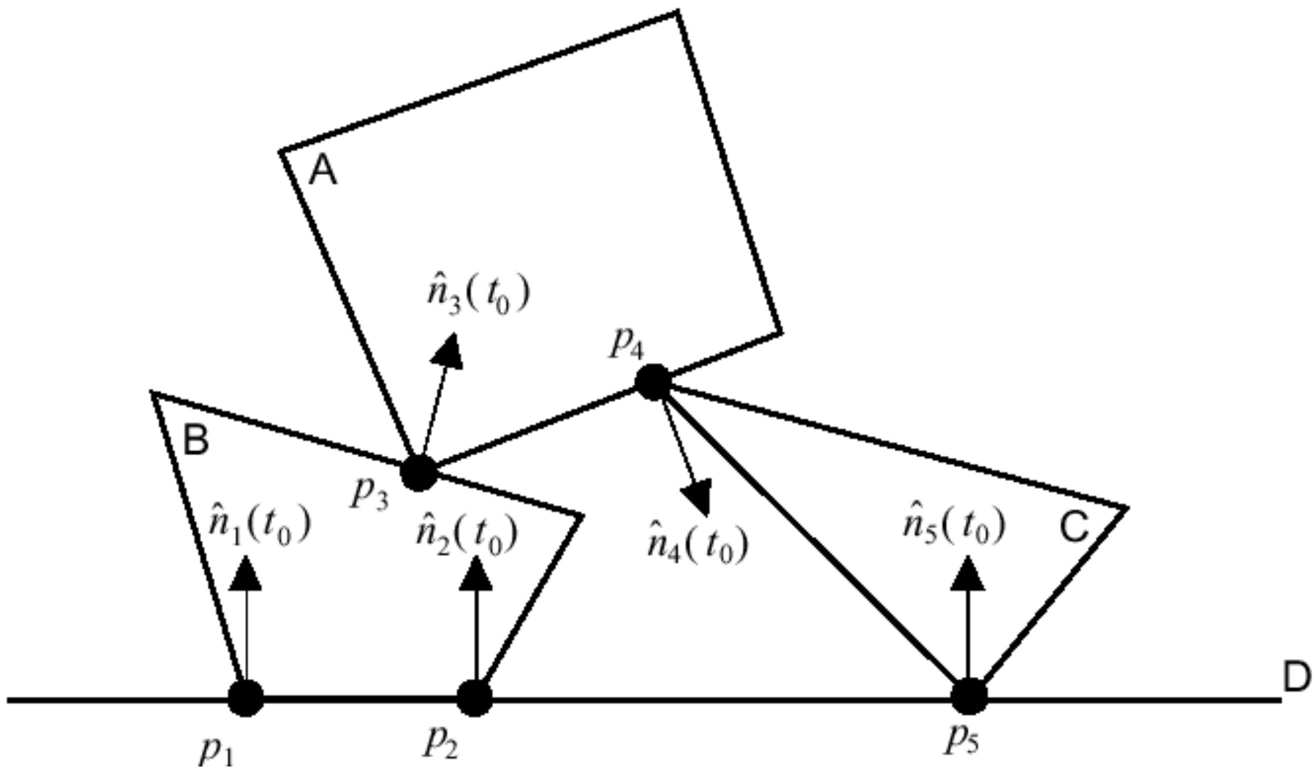
Bodies intersect → classify contacts

- Bodies separating
  - $v_{rel} > \varepsilon$
  - No response required
- Colliding contact
  - $v_{rel} < -\varepsilon$
- Resting contact
  - $-\varepsilon < v_{rel} < \varepsilon$
  - Gradual contact forces avoid interpenetration
  - All resting contact forces must be computed and applied together because they can influence one another

