CS230
Game Implementation Techniques

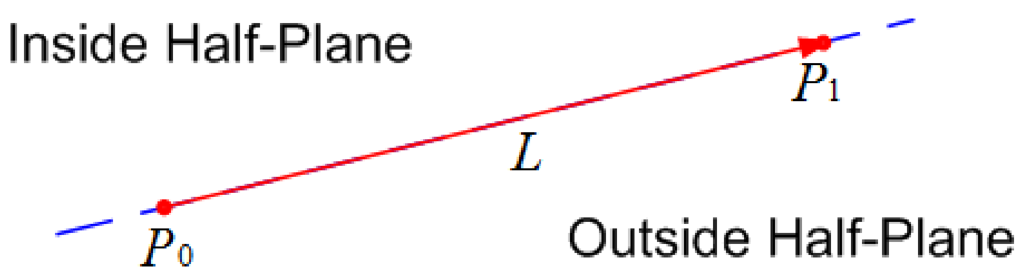
Lecture 16

Overview

- Normal Line Equation
- Animated Point to Line Classification

Line Segment & Half Planes

- Consider directed line segment L from position P_o to position P_1
- Infinite extension of L divides *XY-plane* into two half-planes
  - Half-plane on L's right-hand side is by (our) convention referred to as *outside* (or, *positive*) half-plane
  - Half-plane on L's left-hand side is *inside* (or, *negative*) half-plane

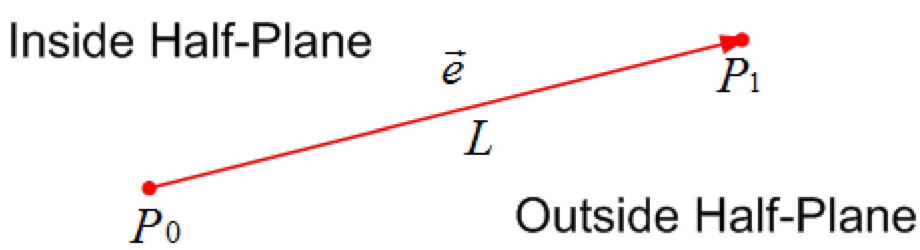


Line Segment: Edge Vector

- Compute L's edge vector

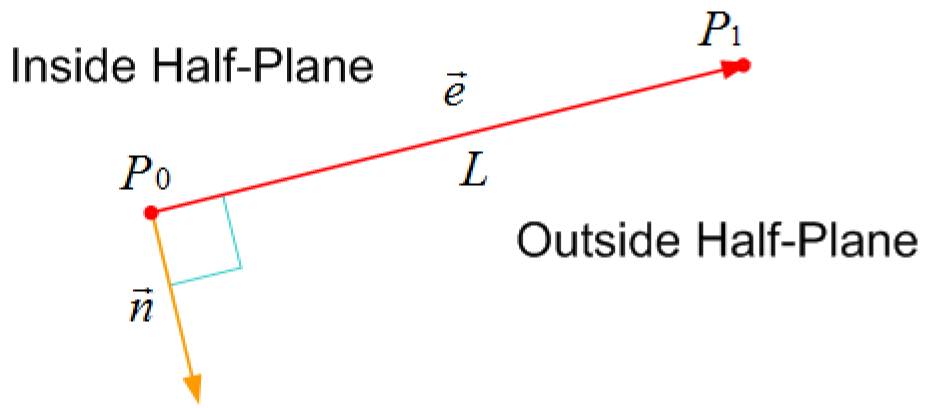
$$\vec{e} = P_1 - P_0 = (e_x, e_y)$$

$$\Rightarrow (e_x, e_y) = (x_1 - x_0, y_1 - y_0)$$



Outward Normal of Line Segment

- What is *outward normal* to line segment L?
  - Vector *n* is orthogonal to L's edge vector *e* such that *n* is oriented from L's inside to outside half-plane



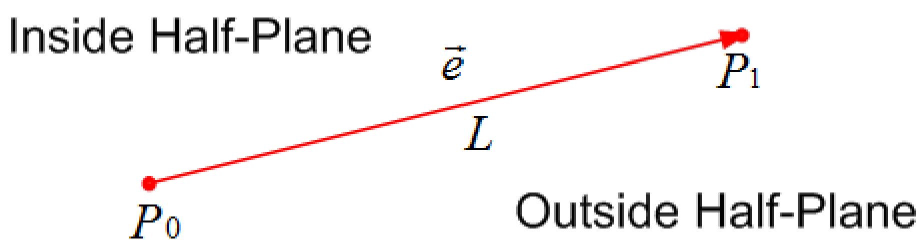
Computing Outward Normal (1/3)

