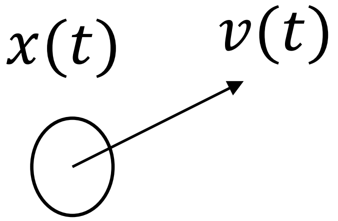
Rigid Bodies and Contacts

Yanzhe Yang
Feb 26, 2020

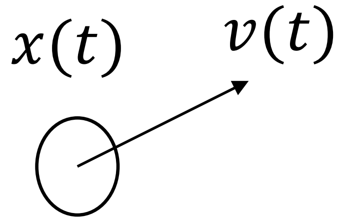
Rigid bodies and contacts

- Particle: State $Y = \binom{x}{v}$, State Derivative $\frac{d}{dt}Y = \frac{d}{dt}\binom{x}{v} = \binom{v}{F/m}$

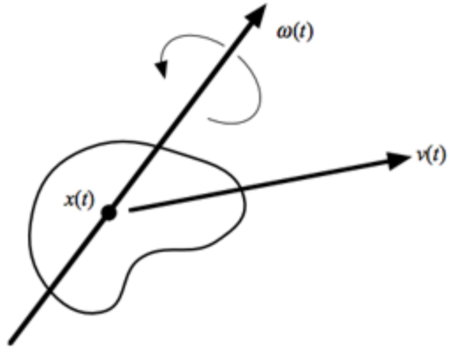


Rigid bodies and contacts

- Particle: State $Y = \binom{x}{v}$, State Derivative $\frac{d}{dt}Y = \frac{d}{dt}\binom{x}{v} = \binom{v}{F/m}$



- Rigid body: State $Y = \begin{pmatrix} x \\ R \\ mv \\ Iw \end{pmatrix}$, State Derivative $\frac{d}{dt}Y = \frac{d}{dt}\begin{pmatrix} x \\ R \\ mv \\ Iw \end{pmatrix} = \begin{pmatrix} v \\ w \times R \\ F \\ \tau \end{pmatrix}$



R:	rotation matrix
mv:	linear momentum
Iw:	angular momentum

Rigid bodies and contacts

Image source:
https://noobtuts.com/unity/2d-angry-birds-game
https://giphy.com/explore/walking-in-the-snow

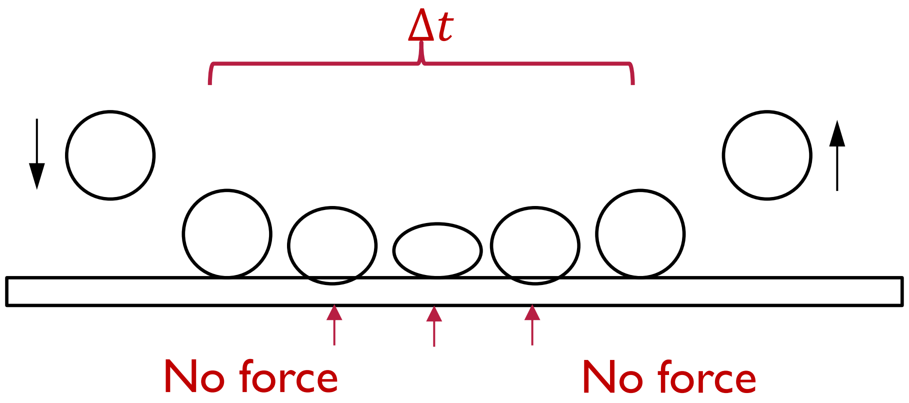
Today's topic:

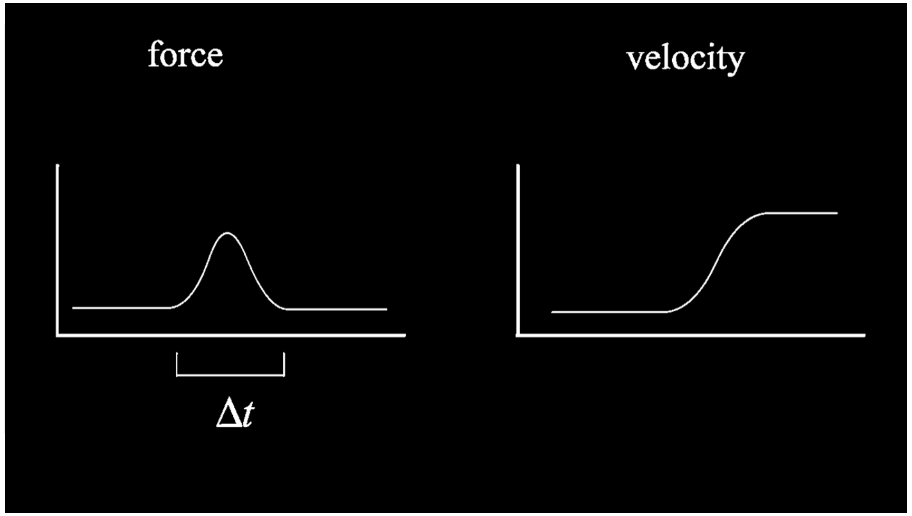
How to simulate contacts for rigid bodies?

Example: ball falling

Reference: Andrew Witkin and David Baraff, Physically Based Modeling: Principles and Practice, Online SIGGRAPH 1997 Course Notes, 1997
https://www.cs.cmu.edu/~baraff/sigcourse/

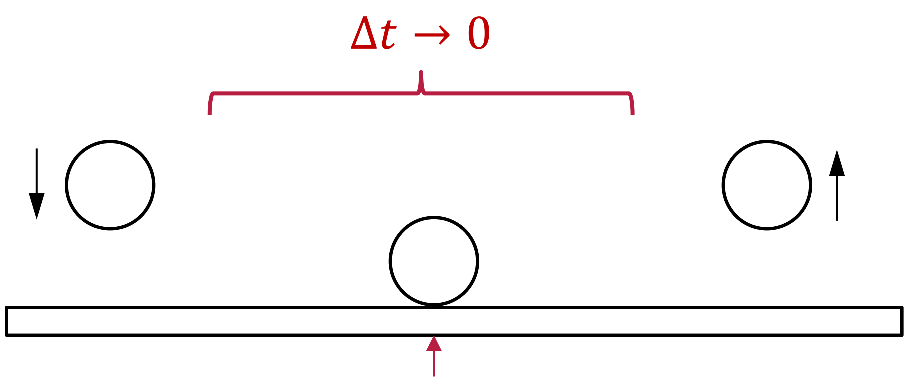
Soft collision

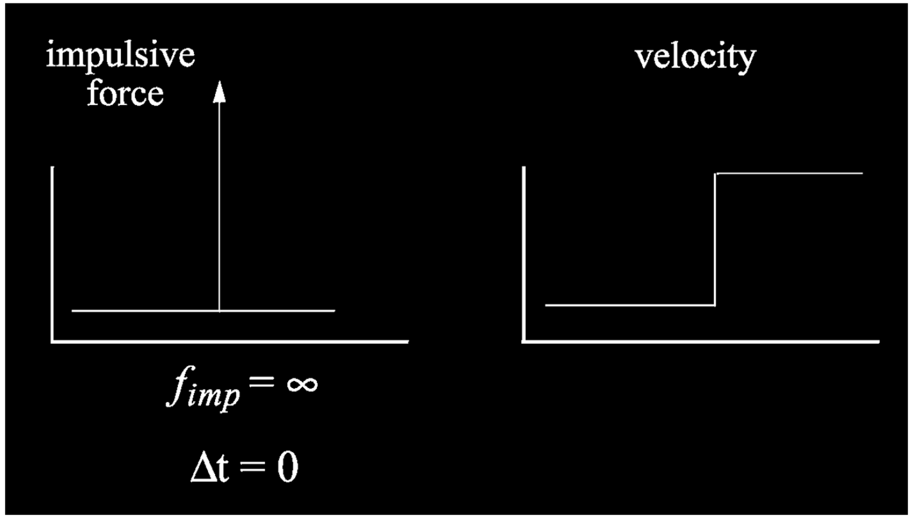




Reference: Andrew Witkin and David Baraff, Physically Based Modeling: Principles and Practice, Online SIGGRAPH 1997 Course Notes, 1997
https://www.cs.cmu.edu/~baraff/sigcourse/

Rigid body collision

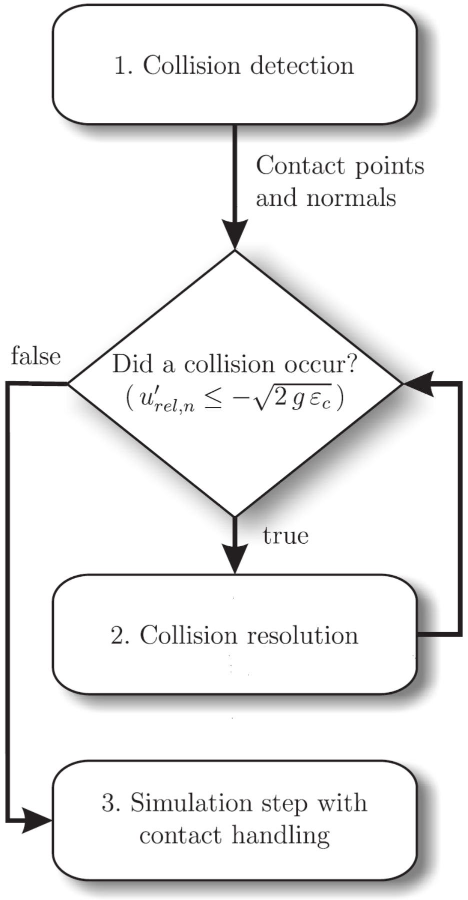




Discontinuity

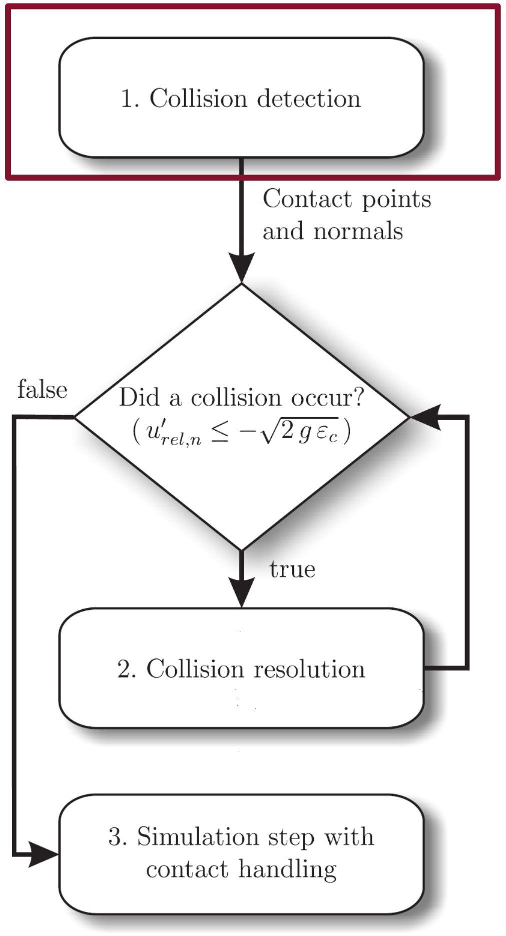
Reference: Andrew Witkin and David Baraff, Physically Based Modeling: Principles and Practice, Online SIGGRAPH 1997 Course Notes, 1997
https://www.cs.cmu.edu/~baraff/sigcourse/