Resting Contact Response

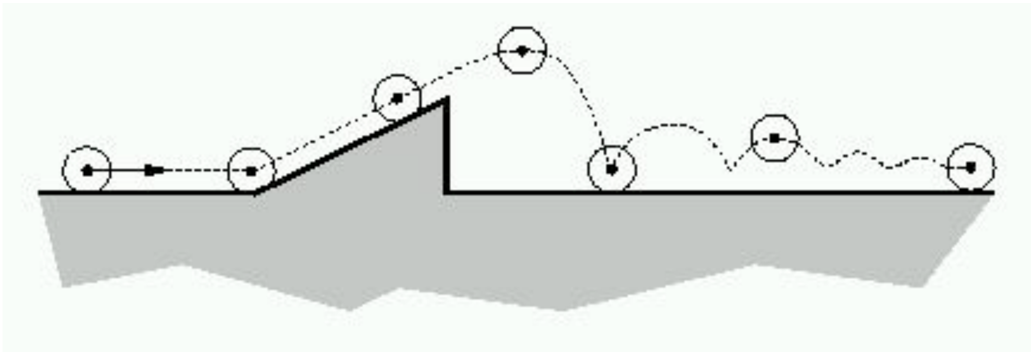


Handling of Resting Contact

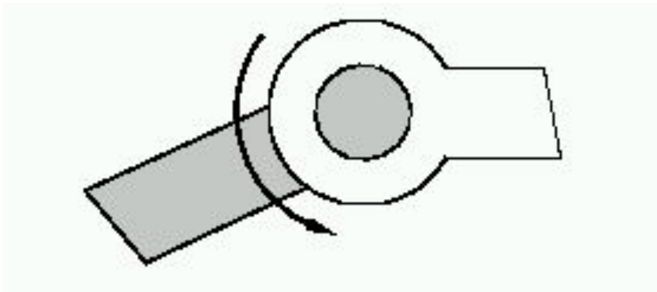
- Resting contact is a **constraint**!
  - Local vs. global methods
  - Impulse-based solution methods
  - Constraint-based solution methods
- Friction

Local vs. Global

- Impulse-based dynamics (local)



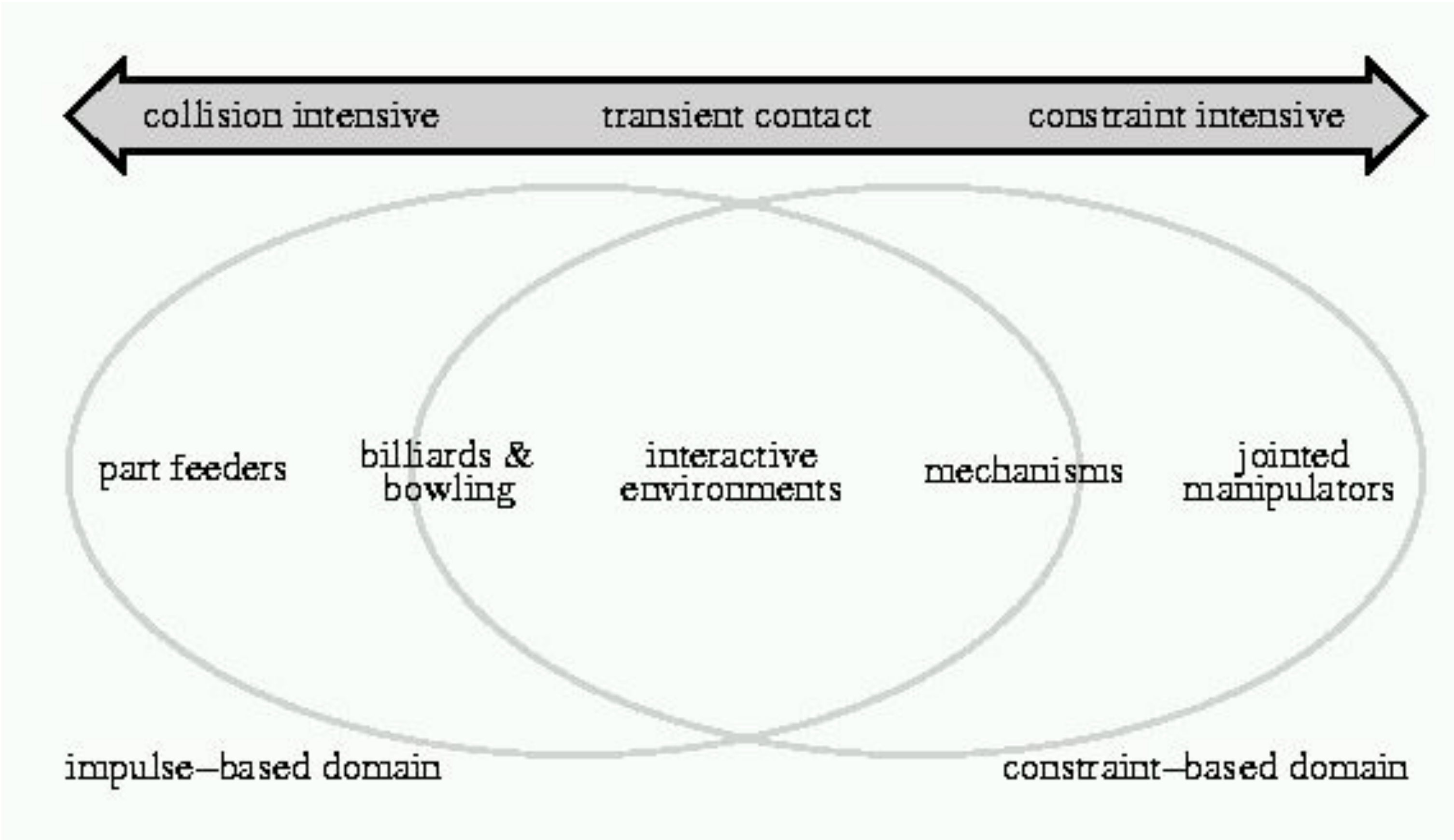
- Constraint-based dynamics (global)



