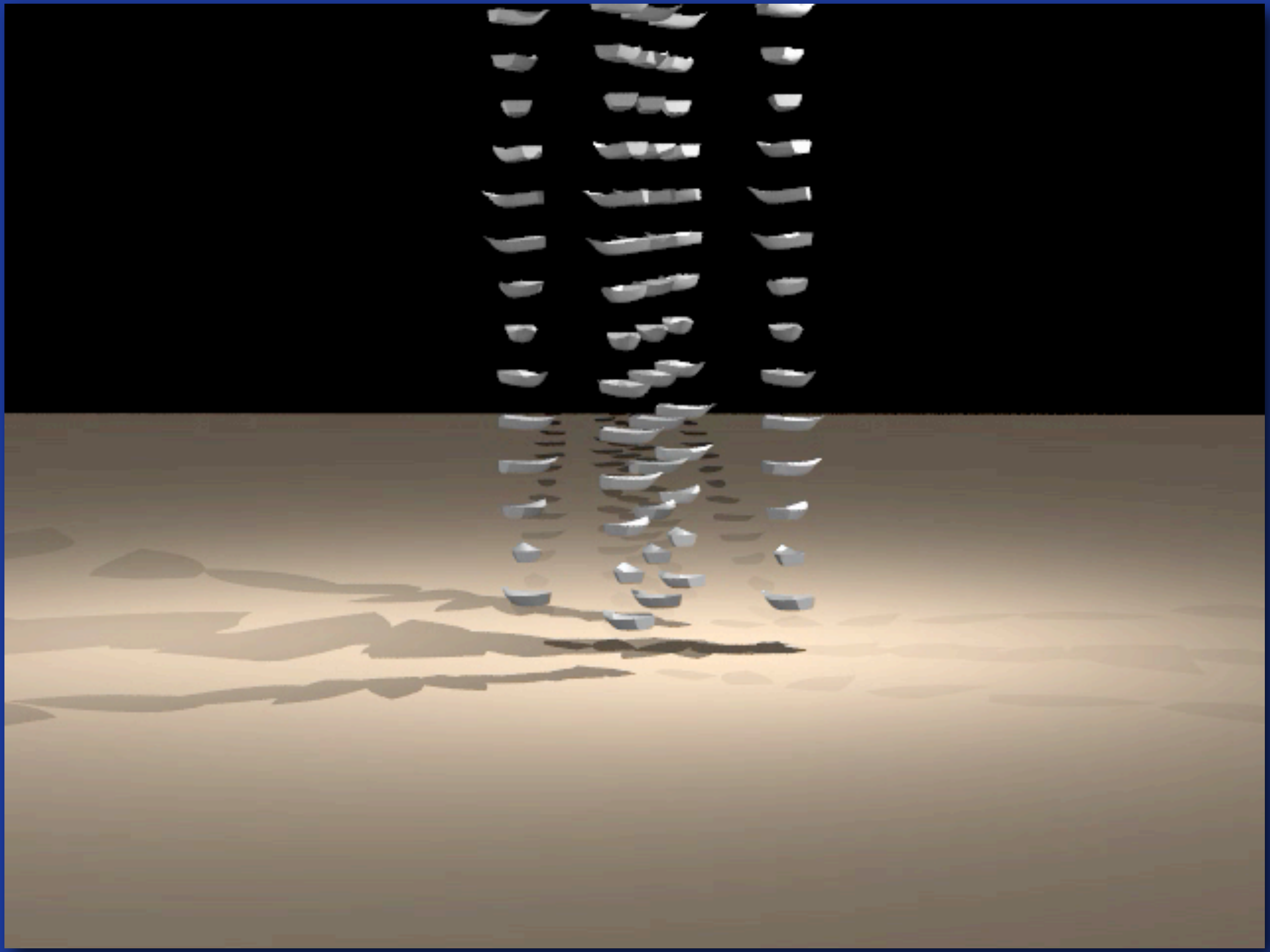
In the Notes – Constraint Forces

Derivations of the non-penetration constraints for contacting polyhedra.

Derivations and code for computing the a_{ij} and b_i coefficients.

Code for computing and applying the constraint forces $f_i \hat{n}_i$.

Example



Example

Question

- What type of discrete geometric representation should we use for a deformable object?
- What sort of forces apply to deformable objects, i.e. in what ways do they resist deformation?
- How can we compute these forces?