A typical simulation loop



Reference: J. Bender and A. Schmitt. Constraint-based collision and contact handling using impulses. In Proceddings of the 19th international conference on computer animation and social agents. July 2006

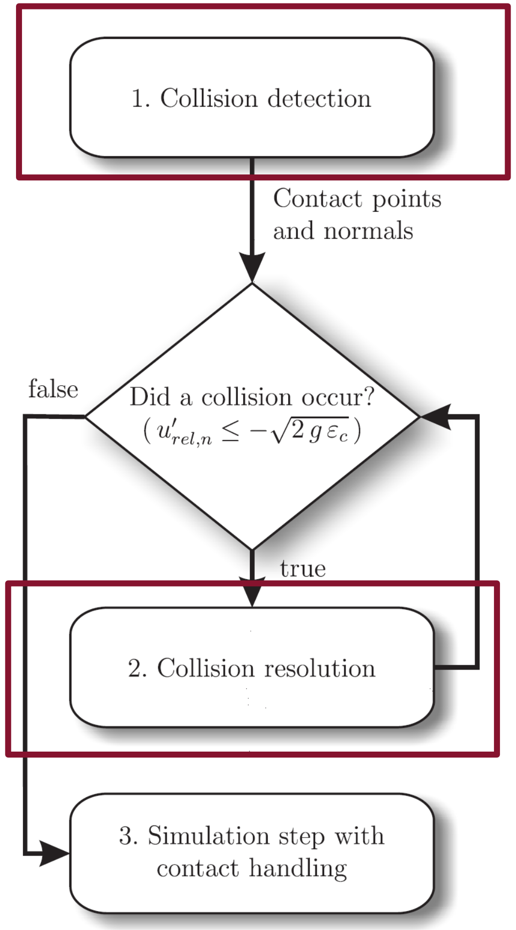
Questions



When and where collision happens?

Reference: J. Bender and A. Schmitt. Constraint-based collision and contact handling using impulses. In Proceddings of the 19th international conference on computer animation and social agents. July 2006

Questions



When and where collision happens?

How to compute impulses?

Reference: J. Bender and A. Schmitt. Constraint-based collision and contact handling using impulses. In Proceddings of the 19th international conference on computer animation and social agents. July 2006

Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

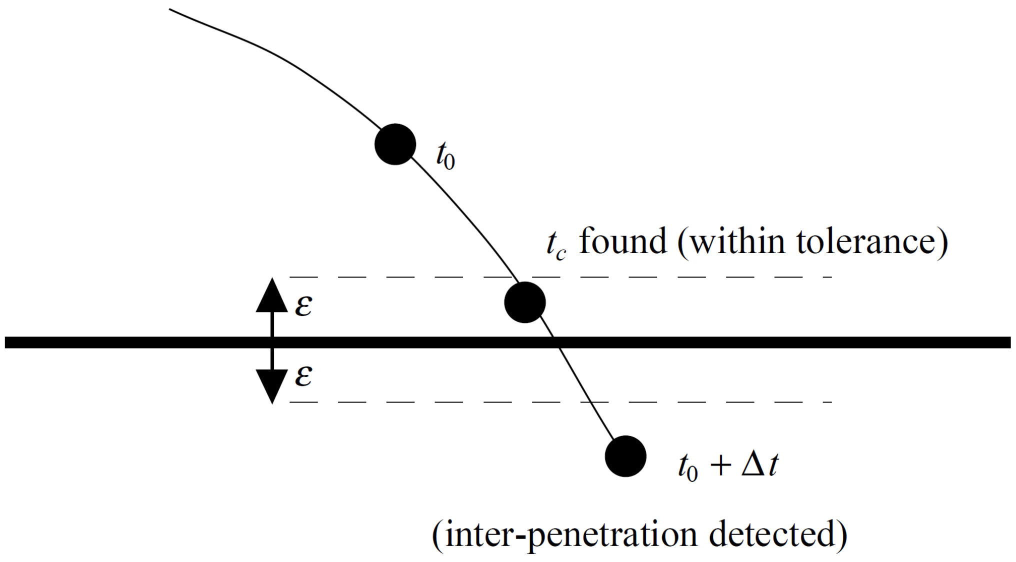
Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

Finding the collision time by backtracking

- Bisection method.

  If we know collision time t_c happens within $[t_0, t_0 + \Delta t)$, then check $t_0 + \Delta t/2$



Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Finding the collision time by backtracking

- Bisection method.
  If we know collision time t_c happens within $[t_0, t_0 + \Delta t)$, then check $t_0 + \Delta t/2$

- Easy to implement and quite robust, but a little slow. Faster convergence could be achieved using the *regula falsi* (false position) method

Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Finding the collision time by backtracking

- Bisection method.
  If we know collision time t_c happens within $[t_0, t_0 + \Delta t)$, then check $t_0 + \Delta t/2$

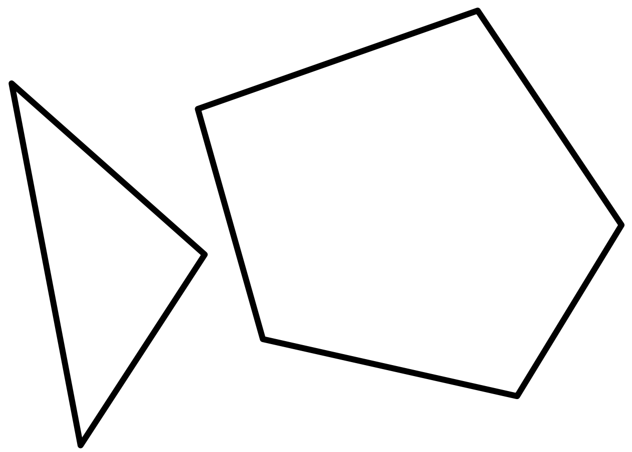
- Easy to implement and quite robust, but a little slow. Faster convergence could be achieved using the *regula falsi* (false position) method

- t_c is not always needed (discuss in penalty-based method)

Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Detecting collision points

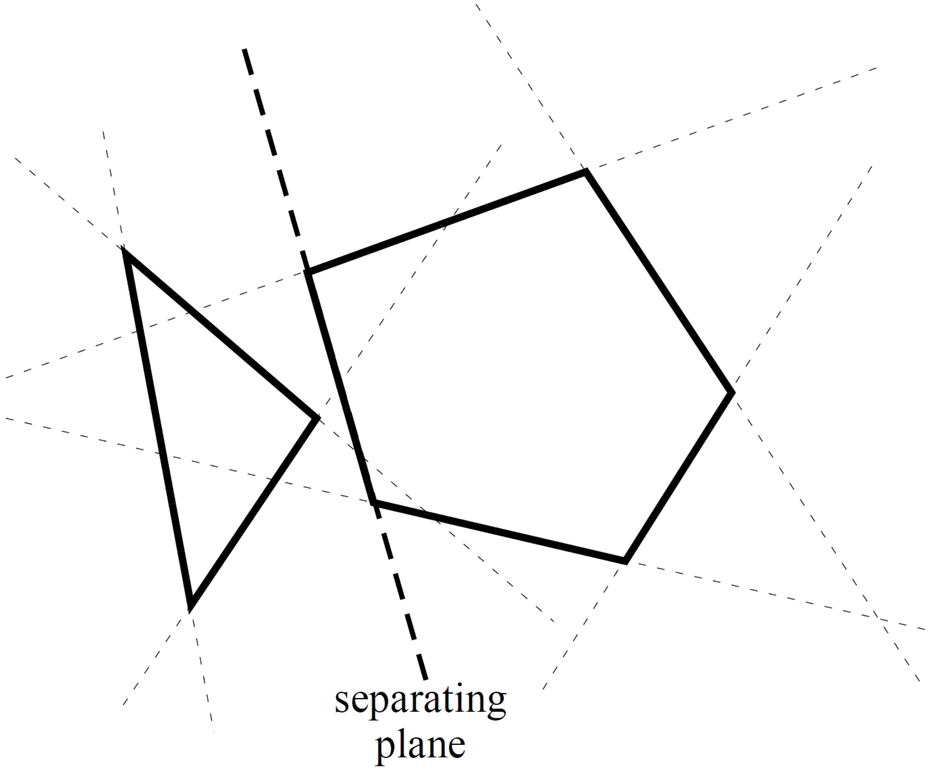
- If a separating plane is found, no inter-penetrating.



Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Detecting collision points: separating plane

- Exhaustive search for a separating plane considering two rigid bodies

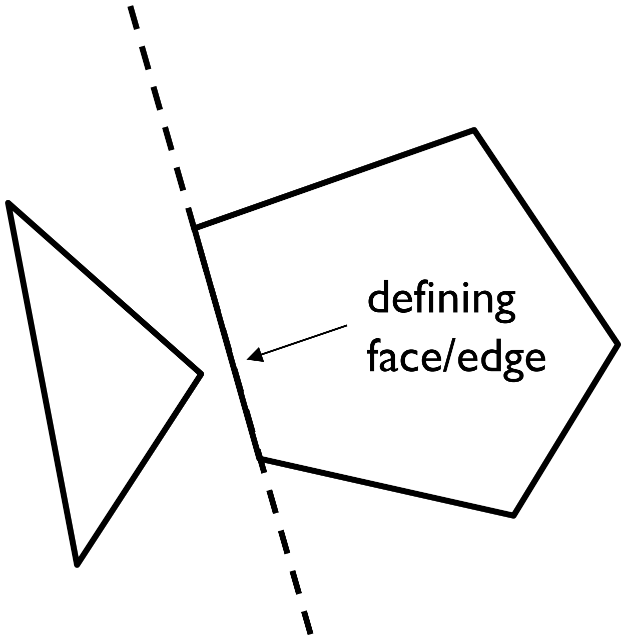


A separating plane either contains a face of one of the convex polyhedral or contains an edge from one convex polyhedral and is parallel to an edge of the other convex polyhedral

Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Detecting collision points: defining face/edge