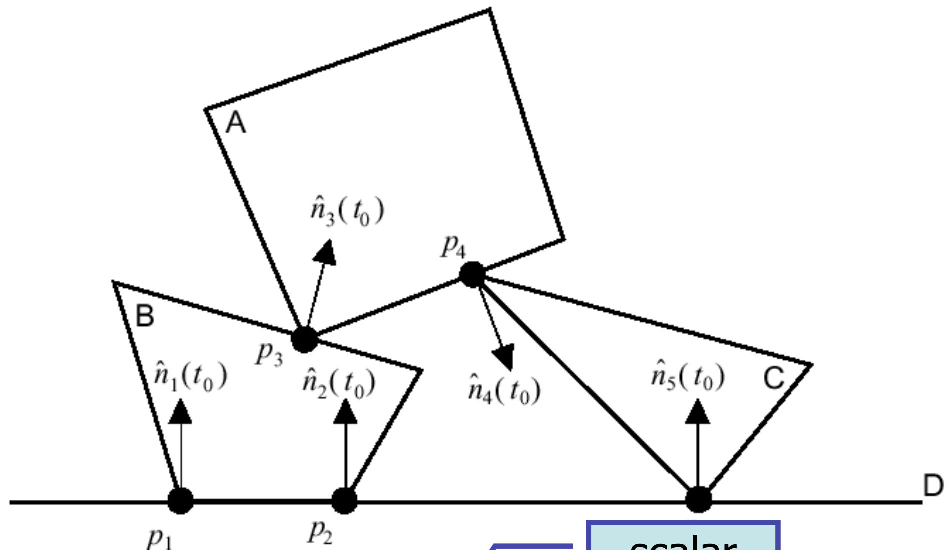
Impulse vs. Constraint

- Impulse-based dynamics (**local**)
  - Faster
  - Simpler
  - No explicit contact constraints
- Constraint-based dynamics (**global**)
  - Must declare each contact to be a resting contact or a colliding contact

Impulse vs. Constraint



Resting Contact Response



scalar

normal

At each contact:

- Apply normal force $f_i\hat{\mathbf{n}}_i$
- All forces computed simultaneously $\rightarrow$ linear system
- Forces subject to three conditions (see next slide)
- Define separation function $d_i(t)$

Resting Contact Response

- The **forces** at each contact must satisfy three criteria
  - **Prevent inter-penetration:** $\ddot{d}_i(t_0) \geq 0$
  - **Repulsive** -- we do not want the objects to be glued together: $f_i \geq 0$
  - Should become zero when the bodies start to **separate** (orthogonality): $f_i \ddot{d}_i(t_0) = 0$
- To implement hinges and pin joints:

$$\ddot{d}_i(t_0) = 0$$

Resting Contact Response

- We can formulate using LCP:

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

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

$$\begin{cases} \ddot{d}_i(t_0) \geq 0 \\ f_i \geq 0 \end{cases} \qquad f_i \ddot{d}_i(t_0) = 0$$

Linear Complimentary Problem (LCP)

- Need to solve a quadratic program to solve for the f_i's
  - General LCP is NP-complete problem
  - A is symmetric positive semi-definite (SPD) making the solution practically possible
- There is an iterative method to solve for without using a quadratic program

[Baraff, Fast contact force computation for nonpenetrating rigid bodies]

[Erin Catto, Sequential impulses]

Linear Complimentary Problem (LCP)

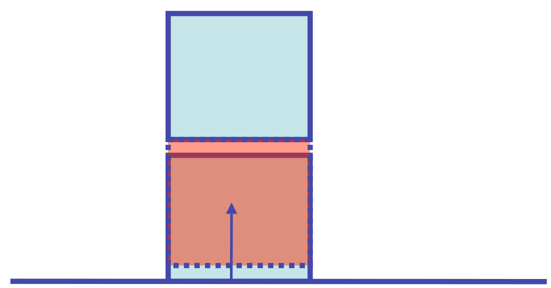
- In general, LCP can be solved with either:
  - **pivoting algos** (like Gauss elimination)
    - they change the matrix
    - do not provide useful intermediate result
    - may exploit sparsity well
  - **iterative algos** (like Conjugate Gradients)
    - only need read access to matrix
    - can stop early for approximate solution
    - faster for large matrices
    - can be warm started (ie. from previous result)

Global vs. local?

