Mathematics in Computer Graphics

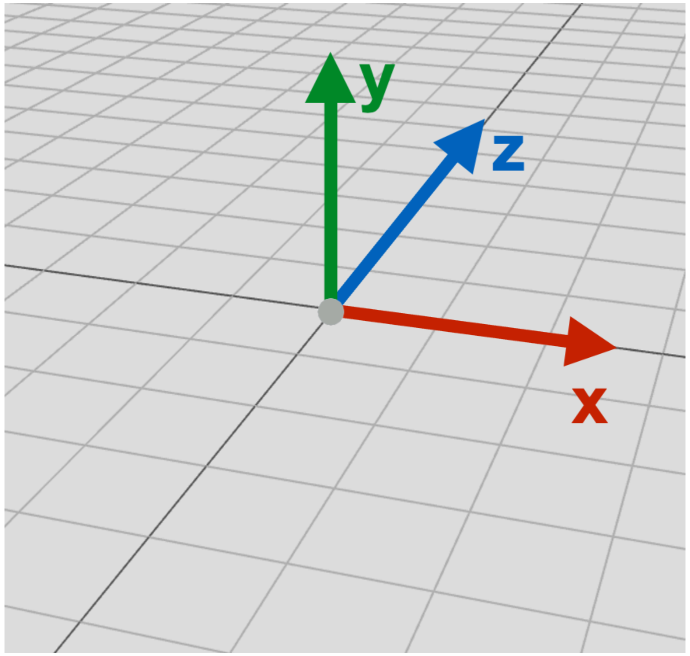
Contents

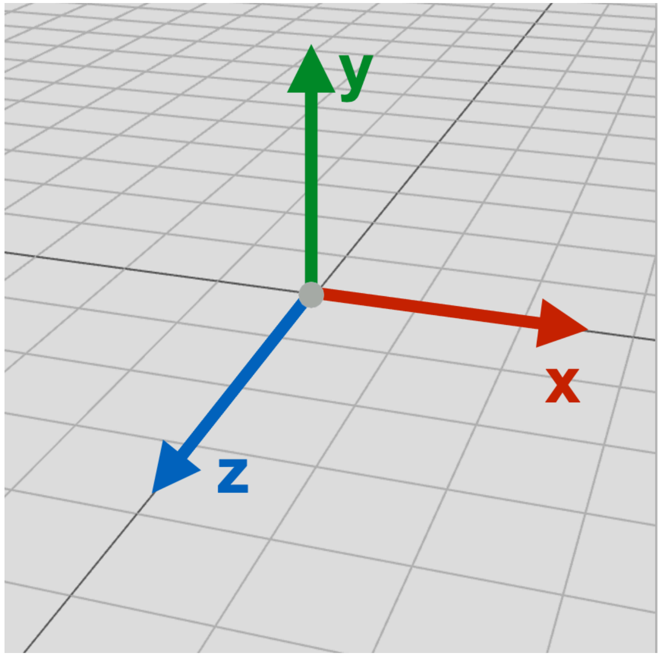
- Geometry (2D, 3D)
- Trigonometry
- Vector spaces
  - Points, vectors, and coordinates
- Dot and cross products
- Linear transforms and matrices

Coordinate Systems

- Orthogonal coordinate systems
  - Left-hand
  - Right-hand
- Meaning of axes, simple computation

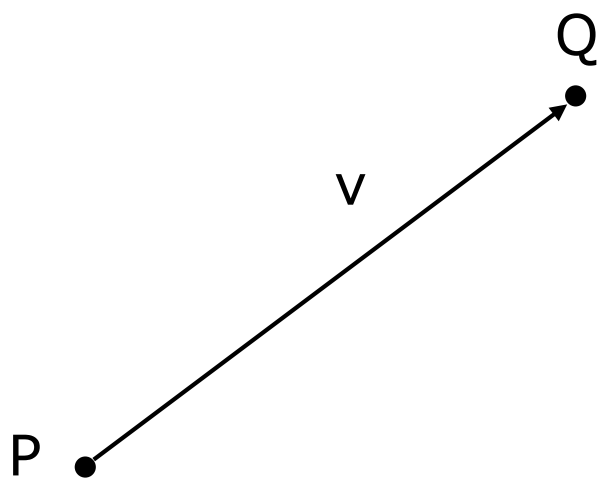
  watching direction, viewer's position, vertical direction





Points

- 2D: P(x, y), 3D: P(x, y, z)
- Point-point subtraction: Q - P = v
  - Result is a vector pointing from P to Q
- Vector-point addition: P + v = Q
  - Result is a new point



Line

- Slope based equation

  $y = mx + b$

  Slope $m= (y - y1) / (x - x1) = (y2 - y1) / (x2 - x1)$

- Parametric equation

  $P = P1 + t(P2-P1)$, as vector based definition

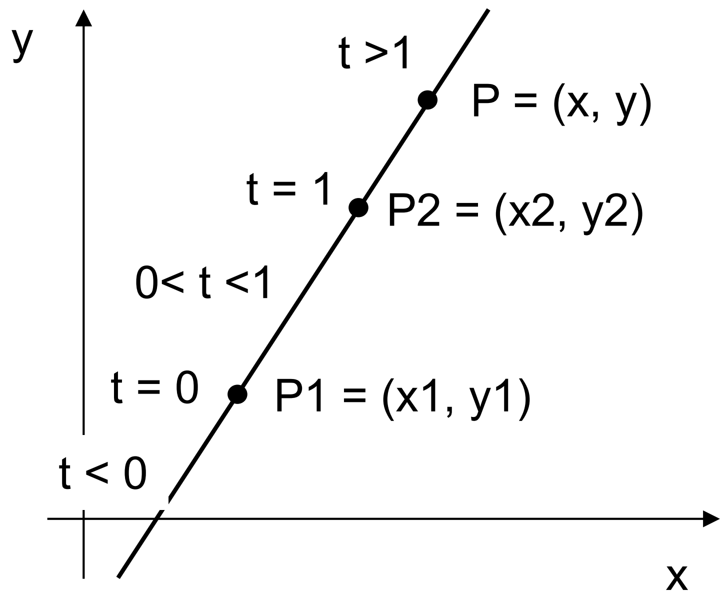
  or

  $P = (1-t)P1 + t\ P2$, as affine combination of two points

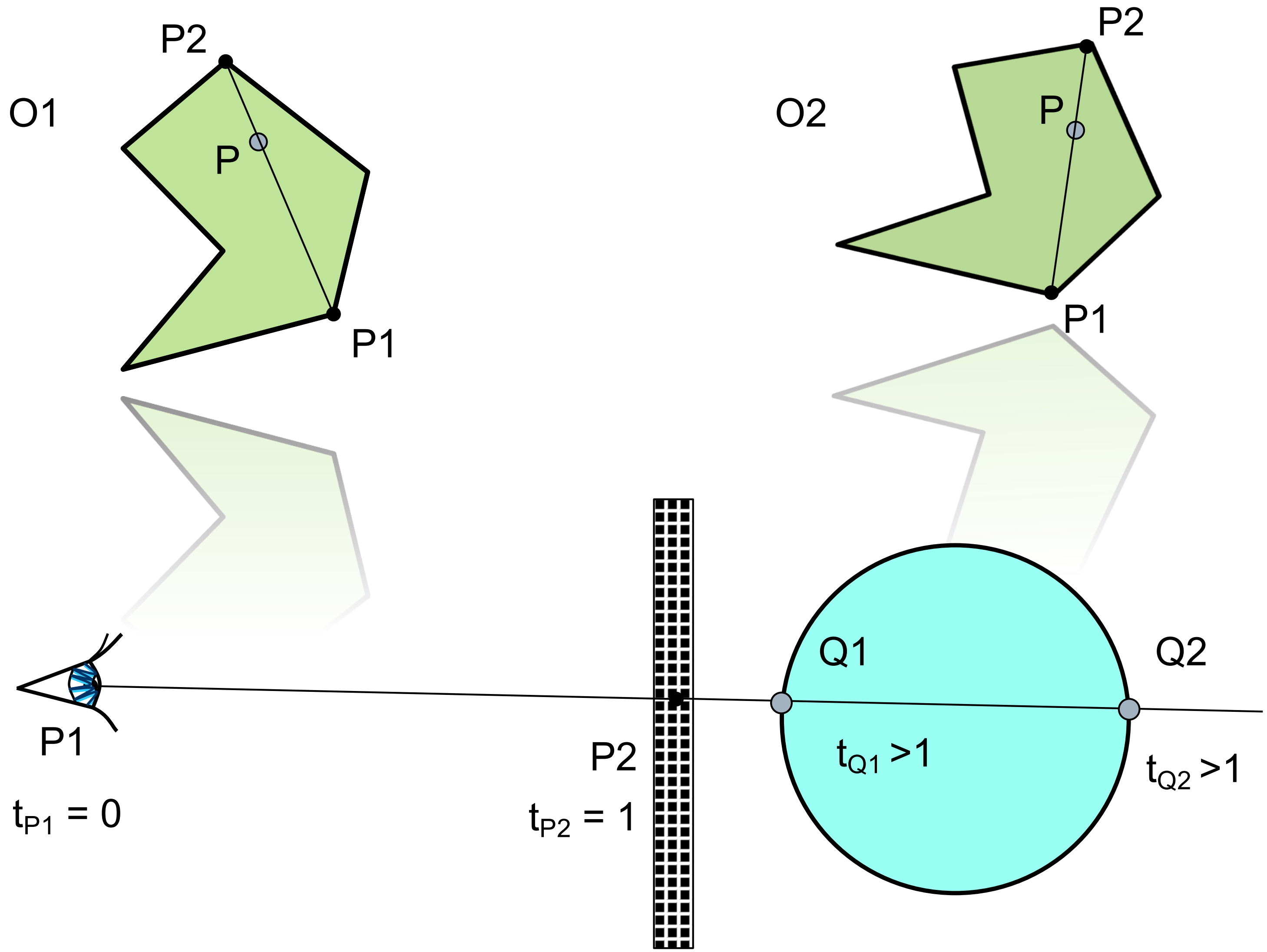
  $x = x1 + t(x2-x1)$

  $y = y1 + t(y2-y1)$

$t<0$,	before P1(x1, y1)
$t=0$,	P=P1
$0<t<1$,	get points between P1 and P2
$t=1$,	P=P2(x2, y2)
$t>1$,	after P2



Line applications



Line related formulas

- Length = sqrt $(x2 - x1)^2 + (y2 - y1)^2$
- Midpoint, p3, between p1 and p2

  p3 = ((x1 + x2) / 2, (y1 + y2) / 2))
- Two lines are perpendicular if:

  m1 = -1/m2

  cos(θ) = 0

  dot product of related vectors is zero (u•v=0)
- Point P3 lays on the line P1P2 if:

  Cartesian space, slope based definition:

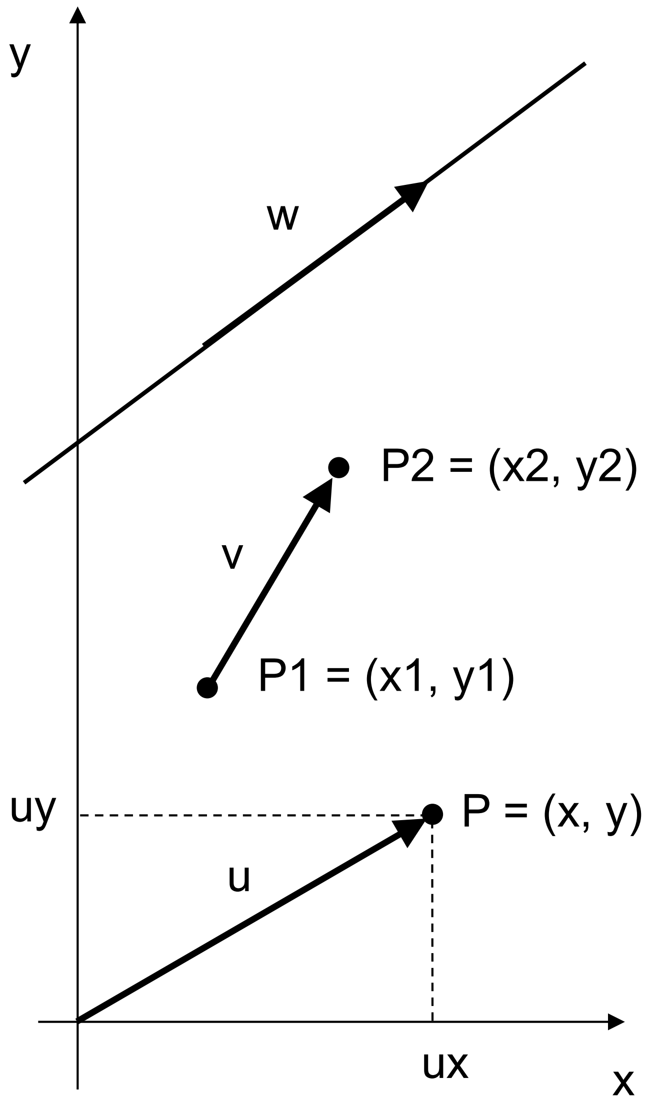
    y3 = mx3 + b, where m(P1,P2), b(P1,P2)

  Affine space, parametric definition:

    P3 = P1 + t(P2-P1), the system of equation on x, y (and z) should have solution. The position depends on the t value (<0, 0, 0..1, 1, 1<)

Vectors

- Two types of elements:
  - Scalars (real numbers): a, b, g, d, …
  - Vectors (n-tuples): u, v, w, …
- Used for:
  - Points in space (i.e., location): u
  - Displacements from point to point: v
  - Direction (i.e., orientation): w
- Operations:
  - Addition
  - Subtraction
  - Dot Product
  - Cross Product
  - Norm

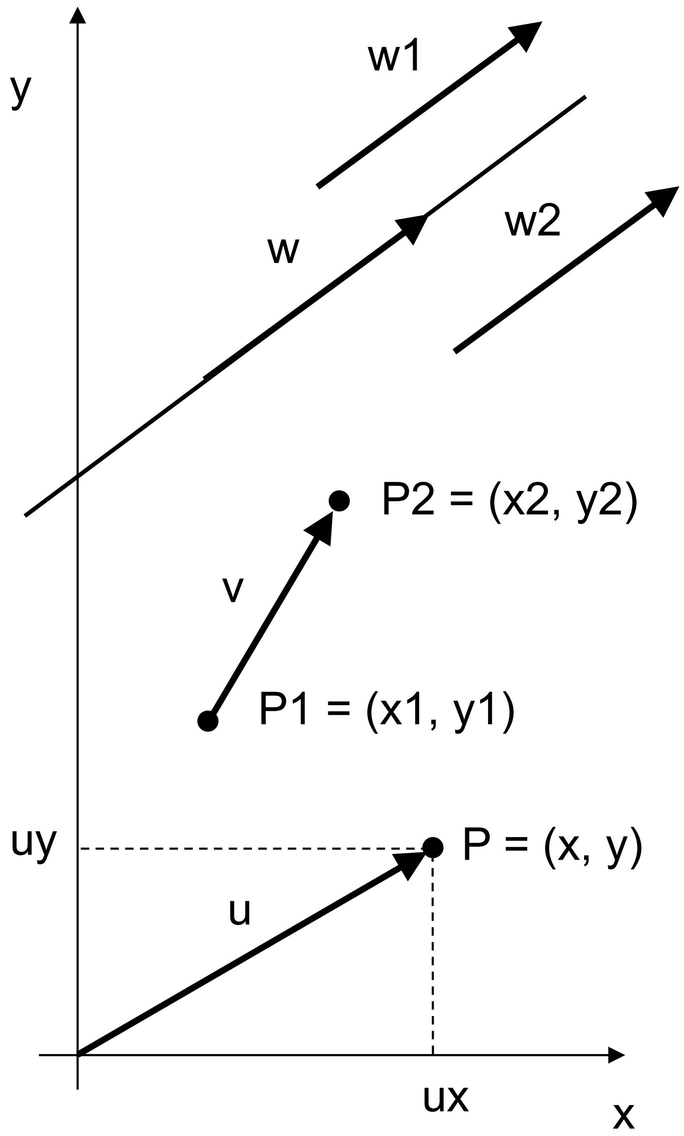


Vectors

$$\mathbf{w} = \begin{bmatrix} w_x \\ w_y \end{bmatrix} \qquad w = w1 = w2$$

$$\mathbf{v} = \begin{bmatrix} x2 - x1 \\ y2 - y1 \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

