Search and Intersection

O'Rourke, Chapter 7

Outline

- Review
  - Barycentric Coordinates
- Primitive Intersection

Segment-Triangle Intersection

<u>Barycentric Coordinates</u>:

Given points $v_1, \ldots, v_n \in \mathbb{R}^d$ (with $n \leq d$) a point p in the $(n-1)$-dimensional hyperplane passing through the $\{v_i\}$ can be expressed as:

$$p = \sum_{i=1}^{n} \lambda_i \cdot v_i \quad \text{with} \quad 1 = \sum_{i=1}^{n} \lambda_i \,.$$

Segment-Triangle Intersection

Barycentric Coordinates:

Given points $v_1, \ldots, v_n \in \mathbb{R}^d$ (with $n-1 \leq d$) a point p in the $(n-1)$-dimensional hyperplane passing through the $\{v_i\}$ can be expressed as:

$$p = \sum_{i=1}^{n} \lambda_i v_i \quad \text{with} \quad 1 = \sum_{i=1}^{n} \lambda_i \,.$$

The point p is in the convex hull of the points if $\lambda_i \geq 0$ for all $1 \leq i \leq n$.

Segment-Triangle Intersection

Barycentric Coordinates:

If a point p is on the plane passing through the $\{v_i\}$ we can get the barycentric coordinates of p by solving the (over-constrained) $n \times d$ linear system:

$$A(\lambda) = p \quad \Leftrightarrow \quad \begin{pmatrix} v_1^1 & \cdots & v_n^1 \\ \vdots & \ddots & \vdots \\ v_1^d & \cdots & v_n^d \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} p^1 \\ \vdots \\ p^d \\ 1 \end{pmatrix}$$

Segment-Triangle Intersection

Barycentric Coordinates:

If a point p is on the plane passing through the $\{v_i\}$ we can get the barycentric coordinates of p by solving the (over-constrained) $n \times d$ linear system:

$$A(\lambda) = p \quad \Leftrightarrow \quad \begin{pmatrix} v_1^1 & \cdots & v_n^1 \\ \vdots & \ddots & \vdots \\ v_1^d & \cdots & v_n^d \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} p^1 \\ \vdots \\ p^d \\ 1 \end{pmatrix}$$

In general, the least-squares solution is given by the solution to the *normal equation*:

$$(A^\top A)\lambda = A^\top p$$

Segment-Triangle Intersection

Barycentric Coordinates:

If a point p is on the plane passing through the $\{v_i\}$ we can get the barycentric coordinates of p by so… …n:

Since p is on the plane passing through the $\{v_i\}$, the least-squares solution is the exact solution.

A …

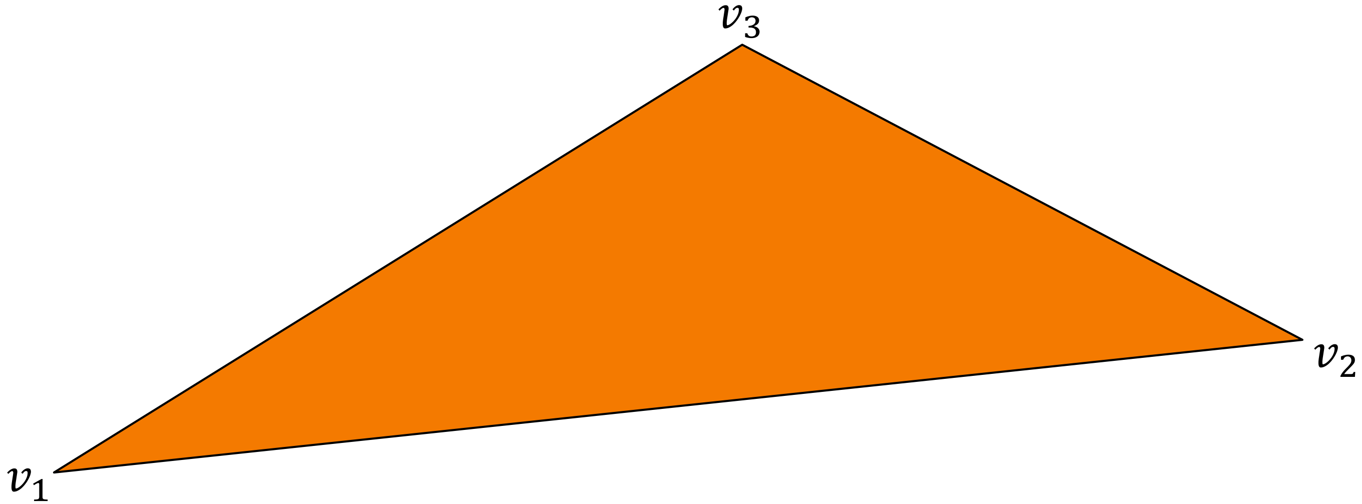
The matrix $A^\top A$ will be non-singular if and only if the $\{v_i\}$ are linearly independent.

In general, the least-squares solution is given by the solution to the *normal equation*:

$$(A^\top A)\lambda = A^\top p$$

Point-Triangle Intersection

Barycentric Coordinates ($n = 3$):

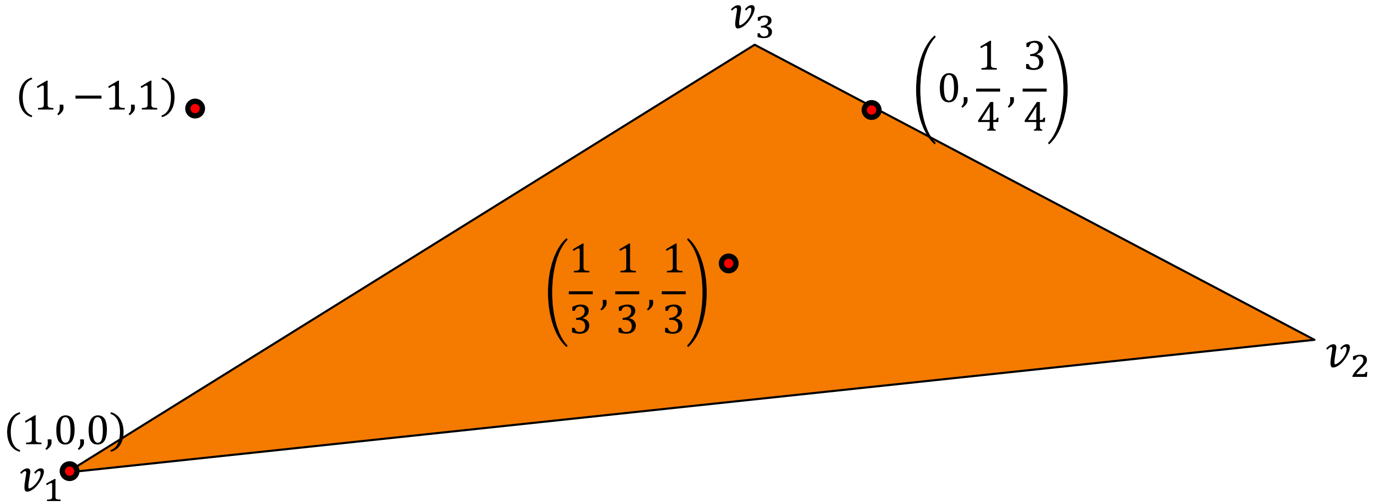


For a point p with barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$:

- p is outside the triangle if $\lambda_i < 0$
- p is on an edge if $0 \leq \lambda_i \leq 1$ and one of the λ_i is 0
- p is on a vertex if $0 \leq \lambda_i \leq 1$ and two of the λ_i are 0
- p is inside if $0 < \lambda_i < 1$.

Point-Triangle Intersection

Barycentric Coordinates ($n = 3$):



For a point p with barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$:

- p is outside the triangle if $\lambda_i < 0$
- p is on an edge if $0 \le \lambda_i \le 1$ and one of the λ_i is 0
- p is on a vertex if $0 \le \lambda_i \le 1$ and two of the λ_i are 0
- p is inside if $0 < \lambda_i < 1$.

Outline

- Review
- Primitive Intersection
  - Segment-Segment (2D)
  - Segment-Triangle (3D)
  - Point-Polygon/Polyhedron (2D/3D)

Intersection

Given primitives A and B, we would like to know **if/how** and **where** the primitives intersect.

Intersection

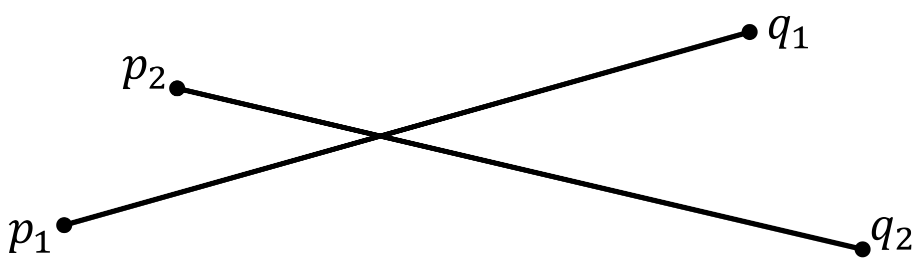
Given primitives A and B, we would like to know **if/how** and **where** the primitives intersect.

In general:

- Answering **if/how** can be done using integer arithmetic.
- Answering **where** requires using floating point precision (or at least rational numbers).

Segment-Segment Intersection

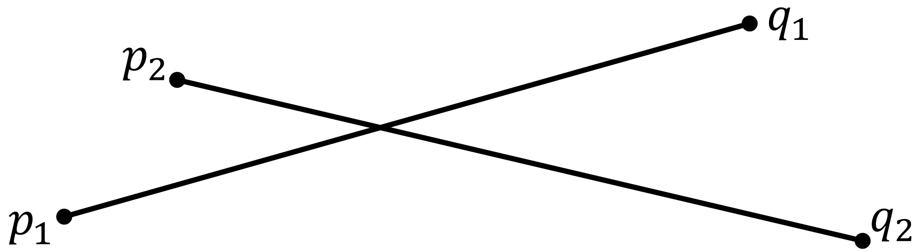
Given segments (p_1, q_1) and (p_2, q_2) in $\mathbb{R}^2$, we would like to determine **if/how** and **where** they intersect.



Segment-Segment Intersection

Points on $\overline{pq}$ can be expressed as:

$$\Phi(t) = p + t \cdot (q - p), \qquad t \in [0,1].$$



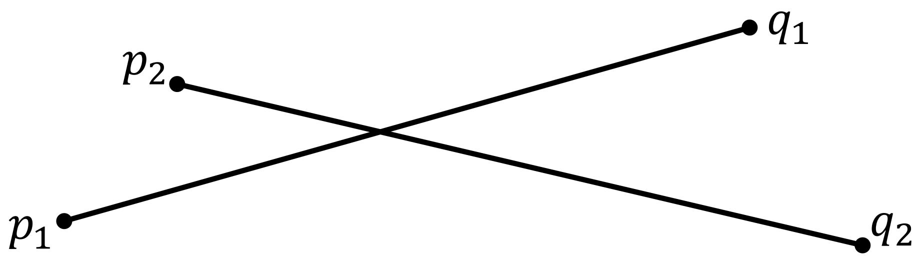
Segment-Segment Intersection

Points on $\overline{pq}$ can be expressed as:

$$\Phi(t) = p + t \cdot (q - p), \qquad t \in [0,1].$$

The **where** question can be computed by first intersecting the lines, solving:

$$p_1 + t_1(q_1 - p_1) = p_2 + t_2(q_2 - p_2)$$



Segment-Segment Intersection

Points on $\overline{pq}$ can be expressed as:

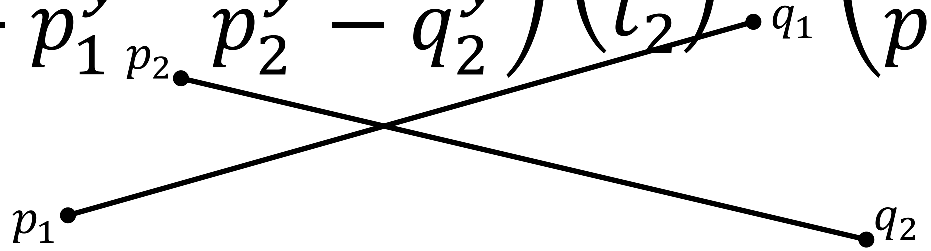
$$\Phi(t) = p + t \cdot (q - p), \qquad t \in [0,1].$$

The **where** question can be computed by first intersecting the lines, solving:

$$p_1 + t_1(q_1 - p_1) = p_2 + t_2(q_2 - p_2)$$

Rewriting, we get:

$$\begin{pmatrix} q_1^x - p_1^x & p_2^x - q_2^x \\ q_1^y - p_1^y & p_2^y - q_2^y \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} p_2^x - p_1^x \\ p_2^y - p_1^y \end{pmatrix}$$



Segment-Segment Intersection

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} q_1^x - p_1^x & p_2^x - q_2^x \\ q_1^y - p_1^y & p_2^y - q_2^y \end{pmatrix}^{-1} \begin{pmatrix} p_2^x - p_1^x \\ p_2^y - p_1^y \end{pmatrix}$$

The matrix is not invertible if the vectors $q_1 - p_1$ and $q_2 - p_2$ are linearly dependent (i.e. the segment directions are parallel).

Otherwise, if $t_i \in [0,1]$ they intersect.

Segment-Segment Intersection

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} q_1^x - p_1^x & p_2^x - q_2^x \\ q_1^y - p_1^y & p_2^y - q_2^y \end{pmatrix}^{-1} \begin{pmatrix} p_2^x - p_1^x \\ p_2^y - p_1^y \end{pmatrix}$$

The matrix is not invertible if the vectors $q_1 - p_1$ and $q_2 - p_2$ are linearly dependent (i.e. the segment directions are parallel).

Otherwise, if $t_1, t_2 \in [0,1]$ they intersect.

$p = p_1 + t_1(q_1 - p_1)$ gives the **where**.

Segment-Segment Intersection

If/How:

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} e \\ f \end{pmatrix} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix}$$

- Parallel $\Leftrightarrow ad - bc = 0$.
- $t_1 = 0 \Leftrightarrow de - bf = 0$.
- $t_2 = 0 \Leftrightarrow -ce + af = 0$.
- $t_1 = 1 \Leftrightarrow de - bf = ad - bc$.
- $t_2 = 1 \Leftrightarrow -ce + af = ad - bc$.

Segment-Segment Intersection

<u>If/How</u>:

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} e \\ f \end{pmatrix} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix}$$

- $\text{Parallel} \Leftrightarrow ad - bc = 0.$
- $t_1 = 0 \Leftrightarrow de - bf = 0.$
- $t_2 = 0 \Leftrightarrow -ce + af = 0.$
- $t_1 = 1 \Leftrightarrow de - bf = ad - bc$

Assuming integer coordinates, we can identify the **if/how** of the intersection using only integer arithmetic, using <u>twice</u> the number of bits of precision.

Segment-Segment Intersection

Parallel Intersection:

We have a parallel intersection if:

- The lines are parallel ($ad - bc = 0$)

And

- Points p_1, q_1, and p_2 are collinear

And

- Point p_1 is between q_1 and q_2, or
- Point p_2 is between q_1 and q_2, or
- Point q_1 is between p_1 and p_2, or
- Point q_2 is between p_1 and p_2.

Segment-Segment Intersection

Parallel Intersection:

We have a parallel intersection if:

- The lines are parallel ($ad - bc = 0$)

And

- Points p_1, q_1, and p_2 are collinear

And

- Point p_1 is between q_1 and q_2, or

We defined these predicates, when performing triangulation.

In the case of parallel segments, we can identify if there is an intersection. And, if there is, we can compute the interval of intersection.

Outline

- Review
- Primitive Intersection
  - Segment-Segment (2D)
  - Segment-Triangle (3D)
  - Point-Polygon/Polyhedron (2D/3D)

Segment-Triangle Intersection

Given a segment $\overline{pq}$ and a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$, an intersection can be computed first intersecting the line with the plane containing the triangle.

Segment-Triangle Intersection

Given a segment $\overline{pq}$ and a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$, an intersection can be computed first intersecting the line with the plane containing the triangle.

The plane containing the triangle is:

$$\pi = \{p \in \mathbb{R}^3 \,|\, \langle p, n\rangle - d = 0\}$$

with normal $n \in \mathbb{R}^3$ and distance $d \in \mathbb{R}$ to the origin:

$$n = (v_2 - v_1) \times (v_3 - v_1) \quad \text{and} \quad d = \langle v_1, n\rangle$$

Segment-Triangle Intersection

Given a segment $\overline{pq}$ and a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$, an intersection can be computed first intersecting the line with the plane containing the triangle.

The plane containing the triangle is:

$$\pi = \{p \in \mathbb{R}^3 \,|\, \langle p, n \rangle - d = 0\}$$

with normal $n \in \mathbb{R}^3$ and distance $d \in \mathbb{R}$ to the origin:

$$n = (v_2 - v_1) \times (v_3 - v_1) \quad \text{and} \quad d = \langle v_1, n \rangle$$

<u>Note</u>:

Representing requires twice the number of bits of precision.

Segment-Plane Intersection

Where:

The point of intersection is the solution to:

$$\langle p + t(q-p), n \rangle - d = 0,$$

or equivalently:

$$t = \frac{d - \langle p, n \rangle}{\langle q - p, n \rangle}$$

Segment-Plane Intersection

<u>Where</u>:

The point of intersection is the solution to:

$$\langle p + t(q - p), n \rangle - d = 0,$$

or equivalently:

$$t = \frac{d - \langle p, n \rangle}{\langle q - p, n \rangle}$$

There is a solution if $q - p$ and n are not orthogonal $\Leftrightarrow$ $\overline{pq}$ is not parallel to π.