- **Global** LCP formulation can work for either constraint-based forces or with impulses
  - Hard problem to solve
  - System very often ill-conditioned, iterative LCP solver slow to converge

Local vs. Global

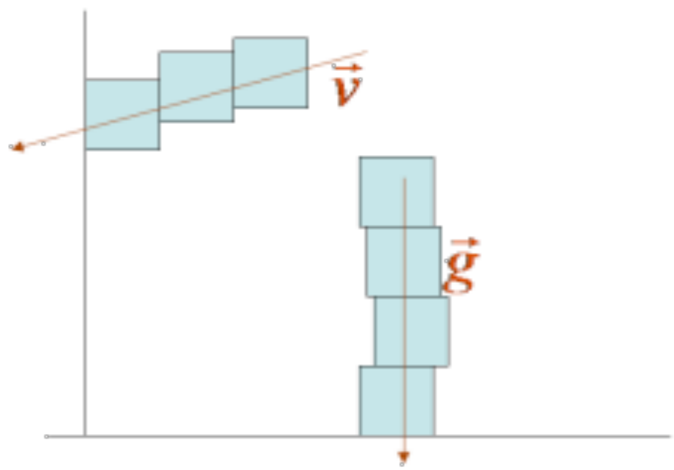
- Impulses often applied in **local** contact resolution scheme
- Applied impulses can break non-penetration constraint for other contacting points



- Often applied **iteratively**, until all resting contacts are resolved

Hard case for local approach

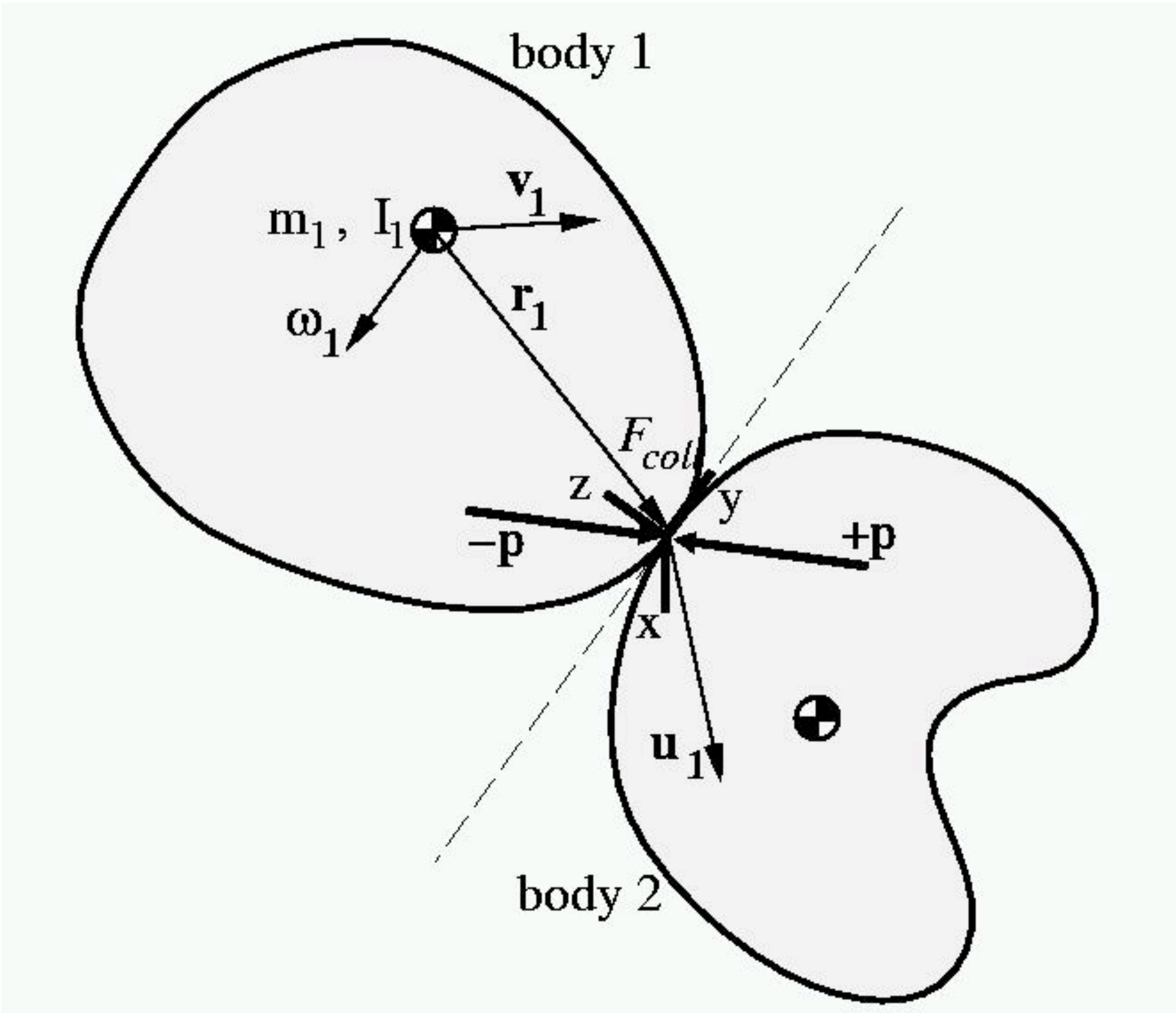
- **Prioritize** contact points along major axes of acceleration (gravity) and velocity
  - Performance improvement:
    25% on scene with 60 stacked objects



