Rigid Body Dynamics (II)

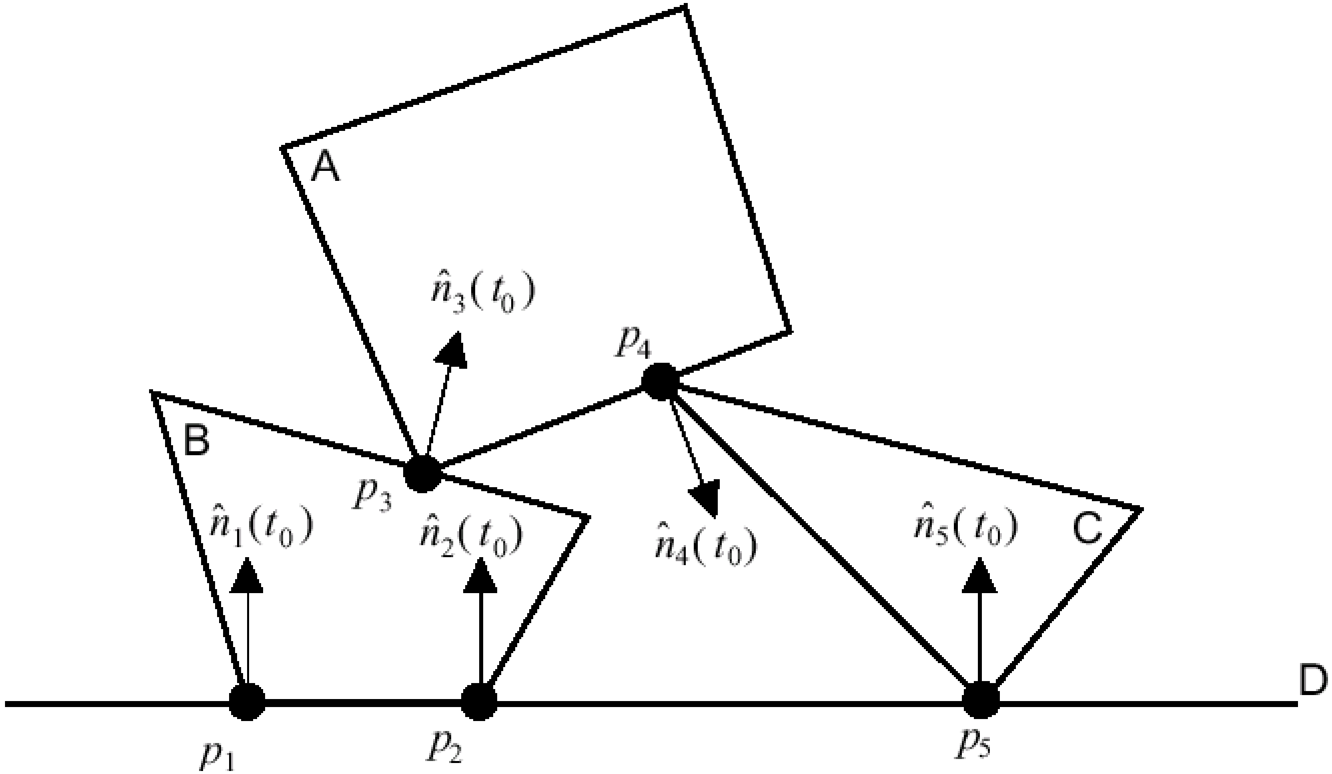
COMP259 March 30, 2006

Nico Galoppo von Borries

Bodies intersect ! classify contacts

- Bodies separating
  - $v_{rel} > \varepsilon$
  - No response required
- Colliding contact (Last time)
  - $v_{rel} < -\varepsilon$
- Resting contact (Today)
  - $-\varepsilon < v_{rel} < \varepsilon$
  - 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

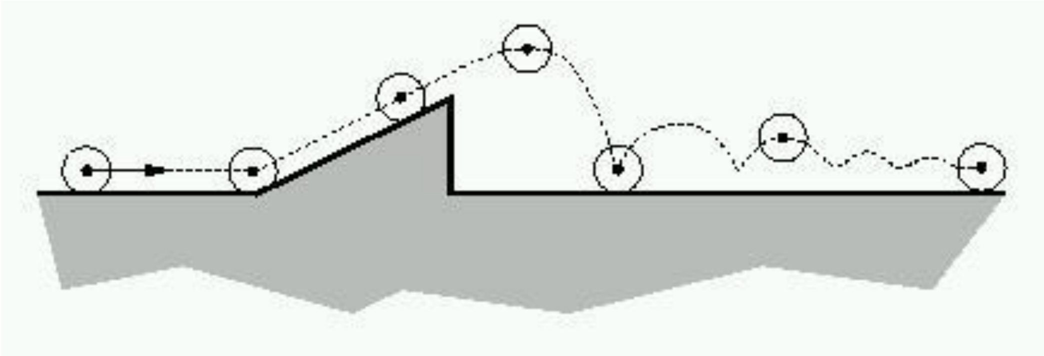
- Global vs. Local Methods
  - Constraint-based vs. Impulse-based
- Colliding Contacts
- Resting Contacts
- Force application
- Friction