Segment-Plane Intersection

Where:

The point of intersection is the solution to:

$$\langle p + t(q-p), n\rangle - d = 0,$$

or equivalently:

$$t = \frac{d - \langle p, n\rangle}{\langle q-p, n\rangle}$$

There is a solution if $q - p$ and n are not orthogonal $\Leftrightarrow$ $\overline{pq}$ is not parallel to π.

Otherwise, if $t \in [0,1]$ they intersect.

Segment-Plane Intersection

If/How:

$$t = \frac{d - \langle p, n \rangle}{\langle q - p, n \rangle}$$

- Parallel $\Leftrightarrow \langle q - p, n \rangle = 0$.
  - In plane $\Leftrightarrow d - \langle p, n \rangle = 0$.
- $t = 0 \Leftrightarrow d - \langle p, n \rangle = 0$.
- $t = 1 \Leftrightarrow d - \langle p, n \rangle = \langle q - p, n \rangle$.

Segment-Plane Intersection

If/How:

$$t = \frac{d - \langle p, n \rangle}{\langle q - p, n \rangle}$$

- Parallel $\Leftrightarrow \langle q - p, n \rangle = 0.$
  - In plane $\Leftrightarrow d - \langle p, n \rangle = 0.$
- $t = 0 \Leftrightarrow d - \langle p, n \rangle = 0.$
- $t = 1 \Leftrightarrow d - \langle p, n \rangle = \langle q - p, n \rangle.$

Assuming integer coordinates, we can answer **if/how** using only integer arithmetic, using three times the number of bits of precision.

Segment-Triangle Intersection

If/How:

Given the point of intersection of $\overline{pq}$ with the plane π, we can use barycenctric coordinates to test if the point of intersection is:

- On a triangle edge
- On a triangle vertex
- Inside the triangle
- Outside the triangle

Segment-Triangle Intersection

<u>If/How</u>:

Given the point of intersection of $\overline{pq}$ with the plane π, we can use barycenctric coordinates to test if the point of intersection is:

- On a triangle edge
- On a triangle vertex
- Inside the triangle
- Outside the triangle

<u>Challenge</u>:
This would entail making a discrete decision using floating point arithmetic (both for computing the point of intersection and the barycentric coordinates).

Segment-Triangle Intersection

After intersecting the segment with the plane, there are two cases:

1. The segment intersects the plane containing the triangle at one of the segment's endpoints.
2. The segment intersects the plane containing the triangle in the interior of the segment.

In either case we would like to answer the **if/how** of segment-triangle intersection.

Segment-Triangle Intersection

If/How (Case 1):

Suppose we have a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$ and a point p in the plane through the $\{v_i\}$.

The barycentric coordinates of the point p w.r.t. the $\{v_i\}$ are the same as the barycentric coordinates of the projection p w.r.t. the projection of the $\{v_i\}$, for any projection that doesn't "collapse" the triangle to a line segment.

Segment-Triangle Intersection

If/How (Case 1):

Suppose we have a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$ and a point p in the plane through the $\{v_i\}$.

The barycentric coordinates of the point p w.r.t. the $\{v_i\}$ are the same as the barycentric coordinates of the projection p w.r.t. the projection of the $\{v_i\}$, for any projection that doesn't "collapse" the triangle to a line segment.

That is, as long as the direction of projection is not perpendicular to the triangle's normal.

Segment-Triangle Intersection

<u>If/How (Case 1)</u>:

Suppose we have a triangle $\Delta v_1 v_2 v_3$ in $\mathbb{R}^3$ and a point p in the plane through the $\{v_i\}$.

The barycentric coordinates of the point p w.r.t. the $\{v_i\}$ are the same as the barycentric coordinates of the projection p w.r.t. the projection of the $\{v_i\}$, for any projection that doesn't "collapse" the triangle to a line segment.

$\Rightarrow$ We can reduce the problem of computing the barycentric coordinates from 2D to 3D by projecting out one of the coordinate axes.

Segment-Triangle Intersection

If/How (Case 1):

⇒ If the segment intersects the plane at one of the segment's endpoints, we have reduced the 3D segment-triangle **if/how** problem to the 2D point-triangle **if/how** problem.

Segment-Triangle Intersection

If/How (Case 1):

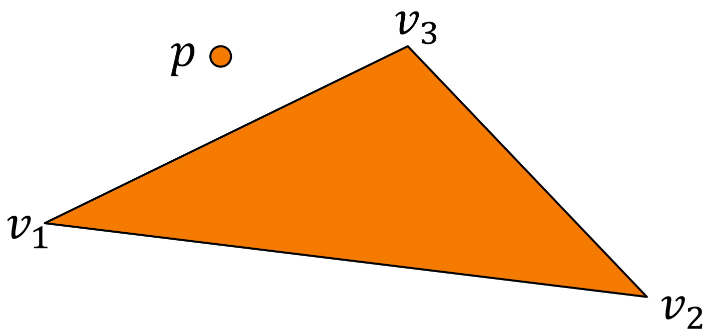
⇒ If the segment intersects the plane at one of the segment's endpoints, we have reduced the 3D segment-triangle **if/how** problem to the 2D point-triangle **if/how** problem.

Note:
In a similar way, we can use projection to reduce the case when the segment is in the triangle's plane to a 2D segment-triangle **if/how** problem.

Point-Triangle Intersection (2D)

Barycentric Coordinates (Geometric):

Compute the ratio of the signed areas of the $\Delta p v_i v_{i+1}$ with the signed area of $\Delta v_1 v_2 v_3$.

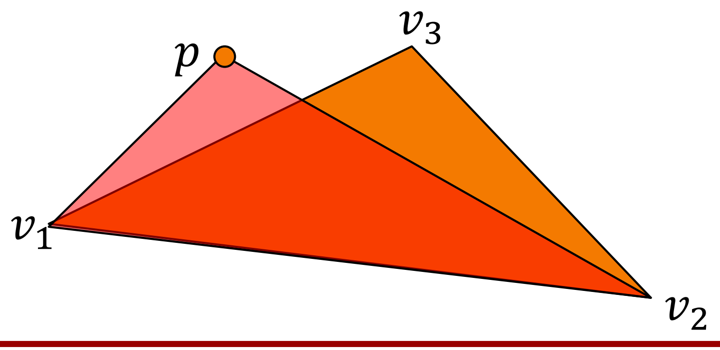


Point-Triangle Intersection (2D)

Barycentric Coordinates (Geometric):

Compute the ratio of the signed areas of the $\Delta p v_i v_{i+1}$ with the signed area of $\Delta v_1 v_2 v_3$.

$$\alpha = \frac{\text{Area}(p, v_1, v_2)}{\text{Area}(v_1, v_2, v_3)} > 0$$



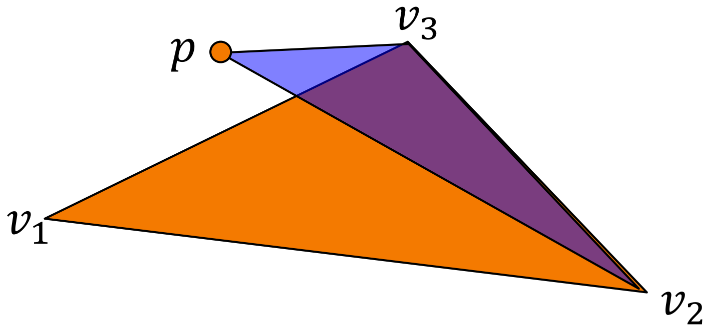
Point-Triangle Intersection (2D)

Barycentric Coordinates (Geometric):

Compute the ratio of the signed areas of the $\Delta p v_i v_{i+1}$ with the signed area of $\Delta v_1 v_2 v_3$.

$$\alpha = \frac{\text{Area}(p, v_1, v_2)}{\text{Area}(v_1, v_2, v_3)} > 0$$

$$\beta = \frac{\text{Area}(p, v_2, v_3)}{\text{Area}(v_1, v_2, v_3)} > 0$$



Point-Triangle Intersection (2D)

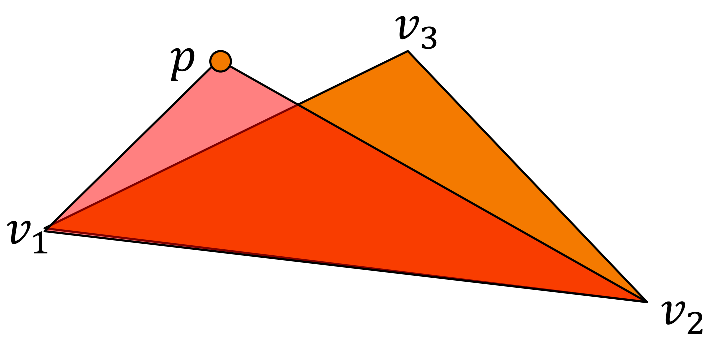
Barycentric Coordinates (Geometric):

Compute the ratio of the signed areas of the $\Delta p v_i v_{i+1}$ with the signed area of $\Delta v_1 v_2 v_3$.

$$\alpha = \frac{\text{Area}(p, v_1, v_2)}{\text{Area}(v_1, v_2, v_3)} > 0$$

$$\beta = \frac{\text{Area}(p, v_2, v_3)}{\text{Area}(v_1, v_2, v_3)} > 0$$

$$\gamma = \frac{\text{Area}(p, v_3, v_1)}{\text{Area}(v_1, v_2, v_3)} < 0$$



Point-Triangle Intersection (2D)

Barycentric Coordinates (Geometric):

Compute the ratio of the signed areas of the $\Delta p v_i v_{i+1}$ with the signed area of $\Delta v_1 v_2 v_3$.

Note:

$$\text{Area}(p, v_1, v_2) + \text{Area}(p, v_2, v_3) + \text{Area}(p, v_3, v_1) = \text{Area}(v_1, v_2, v_3)$$

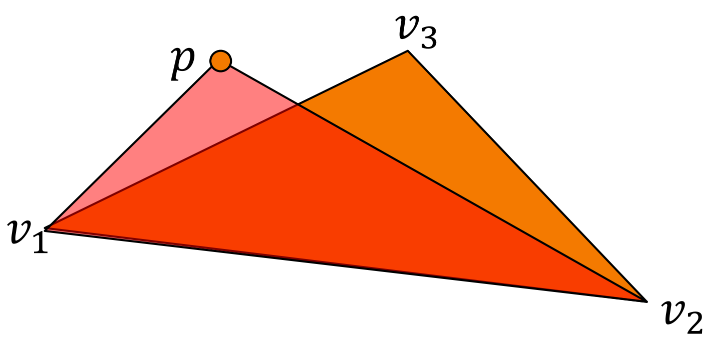
$$\Updownarrow$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha = \frac{\text{Area}(p, v_1, v_2)}{\text{Area}(v_1, v_2, v_3)} > 0$$

$$\beta = \frac{\text{Area}(p, v_2, v_3)}{\text{Area}(v_1, v_2, v_3)} > 0$$

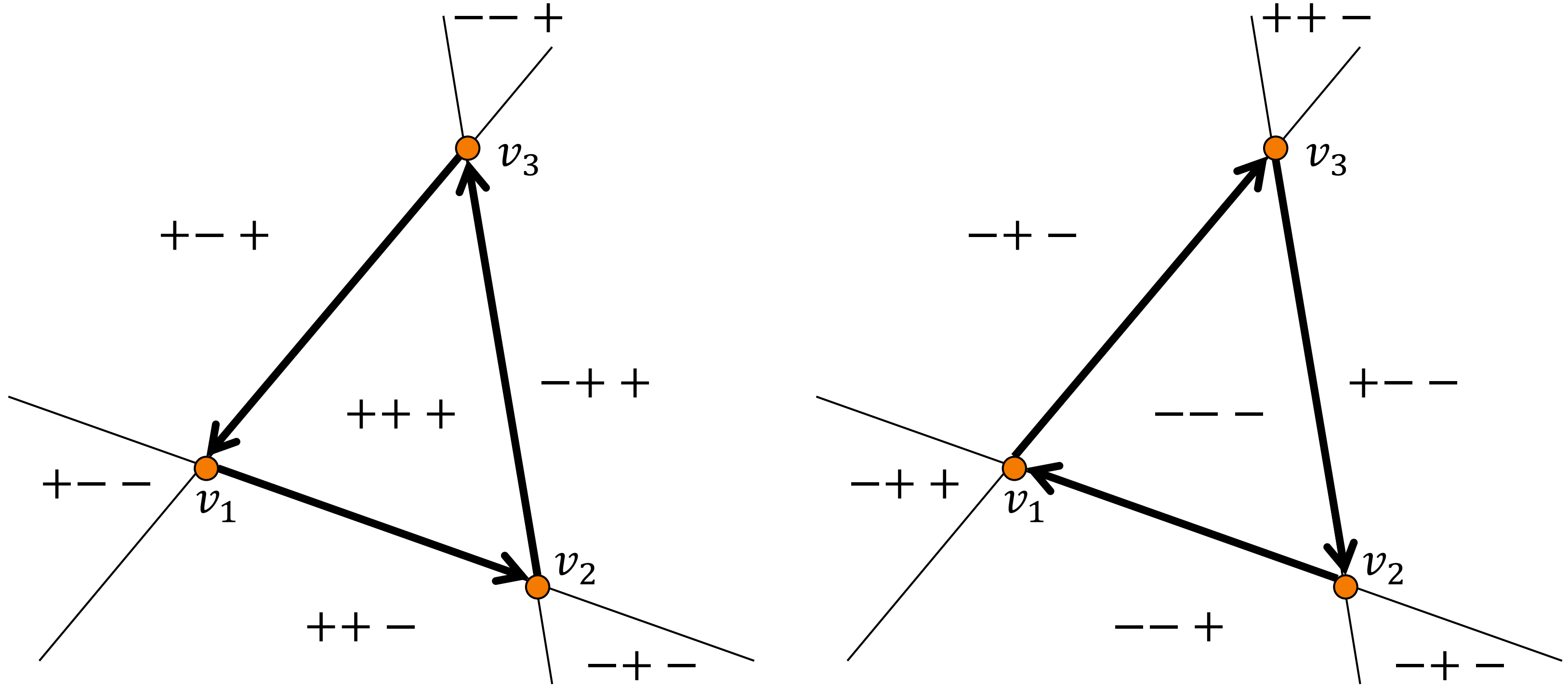
$$\gamma = \frac{\text{Area}(p, v_3, v_1)}{\text{Area}(v_1, v_2, v_3)} < 0$$



Point-Triangle Intersection (2D)

<u>Areas</u>:

We can test if there is an intersection by looking at the signs of the areas of the triangles made between the point and the edges.



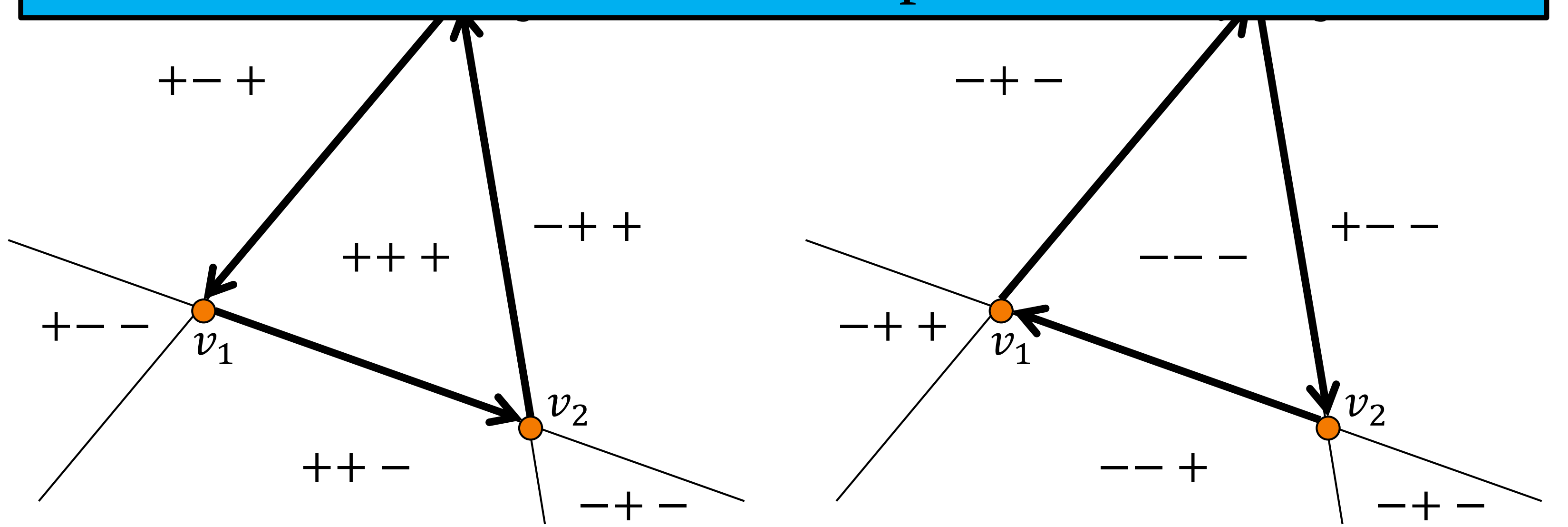
Point-Triangle Intersection (2D)

<u>Classification</u>:

- If all area have the same sign, the point is interior.
- If two have the same sign and one is zero, the point is on an edge.
- If two signs are zero, the point is at a vertex.

Assuming integer coordinates, we can answer the **if/how** question using integer arithmetic with twice number of bits of precision.