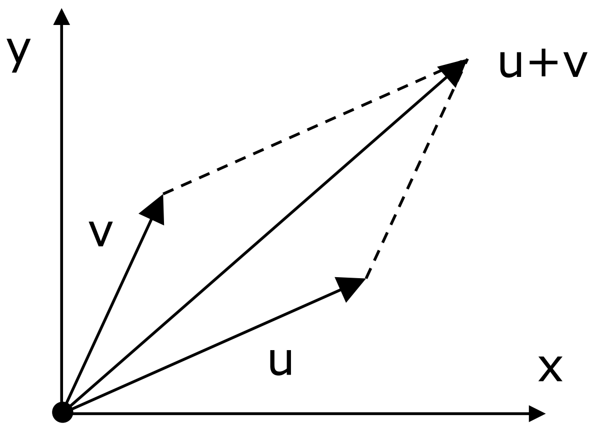


Vector operations

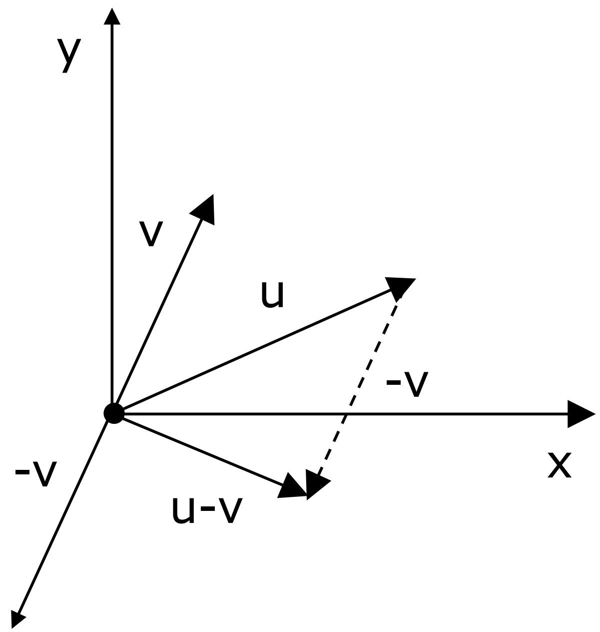
- Addition
- Subtraction
- Multiplication by a scalar
- Distributive rule: $a(u + v) = au + av$

  $(a + b)u = au + bu$

$$a\mathbf{u} = \begin{bmatrix} a u_x \\ a u_y \end{bmatrix}$$



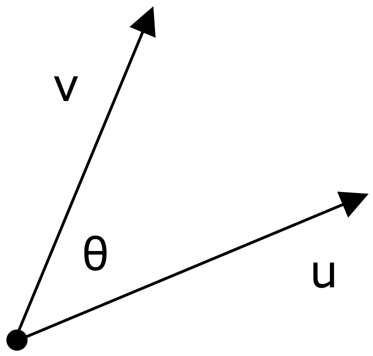
$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} u_x + v_x \\ u_y + v_y \end{bmatrix}$$



Dot product

- The dot product or, more generally, inner product of two vectors is a scalar:

  $u \bullet v = u_x v_x + u_y v_y + u_z v_z$ (in 3D)



- Is commutative

  $u \bullet v = v \bullet u$

- Is distributive with respect to addition

  $u \bullet (v + w) = u \bullet v + u \bullet w$

- Used for:
  - Computing the length (Euclidean Norm) of a vector:
    $\text{length}(v) = |v| = \text{sqrt}(v \bullet v)$
  - Computing the distance between two points P and Q:
    $v = Q - P$, $\text{distance}(P, Q) = \text{length}(v) = |v| = \text{sqrt}(v \bullet v)$
  - Normalizing a vector, making it unit-length: $v = v / |v|$
  - Checking two vectors for orthogonality
    $u \bullet v = 0$

Dot product

- Computing the angle between two vectors:

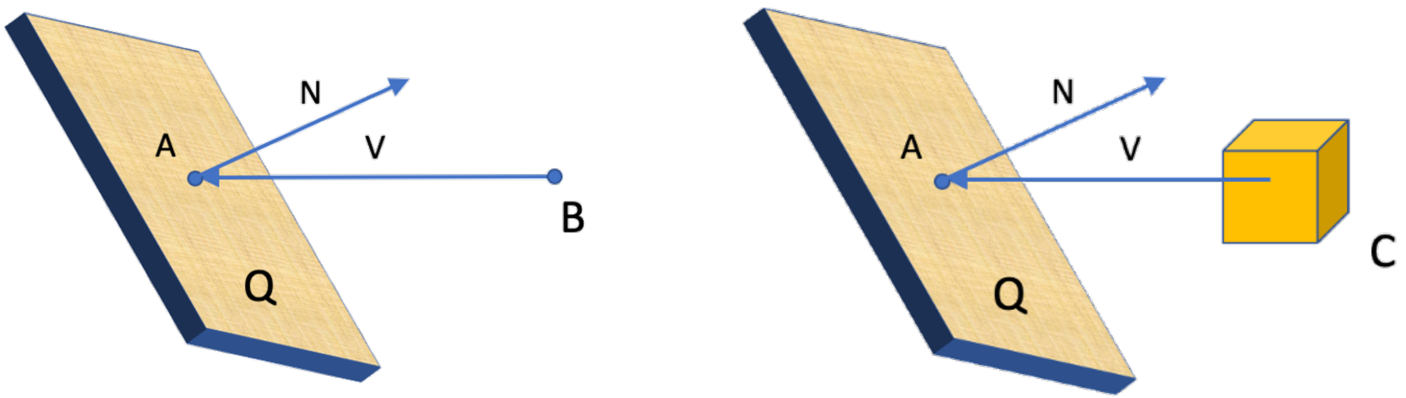
  u • v = |u| |v| cos(θ)

  If u • v < 0 then θ>90°, u • v = 0 then θ=90°, u • v > 0 then θ<90°

- Relative position of a point B against a plane Q

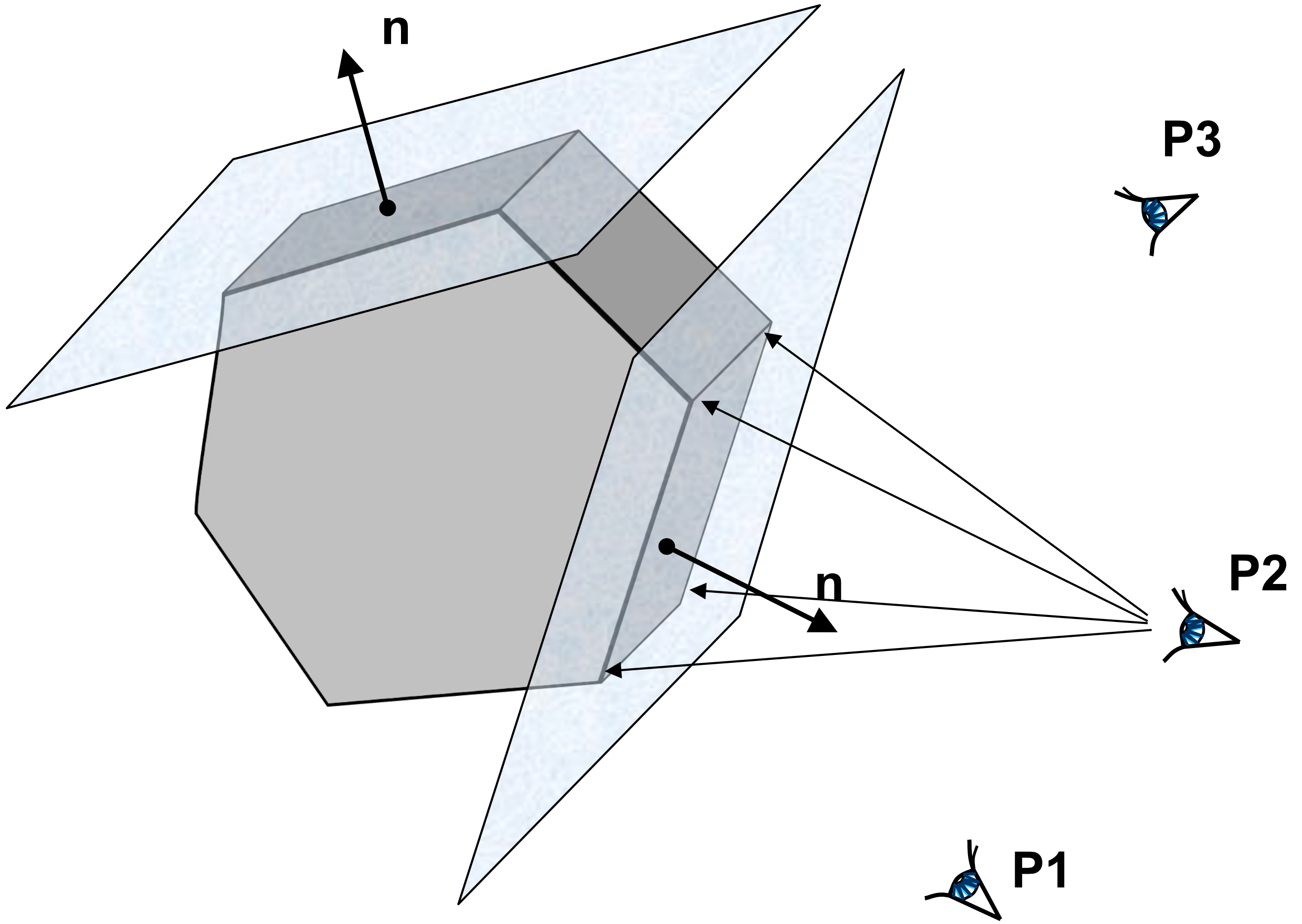
  If the vector v = A-B, where A is any point on the plane, and the normal vector of the plane is N, v and N gives an angle:

  - θ>90° then B is in front of the plane
  - θ=90° then B is on the plane
  - θ<90° then B is on back of the plane



Dot product

- If the viewer (P) is in front of the lateral plane polygon of a convex 3D object, the polygon is visible



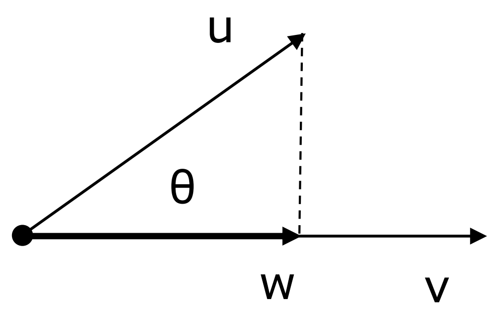
Dot product

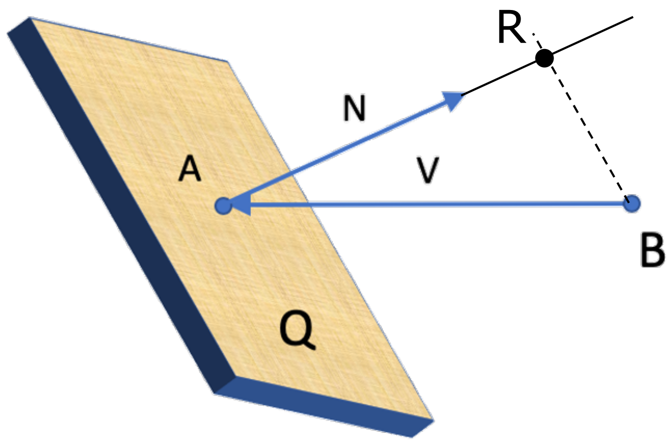
- Projecting a vector onto another
  - If v is a unit vector and we have another vector, u
  - We can project u perpendicularly onto v
  - And the result w has length u • v

$$|w| = ||u|\cos(\theta)| = ||u|\left[\frac{u \bullet v}{|u| \cdot |v|}\right]| = u \bullet v$$

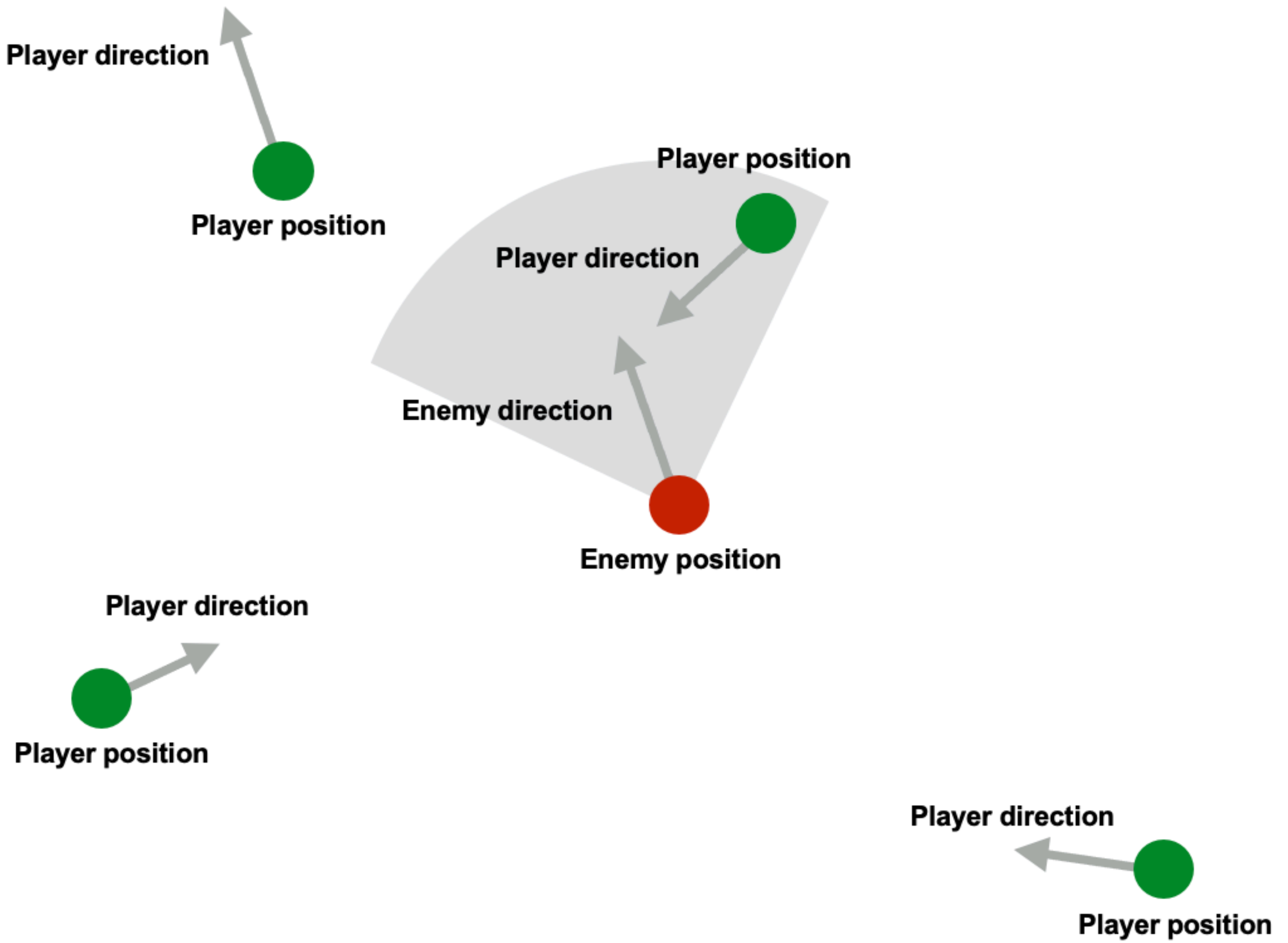
- Computes the distance between poin B and a plane Q.

  Normal vector is normalized (|N|=1), then distance =|A-R|

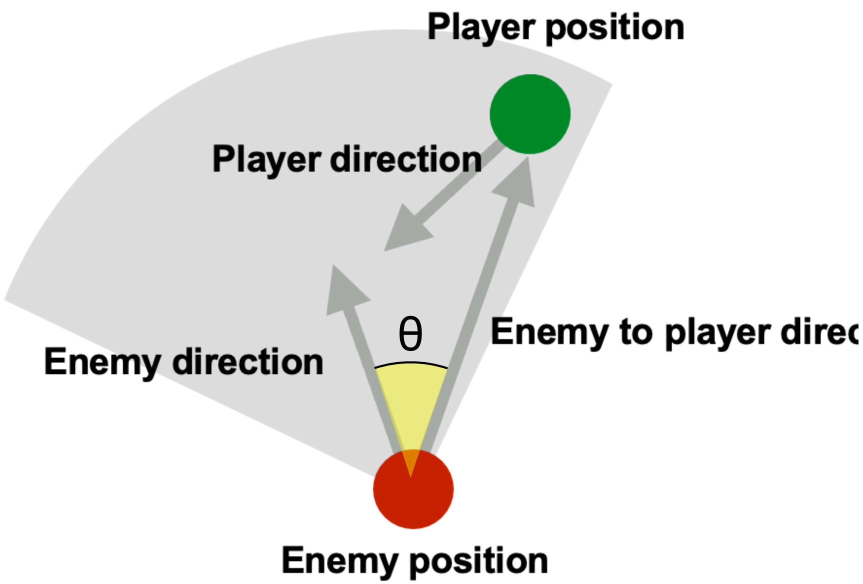


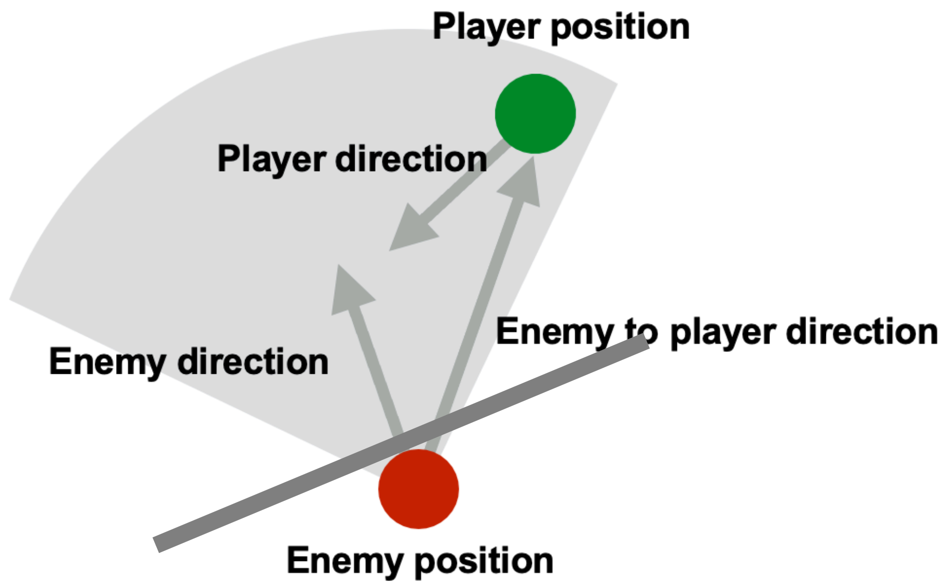


Dot product - application



Dot product - application





Angle θ < 90º

dot product (enemy_direction, enemy_to_player) > 0

Angle θ < FOV/2

cos(θ) > cos(FOV/2)

Player in front of the Enemy_Plane

dot product (enemy_direction, enemy_to_player) > 0

Question: How do you determine if the player and the enemy are heading towards each other.