Impulse vs. Constraint

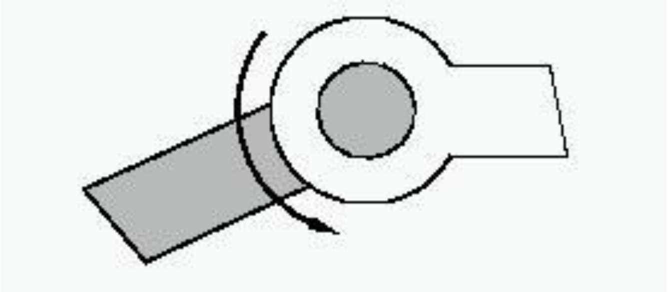
- Impulse-based dynamics (local)
  - Faster
  - Simpler
  - No explicit contact constraints
- Constraint-based dynamics (global)
  - Must declare each contact to be a resting contact or a colliding contact

Impulse vs. Constraint

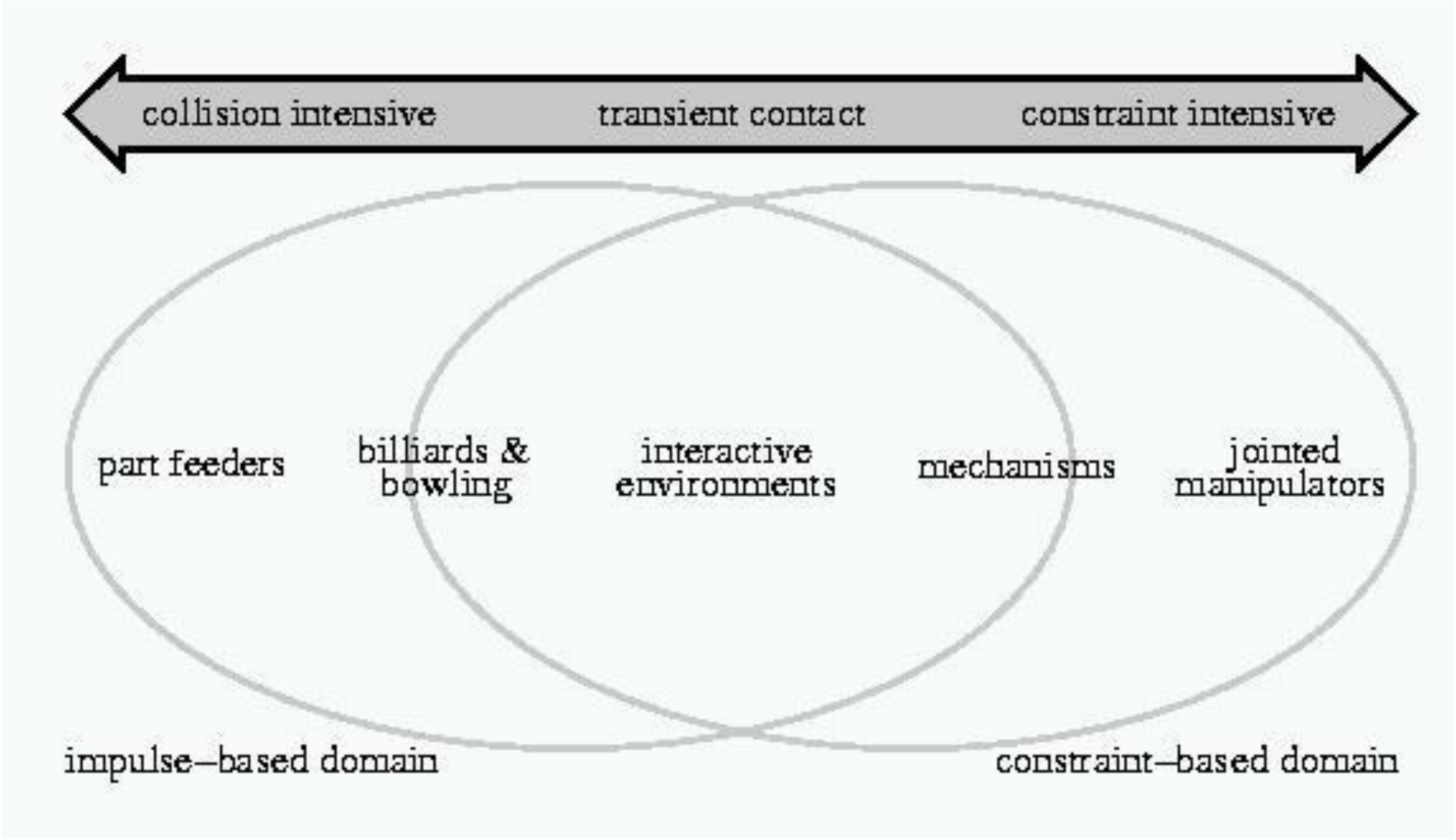
- Impulse-based dynamics (local)



- Constraint-based dynamics (global)



Impulse vs. Constraint



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: $f_i \ddot{d}_i(t_0) = 0$
- 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*]

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)

