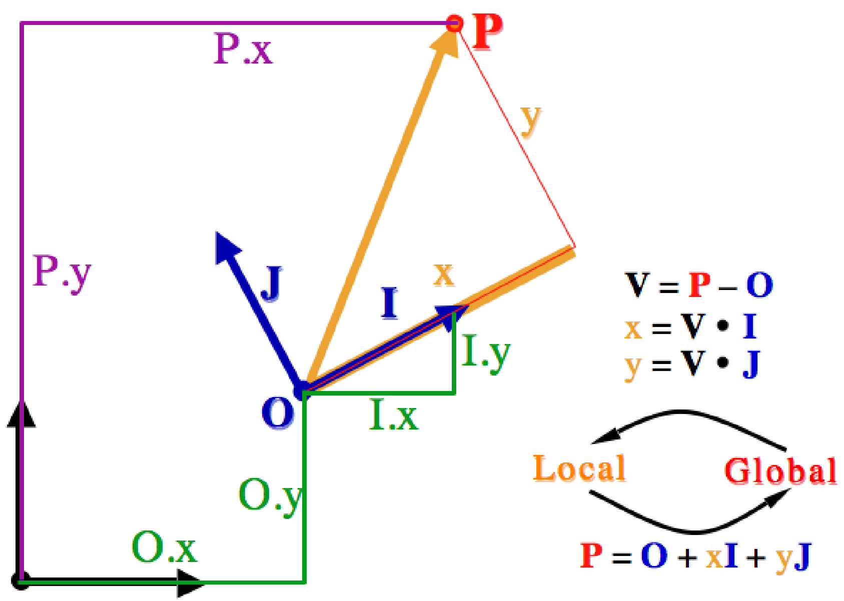
2D geometry

MAGIC Lab
Modeling, Animation, Graphic, Interaction & Compression

- Vectors, points, dot-product
- Coordinates, transformations
- Lines, edges, intersections
- Triangles
- Circles

Updated Sept 2010



Motivation

- The algorithms of Computer Graphics, Video Games, and Digital Animations are about modeling and processing geometric descriptions of shapes, animations, and light paths.

Vectors (Linear Algebra)

- A vector is defined by a **direction** and a…?

  magnitude (also called norm or length)
- A vector may be used to represent what?

  displacement, force…
- What is a **unit** vector?

  a vector with **magnitude 1** (measured in chosen unit)
- What does a **unit** vector represent?

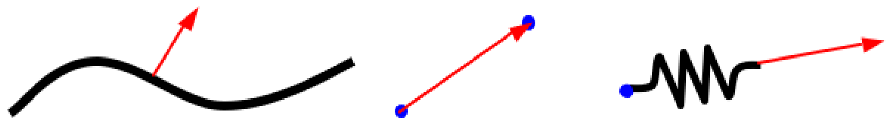
  a **direction** (tangent, outward normal)
- What is sV, where s is a scalar?

  a vector with direction of V, but norm **scaled** by s
- What is U+V?

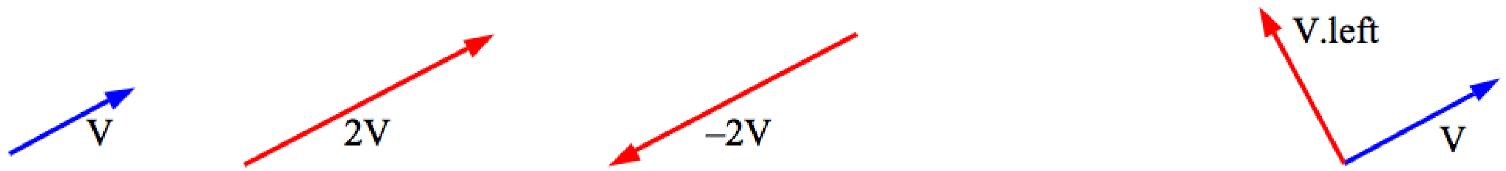
  the **sum** of the displacements of U and of V

Unary vector operators

- Defines a direction (and displacement magnitude)
- Used to represent a basis vector, tangent, normal, force…

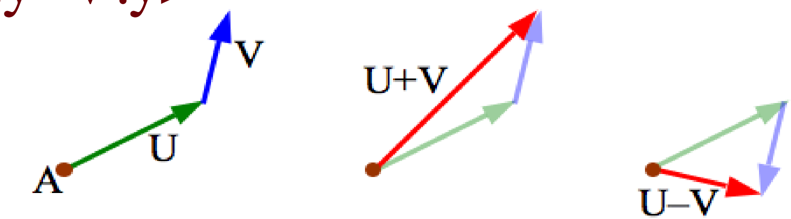


- **Coordinates**: $V = < V.x , V.y >$
- **Opposite**: $–V = < –V.x , –V.y >$
- **Norm** (or length, or magnitude): $n(V)= \sqrt{(V.x^2+V.y^2)}$
- **Null** vector: $o = < 0 , 0 >$, $n(o)=0$
- **Scaling**: $sV = < sV.x , sV.y >$, $V/s = < V.x/s, V.y/s >$
- **Direction** (unit vector): $U(V) = V/n(V)$, assume $n(V)\neq 0$
- **Rotated** 90 degrees: $R(V) = < –V.y , V.x >$

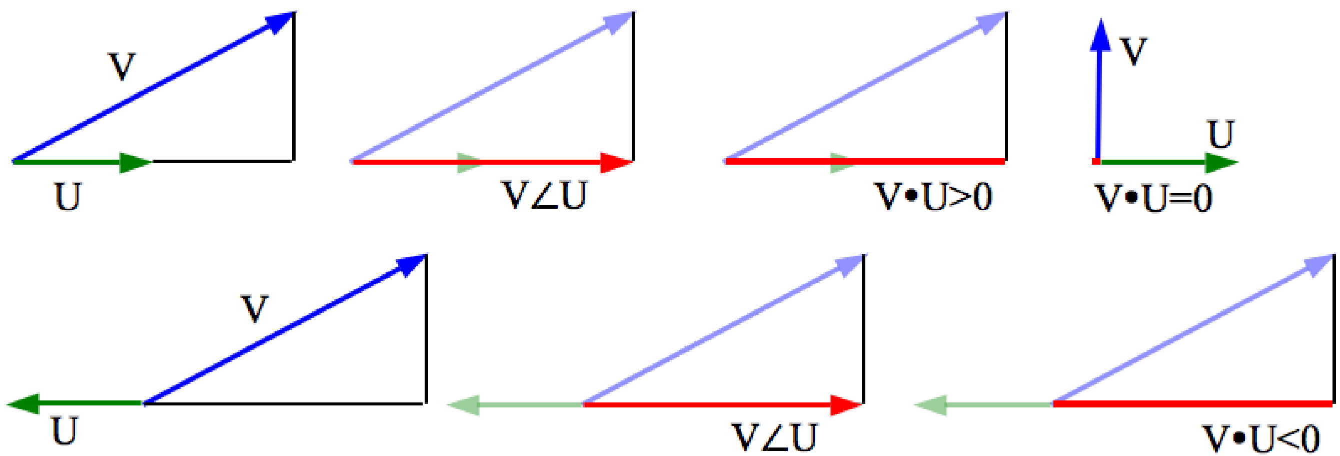


Binary vector operators

- **Sum**: U+V = < U.x+V.x , U.y+V.y>,
- **Difference**: U–V = < U.x–V.x , U.y–V.y>



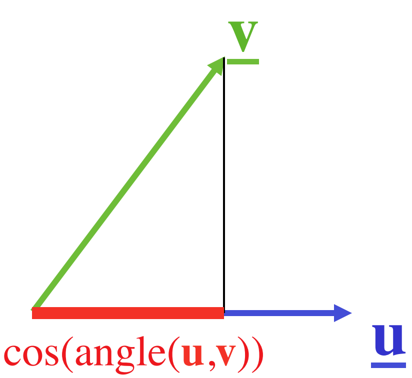
- **Dot product (scalar)**: $V \bullet U = U.xV.x + U.yV.y$
- **Norm squared**: $V^2 = V \bullet V = (n(V))^2$
- **Tangential component of V wrt U**: $V\angle U = (V \bullet U)\, U / U^2$



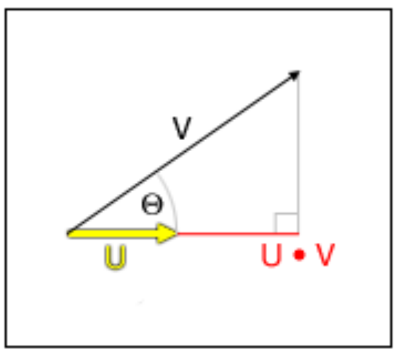
Dot Product: "Your best friend"

