Collision

- ➢ Detection
- ➢ Resolution

Detection:
top-down (Bounding Volume Hierarchy)

See also
https://en.wikipedia.org/wiki/Collision_detection
https://en.wikipedia.org/wiki/Collision_response

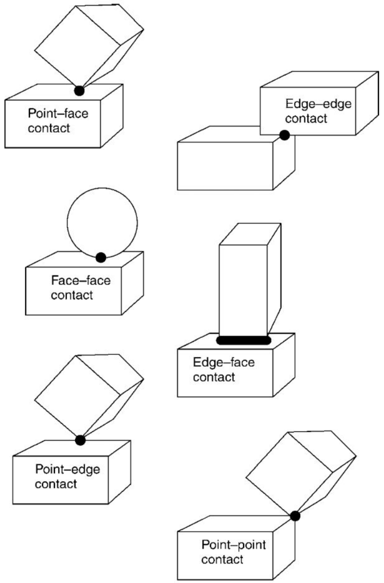
Collision Detection

Detection:
top-down (Bounding Volume Hierarchy)

Broad phase
• Disregard pairs of objects that cannot collide.
Ø Model and space partitioning.

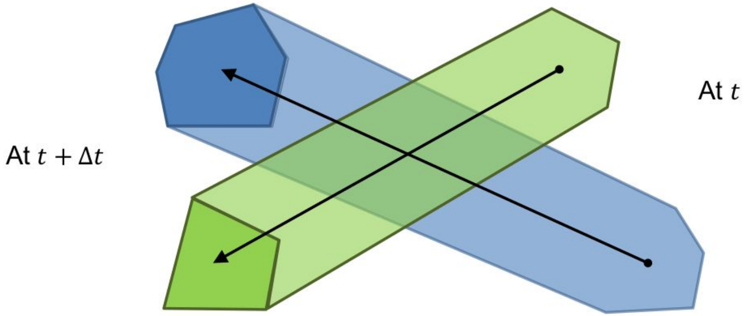
Mid phase
• Determine potentially-colliding primitives.
Ø Movement bounds.

Narrow phase
• Determine exact contact between two shapes.
• Convex object intersection (GJK algorithm)
Ø Triangle-triangle intersections.