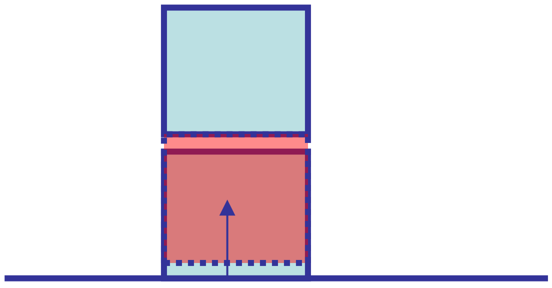
Slide courtesy of Moravanszky (ETHZ 2002)

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

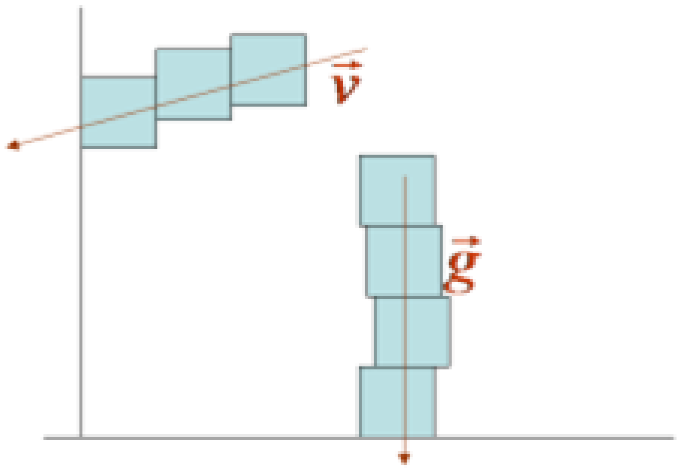
- Impulses often applied in **local** contact resolution scheme
- Applied impulses can break non-penetration constraint for other contacting points



- 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*



Global Resting Contact Resolution

X Ã InitializeState()

For t=t_0 to t_{final} with step Δt

ClearForces(F(t), τ(t))
AddExternalForces(F(t), τ(t))
X_{new} Ã Solver::Step(X, F(t), τ(t), t, Δt)

t Ã findCollisionTime()
X_{new} Ã Solver::Step(X, F(t), τ(t), t, Δt)

C Ã Contacts(X_{new})
while (!C.isColliding())
applyImpulses(X_{new})
end if

X Ã X_{new}
t Ã t + Δt

End for

C Ã Contacts(X_{new})

while (!C.isColliding())

applyImpulses(X_{new})

end if

restingC Ã C.restingSet()

solveLCP(restingC, X_{new})