$U \bullet V = \|U\| \cdot \|V\| \cdot \cos(\text{angle}(U,V))$

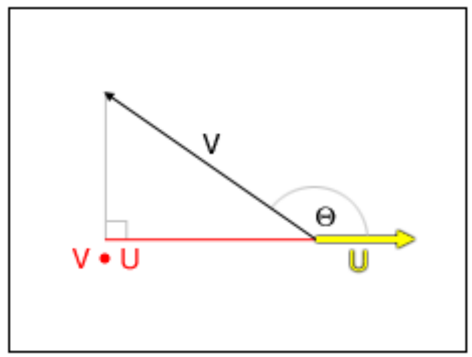
- $U \bullet V$ is a scalar.
- if U and V are orthogonal $\Rightarrow$ then $U \bullet V == 0$
- $U \bullet V == 0 \quad \Rightarrow \quad U == 0$ or $V == 0$ or (U and V are orthogonal)
- $U \bullet V$ is positive if the angle between U and V is less than 90^o
- $U \bullet V = V \bullet U$, because: $\cos(a) = \cos(-a)$.
- $\|u\| = \|v\| = 1 \quad \Rightarrow \quad u \bullet v = \cos(\text{angle}(u,v))$ # unit vectors
- $V \bullet u$ = signed length of projection of V onto the direction (unit vector)



cos(angle(**u**,**v**))



V•U = U•V > 0 here



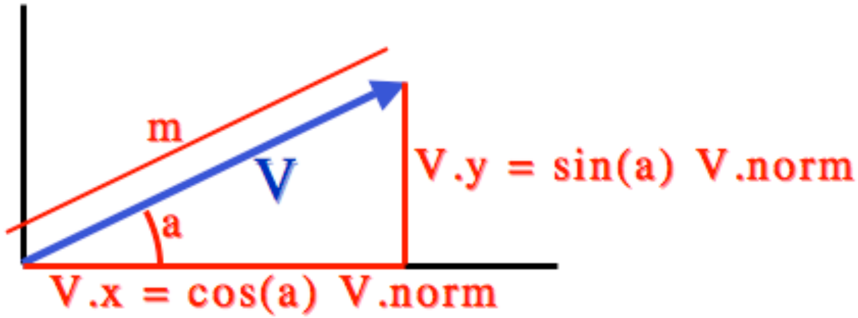
V•U = U•V < 0 here

Dot product quiz

- What does the dot product V•U measure when U is unit?
  The projected displacement of V onto U
- What is V•U equal to when U and V are unit?
  cos(angle(U,V))
- What is V•U equal to for general U and V?
  cos(angle(U,V)) n(V) n(U)
- When is V•U=0?
  n(U)=0 OR n(V)=0 OR U and V are orthogonal
- When is V•U>0?
  the angle between them is less than 90°
- How to compute V•U?
  U.xV.x+U.yV.y
- What is V^2?
  $V^2 = V \bullet V = sq(n(V))$

Angles between vectors

- **Polar** coordinates of a vector: (m=n(V), a=atan2(V.y,V.x))
  $a \in [-\pi,\pi]$

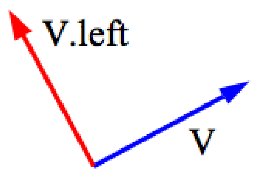


Assume V≠0, U≠0

- **Angle** between two vectors: cos(a) = V•U/(n(V)*n(U))

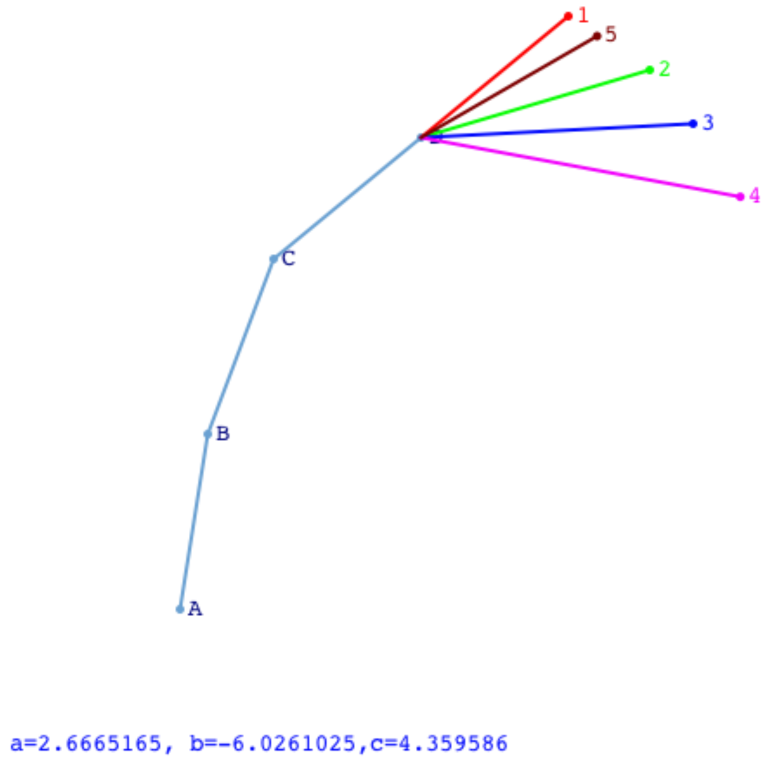
Use difference between polar coordinates to sort vectors by angle

- V and U are **orthogonal** (i.e. perpendicular) when V•U=0
  V.xU.x + V.yU.y = 0
- V and U are **parallel** when V•R(U)=0
  V.xU.y = V.yU.x



Application: Motion prediction

- Based on last 4 positions, how to predict the next one?



Change of orthonormal basis — important!

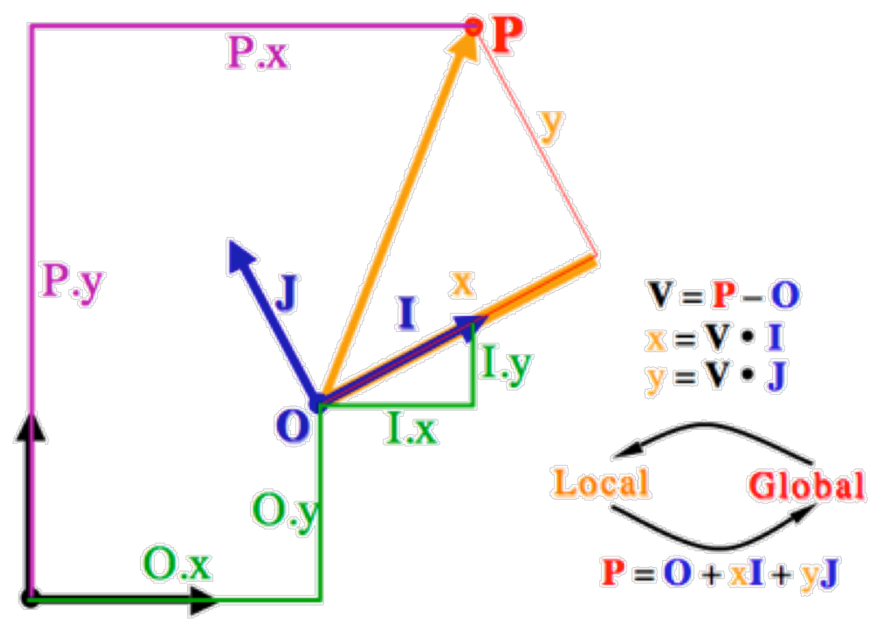
- A **basis** is two non-parallel vectors {I,J}. It is **orthonormal** if I^2==1 and J=R(I)
- What is the vector with coordinates <x,y> in basis {I,J}?

  xI+yJ
- What is the vector <x,y> if we do not specify a basis?

  xX+yY, X is the horizontal, Y the vertical unit vector
- What are the coordinates <x,y>

of V in orthonornal basis (I,J)?

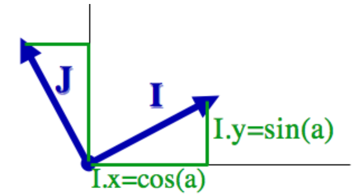
x=V•I, y=V•J



Rotating a vector

- What is the **rotation** (I,J) of basis (X,Y) by angle a?

  $(I,J) = (< \cos(a) , \sin(a) > , < - \sin(a) , \cos(a) >)$



- How to rotate vector <x,y> by angle a?

  compute (I,J) as above, then compute xI+yJ

- What are the **coordinates** of V rotated by angle a?

  $V.rotate(a) =$

  $= V.x < \cos(a) , \sin(a) > + V.y < - \sin(a) , \cos(a) >$

  $= < \cos(a)\, V.x - \sin(a)\, V.y , \sin(a)\, V.x + \cos(a)\, V.y >$

- What is the **matrix form** of this **rotation**?

$$\begin{pmatrix} \cos(a)\, V.x - \sin(a)\, V.y \\ \sin(a)\, V.x + \cos(a)\, V.y \end{pmatrix} = \begin{pmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{pmatrix} \begin{pmatrix} V.x \\ V.y \end{pmatrix}$$

Vector coordinates quiz

- When are two vectors **orthogonal** (to each other)

  When the angle between their directions is ±90°
- What is an **orthonormal basis**?

  two orthogonal unit vectors (I,J)
- What is the vector with **coordinates** <V.x,V.y> in (I,J)?

  V.x I + V.y J
- What are the coordinates of vector **combination** U+sV?

  < U.x+sV.x, U.y+sV.y >
- What is the **norm** of V?

  n(V) = V.norm = sqrt(V.x2+V.y2) (always ≥ 0)
- What are the coordinates of V **rotated** by 90°

  R(V)= < –V.y , V.x >, *verify that V•R(V) = 0*

Radial coordinates and conversions

- What are the **radial** coordinates {r,a} of V?