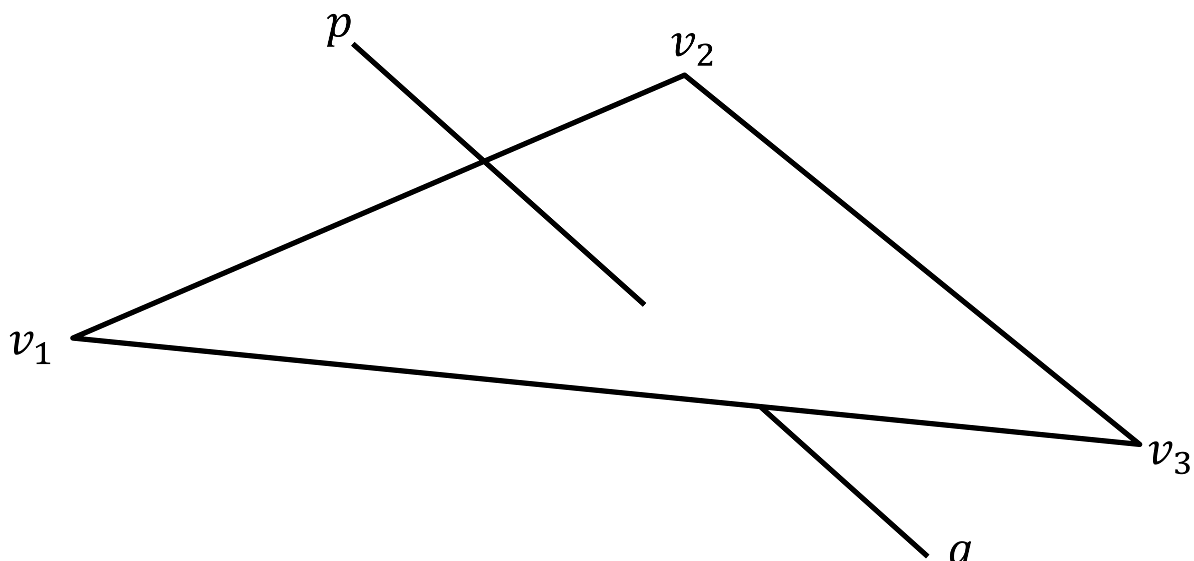
b



Segment-Triangle Intersection

If/How (Case 2):

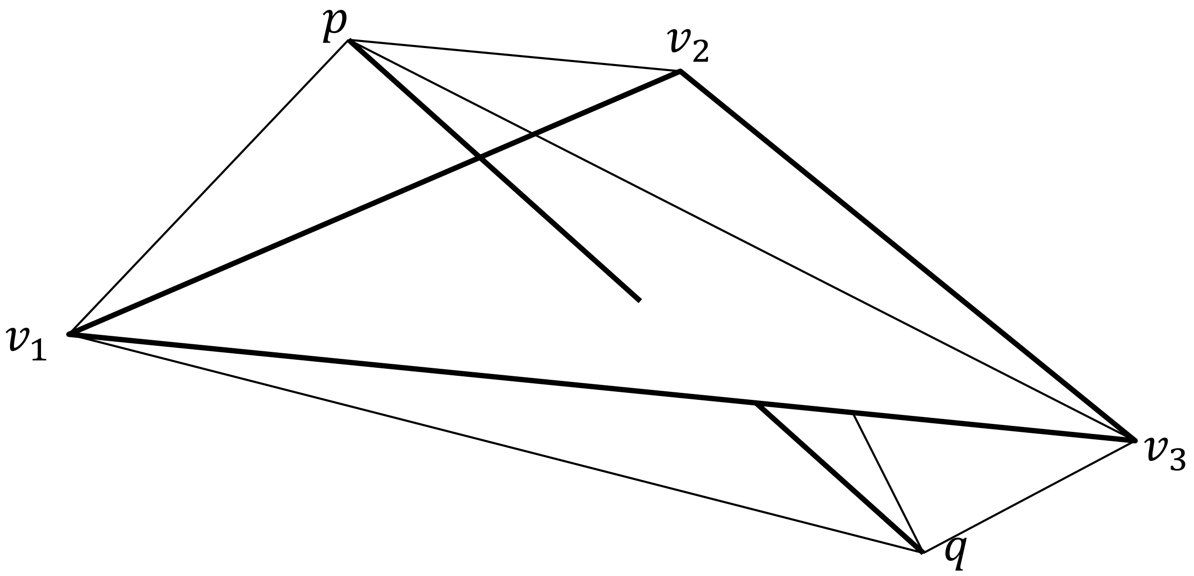
If the $\overline{pq}$ intersects π properly, we can classify the point of intersection by considering the volumes of the three tetrahedra.



Segment-Triangle Intersection

If/How (Case 2):

If the $\overline{pq}$ intersects π properly, we can classify the point of intersection by considering the volumes of the three tetrahedra.

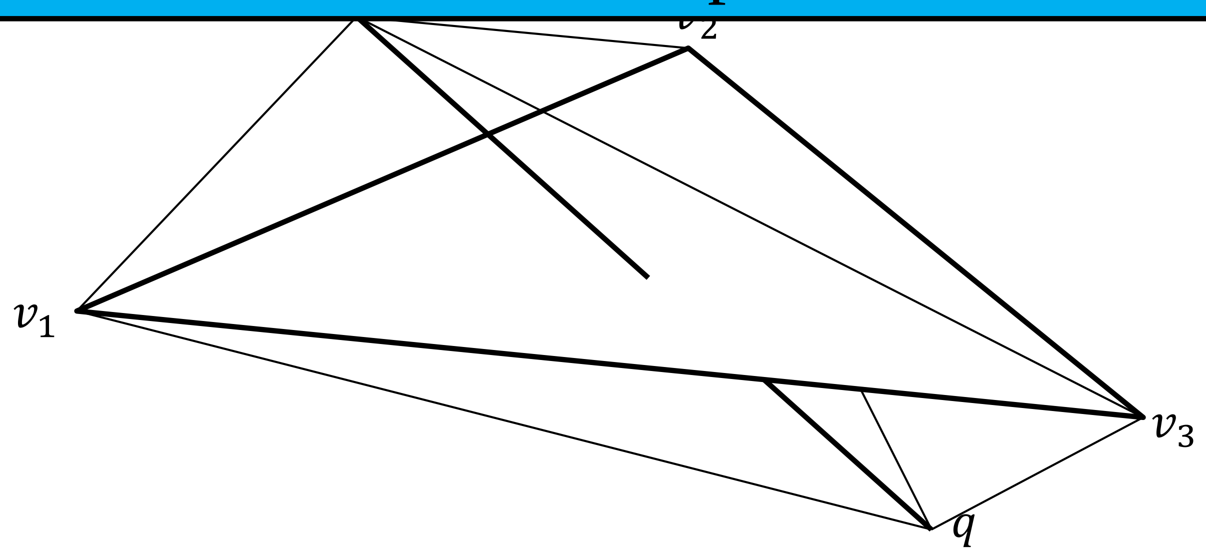


Segment-Triangle Intersection (3D)

Classification:

- If all volumes have the same sign, the point is interior.
- If two have the same sign and one is zero, the point is on an edge.
- If two signs are zero, the point is at a vertex.

Assuming integer coordinates, we can answer the **if/how** question using integer arithmetic with thrice the number of bits of precision.



Segment-Triangle Intersection

If/How:

- If end-points are on the plane:
  - » Do point-triangle (2D) intersection
- If interior of the edge crosses:
  - » Do segment-triangle (3D) intersection
- If segment and plane are parallel
  - » …
- Otherwise
  - » No intersection

Note:

- All the predicates can be performed using integer arithmetic.
- Implementation requires three times the number of bits of precision.

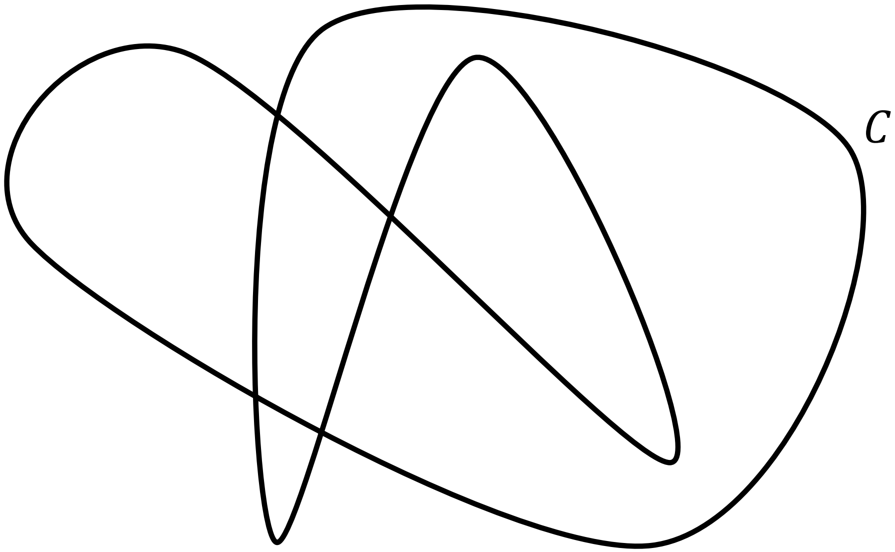
Outline

- Review
- Primitive Intersection
  - Segment-Segment (2D)
  - Segment-Triangle (3D)
  - Point-Polygon/Polyhedron (2D/3D)
    - » Winding Number
    - » Parity Test

Point-Polygon/Polyhedron

Winding Number:

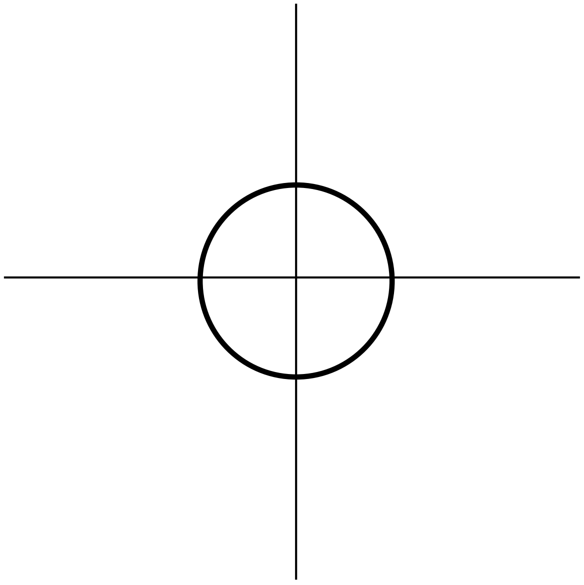
Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