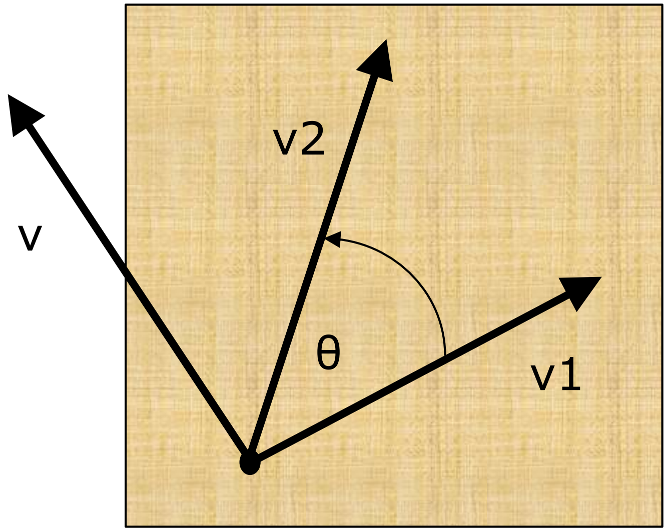
Cross product

- The cross product or vector product of two vectors is a vector orthogonal to both,

  $$v = v1 \times v2 = n\,|v1|\,|v2| \sin(\theta),$$

  where n is normal vector to both v1 and v2
- Right-hand rule dictates direction of cross product

$$v1 = \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} \quad v2 = \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix} \quad v = v1xv2 = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{bmatrix} \quad v = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x1 & 0 & 0 \\ 0 & y2 & 0 \end{bmatrix}$$



i=(1,0,0), **j**=(0,1,0), **k**=(0,0,1) – Cartesian system

$$v = v1xv2 = \begin{bmatrix} y1\,z2 - y2\,z1 \\ -(x1\,z2 - x2\,z1) \\ x1\,y2 - x2\,y1 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ 0 \\ x1\,y2 \end{bmatrix}$$

Cross product

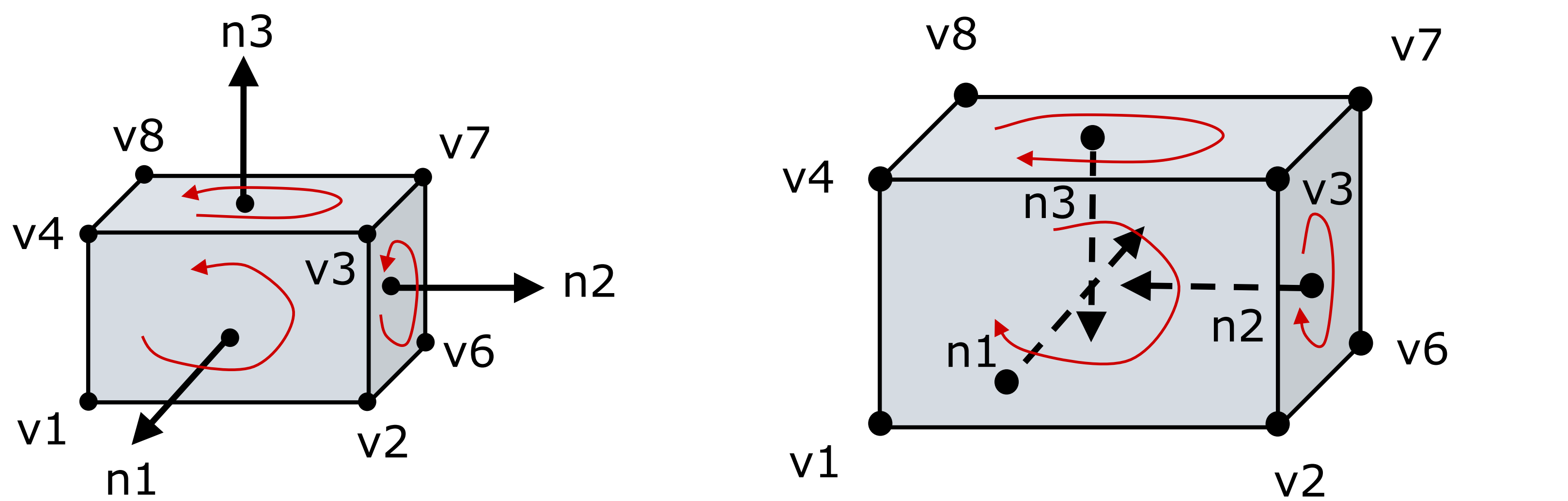
- Application:

  cross product computes the normal vector to a plane polygon

  n1 = e1 x e2, where e1 and e2 could be given by two consecutive edges of a poligon v1v2 and v2v3 respectively.

- Convention

  1. Normal vector points away of the object
     - Counterclockwise vertex sequence, e.g. v1, v2, v3, v4
  2. Normal vector points into the object
     - Clockwise vertex sequence, e.g. v1, v4, v3, v2

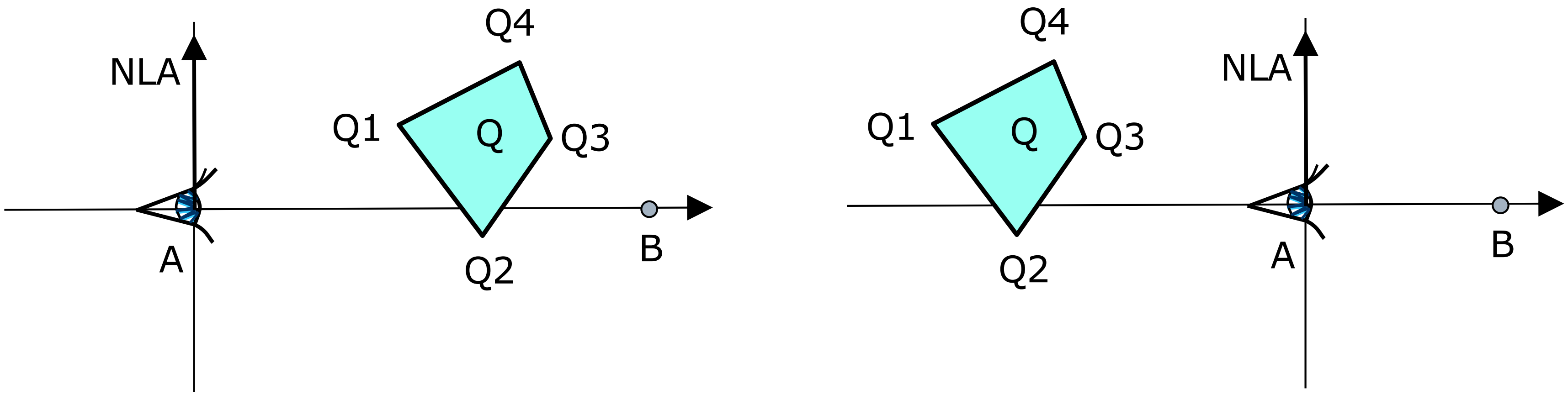


Dot and cross products – in front

- The position of the object Q relative to the viewer may be determine by using the dot or cross product.
- If the object Q is in front of the viewer:

  - dot product of vector AB and position vectors AQi, where i=1,2,3,4, is positive (same direction within 90°), AB · AQi >0

  - cross product of vector NLA (left normal vector of AB, placed on A) and position vectors AQi, where i=1,2,3,4, is negative (right side, within 180°), NLA x AQi<0

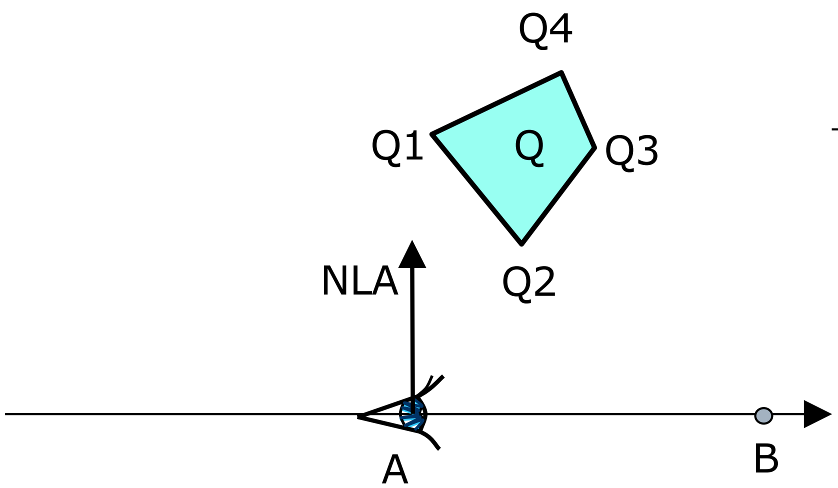


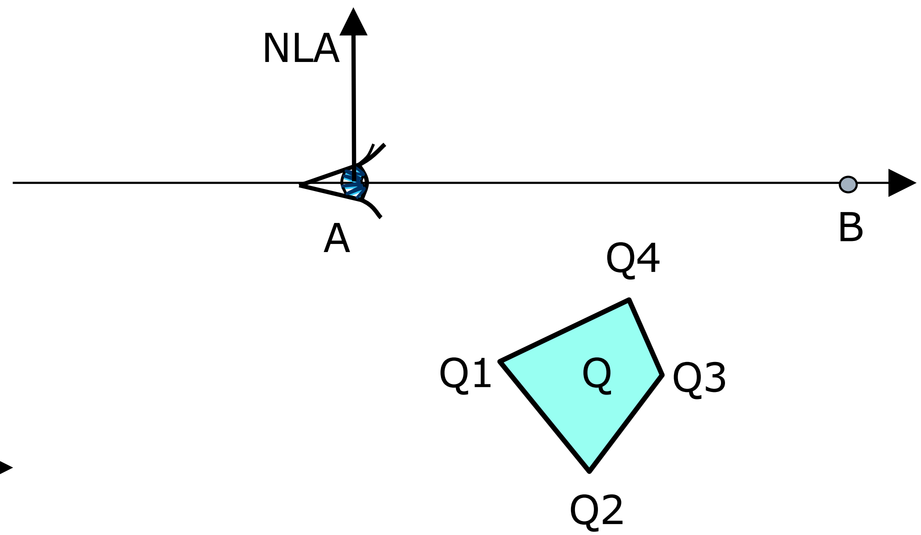
Dot and cross products – on left side

- The position of the object Q relative to the viewer may be determine by using the dot or cross product.
- If the object Q is on left side of the viewer:

  - cross product of vector AB and position vectors AQi, where i=1,2,3,4, is positive (left side within 180°), AB x AQi >0

  - dot product of vector NLA (left normal vector of AB, placed on A) and position vectors AQi, where i=1,2,3,4, is positive (same direction within 90°), NLA · Aqi>0





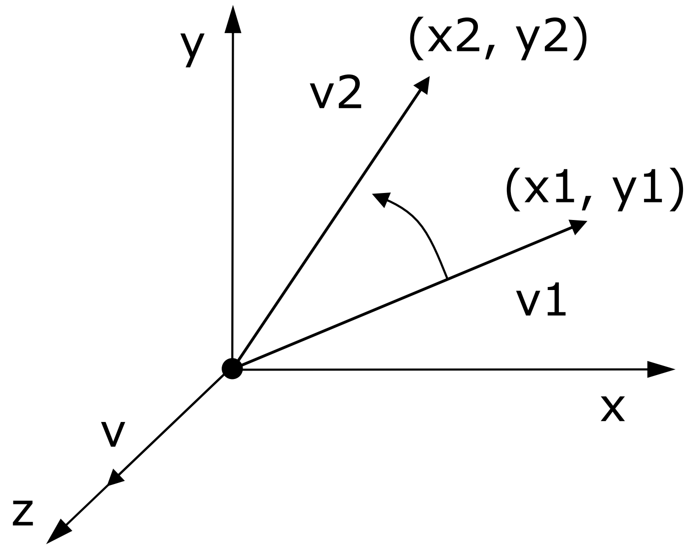
Triangle computation

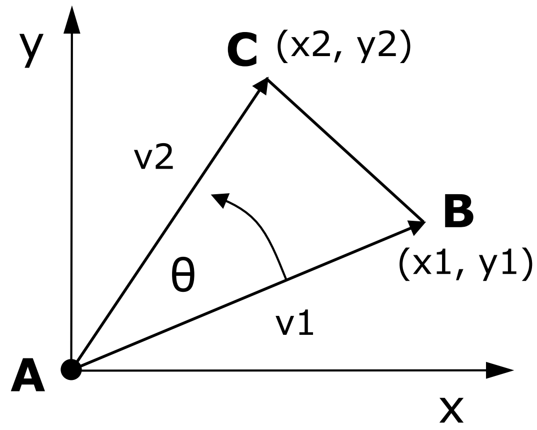
- Consider the general case of the cross product:

$$v = v1xv2 = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x1 & y1 & 0 \\ x2 & y2 & 0 \end{bmatrix}$$

$v = (x1\ y2 - x2\ y1)\ \mathbf{k}$

The cross product is a vector v along z, of module | x1 y2 – x2 y1|.





Area of a triangle

The triangle ABC has the area:

$\frac{1}{2}$ (x1-x2) (y1+y2) +

$\frac{1}{2}$ (x2-x0) (y2+y0) +

$\frac{1}{2}$ (x0-x1) (y0+y1) =

$\frac{1}{2}$ (x1y1+x1y2-x2y1-x2y2 +
x2y2+x2y0-x0y2-x0y0 +
x0y0+x0y1-x1y0-x1y1) =

$\frac{1}{2}$ (x0y1-x1y0+x1y2-x2y1+x2y0-x0y2)

where x0, y0 = 0

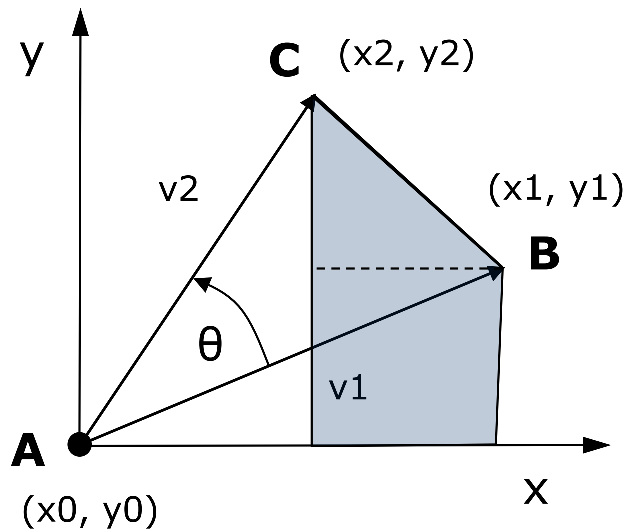
S = $\frac{1}{2}$ (x1y2-x2y1)

therefore

S = $\frac{1}{2}$ |v1 x v2|

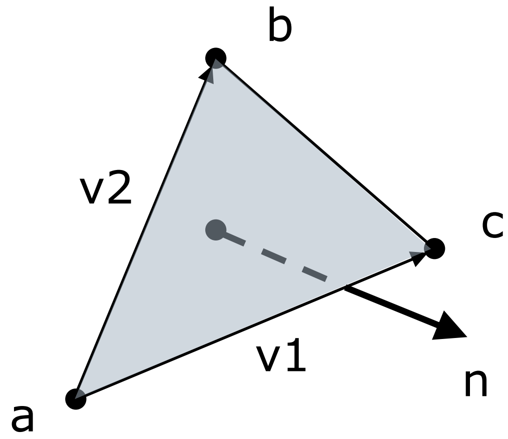
Moreover

v1x v2 = |v1| |v2| sinθ



Triangle computation

- Consider a triangle, (a, b, c)
  - a,b,c = (x,y,z) tuples
- Surface area by cross product
  - S = ½ |v1 x v2| = ½ * |(c–a) x (b-a)|
- Normal vector
  - N = e2xe1= (a-b) x (c-a) = (-(b-a)) x (c-a)
  - = (c-a) x (b-a) = v1xv2
- Normalized normal vector
  - n = (1/2S) * N = (1/2S) * (c-a) x (b-a) = (1/2S) * v1xv2