  { V.norm, atan2(V.y,V.x) }

- What are the **Cartesian** coordinates of {r,a} ?

  < r cos(a) , r sin(a) >

Reflection: used in collision and ray tracing

Consider a **line** L with **tangent** direction T

- What is the **normal** N to L?

  N=R(T)

- What is the **normal component** of V?

  (V•N)N (it is a vector)

- What is the **tangent component** of V?

  (V•T)T

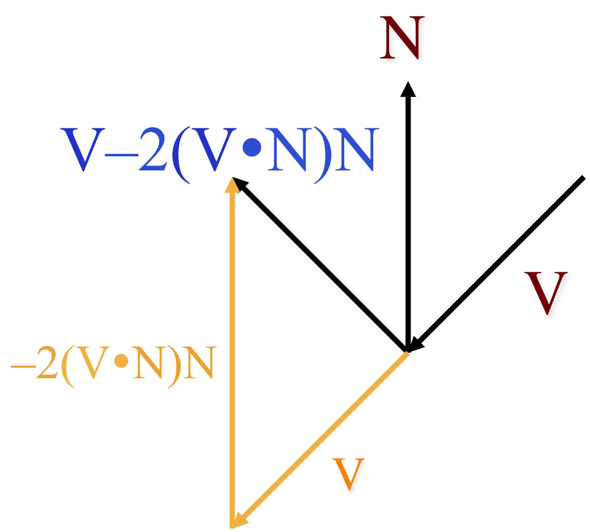
- What is the **reflection** of V on L?

  (V•T)T–(V•N)N (reverse the normal component)

- What is the **reflection** of V on L (simpler form not using T)?

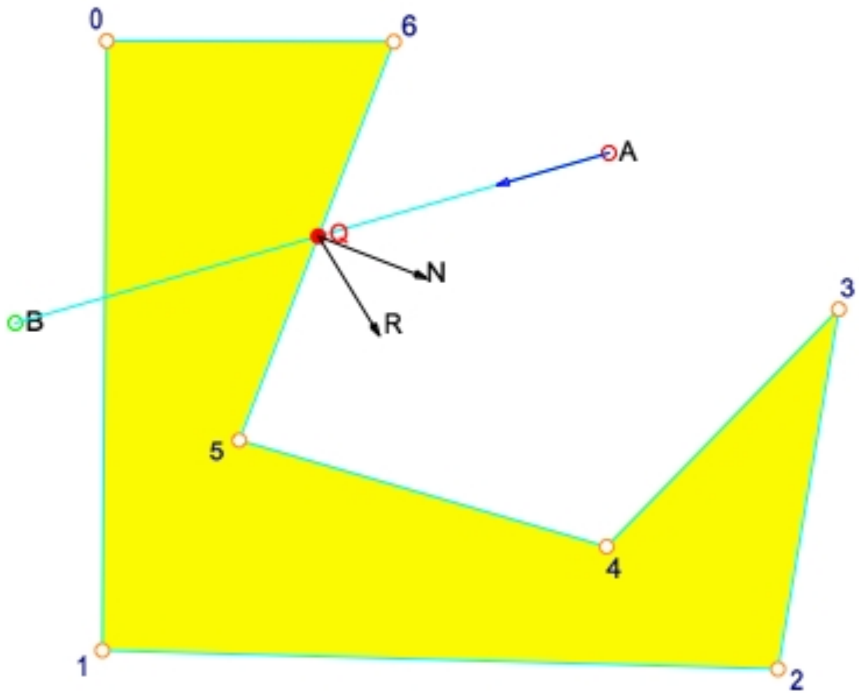
  V–2(V•N)N (cancel, then subtract normal component)

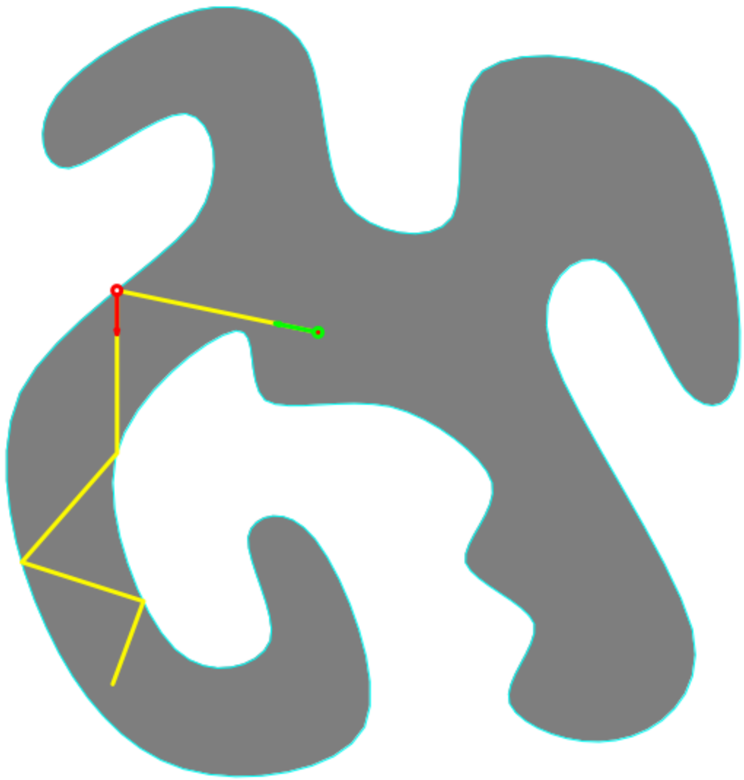
  This one works in 3D too (where T is not defined)



Appliction of reflection: Photon tracing

- Trace the path of a photon as it bounces off mirror surfaces (or mirror edges for a planar version)





Cross-product: "Your other best friend"

- The **cross-product** U×V of two vectors is a **vector** that is orthogonal to both, U and V and has for magnitude the product of their lengths and of the sine of their angle

$$\|U\times V\| = \|U\| \; \|V\| \; \sin(\text{angle}(U,V))$$

- Hence, the cross product of two vectors in the plane of the screen is a vector orthogonal to the screen.

OPERATOR OVERLOADING FOR 2D CONSTRUCTIONS

- When dealing with 2D constructions, we define U×V as a **scalar**:

$$U\times V = \|U\| \; \|V\| \; \sin(\text{angle}(U,V))$$

The 2D cross product is the z-component of the 3D cross-product.

Verify that in 2D: $U\times V = U\bullet R(V)$

Change of **arbitrary** basis

- What is the vector with coordinates <x,y> in basis {I,J}?

  xI+yJ

- What are the coordinates <x,y> of V in basis (I,J)?

  Solve V=xI+yJ,

  a system of two linear equations with variables x and y

  (two vectors are equal is their x and their y coordinates are)

The solution (using Cramers rule):

$x = V \times J \,/\, I \times J$ and $y = V \times I \,/\, J \times I$

Proof

$V = xI + yJ \Rightarrow V \times J = xI \times J + yJ \times J \Rightarrow V \times J = xI \times J \Rightarrow V \times J \,/\, I \times J = x$

Points (Affine Algebra)

- **Define a location**
- **Coordinates** P = (P.x,P.y)
- Given **origin** O: P is defined by vector OP=P–G=<P.x,P.y>

- **Subtraction**: PQ = Q–P = <Q.x–P.x,Q.y–P.y>
- **Translation** (add vector): Q = P+V = (P.x+V.x , P.y+V.y)

Incorrect but convenient notation:

- **Average**: (P+Q)/2 = ((P.x+Q.x)/2 , (P.y+Q.y)/2)
  correct form: P+PQ/2
- **Weighted average**: $\sum w_i P_i$, with $\sum w_i = 1$
  correct form: $O+\sum w_j OP_j$

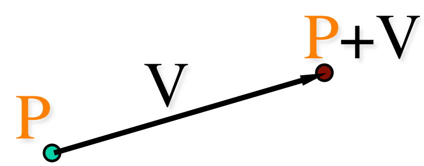
Practice with points

- What does a **point** represent?

  a location

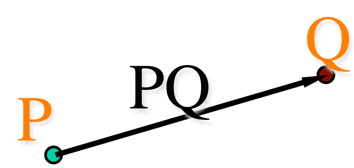
- What is P+V?

  P translated by displacement V



- What is the **displacement** from P to Q?

  PQ = Q – P (**vector**)

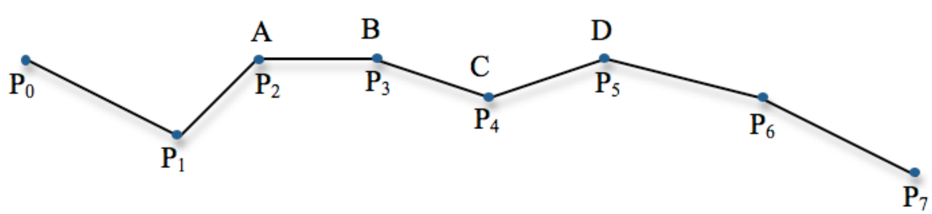


- What is the midpoint between P and Q?