- How is the outward normal *n* to line segment L computed?
  - Rotate edge vector *e* thro' –90° about Z-axis
  - That is, edge vector *e* is rotated about Z-axis in *clockwise direction* through 90°

Computing Outward Normal (2/3)

- First, compute directed line segment L's edge vector *e*:

$$\vec{e} = (e_x, e_y) = P_1 - P_0 = (x_1 - x_0, y_1 - y_0)$$

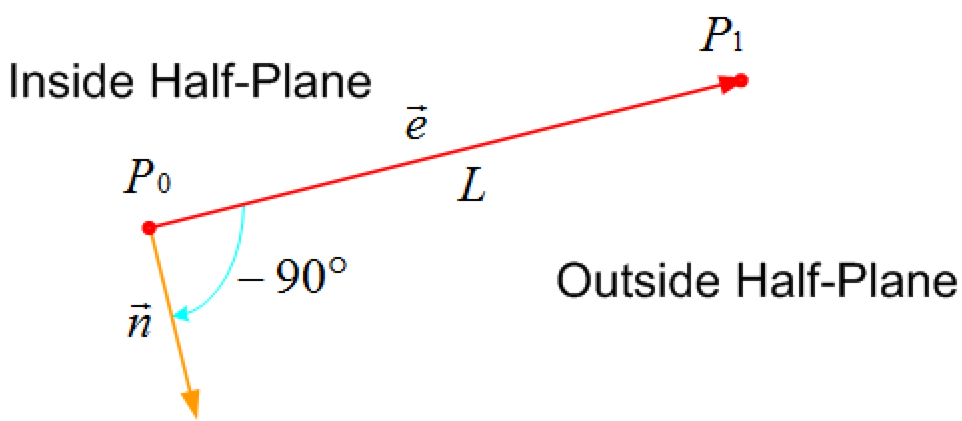


Computing Outward Normal (3/3)

- To compute outward normal n, rotate edge vector e about Z-axis thro' −90°

$$\begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} \cos(-90^\circ) & -\sin(-90^\circ) \\ \sin(-90^\circ) & \cos(-90^\circ) \end{bmatrix} \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$

$$\Rightarrow \vec{n} = (n_x, n_y) = (e_y, -e_x)$$



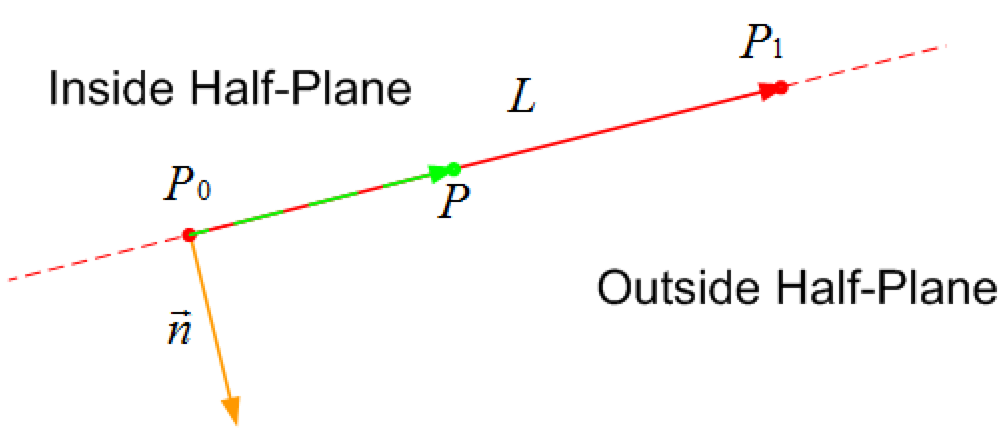
Overview

- Normal Line Equation
- Animated Point to Line Classification

Point-Normal Line Equation (1/2)

- Let *P(x, y)* be an arbitrary point on *L's* infinite extension
- Point-normal equation of L is:

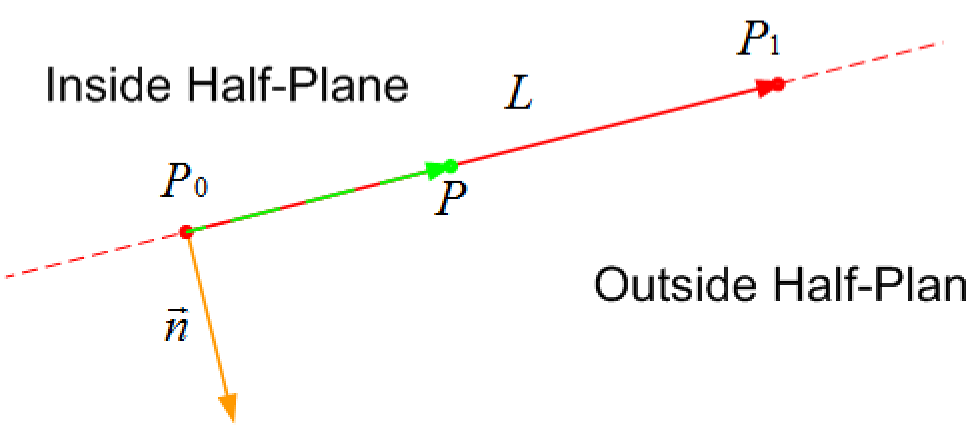
$$\vec{n} \bullet (P - P_0) = 0 \Rightarrow \vec{n} \bullet P - \vec{n} \bullet P_0 = 0$$



Point-Normal Line Equation (2/2)

- Better to use normalized outward normal
- Using normalized outward normal, point-normal equation of line segment L from point P_0 to point P_1 is written as:

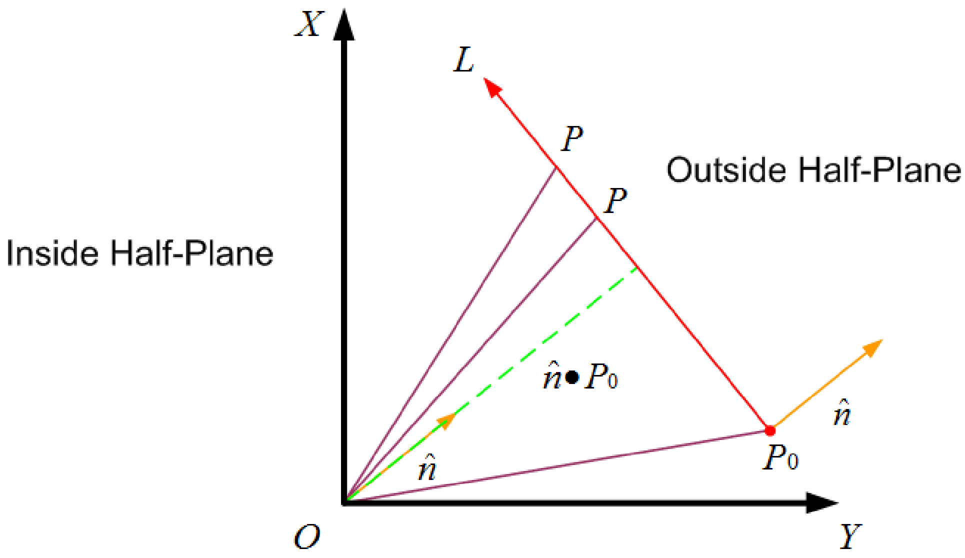
$$L : \hat{n} \bullet P - \hat{n} \bullet P_0 = 0$$



$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|}$$

Geometrical Interpretation

- Projections of position vectors of each of the infinite points *P* on L onto normalized outward normal *n* will result in same value
  - Value is *orthogonal* (or, *shortest*) distance from coordinate system origin to L

