CS**5643**

10 Resolving systems of collisions

Steve Marschner
Cornell University
Spring 2025

Overview

How systems of collisions arise

- resting contact
- deformable vs. rigid

1: resolving systems of collisions with particles

- kinematics of 3DOF per object, friction makes no sense
- establishes problem structure in simpler setting

2: resolving systems of frictionless collisions with rigid bodies

- similar to (1) but with kinematics that has position and orientation

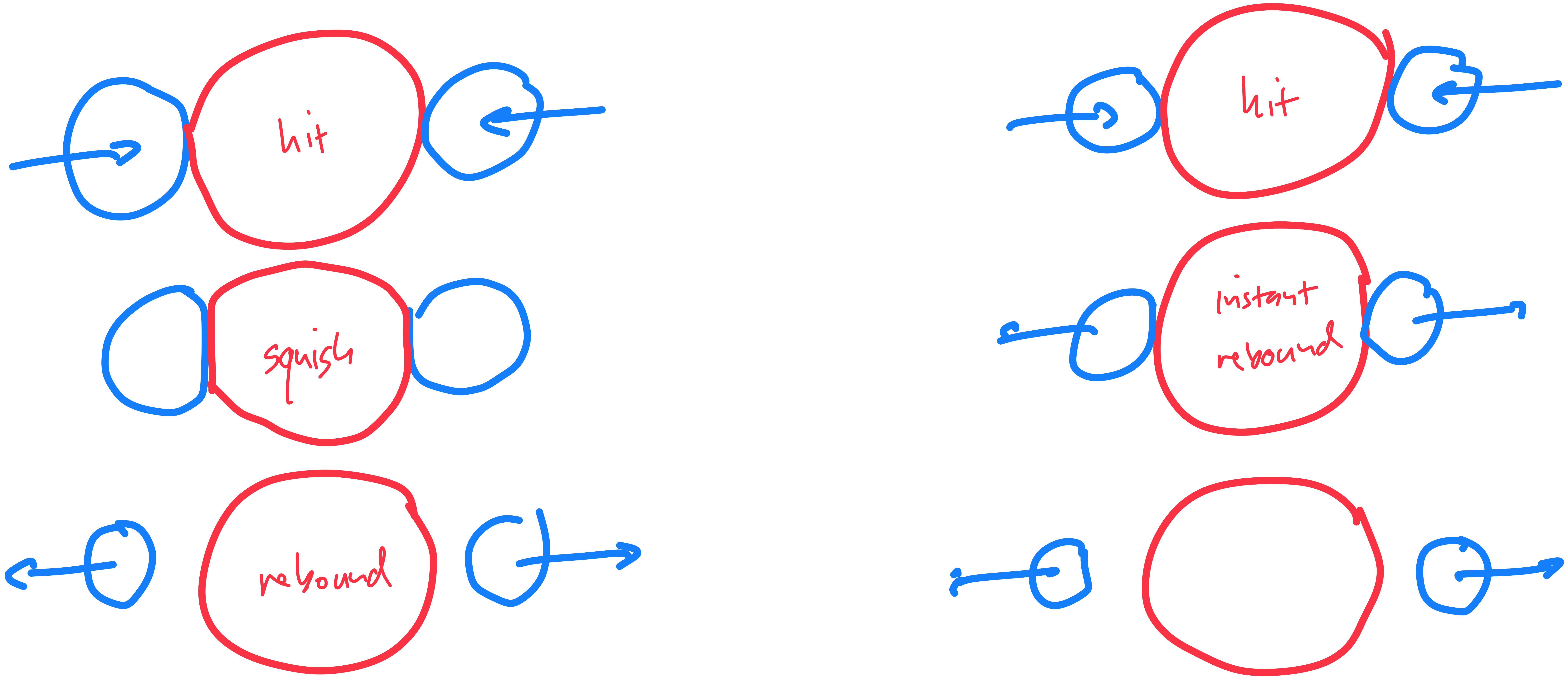
3: resolving systems of collisions with friction (rigid bodies)

- reuses similar machinery to (2) to also solve for frictional forces

Resolving a system of coupled collisions

Sometimes many collisions are coupled together at a single time

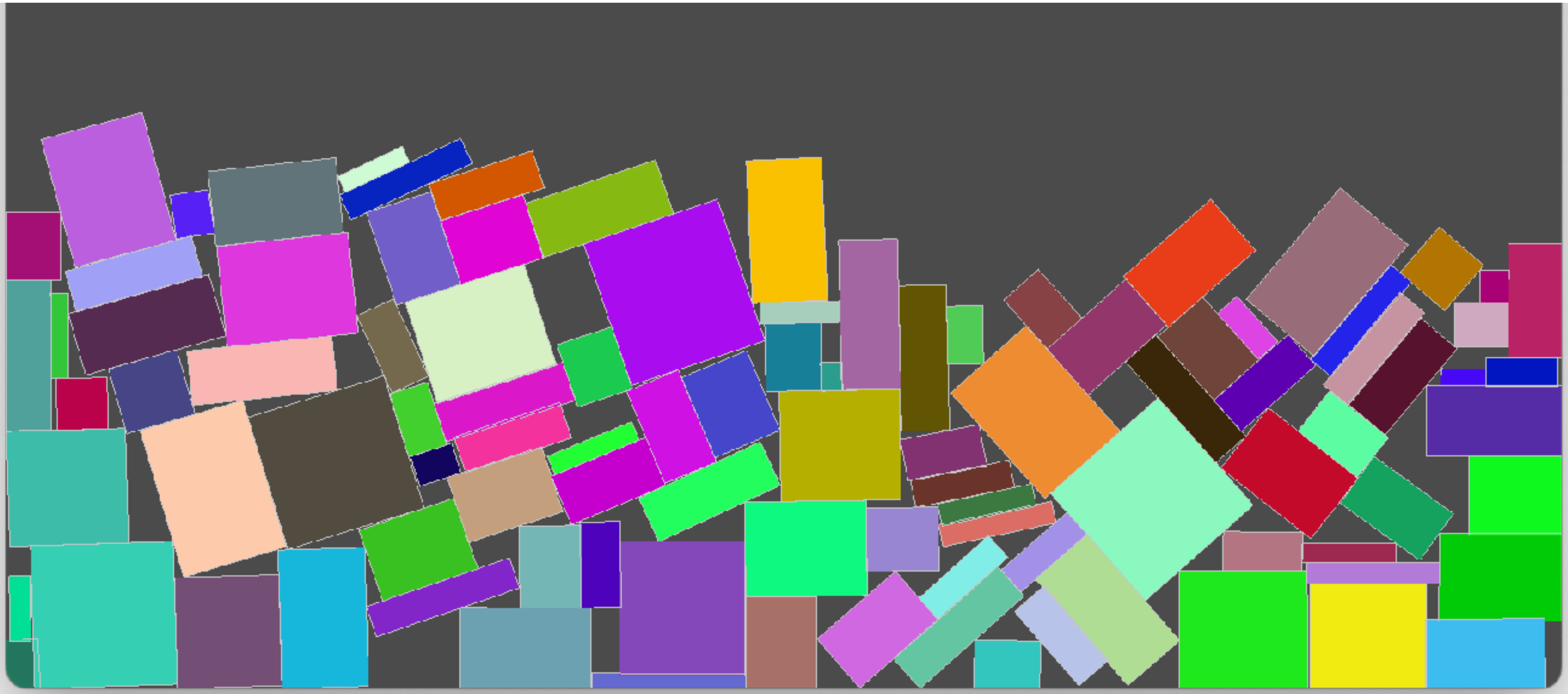
- deformable objects insulate contacts from one another
- rigid objects transmit impulses instantly



Common case: resting contact

In the presence of gravity, objects end up piled up

- contacts persist over time
- large systems of coupled contacts are unavoidable
- sequential resolution does not scale



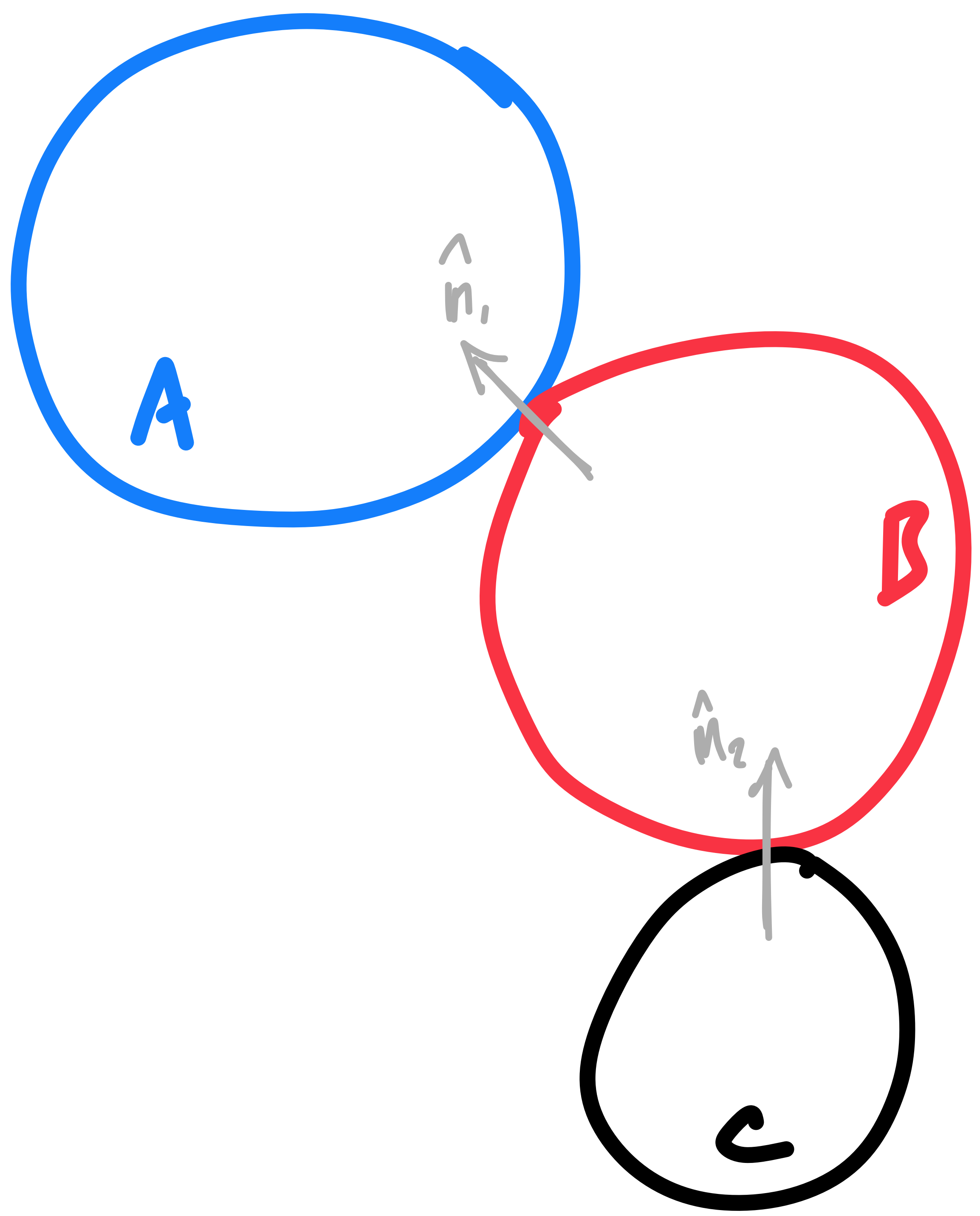
One collision in the context of another

Suppose an object is involved in two simultaneous collisions

- one we are computing the impulse for
- someone has told us the impulse for the other one