X Ã X_{new}

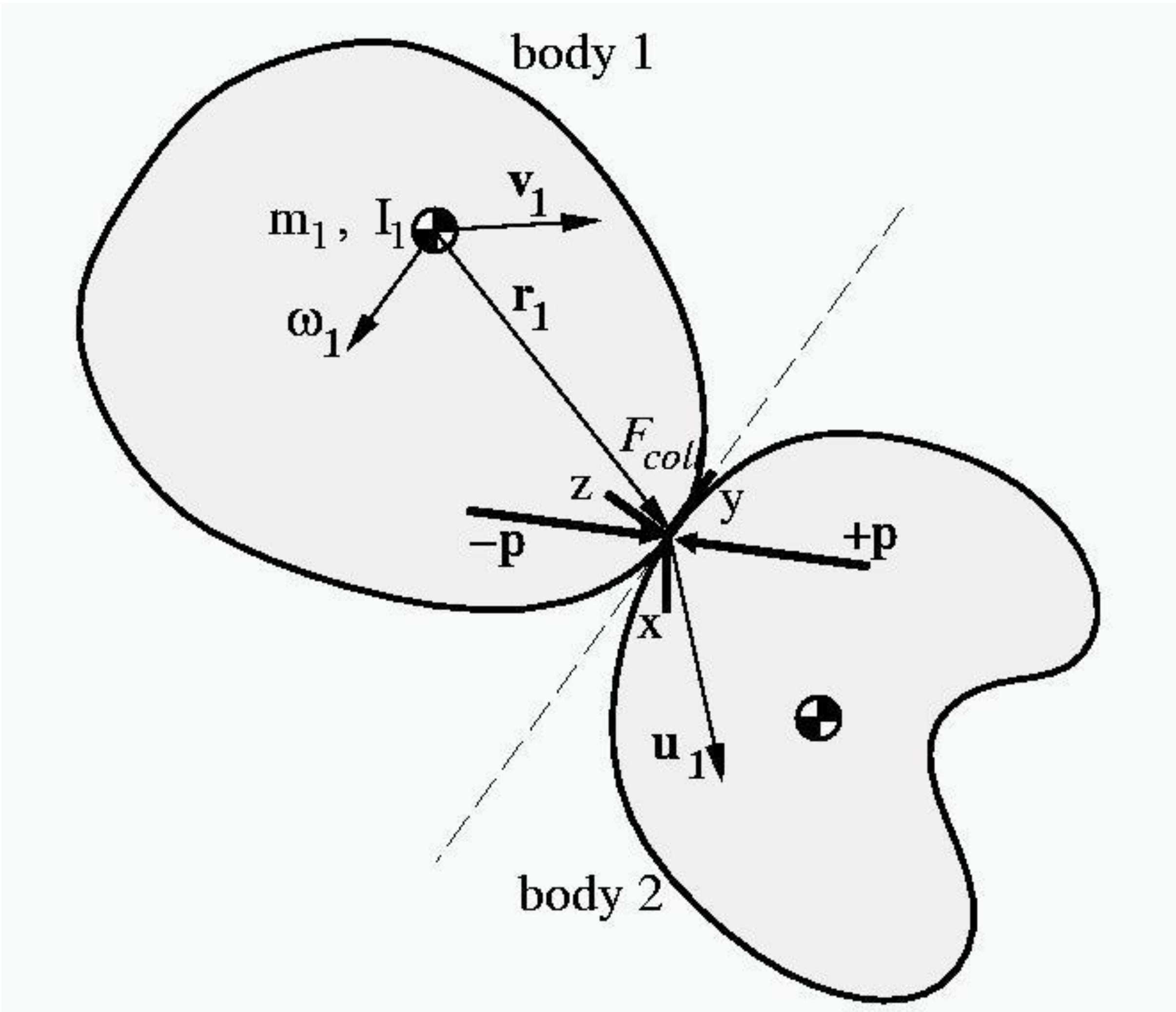
t Ã t + Δt

End for

Frictional Forces Extension

- Constraint-based dynamics
  - Reformulate constraints and solve
  - No more on this here
- Impulse-based dynamics
  - Must not add energy to the system in the presence of friction so we have to reformulate the impulse to be applied

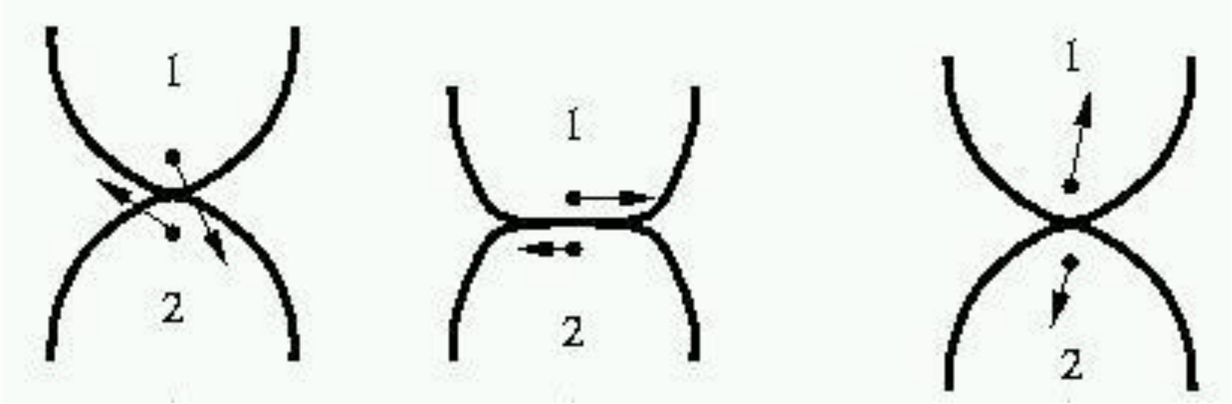
Collision Coordinate System



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

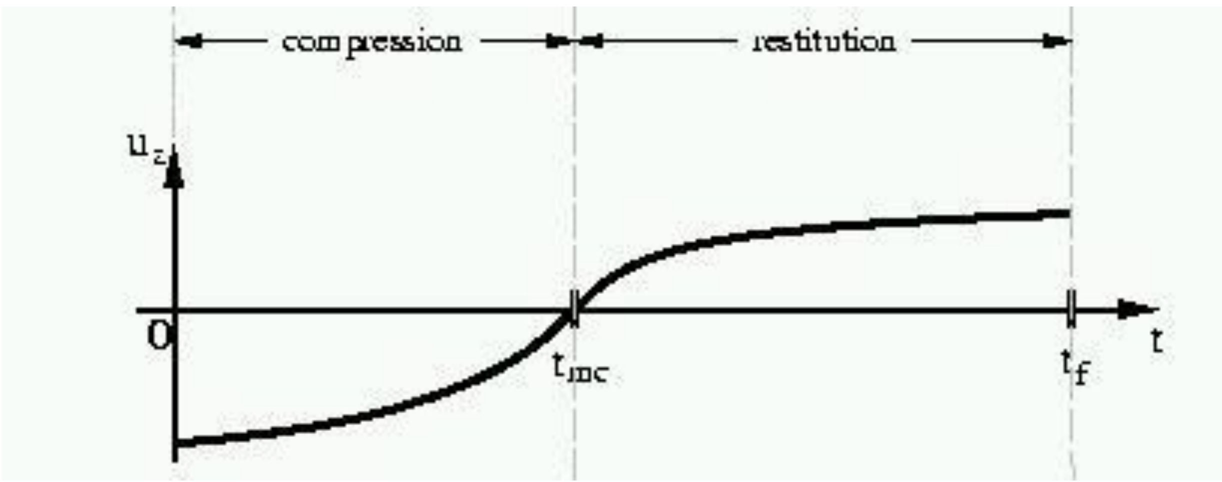
Impulse Reformulation

- 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.