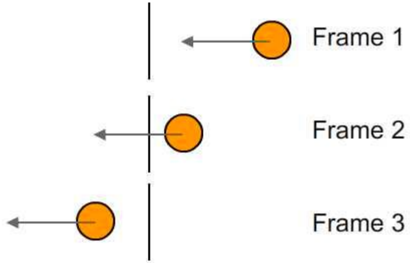
Collision Detection

Sweep motion: intersecting paths



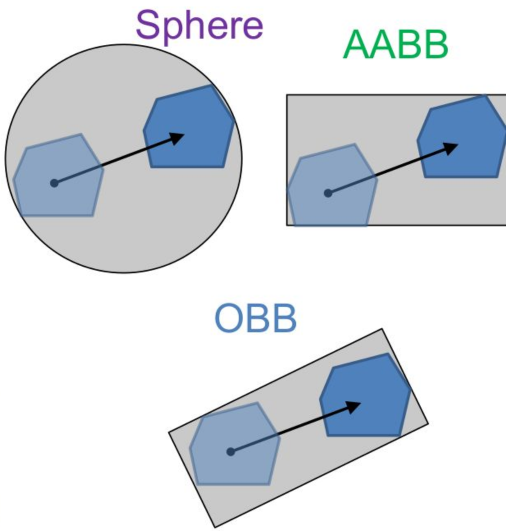
Tunneling: false negatives

Small, fast-moving → time step

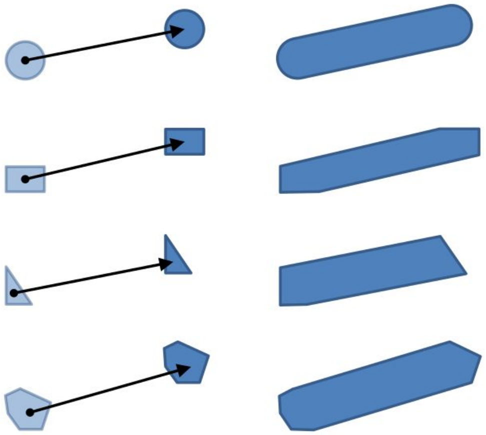


Collision Detection

Movement bounds: sphere, AABB, OBB



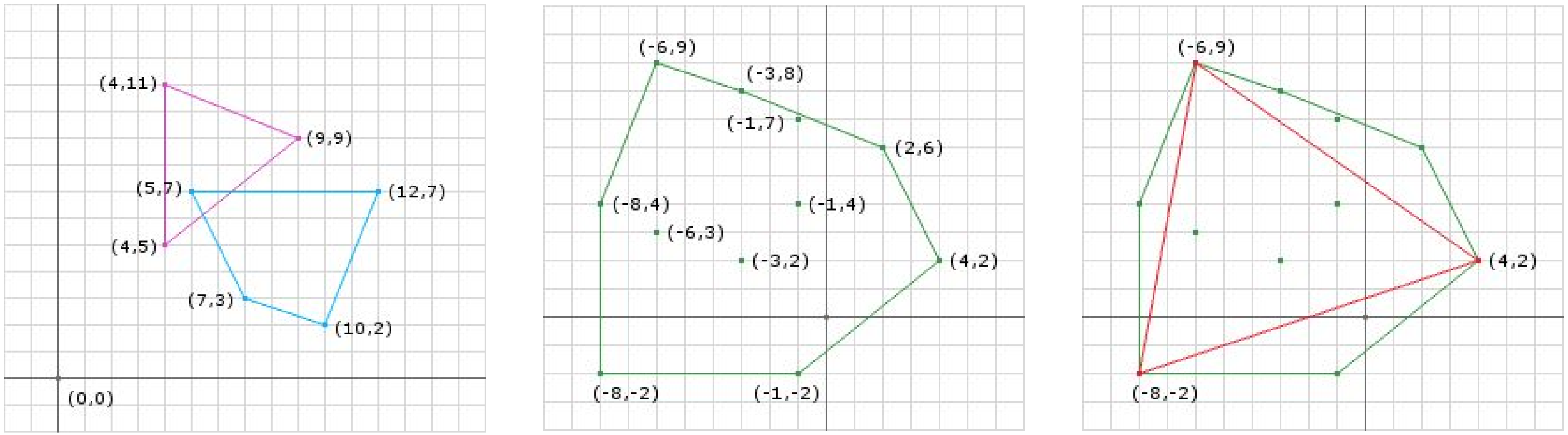
Sweep bounding box:
Minkowski sum



Collision Detection

GJK Algorithm: origin in Minkowski Difference (MD)
→ iteratively approximate MD by simplices

http://www.dyn4j.org/2010/04/gjk-gilbert-johnson-keerthi/

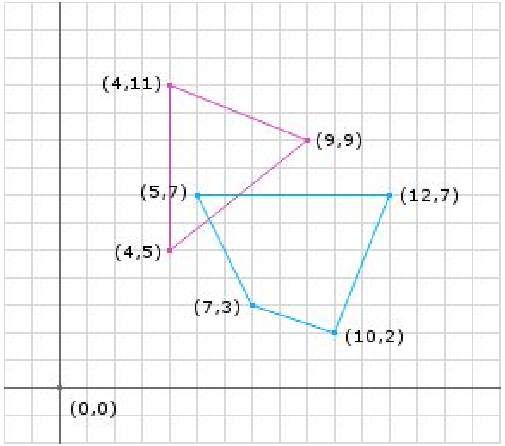


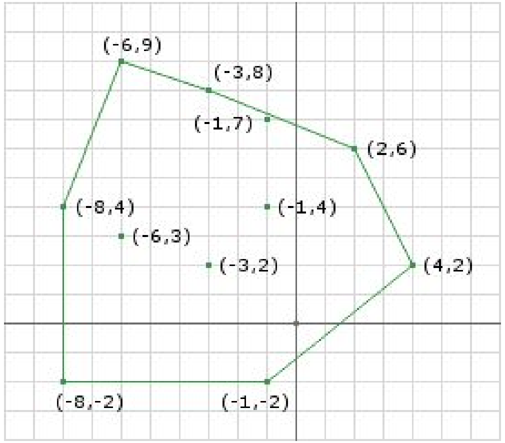
Dir = 1,0 -1,0 0,1

(9,9)-(5,7) (4,5)-(12,7) (4,11)-(10,2)