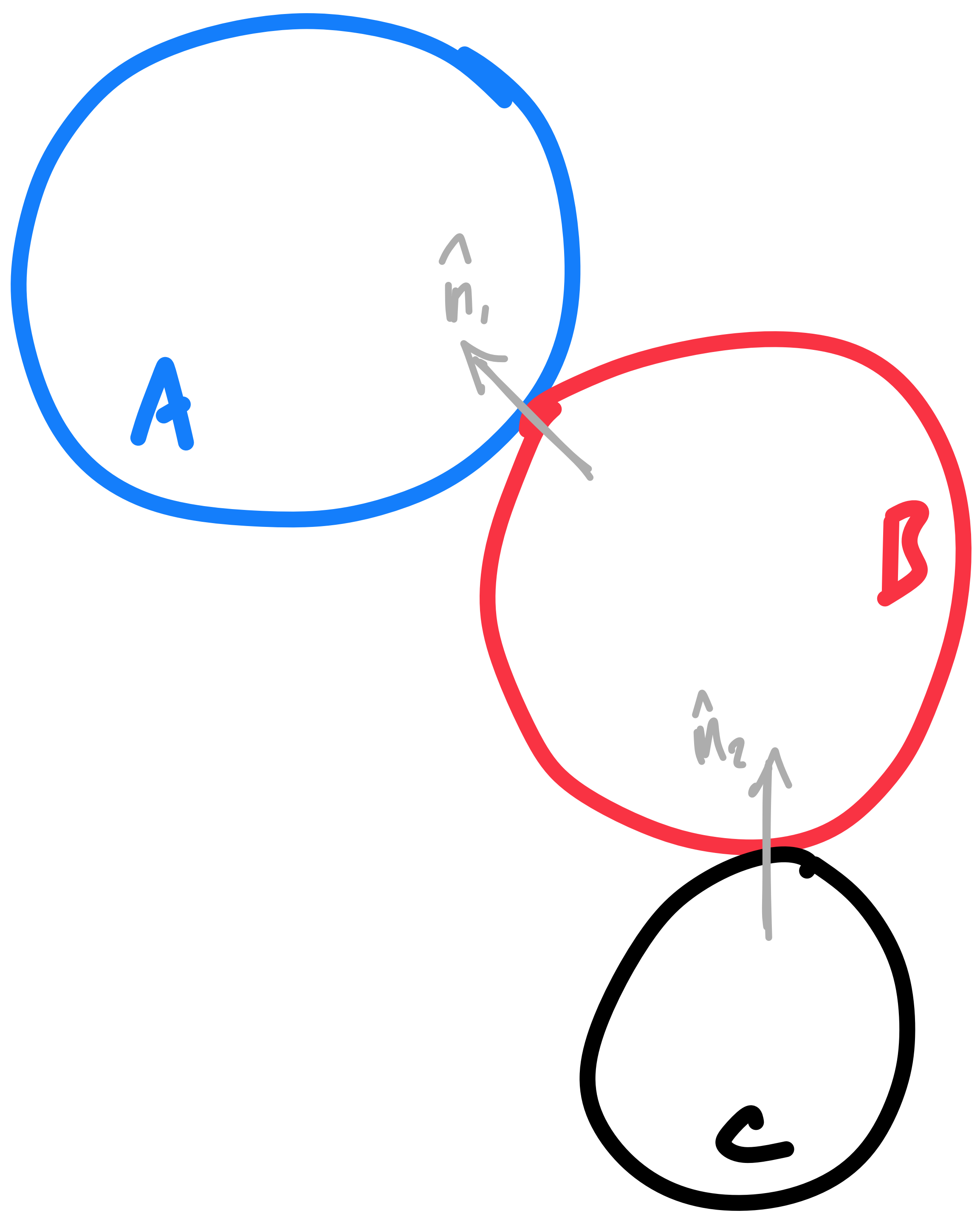
Call the objects A and B, the collisions 1 and 2

- pre-collision velocities $\mathbf{v}_a^-$ and $\mathbf{v}_b^-$; post-collision $\mathbf{v}_a^+$ and $\mathbf{v}_b^+$
- collision normals $\mathbf{n}_1$ and $\mathbf{n}_2$
- restitution hypothesis: $v_1^+ = -c_r v_1^-$ where $v_1 = \mathbf{n}_1 \cdot (\mathbf{v}_a - \mathbf{v}_b)$
- collision impulses are $\gamma_1 \mathbf{n}_1$ (unknown) and $\gamma_2 \mathbf{n}_2$ (known)



One collision in the context of another

- velocities after collision
  - $\mathbf{v}_a^+ = \mathbf{v}_a^- + m_a^{-1}\gamma_1\mathbf{n}_1$
  - $\mathbf{v}_b^+ = \mathbf{v}_b^- - m_b^{-1}\gamma_1\mathbf{n}_1 + m_b^{-1}\gamma_2\mathbf{n}_2$
  - $v_1^+ = \mathbf{n}_1 \cdot (\mathbf{v}_a^+ - \mathbf{v}_b^+)$
  - $v_1^+ = \mathbf{n}_1 \cdot (\mathbf{v}_a^- - \mathbf{v}_b^-) + (m_a^{-1} + m_b^{-1})\gamma_1 - \mathbf{n}_1 \cdot m_b^{-1}\gamma_2\mathbf{n}_2$
- solving for impulse
  - $v_1^+ = -c_r v_1^- = v_1^- + (m_a^{-1} + m_b^{-1})\gamma_1 - \mathbf{n}_1 \cdot m_b^{-1}\gamma_2\mathbf{n}_2$
  - $(m_a^{-1} + m_b^{-1})\gamma_1 = -(1 + c_r)v_1^- + m_b^{-1}\gamma_2(\mathbf{n}_1 \cdot \mathbf{n}_2)$
  - $\gamma_1 = m_{\text{eff}}\left(-(1 + c_r)v_1^- + m_b^{-1}\gamma_2\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2\right)$
  - where $m_{\text{eff}} = \left(m_a^{-1} + m_b^{-1}\right)^{-1}$



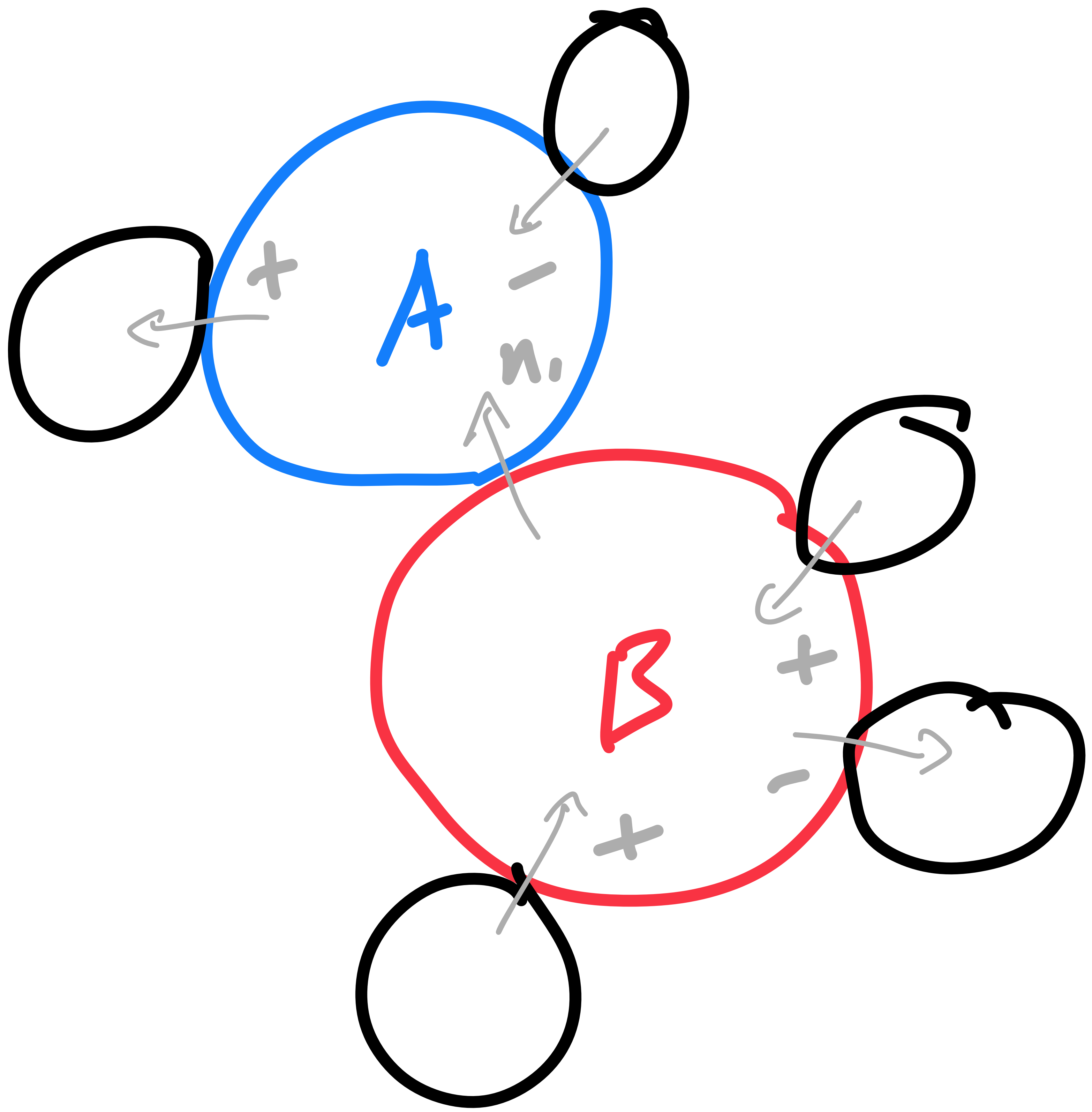
One collision in the context of many

The same idea extends to as many other collisions as required

$$\gamma_i = m_{\text{eff}} \left(-(1+c_r)v_i^- - m_a^{-1}\hat{\mathbf{n}}_i \cdot \boldsymbol{\gamma}_{ia} + m_b^{-1}\hat{\mathbf{n}}_i \cdot \boldsymbol{\gamma}_{ib} \right)$$

$$\boldsymbol{\gamma}_{ix} = \sum_{j \neq i} s_{jx}\gamma_j \hat{\mathbf{n}}_j$$

- where s_{jx} is +1 if object X is the first object in collision j and, –1 if X is the second object in collision j, and 0 if X is not involved in collision j.
- for efficiency compute $\boldsymbol{\gamma}_a$ and $\boldsymbol{\gamma}_b$ first
  - more on this later



Iterating to resolve simultaneous collisions

Since we don't know any of the γ to start, just use our best esimate

- compute object velocities, detect all collisions
- initialize all γ_i to zero
- solve for each γ_i assuming the other γs are correct
  - if γ_i wants to be negative, set it to zero (collisions can push but not pull!)
- repeat until convergence
- update velocities using impulses, compute new positions from velocities

To resolve redsidual errors, add an overlap-repair impulse

- bias target velocity in normal direction proportional to overlap
- very effective at removing residual overlap
- unstable if turned up too much to repair major overlap problems

Some implementation issues

Summing influences of related collisions

- searching all collisions for related ones is O(N^2)
- maintaining some graph data structure adds extra complexity
- there is a nice trick for maintaining these sums efficiently per object
- see lecture notes for details

This works, mostly! (demo…)

- it does converge
- it does not always converge very quickly
- errors can accumulate and lead to persistent overlap between objects

Why does this work?

If we stand back from the process we have been using, it looks like this:

1. Write the new and old normal velocities as a function of the new and old object velocities
2. Write the objects' new velocities as a function of their old velocities and the collision impulses
3. Use the restitution hypothesis to write an equation that can be solved for the collision impulses

We can formalize this computation in terms of matrices

It will lead to a matrix system with a well defined solution…

1. Normal velocities from object velocities

Normal velocity for collision 1, v_1, is a linear function of object velocities

$$v_1 = \hat{\mathbf{n}}_1 \cdot \mathbf{v}_a - \hat{\mathbf{n}}_1 \cdot \mathbf{v}_b = \begin{bmatrix} \cdots & \hat{\mathbf{n}}_1^T & \cdots & -\hat{\mathbf{n}}_1^T & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{v}_a \\ \vdots \\ \mathbf{v}_b \\ \vdots \end{bmatrix} = \mathbf{J}_1 \mathbf{v}$$

- same can be done for all collisions, stacked into a matrix $\mathbf{J}$
- then $\mathbf{v}_n = \mathbf{J}\mathbf{v}$ where $\mathbf{v}_n = [v_1 \ \cdots \ v_k]^T$
- this can be used before or after the collision:

$$\mathbf{v}_n^- = \mathbf{J}\mathbf{v}^-$$

$$\mathbf{v}_n^+ = \mathbf{J}\mathbf{v}^+$$

2. Velocity changes from collision impulses

Collision impulse 1 changes the velocities for objects A and B

$$\mathbf{v}_a^+ = \mathbf{v}_a^- + m_a^{-1}\gamma_1\hat{\mathbf{n}}_1$$

$$\mathbf{v}_b^+ = \mathbf{v}_b^- - m_b^{-1}\gamma_1\hat{\mathbf{n}}_1$$

- package the update to the whole system velocity in a vector

$$\begin{bmatrix} \vdots \\ \mathbf{v}_a^+ \\ \vdots \\ \mathbf{v}_b^+ \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{v}_a^- \\ \vdots \\ \mathbf{v}_b^- \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ m_a^{-1}\hat{\mathbf{n}}_1 \\ \vdots \\ -m_b^{-1}\hat{\mathbf{n}}_1 \\ \vdots \end{bmatrix}\gamma_1 \qquad \mathbf{M} = \begin{bmatrix} m_1 & 0 & \cdots & 0 & 0 \\ 0 & m_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & m_N & 0 \\ 0 & 0 & \cdots & 0 & m_N \end{bmatrix}$$