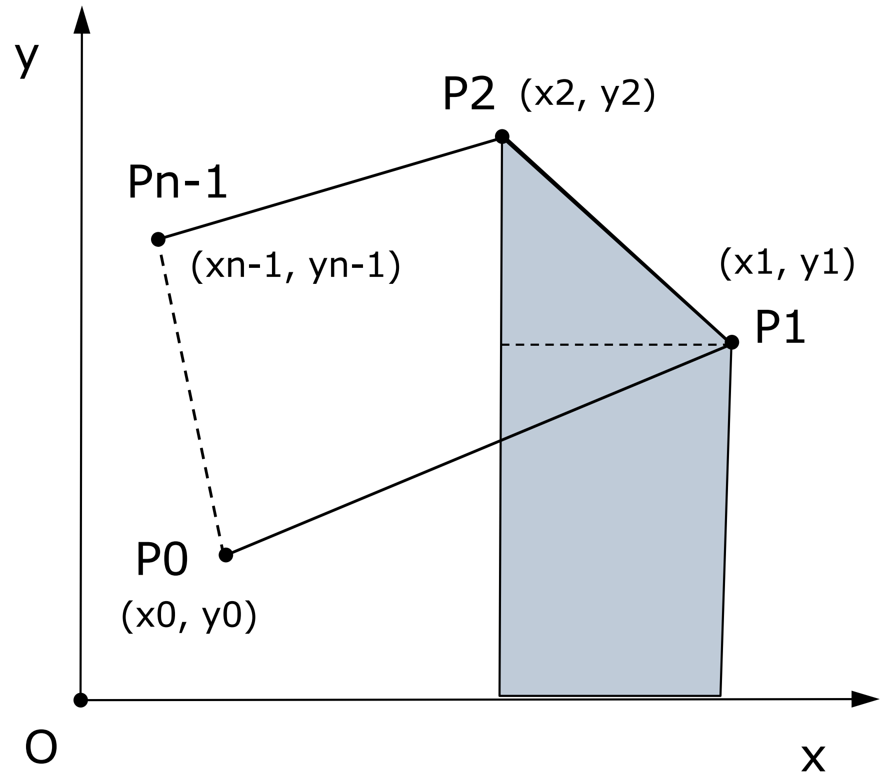
Area of a polygon

Polygonal area:

$$\tfrac{1}{2}(x0-x1)(y0+y1) +$$
$$\tfrac{1}{2}(x1-x2)(y1+y2) +$$
$$\ldots$$
$$\tfrac{1}{2}(xn-x0)(yn+y0) =$$

$$\tfrac{1}{2}(x0y0+x0y1-x1y0-x1y1 +$$
$$x1y1+x1y2-x2y1-x2y2 +$$
$$\ldots$$
$$xnyn+xny0-x0yn-x0y0)$$

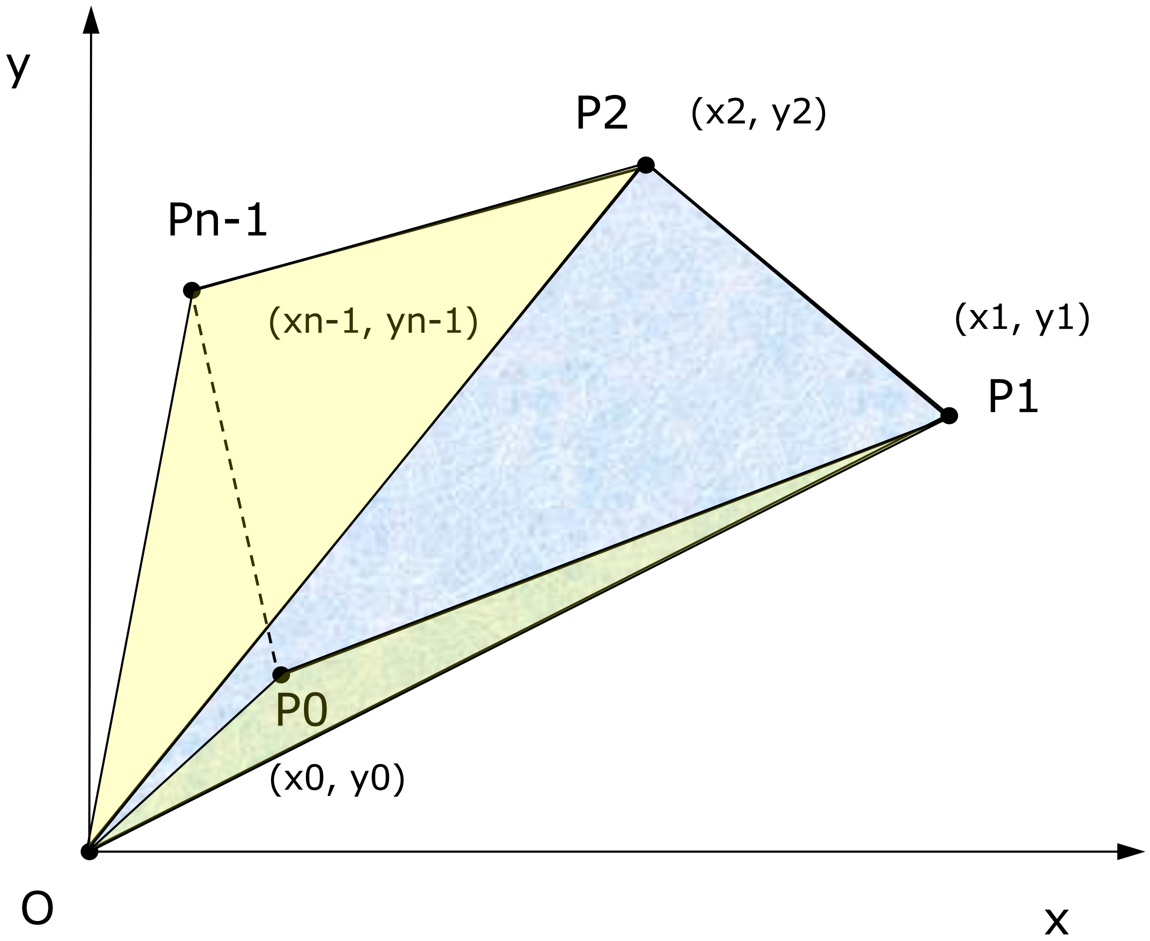
$$S = 1/2 \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$



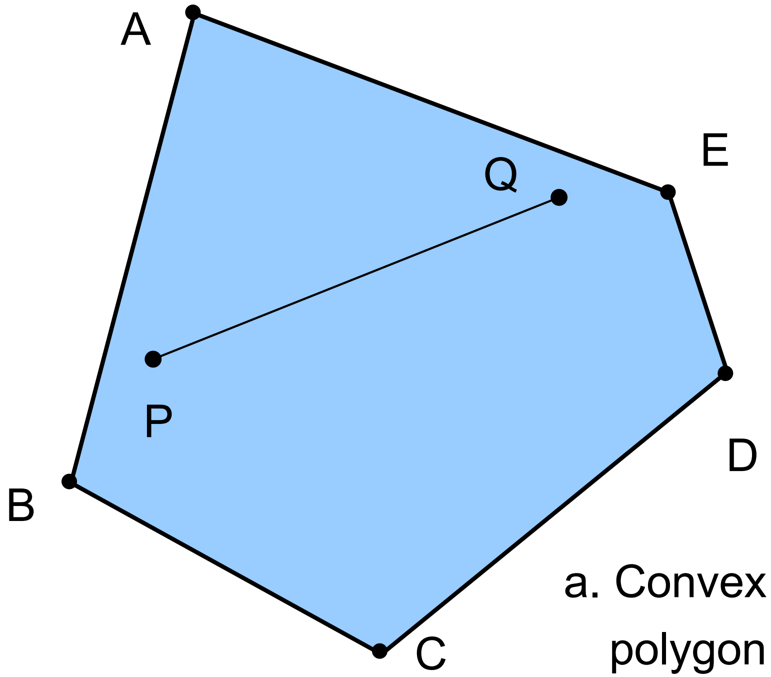
Area of a polygon

Polygonal area:

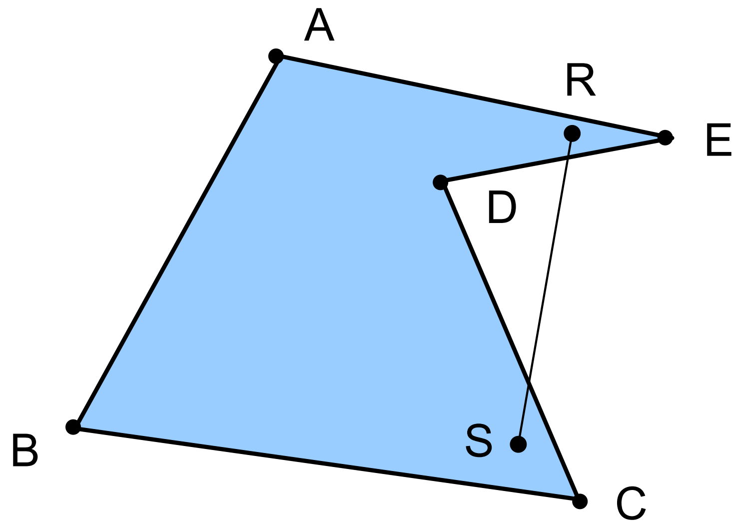
$$S = 1/2 \sum_{i=0}^{n-1} (v_i \times v_{i+1})$$



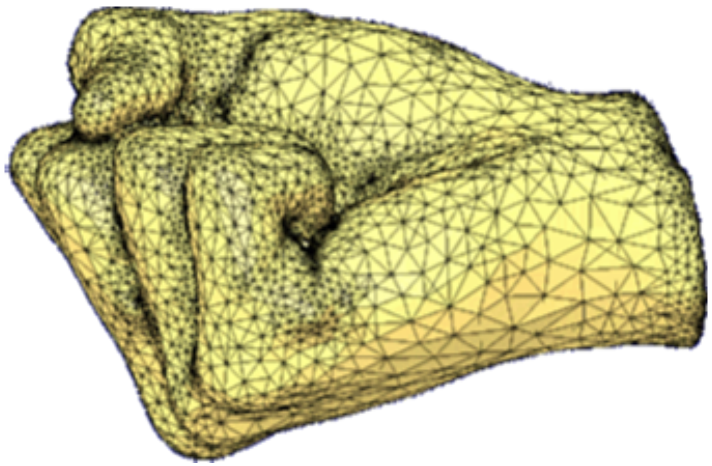
Convex and concave polygon

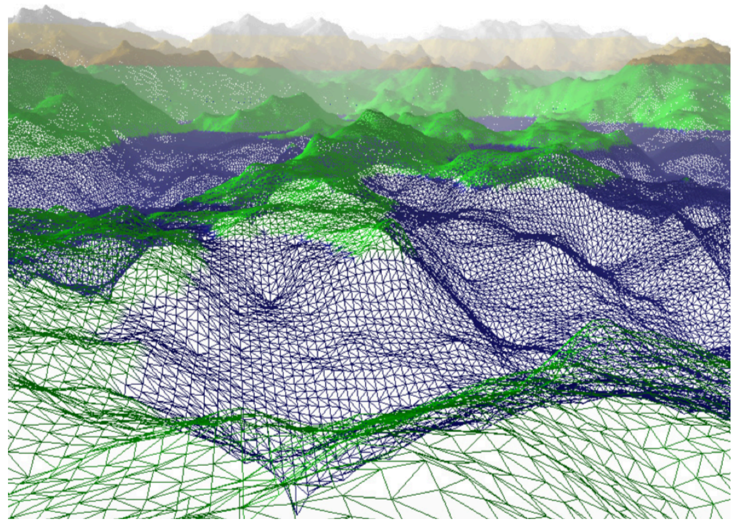


a. Convex polygon

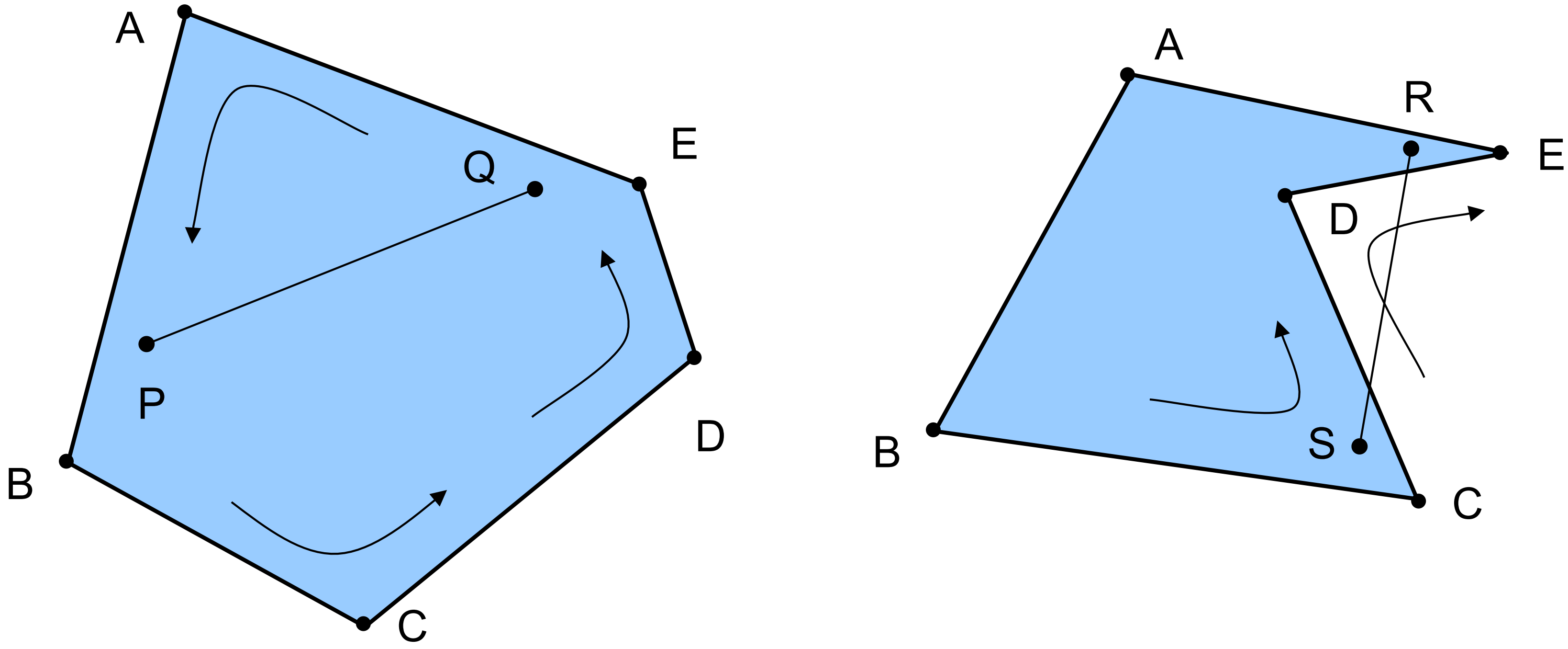


b. Concave polygon





Convex and concave polygon



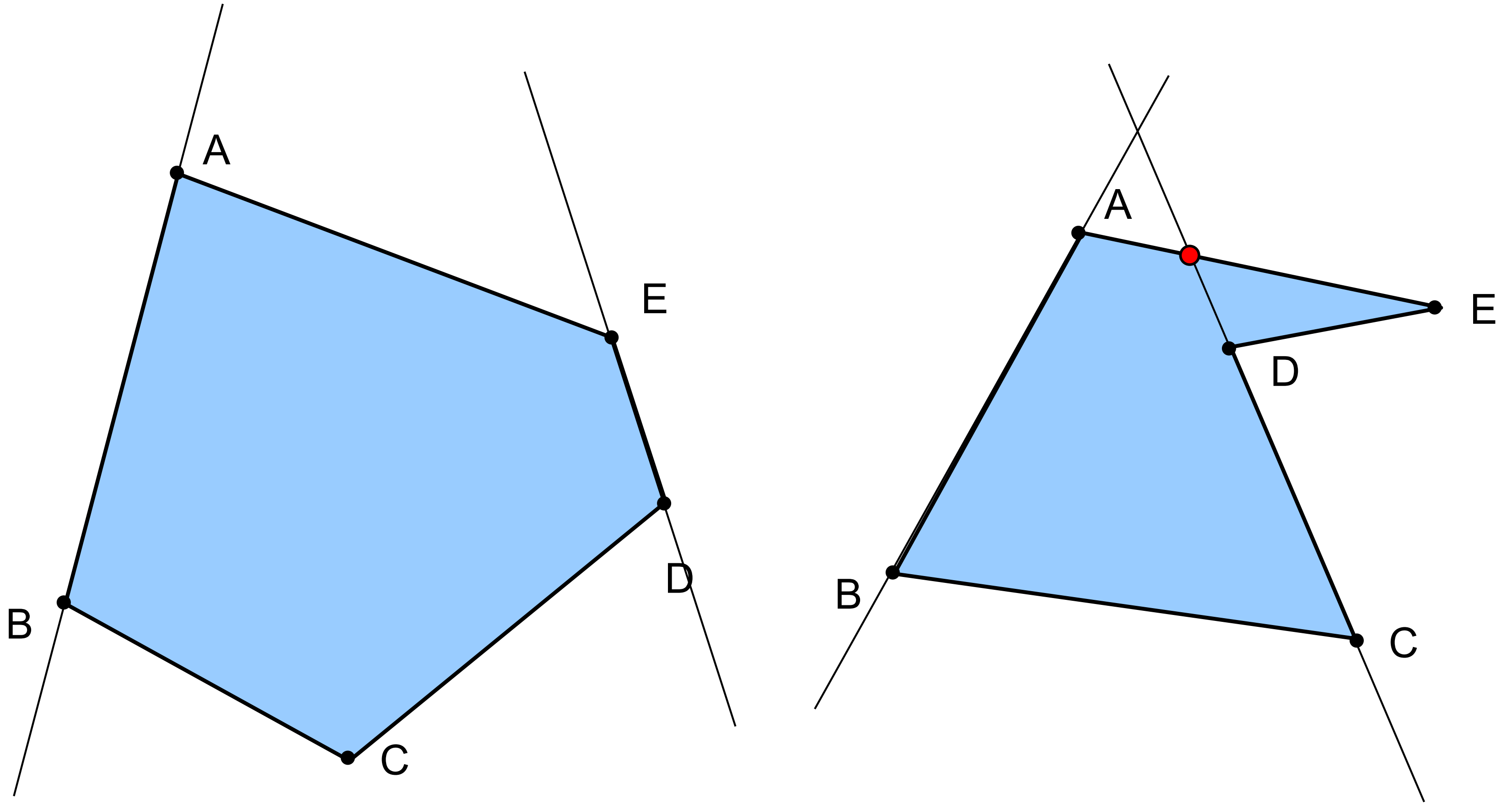
a. Convex polygon

b. Concave polygon

- Cross product of successive edges ($n = e1\ e2\ \sin(\theta)$)
- Turn directions
- Order of three successive vertices
- Angles
- Intersection between an edge and the polygon

Convex and concave polygon

- At least one intersection between an edge and the polygon



Test for convex and concave polygons

For the convex polygon all the cross products of adjacent edges will be the same sign.