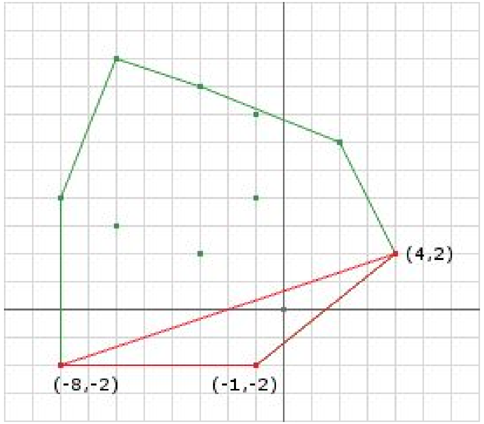
Collision Detection

GJK Algorithm: origin in Minkowski Difference (MD)
→ iteratively approximate MD by simplices





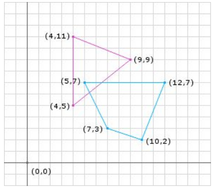
Dir = 1,0 -1,0   0,-1

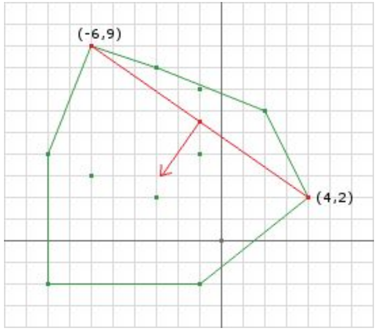


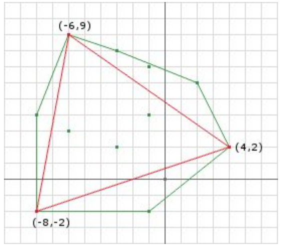
(9,9)-(5,7)   (4,5)-(12,7)   (4,5)-(5,7)

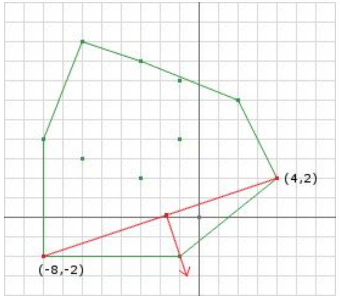
Collision Detection

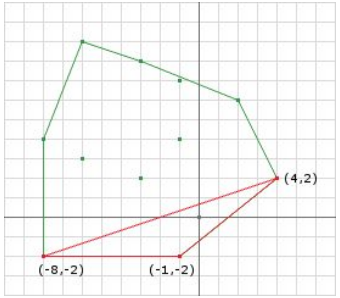
GJK Algorithm: origin in Minkowski Difference (MD)
→ iteratively approximate MD by simplices
Choose direction → origin   use barycentric coords







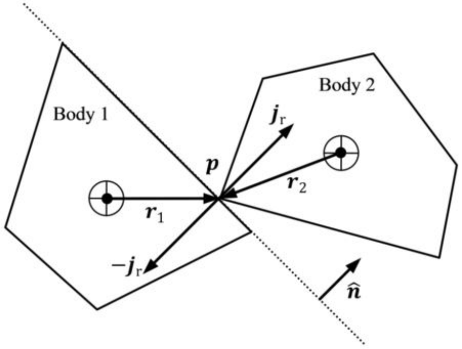




Dir = ((b-a)x(-a))x(b-a)

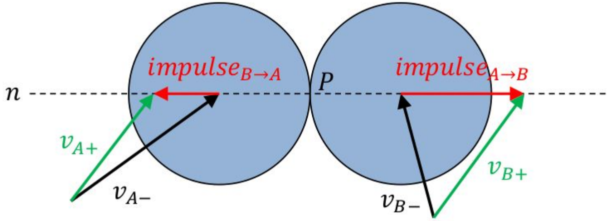
Collision Response

Direction & magnitude:
Elastic, fracture, stick,...



Contact point and normal
(normal of some separating plane)

Contact velocities



Impulse = change in momentum
Momentum = mass*velocity

Collision Response

Direction & magnitude:

- Elastic: billiard balls
- Inelastic: football players

Impulse J = momentum change = mass(v^+-v) = $\int_t^{t+} force$

(also for rotational component) **v** = velocity, $\mathbf{v}^+$ = velocity at next time step

Conserved: $m_A \mathbf{v}_A + m_B \mathbf{v}_B = m_A \mathbf{v}^+_A + m_B \mathbf{v}^+_B$