- $\mathbf{v}^+ = \mathbf{v}^- + \mathbf{M}^{-1}\mathbf{J}_1^T\gamma_1$ or for all collisions at once:

$$\begin{aligned} \mathbf{v}^+ &= \mathbf{v}^- + \mathbf{M}^{-1}\mathbf{J}_1^T\gamma_1 + \cdots + \mathbf{M}^{-1}\mathbf{J}_k^T\gamma_k \\ &= \mathbf{v}^- + \mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\gamma} \end{aligned}$$

3. Global system from restitution hypothesis

Restitution hypothesis as a statement about all collisions:

$$\mathbf{v}_n^+ = -c_r \mathbf{v}_n^-$$

- (1) and (2) let us write the two velocities

$$\mathbf{v}_n^- = \mathbf{J}\mathbf{v}^-$$

$$\mathbf{v}_n^+ = \mathbf{J}\mathbf{v}^+ = \mathbf{J}\mathbf{v}^- + \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\gamma}$$

- and substituting we get a linear system

$$\mathbf{J}\mathbf{v}^- + \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\gamma} = -c_r\mathbf{J}\mathbf{v}^-$$

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\gamma} = -(1 + c_r)\mathbf{J}\mathbf{v}^-$$

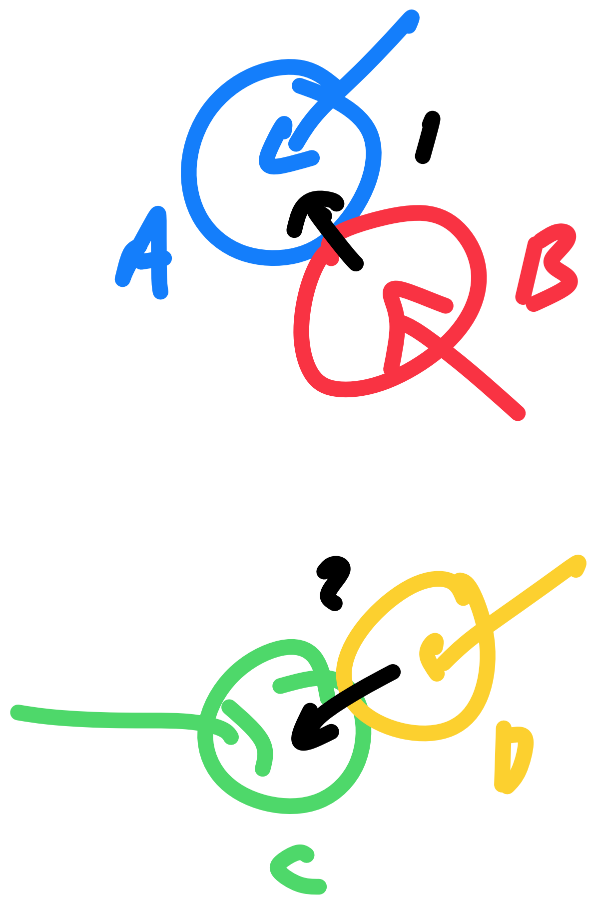
$$\mathbf{A}\boldsymbol{\gamma} = \mathbf{b}$$

- this is a square, k by k, matrix system
  - one row per collision, one column per collision

Example: independent collisions

$$J = \begin{matrix} 1 \\ 2 \end{matrix}\begin{bmatrix} n_1^T & -n_1^T & & \\ & & n_2^T & -n_2^T \end{bmatrix}$$
$$\quad\;\; \begin{matrix} a & \;\; b & \;\; c & \;\; d \end{matrix}$$

$$v = \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix}$$



$$M = \begin{bmatrix} m_a & & & \\ & m_b & & \\ & & m_c & \\ & & & m_d \end{bmatrix}$$

$$M^{-1}J^T = \begin{bmatrix} n_1/m_a & \\ -n_1/m_b & \\ & n_2/m_c \\ & -n_2/m_d \end{bmatrix}$$

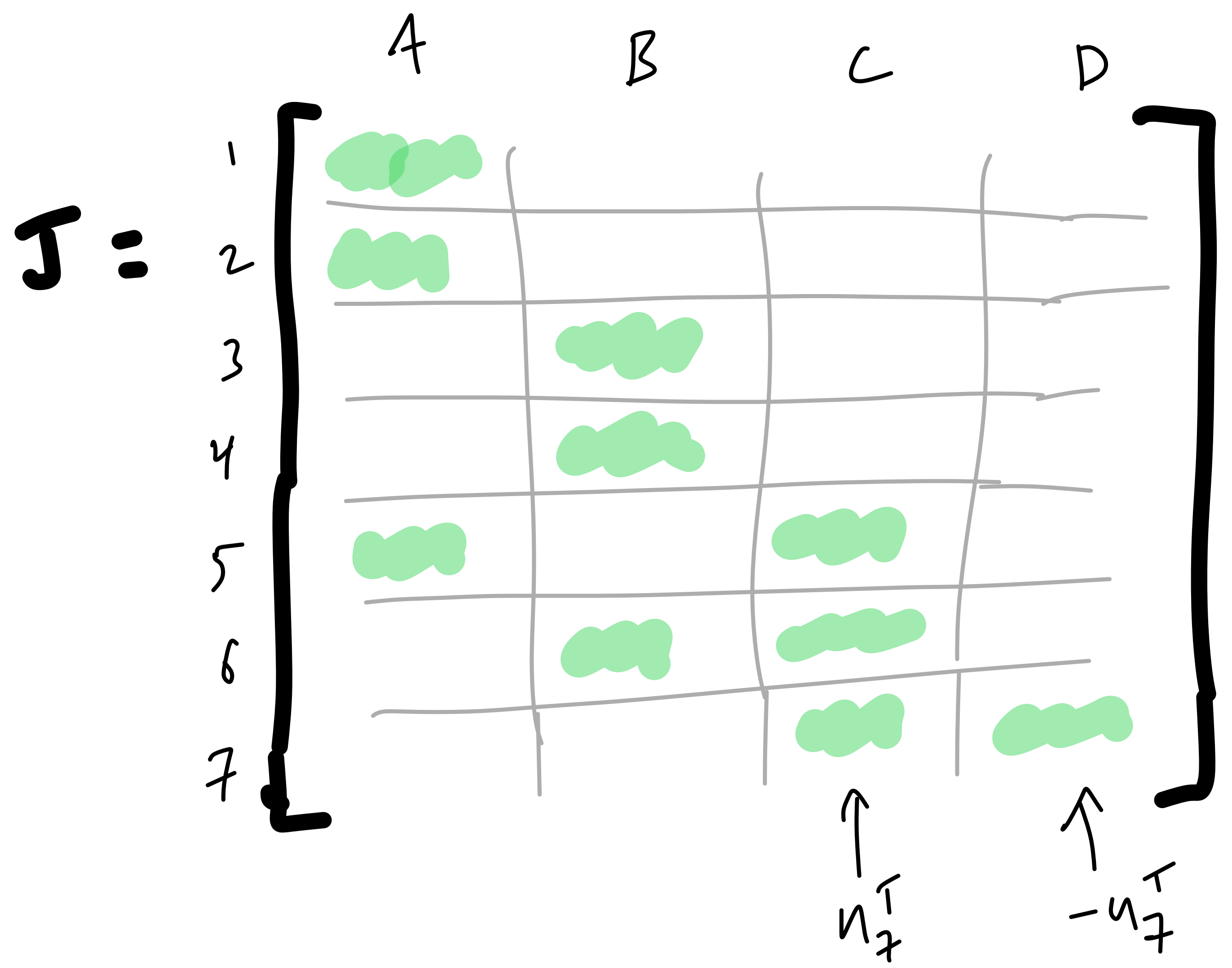
$$JM^{-1}J^T = \begin{bmatrix} \frac{1}{m_a} + \frac{1}{m_b} & \\ & \frac{1}{m_c} + \frac{1}{m_d} \end{bmatrix}$$

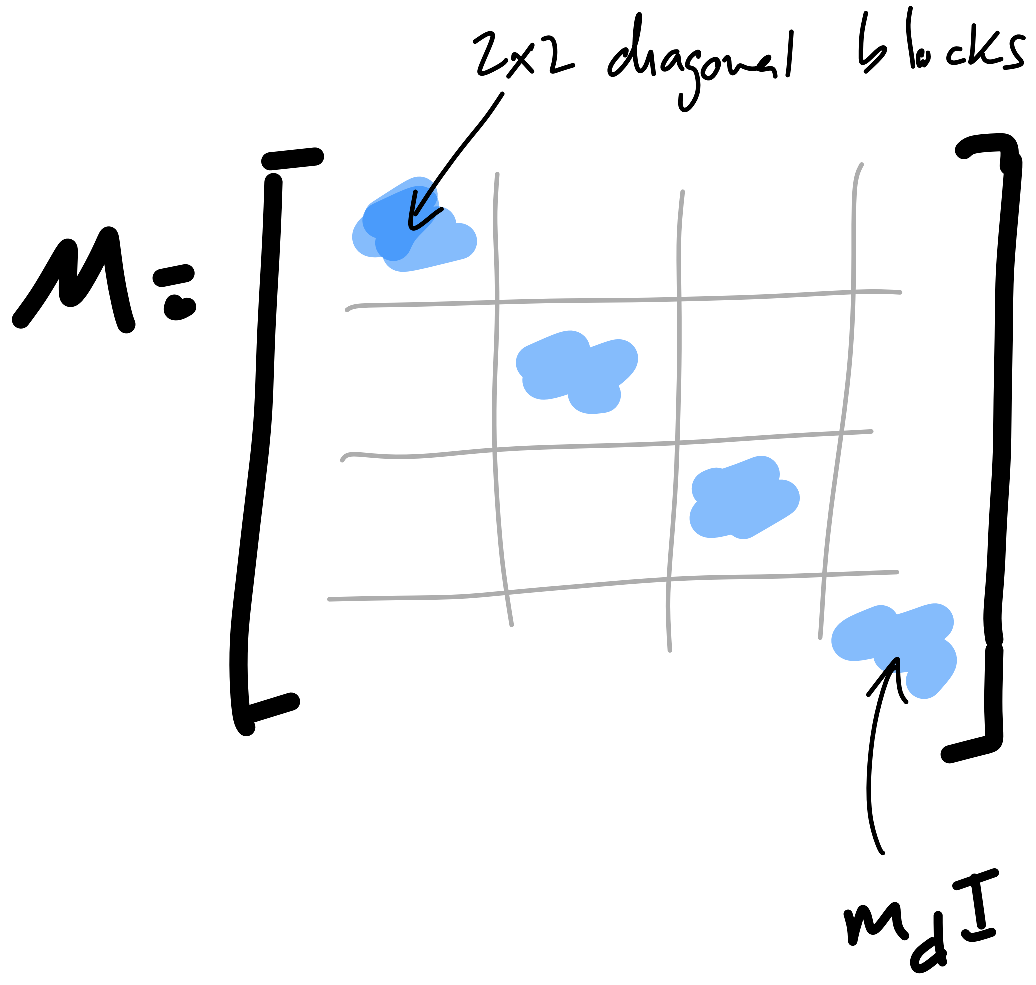
$$Jv^- = \begin{bmatrix} n_1 \cdot (v_a - v_b) \\ n_2 \cdot (v_c - v_d) \end{bmatrix}$$