Frictional Forces Extension

- Constraint-based dynamics
  - Reformulate constraints and solve
  - This is an advantage for constraint-based dynamics!
- Impulse-based dynamics
  - Must not add energy to the system in the presence of friction
  - We will **integrate work** performed by contact impulses to track energy change

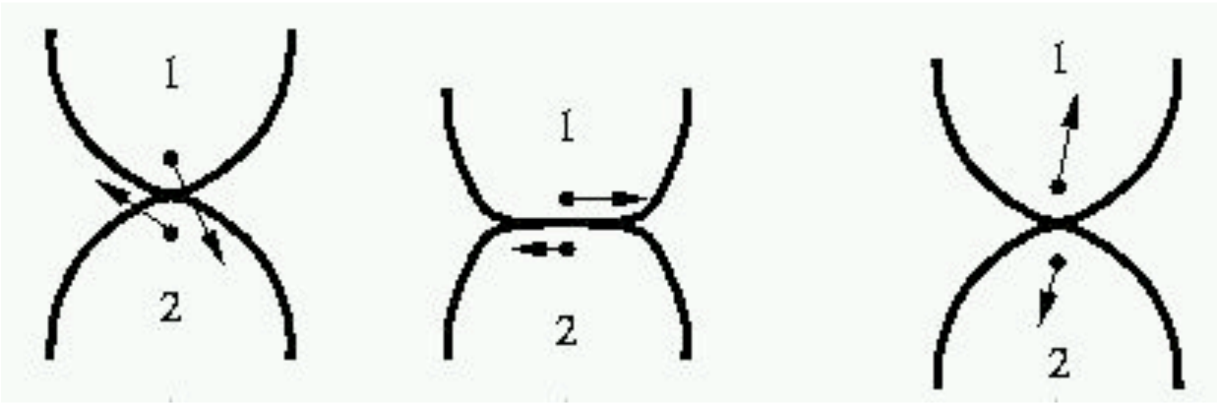
Collision Coordinate System



- **p** is the applied impulse. We use **j** because **P** is for linear momentum

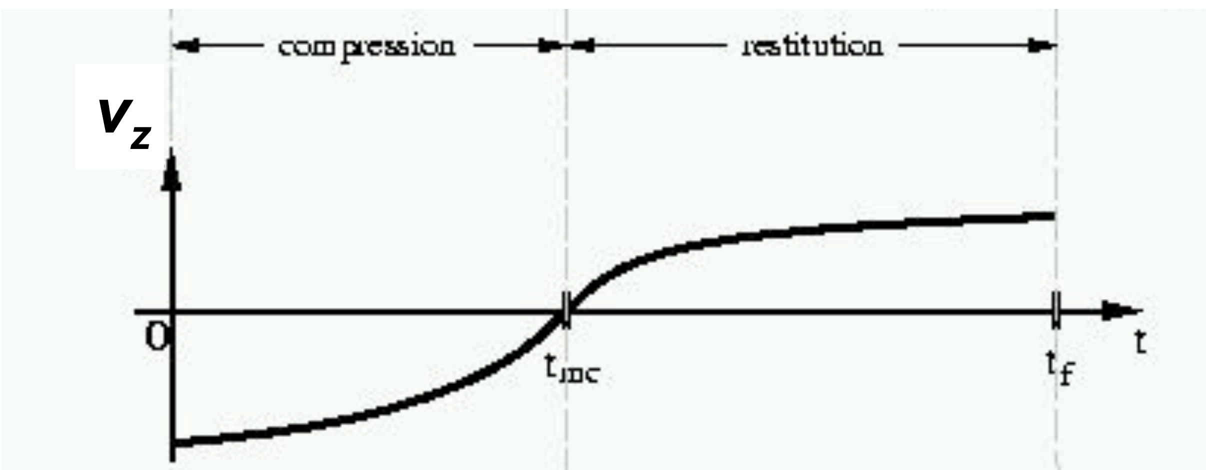
Impulse Reformulation

- When two real bodies collide there is a period of deformation during which elastic energy is stored in the bodies followed by a period of restitution during which some of this energy is returned as kinetic energy and the bodies rebound of each other.