  P + 0.5PQ (also written P+PQ/2 or wrongly (P+Q)/2)

- What is the center of mass G of triangle area (A,B,C)?

  G=(A+B+C)/3, properly written G=A+(AB+AC)/3



vector ≠ point

	Vectors U,V,W	Points P,Q
Meaning	displacement	location
Translation	forbidden	P+V
Addition	U+V	forbidden
Subtraction	W=U–V	V (= PQ) = Q–P
Dot product	s=U•V	forbidden
Cross product	W=U×V	forbidden

Orientation and point-triangle inclusion

- When is the sequence A,B,C a **left turn**?

  cw(A,B,C) = AB×BC>0

  (also = R(AB)•BC>0 and also AB×AC>0 …)

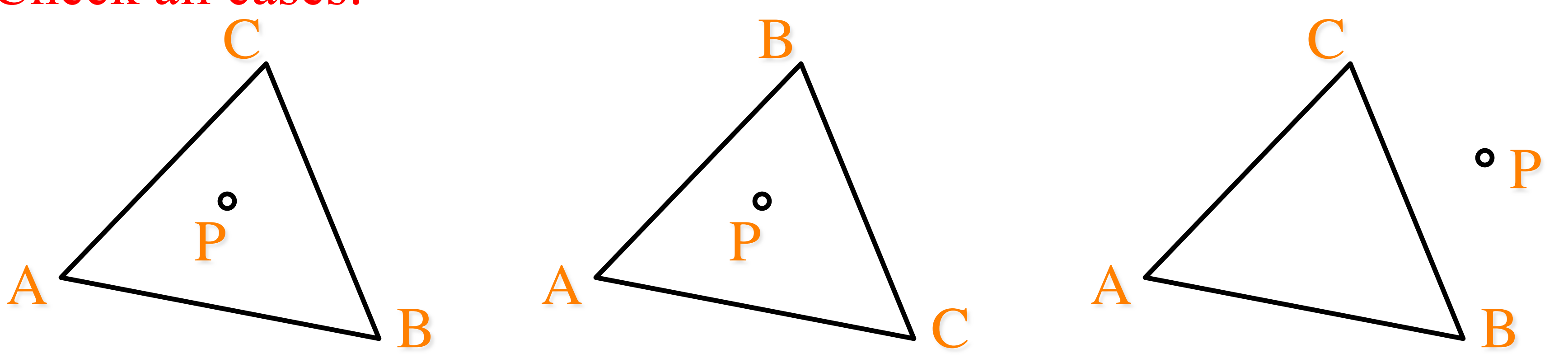
- When is triangle(A,B,C) **cw** (clockwise)?

  cw(A,B,C)

- When is point **P in triangle**(A,B,C)?

  cw(A,B,P) == cw(B,C,P) && cw(A,B,P) == cw(C,A,P)

Check all cases:



Edge intersection test

- Linear parametric expression of the **point** P(s) **on edge**(A,B)?

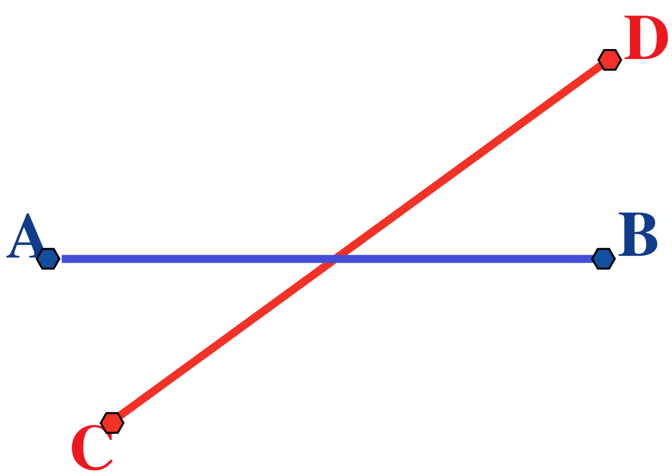
  P(s) = A+sAB (also written (1–s)A+sB) for s in [0,1]

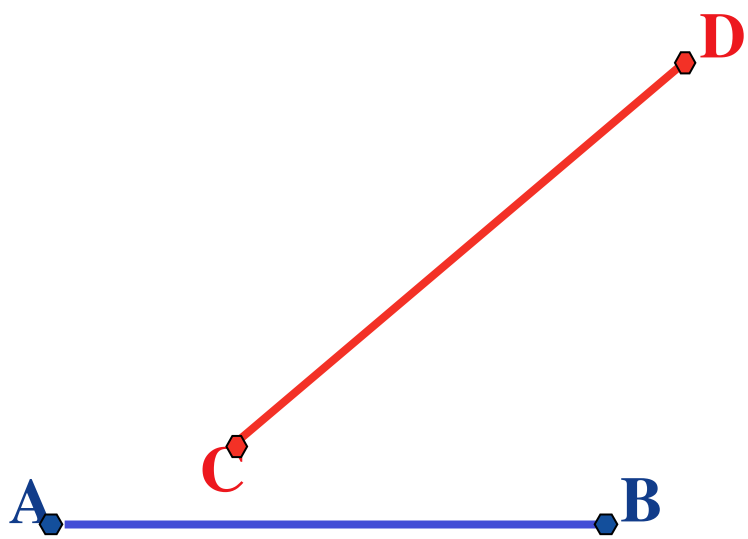
  my Processing implementation is called L(A,s,B) or I(A,s,B)

- When do edge(A,B) and edge(C,D) intersect?

  cw(A,B,C) != cw(A,B,D)) && (cw(C,D,A) != cw(C,D,B)

  (special cases of collinear triplets require additional tests)





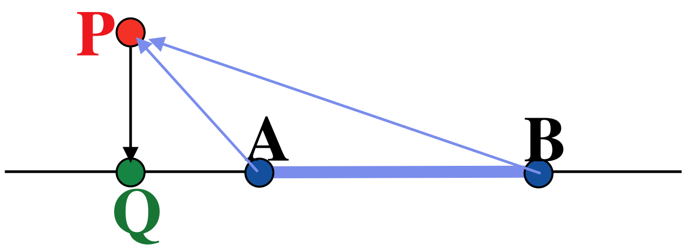
Normal projection on edge

- When does the projection Q of point P onto Line(A,B) fall between A and B

  i.e.: when does P project onto edge(A,B)?

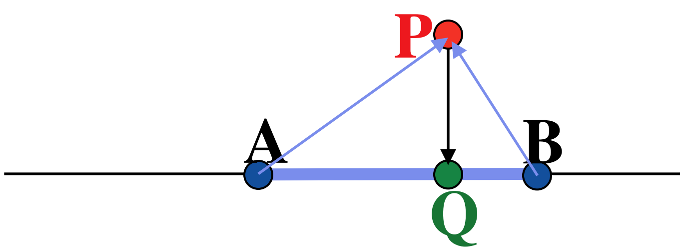
  when:

$0 \leq AP \bullet AB \leq AB \bullet AB$

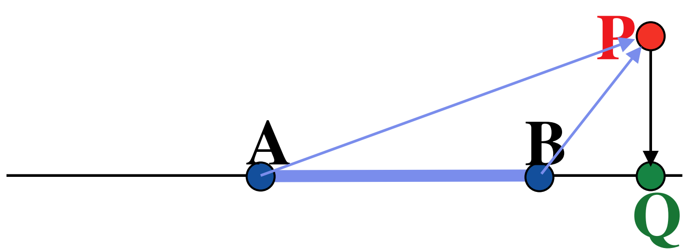


or equivalently, when:

$0 \leq AP \bullet AB \;\&\&\; 0 \leq BP \bullet BA$



explain why:



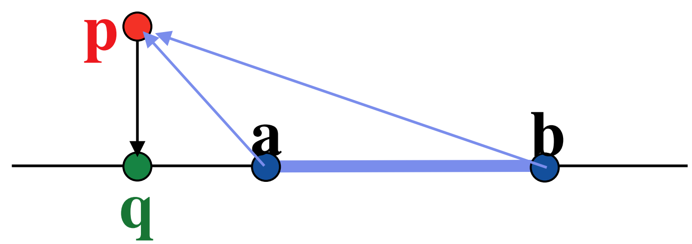
PinE(point,edge): Point-in-edge test

When is point P in edge(a,b)?

when $|ab \times ap| < \varepsilon \, \|ab\|$ && $ab \cdot ap > 0$ && $ba \cdot bp > 0$

PROOF:

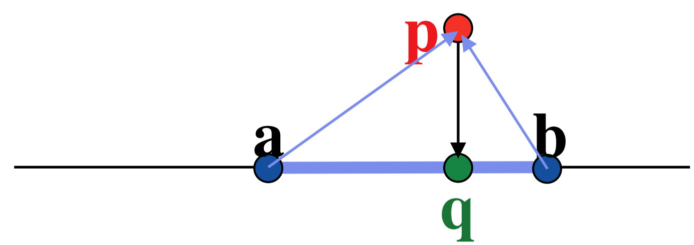
q = projection of p onto the line (a,b)



The **distance** $\|pq\|$ from p to q is

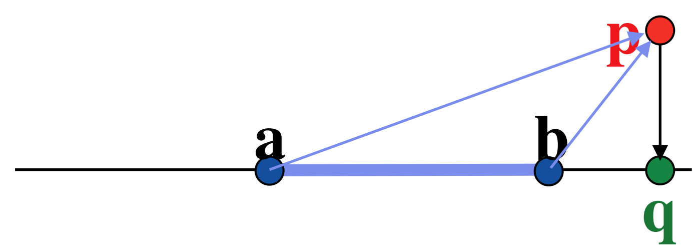
$$|ab \times ap| \; / \; \|ab\|$$

It needs to be less than a threshold **ε**



We also want the projection q to be inside edge(a,b), hence:

$ab \cdot ap > 0$ && $ba \cdot bp > 0$



Parallel lines

- When are line(P,T) and line(Q,U) **parallel**

  $T \times U == 0$

or equivalently when

  $T \bullet R(U) == 0$

Ray/line intersection

- What is the expression of **point on ray**(S,T)?