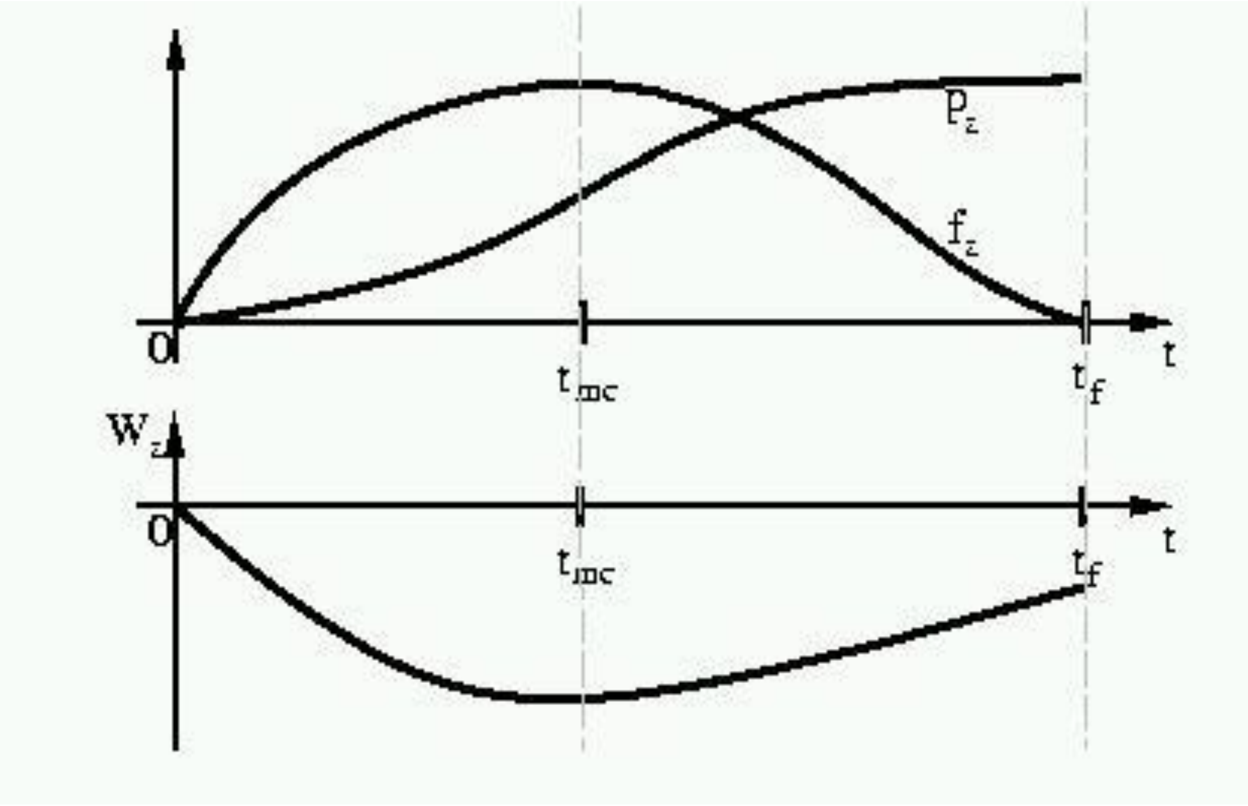
Impulse Reformulation

- 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



$\boldsymbol{u_z}$ is the relative normal velocity. We used $\boldsymbol{v_{rel}}$ before. From now on we will use $\boldsymbol{v_z}$.

Impulse Reformulation



- ***p_z*** is the impulse magnitude in the normal direction. We used ***j*** before. From now on we will use ***j_z***.
- ***W_z*** is the work done in the normal direction.

Impulse Reformulation (I)

- $v^{-}=v(0)$, $v^{0}=v(t_{mc})$, $v^{+}=v(t_f)$, $v_{rel}=v_z$
- **Newton's Empirical Impact Law:**

$$v_z^+ = -\varepsilon v_z^-$$

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

- **Poisson's Hypothesis:** $\begin{cases} j_z^+ - j_z^0 = \varepsilon j_z^0 \\ j_z^+ = (1+\varepsilon) j_z^0 \end{cases}$

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*

$$\begin{cases} W_z^+ - W_z^0 = -\varepsilon^2 W_z^0 \\ W_z^+ = (1-\varepsilon^2) W_z^0 \end{cases}$$

Energy of the bodies does not increase when friction present

Friction Formulae

- Assume the Coulomb friction law:
  - At some instant during a collision between bodies 1 and 2, let $\boldsymbol{v}$ be the contact point velocity of body 1 relative to the contact point velocity of body 2. Let $\boldsymbol{v_t}$ be the tangential component of $\boldsymbol{v}$ and let $\hat{\boldsymbol{v}}_t$ be a unit vector in the direction of $\boldsymbol{v_t}$. Let $\boldsymbol{f_z}$ and $\boldsymbol{f_t}$ be the normal and tangential (frictional) components of force exerted by body 2 on body 1, respectively.