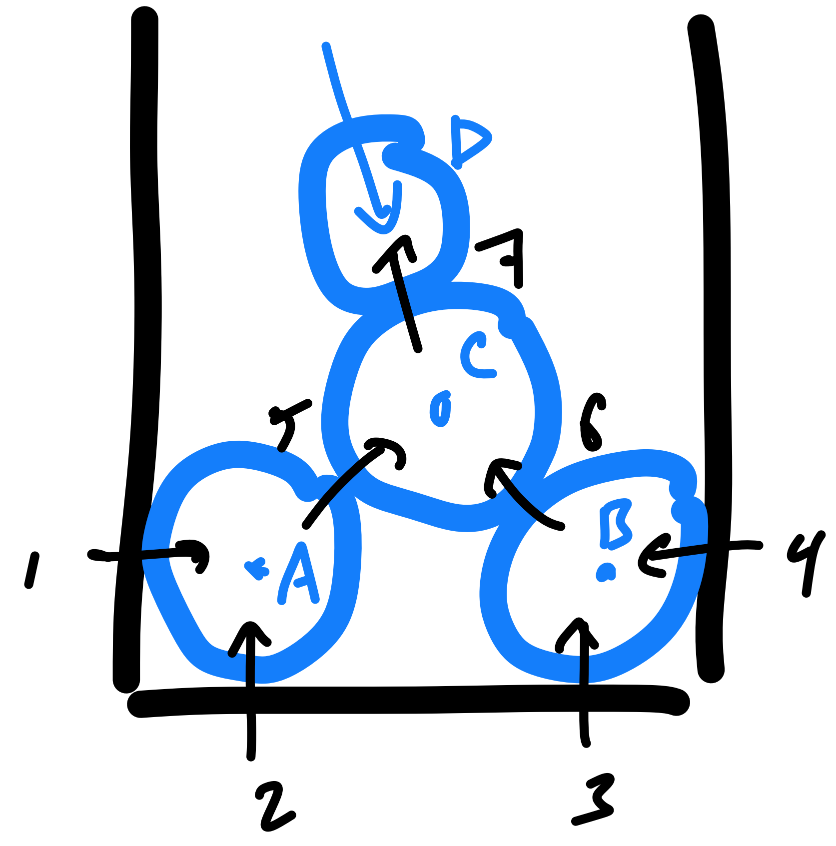
$$\gamma_1 = -(1 + c_r)\,\hat{\mathbf{n}}_1 \cdot (\mathbf{v}_a^- - \mathbf{v}_b^-) \Big/ \left(m_a^{-1} + m_b^{-1}\right)$$

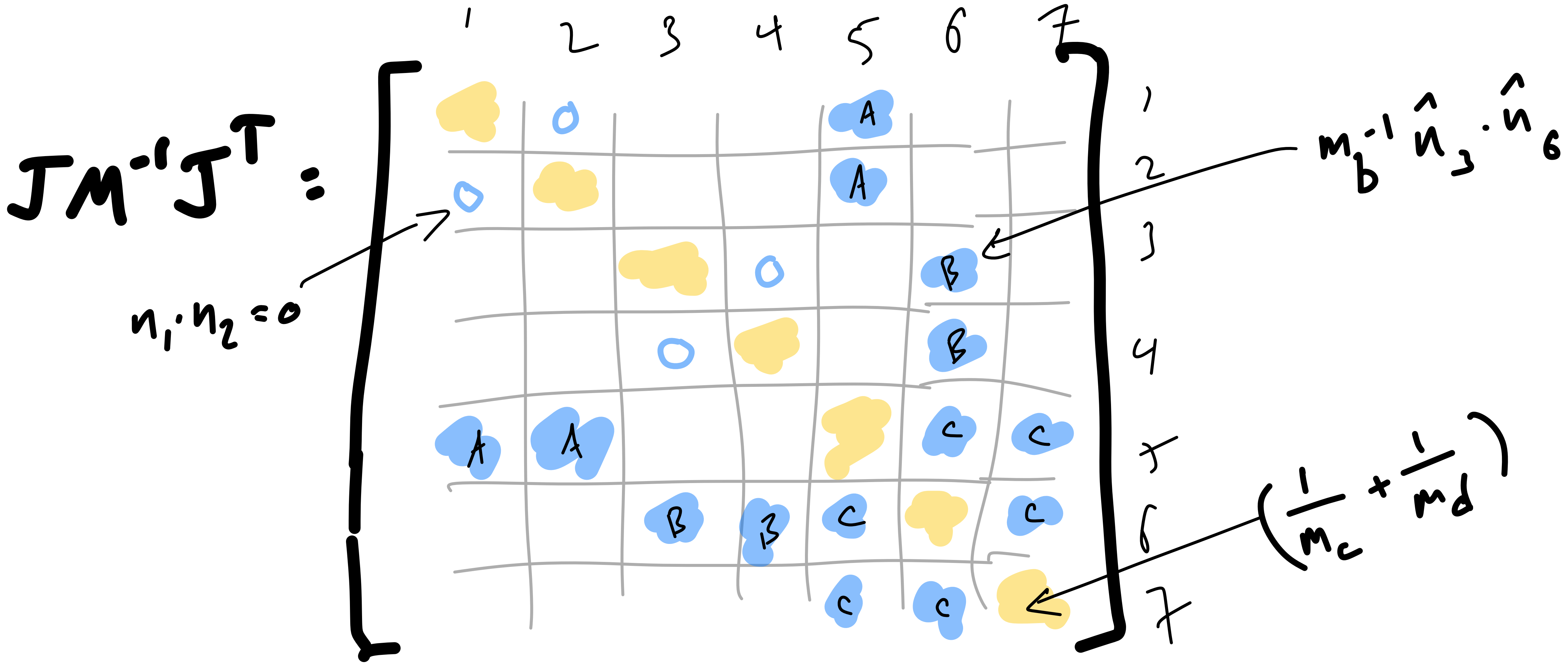
$$\gamma_2 = -(1 + c_r)\,\hat{\mathbf{n}}_2 \cdot (\mathbf{v}_c^- - \mathbf{v}_d^-) \Big/ \left(m_c^{-1} + m_d^{-1}\right)$$

Example: coupled collisions





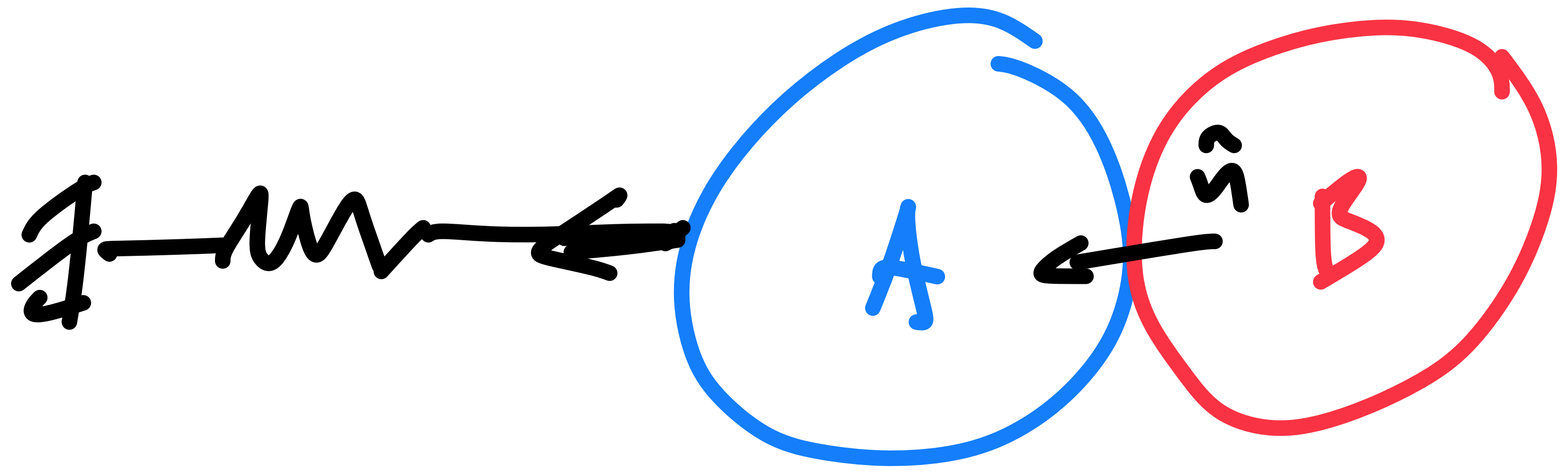




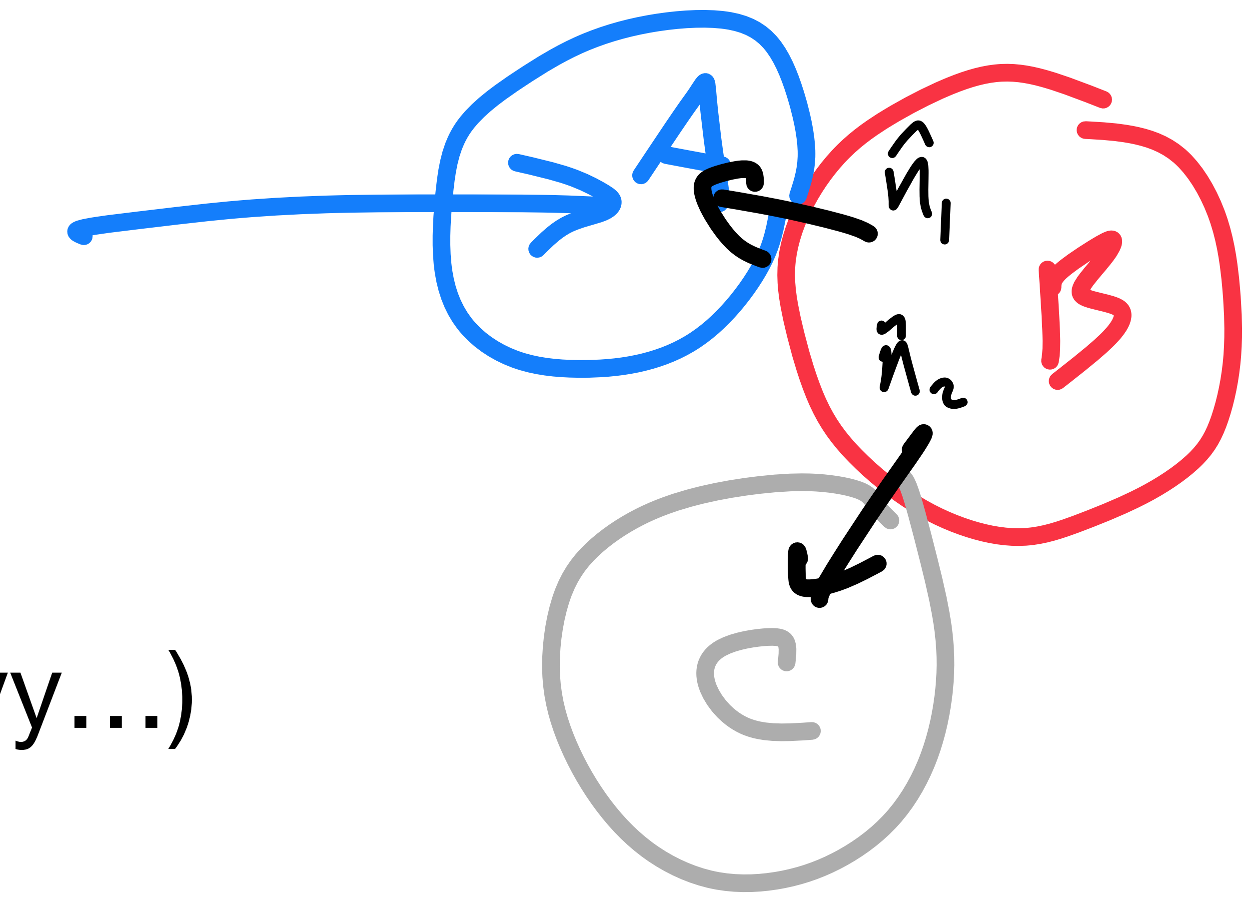
Problem: pulling impulses

In some situations we don't want to solve the equation we wrote

- e.g. single contact with force pulling objects apart



- if objects were stationary, equations ask for zero relative velocity
  - so system computes a negative γ that will bring B with A
  - solution here: just clamp γ at zero
- more complex e.g.: two contacts with impact pushing balls apart
  - clamping γ_2 to zero after solution leaves γ_1 wrong (e.g. C is heavy…)



How to say what we want?