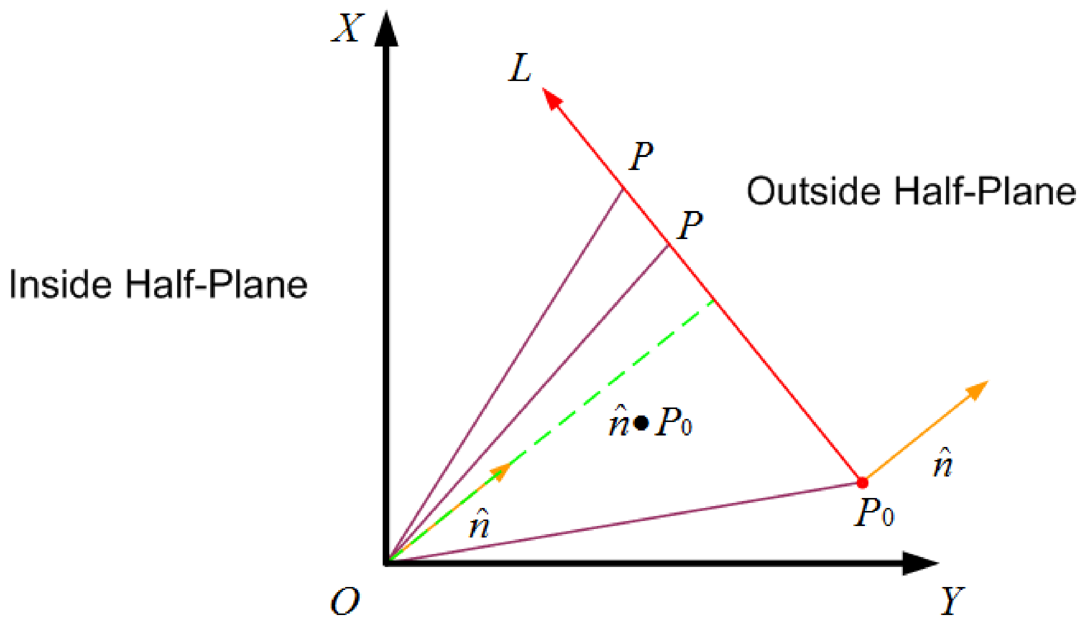


$$L : \hat{n} \bullet P - \hat{n} \bullet P_0 = 0$$

Point-Line Classification (1/3)

$\hat{n} \bullet P - \hat{n} \bullet P_0 = 0 \Leftrightarrow$

P is co-linear with L

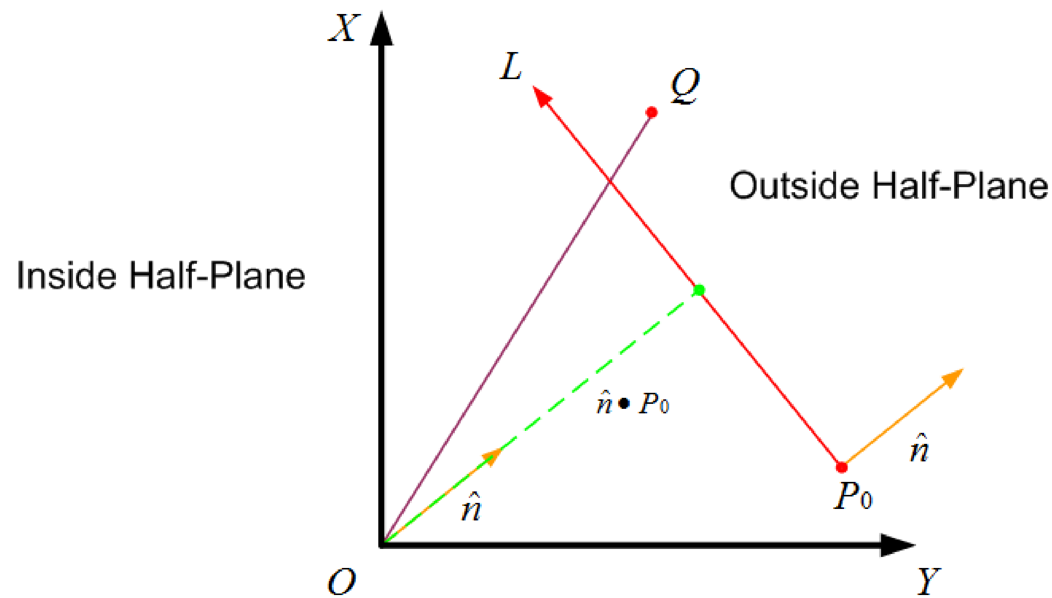


- Distance between origin and arbitrary point *P* (measured along normalized normal to L) is equal to shortest distance from origin to line segment L
- This implies that *P* must lie on infinite extension of L