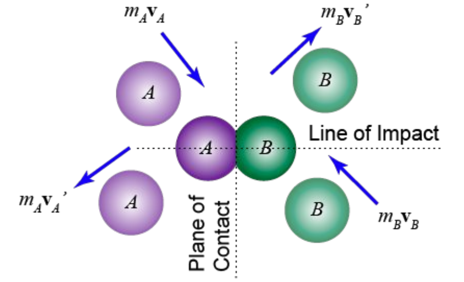


Figure 9.4-2: Line of impact and plane of contact

Collision Response

Impulse = momentum change = $mass(\mathbf{v}^+ - \mathbf{v}) = \int_t^{t^+} force$

Conserved: $m_A \mathbf{v}_A + m_B \mathbf{v}_B = m_A \mathbf{v}_A^+ + m_B \mathbf{v}_B^+$

$m_A \mathbf{v}_A + j\mathbf{n} = m_A \mathbf{v}_A^+$ $\quad$ $m_B \mathbf{v}_B - j\mathbf{n} = m_B \mathbf{v}_B^+$

(j=magnitude of J, n=normal)

3rd equation needed in 3-space!

Const $= -(\mathbf{v}_A^+ - \mathbf{v}_B^+) \cdot \mathbf{n} / (\mathbf{v}_A - \mathbf{v}_B) \cdot \mathbf{n}$ (coeff of restitution)

can include angular velocity

Const = 0 → stick Const = 1 → fully elastic Const <1 → damping

Collision Response

Impulse Conserved: $m_A\mathbf{v}_A + j\mathbf{n} = m_A\mathbf{v}_A^+$ $\quad m_B\mathbf{v}_B - j\mathbf{n} = m_B\mathbf{v}_B^+$

$(\cdot\mathbf{n}\ ) \rightarrow$ $j = m_A(\mathbf{v}_A^+ - \mathbf{v}_A)\cdot\mathbf{n}$ $\quad j = -m_B(\mathbf{v}_B^+ - \mathbf{v}_B)\cdot\mathbf{n}$ $\quad C = -\dfrac{(\mathbf{v}_A^+ - \mathbf{v}_B^+)\cdot\mathbf{n}}{(\mathbf{v}_A - \mathbf{v}_B)\cdot\mathbf{n}}$

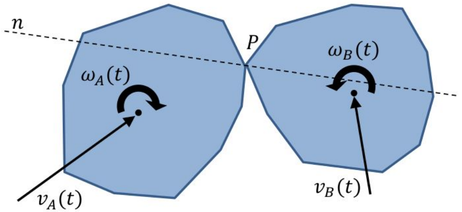
$$\frac{m_B}{m_A}j + j = m_B((\mathbf{v}_A^+ - \mathbf{v}_A)\cdot\mathbf{n} - (\mathbf{v}_B^+ - \mathbf{v}_B)\cdot\mathbf{n}) = -m_B(1+C)(\mathbf{v}_A - \mathbf{v}_B)\cdot\mathbf{n}$$

$$j = -(1+C)\frac{(\mathbf{v}_A - \mathbf{v}_B)\cdot\mathbf{n}}{(1/m_A + 1/m_B)}$$

$$\mathbf{v}_A^+ = \mathbf{v}_A + \frac{j}{m_A}\mathbf{n} \qquad \mathbf{v}_B^+ = \mathbf{v}_B - \frac{j}{m_B}\mathbf{n}$$

Larger mass → less velocity change

Collision Response



Angular Velocity (w) Separation normal off center

Collision Arm (r_A) = contact point - center

$\mathbf{V}_A = \mathbf{v}_A + \boldsymbol{\omega}_A \times \mathbf{r}_A$ (linear + rotational velocity)

(total velocity = CoM velocity + cross product of angular velocity and collision arm)

$\boldsymbol{\omega}_A^+ = \boldsymbol{\omega}_A + I^{-1}(\mathbf{r}_A \times j\mathbf{n})$ (I^{-1} = inverse of inertia tensor (3x3 matrix))

$\mathbf{v}_A^+ = \mathbf{v}_A + \frac{j}{m_A}\mathbf{n}$

$$j = \frac{-(1 + C_R)[(\bar{v}_{A-} - \bar{v}_{B-}) \cdot \hat{n}]}{\left(\frac{1}{m_A} + \frac{1}{m_B}\right) + [(\vec{r}_A \times \hat{n})^T I_A^{-1}(\vec{r}_A \times \hat{n}) + (\vec{r}_B \times \hat{n})^T I_B^{-1}(\vec{r}_B \times \hat{n})]}$$