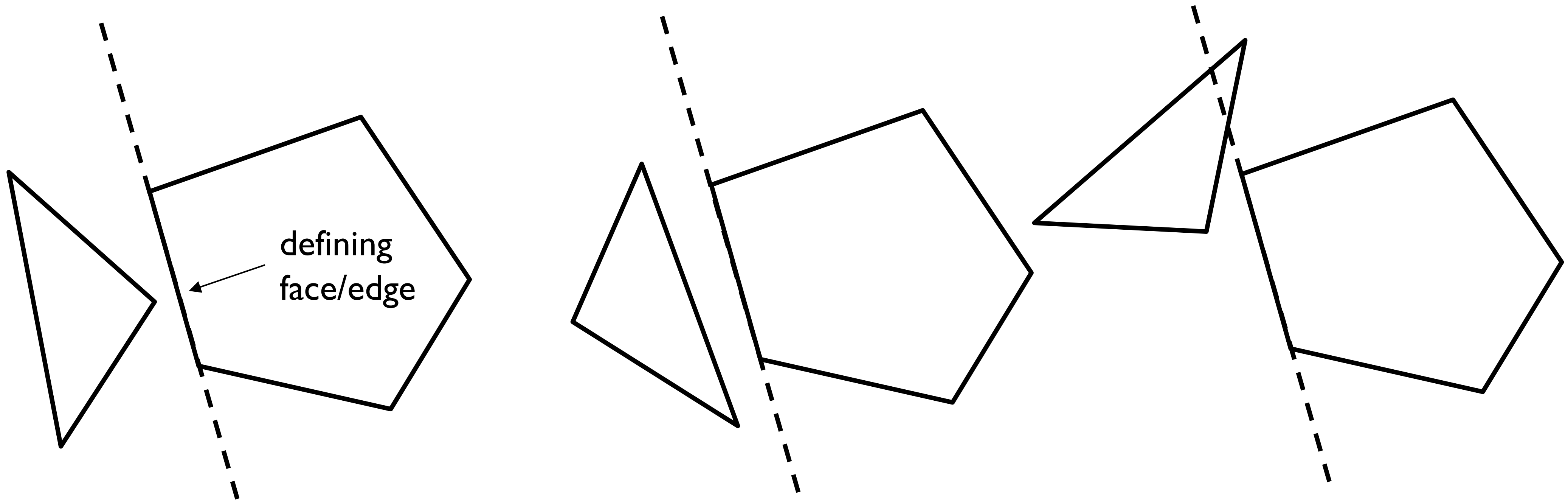
- The face/edge that is contained in the separating plane is called *defining* face/edge



Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Detecting collision points: subsequent time steps

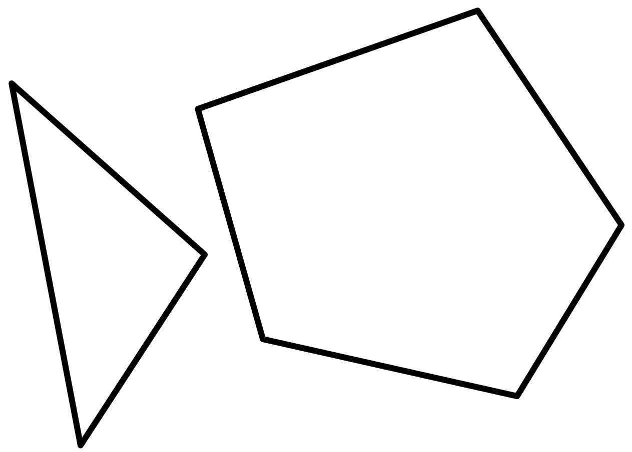
- For subsequent time steps, we will still use the defining face/edge to define a separating plane until it no longer does so.



Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

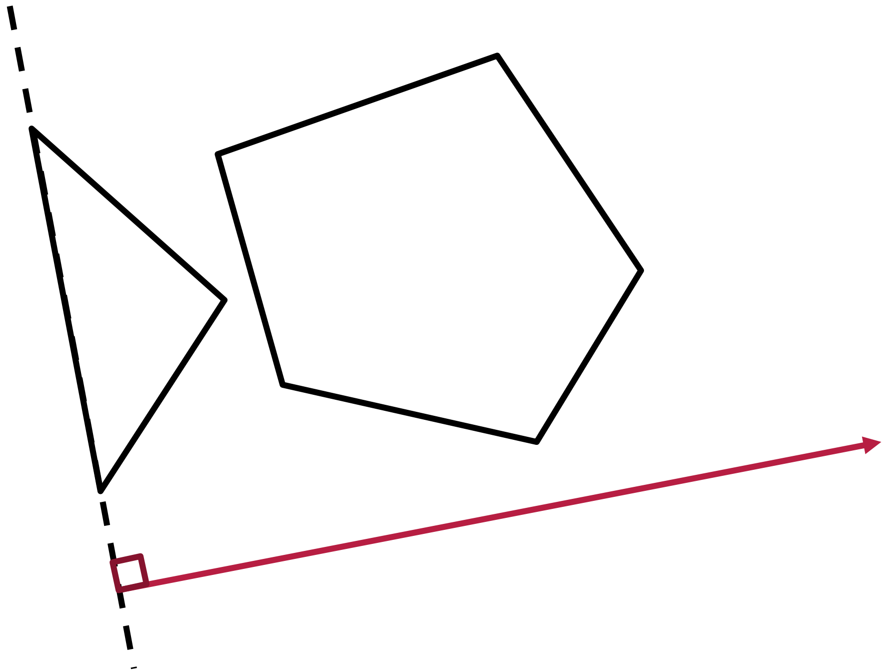
Separating axis theorem (SAT)

- Exhaustive search for projection gap in normal axes



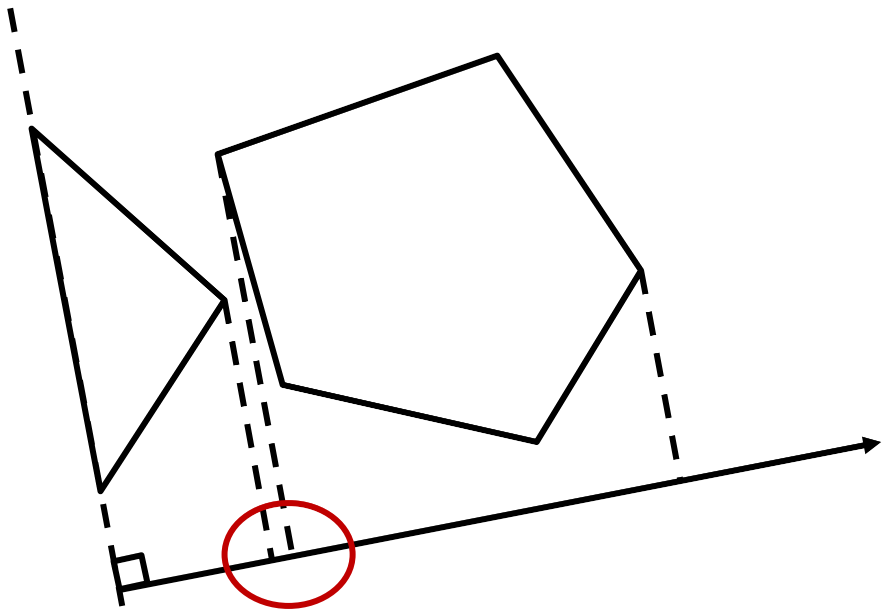
Separating axis theorem (SAT): axes

- In 2d cases, axes could be obtained using the unique normal of edges



Separating axis theorem (SAT): projection gap

- If there is a gap in projection region, there is no inter-penetration/collision
- Vertex projection can be computed using dot production



Separating axis theorem (SAT): 3d case

- If there is a gap in projection region, there is no inter-penetration/collision
- Vertex projection can be computed using dot production
- In 3d case, axes are normal of faces or the cross product of two edges (one from each object)

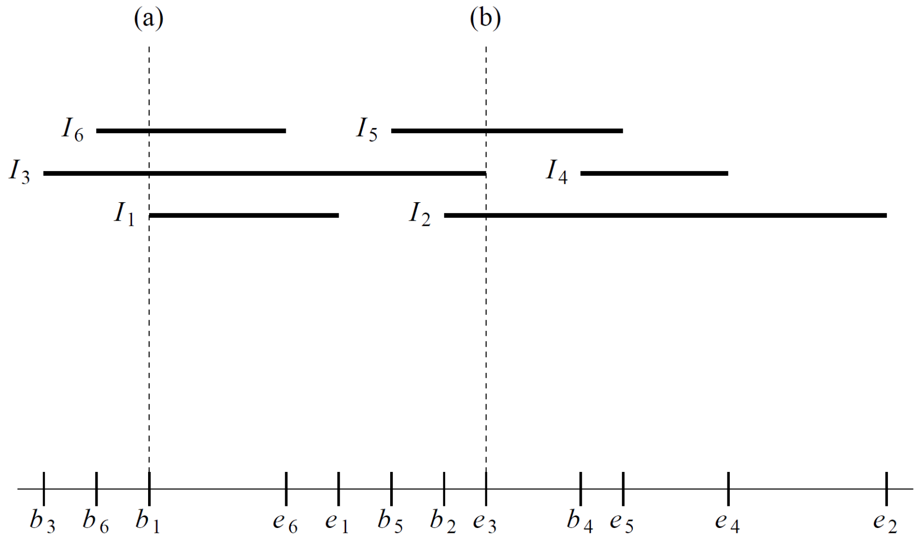
Collision detection is only done when bounding boxes overlap

- Naïve algorithm $O(n^2)$

Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

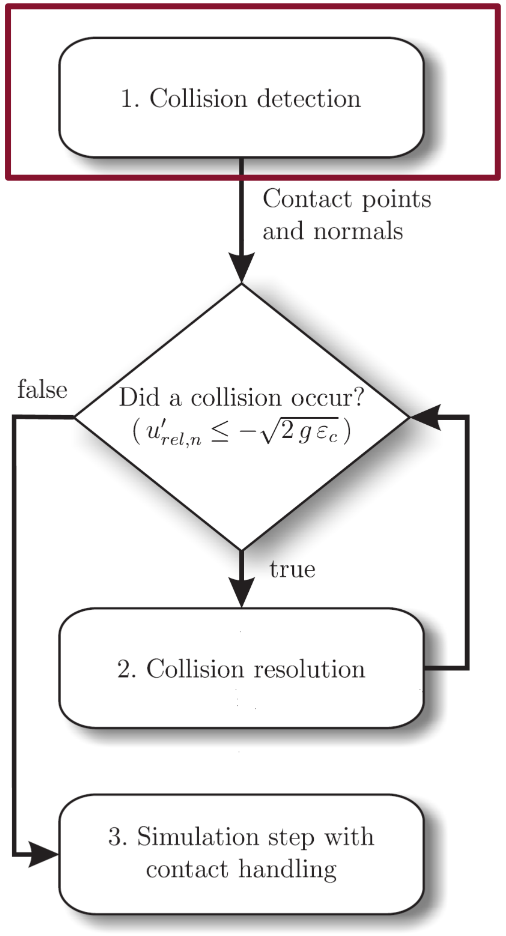
Collision detection is only done when bounding boxes overlap

- Naïve algorithm $O(n^2)$
- Sweep/sort algorithm $O(n \log n + k)$



Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

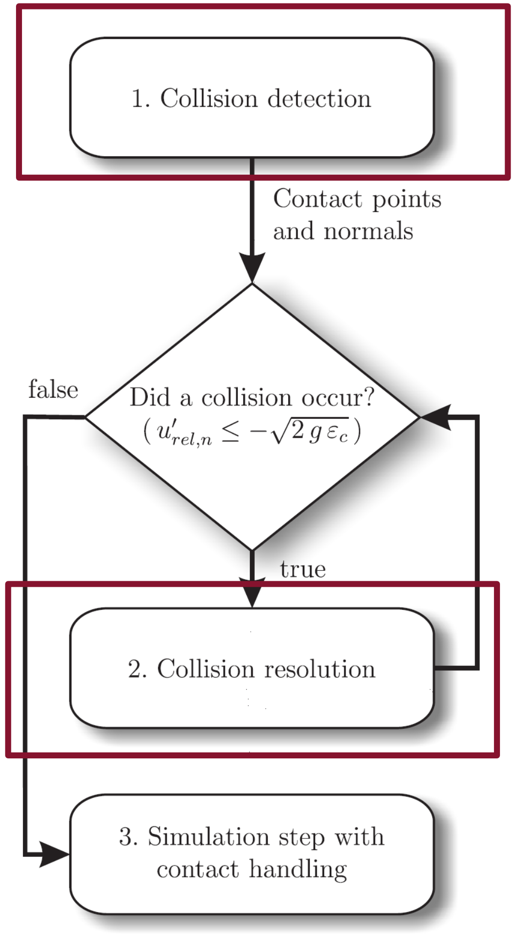
Questions



When and where collision happens?