Point-Line Classification (2/3)

$$\hat{n} \bullet Q - \hat{n} \bullet P_0 > 0 \Leftrightarrow$$

Q in outside half-plane of L

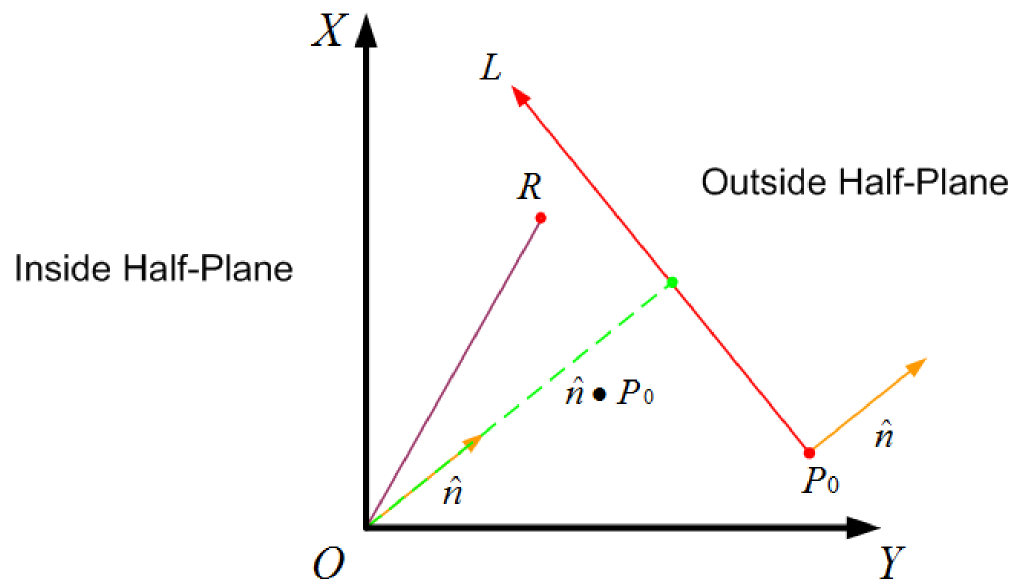


- Distance between origin and arbitrary point Q (measured along normalized normal to L) is greater than shortest distance from origin to line segment L
- This implies that Q must lie in outside half-plane of L

Point-Line Classification (3/3)

$$\hat{n} \bullet R - \hat{n} \bullet P_0 < 0 \Leftrightarrow$$

R in the inside half-plane of L



- Distance between origin and arbitrary point *R* (measured along normalized normal to L) is smaller than shortest distance from origin to line segment L
- This implies that *R* must lie in inside half-plane of L

Boundary Condition of Point

- Evaluation of an arbitrary point in line segment's point-normal equation is called *boundary condition of* arbitrary point with respect to the line segment
- Boundary condition of arbitrary point *P with respect to* line segment L is:

$$BC_L^P = \hat{n} \,.\, (P - P_0)$$

- Boundary condition BC_L^P evaluates to three results:
  - Positive ⇔ Point *P* in outside half-plane of line segment *L*
  - Negative ⇔ Point *P* in inside half-plane of line segment *L*
  - Zero ⇔ Point *P* on line segment *L*

Overview

- Static Collision
  - Point/Circle
  - Circle/Circle
  - Point/Rectangle
  - Rectangle/Rectangle
- Normal Line Equation
- Animated Point to Line Classification

Collision Experiment

- Given:
  - Static wall of finite length and infinitesimal thickness
  - Animated pinball with an infinitesimal radius
- Problem:
  - Ensure animated pinball correctly collides and bounces off wall

Geometrical and Mathematical Model of Wall

- Geometrical model of wall
  - Directed line segment from position P_0 to position P_1
- Mathematical model of wall
  - Infinite extension of directed line segment from position P_0 to position P_1
  - L: $n \bullet P - n \bullet P_0 = 0$
    - n is the normalized outward normal of directed line segment from position P_0 to position P_1
    - P is any arbitrary point on infinite extension of line segment
    - $n \bullet P_0$ is the orthogonal distance from origin to line segment

Modeling Pinball Animation

- Pinball modeled as an infinitesimal point
- Located at points B_s and B_e at times t_s (frame start time) and t_e (frame end time), respectively within the current frame
- Moving with speed k units along direction given by normalized vector v
- Pinball location during current frame is modeled as

$$\boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \boldsymbol{k}\hat{\boldsymbol{v}}\boldsymbol{t}, \boldsymbol{t} \in [0,1]$$

- Velocity per **frame** $\vec{\boldsymbol{v}} = \overrightarrow{\boldsymbol{B}_s\boldsymbol{B}_e}$

$$\Rightarrow \boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}, \boldsymbol{t} \in [0,1]$$