Coulomb Friction model

- **Sliding (dynamic) friction**

$$v_t \neq 0 \Rightarrow f_t = -\mu \|f_n\| \hat{v}_t$$

- **Dry (static) friction**

$$v_t = 0 \Rightarrow \|f_t\| \leq \mu \|f_n\|$$

(ie. 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 $\Delta L = r\pounds j^T n$
- All are parameterized by time

Impulse with Friction

$$\Delta v(t) = \left[\left(\frac{1}{m_1}+\frac{1}{m_2}\right)1 - \left(\tilde{r}_1 I_1^{-1}\tilde{r}_1 + \tilde{r}_2 I_2^{-1}\tilde{r}_2\right)\right] j(t) = Kj(t)$$

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

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

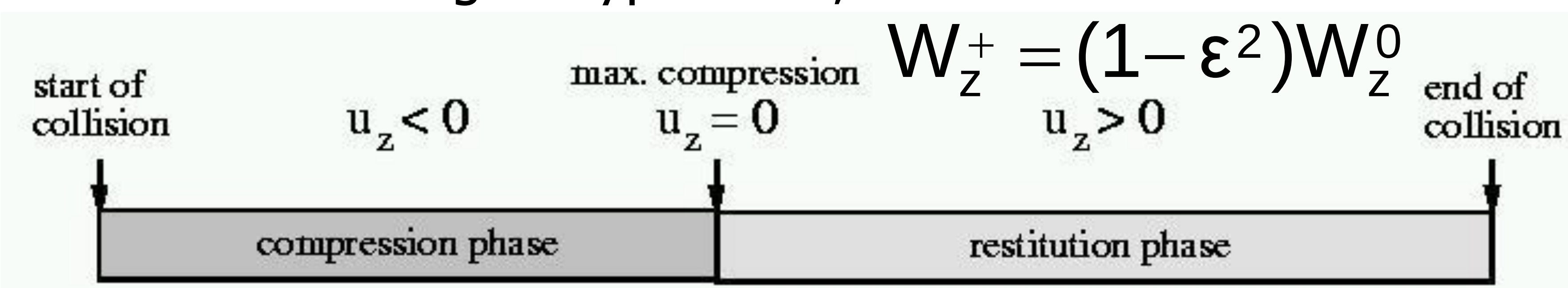
Collision Functions

- We assume collision to occur over zero time interval ! velocities discontinuous over time
- Reparameterize $\Delta v(t) = K\, j(t)$ from ***t*** to γ
- Take γ such that it is monotonically increasing during the collision: $\Delta \mathrm{v}(\boldsymbol{\gamma}) = \mathrm{Kj}(\boldsymbol{\gamma})$
- 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

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



Sliding Formulation

- Compression phase equations are:

$$\frac{d}{dv_z}\begin{bmatrix} v_x \\ v_y \\ W_z \end{bmatrix} = \frac{1}{k_z \xi(\theta)} \begin{bmatrix} k_x \xi(\theta) \\ k_y \xi(\theta) \\ v_z \end{bmatrix}$$

Sliding Formulation

- Restitution phase equations are:

$$\frac{d}{dW_z}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{1}{v_z} K\,\xi(\theta) = \frac{1}{v_z}\begin{bmatrix} k_x \xi(\theta) \\ k_y \xi(\theta) \\ k_z \xi(\theta) \end{bmatrix}$$

Sliding Formulation

where the sliding vector is:

$$\xi(\theta) = \begin{bmatrix} -\mu\cos\theta \\ -\mu\sin\theta \\ 1 \end{bmatrix} = \begin{bmatrix} -\mu v_x \Big/ \sqrt{v_x^2 + v_y^2} \\ -\mu v_y \Big/ \sqrt{v_x^2 + v_y^2} \\ 1 \end{bmatrix}$$

Sliding Formulation

- Notice that there is a problem at the point of maximum compression because $\boldsymbol{v_z} = 0$:

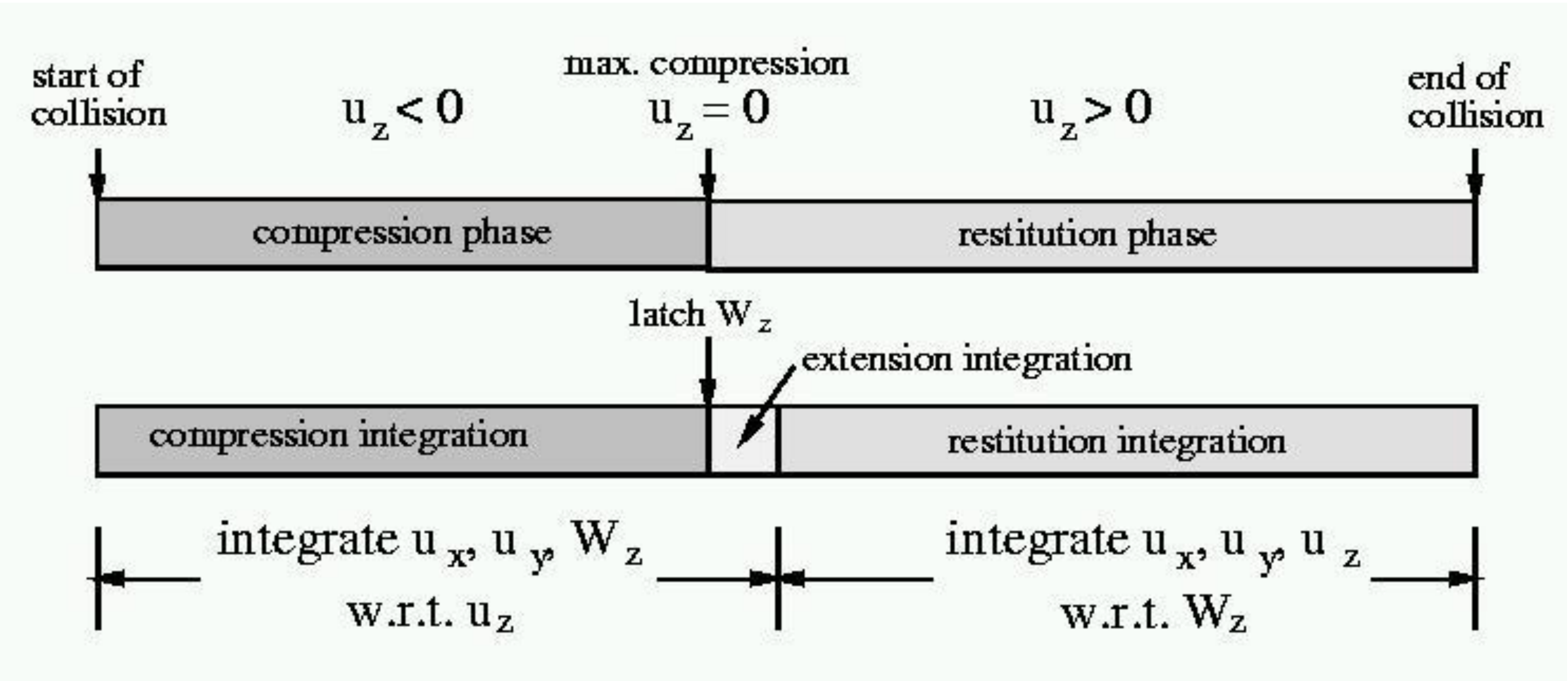
$$\frac{d}{dW_z}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{1}{\boxed{v_z}} K\,\xi(\theta) = \frac{1}{\boxed{v_z}}\begin{bmatrix} k_x\xi(\theta) \\ k_y\xi(\theta) \\ k_z\xi(\theta) \end{bmatrix}$$

Sliding Formulation

- Let us integrate using $\boldsymbol{v_z}$ a bit into the restitution phase (extension integration) so that we never divide by 0.
- But we don't want to integrate too far! Otherwise we exceed the amount of work that is to be done in the restitution phase.
- We are safe if we stop at:

$$v_z = \sqrt{2\left(W_z^{+} - W_z^{0}\right)\left(K_{33} - \mu\sqrt{K_{31}^2 + K_{32}^2}\right)}$$