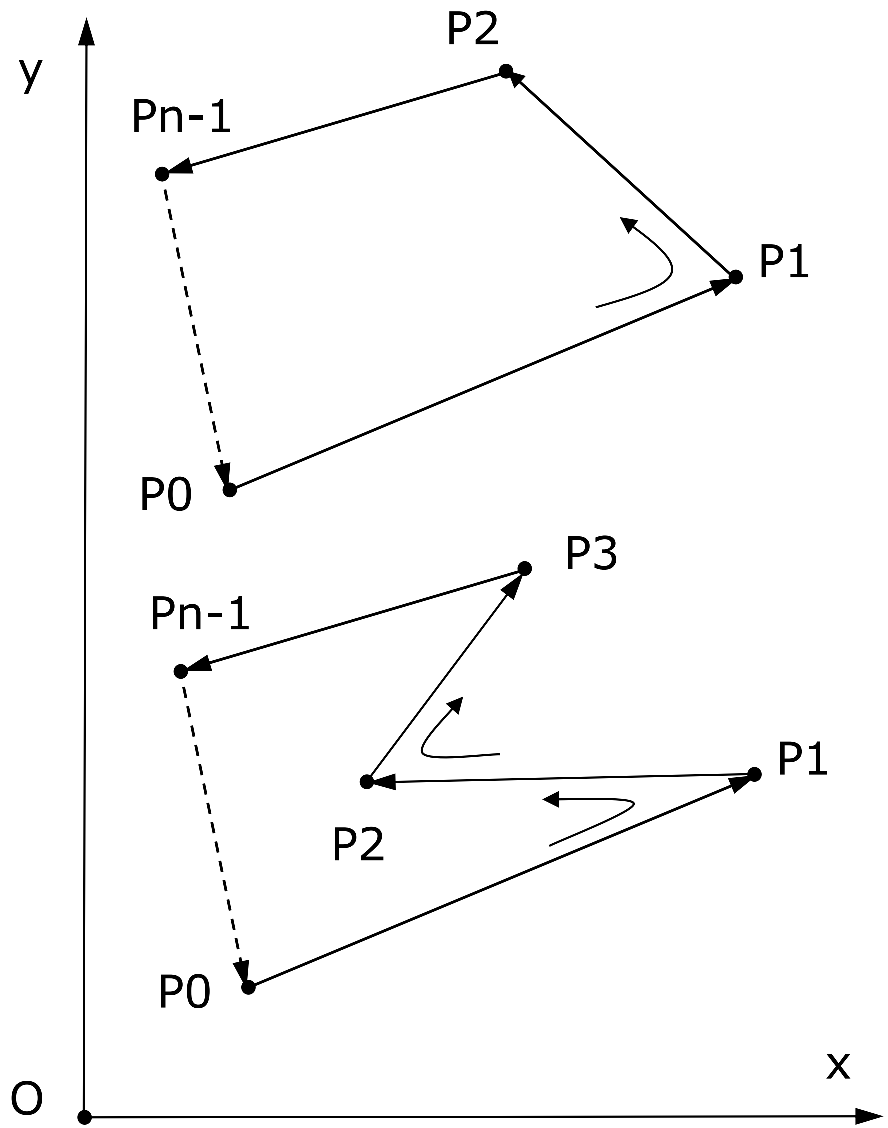
sign (e1xe2) = sign (e2xe3) = …
= sign (enxe1)

Obs: moving along the edges of the convex polygon it always turns left.

The concave polygon has a mixture of cross product signs.

NO: sign (e1xe2) = sign (e2xe3) = …
= sign (enxe1)

Obs: moving along the edges of the concave polygon it sometimes turns right.



Plane in Cartesian Space

- Plane equation:
  $$Ax + By + Cz + D = 0$$
- Normal vector:
  $$N\ [A, B, C]$$
- Equation by one point P(x0,y0,z0) and normal vector N:
  $$A(x-x0) + B(y-y0) + C(z-z0) + D = 0$$
- Plane defined by 3 points:
  $$P(x1, y1, z1),\ Q(x2, y2, z2),\ R(x3, y3, z3)$$

$$\begin{vmatrix} x & y & z & 1 \\ x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \end{vmatrix} = 0$$

$$A = \begin{vmatrix} y1 & z1 & 1 \\ y2 & z2 & 1 \\ y3 & z3 & 1 \end{vmatrix} \qquad B = -\begin{vmatrix} x1 & z1 & 1 \\ x2 & z2 & 1 \\ x3 & z3 & 1 \end{vmatrix}$$

$$C = \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix} \qquad D = -\begin{vmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{vmatrix}$$

Plane in Affine Space

- Parametric plane equation by 3 points P, Q, and R:

  $A(s,t) = (1-s)((1-t)P + tQ) + sR$

  Or another form $A(s,t) = P + t(Q-P) + s(R-(P + t(Q-P)))$

- Equation of the plane through P and normal vector v as dot product:

  $(X - P)\bullet v = 0$

- Intersection of a line with the plane:

  Plane equation $(X - P)\bullet v = 0$

  Line equation $Q(t) = Q + tw$

  Replace the point X by Q(t) for the intersection

  $$(Q + tw - P)\bullet v = 0$$

  Gives $t = \dfrac{((P - Q)\bullet v}{w\bullet v}$

  And the intersection point is $Q + \dfrac{((P - Q)\bullet v}{w\bullet v}\, w$

Distance to a line

- Line equation is:

  $P(t) = P0 + t\ v$, t parameter, v direction vector

- Distance from R to the line:

  It is the point P(t) for which the distance is minimum

  $(R - P(t))\bullet v = 0$, and expanding

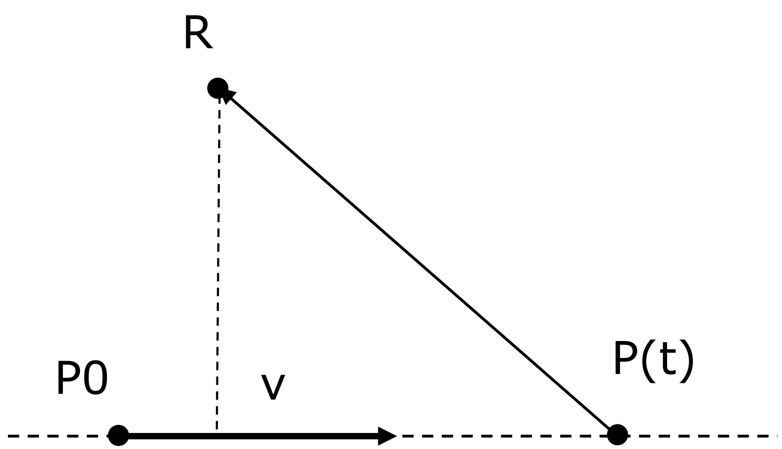
  $(R - P0 - t\ v)\bullet v = 0$

  $(R - P0)\bullet v = t\ v\bullet v$

  Result $t = \dfrac{((R - P0)\bullet v}{v\bullet v}$

  And the closest point to R is:

  $$P0 + \frac{((R - P0)\bullet v}{v\bullet v}\ v$$

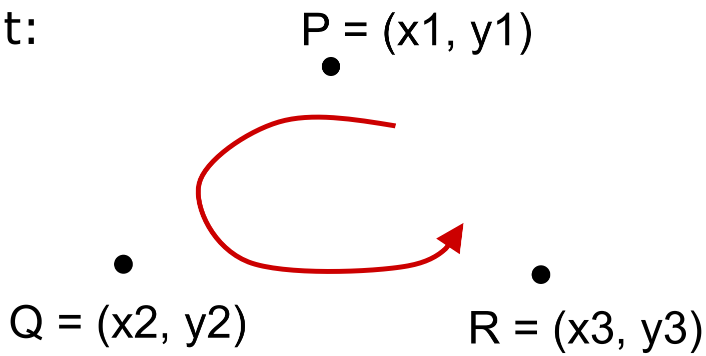


Point to point relationship

- Ordered points

  P, Q, and R counterclockwise order

  The order expressed as a determinant:



$$D = \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix}$$

If sign(D)=1 (+) then counterclockwise direction
sign(D)=0, collinear points
sign(D)=-1 (-), clockwise direction

Point to line relationship

- P(x0, y0) on the line

  P checks the equation: $P = P1 + t(P2-P1)$

  The system should have a real solution on t:

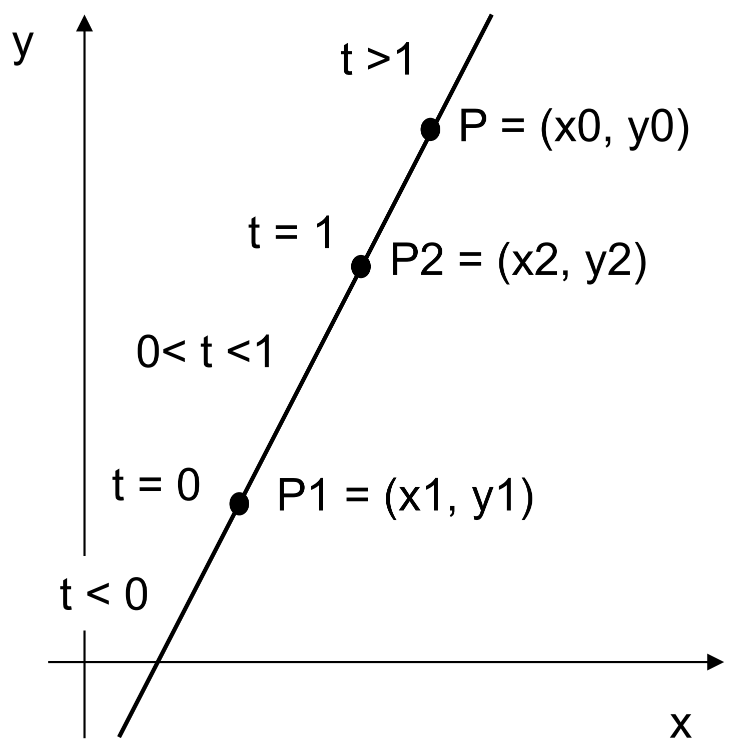
  $x0 = x1 + t(x2-x1)$

  $y0 = y1 + t(y2-y1)$

  Otherwise P is ouside of the line

  Discussion on the t:

t=0,	P=P1(x1, y1)
t=1,	P=P2(x2, y2)
0<t<1,	the P position between P1 and P2

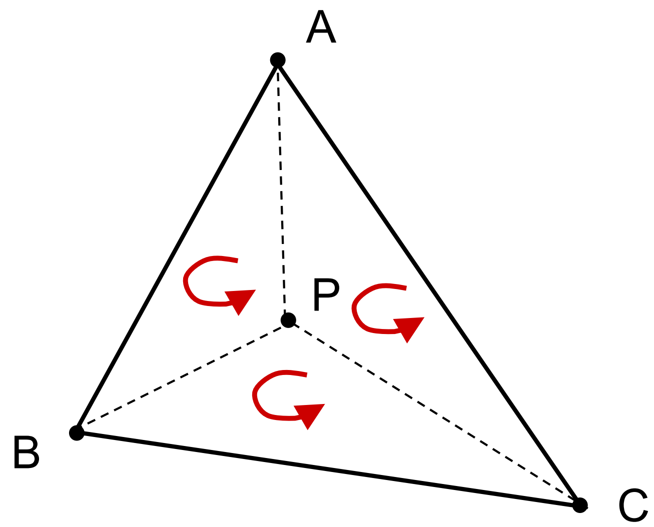


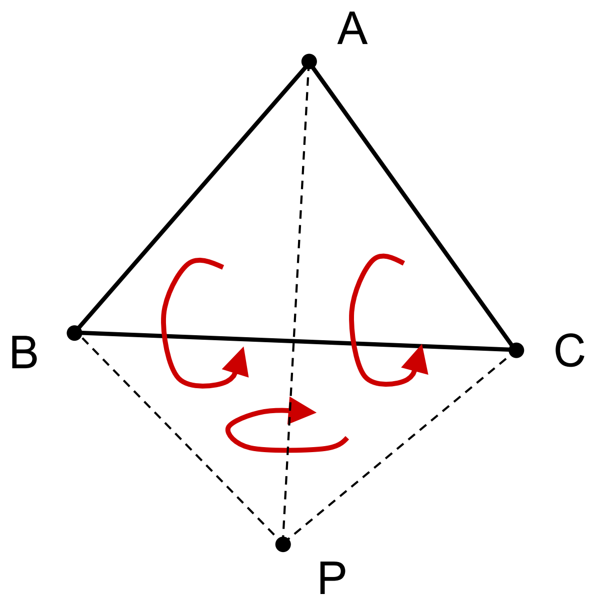
Point to triangle relationship

- Compute the order relationships of P against the triangle vertices A, B, and C
- Inside of the triangle

  sign(A,B,P)=sign(B,C,P)=sign(C,A,P)
- Outside of the triangle

  NO: sign(A,B,P)=sign(B,C,P)=sign(C,A,P)





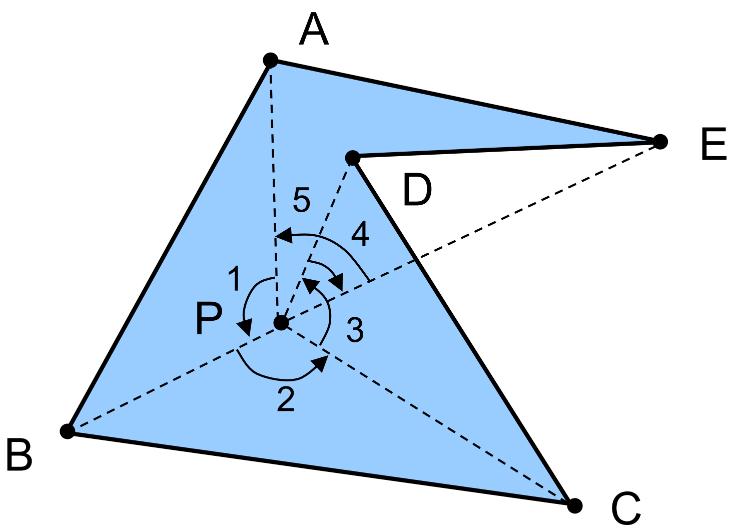
Point to polygon relationship

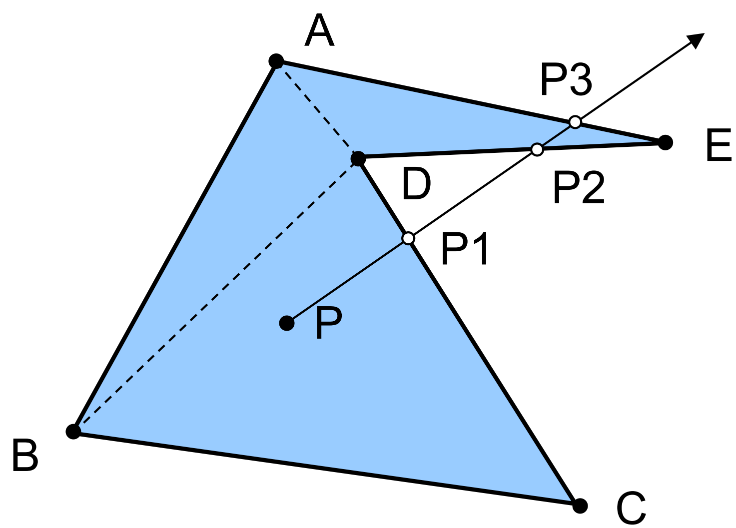
- P lays inside the polygon
  - Sum of angles given by the edges is 360°