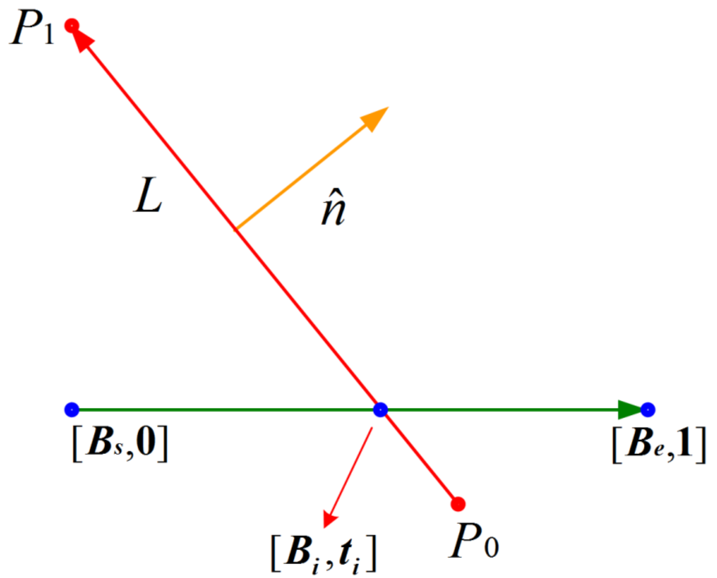
Intersection Between Wall and Animated Ball

Ball modeled as : $\boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$

Wall modeled as $\boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \boldsymbol{0}$

$$\boldsymbol{t}_i = \frac{\hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 - \hat{\boldsymbol{n}} \bullet \boldsymbol{B}_s}{\hat{\boldsymbol{n}} \bullet \vec{\boldsymbol{v}}} \text{ and } \boldsymbol{t}_i \in [0,1]$$

$$\boldsymbol{B}_i = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\left(\frac{\hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 - \hat{\boldsymbol{n}} \bullet \boldsymbol{B}_s}{\hat{\boldsymbol{n}} \bullet \vec{\boldsymbol{v}}}\right)$$

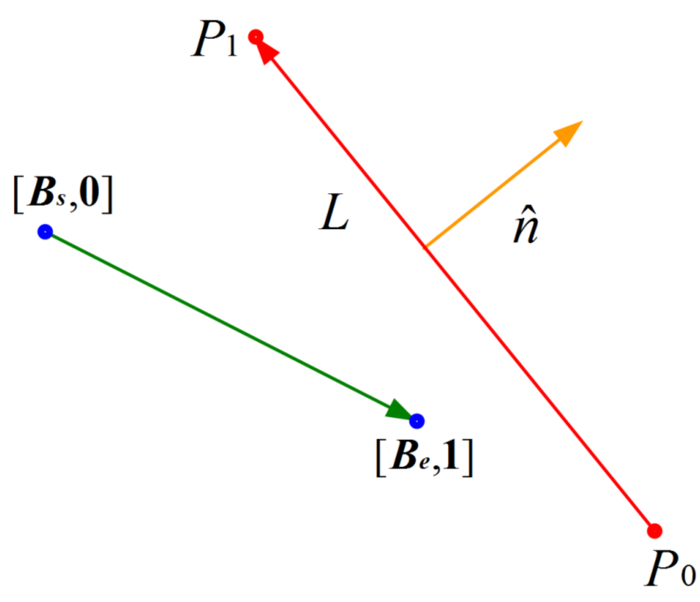


Test for Non-Collision (1/5)

$$\text{Ball modeled as}: \boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B_s} + \vec{\boldsymbol{v}}\boldsymbol{t}$$

$$\text{Wall modeled as } \boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P_0} = \boldsymbol{0}$$

$$(\hat{\boldsymbol{n}} \bullet \boldsymbol{B_s} < \hat{\boldsymbol{n}} \bullet \boldsymbol{P_0}) \,\&\&\, (\hat{\boldsymbol{n}} \bullet \boldsymbol{B_e} < \hat{\boldsymbol{n}} \bullet \boldsymbol{P_0})$$