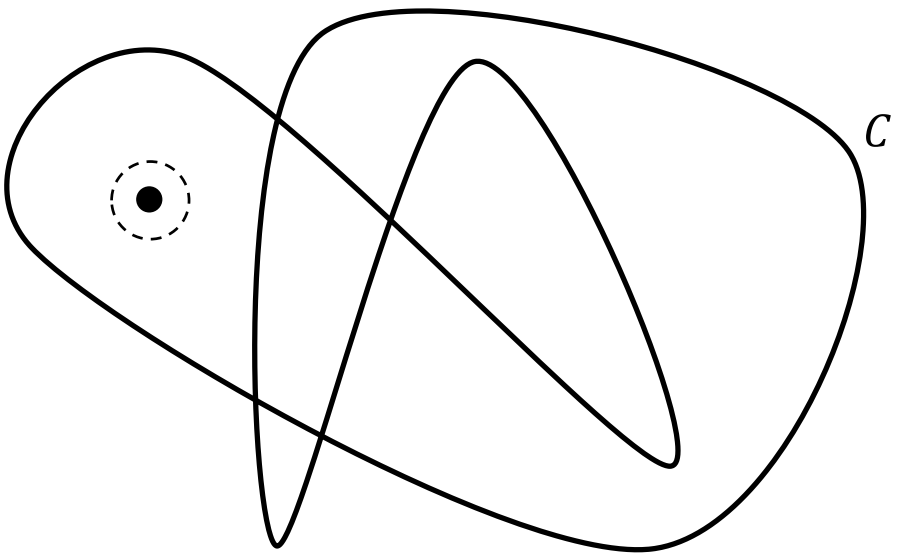


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

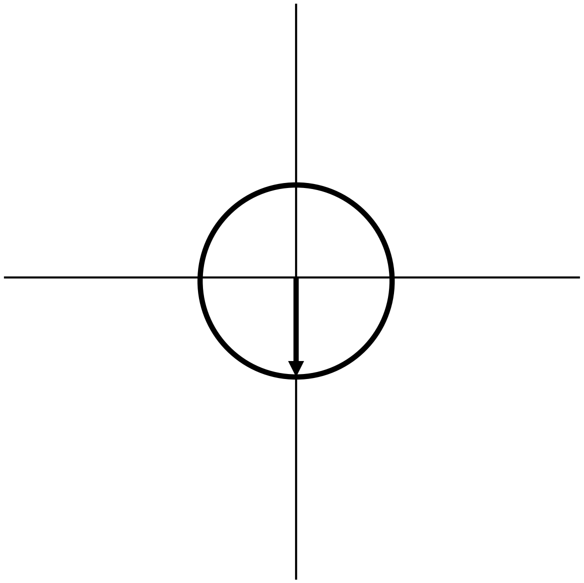


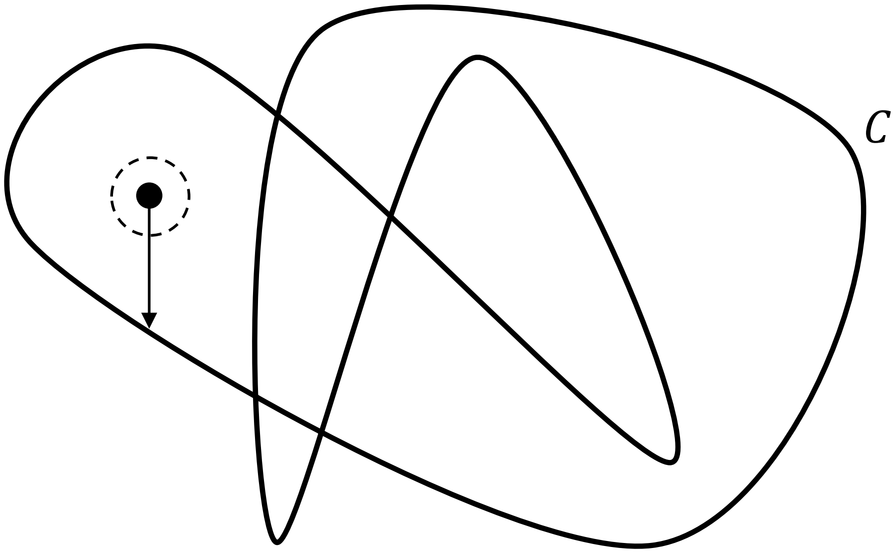


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

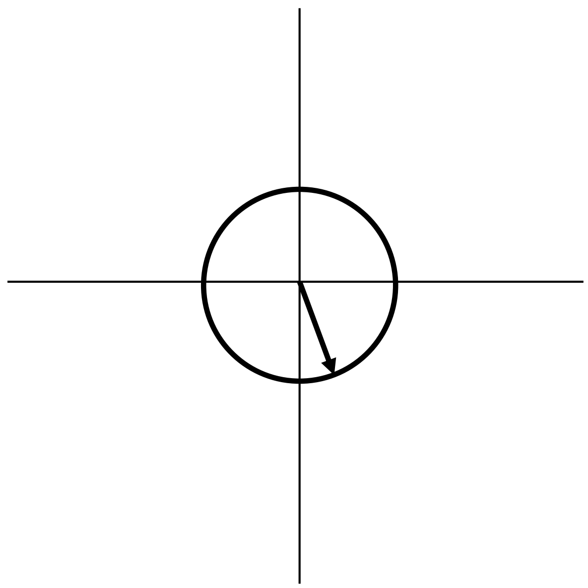


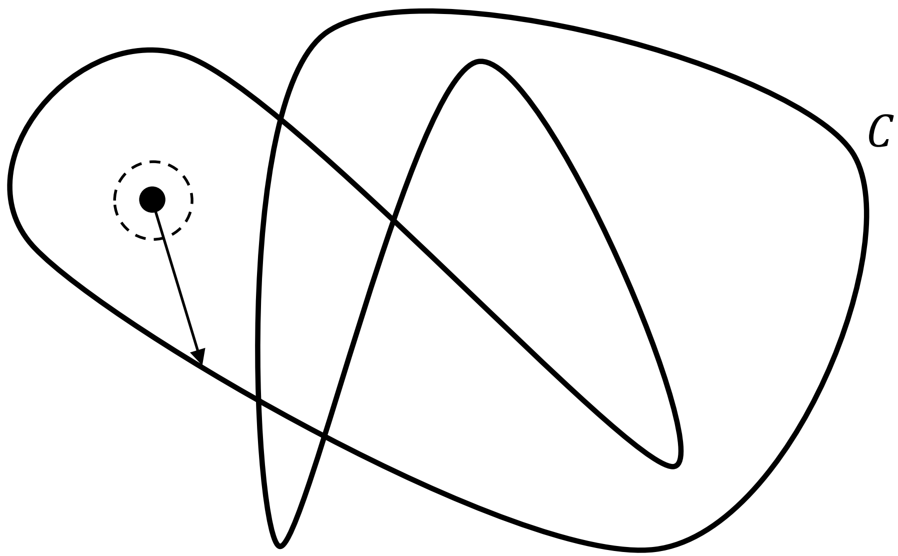


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

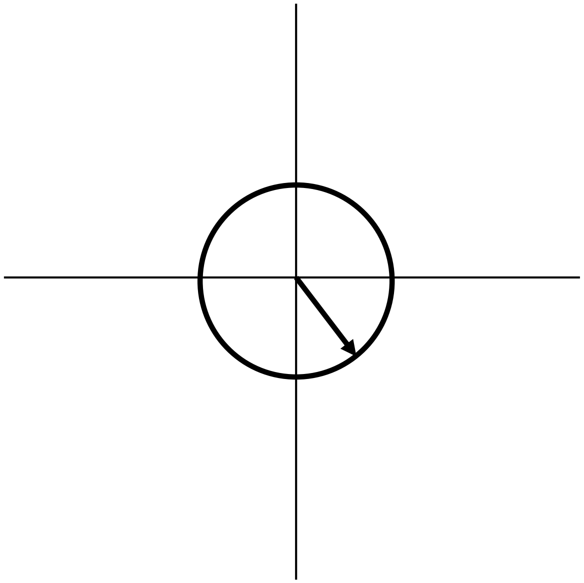


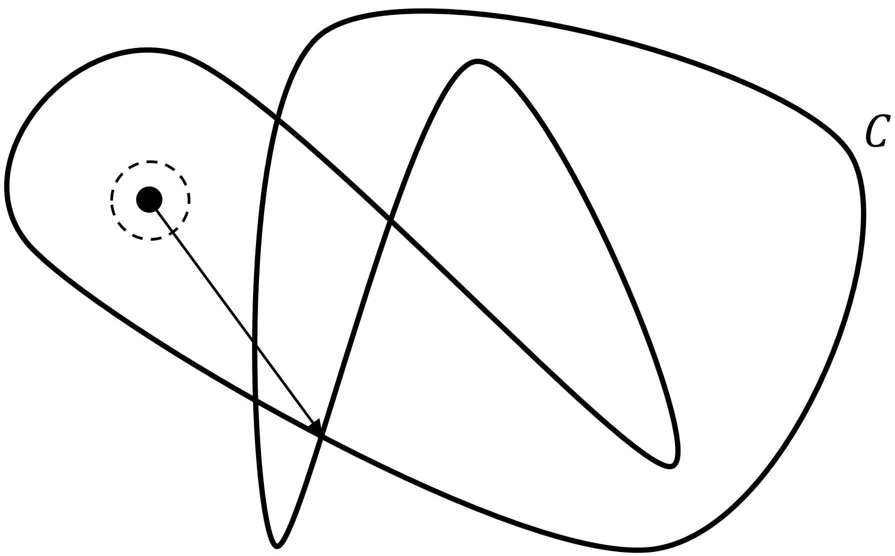


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

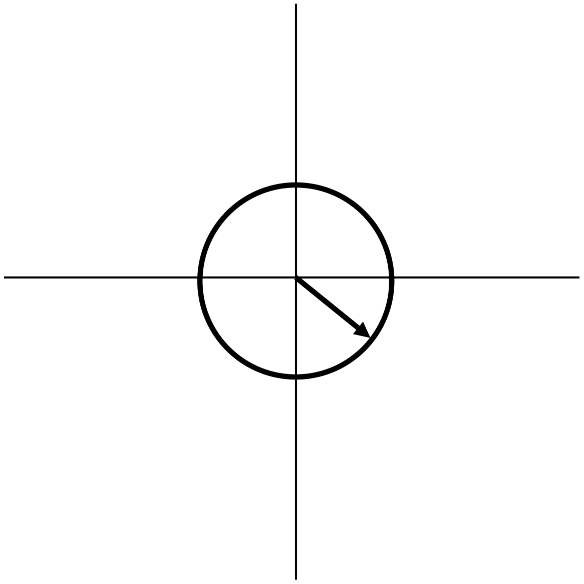


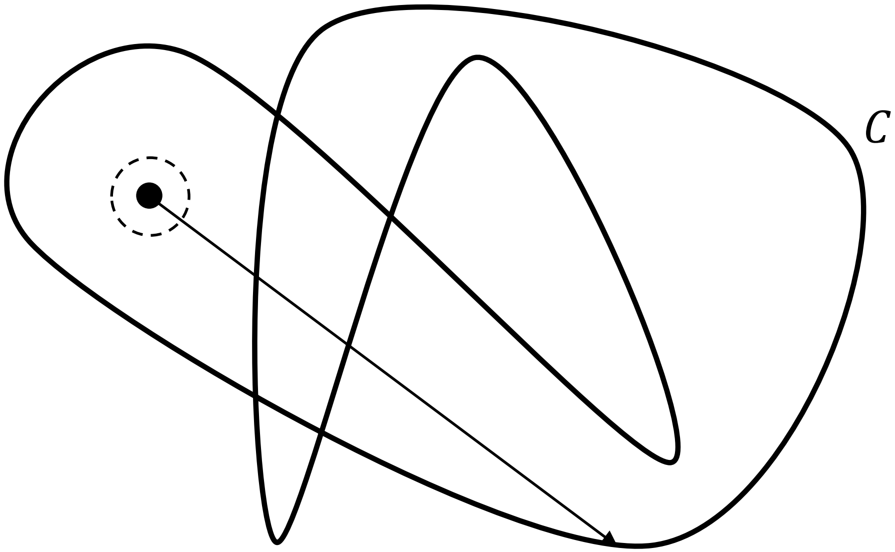


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

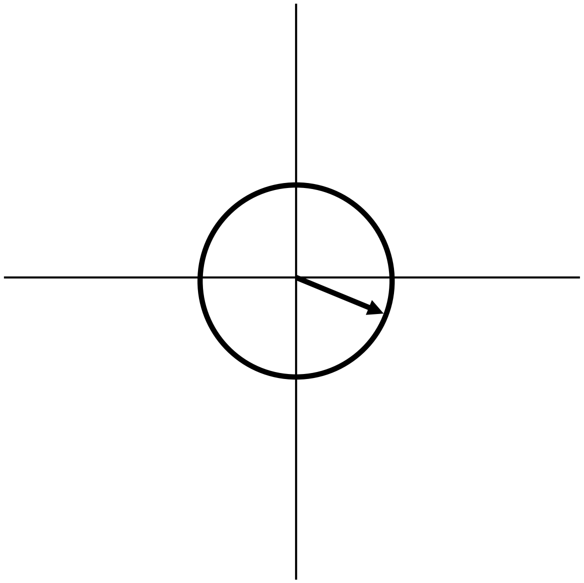


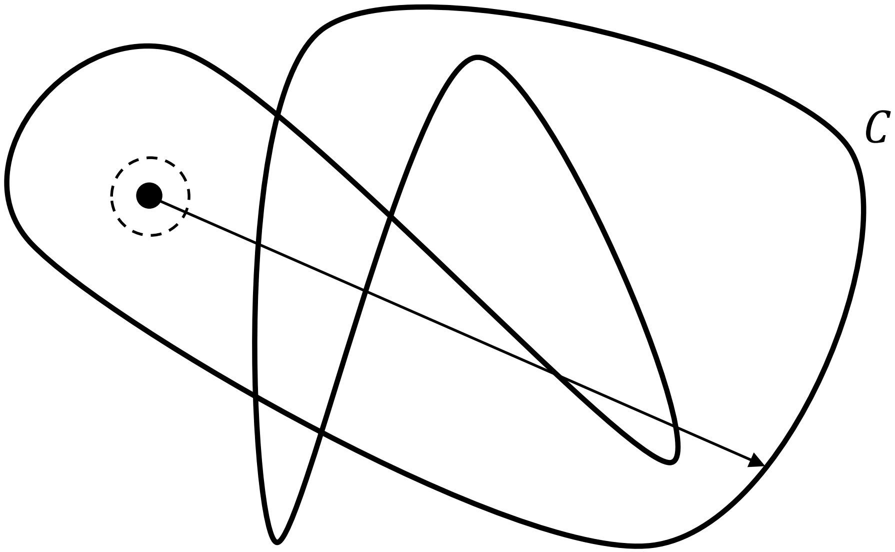


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

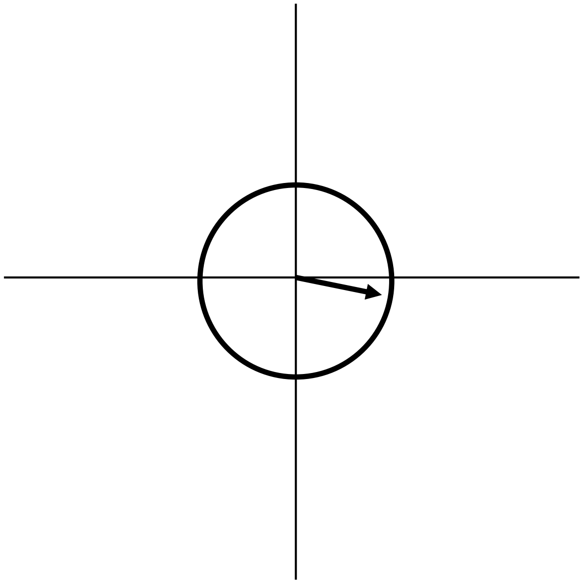


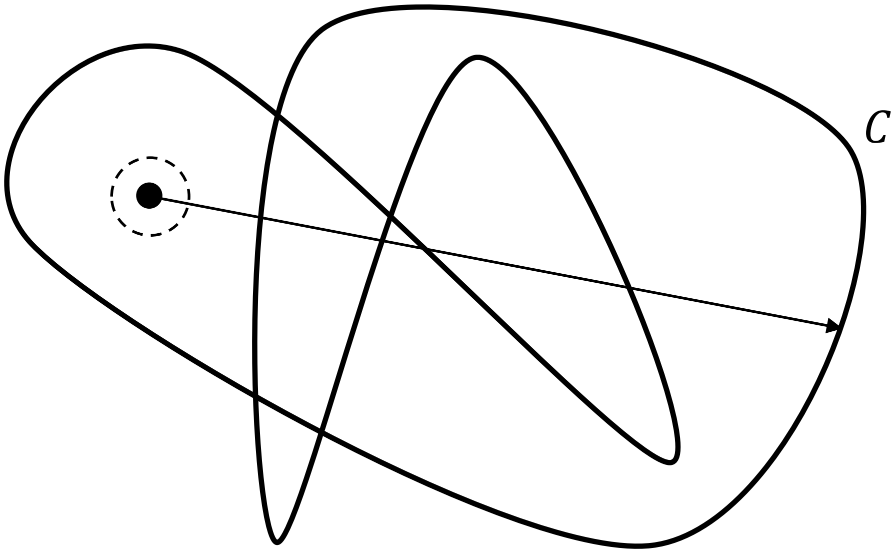


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

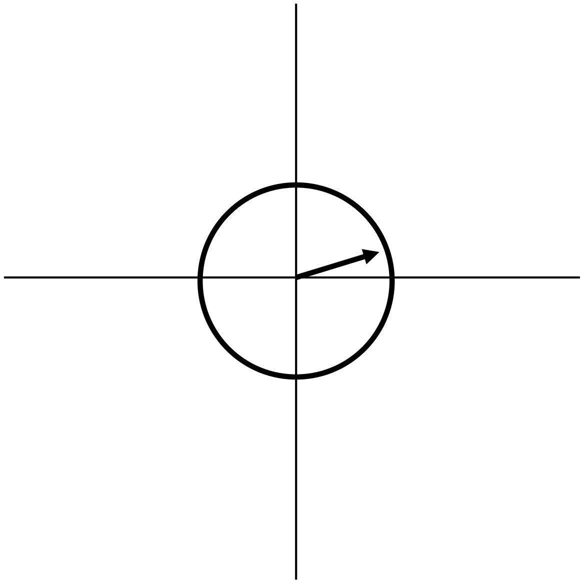


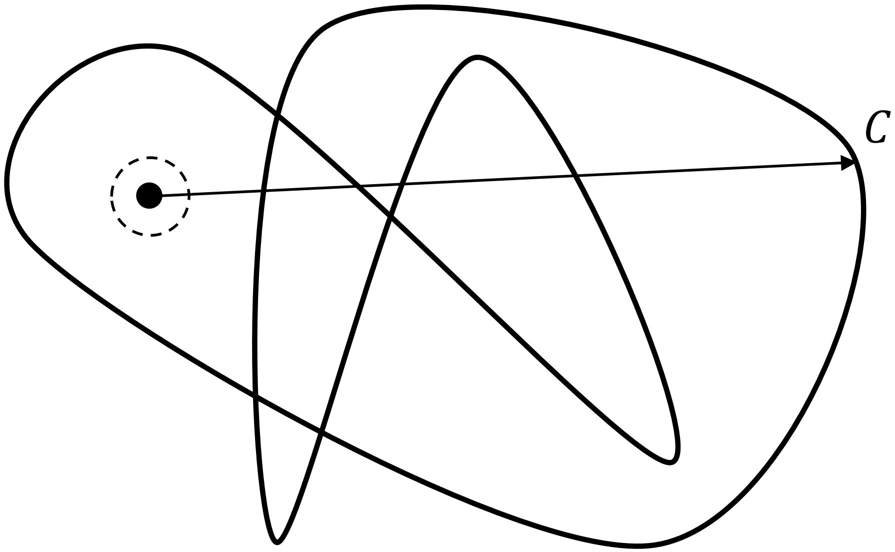


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