Reference: J. Bender and A. Schmitt. Constraint-based collision and contact handling using impulses. In Proceddings of the 19th international conference on computer animation and social agents. July 2006

Questions



When and where collision happens?

How to resolve the collision?

(How to compute impulses or contact forces?)

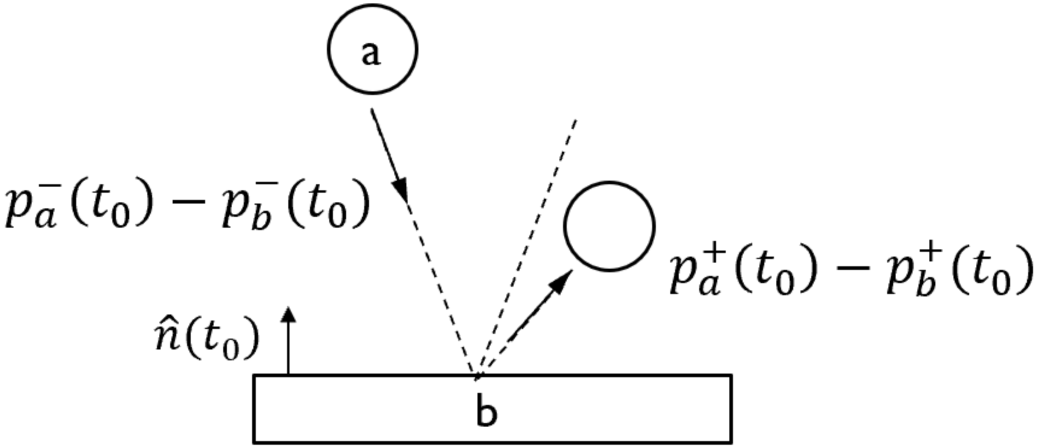
Reference: J. Bender and A. Schmitt. Constraint-based collision and contact handling using impulses. In Proceddings of the 19th international conference on computer animation and social agents. July 2006

Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

Force based

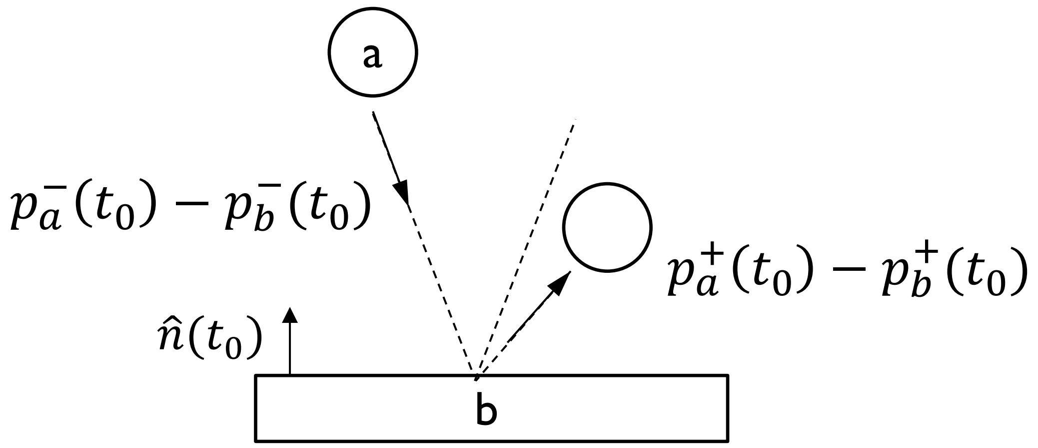
A simple example of collision



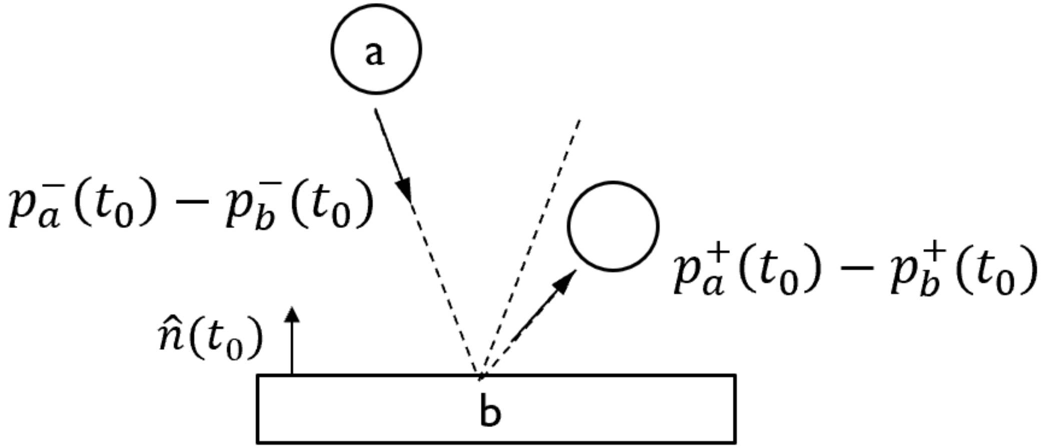
For rigid bodies, $p_a(t_0) = x_a(t_0) + R_a(t_0) r_a$
r_a is a vector from center of mass to collision point

Newton's law for restitution

- $v_{rel}^{+} = -\epsilon v_{rel}^{-}\ , \quad 0 \leq \epsilon \leq 1$
- $v_{rel}^{+} = \hat{n}(t_0)(\dot{p_a}^{+}(t_0) - \dot{p_b}^{+}(t_0))$



What is the impulse in this process?



Solve $J = j\hat{n}(t_0)$, given newton's law on restitution $v_a^+(t_0) = v_a^-(t_0) + j\hat{n}(t_0)/m_a$ and rigid body dynamics

Solving $J = j\hat{n}(t_0)$

$$v_{rel}^{+} = \hat{n}(t_0)\left(\dot{p_a}^{+}(t_0) - \dot{p_b}^{+}(t_0)\right) = -\epsilon v_{rel}^{-}$$

Newton's law for restitution

Solving $J = j\hat{n}(t_0)$

$$v_{rel}^{+} = \hat{n}(t_0)\left(\dot{p_a}^{+}(t_0) - \dot{p_b}^{+}(t_0)\right) = -\epsilon v_{rel}^{-}$$

Newton's law for restitution

$$\dot{p_a}^{+}(t_0) = v_a^{+}(t_0) + w_a^{+}(t_0) \times r_a$$

$$v_a^{+}(t_0) = v_a^{-}(t_0) + j\hat{n}(t_0)/m_a$$

$$w_a^{+}(t_0) = w_a^{-}(t_0) + I_a^{-1}(t_0)(r_a \times j\,\hat{n}(t_0))$$

$$\Rightarrow \dot{p}_a^{+}(t_0) = \dot{p}_a^{-}(t_0) + j(\frac{\hat{n}(t_0)}{m_a} + I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right) \times r_a$$

Rigid body dynamics for a

Solving $J = j\hat{n}(t_0)$

$$v_{rel}^{+} = \hat{n}(t_0)\left(\dot{p_a}^{+}(t_0) - \dot{p_b}^{+}(t_0)\right) = -\epsilon v_{rel}^{-}$$

Newton's law for restitution

$$\dot{p_a}^{+}(t_0) = v_a^{+}(t_0) + w_a^{+}(t_0) \times r_a$$

$$v_a^{+}(t_0) = v_a^{-}(t_0) + j\hat{n}(t_0)/m_a$$

$$w_a^{+}(t_0) = w_a^{-}(t_0) + I_a^{-1}(t_0)(r_a \times j\,\hat{n}(t_0))$$

$$\Rightarrow \dot{p}_a^{+}(t_0) = \dot{p}_a^{-}(t_0) + j(\frac{\hat{n}(t_0)}{m_a} + I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right) \times r_a$$

Rigid body dynamics for a

Similarly,

$$\dot{p}_b^{+}(t_0) = \dot{p}_b^{-}(t_0) - j(\frac{\hat{n}(t_0)}{m_b} + I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right) \times r_b$$

Rigid body dynamics for b

Solving $J = j\hat{n}(t_0)$

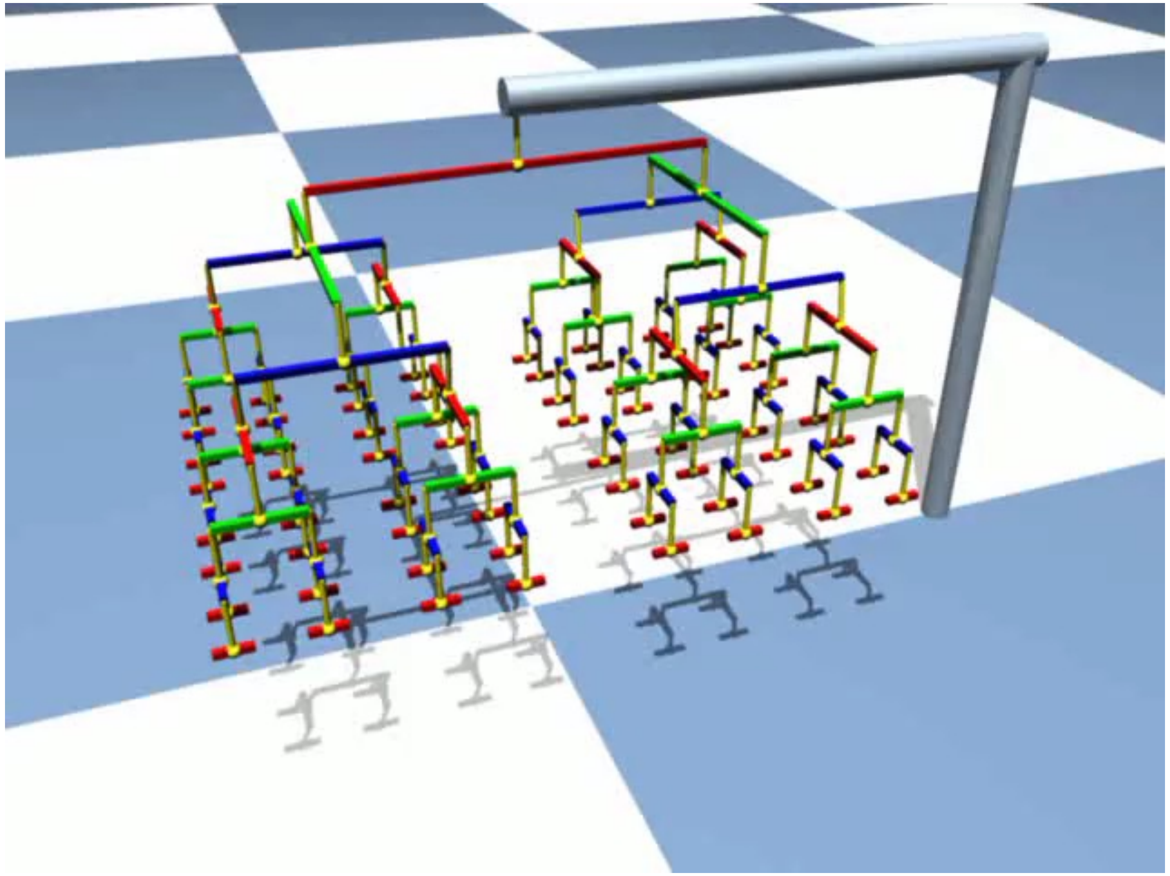
$$j = \frac{-(1+\epsilon)v_{rel}^{-}}{\frac{1}{m_a} + \frac{1}{m_b} + \hat{n}(t_0) \cdot z_a + \hat{n}(t_0) \cdot z_b}$$

$$z_a = (I_a^{-1}(t_0)(r_a \times \hat{n}(t_0)))\times r_a$$

$$z_b = (I_b^{-1}(t_0)(r_b \times \hat{n}(t_0)))\times r_b$$

Reference: David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989
https://www.cs.cmu.edu/~baraff/papers/sig89.pdf