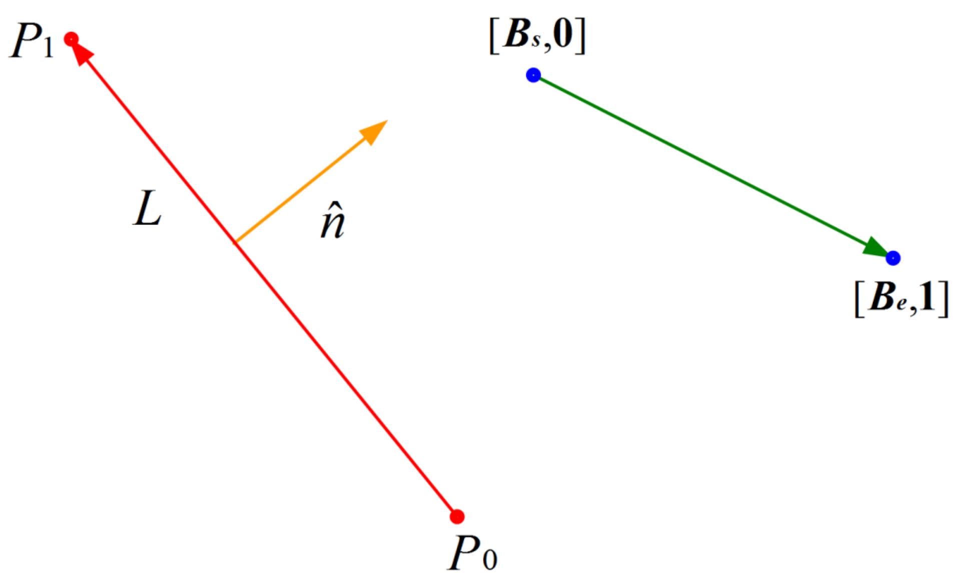


Test for Non-Collision (2/5)

$$\text{Ball modeled as}: \boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$$

$$\text{Wall modeled as } \boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \boldsymbol{0}$$

$$(\hat{\boldsymbol{n}} \bullet \boldsymbol{B}_s > \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0) \,\&\&\, (\hat{\boldsymbol{n}} \bullet \boldsymbol{B}_e > \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0)$$

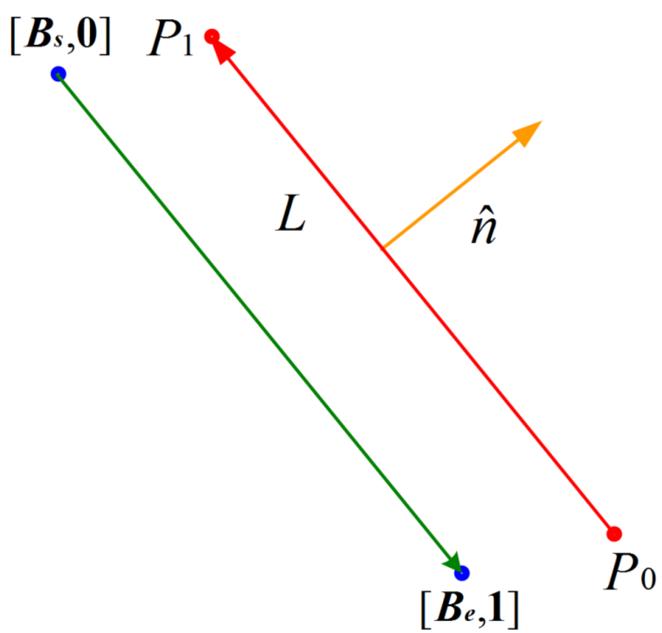


Test for Non-Collision (3/5)

$$\text{Ball modeled as}: \boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$$

$$\text{Wall modeled as } \boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \boldsymbol{0}$$