    ^1 + ^2 + ^3+ ^4 + ^5 = 360°
  - P lays inside at least one triangle of the triangulation set

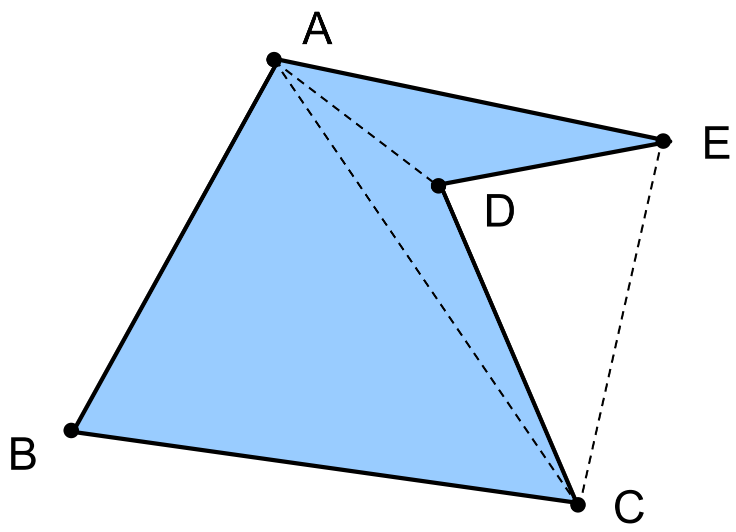
    ΔABD, ΔBDC, ΔDEA, P ∈ ΔABD
  - Number of intersection between a line starting on P and the triangle

    Odd number of intersection: P1, P2, P3





Polygon triangulation



1. Input circular list: IL = A, B, C, D, E

   Output list of triangles: OL
2. IL = A, B, C, D, E, OL = Ø,

   Analyze Δ ABC, Δ ABC ⊂ P

   OL ← Δ ABC, cut off B from IL
3. IL = A, C, D, E, OL = Δ ABC,

   analyze Δ CDE, Δ CDE ⊄ P

   OL ← Ø
4. IL = A, C, D, E, OL = Δ ABC,

   Analyze Δ ACD, Δ ACD ⊂ P

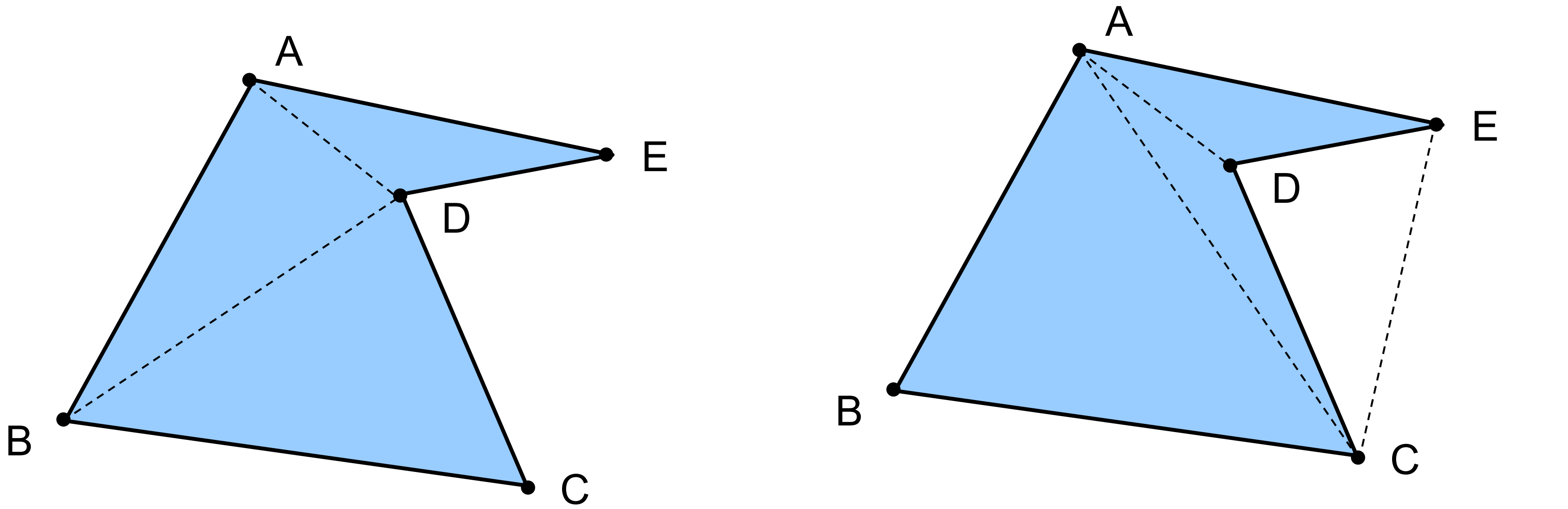
   OL ← Δ ACD, cut off C from IL
5. IL = A, D, E, OL = Δ ABC, Δ ACD

   Analyze Δ ADE, Δ ADE ⊂ P

   OL ← Δ ADE, cut off D from IL
6. IL = A, E, then stop

   OL = Δ ABC, Δ ACD, Δ ADE

Polygon triangulation



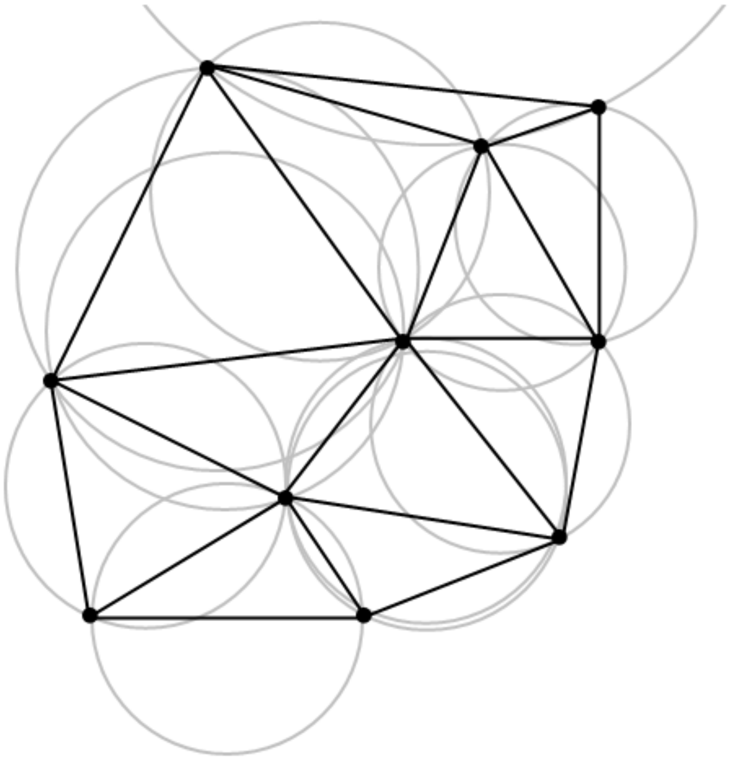
- Another approach: Delaunay triangulations

Delaunay triangulation

- Delaunay triangulation:

  for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P).

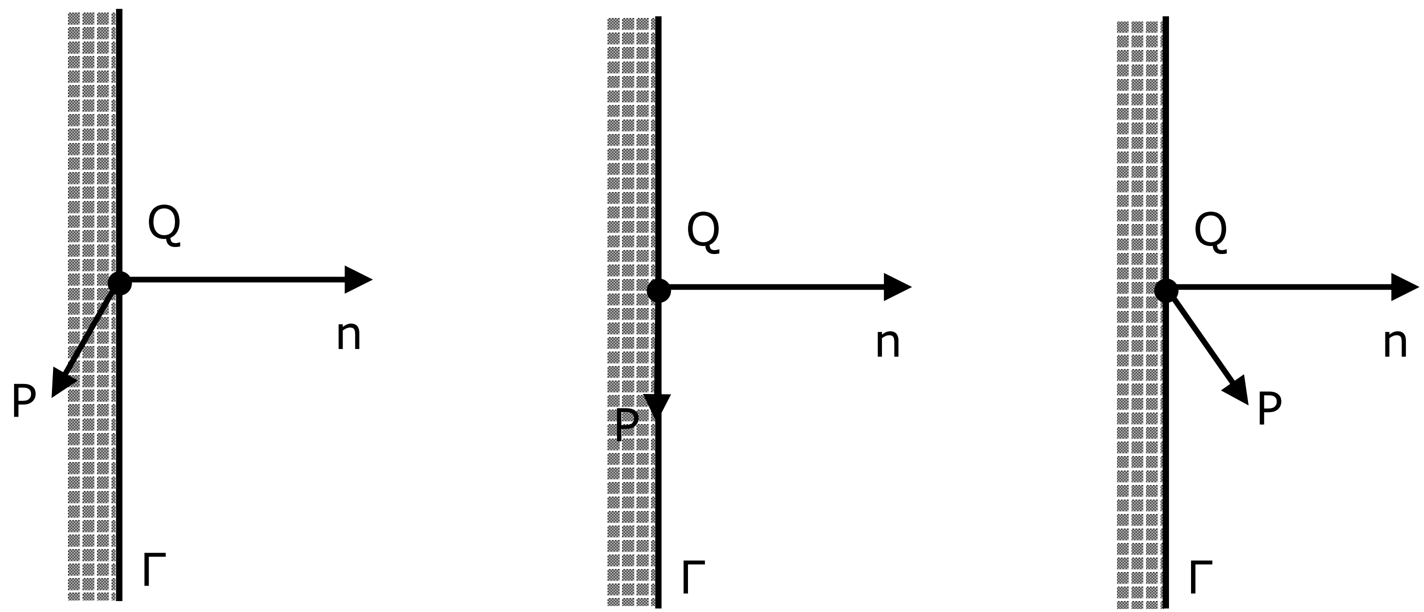
- Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation. They tend to avoid skinny triangles.
- The triangulation was invented by Boris Delaunay in 1934.



Point to plane relationship

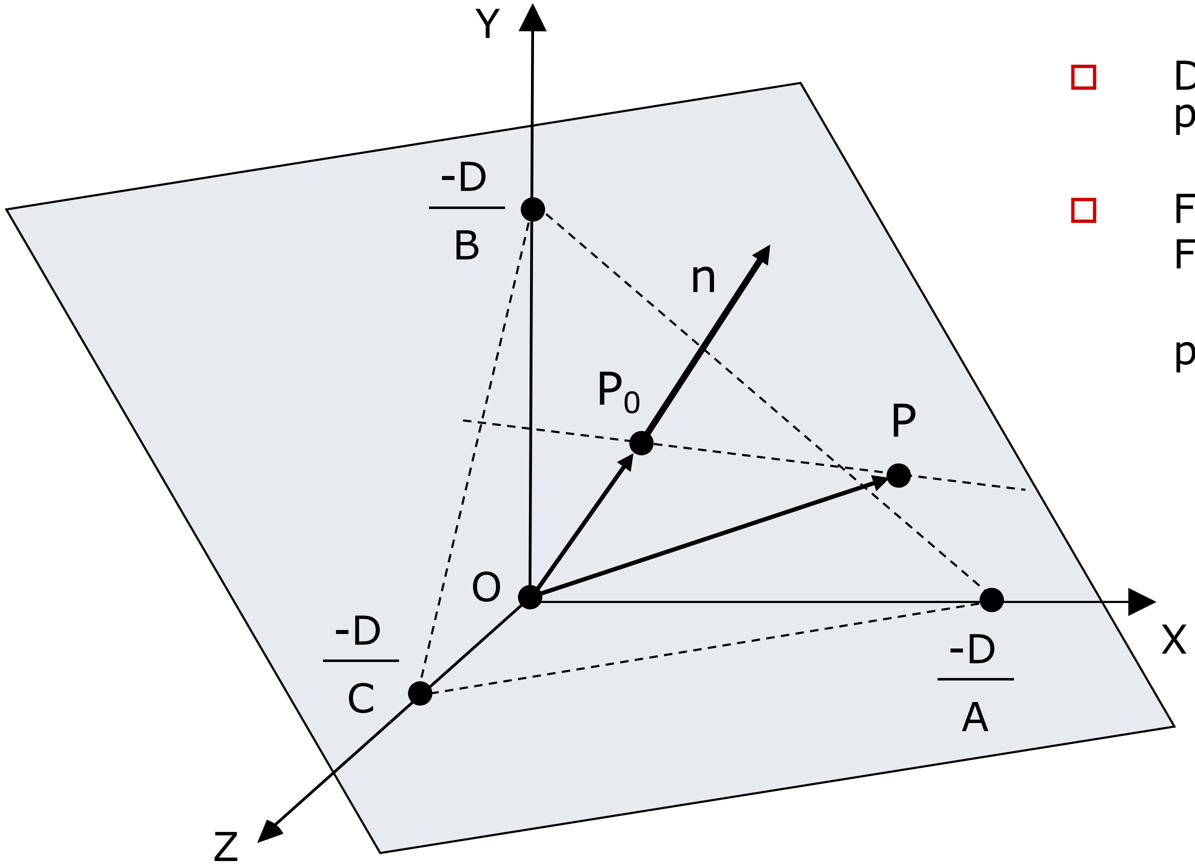
- A very general test to determine if a point P is "inside" a plane Γ, defined by the point Q and the normal vector n:

$(P - Q) \bullet n < 0$:	P inside Γ
$(P - Q) \bullet n = 0$:	p on Γ
$(P - Q) \bullet n > 0$:	p outside Γ



Point to plane relationship

- Plane Γ : Ax + By + Cz + D = 0
- Normal vector n [A B C]
- Plane Γ intersects the axes on -D/A, -D/B and -D/C
  If the intersection points are on the positive axes then A, B, C are positive and D is negative



- Distance from the origin O to the plane Γ is |D|
- For function
  F(x,y,z)= Ax + By + Cz + D,

  position of the point
  Q(xq, yq, zq) is:

  if F(Q) > 0 then Q in front of the plane Γ (semispace of n)

  if F(Q) = 0 then Q on the plane Γ

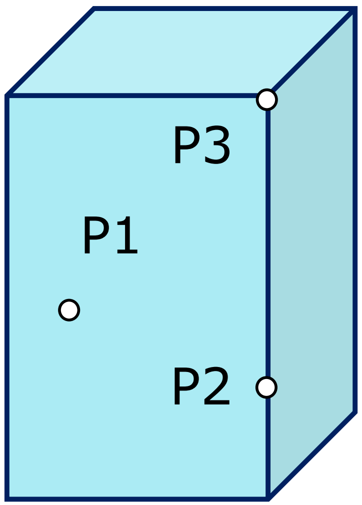
  if F(Q) < 0 then Q behind the plane Γ

Point to 3D convex object

- Compute the plane equation and the function:

  $$F_i(x,y,z)= A_i x + B_i y + C_i z + D_i$$

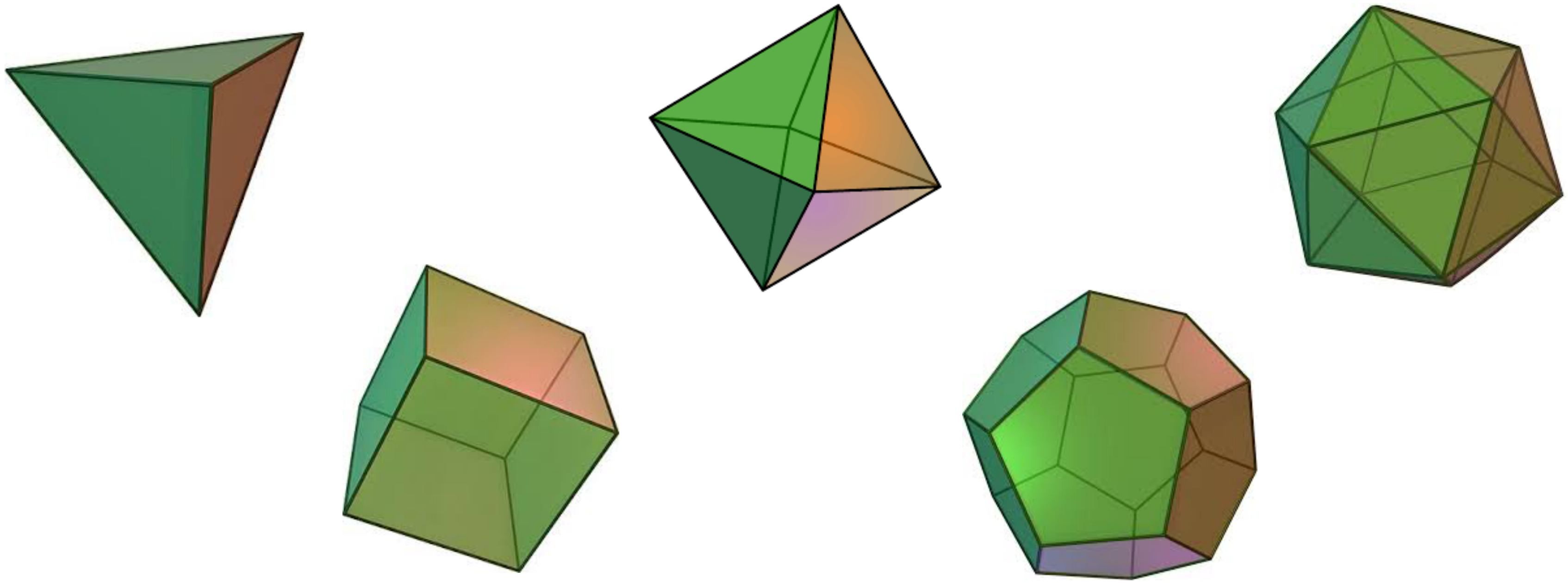
  for all polygonal faces of the object Ω
- Compute the normal vectors to point away from the 3D polygonal object
- The point P(xp,yp,zp) is located as follows:



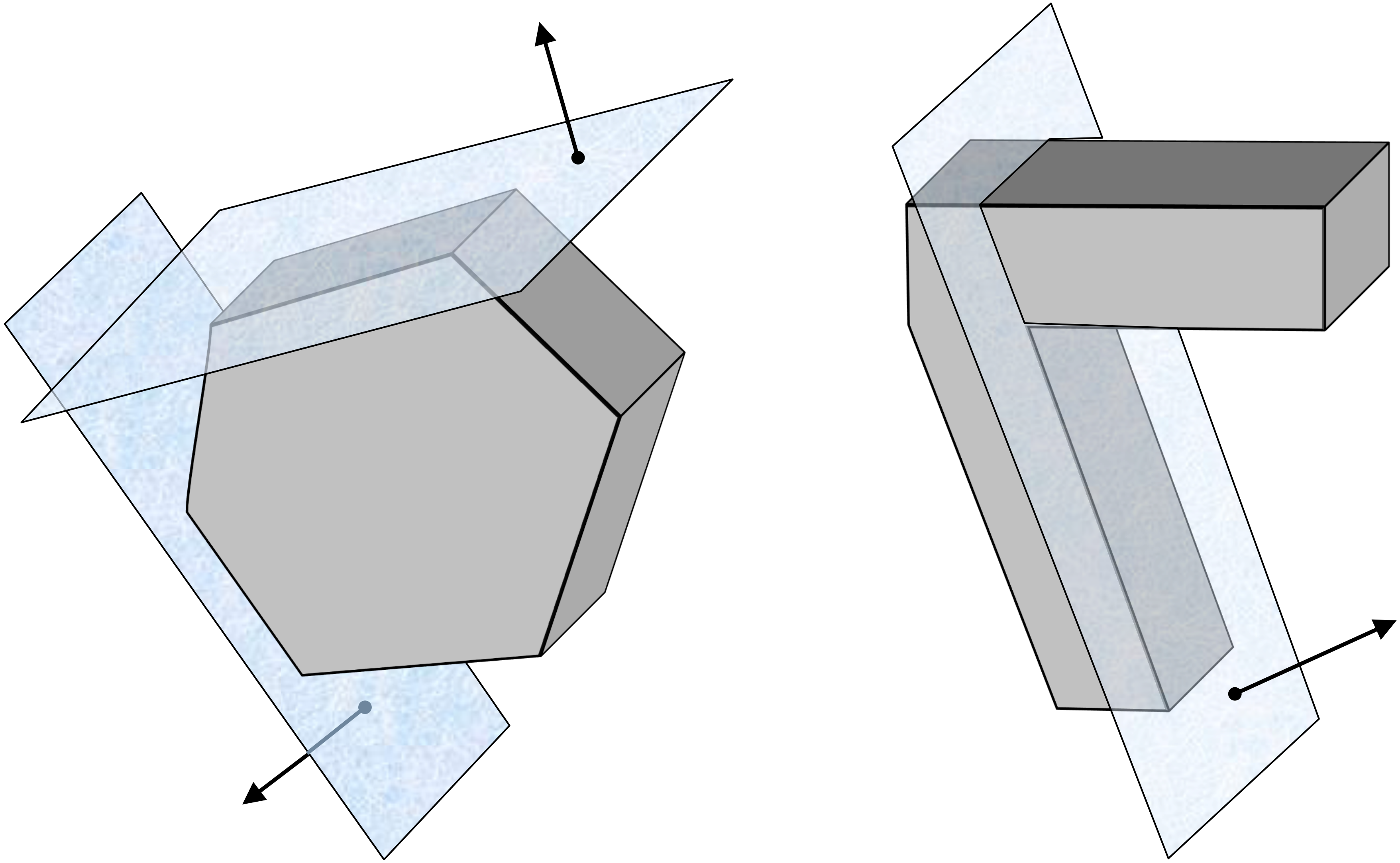
1. If at least one function gives $F(P)>0$, then P outside of Ω
2. Else if one function gives $F_k(P)=0$, then P on the k polygonal face
3. Else if two functions give $F_k(P)= F_q(P)=0$, then P on the edge between the k and q polygonal faces
4. Else if r functions give $F_{k1}(P)= F_{k2}(P)= ..=F_{kr}(P)= 0$, then P on the vertex between the k1, k2,…,kr polygonal faces
5. Else if all functions give $F_k(P)<0$, then P inside of Ω
6. Otherwise error

3D Objects - Polyhedrons

- Tetrahedron (4 faces), Cube (6), Octahedron (8), Dodecahedron (12), Icosahedron (20)
- Euler's Theoreme: V – E + F = 2



Convex and concave 3D object



Line to line relationship

- Resolve the system of two line equation

$$P = (1-t)P1 + t\ P2$$
$$P = (1-s)Q1 + s\ Q2$$

The equation should have real solution on t and s:

$$(1-t)P1 + t\ P2 = (1-s)Q1 + s\ Q2$$

In fact it is:

$$(1-t)xp1 + t\ xp2 = (1-s)xq1 + s\ xq2$$
$$(1-t)yp1 + t\ yp2 = (1-s)yq1 + s\ yq2$$

Discussion on the t and s:

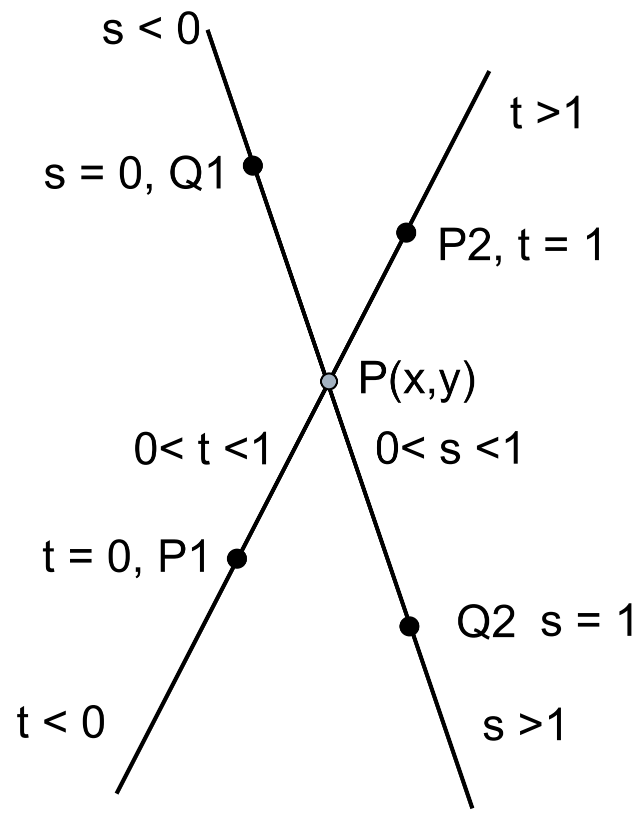
$t<0, s<0$ P before P1 and Q1

. . .

$0<t<1$ and $0<s<1$, P1P2 intersects Q1Q2

. . .

$t>1, s>1$ P after P2 and Q2



Line - plane intersection

- Parametric definition of edge:

  $L(t) = L_0 + (L_1 - L_0)t$

  If $t = 0$ then $L(t) = L_0$
  If $t = 1$ then $L(t) = L_1$
  Otherwise, $L(t)$ is part way from L_0 to L_1

- Edge intersects plane Γ where E(t) is on Γ

  Q is a point on Γ
  n is the normal vector to Γ

  $(L(t) - Q) \bullet n = 0$
  $(L_0 + (L_1 - L_0)t - Q) \bullet n = 0$
  $t = [(Q - L_0) \bullet n] / [(L_1 - L_0) \bullet n]$

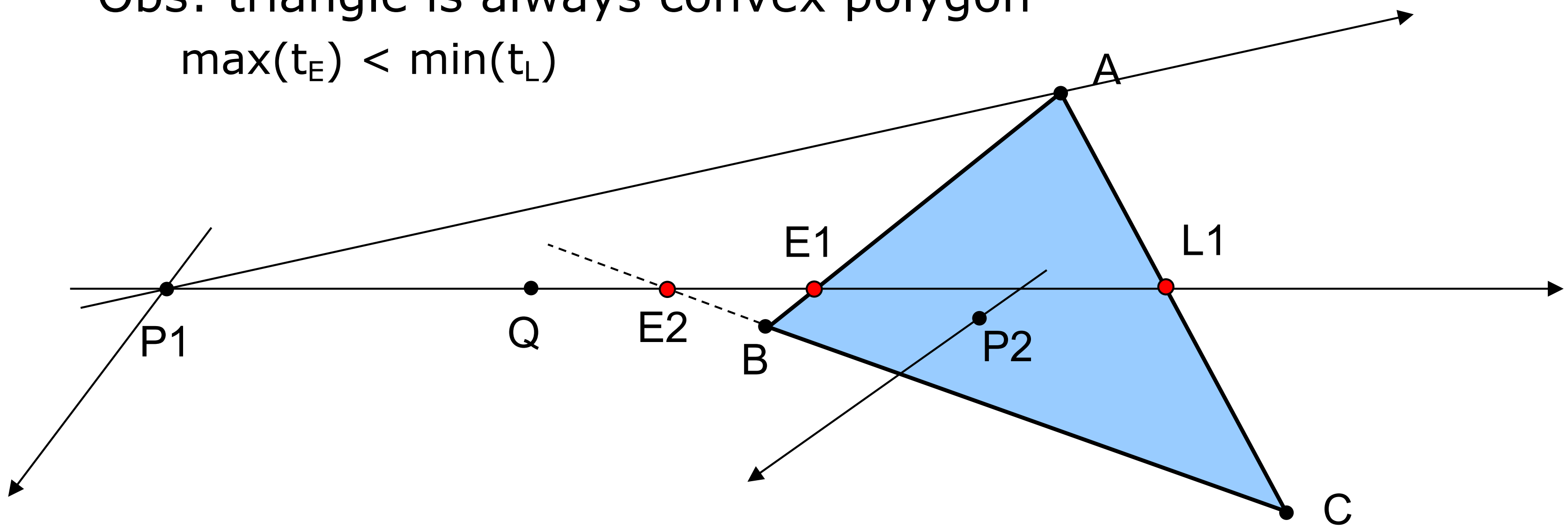
  The intersection point $I = L(t)$ for this value of t

Line to triangle relationship

- P and Q inside the triangle
- P and Q outside
  - PQ intersects the triangle
  - PQ does not intersects the triangle
- Solutions:
  - Analyse the line to edges relationships
  - Cyrus-Beck algorithm

  Obs: triangle is always convex polygon

  $\max(t_E) < \min(t_L)$



Cyrus-Beck algorithm

Cyrus-Beck algorithm (for convex polygons only):

1. Classify the edges by front face and back face;

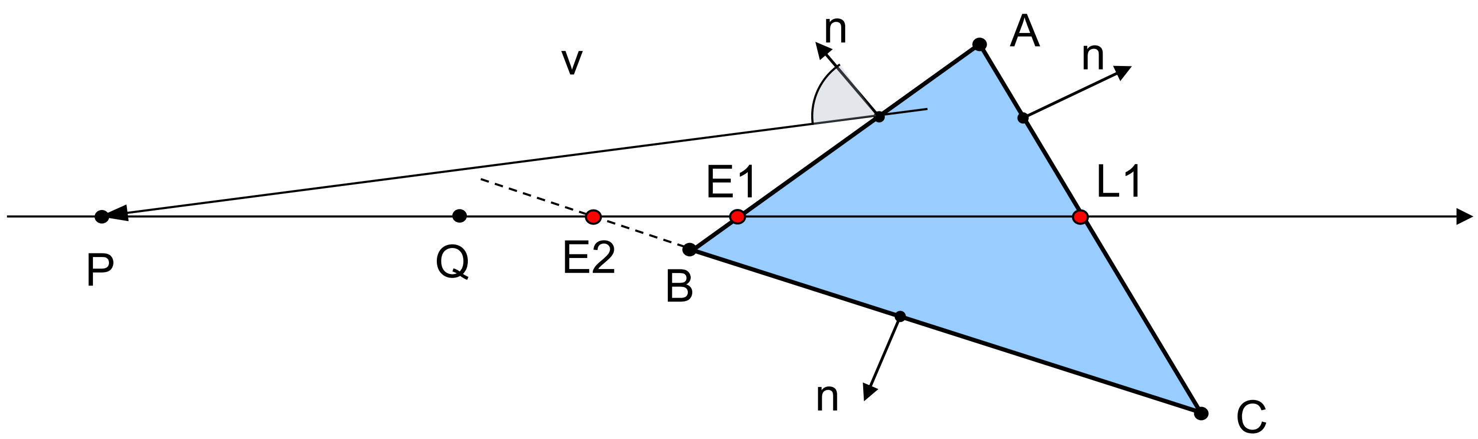
   dot product $v \bullet n > 0$ (front face, entering face, near face),
   $v \bullet n < 0$ (back face, leaving face, far face)

2. For each edge compute intersection with view ray;

   get a set of t_{near} and t_{far} for intersection points E1, E2, L1

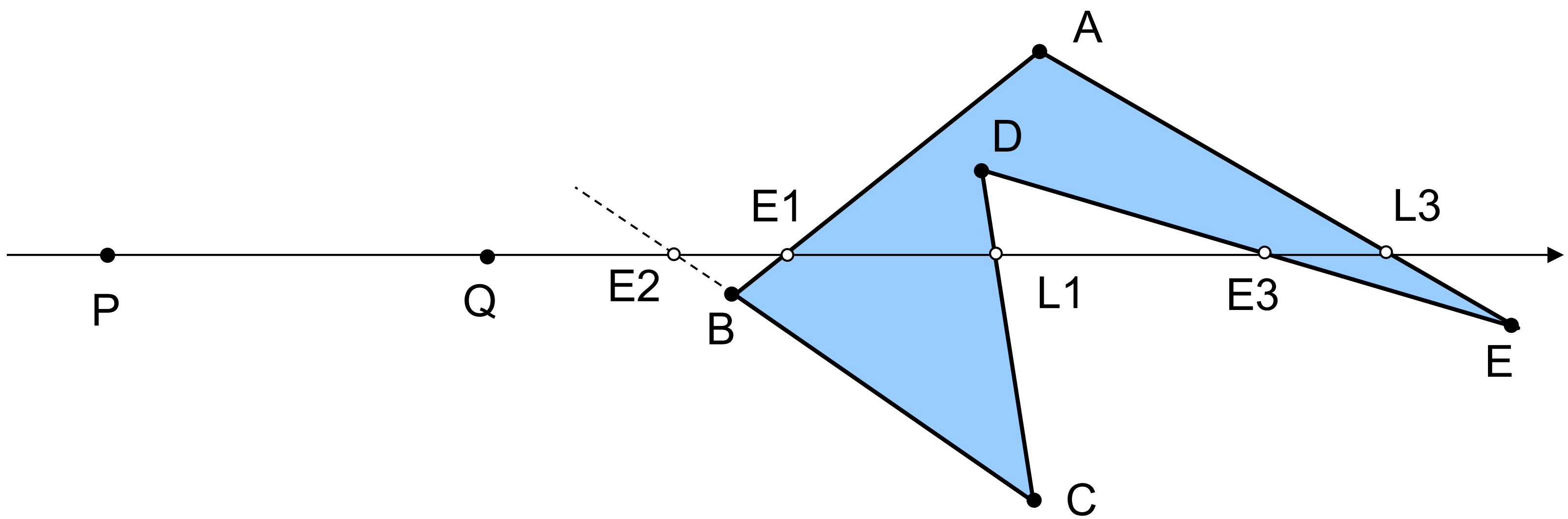
3. Compute $\max(t_{near})$ and $\min(t_{far})$;