  P(t) = S+tT, ray starts at S and has tangent T
- What is the constraint for **point** P to be **on line**(Q,N)?

  QP•N=0, normal component of vector QP is zero
- What is the intersection X of ray(S,T) with line(Q,N)?

  X = P(t) = S+tT, with t defined as the solution of

  QP(t)•N=0
- How to compute parameter t for the intersection above?

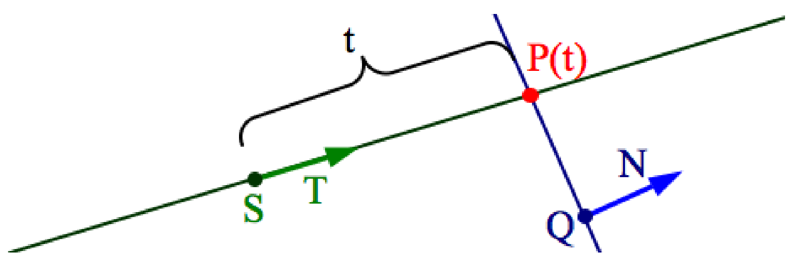
  (P(t)–Q)•N=0

  (S+tT–Q)•N=0

  (QS+tT)•N=0

  QS•N + tT•N=0 , distributing • over +

  t = – (QS•N) / (T•N)



Lines intersections

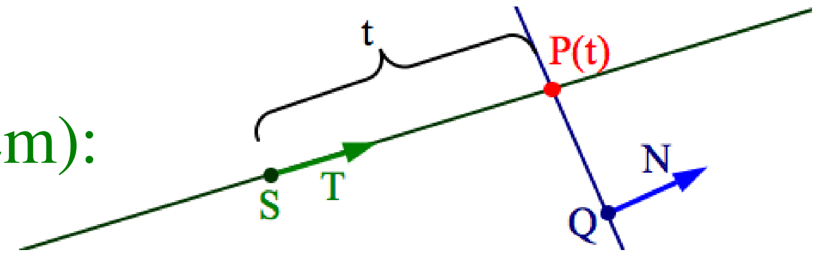
Two useful representations of a line:

- **Parametric** form, LineParametric(S,T): P(t)=S+tT
- **Implicit** form, LineImplicit(Q,N): QP•N=0

LineParametric(S,T) = LineImplicit(S,R(T))

Intersection: LineParametric(S,T) ∩ LineImplicit(Q,N)

- Substitute P(t)=S+tT for P into QP•N=0
- Solve for the parameter value: t=(SQ•N)/(T•N)
- Substitute back: P(t) = S + (SQ•N)/(T•N) T



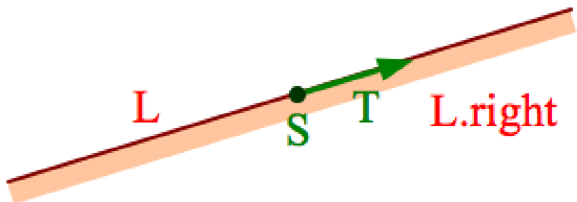
Other approaches (solve linear system):

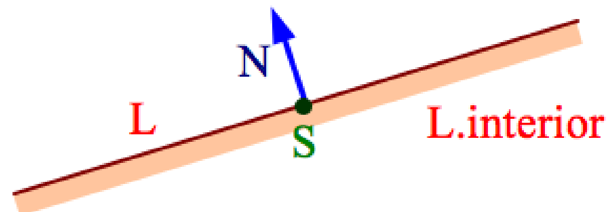
S+tT==S'+uT' or QP•N==0

Q'P•N'==0

Half-space

- Linear **half-space** H(S,N) ={P : SP•N < 0 }
  set of points P such that they are "behind" S with respect to N
  N is the **outward normal** to the half-space
  H(S,N) does not contain line {P : SP•N==0 } (is **topologically open**)





- L = line(S,T) (through S with tangent T)
- L.right = H(S,R(T))
  - N=R(T) is the **outward normal to the half-space**

  L.right is shown on the left in a Processing canvas (Y goes down)
  L.right does not contain L (**topologically open**)

Transformations

- **Translation** of P=(x,y) by vector V: $T_V(P) = P+V$
- **Rotation** : $R_a(P) = (x \cos(a) – y \sin(a) , x \sin(a) + y \cos (a))$
  by angle a **around the origin**
- **Composition**: $T_V(R_a(P))$, rotates by a, then translates by V
- **Translations commute**: $T_U(T_V(P))=T_V(T_U(P)) =T_{U+V}(P)$
- **2D rotations commute**: $R_b(R_a(P))=R_a(R_b(P))=R_{a+b}(P)$
- **Rotations/translations do not commute**: $T_V(R_a(P))\neq R_a(T_V(P))$
- **Canonical** representation of compositions of transformations:
  Want to represent $\ldots T_W(R_c(T_U(R_b(P))\ldots$ as $T_V(R_a(P))$
  How to compute V and a?
  How to apply it to points and vectors?
  Answer: represent a composed transformation by a **coordinate system**

Coordinate system ("frame")

- **Coordinate system** [I,J,O]

  O is the origin of the coordinate system (a translation vector)

  {I,J} is an **ortho-normal basis**: I.norm=1, J=R(I)

  {I,J} captures the rotation part of the transformation
- Given **local coordinates** (x,y) of P in [I,J,O]

  P=O+xI+yJ, "start at O, move by x along I, move by y along J"
- **Given P, O, I, J, compute (x,y)**

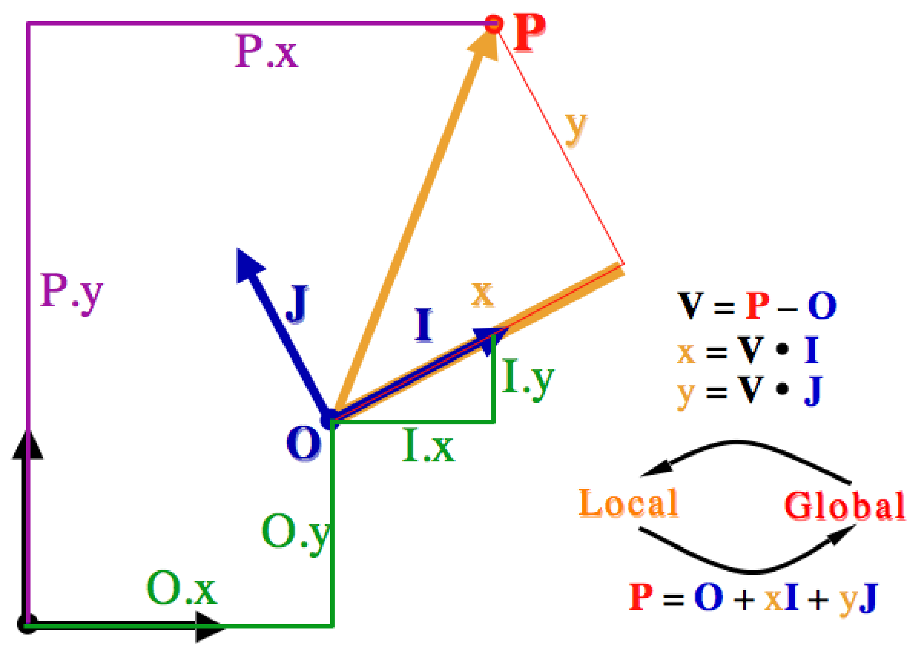
  x=OP•I, y=OP•J

  proof: OP•I=xI•I+yJ•I=xI•I
- For a **vector V**, **no translation**

  Local coordinates <x,y>

  Vector V = xI+yJ

  Inverse: x=V•I, y=V•J



Rotation around center C

- What is the **result** P’ of rotatin a point P by angle a around C?