$$\hat{\boldsymbol{n}} \bullet \vec{\boldsymbol{v}} = \boldsymbol{0}$$

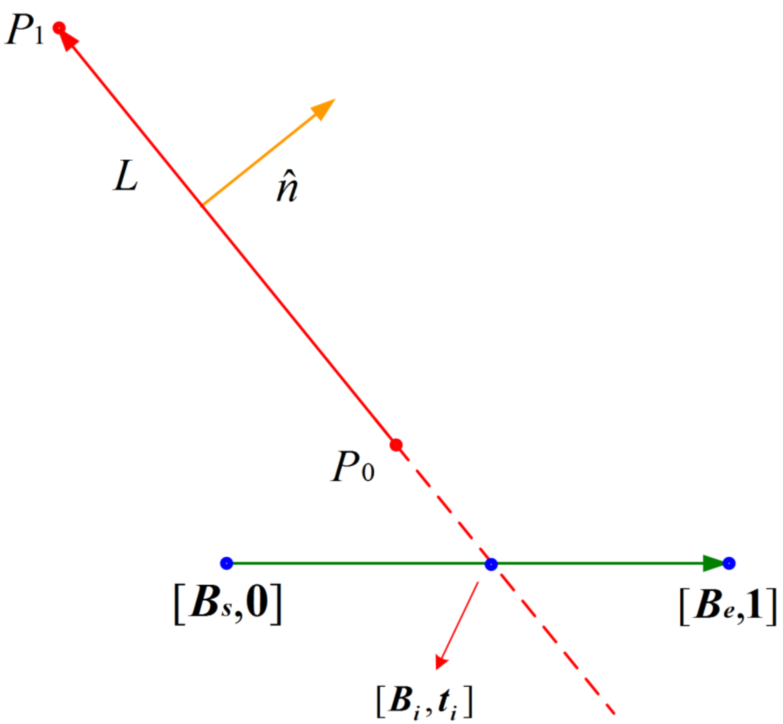


Test for Non-Collision (4/5)

Ball modeled as : $\boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$

Wall modeled as $\boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \boldsymbol{0}$

$(\boldsymbol{B}_i - \boldsymbol{P}_0) \bullet (\boldsymbol{P}_1 - \boldsymbol{P}_0) < \boldsymbol{0}$



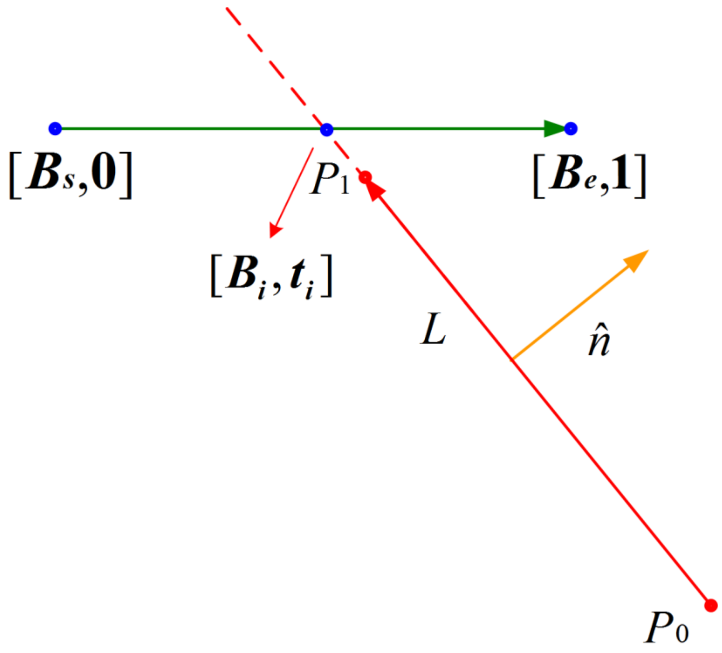
- Ball collides with infinite extension of wall … not finite wall!

Test for Non-Collision (5/5)

$$\text{Ball modeled as}: \boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$$

$$\text{Wall modeled as } \boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \mathbf{0}$$

$$(\boldsymbol{B}_i - \boldsymbol{P}_1) \bullet (\boldsymbol{P}_0 - \boldsymbol{P}_1) < \mathbf{0}$$



- Ball collides with infinite extension of wall … not finite wall!

Collision of Animated Ball with Wall

Ball modeled as : $\boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\boldsymbol{t}$

Wall modeled as $\boldsymbol{L}: \hat{\boldsymbol{n}} \bullet \boldsymbol{P} - \hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 = \boldsymbol{0}$

$$\boldsymbol{t}_i = \frac{\hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 - \hat{\boldsymbol{n}} \bullet \boldsymbol{B}_s}{\hat{\boldsymbol{n}} \bullet \vec{\boldsymbol{v}}} \text{ and } \boldsymbol{t}_i \in [0,1]$$

$$\boldsymbol{B}_i = \boldsymbol{B}_s + \vec{\boldsymbol{v}}\left(\frac{\hat{\boldsymbol{n}} \bullet \boldsymbol{P}_0 - \hat{\boldsymbol{n}} \bullet \boldsymbol{B}_s}{\hat{\boldsymbol{n}} \bullet \vec{\boldsymbol{v}}}\right)$$