4. If $\max(t_{near}) < \min(t_{far})$, then PQ intersects the polygon;

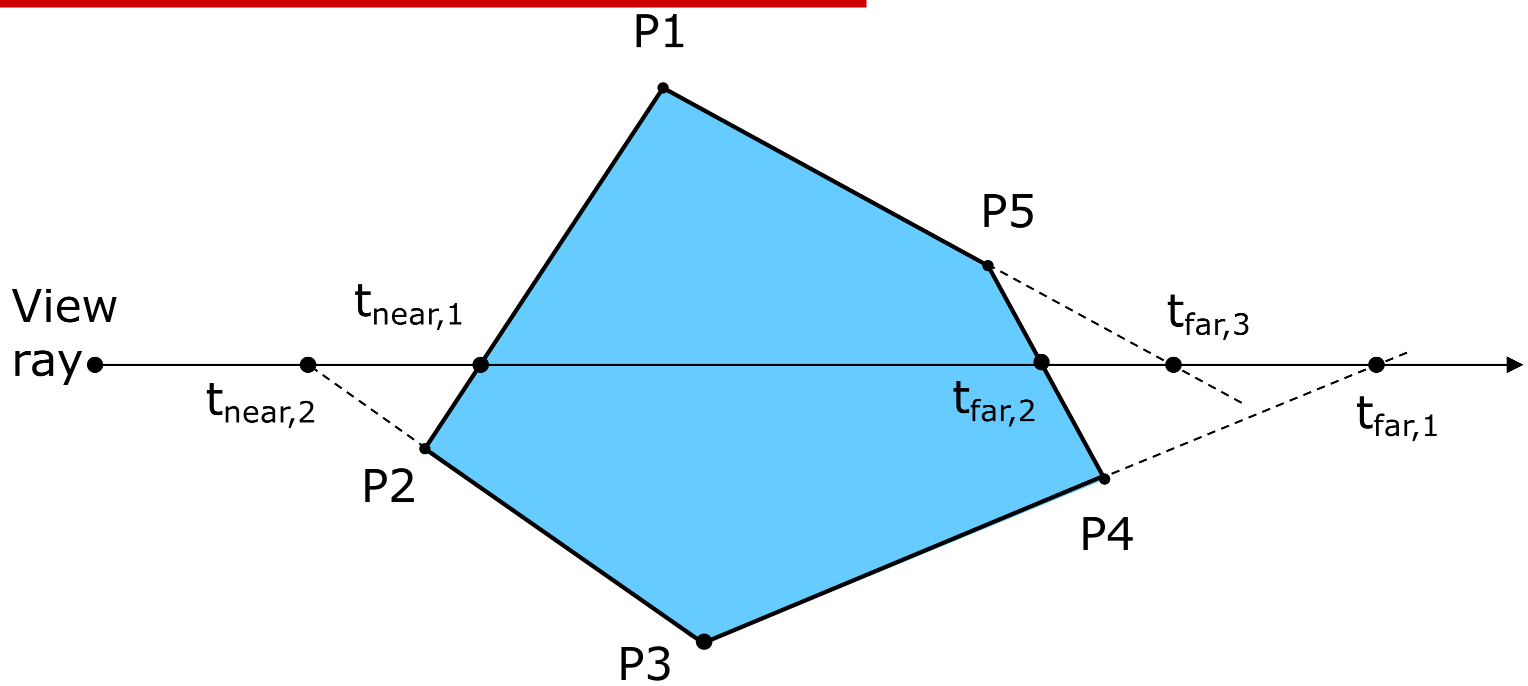


Line to polygon relationship

- Convex and concave polygons
- Triangulation
- Convex polygons
  - Analyses the line to edges relationships
  - Cyrus-Beck algorithm
    (for convex polygon only!)



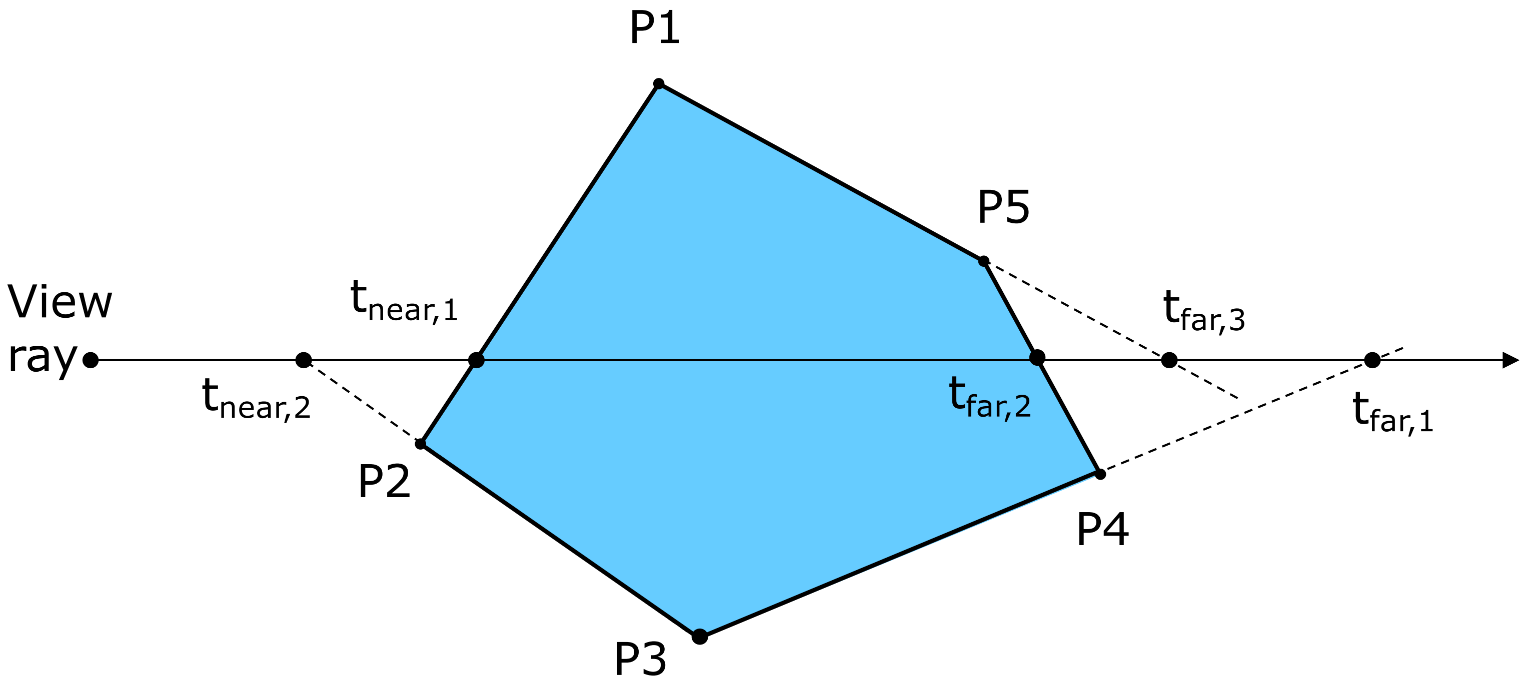
Intersection: line–convex polyhedron



Cyrus-Beck algorithm:

1. Classify the face polygons by front face and back face;
2. For each face compute intersection with view ray;
3. Compute max(t_{near}) and min(t_{far});
4. If max(t_{near}) < min(t_{far}) there is intersection between ray and the convex 3D object;

Intersection: line–convex polyhedron



Cyrus-Beck algorithm:

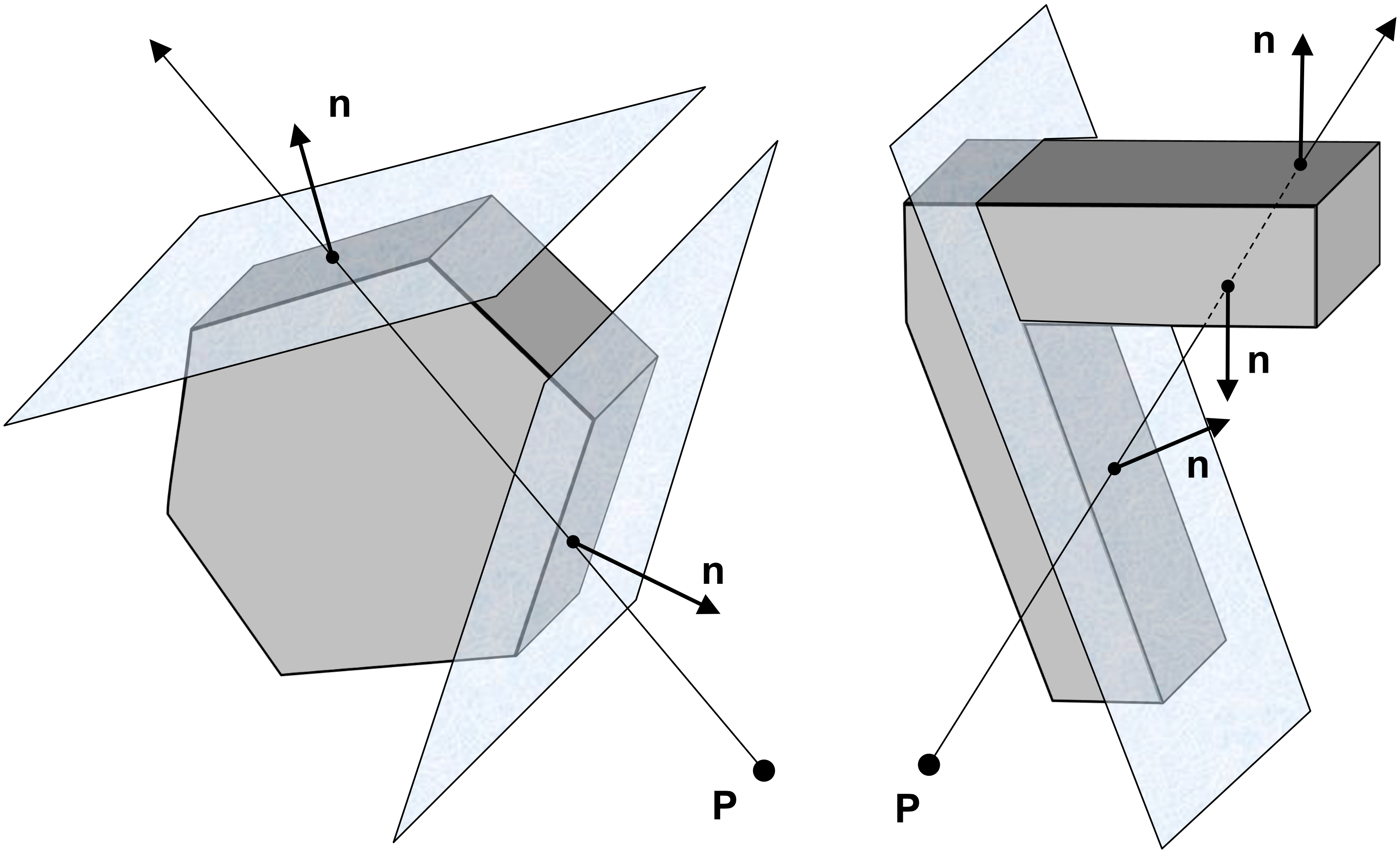
Initialize t_{near} to large negative value, t_{far} to large positive value

If (plane is back-facing) and ($t < t_{far}$) then $t_{far} = t$

If (plane is front-facing) and ($t > t_{near}$) then $t_{near} = t$

If ($t_{near} > t_{far}$) then (exit – ray misses)

Line and 3D object intersection



Questions and proposed problems

1. What is the motivation of using in graphics systems of one instead of another coordinate system: left-hand or right-hand?
2. What could be the meaning of the coordinate axes?
3. What is the main difference between the parametric equations of the line: $P = P1 + t(P2-P1)$, and $P = (1-t)P1 + t\,P2$?
4. Explain the using of the parametric equation of a line for modelling the viewing ray.
5. Explain an approach to determine if an object lies in front side, backside, and aside of the viewer.
6. Let be two lines d and f in plane. Explain an approach to determine if the two lines are in one of the following cases (a) intersects each other; (b) are orthogonal; (c) form a given angle (e.g. 35^o).
7. Let be a wall and a person. Determine if the person lies in front of the wall.
8. Let be a pentagonal room and a person. Determine if the person lies inside the room. Determine if the person lies outside the room.

Questions and proposed problems

9. Let be a person and a pentagonal building. Explain an approach to determine the visible and non-visible walls of the building.
10. Let be a triangle in the plane. Explain an approach to compute the distance from a point P to the triangle.
11. Let be a pentagonal building. Explain an approach to compute the distance of a person to the building.
12. Let be two triangles in the plane. Explain an approach to compute the distance between the two triangles.
13. Let be a pentagon in the plane, given by its vertices in the counterclockwise direction. Compute the normal vectors to the edges.
14. Let be the polygon in the plane, given by its vertices V1, V2, …, V10. Explain an approach to compute the area of the polygon.
15. Let be two vectors AB and BC. Explain an approach to determine if the vector BC goes relative to AB (a) forward; (b) backward; (c) left; (d) right.

Questions and proposed problems

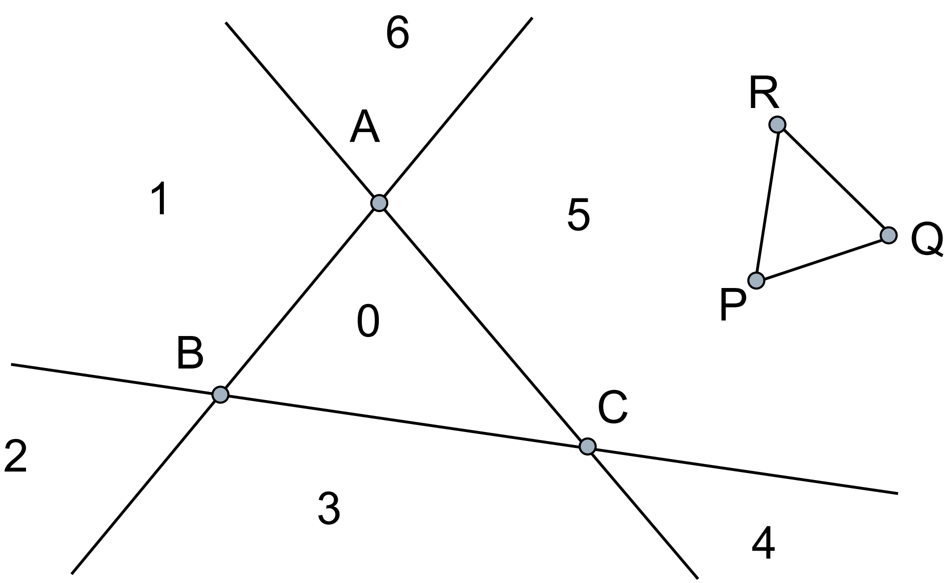
16. Let be a polygon in the plane, given by its vertices V1, V2, …, V5. Explain an approach to determine if the polygon is convex or concave.
17. Let be a convex polygon in the plane, given by its vertices V1, V2, …, V5. Explain an approach to determine if the point P lies outside the polygon.
18. Let be a concave polygon in the plane, given by its vertices V1, V2, …, V5. Explain an approach to determine if the point P lies inside the polygon.
19. Let be a convex polygon in the plane, given by its vertices V1, V2, …, V5. Explain an approach to determine if the segment AB lies outside the polygon.
20. Let be a convex polygon in the plane, given by its vertices V1, V2, …, V5. Explain an approach to determine if the triangle ABC lies outside the polygon.
21. Let be two triangles in the plane. Explain an approach to compute the relationship between the two triangles: (a) intersected; (b) disjoint; (c) included.

Questions and proposed problems

22. Let be a segment AB and a point P in the plane. Compute the distance between the point P and the line AB. How do you compute the distance between the point P and the segment AB?

23. Let be a segment AB and a point P in 3D space. Compute the distance between the point P and the line AB. How do you compute the distance between the point P and the segment AB?

24. Let be the triangle ABC in the following figure. Determine if the triangle PQR lies inside one of the regions: 2, 3 or 5.



Questions and proposed problems

25. Let be an oriented segment AB. Explain an approach to determine if a point P lies on left side or right side of the segment.
26. Let be a line in the plane, given by the equation $y = 0.5x + 7$. Determine if a point P(3,5) lies below or above the line.
27. Let be the linear function $F(x,y)=0.5x - y + 7$. Determine if the points below the line $y=0.5x+7$ give positive or negative values.
28. Let be the function $F(x,y)=ax+by+c$. Explain if the points above the line $ax+by+c=0$ give positive or negative values.
29. Let be the function $F(x,y,z)=Ax+By+Cz+D$. Explain if the points in front of the plain $Ax+By+Cz+D=0$ give positive or negative values.
30. Let be the pentagonal polygon in the plane, and the viewer P is looking toward the point Q. Explain an approach to determine if the viewer also looks toward the polygon.
31. Let be a cube in the 3D space, and the viewer P is looking toward the point Q. Explain an approach to determine if the viewer also looks toward the cube.

Questions and proposed problems

32. Let be a 3D polygonal object, where its surface is modelled by triangles. Explain an approach to determine the position of a point P relative to the object, if the 3D object is (a) convex; (b) concave.
33. Let be two cubes of any dimension and orientation. Explain an approach to compute the distance between the two cubes.
34. Let be two cubes of any dimension and orientation. Explain an approach to determine the relative position of the two cubes as: (a) disjoint; (b) intersected; (c) surrounded.