  Rotate vector CP and add it to C

  P’=C+CP.rotate(a)

  Hence: P’=C + CP.x <cos a , sin a> + CP.y <sin a , cos a>

  This can be executed in Processing (and OpenGL) as 3 transforms:

  - Translate by CO (now C is at the origin and P is at O+CP)
  - Rotate by angle a (rotates CP around origin: O+CP.rotate(a))
  - Translate by OC (to put things back: O+CP.rotate(a)+OC)

A different (faster?) implementation

$(\cos(a)\, P.x - \sin(a)\, P.y,\ \sin(a)\, P.x + \cos(a)\, P.y)$

may also be implemented as:

$P.x \mathrel{-}= \tan(a/2)\, P.y$

$P.y \mathrel{+}= \sin(a)\, P.x$

$P.x \mathrel{-}= \tan(a/2)\, P.y$

- *Which one is it faster to compute (this or the matrix form)?*

For animation, or to trace a circle*:*

- *pre-compute tan(a/2) and sin(a)*
- *at each frame,*

  update P.x and P.y

  add displacement OC if desired before rendering

Practice with Transforms

- What is the **translation of point** P by displacement V?

  P+V
- What is the **translation of vector** U by displacement V?

  U (vectors do not change by translation)
- What is the rotation (around origin) of point P by angle a?

  same as O + rotation of OP

  (cos(a) P.x– sin(a) P.y, sin(a) P.x + cos (a) P.y)

- What is the matrix form of this rotation?

$$\begin{pmatrix} \cos(a)\, P.x - \sin(a)\, P.y \\ \sin(a)\, P.x + \cos(a)\, P.y \end{pmatrix} = \begin{pmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{pmatrix} \begin{pmatrix} P.x \\ P.y \end{pmatrix}$$

Change of frame

Let (x_1,y_1) be the coordinates of P in $[I_1\ J_1\ O_1]$

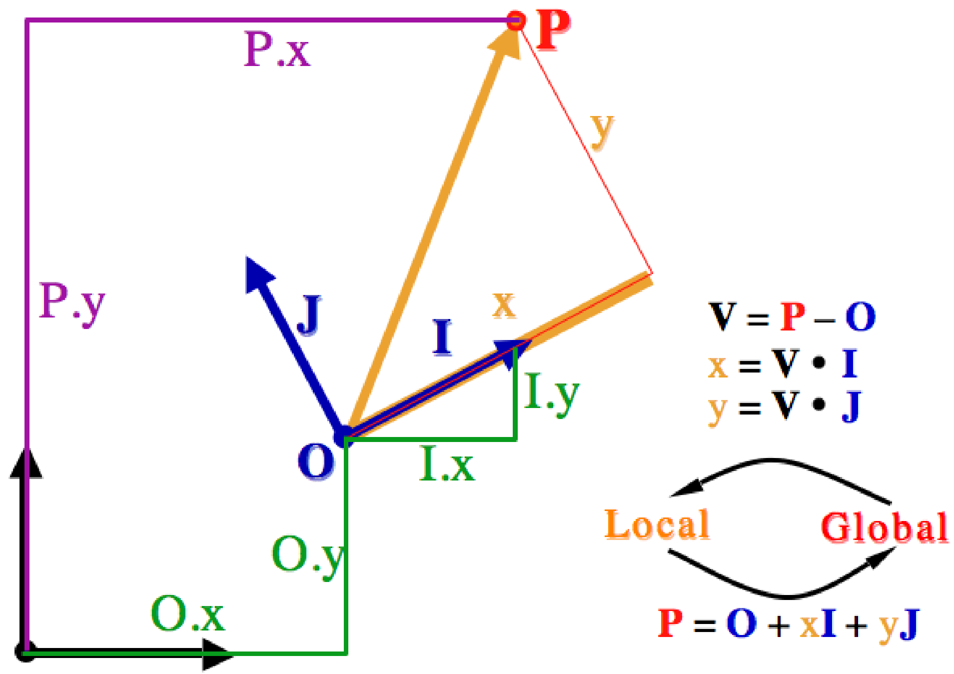
What are the coordinates (x_2,y_2) of P in $[I_2\ J_2\ O_2]$?

$P = O_1 + x_1 I_1 + y_1 J_1$ (convert local to global)

$x_2 = O_2P \bullet I_2$ (convert global to local)

$y_2 = O_2P \bullet J_2$

Applications:



What is in a rigid transform matrix?

- **Why** do we use homogeneous transforms?

  To be able to represent the cumulative effect of rotations, translations, (and scalings) into a single matrix form

$$\begin{bmatrix} P.x \\ P.y \\ 1 \end{bmatrix} = \begin{pmatrix} I.x & J.x & O.x \\ I.y & J.y & O.y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

What do the **columns of M** represent?

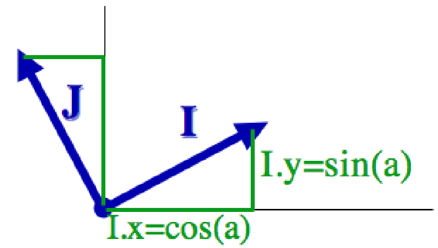
- A canonical transformation $T_O(R_a(P))$

  O = <O.x,O.y> is the translation vector

  [I,J] is the local basis (image of the global basis)

  (I J) is a 2×2 rotation matrix: I.x = J.y = cos(a), I.y = –J.x = sin(a)

  a = atan2(I.y,I.x) is the rotation angle, with a∈[–π,π]



Homogeneous matrices

- Represent a 2D coordinate system by a 3×3 homogeneous matrix
- Transform points and vectors through matrix-vector multiplication

  For point P with local coordinate (x,y) use <x,y,**1**>

  For vector V with local coordinate <x,y>, use <x,y,**0**>

- Computing the global coordinates of P from local (x,y) ones

  <P.x,P.y,1> = [**I.h J.h O.h**](x,y,1) = x**I.h** + y**J.h** + **O.h**.

$$\begin{bmatrix} P.x \\ P.y \\ 1 \end{bmatrix} = \begin{pmatrix} I.x & J.x & O.x \\ I.y & J.y & O.y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Vectors are not affected by origin (no translation)

Inverting a homogeneous matrix

- The inverse of R_a is R_{-a}
- The inverse of a rotation matrix is its **transpose**

  I.x = J.y = cos(a) remain unchanged since cos(–a) = cos(a)

  I.y = –J.x = sin(a) change sign (swap places) since sin(–a) = –sin(a)

$$\begin{bmatrix} P.x \\ P.y \\ 1 \end{bmatrix} = \begin{pmatrix} I.x & J.x & O.x \\ I.y & J.y & O.y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- The inverse of T_V is T_{-V}
- The inverse of $T_V(R_a(P))$ is $R_{-a}(T_{-V}(P))$

  It may also be computed directly as x=OP•I, y=OP•J

Examples of questions for quiz…

- What is the dot product <1,2>•<3,4>?
- What is R(<1,2>)?
- What is V^2, when V=<3,4>?
- What is the rotation by –30° of point P around point C?
- Let (x_1,y_1) be the coordinates of point P in $[\ I_1\ , J_1\ , O_1\ ]$. How would you compute its coordinates (x_2,y_2) in $[\ I_2\ , J_2\ , O_2\ ]$? (Do not use matrices, but combinations of points and vectors.)
- Point P will travel at constant velocity V. When will it hit the line passing through Q and tangent to T?

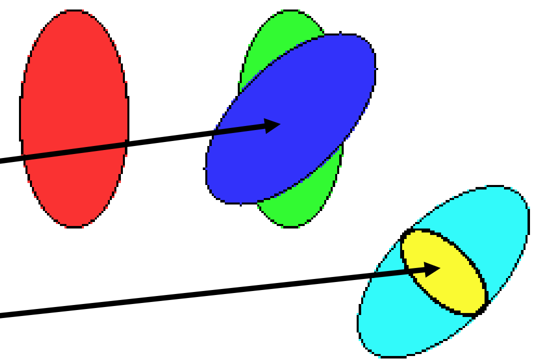
Transformations in graphics libraries

- translate(V.x,V.y);   # implement $T_V(P)$
- rotate(a);   # implements $R_a(P)$
- translate(V.x,V.y); rotate(a);   # implements $T_V(R_a(P))$

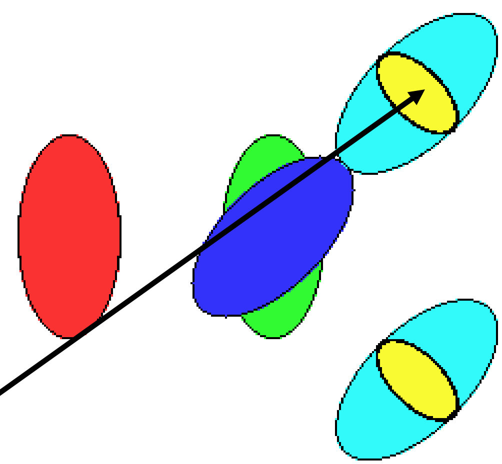
  Notice left-to-right order. Think of moving global CS.
- Scale(u,v);   # implements (uP.x,vP.y)

Push/pop operators

- {fill(red); paint();
  translate(100,0); fill(green); paint();
  rotate(PI/4);
  fill(blue); paint();
  translate(100,0); fill(cyan); paint();
  scale(1.0,0.25); fill(yellow); paint(); }



- {fill(red); paint();
  translate(100,0); fill(green); paint();
  rotate(PI/4);
  fill(blue); paint();
  pushMatrix();
  translate(100,0); fill(cyan); paint();
  scale(1.0,0.25); fill(yellow); paint();
  popMatrix();
  translate(0, -100); fill(cyan); paint();
  scale(1.0,0.25); fill(yellow); paint(); }



Circles and disks

- How to identify all points P on circle(C,r) of center C and radius r?