Impulse-based dynamic simulation in linear time



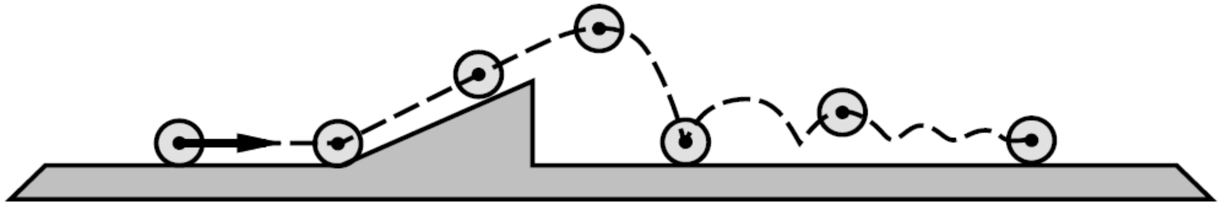
Reference: Jan Bender. Impulse-based dynamic simulation in linear time. Computer Animation and Virtual Worlds. 2007

Impulse-based methods: summary

- Newton's law of restitution: $v_{rel}^{+} = -\epsilon v_{rel}^{-}$, $0 \leq \epsilon \leq 1$
- $j = \frac{-(1+\epsilon)v_{rel}^{-}}{\frac{1}{m_a}+\frac{1}{m_b}+\hat{n}(t_0)\cdot(I_a^{-1}(t_0)(r_a\times\hat{n}(t_0)))\times r_a+\hat{n}(t_0)\cdot(I_b^{-1}(t_0)(r_b\times\hat{n}(t_0)))\times r_b}$

$$J = j\hat{n}(t_0)$$

- Impulses often applied in **local** contact resolution scheme



Reference:

James K. Hahn. Realistic animation of rigid bodies. Computer Graphics, Volume 22, Number 4, August 1988

Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4

Brian Vincent Mirtich. Impulse-based Dynamic Simulation of Rigid Body Systems. PhD thesis, Fall 1996

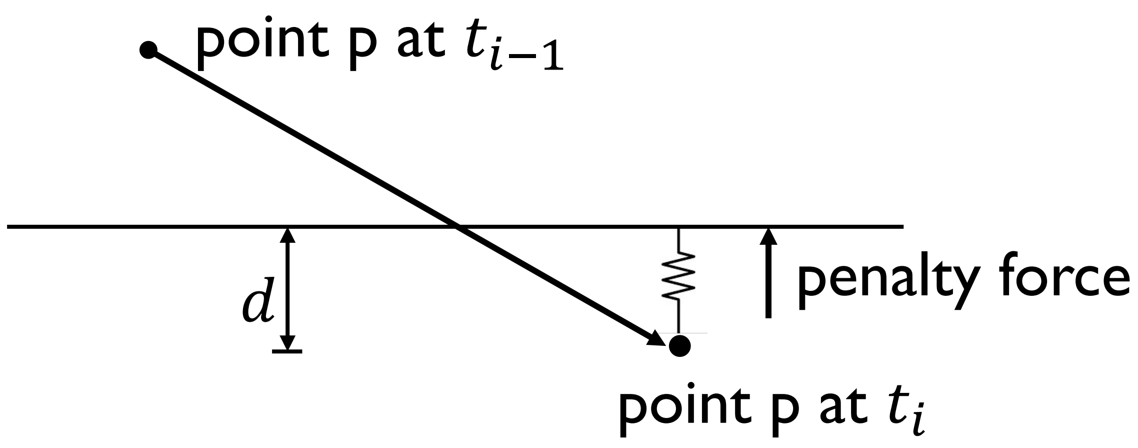
Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

} Force based

Prevent penetration using a spring

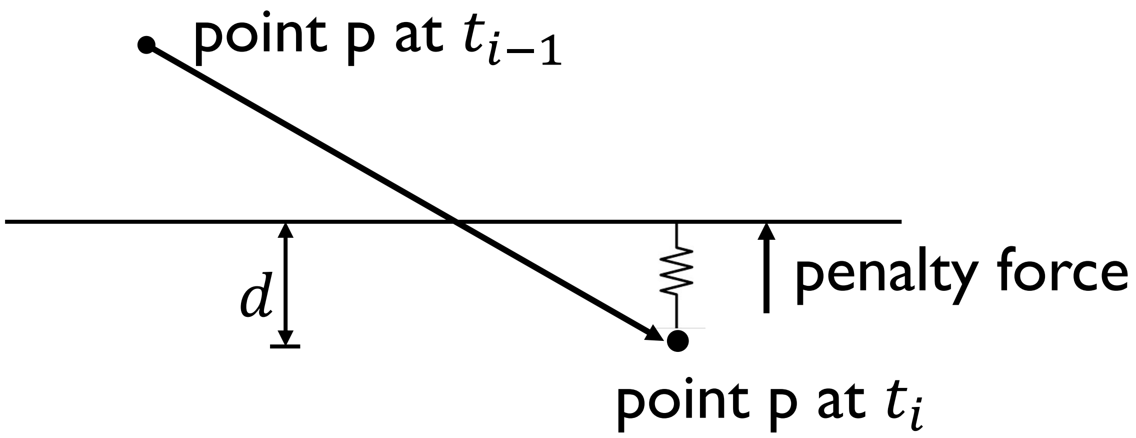
- A spring with zero resting length is attached to prevent interpenetration.
- t_c is not needed



Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4

Control parameter m, k_p

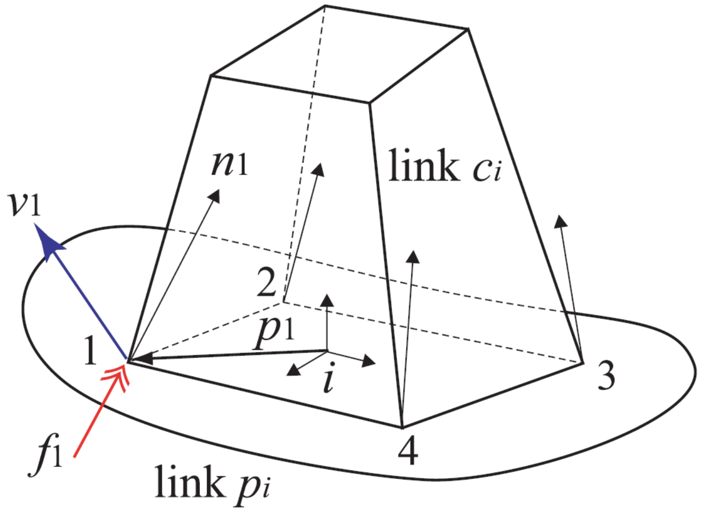
- $F = -k_p d,\ \ a = F/m.$ Here m, k_p are control parameter for the collision simulation



Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4
Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006

Control parameter m, k_p, k_d

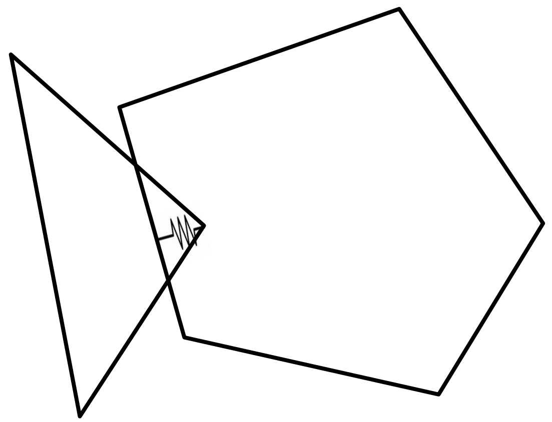
- $F = -\frac{d(k_p - k_d v_n)}{n_C}$. Here, v_n is the normal velocity and n_c is the number of contact points.



Reference: Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006
https://ieeexplore.ieee.org/abstract/document/1641984

Penalty based methods: convex polyhedral

- The penalty force will give rise to torque when not acting in line with the center of mass of an object
- The spring is attached to both objects and imparts an equal but opposite force on the two to restore nonpenetration.



Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4

Penalty-based methods: summary

Penalty-based methods compute the contact forces based on the penetration depth and normal velocity of a pair of objects using linear or nonlinear spring-damper model.

Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4
Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006

Penalty-based methods: summary

Penalty-based methods compute the contact forces based on the penetration depth and normal velocity of a pair of objects using linear or nonlinear spring-damper model.

Pros:

- Easy to incorporate with other forces like gravity
- Scales well with the complexity of the scene

Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4
Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006

Penalty-based methods: summary

Penalty-based methods compute the contact forces based on the penetration depth and normal velocity of a pair of objects using linear or nonlinear spring-damper model.

Pros:

- Easy to incorporate with other forces like gravity
- Scales well with the complexity of the scene

Cons:

- Need to specify control parameters in a trial-and-error process
- Unclear in determining where contact between objects should break

Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4
Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006

Penalty-based methods: summary

Penalty-based methods compute the contact forces based on the penetration depth and normal velocity of a pair of objects using linear or nonlinear spring-damper model.