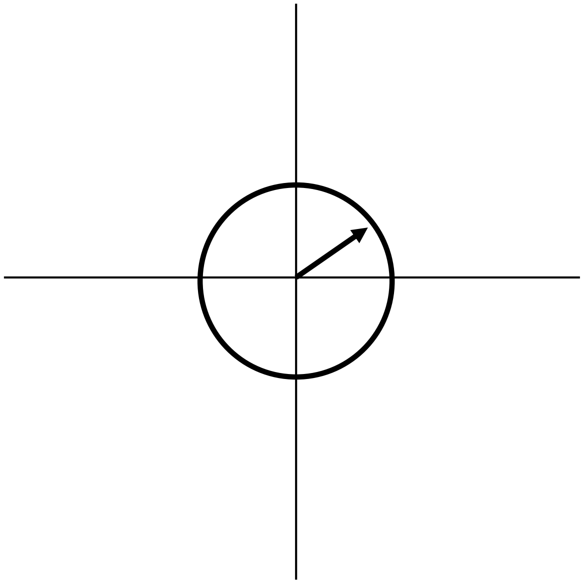


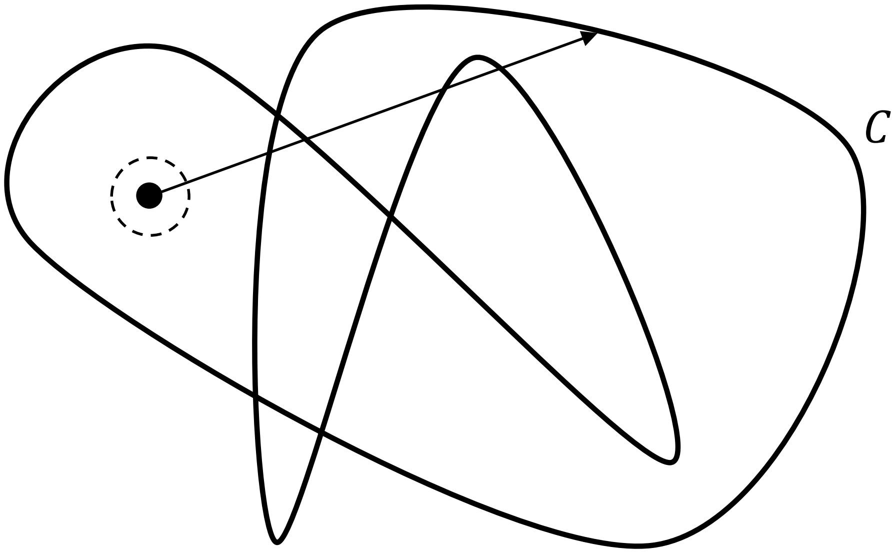


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

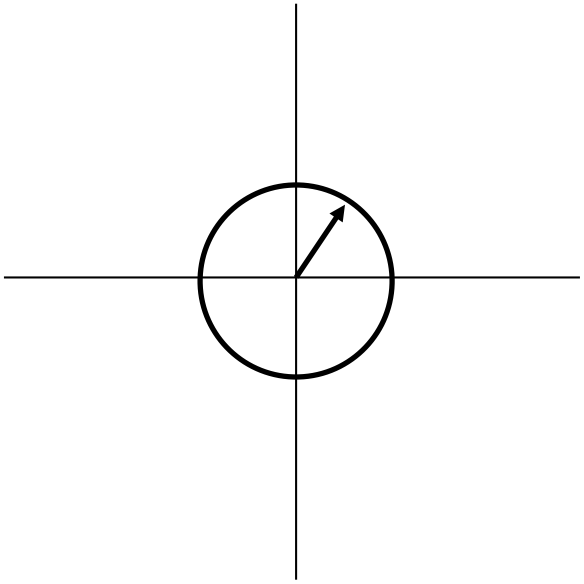


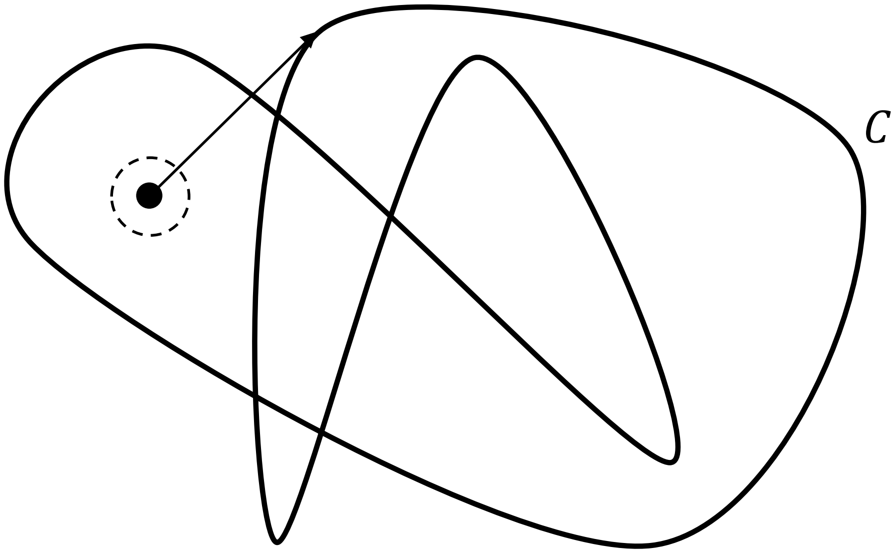


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

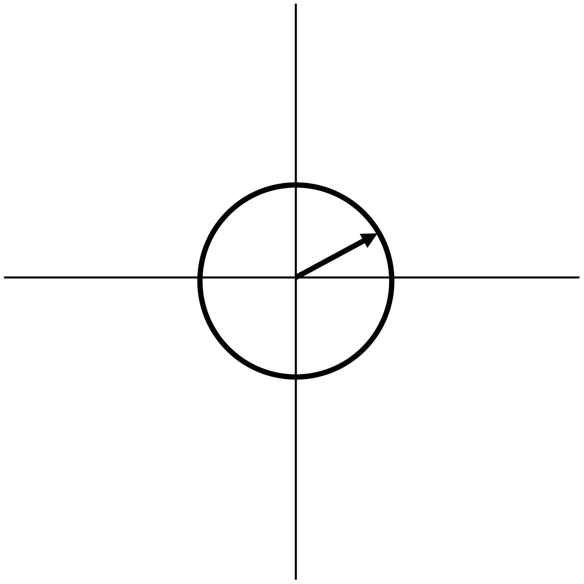


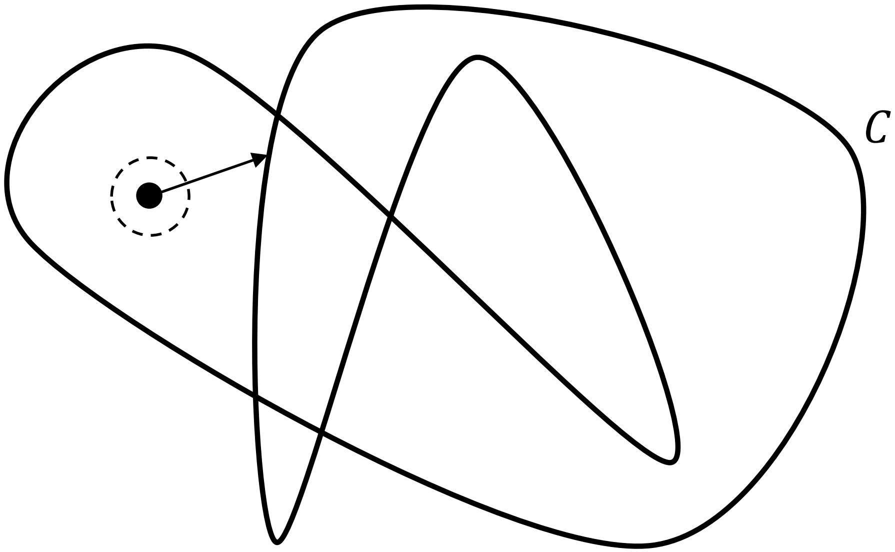


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

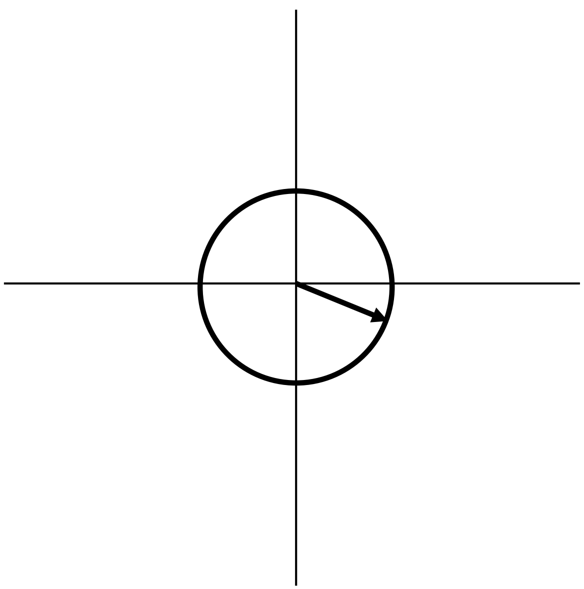


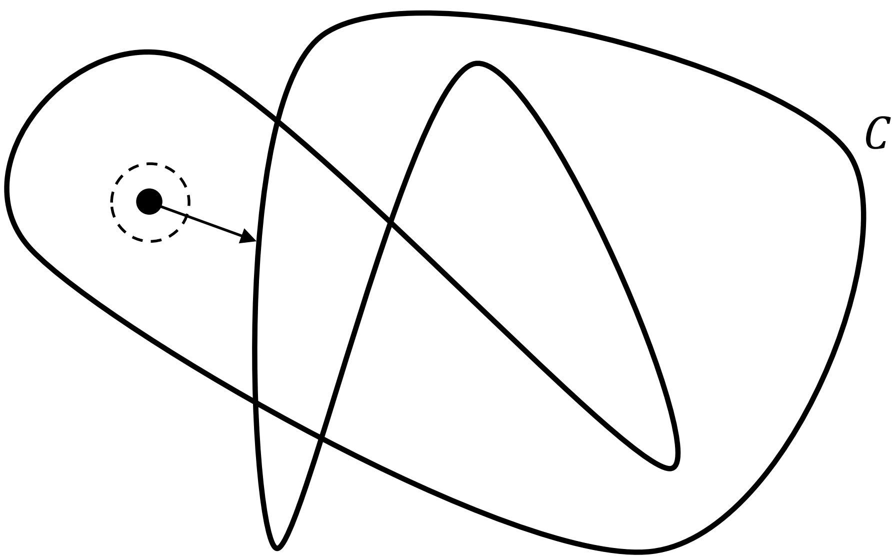


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

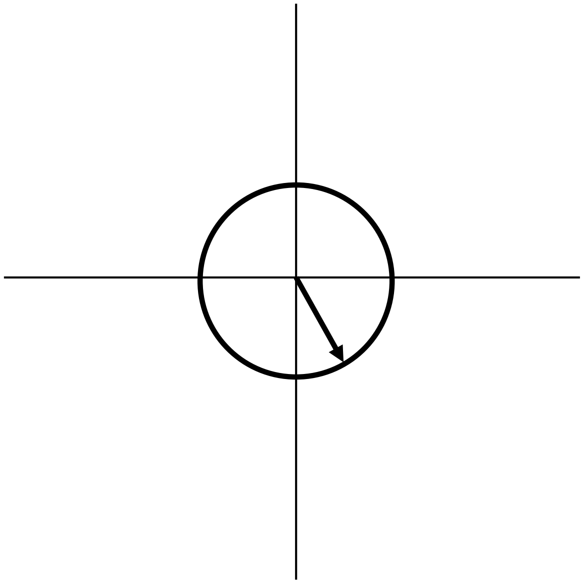


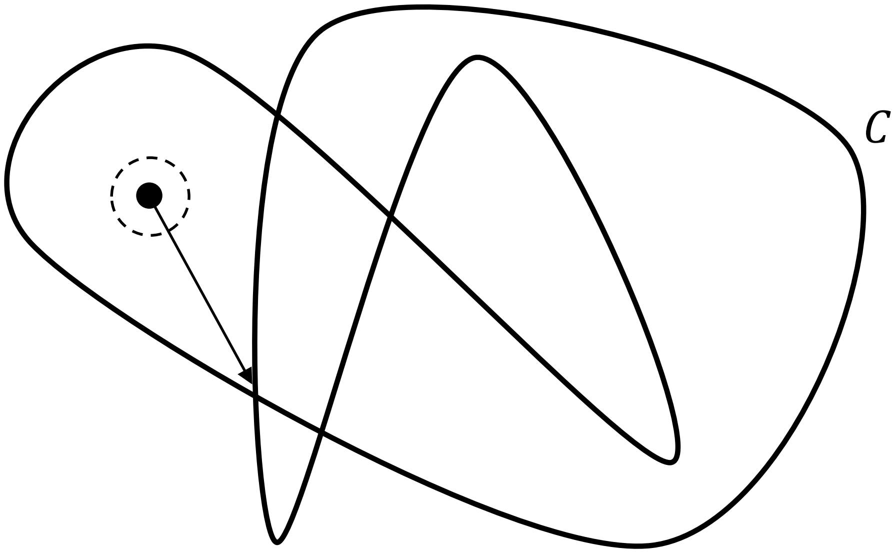


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

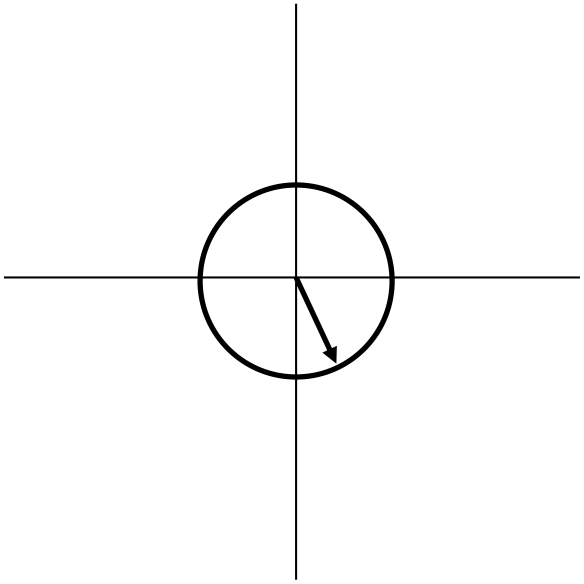


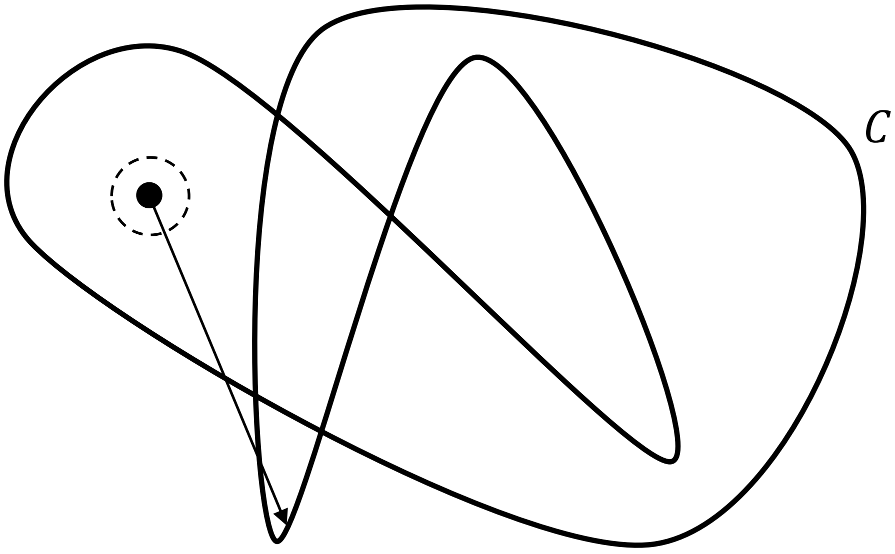


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

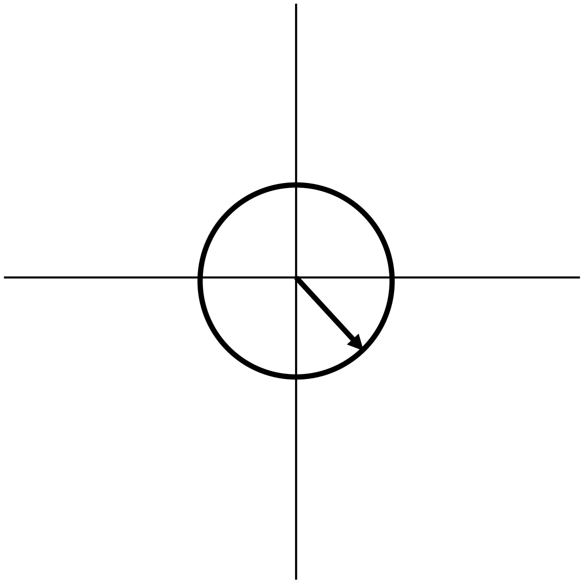


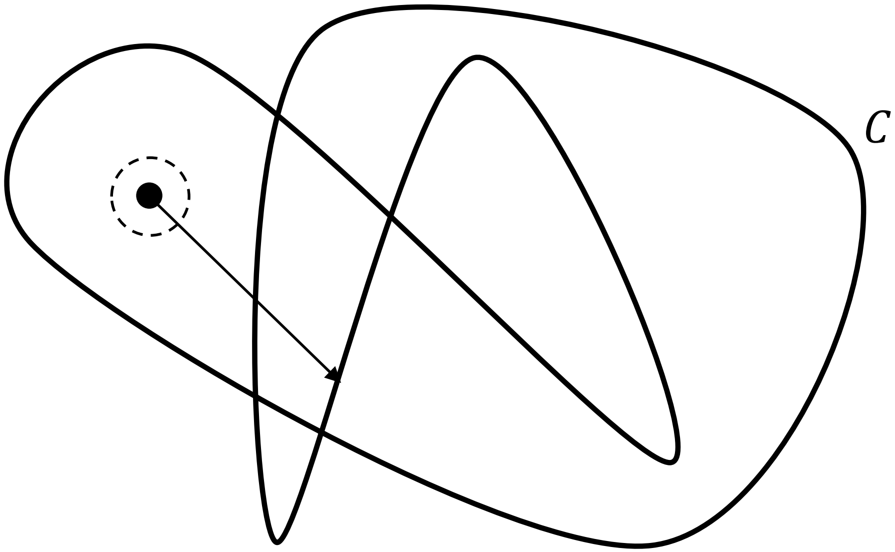


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

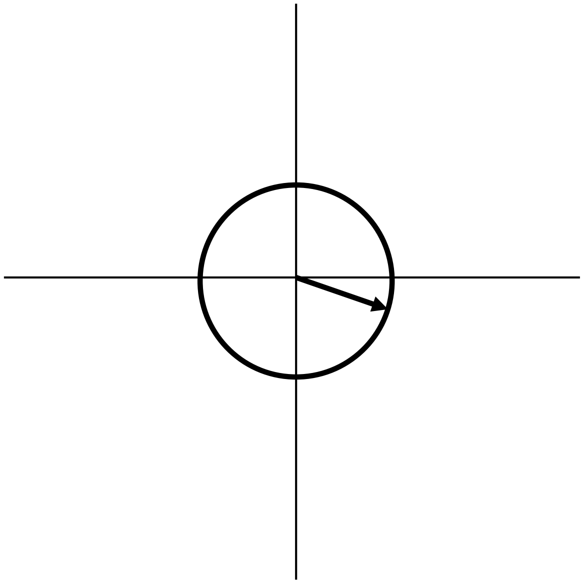


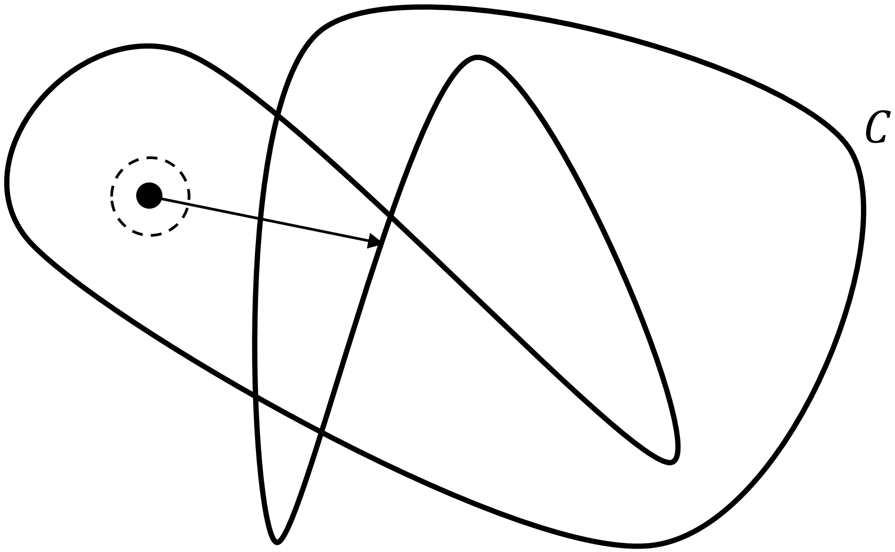


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