  $\{ P : PC^2 = r^2 \}$

- How to identify all points P in disk(C,r)?

  $\{ P : PC^2 \leq r^2 \}$

- When do disk(C_1,r_1) and disk(C_2,r_2) interfere?

  $C_1C_2^2 < (r_1 + r_2)^2$

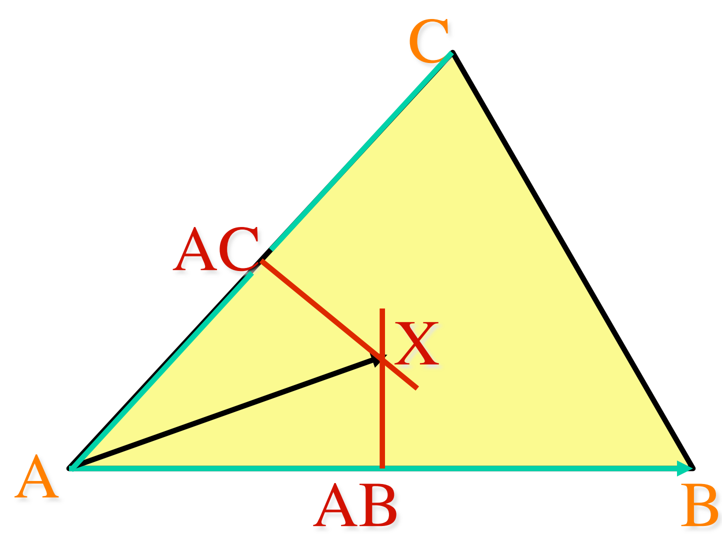
Circles and intersections

- **Circle** of center C and radius r, Circle(C,r): $\{P: CP^2=r^2\}$
  where $CP^2 = CP \bullet CP$
- **Disk** of center C and radius r, Disk(C,r): $\{P: CP^2<r^2\}$
- Disk(C_1,r_1) and Disk(C_2,r_2) **interfere** when $(C_1C_2)^2<(r_1+r_2)^2$
- The intersection of LineParametric(S,T) with Circle(C,r):
  Replace P in $CP^2 = r^2$ by S+tT
  $CP = P–C = P–S–tT = SP–tT$
  $(SP–tT)\bullet(SP–tT) = r^2$
  $(SP\bullet SP)–2(SP\bullet T)t+(T\bullet T)t^2 = r^2$
  $t^2–2(SP\bullet T)t+(SP^2–r^2)=0$
  Solve for t: real roots, t_1 and t_2, assume $t_1<t_2$
  Points S+tT when $t\in]t_1,t_2[$ are in Disk(C,r)

Circumcenter

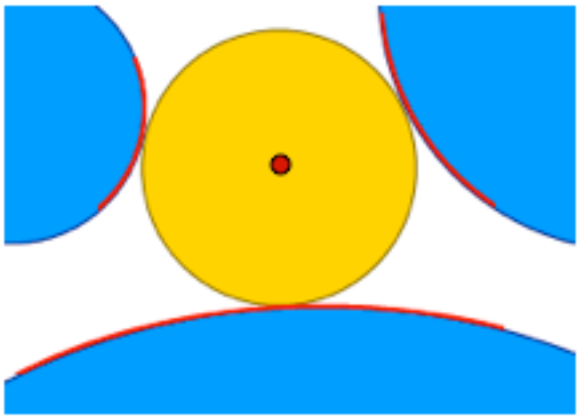
Solve for X:

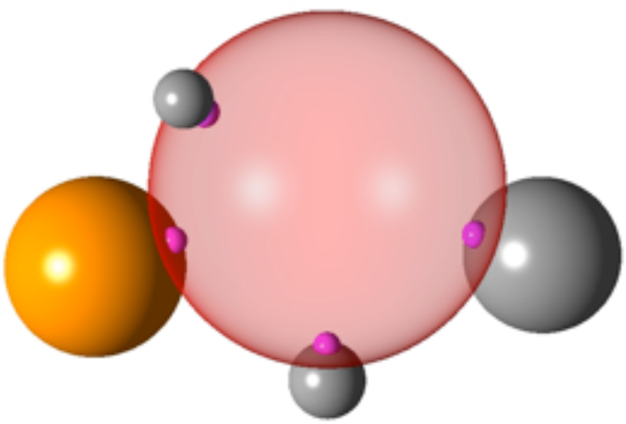
$$\begin{cases} 2AB \bullet AX = AB \bullet AB \\ 2AC \bullet AX = AC \bullet AC \end{cases}$$

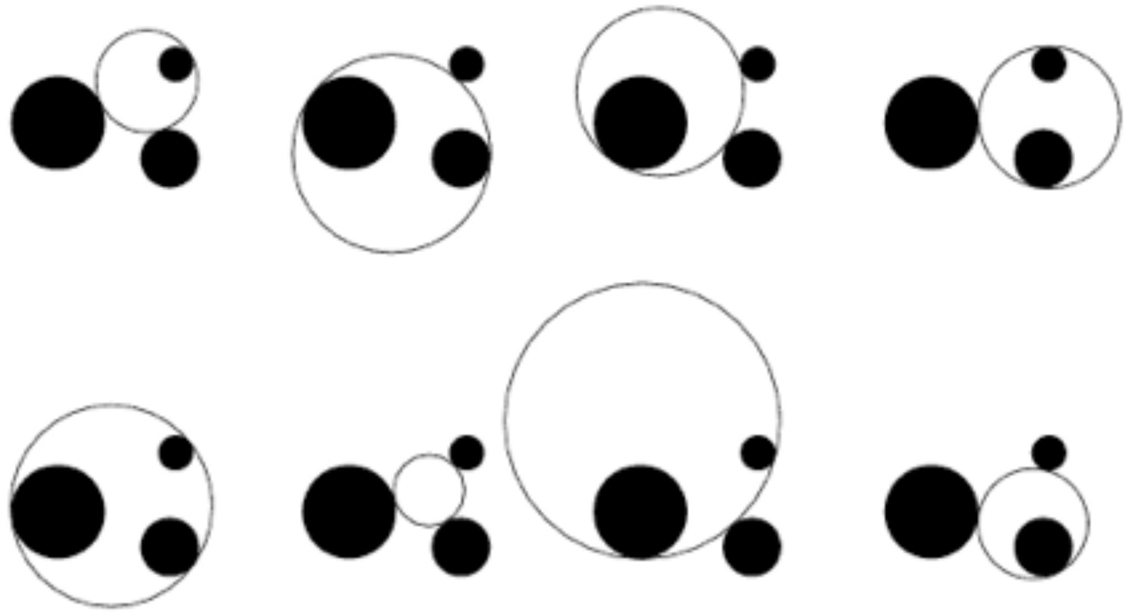


Circles & spheres tangent to others

- Compute circle tangent to 3 given ones
- In 3D, compute sphere tangent to 4 given ones.







Examples of questions for quiz…

- What is the implicit equation of circle with center C and radius r?
- What is the parametric equation of circle (C,r)?
- How to test whether a point is in circle (C.r)?
- How to test whether an edge intersects a circle?
- How to compute the intersection between an edge and a circle?
- How to test whether two circles intersect?
- How to compute the intersection of two circles
- Assume that disk(C_1,r_1) starts at t=0 and travels with constant velocity V. When will it collide with a static disk(C_2,r_2)?
- Assume that a disk(C_1,r_1) arriving with velocity V has just collided with disk(C_2,r_2). Compute its new velocity V'.

Geometry Practice

1) Point on line

When is a point P on the line passing through point Q and having unit normal vector N?

QP•N=0 , the vector from Q to a point on the line is orthogonal to N

2) Linear motion of point

- Point P starts at S and moves with constant velocity V. Where is it after t time units?

$P(t)=S+tV$, the displacement is time*velocity

3) Collision

- When will P(t) of question 2 collide with the line of question 1

  line through Q with unit normal vector N

$QP(t)=P(t)–Q=S+tV–Q=(S–Q)+tV=QS+tV$

$(QS+tV)\bullet N=0$, condition for P(t) to be on the line

Solving for t by distributing • over +

$t=(SQ\bullet N)/(V\bullet N)$, notice that $SQ=–QS$

When $V\bullet N=0$: no collision

Q may already be on the line

4) Intersection

- Compute the intersection of a line through S with tangent V with line L' through Q with **normal** N.