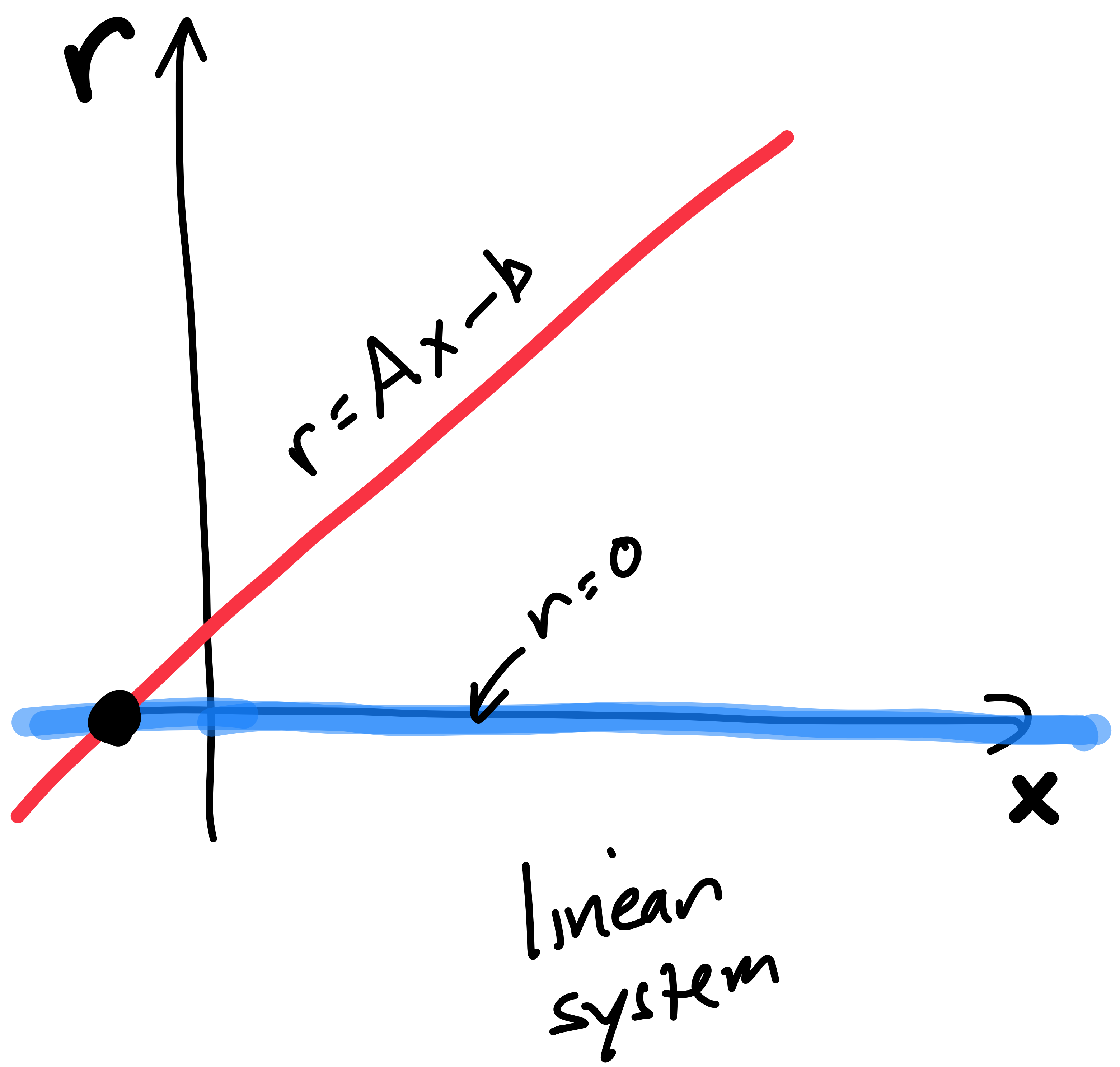
We want $\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\gamma} = -(1 + c_r)\mathbf{J}\mathbf{v}^-$, aka. $\mathbf{A}\boldsymbol{\gamma} = \mathbf{b}$

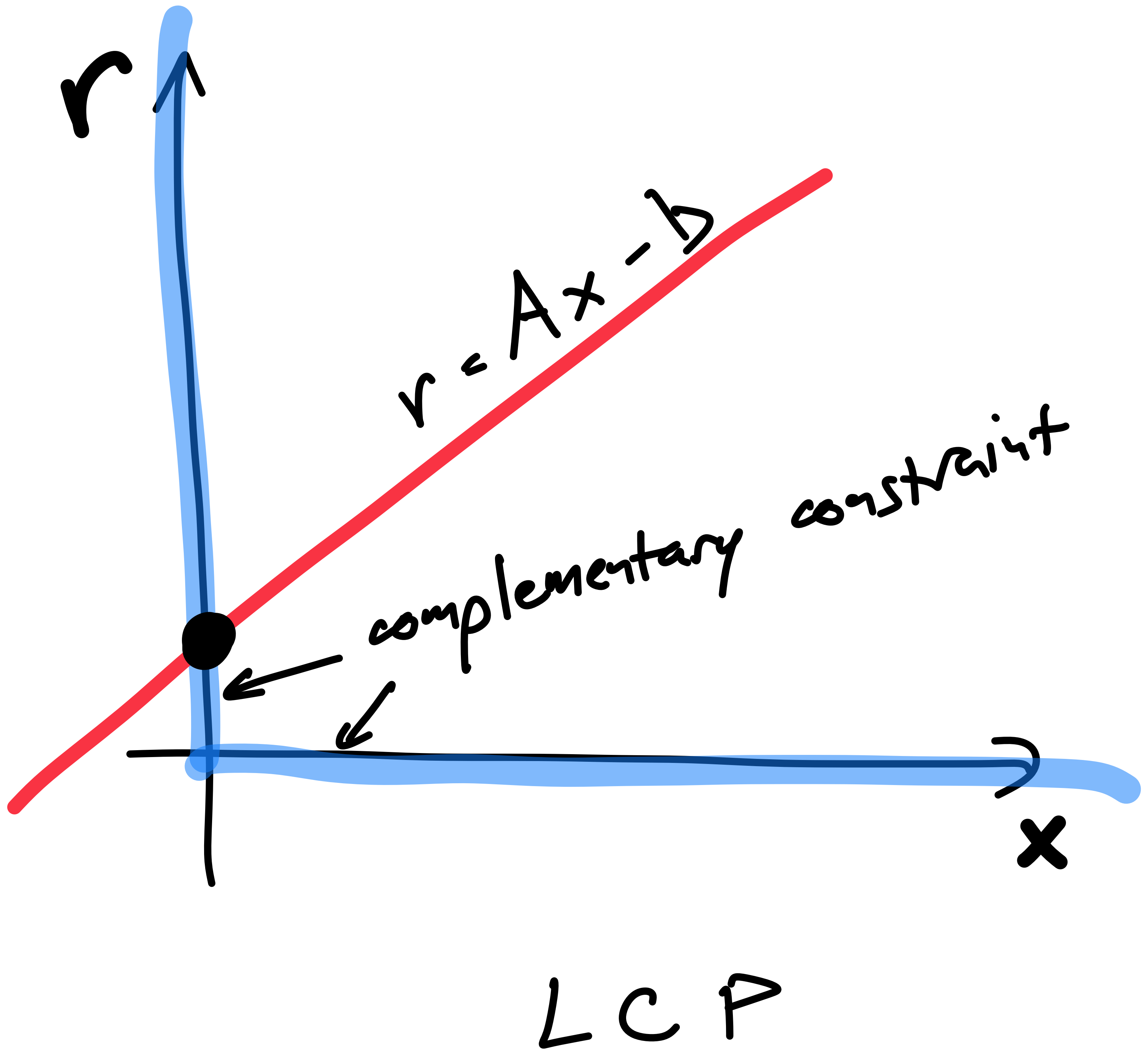
- but wait, actually, not always — the components of $\boldsymbol{\gamma}$ should not be negative
- if γ_i would be negative we want to set $\gamma_i = 0$ and let $v_i^+ > -c_r v_i^-$
- what we have here is a pair of complementary constraints for each i:
  - ($\gamma_i > 0$ and $\mathbf{A}_i\boldsymbol{\gamma} - b_i = 0$) or ($\mathbf{A}_i\boldsymbol{\gamma} - b_i > 0$ and $\gamma_i = 0$)
- stated a little too cleverly as a whole system:
  - $\mathbf{A}\boldsymbol{\gamma} - \mathbf{b} \geq 0$ and $\boldsymbol{\gamma} \geq 0$ and $(\mathbf{A}\boldsymbol{\gamma} - \mathbf{b}) \cdot \boldsymbol{\gamma} = 0$
- this kind of problem is known as a *linear complementarity problem* or LCP

A little LCP intuition

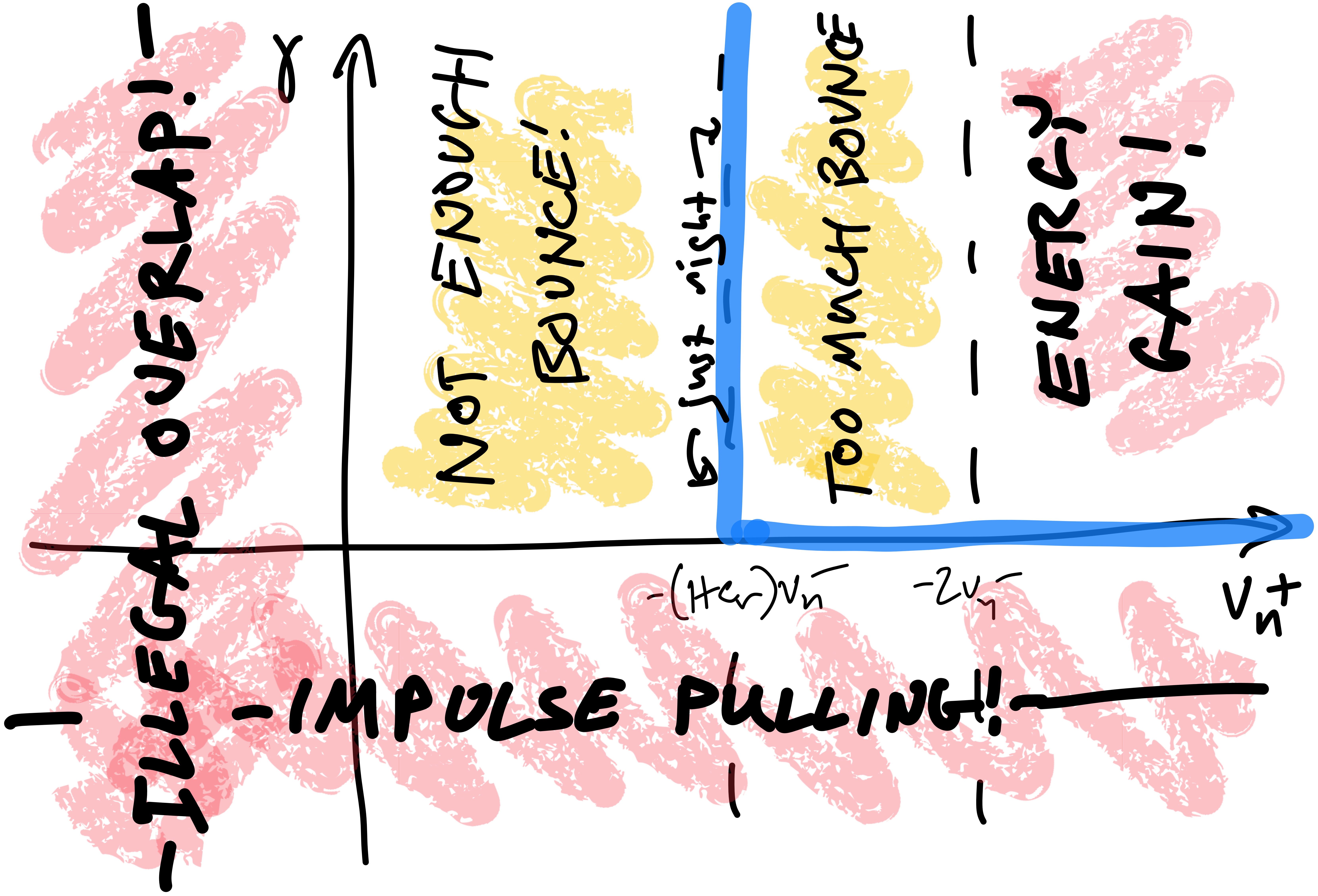
It's not really so different from a regular linear system

- linear system is intersecting $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ with $\mathbf{r} = \mathbf{0}$
- LCP is intersecting $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ with L-shaped complementary constraint
- this is not an inequality constrained optimization problem despite the appearance of "$\geq$"





LCP constraint in the context of collisions



Solving the LCP system

Popular and simple approach: Projected Gauss-Seidel

- use basic iterative solver but enforce constraint at each step by clamping $\gamma > 0$
- Gauss-Seidel algorithm is a suitable choice: solve rows sequentially
  - find x_i assuming all x_j for $i \neq j$ are known
  - use latest values for x_j
  - row i reads $\sum_{j=0}^{N} a_{ij}x_j = b_i$ or $\sum_{j=0}^{i-1} a_{ij}x_j + a_{ii}x_i + \sum_{j=i+1}^{N} a_{ij}x_j = b_i$
  - solve: $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=0}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{N} a_{ij}x_j\right)$
  - after updating all x_i, start back at the top and repeat whole process until convergnece

PGS iteration applied to contact

Fill in the problem details for the xs and bs…

- $$\gamma_i = m_{\text{eff}} \left(-(1-c_r)v_i^- - m_a^{-1} \sum_k s_{ak}\gamma_k \hat{\mathbf{n}}_k \cdot \hat{\mathbf{n}}_i + m_b^{-1} \sum_k s_{bk}\gamma_k \hat{\mathbf{n}}_k \cdot \hat{\mathbf{n}}_i \right)$$
    - …and clamp all $\gamma_i \geq 0$ at each iteration
- this looks familiar … it's the same thing we derived intuitively before!