Pros:

- Easy to incorporate with other forces like gravity
- Scales well with the complexity of the scene

There are advanced work to do it right

Cons:

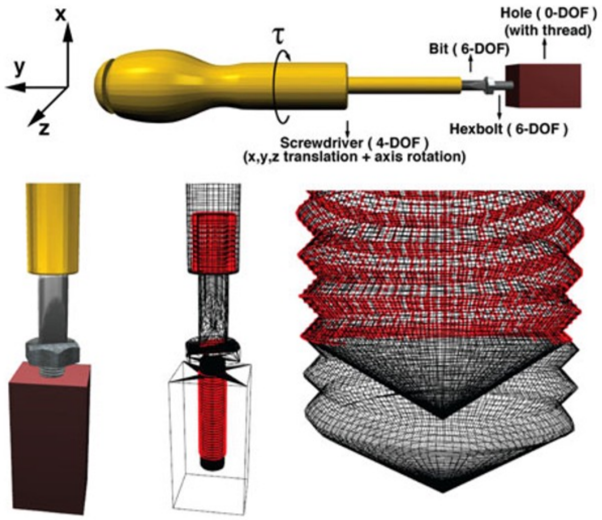
- Need to specify control parameters in a trial-and-error process
- Unclear in determining where contact between objects should break

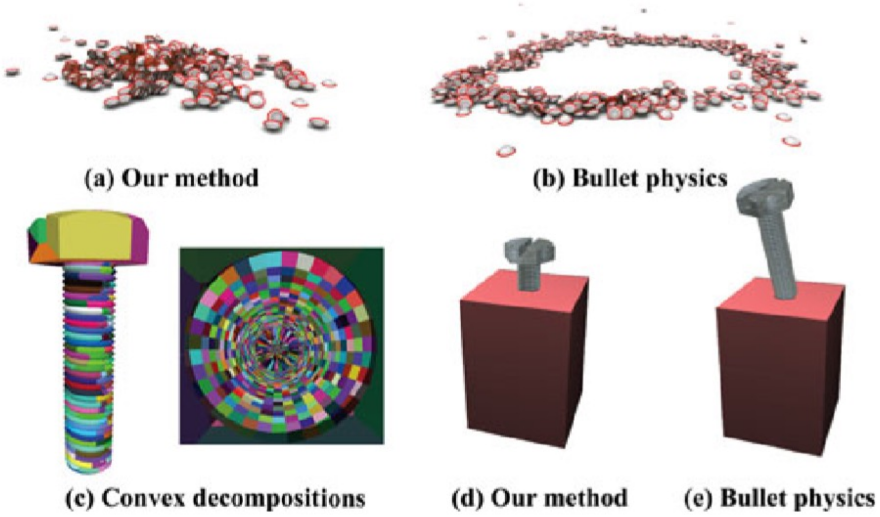
Reference: Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4
Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006

IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 20, NO. 9, SEPTEMBER 2014

Implicit Multibody Penalty-Based Distributed Contact

Hongyi Xu, *Member, IEEE*, Yili Zhao, *Member, IEEE*, and Jernej Barbič, *Member, IEEE*





IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 20, NO. 9, SEPTEMBER 2014

Implicit Multibody Penalty-Based Distributed Contact

Hongyi Xu, *Member, IEEE*, Yili Zhao, *Member, IEEE*, and Jernej Barbič, *Member, IEEE*

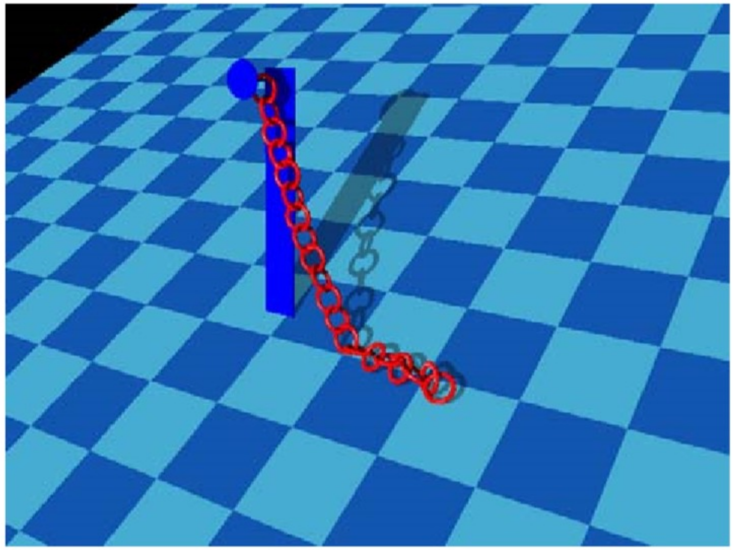
It addressed the stability problems due to the **highly variable and unpredictable net stiffness** by employing exact analytical contact gradients, symbolic Gaussian elimination, SVD solver, and semi-implicit integration.

Comparison to Bullet Physics

Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Stable Penalty-Based Model of Frictional Contacts

Katsu Yamane and Yoshihiko Nakamura
Department of Mechano-Informatics, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
yamane@ynl.t.u-tokyo.ac.jp



Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Stable Penalty-Based Model of Frictional Contacts

Katsu Yamane and Yoshihiko Nakamura
Department of Mechano-Informatics, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
yamane@ynl.t.u-tokyo.ac.jp

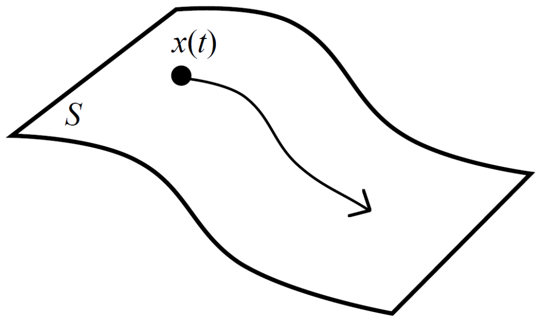
- It solves the problems in implementing Coulomb's friction model, i.e. how to handle static/dynamic friction forces.

Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

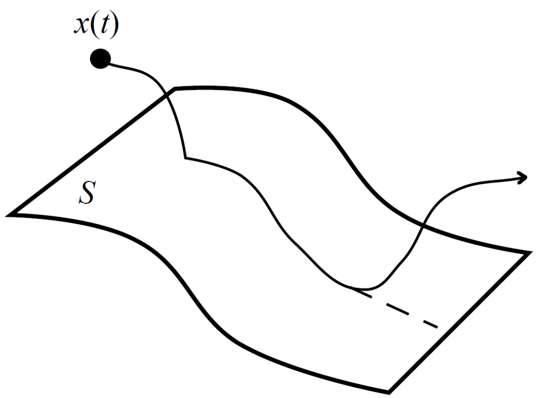
Constraint based method

- Computes constraint forces that are designed to exactly cancel any external accelerations that would result in interpenetration



$C(x(t)) = 0$

Equality constraints

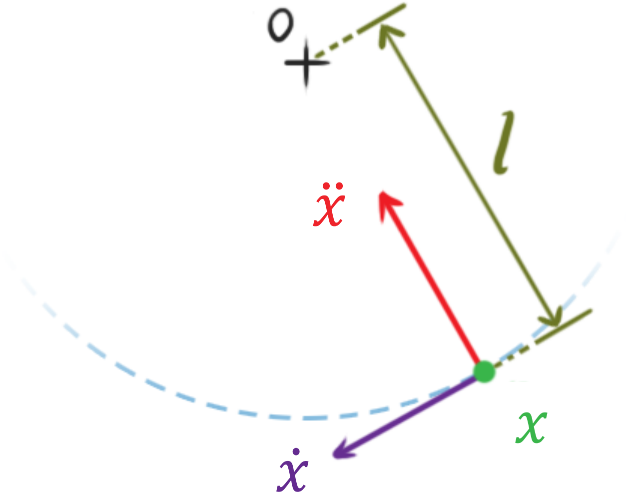


$C(x(t)) \geq 0$

Inequality constraints

Reference: David Baraff. Non-penetrating Rigid Body Simulation. Eurographics 1993 State of the Art Reports.
https://www.cs.cmu.edu/~baraff/papers/eg93.pdf

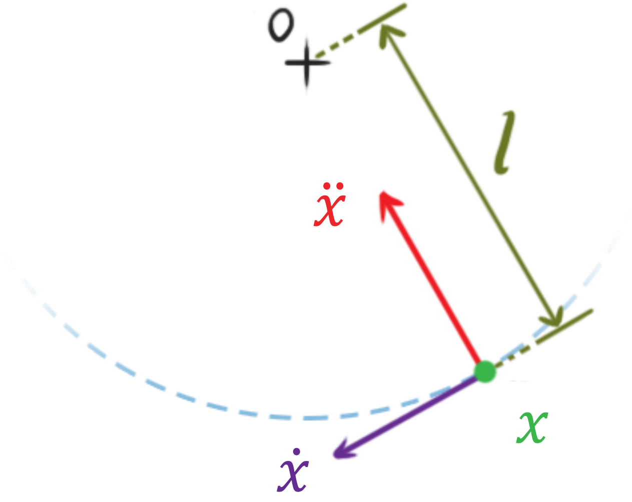
A simple example of equality constraint



- $C(x) = \frac{1}{2}(x \cdot x - l^2)$
- $C(x) = 0$
- In addition, $\dot{C}(x) = 0, \ddot{C}(x) = 0$

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Solving equality constraints



- From equality constraints, we have

$$C(x) = \frac{1}{2}(x \cdot x - l^2) = 0 \quad [1]$$

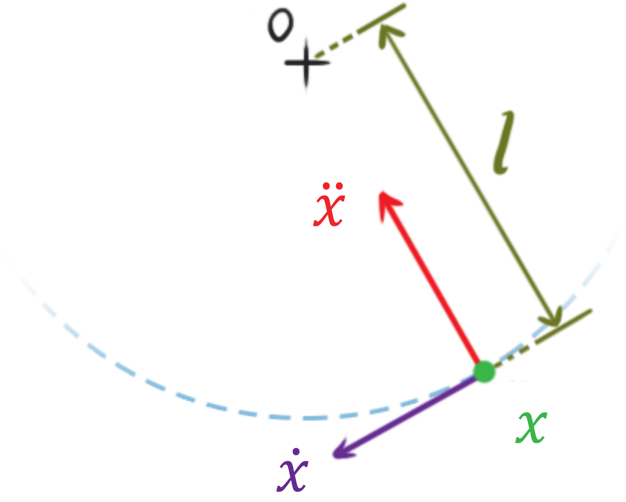
$$\dot{C(x)} = x \cdot \dot{x} = 0 \quad [2]$$

$$\ddot{C(x)} = \ddot{x} \cdot x + \dot{x} \cdot \dot{x} = 0 \quad [3]$$

- From newton's 2nd law: $\ddot{p} = (f_{ext} + f_C)/m \quad [4]$

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Solving equality constraints



- From equality constraints, we have

$$C(x) = \frac{1}{2}(x \cdot x - l^2) = 0 \quad [1]$$

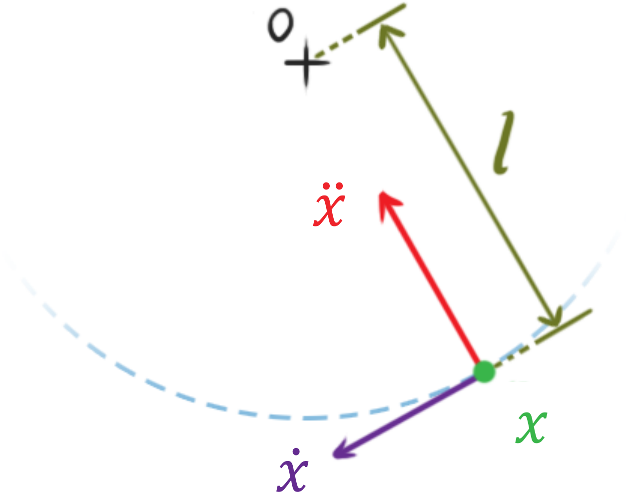
$$\dot{C(x)} = x \cdot \dot{x} = 0 \quad [2]$$

$$\ddot{C(x)} = \ddot{x} \cdot x + \dot{x} \cdot \dot{x} = 0 \quad [3]$$

- From newton's 2nd law: $\ddot{x} = (f_{ext} + f_C)/m \quad [4]$
- From [3][4], we can derive $f_C \cdot x = -f_{ext} \cdot x - m\,\dot{x} \cdot \dot{x}$ [5]