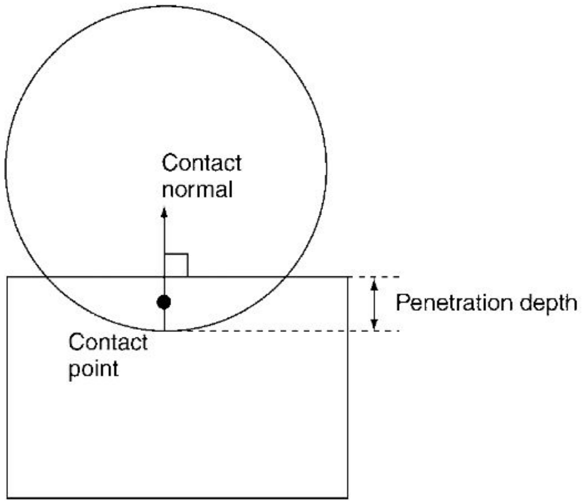
Augmented mass and inertia

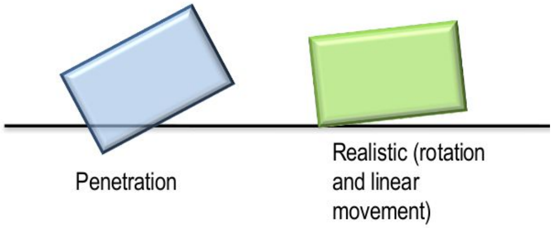
Pre-compute $I_A^{-1} I_B^{-1}$ in object coords map to world coords (Rot' I^{-1} Rot) $\rightarrow$ I^{-1}

Collision & Contact

Recall Tunneling
→ Interpenetration resolution →
Unrealistic for rotation

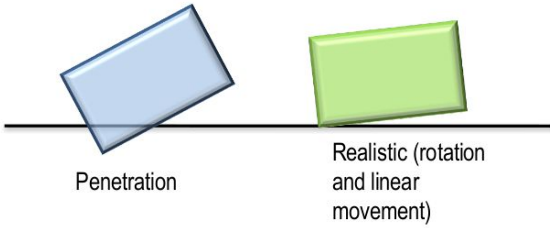
1. Push back in normal direction or
2. Push back & rotate

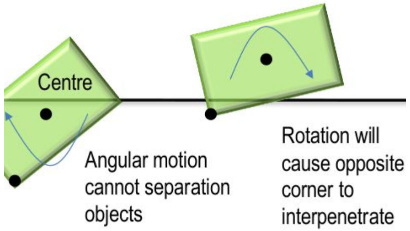


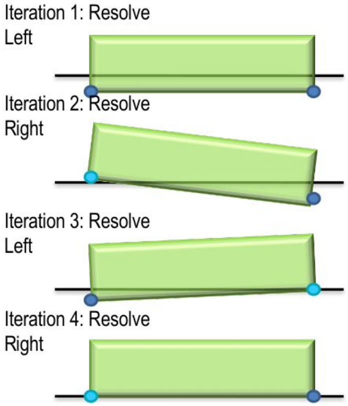


Collision & Contact

1. Push back in normal direction or
2. Push back & rotate
3. Avoid jitter by reducing C → 0







Collision Resolution

1. Collision Detection → Contact points, normal
2. Resolve interpenetration → Push back & rotate
3. Conservation of momentum → impulse
4. → Linear & angular velocity

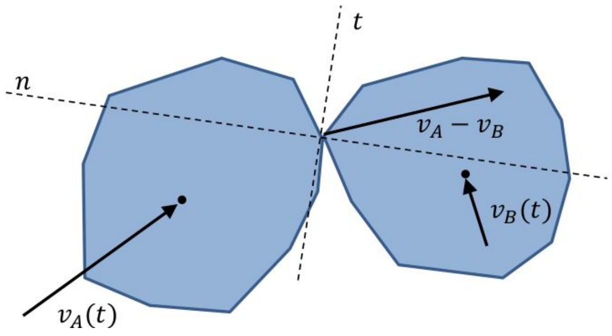
Collision Resolution

Friction force (tangential: $(n \times (v_A - v_b)) \times n$)

→ impulse

$v^+_A = v_A + j(n + \text{fric } t) / m_A$

etc.



- Two colliding non-rotating objects can create rotation.
- Is it creating angular momentum from linear momentum? No!
  - Both momentums are conserved, and are separable entities.
- Angular momentum is relative to a point and measures the entire system.
- Neither object rotates around its COM, but the system has angular momentum around the mutual COM.