Compute $t=(SQ \bullet N)/(V \bullet N)$, as in the previous slide and substitute this expression for t in $P=S+tV$, yielding:
$P=S+((SQ \bullet N)/(V \bullet N))V$

If $V \bullet N=0$: no intersection

5) Medial

- When is $P(s)=S+tV$, with $|V|=1$ at the same distance from S as from the line L' through Q with **normal** N

Since $|V|=1$, $P(t)$ has traveled a distance of t from S.
The distance between $P(t)$ and the line through Q with normal N is $QP(t)\bullet N$
Hence, we have two equations: $P(t)=S+tV$ and $QP(t)\bullet N=t$

Solve for t by substitution: $t = (QS\bullet N)/(1-V\bullet N)$
If $V\bullet N=0$, use $-N$ instead of N

6) Point/line distance

- What is the distance between point P and the line through Q with normal N

d = QP•N, as used in the previous question

7) Tangent circle

- Compute the radius r and center G of the circle tangent at S to a line with normal V and tangent to a line going through Q with normal N

From question 5:

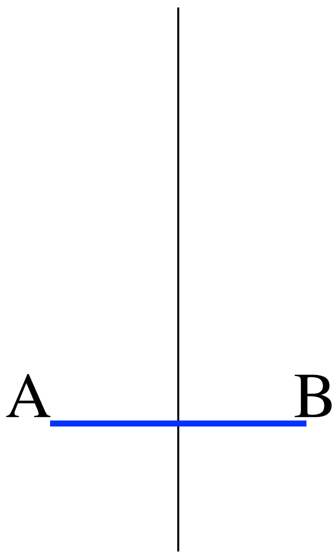
$r = (QS \bullet N)/(1 - V \bullet N)$

When $V \bullet N = 0$, use $-N$

$G = S + rV$

8) Bisector

- What is the bisector of points A and B?



Line through (A+B)/2
With normal N=AB.left.unit

9) Radius

- Compute the radius and center of the circle passing through the 3 points: A, B, and C

We compute the bisectors of AB and BC and use the result of question 3

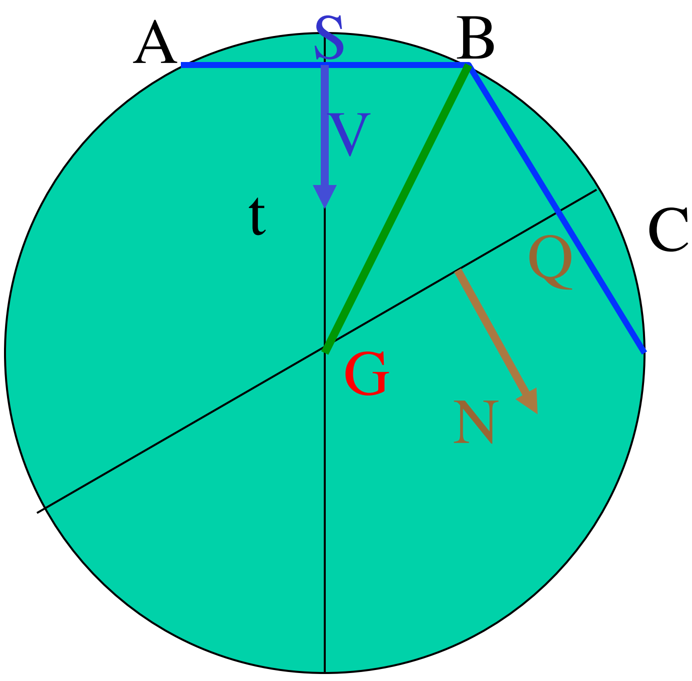
S = (A+B)/2; V = BA.left.unit

Q = (B+C)/2; N = BC.unit

t = (SQ•N)/(V•N) *(from question 3)*

G = S+tV

r = GB.unit



10) Distance

- What is the square distance between points P and Q

PQ•PQ

11) Equidistant

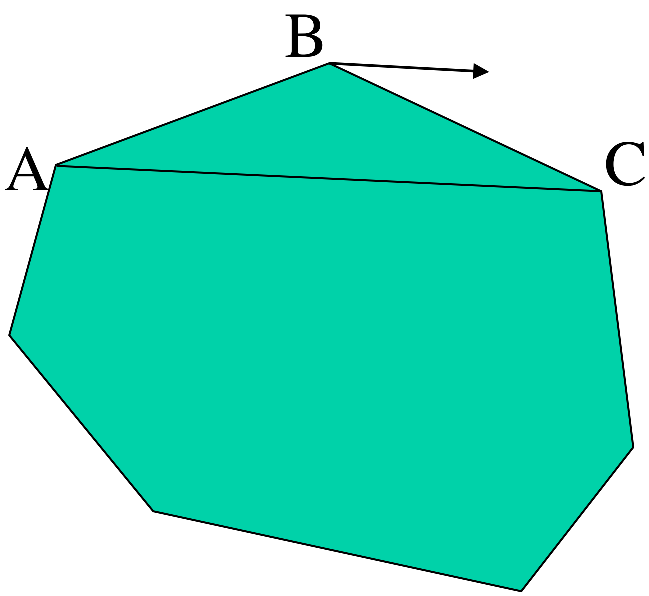
- Let $P(t)=S+tV$, with $|V|=1$. When will $P(t)$ be equidistant from points S and Q?

Similarly to question t, we have $P(t)=S+tV$
and want t such that $(QP(t))^2=t^2$
using W^2 is $W \bullet W$
$(QS+tV) \bullet (QS+tV)=t^2$
Distributing and using $V \bullet V = 1$ permits to eliminate t^2

Solving for t:
$t = QS^2/(2QS \bullet V)$

12) Tangent

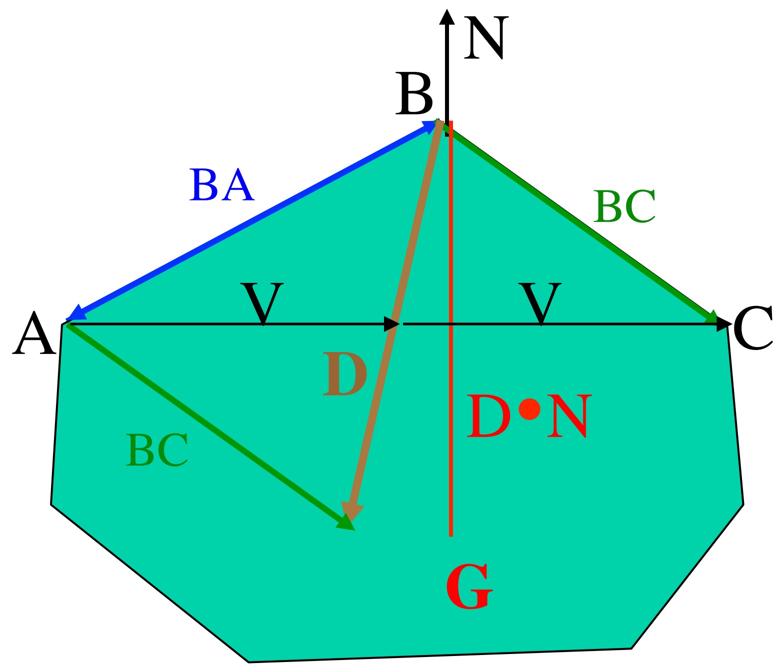
- Estimate the tangent at B to the curve that interpolates the polyloop … A, B, C…



AC.unit

13) Center of curvature

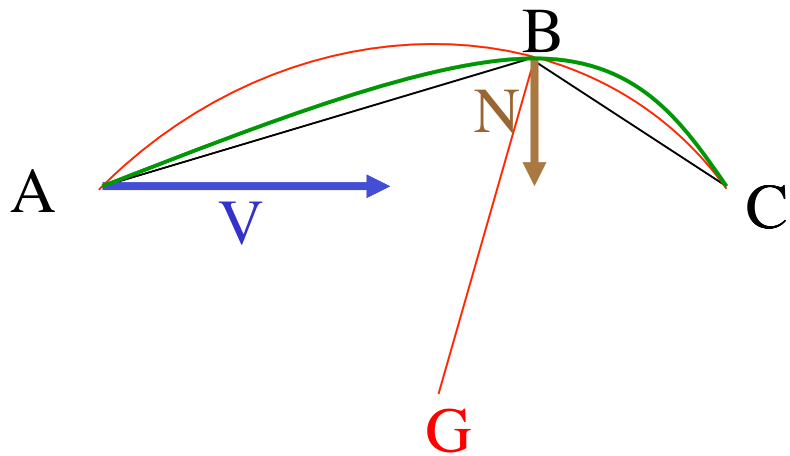
- Estimate the radius r and center G of curvature at point B the **curve approximated** by the polyline containing vertices A, B, C



Velocity: $V = AC/2$
Normal: $N = V.left.unit$
Acceleration: $D = BA+BC$
Normal acceleration: $-D \bullet N$
$\mathbf{r = -V^2/D \bullet N}$
The center of the osculating circle
$\mathbf{G=B-rN}$

Vector formulae for G in 2D and 3D?

- How to compute the center of curvature G in the previous question



V:= AC/2 ;
N:= BA + ((AB•V)/(V•V)) V ;
G:= B + ((V•V)/(2N•N)) N ;

Practice: Circle/line intersection

- When does line(P,T) intersect disk(C,r)?

$|PC\bullet(T.\text{left})| \leq r$

- Where does line(S,T) intersect disk(C,r)?

$CP = P–C = P–S–tT = SP–tT$

$(SP–tT)\bullet(SP–tT) = r^2$

$(SP\bullet SP)–2(SP\bullet T)t+(T\bullet T)t^2 = r^2$ (distribute • over –)

$t^2–2(SP\bullet T)t+(SP^2–r^2)=0$

Solve for t: real roots, t_1 and t_2, assume $t_1<t_2$

Points S+tT when $t\in]t_1,t_2[$ are in Disk(C,r)