What have we achieved

- we now can inherit a proof of convergence from PGS
- we have a more mechanical and maybe less error-prone way to derive these equations
- we now can read papers about collision and contact without glazing over when the $\mathbf{J}$s appear

Rigid bodies

We can now run the same program for rigid bodies… it's similar but with more state variables!

- recall the steps of resolving a rigid body collision:
  - write normal velocity in terms of object velocities

$$v_i = \hat{\mathbf{n}}_i \cdot \mathbf{v}_{\text{rel}} = \hat{\mathbf{n}} \cdot \left(\mathbf{v}_a - \mathbf{v}_b + \omega_a \times \mathbf{r}_a - \omega_b \times \mathbf{r}_b\right)$$

  - write new velocities in terms of collision impulse

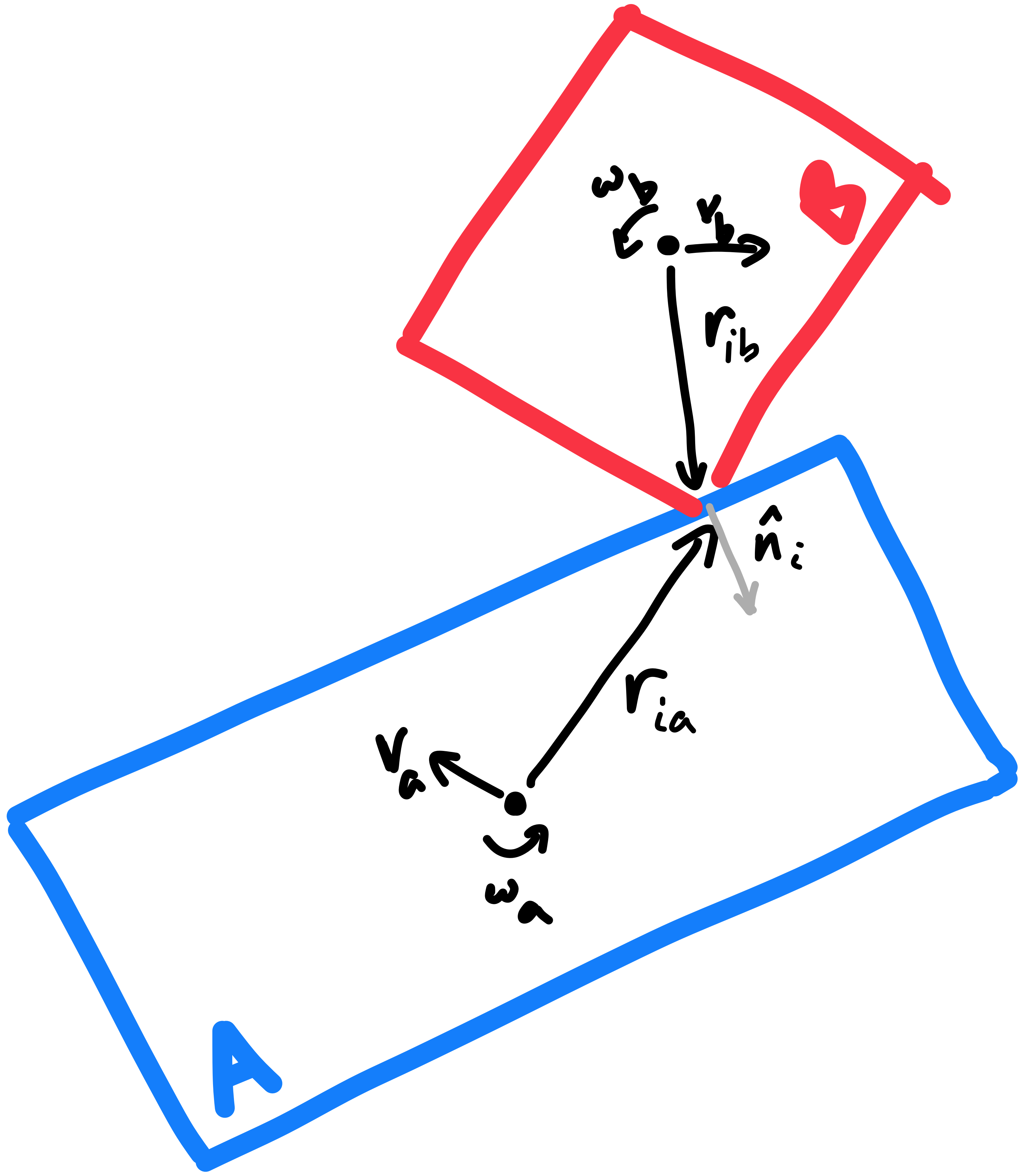
$$\Delta \mathbf{v}_a = m_a^{-1} \gamma_i \hat{\mathbf{n}}_i \qquad \Delta \omega_a = I_a^{-1} \mathbf{r}_{ia} \times \gamma_i \hat{\mathbf{n}}_i$$

$$\Delta \mathbf{v}_b = - m_b^{-1} \gamma_i \hat{\mathbf{n}}_i \qquad \Delta \omega_b = - I_b^{-1} \mathbf{r}_{ib} \times \gamma_i \hat{\mathbf{n}}_i$$

  - substitute into restitution hypothesis and solve

$$\gamma_i = -(1 + c_r) m_{\text{eff},i} v_i^-$$

$$m_{\text{eff},i} = \left(m_a^{-1} + m_b^{-1} + I_a^{-1} \hat{\mathbf{n}} \cdot (\mathbf{r}_{ia} \times \hat{\mathbf{n}}_i) \times \mathbf{r}_{ia} + I_b^{-1} \hat{\mathbf{n}} \cdot (\mathbf{r}_{ib} \times \hat{\mathbf{n}}_i) \times \mathbf{r}_{ib}\right)^{-1}$$



- if there are other contacts, their impulses contribute to the velocities
  - $$\Delta \mathbf{v}_a = m_a^{-1}\gamma_i \hat{\mathbf{n}}_i + m_a^{-1}\sum_{j\neq i} s_{ja}\gamma_j \hat{\mathbf{n}}_j \qquad \Delta \omega_a = I_a^{-1}\mathbf{r}_{ia}\times \gamma_i \hat{\mathbf{n}}_i + I_a^{-1}\sum_{j\neq i} s_{ja}\mathbf{r}_{ja}\times \gamma_j \hat{\mathbf{n}}_j$$
  - $$\Delta \mathbf{v}_b = \underbrace{-m_b^{-1}\gamma_i \hat{\mathbf{n}}_i}_{\Delta \mathbf{v}_b^{\text{self}}} + \underbrace{m_b^{-1}\sum_{j\neq i} s_{jb}\gamma_j \hat{\mathbf{n}}_j}_{\Delta \mathbf{v}_b^{\text{other}}} \qquad \Delta \omega_b = \underbrace{-I_b^{-1}\mathbf{r}_{ib}\times \gamma_i \hat{\mathbf{n}}_i}_{\Delta \omega_b^{\text{self}}} + \underbrace{I_b^{-1}\sum_{j\neq i} s_{jb}\mathbf{r}_{jb}\times \gamma_j \hat{\mathbf{n}}_j}_{\Delta \omega_b^{\text{other}}}$$
- when we compute the post-collision relative velocity this produces extra terms

$$\mathbf{v}_{\text{rel}}^{+} = \mathbf{v}_{\text{rel}}^{-} + \left(\Delta \mathbf{v}_a + \Delta \omega_a \times \mathbf{r}_{ia}\right) - \left(\Delta \mathbf{v}_b + \Delta \omega_b \times \mathbf{r}_{ib}\right)$$

  - $$= \mathbf{v}_{\text{rel}}^{-} + \left(m_a^{-1}\hat{\mathbf{n}}_i + m_b^{-1}\hat{\mathbf{n}}_i + I_a^{-1}(\mathbf{r}_{ia}\times \hat{\mathbf{n}}_i)\times \mathbf{r}_{ia} + I_b^{-1}(\mathbf{r}_{ib}\times \hat{\mathbf{n}}_i)\times \mathbf{r}_{ib}\right)\gamma_i + \Delta \mathbf{v}_a^{\text{other}} - \Delta \mathbf{v}_b^{\text{other}} + \Delta \omega_a^{\text{other}}\times \mathbf{r}_{ia} - \Delta \omega_b^{\text{other}}\times \mathbf{r}_{ib}$$
- and they also propagate into the normal velocity

$$v_i^{+} = \hat{\mathbf{n}}_i \cdot \mathbf{v}_{\text{rel}}^{+}$$

  - $$= v_i^{-} + m_{\text{eff},i}^{-1}\gamma_i + \hat{\mathbf{n}}\cdot\left(\Delta \mathbf{v}_a^{\text{other}} - \Delta \mathbf{v}_b^{\text{other}} + \Delta \omega_a^{\text{other}}\times \mathbf{r}_{ia} - \Delta \omega_b^{\text{other}}\times \mathbf{r}_{ib}\right)$$

- finally solving for γ_i we get
  - $\gamma_i = -\, m_{\text{eff},i} \left[(1 + c_r) v_i^- + \hat{\mathbf{n}} \cdot \left(\Delta \mathbf{v}_a^{\text{other}} - \Delta \mathbf{v}_b^{\text{other}} + \Delta \omega_a^{\text{other}} \times \mathbf{r}_{ia} - \Delta \omega_b^{\text{other}} \times \mathbf{r}_{ib} \right) \right]$
- which we can compare to the result for an isolated collision from 2 slides back
  - $\gamma_i = -\,(1 + c_r) m_{\text{eff},i} v_i^-$ —if there are no other collisions involving A or B

This leads to an iterative algorithm in exactly the same way as with particles

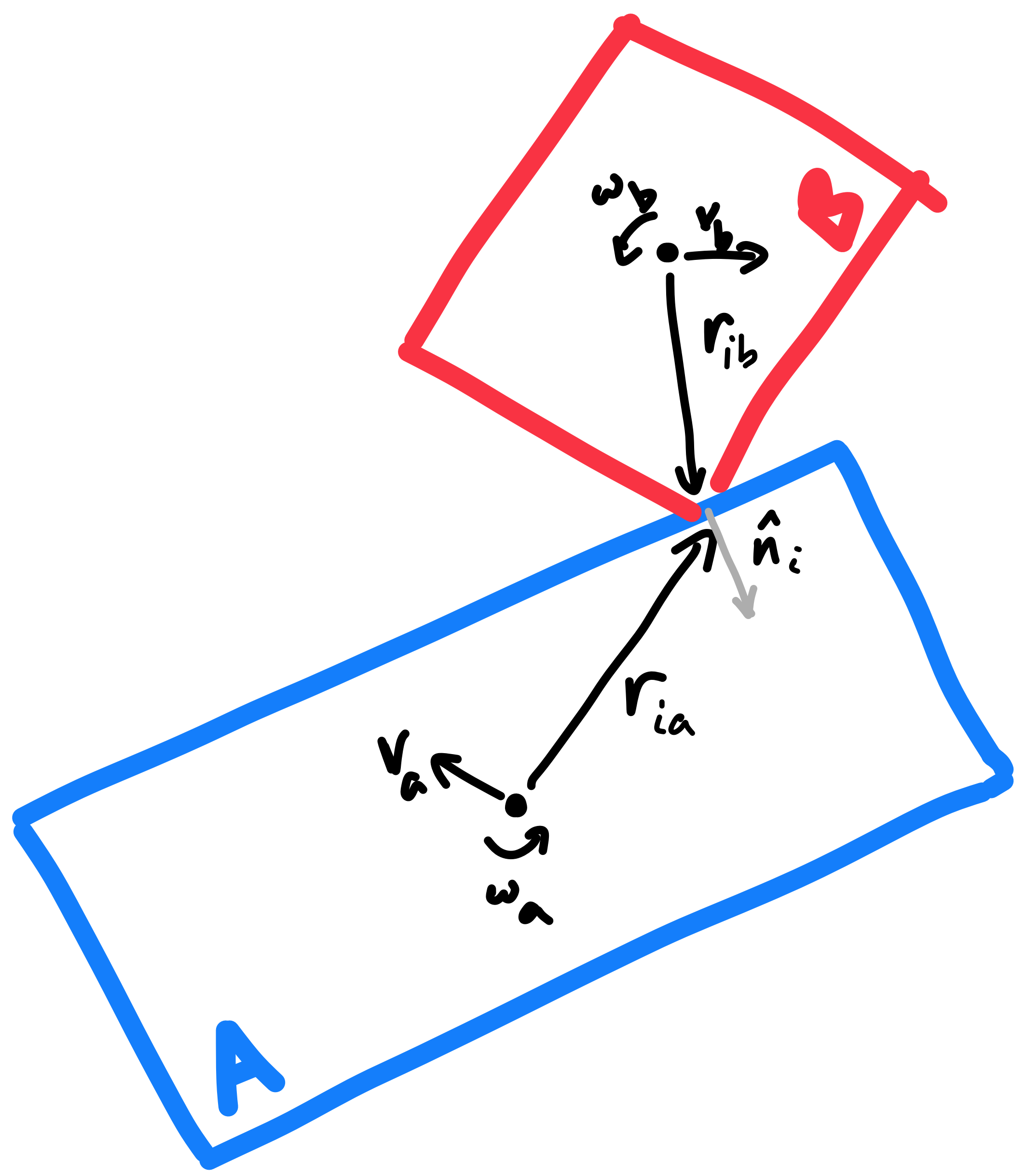
- compute each collision impulse magnitude assuming the other impulses are correct
- iterate in Gauss-Seidel fashion
  - this means the new value of each γ is used in computing all subsequent γs
- project to account for non-pulling constraint
  - this means every computed γ gets clamped at zero