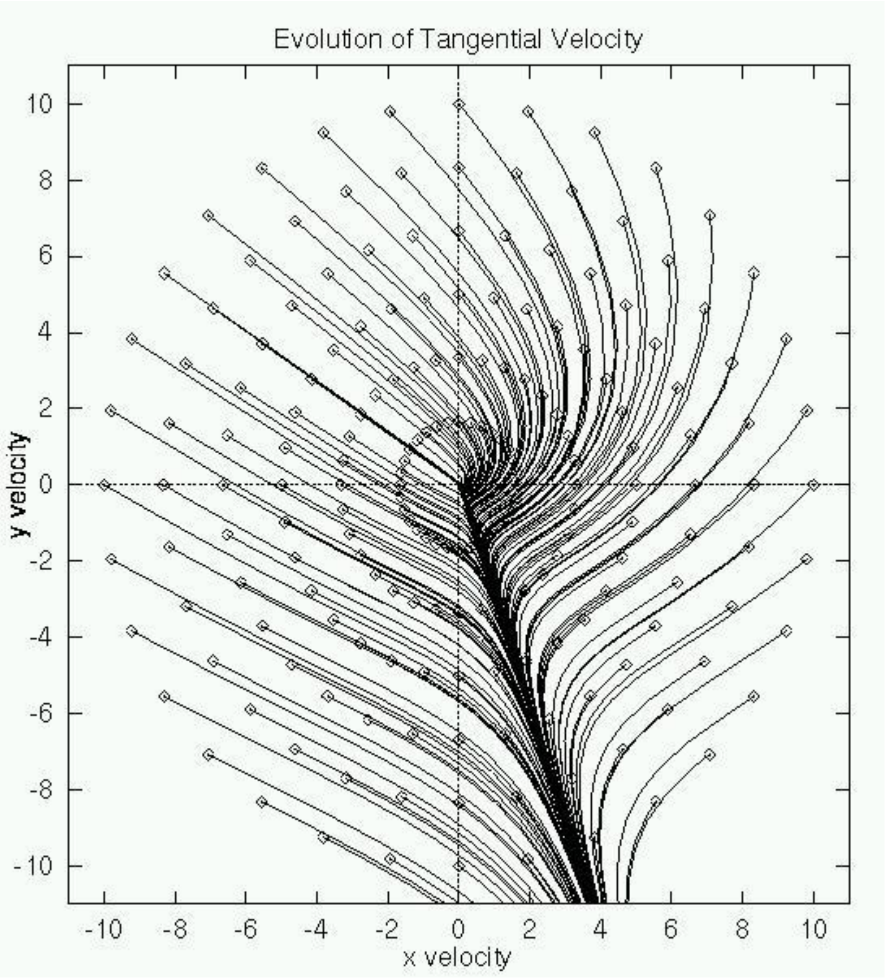
Sliding Formulation



Sliding Formulation

- There is another problem if the tangential velocity becomes 0 because the equations that we have derived were based on
$v_t \neq 0 \Rightarrow f_t = -\mu \|f_n\| \hat{v}_t$ which no longer holds.
- This brings us to the sticking formulation

Sticking Formulation



Sticking Formulation

- Stable if $\left(K_{13}^{-1}\right)^2 + \left(K_{23}^{-1}\right)^2 \le \mu^2 \left(K_{33}^{-1}\right)^2$
  - This means that static friction takes over for the rest of the collision and $\boldsymbol{v_x}$ and $\boldsymbol{v_y}$ remain 0
- If instable, then in which direction do $\boldsymbol{v_x}$ and $\boldsymbol{v_y}$ leave the origin of the $\boldsymbol{v_x}$, $\boldsymbol{v_y}$ plane?
  - There is an equation in terms of the elements of K which yields 4 roots. Of the 4 only 1 corresponds to a diverging ray – a valid direction for leaving instable sticking.

Sticking Formulation

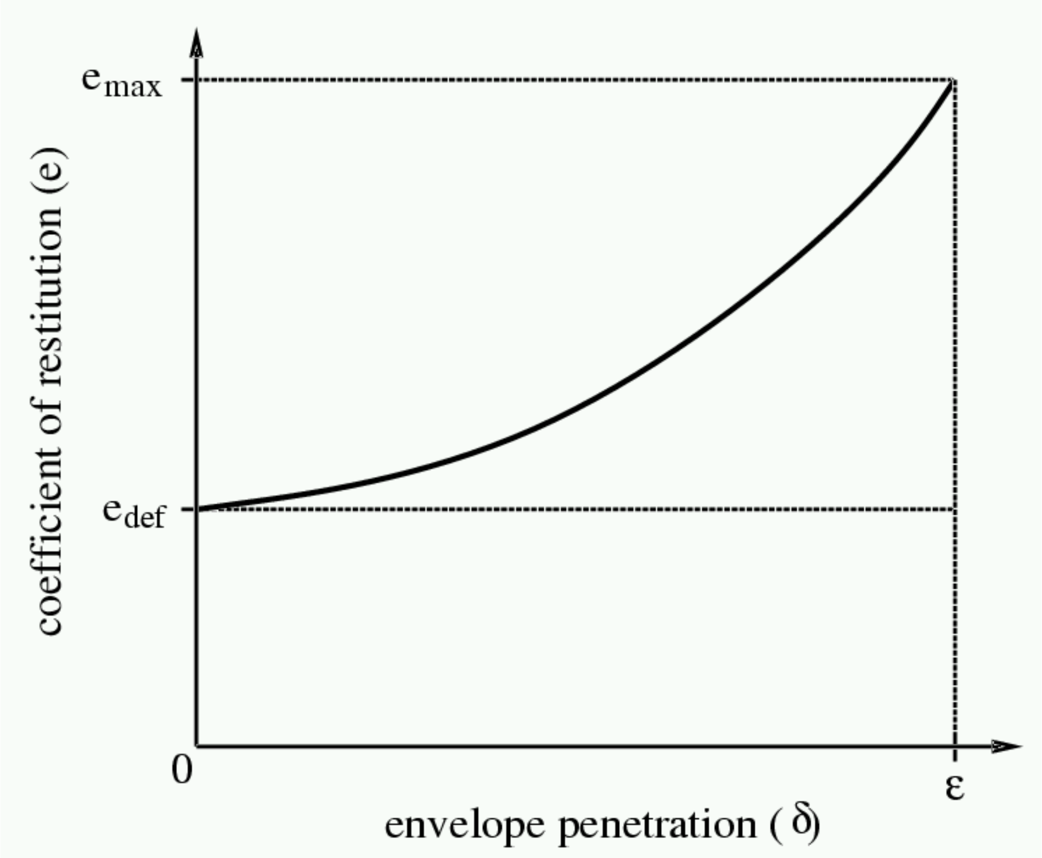
- If sticking occurs, then the remainder of the collision may be integrated analytically due to the existence of closed form solutions to the resulting simplified equations.

Resting Contacts with Impulses

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

Artificial restitution for microcollisions

- ϵ = f(Distance(A,B))



- Other problems arise:
  - Boosted elasticity from microcollisions 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
- Microcollisions are an ad-hoc solution!
- Constrained-based approaches are a better solution for these situations