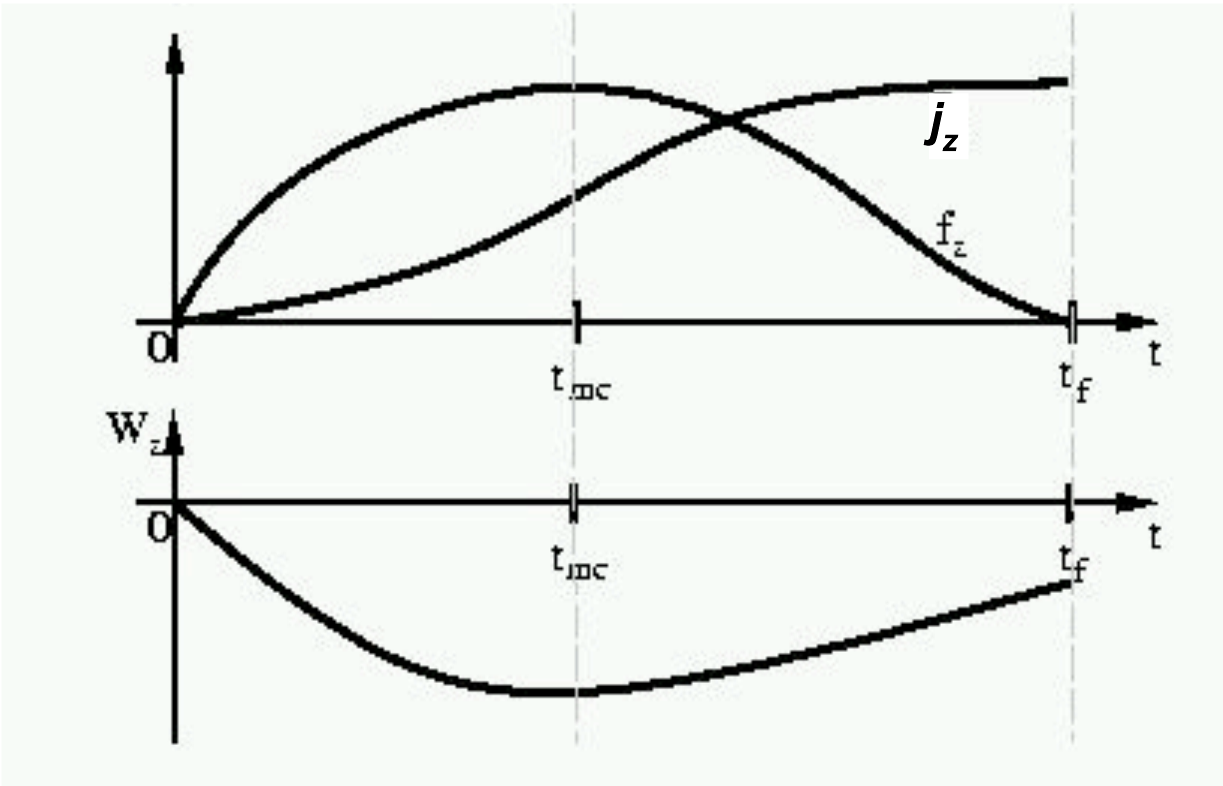
Impulse Reformulation

- The collision is instantaneous but we can assume that it occurs over a very small period of time: $0 \rightarrow t_{mc} \rightarrow t_f$.
- t_{mc} is the time of maximum compression



v_z is the relative normal velocity.

Impulse Reformulation



- j_z is the impulse magnitude in the normal direction.
- W_z is the work done in the normal direction.

Impulse Reformulation (I)

- **Newton's Empirical Impact Law:**

  Coefficient of restitution ε relates before-collision to after-collision relative velocity

- **Poisson's Hypothesis:**

  The normal component of impulse delivered during restitution phase is ε times the normal component of impulse delivered during the compression phase

Both these hypotheses can cause increase of energy when friction is present!