Matrix form for rigid bodies

$$J_i = \left[\cdots \; \hat{n}_i^T \quad (r_i \times \hat{n})^T \; \cdots \; -\hat{n}_i \quad -(r_i \times \hat{n}_i)^T \cdots \right]$$

$$\qquad\quad v_a \qquad\quad \omega_a \qquad\qquad v_b \qquad\quad \omega_b$$

$$u = \begin{bmatrix} v_1 \\ \omega_1 \\ \vdots \\ \vdots \\ v_n \\ \omega_n \end{bmatrix}$$



$$v_i = J_i u$$

$$v_n = J u$$

$$M = \begin{bmatrix} m_1 & & & & & & \\ & m_1 & & & & & \\ & & I_1 & & & & \\ & & & \ddots & & & \\ & & & & m_n & & \\ & & & & & m_n & \\ & & & & & & I_n \end{bmatrix}$$

It all goes through exactly the same way with velocity and angular velocity gathered into $\mathbf{u}$, more columns of $\mathbf{J}$, and longer diagonal for $\mathbf{M}$.

$$u^+ = u^- + \cdots + M^{-1} J_i^T \lambda_i + \cdots \quad = u^- + M^{-1} J^T \lambda$$

$$v_n^+ = -c_r v_n^-$$

$$J u^+ = -c_r J u^- = J u^- + J M^{-1} J^T \lambda \quad \leadsto \quad \boxed{J M^{-1} J^T \lambda = -(1 + c_r) J u^-}$$

Friction

So far all impacts and resting contacts have been frictionless

- works OK for dynamic motion
- some pretty serious limitations for slow/resting contact
    - stacks can be taken apart by miniscule sideways forces
    - objects will not stay put on the slightest incline
    - in practice objects will not stay put at all :)

Solution is to include a model for friction

- a force which opposes sliding (tangential) motion
- one model: viscous drag
    - opposing force proportional to tangential velocity
- better model: "dry friction"
    - can exert a force even with no velocity

Coulomb friction model

A time-honored pretty-good model for complex contact forces

Two rules:

- frictional force opposes tangential velocity
  - when the contact is sliding, frictional force opposes the motion
  - when the contact is stuck, frictional force resists starting to move
  - friction never increases velocity
- magnitude of frictional force is limited to μ times the normal force
  - if it can keep velocity at zero it will
  - if not it will push at the maximum force

Modeling friction mathematically

I'll show a velocity/impulse formulation, in 2D for simplicity

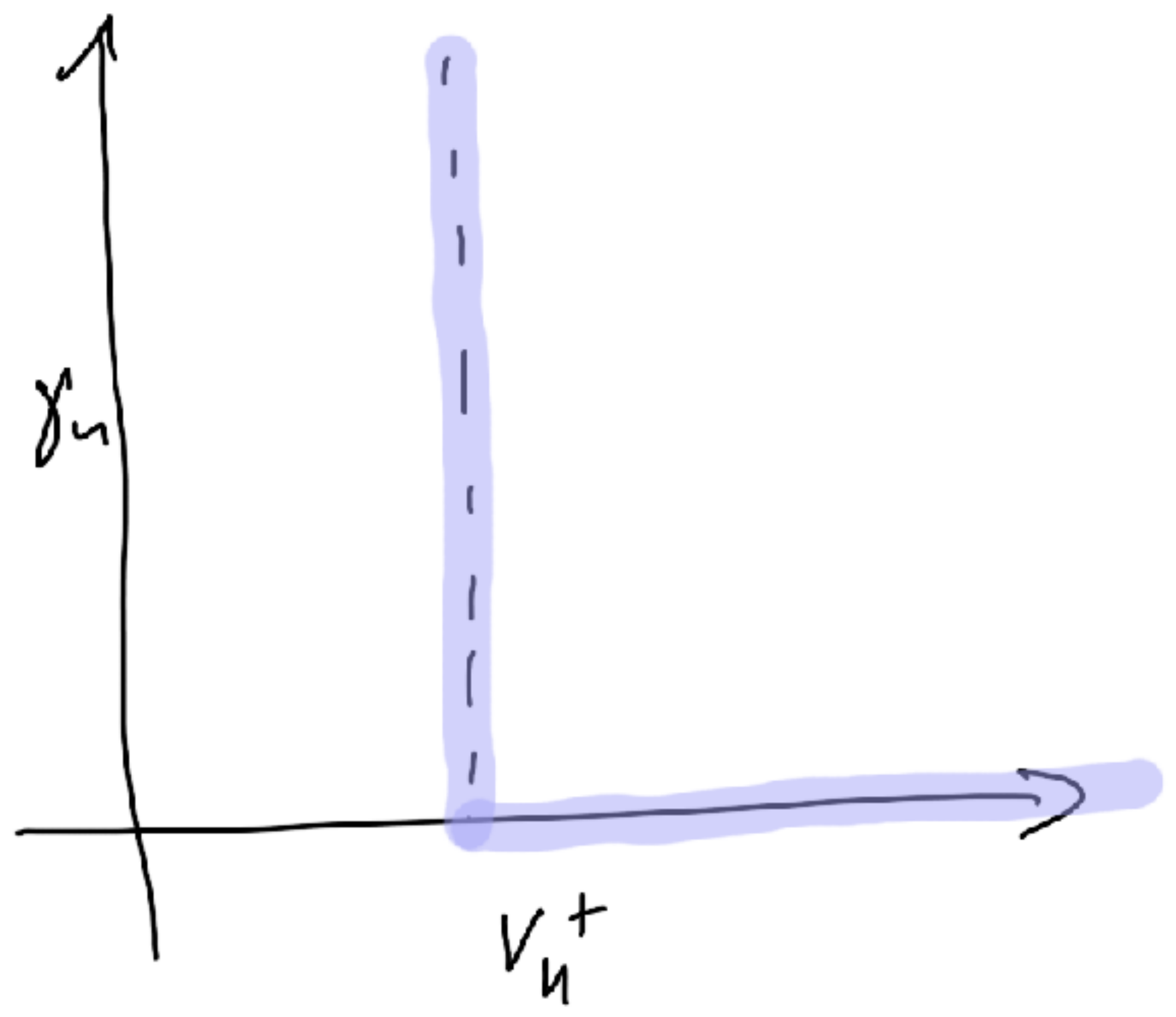
Separate relative velocity and contact impulse into normal and tangential

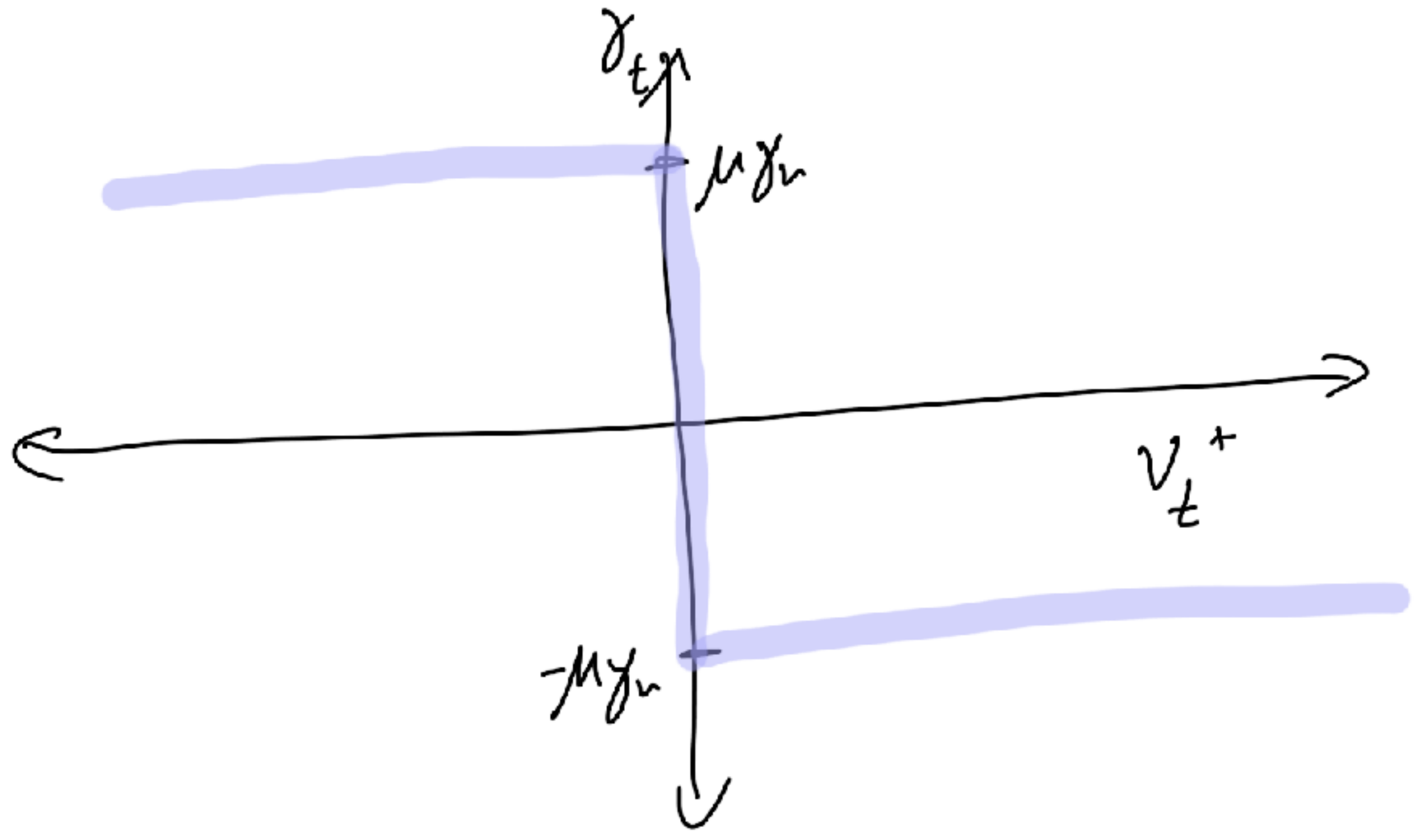
- $\mathbf{v}_{\text{rel}} = v_n\hat{\mathbf{n}} + v_t\hat{\mathbf{t}}$
- $\mathbf{j} = \gamma_n v_n\hat{\mathbf{n}} + \gamma_t v_t\hat{\mathbf{t}}$