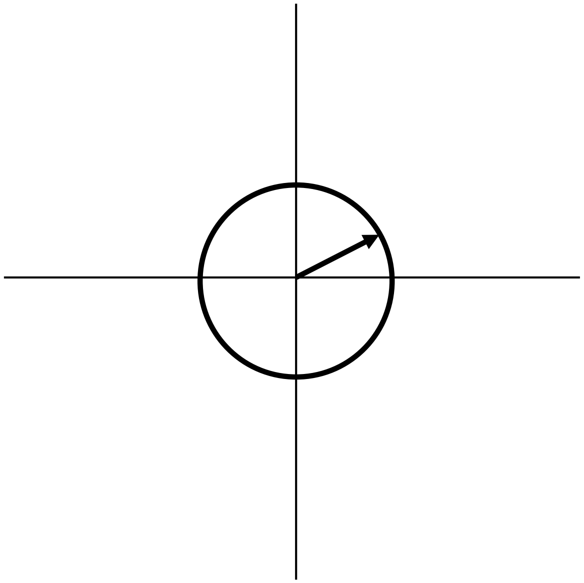


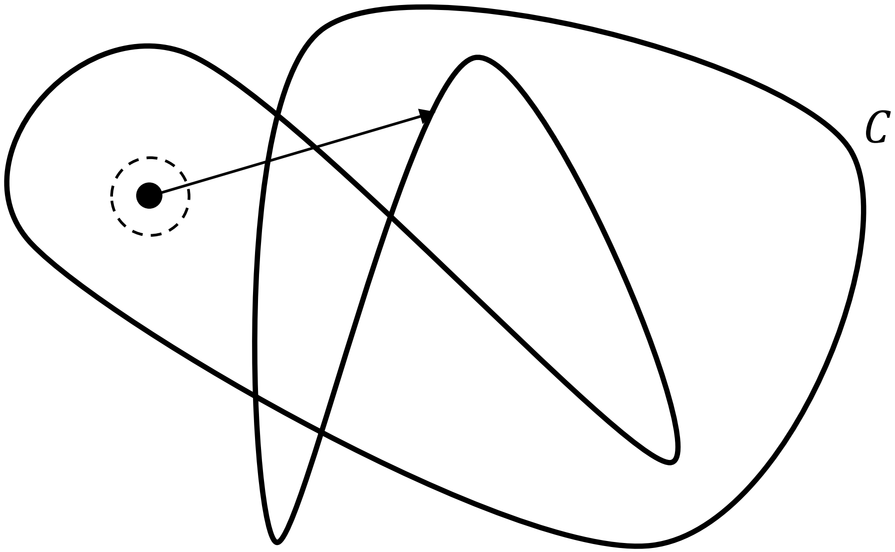


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

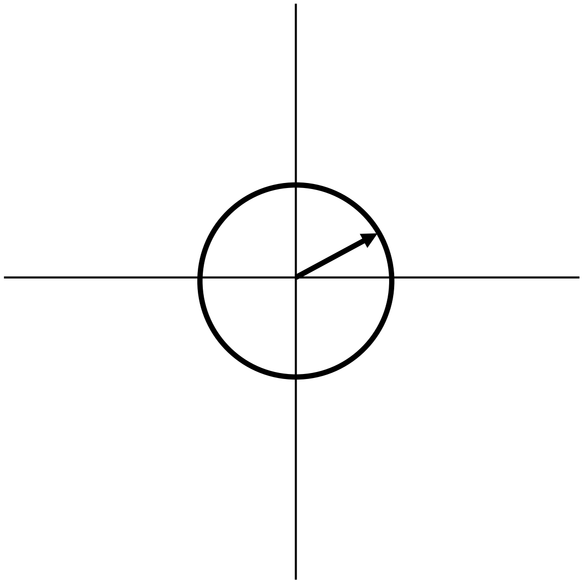


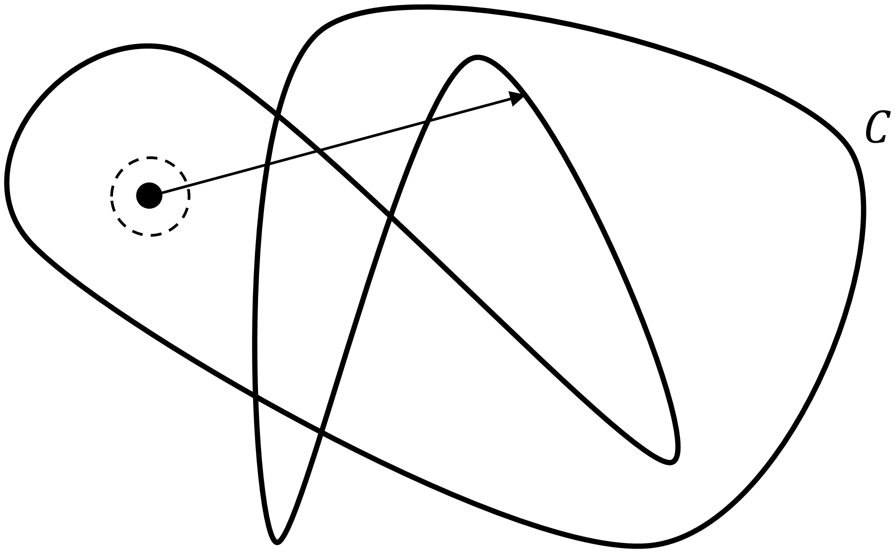


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

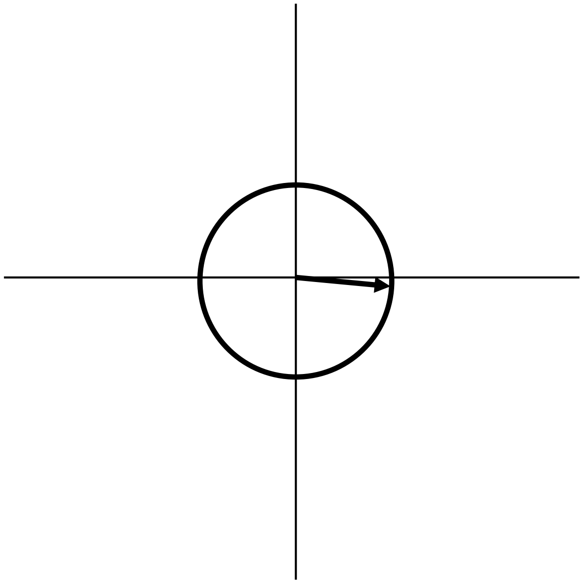


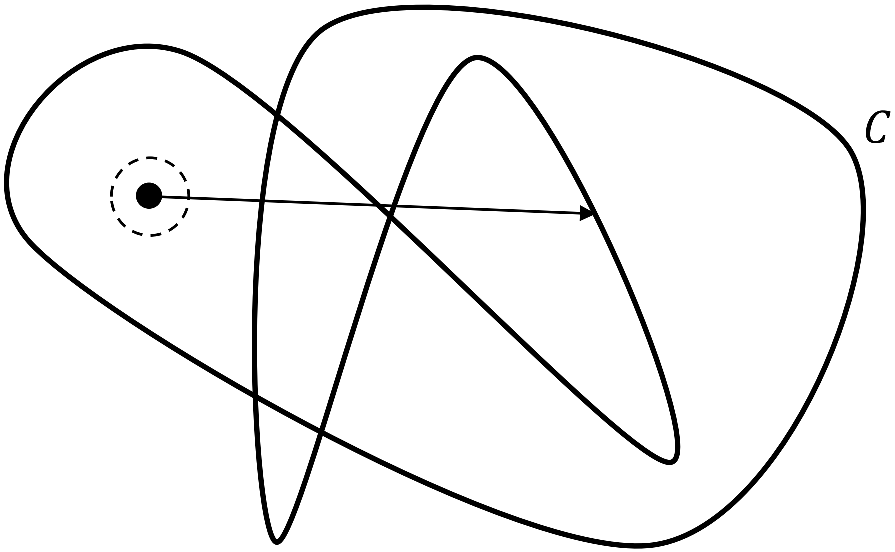


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

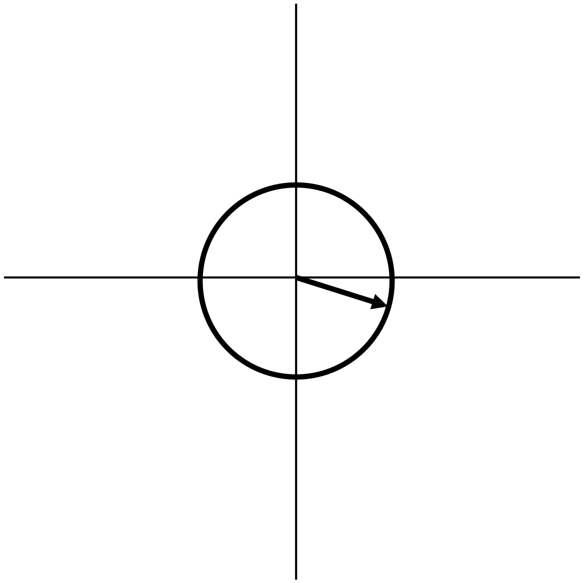


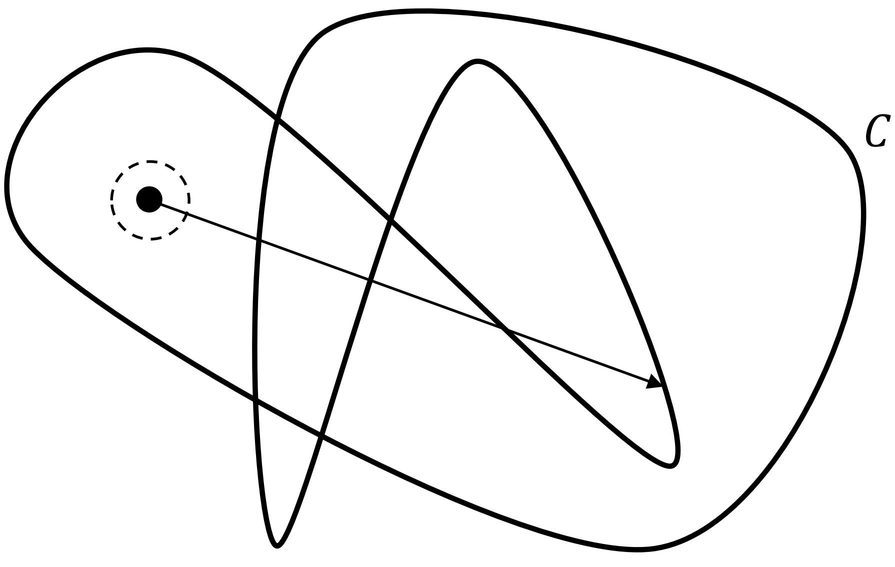


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

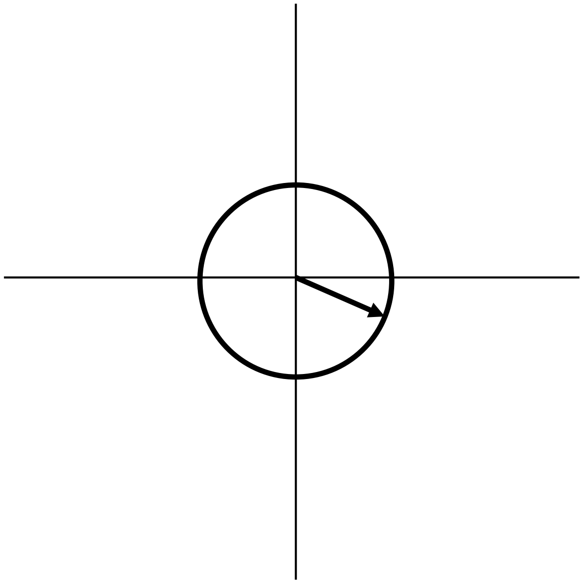


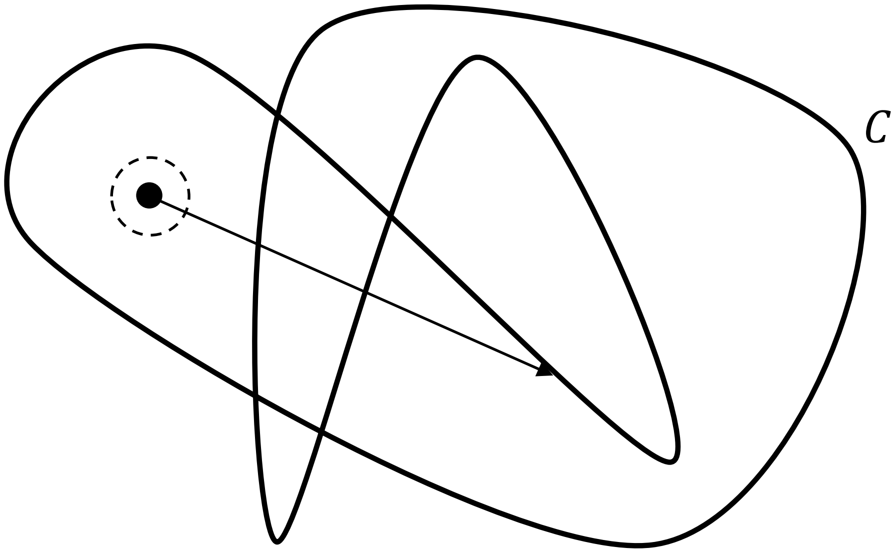


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

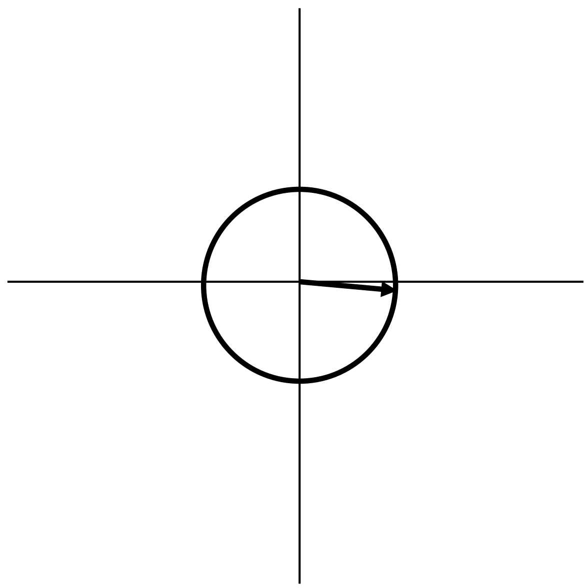


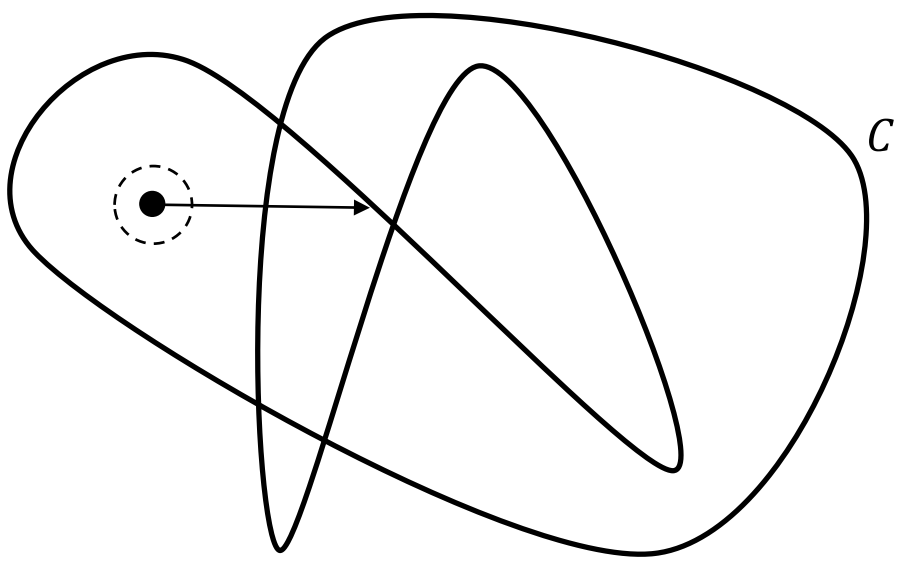


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

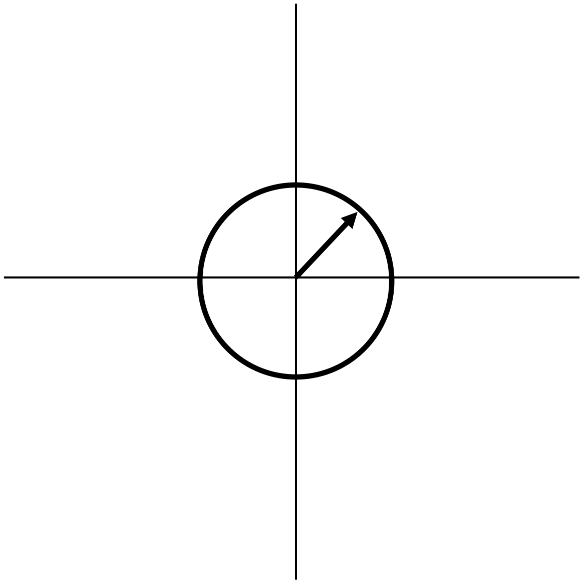


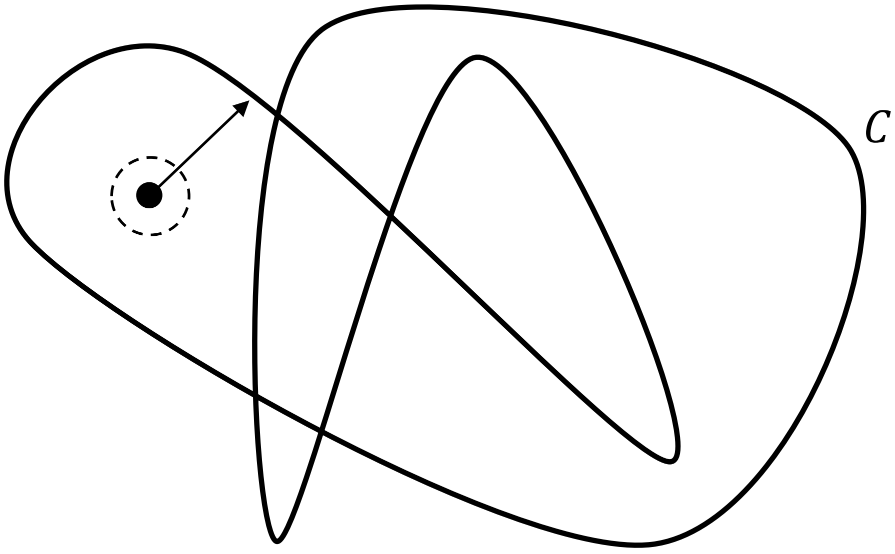


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

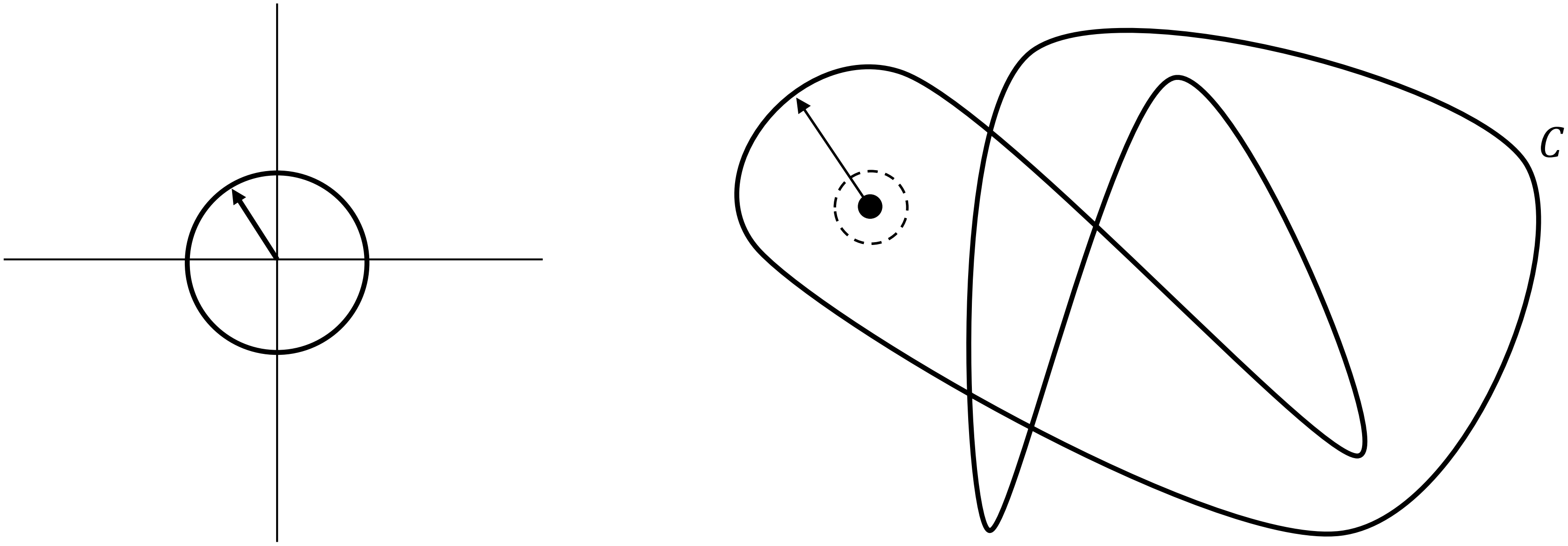




Point-Polygon/Polyhedron

Winding Number:

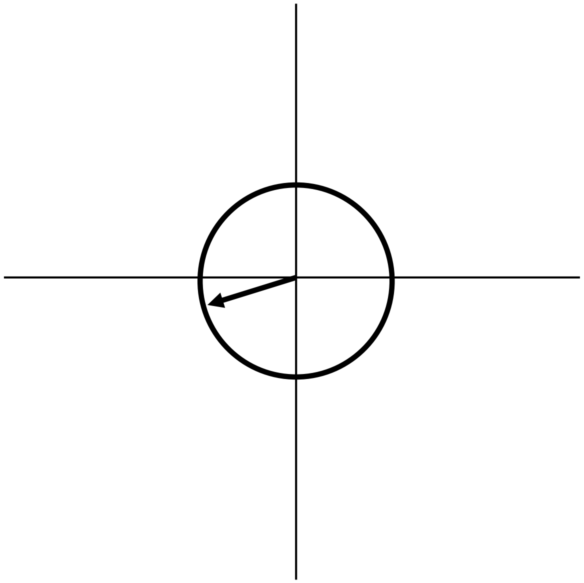
Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

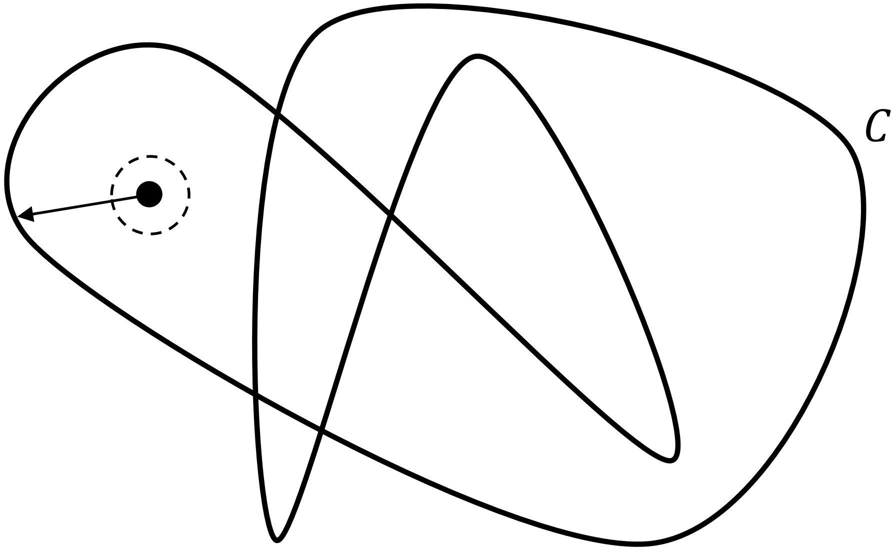


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

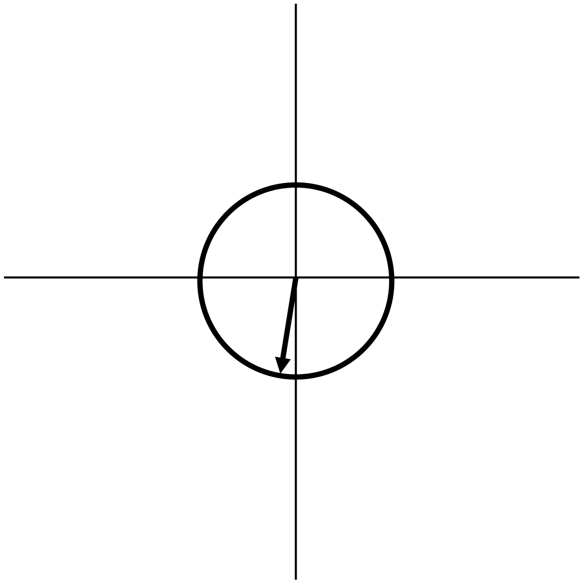


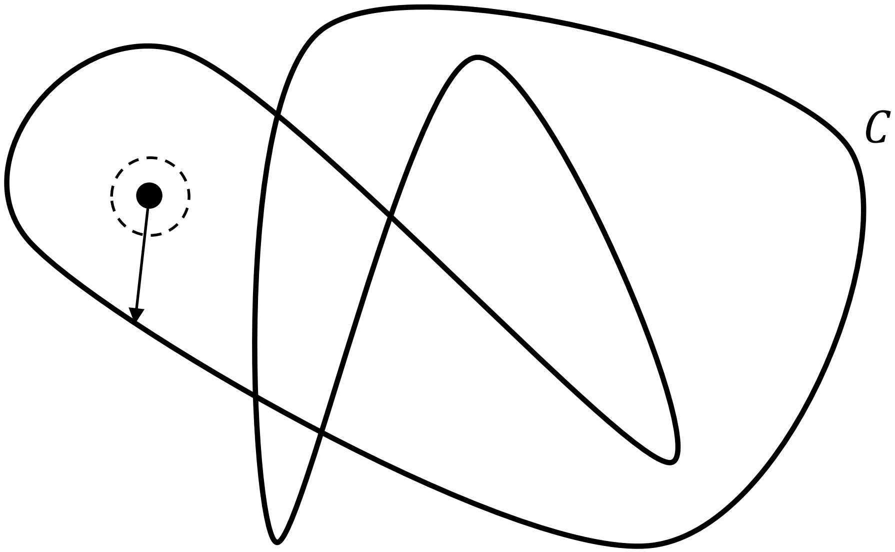


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.



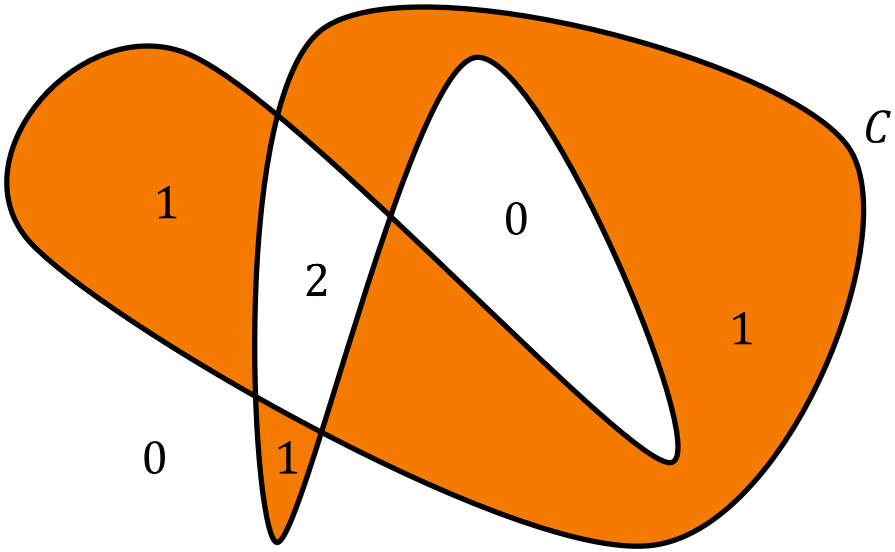


Point-Polygon/Polyhedron

Winding Number:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

A point is interior if its winding number is odd/one.

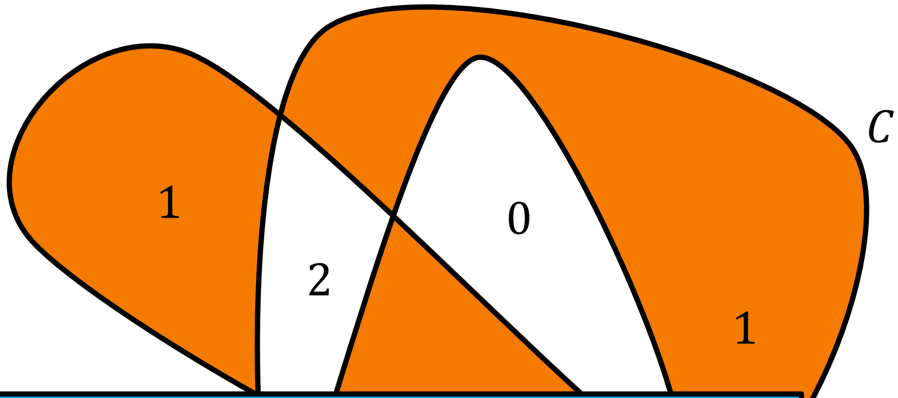


Point-Polygon/Polyhedron

<u>Winding Number</u>:

Given a point p and a curve C in the plane, we can compute the number of times the curve winds around p.

A point is interior if its winding number is odd/one.

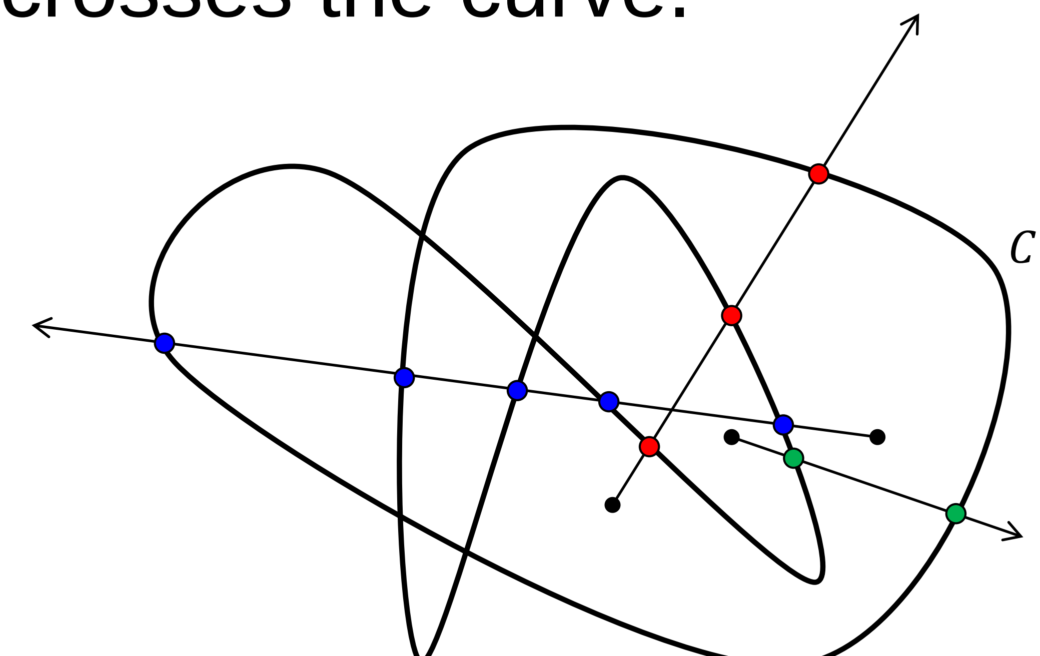


A similar approach (measuring steridians instead of angles) can be used to test for points in polyhedra.

Point-Polygon/Polyhedron

Parity Test:

Given a point p and a curve C in the plane, we can compute the number of times a ray emanating from p crosses the curve.

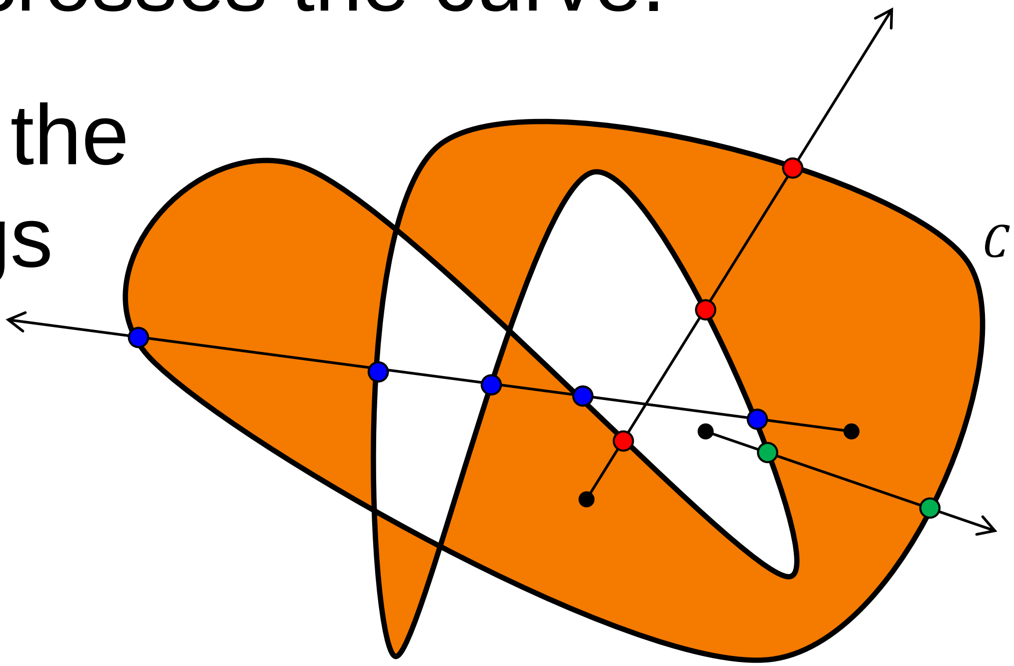


Point-Polygon/Polyhedron

<u>Parity Test</u>:

Given a point p and a curve C in the plane, we can compute the number of times a ray emanating from p crosses the curve.

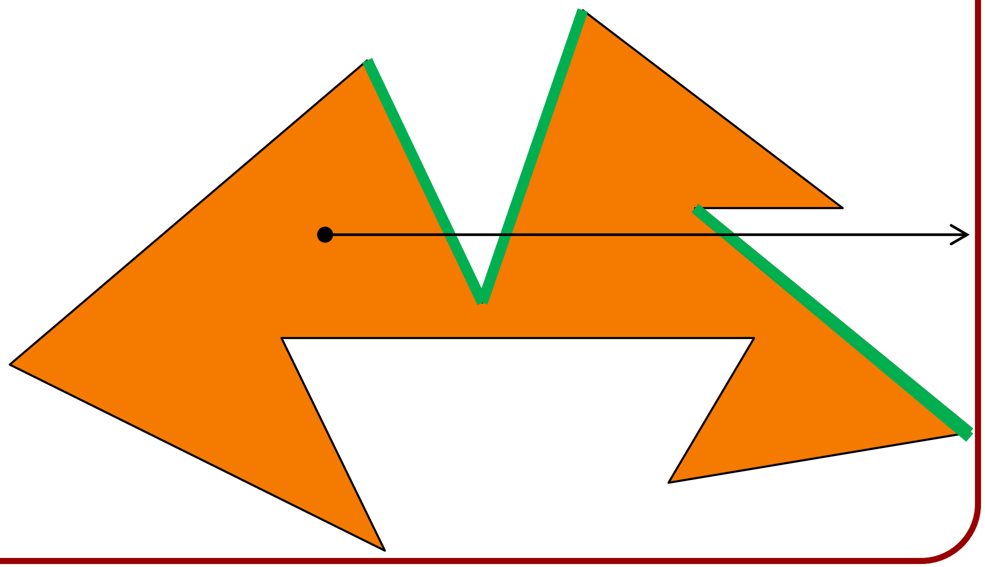
A point is interior if the number of crossings is odd.



Point-Polygon/Polyhedron

Parity Test:

When the curve is a polygon, we can test using, e.g. a ray directed along the positive x-axis, and test for intersection with edges.

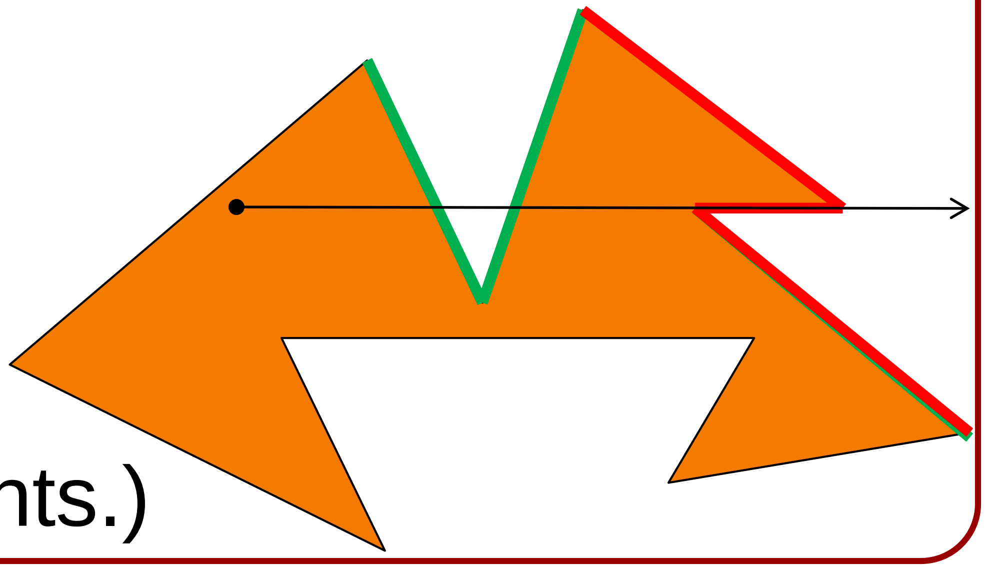


Point-Polygon/Polyhedron

Parity Test:

When the curve is a polygon, we can test using, e.g. a ray directed along the positive x-axis, and test for intersection with edges.