Impulse Reformulation (II)

- Stronge's Hypothesis:

The positive work done during the restitution phase is $-\varepsilon^2$ times the negative work done during compression

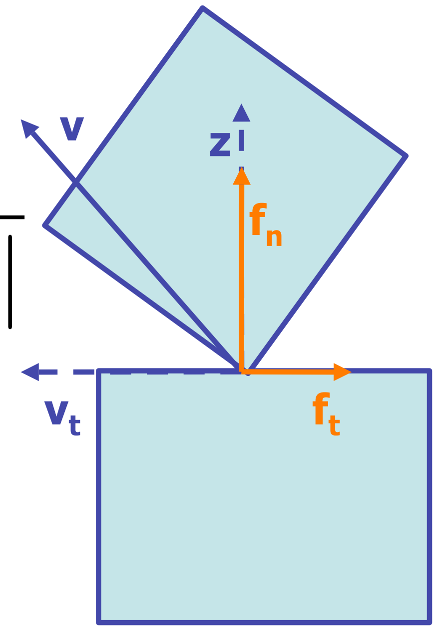
$$\begin{aligned} W_z^+ - W_z^0 &= -\epsilon^2 W_z^0 \\ W_z^+ &= (1-\epsilon^2) W_z^0 \end{aligned}$$

Energy of the bodies does not increase when friction present

Coulomb Friction model

- **Sliding (dynamic) friction**

$$\mathbf{v}_t \neq 0 \Rightarrow \mathbf{f}_t = -\mu \|\mathbf{f}_n\| \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|}$$



- **Dry (static) friction**

$$\mathbf{v}_t = 0 \Rightarrow \mathbf{f}_t \leq \mu \|\mathbf{f}_n\|$$

(i.e. the friction cone)

- Assume no rolling friction

Impulse with Friction

- Recall that the impulse looked like this for frictionless collisions:

$$j = \frac{-(1+\epsilon)v_{rel}^{-}}{\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0)\cdot\left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \hat{n}(t_0)\cdot\left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b}$$

$$p(t) = \int_0^t f(\tau)\,d\tau$$

- Remember: $p_z(t) = j(t)$
- Recall also that $\Delta v_z = j/M$ and $\Delta L = r \times j^T n$
- All are parameterized by time

Impulse with Friction

$$\Delta \mathbf{v}_t = \left[\left(\frac{1}{m_1} + \frac{1}{m_1} \right) - (\mathbf{r}_1^\star I_1^{-1} \mathbf{r}_1^\star + \mathbf{r}_2^\star I_2^{-1} \mathbf{r}_2^\star) \right] \mathbf{j}(t) = \mathbf{K}\mathbf{j}(t)$$

$$\Delta \mathbf{v}_t = \mathbf{K}\mathbf{j}(t)$$

where:

- **r** = (p-x) is the vector from the center of mass to the contact point

$$r^* = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

The K Matrix

- K is constant over the course of the collision, nonsingular, symmetric, and positive definite

$$K = \begin{bmatrix} k_x & & \\ & k_y & \\ & & k_z \end{bmatrix}$$

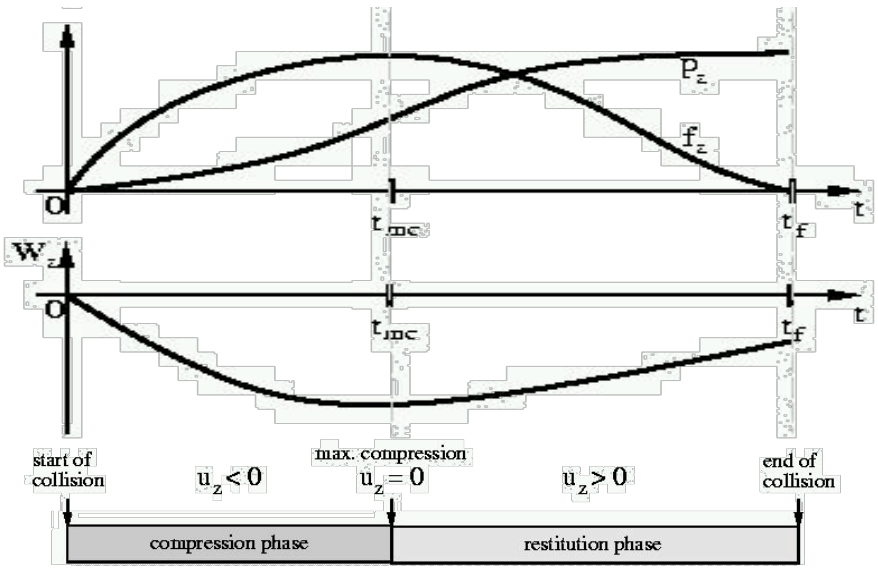
Collision Functions

- We assume collision to occur over zero time interval → velocities discontinuous over time
  - Discontinuities **bad for integration**!
- Reparameterize Δv(t) = **K** j(t) from ***t*** to γ
- Take γ such that it is monotonically increasing during the collision: $\Delta\mathbf{v}(\gamma) = \mathbf{K}\mathbf{j}(\gamma)$
- Let the duration of the collision → 0.
- The functions ***v***, ***j***, ***W***, all evolve *continuously* over the compression and the restitution phases with respect to γ.