Solve for impulses in terms of relations between velocity and impulse

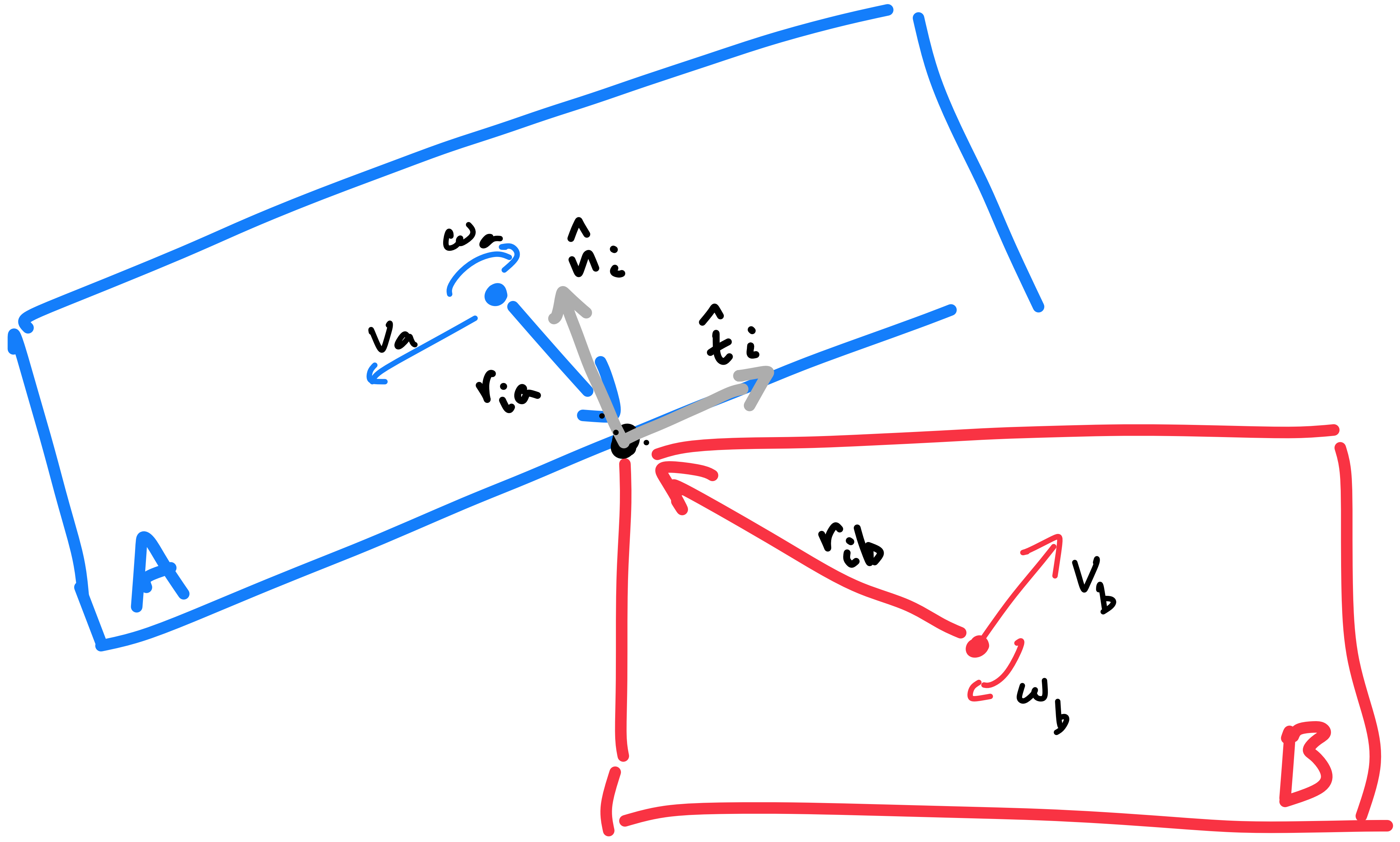
- for normal direction, $v_n^+ \geq -c_r v_n^-$ and $\gamma_n = 0$ or $v_n^+ = -c_r v_n^-$ and $\gamma_n \geq 0$
- for tangent direction, three cases:
  - sliding to the right: $v_t \geq 0$ and $\gamma_t = \mu\gamma_n$, or
  - sliding to the left: $v_t \leq 0$ and $\gamma_t = -\mu\gamma_n$, or
  - stuck: $v_t = 0$ and $|\gamma_t| \leq |\mu\gamma_n|$

Frictional contact relations in pictures





- Start with relative velocity but keep normal and tangential components
  - $v_i^n = \hat{\mathbf{n}}_i \cdot \mathbf{v}_{\text{rel}} = \hat{\mathbf{n}}_i \cdot \left(\mathbf{v}_a - \mathbf{v}_b + \omega_a \times \mathbf{r}_{ia} - \omega_b \times \mathbf{r}_{ib}\right)$
  - $v_i^t = \hat{\mathbf{t}}_i \cdot \mathbf{v}_{\text{rel}} = \hat{\mathbf{t}}_i \cdot \left(\mathbf{v}_a - \mathbf{v}_b + \omega_a \times \mathbf{r}_{ia} - \omega_b \times \mathbf{r}_{ib}\right)$
- Introduce unknown impulses in both directions
  - $\Delta \mathbf{v}_x = m_x^{-1} \sum_i s_{ix} \left(\gamma_i^n \hat{\mathbf{n}}_i + \gamma_i^t \hat{\mathbf{t}}_i\right)$
  - $\Delta \omega_x = I_x^{-1} \sum_i s_{ix} \left(\gamma_i^n \mathbf{r}_{ix} \times \hat{\mathbf{n}}_i + \gamma_i^t \mathbf{r}_{ix} \times \hat{\mathbf{t}}_i\right)$
- Solve for impulses
  - $\Delta \gamma_i^n = -m_{\text{eff},i}^n \left[(1 + c_r) v_i^{n-} + \hat{\mathbf{n}} \cdot \left(\Delta \mathbf{v}_a - \Delta \mathbf{v}_b + \Delta \omega_a \times \mathbf{r}_{ia} - \Delta \omega_b \times \mathbf{r}_{ib}\right)\right]$
  - $m_{\text{eff},i}^n = \left(m_a^{-1} + m_b^{-1} + I_a^{-1} \hat{\mathbf{n}} \cdot (\mathbf{r}_{ia} \times \hat{\mathbf{n}}_i) \times \mathbf{r}_{ia} + I_b^{-1} \hat{\mathbf{n}} \cdot (\mathbf{r}_{ib} \times \hat{\mathbf{n}}_i) \times \mathbf{r}_{ib}\right)^{-1}$
  - $\Delta \gamma_i^t = -m_{\text{eff},i}^t \left[v_i^{t-} + \hat{\mathbf{t}} \cdot \left(\Delta \mathbf{v}_a - \Delta \mathbf{v}_b + \Delta \omega_a \times \mathbf{r}_{ia} - \Delta \omega_b \times \mathbf{r}_{ib}\right)\right]$
  - $m_{\text{eff},i}^t = \left(m_a^{-1} + m_b^{-1} + I_a^{-1} \hat{\mathbf{t}} \cdot (\mathbf{r}_{ia} \times \hat{\mathbf{t}}_i) \times \mathbf{r}_{ia} + I_b^{-1} \hat{\mathbf{t}} \cdot (\mathbf{r}_{ib} \times \hat{\mathbf{t}}_i) \times \mathbf{r}_{ib}\right)^{-1}$



$$v_i^n = \underbrace{\begin{bmatrix} \cdots & \hat{n}_i^T & (r_{ia}\times\hat{n}_i)^T & \cdots & -\hat{n}_i & -(r_{ia}\times\hat{n}_i)^T & \cdots \end{bmatrix}}_{J_i^n} \underbrace{\begin{bmatrix} v_1 \\ \omega_1 \\ \vdots \\ v_n \\ \omega_n \end{bmatrix}}_{u}$$

$$v_i^t = \underbrace{\begin{bmatrix} \hat{t}_i^T & (r_{ia}\times\hat{t}_i)^T & \cdots & -t_i^T & -(r_{ib}\times\hat{t}_i)^T & \cdots \end{bmatrix}}_{J_i^t} u$$

$$\begin{bmatrix} v_1^n \\ v_1^t \\ \vdots \\ v_k^n \\ v_k^t \end{bmatrix} = \begin{bmatrix} - J_1^n - \\ - J_1^t - \\ \\ - J_k^n - \\ - J_k^t - \end{bmatrix} \begin{bmatrix} | \\ u \\ | \end{bmatrix}$$

$$v_c = J u$$

$$\begin{bmatrix} | \\ \Delta u \\ | \end{bmatrix} = \begin{bmatrix} \ddots & & & & & \\ & m_a & & & & \\ & & m_b & & & \\ & & & I_a & & \\ & & & & \ddots & \\ & & & & & m_b \; m_b \; I_b \; \ddots \end{bmatrix}^{-1} \begin{bmatrix} \hat{n}_i & \hat{t}_i \\ r_{ia}\times\hat{n}_i & r_{ia}\times\hat{t}_i \\ & \\ -\hat{n}_i & -\hat{t}_i \\ -r_{ib}\times n_i & -r_{ib}\times\hat{t}_i \end{bmatrix} \begin{bmatrix} \gamma_i^n \\ \gamma_i^t \end{bmatrix}$$

$$\Delta u = M^{-1} J^T \gamma$$

Δu: object velocities of whole system

γ: normal and tangential impulse magnitudes per collision

$$\Delta v_c = J M^{-1} J^T \gamma$$

Δv_c: normal and tangential velocities per collision

γ: normal and tangential impulse magnitudes per collision

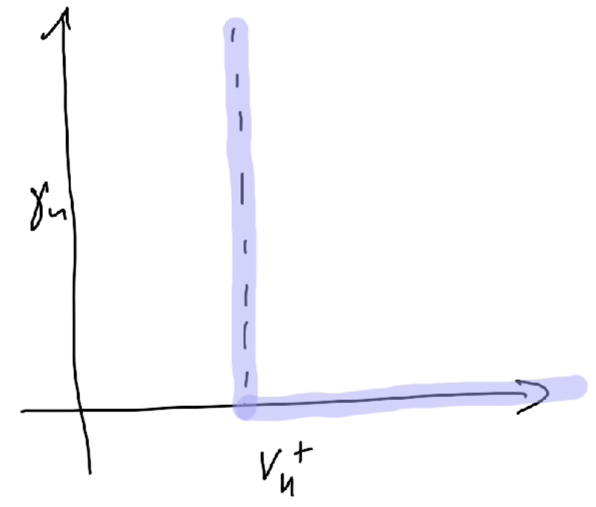
Solving contact with friction

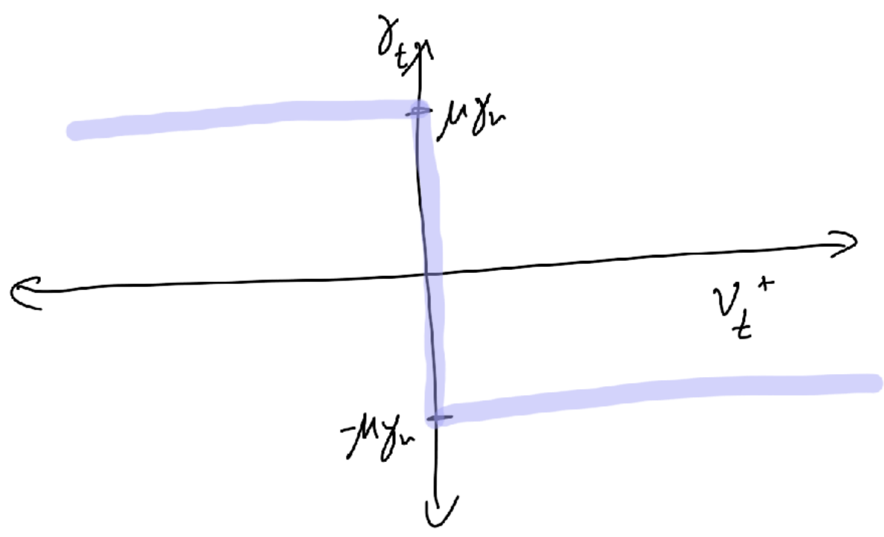
System has the same form as without friction, with two differences

- there are two kinds of γs, one with only lower bounds and one with upper and lower bounds
- the bounds for each γ^t are dependent on the value of the corresponding γ^n

$$\underbrace{\begin{bmatrix} v_1^n \\ v_1^t \\ \vdots \\ v_k^n \\ v_k^t \end{bmatrix} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \begin{bmatrix} \gamma_1^n \\ \gamma_1^t \\ \vdots \\ \gamma_k^n \\ \gamma_k^t \end{bmatrix}}_{\text{linear equations}}$$

$$\underbrace{\begin{array}{l} \left.\begin{array}{ll} \Delta v_i^n = -(1+c_r)v_i^{n-} \quad \text{and} & \gamma_i^n \geq 0 \\ \Delta v_i^n \geq -(1+c_r)v_i^{n-} \quad \text{and} & \gamma_i^n = 0 \end{array}\right\} \text{or} \\ \left.\begin{array}{ll} \Delta v_i^t = -v_i^{t-} \quad \text{and} & -\mu\gamma_i^n \leq \gamma_i^t \leq \mu\gamma_i^n \\ \Delta v_i^t \leq -v_i^{t-} \quad \text{and} & \gamma_i^t = \mu\gamma_i^n \\ \Delta v_i^t \geq -v_i^{t-} \quad \text{and} & -\mu\gamma_i^n = \gamma_i^t \end{array}\right\} \text{or} \end{array}}_{\text{(almost) linear constraints}}$$





PGS for friction

Same algorithm with a couple of tweaks

- for each iteration
  - for each impulse γ_i^x to be determined (considering normal and tangential separately)

    compute an update to γ_i^x

    update the bounds $\gamma_{\min} = 0$ or $-\mu\gamma_i^n$ and $\gamma_{\max} = \infty$ or $\mu\gamma_i^n$

    clamp to the range $\gamma_{\min} \leq \gamma_i^x \leq \gamma_{\max}$