What happens if the intersection is degenerate?
(We cannot use the parity of the count of connected components.)

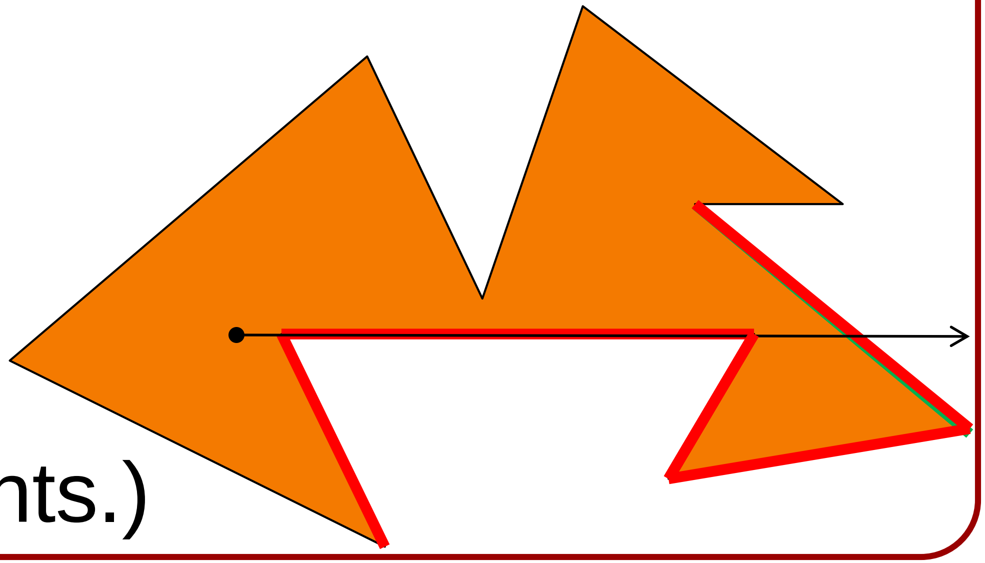


Point-Polygon/Polyhedron

Parity Test:

When the curve is a polygon, we can test using, e.g. a ray directed along the positive x-axis, and test for intersection with edges.

What happens if the intersection is degenerate?
(We cannot use the parity of the count of connected components.)

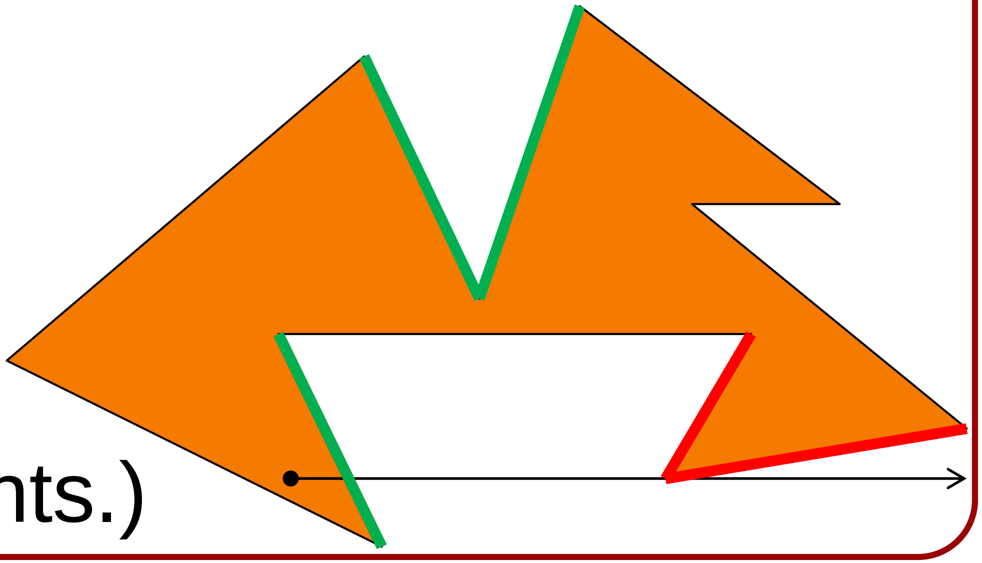


Point-Polygon/Polyhedron

Parity Test:

When the curve is a polygon, we can test using, e.g. a ray directed along the positive x-axis, and test for intersection with edges.

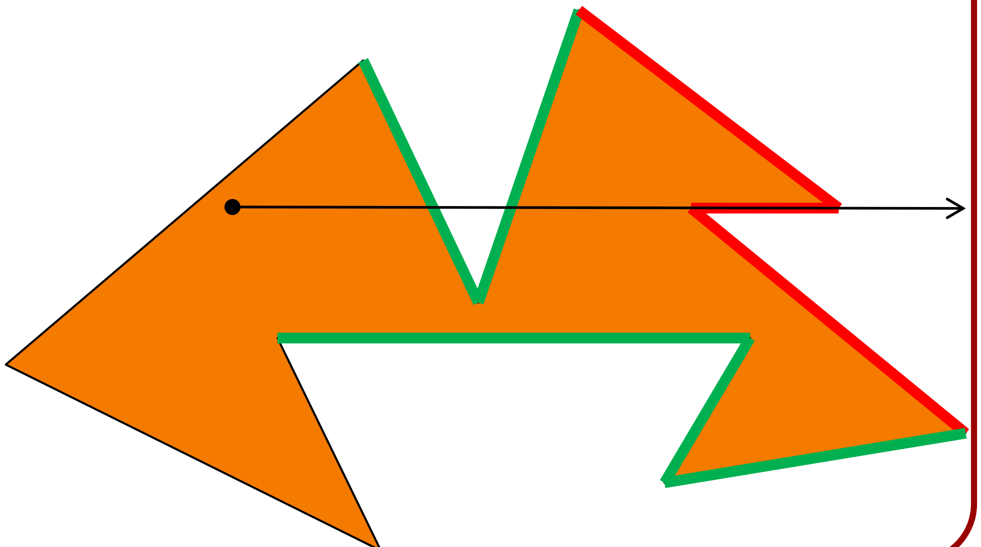
What happens if the intersection is degenerate?
(We cannot use the parity of the count of connected components.)



Point-Polygon/Polyhedron

Parity Test (Avoid Degeneracies):

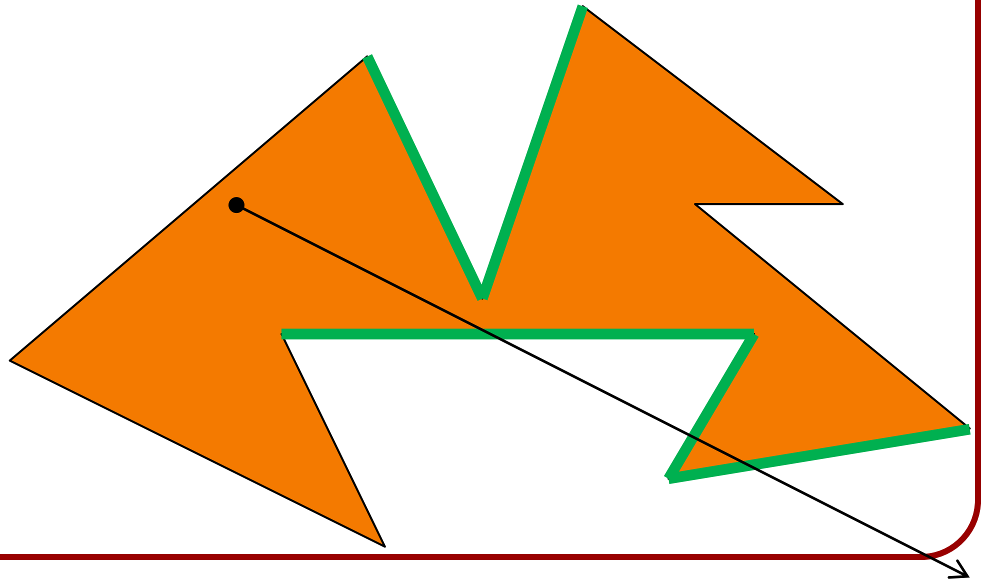
- Test for degeneracies, and if encountered, cast a different ray in some other (random) direction.
- With high likelihood, that ray won't be degenerate.
- Otherwise, cast again.



Point-Polygon/Polyhedron

Parity Test (Avoid Degeneracies):

- Test for degeneracies, and if encountered, cast a different ray in some other (random) direction.
- With high likelihood, that ray won't be degenerate.
- Otherwise, cast again.

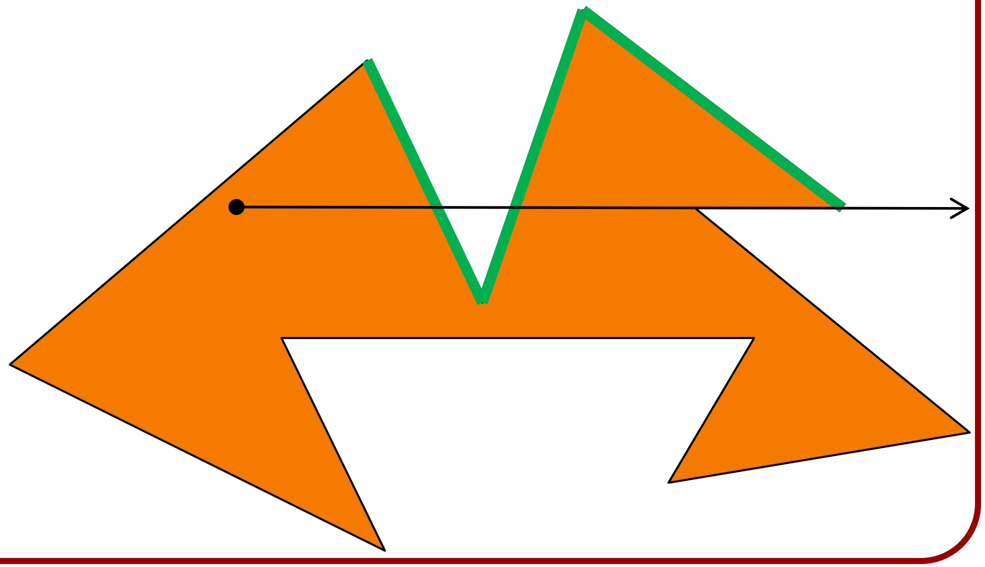


Point-Polygon/Polyhedron

Parity Test (Handle Degeneracies):

- Define a ray-edge intersection if the ray intersects and one of the end-points is above.

(Equivalent to shifting the ray up by a tiny amount.)

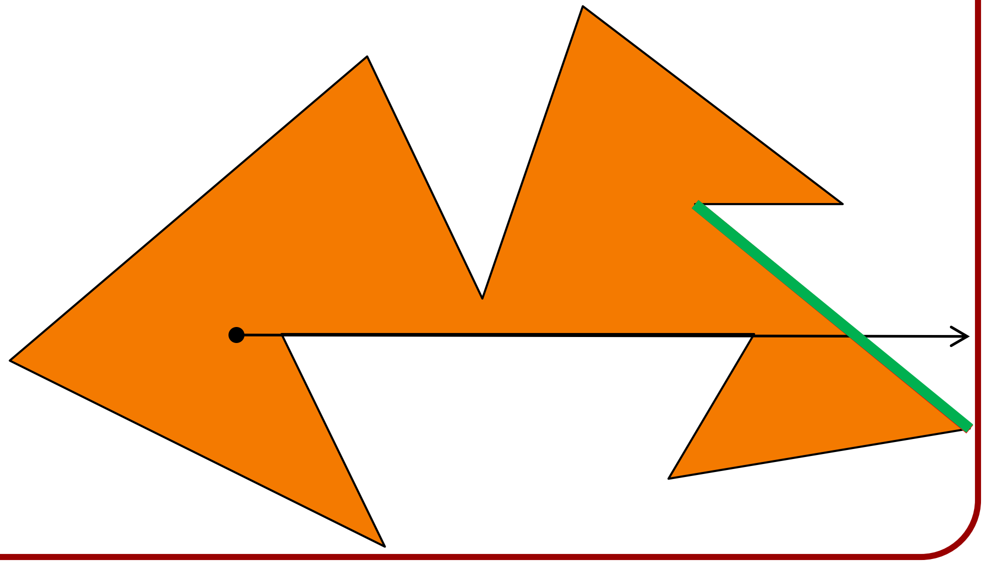


Point-Polygon/Polyhedron

Parity Test (Handle Degeneracies):

- Define a ray-edge intersection if the ray intersects and one of the end-points is above.

(Equivalent to shifting the ray up by a tiny amount.)

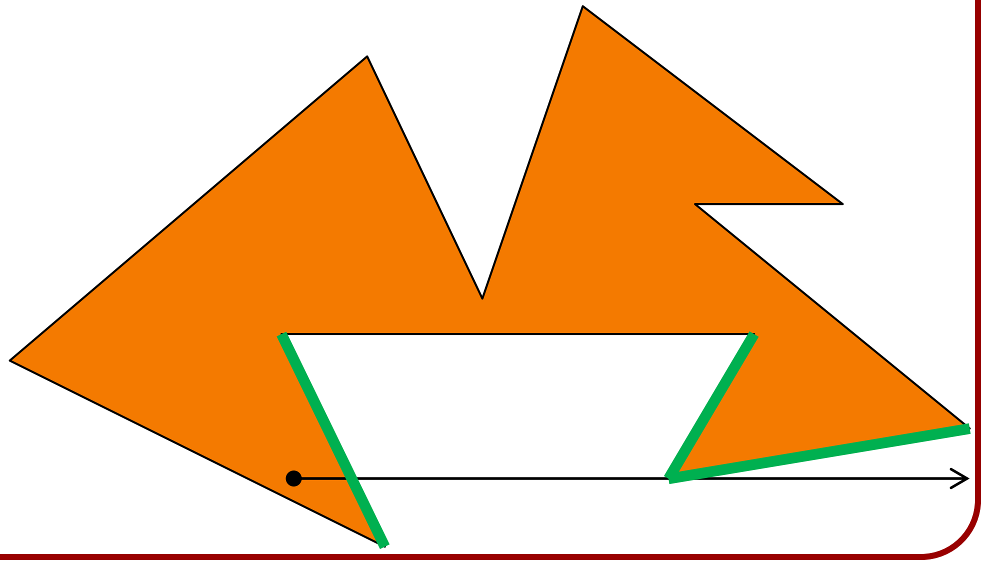


Point-Polygon/Polyhedron

<u>Parity Test (Handle Degeneracies)</u>:

- Define a ray-edge intersection if the ray intersects and one of the end-points is above.

(Equivalent to shifting the ray up by a tiny amount.)



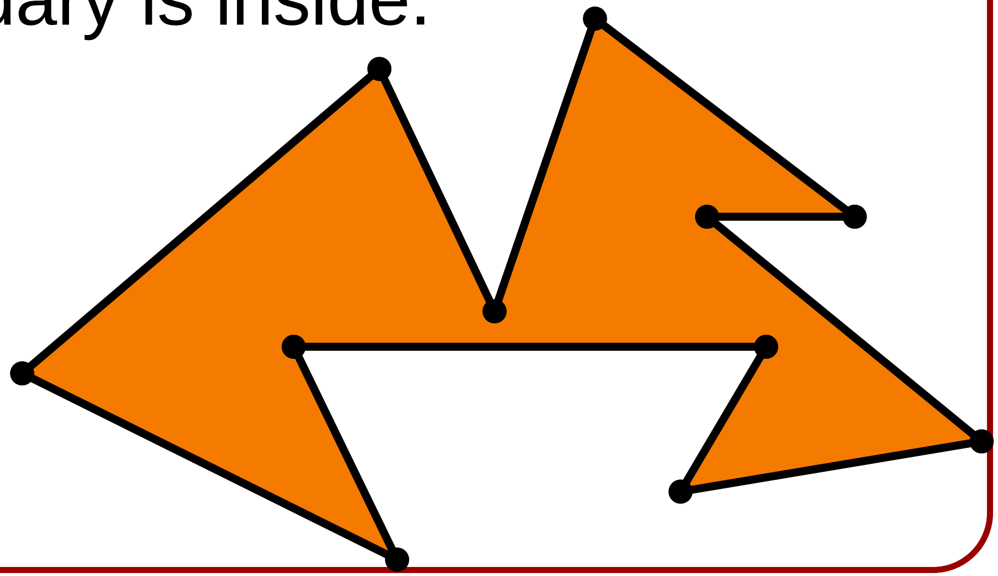
Point-Polygon/Polyhedron

Parity Test (Handle Degeneracies):

- Define a ray-edge intersection if the ray intersects and one of the end-points is above.

Boundary (Closed Polygons):

- A point on the boundary is inside.



Point-Polygon/Polyhedron

Parity Test (Handle Degeneracies):

- Define a ray-edge intersection if the ray intersects and one of the end-points is above.

Boundary (Partitioning Polygons):

- A point on an edge is inside if the points to the right are.

(Equivalent to shifting the ray up and right by a tiny bit.)