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Solving equality constraints



- From equality constraints, we have

$$C(x) = \frac{1}{2}(x \cdot x - l^2) = 0 \quad [1]$$

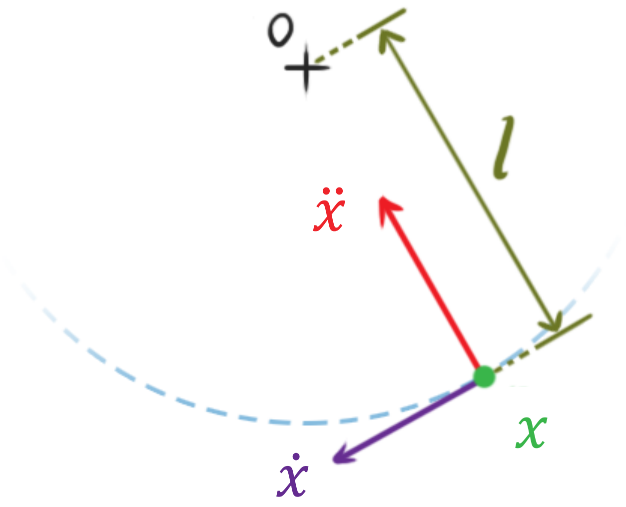
$$\dot{C(x)} = x \cdot \dot{x} = 0 \quad [2]$$

$$\ddot{C(x)} = \ddot{x} \cdot x + \dot{x} \cdot \dot{x} = 0 \quad [3]$$

- From newton's 2nd law: $\ddot{x} = (f_{ext} + f_C)/m \quad [4]$
- From [3][4], we can derive $f_C \cdot x = -f_{ext} \cdot x - m\,\dot{x} \cdot \dot{x}$ [5]
- Additional condition: $f_C \cdot \dot{x} = 0$ [6]
- From [2][6], we can get $f_C = \lambda x$ [7]

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Solving equality constraints



- From equality constraints, we have

$$C(x) = \frac{1}{2}(x \cdot x - l^2) = 0 \quad [1]$$

$$C(\dot{x}) = x \cdot \dot{x} = 0 \qquad [2]$$

$$C(\ddot{x}) = \ddot{x} \cdot x + \dot{x} \cdot \dot{x} = 0 \quad [3]$$

- From newton's 2nd law: $\ddot{x} = (f_{ext} + f_C)/m \quad [4]$
- From [3][4], we can derive $f_C \cdot x = -f_{ext} \cdot x - m\,\dot{x} \cdot \dot{x}$ [5]
- Additional condition: $f_C \cdot \dot{x} = 0$ [6]
- From [2][6], we can get $f_C = \lambda x$ [7]
- From [5][7], we can obtain $\lambda = \frac{-f_{ext} \cdot x - m\dot{x} \cdot \dot{x}}{x \cdot x}$

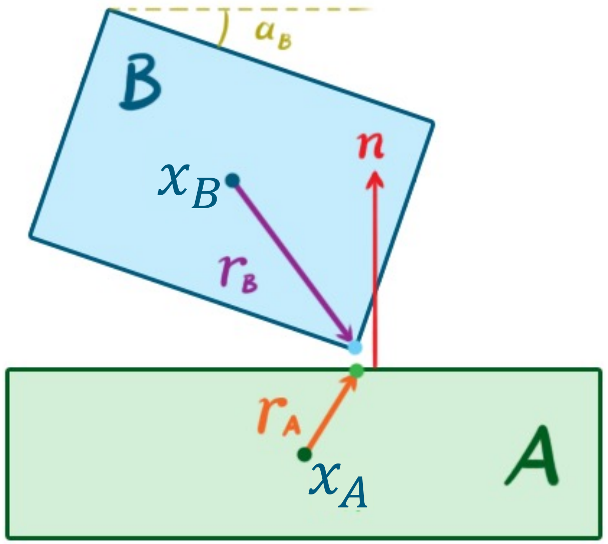
Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Solving equality constraints

- λ is also known as Lagrange multiplier
- For any constraints, the calculation involves determining the direction of force and its magnitude λ
- For rigid bodies, $JM^{-1}J^T\lambda = -\dot{J}\dot{q} - JM^{-1}F_{ext}$
  Here, q is a state vector for position and rotation, M is a matrix of mass and inertia, J is the Jacobian matrix of constraints

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

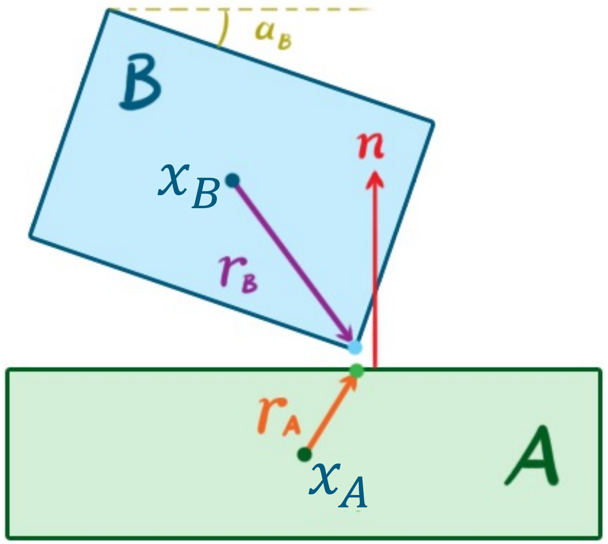
A simple example of inequality constraint



- A and B are colliding
- $R(\alpha_B) = \begin{bmatrix} \cos\alpha_B & -\sin\alpha_B \\ \sin\alpha_B & \cos\alpha_B \end{bmatrix}$

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

A simple example of inequality constraint

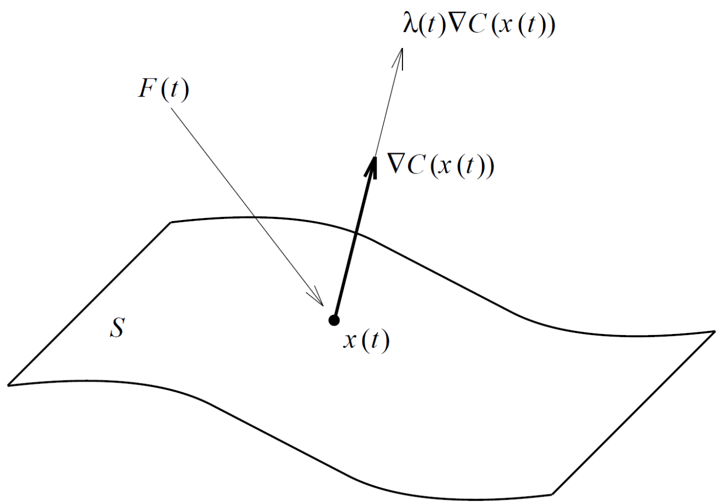


- A and B are colliding
- $R(\alpha_B) = \begin{bmatrix} \cos\alpha_B & -\sin\alpha_B \\ \sin\alpha_B & \cos\alpha_B \end{bmatrix}$
- $C(x_B, \alpha_B, x_A, \alpha_A) = (x_B + R(\alpha_B)r_B) - (x_A + R(\alpha_A)r_A)$
- $n^T C(x_B, \alpha_B, x_A, \alpha_A) \geq 0$ (i.e. penetration depth is positive)

Reference: https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation

Constraint forces to avoid penetration

- Normal of surface $\nabla C(x(t))$
- $F_c(t) = \lambda(t)\ \nabla C\big(x(t)\big)$, where $\lambda(t)$ is a scalar. $\lambda(t) \geq 0$ so that the force will only push not pull the two objects



Reference: David Baraff. Non-penetrating Rigid Body Simulation. Eurographics 1993 State of the Art Reports.
https://www.cs.cmu.edu/~baraff/papers/eg93.pdf

Formulate as Linear Complementarity Problem (LCP)

- The constraint problem could be formulated as an LCP problem
- Given a real matrix M and vector q, the linear complementarity problem LCP(M, q) seeks vectors z and w which satisfy the following constraints:

$$w, z \geq 0$$
$$z^T w = 0$$
$$w = Mz + q$$

Refernce: http://www.cs.unc.edu/~lin/COMP768-F07/LEC/rbd2.pdf

Resting contact

- Body separating (no response required)

$$v_{rel} > \epsilon$$

- Colliding contact

$$v_{rel} < -\epsilon$$

- Resting contact

$$-\epsilon < v_{rel} < \epsilon$$

Reference: http://www.cs.unc.edu/~lin/COMP768-F07/LEC/rbd2.pdf

Resting constraint forces properties

- Prevent interpenetration $\ddot{d}_i(t_0) \geq 0$
- Repulsive normal force: $f_N(t) \geq 0$
- Normal force is zero when bodies start to separate: $f_i \ddot{d}_i(t_0) = 0$

Refernce: http://www.cs.unc.edu/~lin/COMP768-F07/LEC/rbd2.pdf

Linear Complementarity Problem (LCP)

- $\ddot{d}_i(t_0) = a_{i1}f_1 + a_{i2}f_2 + \cdots + a_{in}f_n + b_i$

$$\begin{pmatrix} \ddot{d}_1(t_0) \\ \vdots \\ \ddot{d}_n(t_0) \end{pmatrix} = A \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

$$\begin{cases} \ddot{d}_i(t_0) \geq 0 & \text{Prevent interpenetration} \\ f_i \geq 0 & \text{Repulsive} \\ f_i \ddot{d}_i(t_0) = 0 & \text{Is zero when bodies are starting to come apart} \end{cases}$$

Refernce: http://www.cs.unc.edu/~lin/COMP768-F07/LEC/rbd2.pdf

Solving Linear Complementarity Problem (LCP)

- Pivoting algorithms (like Gaussian elimination)
  - Need read and write access to matrices
  - Do not provide useful intermediate results
  - May exploit sparsity well
- Iterative algorithms (like Conjugate gradients)
  - Only need read access to matrices
  - Can stop early for approximation
  - Faster for large matrices
  - Can be warm started (i.e. from previous results)

Constraint-based methods: summary

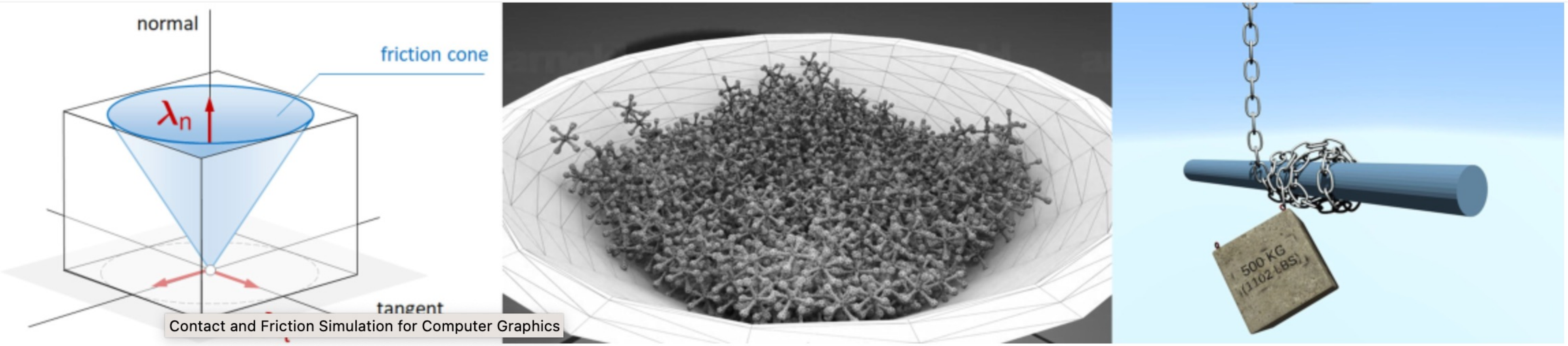
- Computes constraint forces that are designed to exactly cancel any external accelerations that would result in interpenetration
- Useful when the scene is composed of relatively small number of objects with simple shape
- Does not scale well with the complexity of the scene
(the penalty based method scales well with the complexity of the scene)