Sliding Formulation

- For the **compression phase**, use $\gamma = v_z$
  - v_z is the relative normal velocity at the start of the collision (we know this)
  - At the end of the compression phase, $v_z^0=0$
- For the **restitution phase**, use $\gamma = W_z$
  - W_z^0 is the amount of work that has been done in the compression phase
  - From Stronge's hypothesis, we know that

$$W_z^+ \quad = \quad (1 - \epsilon^2) W_z^0$$

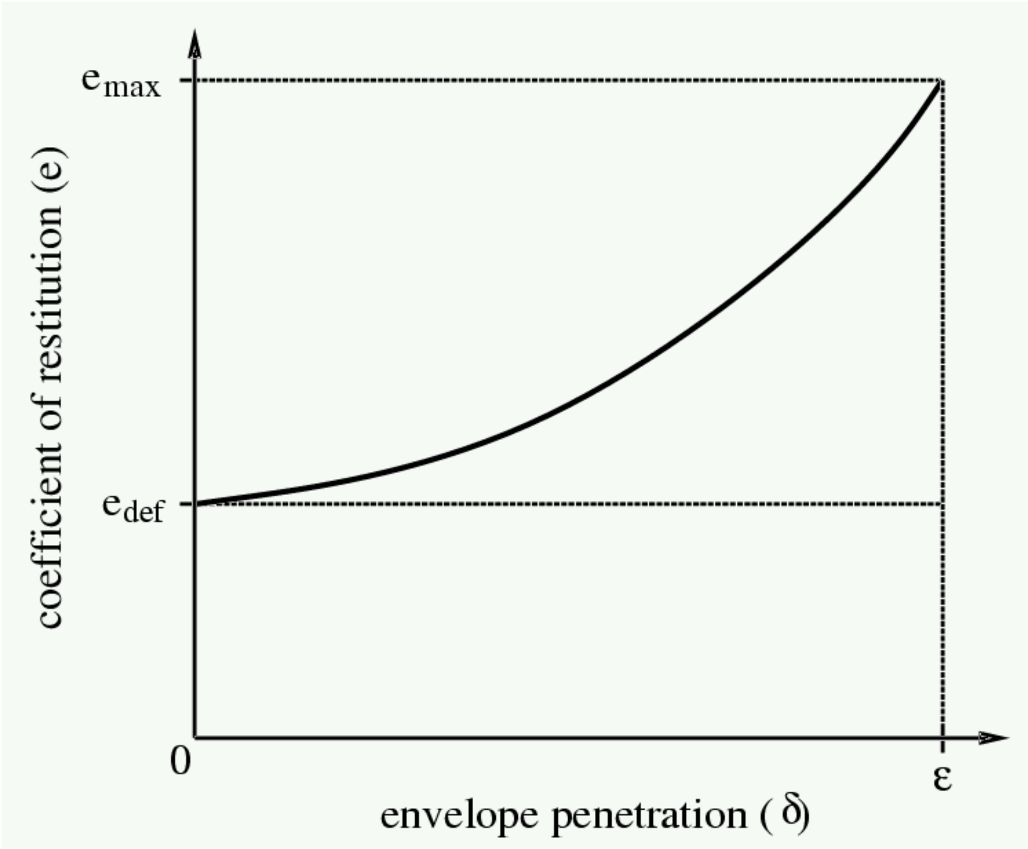


Resting Contacts with Impulses

- Modeled by artificial train of collisions
- The resulting collision impulses model a constant reaction force (doesn’t work for stationary objects)
- Problem: book on table: through collisions, energy steadily decreases, book sinks into table
- #of collisions increases, simulator comes to grinding halt!
- Introduce micro-collisions
  - Micro-collision impulses are not computed in the standard way, but with **artificial coefficient of restitution** $e(\delta)$
  - Applied only if normal velocity is ‘small’

Artificial restitution for

- e = f(Distance(A,B))



Micro-collisions issues

- Other problems arise:
  - Boosted elasticity from micro-collisions makes box on ramp ‘bounce’ as if ramp were vibrating
  - Stacked books cause too many collision impulses, propagated up and down the stack
  - Weight of pile of books causes deep penetration between table and bottom book → large reaction impulses cause instabilities
- Micro-collisions are an ad-hoc solution!
- Constrained-based approaches are a better solution for these situations