Outline

- Collision detection basics
- Computing impulses using a coefficient of restitution
- Penalty based method
- Constraint based method
- Recent work

SIGGRAPH'22 Course: Contact and Friction Simulation for Computer Graphics

Changelog



Sheldon Andrews[1]
[1]École de technologie supérieure

Kenny Erleben[2]
[2]University of Copenhagen

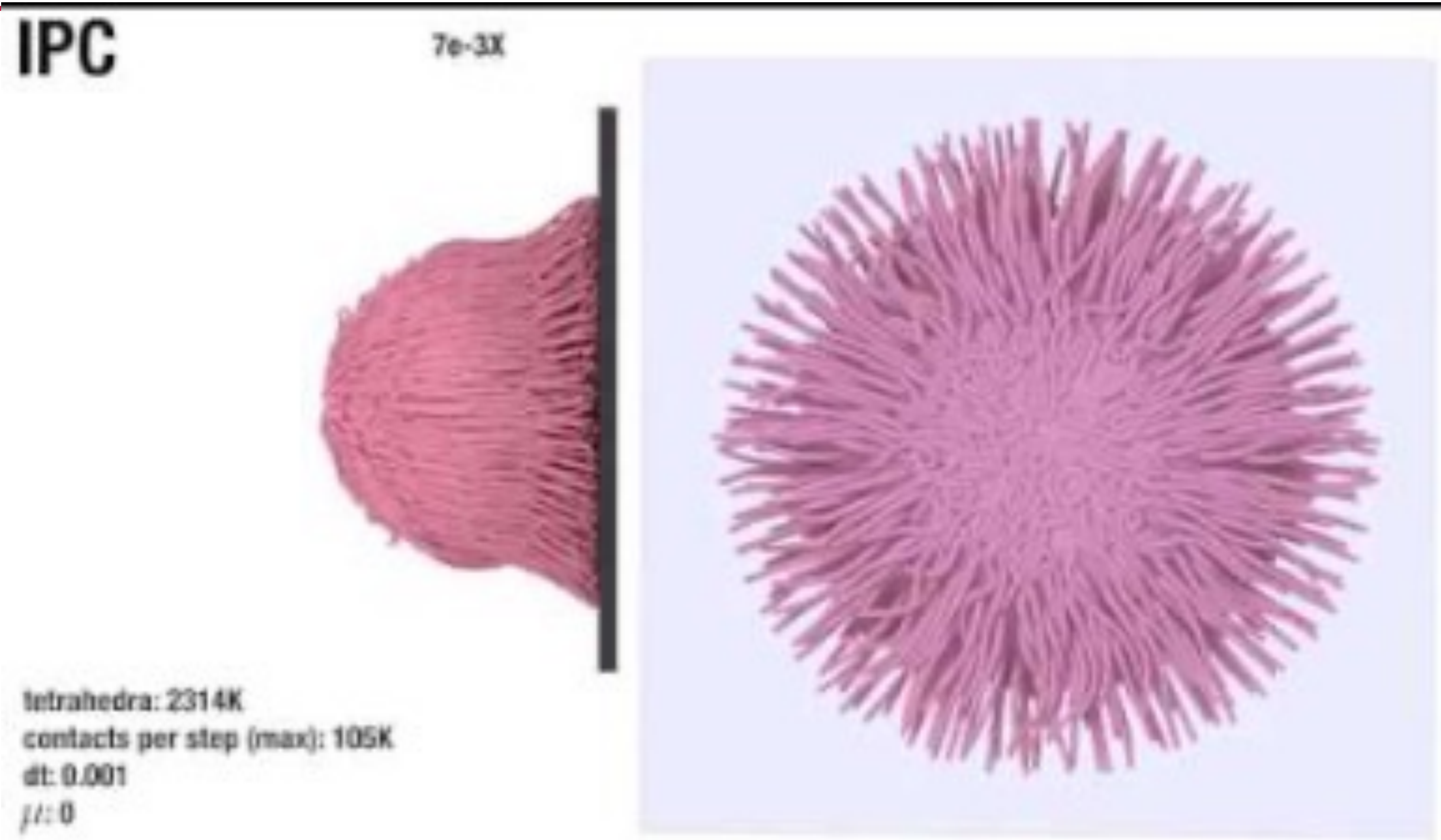
Zachary Ferguson[3]
[3]New York University

This course covers fundamental topics on the nature of contact modeling and simulation for computer graphics. Specifically, we provide mathematical details about formulating contact as a complementarity problem in rigid body and soft body animations. We briefly cover several approaches for contact generation using discrete collision detection. Then, we present a range of numerical techniques for solving the associated LCPs and NCPs. The advantages and disadvantages of each technique are further discussed in a practical manner, and best practices for implementation are discussed. Finally, we conclude the course with several advanced topics such as methods for soft body contact problems, barrier functions, and anisotropic friction modeling. Programming examples are provided in our appendix as well as on the course website to accompany the course notes.

Updated for SIGGRAPH '22: This version of the course adds coverage of multivariate continuous collision detection, an extended treatment of penalty-based methods to include barrier methods, a new chapter on implicit time integration schemes and methods for solving soft body contact problems has been added. Additionally, we now include a more in-depth coverage of anisotropic friction simulation, as well as a tutorial that guides the reader through the steps of creating a rigid body simulator with frictional contact

https://siggraphcontact.github.io/

Incremental Potential Contact (SIGGRAPH 2020)



Rigid IPC (SIGGRAPH 2021)

Affine Body Dynamics (SIGGRAPH 2022)

Affine Body Dynamics:

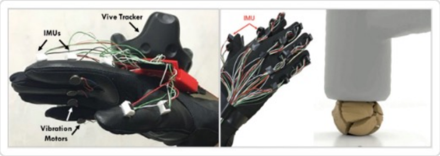
Fast, Stable, and Intersection-free Simulation of Stiff Materials

Lei Lan[1,2], Danny M. Kaufman[3], Minchen Li[4,5], Chenfanfu Jiang[4,5], Yin Yang[1,2,5]

1. 2. THE UNIVERSITY OF UTAH 3. Adobe 4. UCLA 5. TimeStep Inc.

Minchen Li

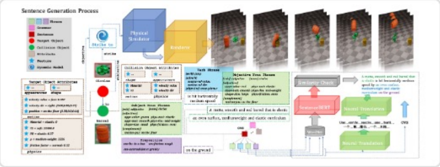
- Will give a talk here at CMU March 13, 3pm



Reconfigurable Data Glove for Reconstructing Physical and Virtual Grasps

Hangxin Liu, Zeyu Zhang, Ziyuan Jiao, Zhenliang Zhang, **Minchen Li**, Chenfanfu Jiang, Yixin Zhu, Song-Chun Zhu

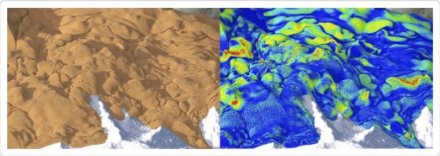
Engineering, 2023 Paper Project Page



TPA-Net: Generate A Dataset for Text to Physics-based Animation

Yuxing Qiu, Feng Gao, **Minchen Li**, Govind Thattai, Yin Yang, Chenfanfu Jiang

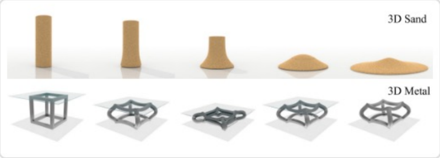
Arxiv 2211.13887 Preprint Gallery



A Sparse Distributed Gigascale Resolution Material Point Method

Yuxing Qiu, Samuel T. Reeve, **Minchen Li**, Yin Yang, Stuart R. Slattery, Chenfanfu Jiang

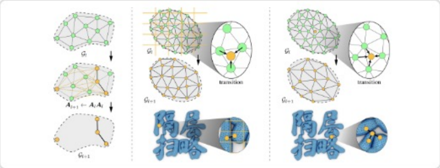
ACM Transactions on Graphics, 2022 (presentation at SIGGRAPH) Paper



PlasticityNet: Learning to Simulate Metal, Sand, and Snow for Optimization Time Integration

Xuan Li, Yadi Cao, **Minchen Li**, Yin Yang, Craig Schroeder, Chenfanfu Jiang

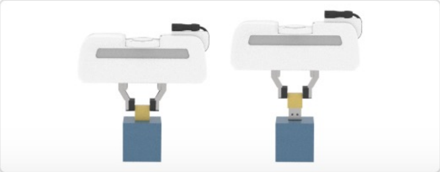
Neural Information Processing Systems (NIPS), 2022 Paper Video



Bi-Stride Multi-Scale Graph Neural Network for Mesh-Based Physical Simulation

Yadi Cao, Menglei Chai, **Minchen Li**, Chenfanfu Jiang

Arxiv 2210.02573 Preprint



Midas: A Multi-Joint Robotics Simulator with Intersection-Free Frictional Contact

Yunuo Chen, **Minchen Li**, Wenlong Lu, Chuyuan Fu, Chenfanfu Jiang

Arxiv 2210.00130 Preprint

Summary

- Collision detection
  - backtracking t_c,
  - determining interpenetration using separation plane
  - determining interpeneration using separating axis theorem (SAT)
- Impulse based method. Computing impulse based on restitution coefficient ϵ
- Penalty based method
- Constraint based method

} Force based

References

- David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. SIGGRAPH 1989 https://www.cs.cmu.edu/~baraff/papers/sig89.pdf
- Andrew Witkin and David Baraff, Physically Based Modeling: Principles and Practice, Online SIGGRAPH 1997 Course Notes https://www.cs.cmu.edu/~baraff/sigcourse/
- David Baraff. Non-penetrating Rigid Body Simulation. Eurographics 1993 State of the Art Reports. https://www.cs.cmu.edu/~baraff/papers/eg93.pdf
- Rick Parent. Compute Animation: Algorithm and Technique. Chapter 7.4 2012
- Katsu Yamane and Yoshihiko Nakamura. Stable Penalty-based Model of Frictional Contacts. ICRA 2006 https://ieeexplore.ieee.org/document/1641984
- UNC COMP768 Physically-Based Modeling, Simulation and Animation, Course Slides. Ming C. Lin 2007
- Crispin Deul, Patrick Charrier, Jan Bender. Position-Based Rigid Body Dynamics. Computer Animation and Virtual Worlds 2014
- Jan Bender, Matthias Müller and Miles Macklin, A Survey on Position Based Dynamics, 2017, In *Tutorial Proceedings of Eurographics*, 2017