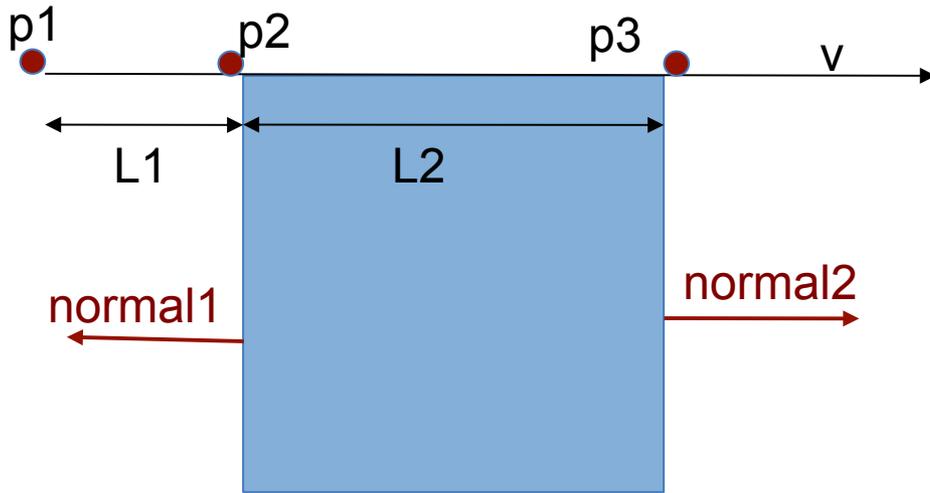


Shape Conventions

Distance & Safety

Convex shape with flat surfaces, conventions for 'scratching' directions

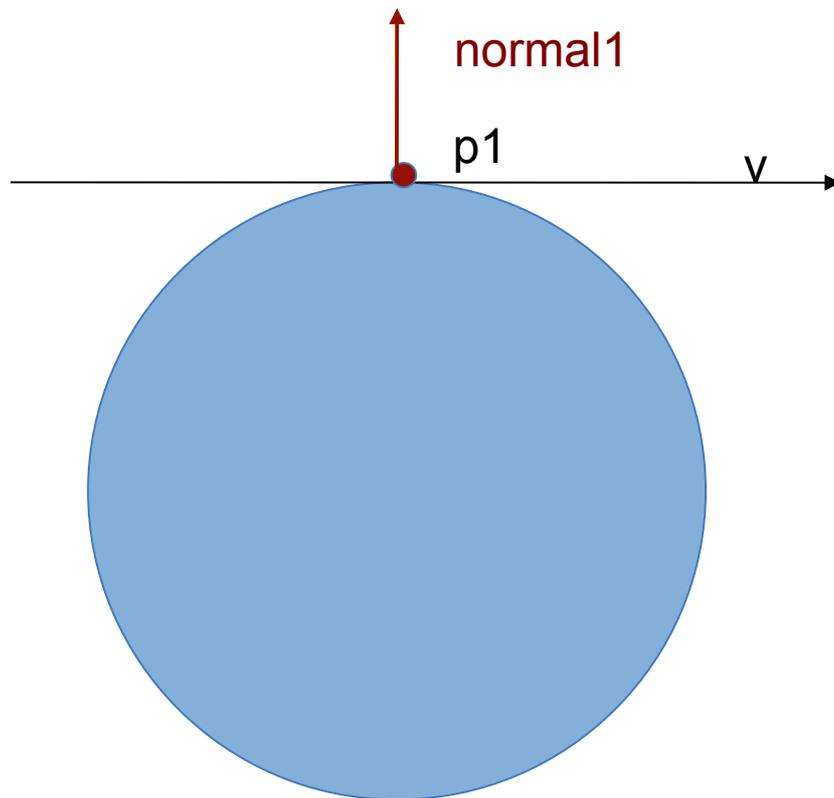


p1 : $\text{DistanceToIn}(p1,v) = L1$

p2 : $\text{DistanceToIn}(p2,v) = 0, \quad (\text{normal1},v) < 0$
 $\text{DistanceToOut}(p2,v) = L2, \quad (\text{normal1},v) < 0$

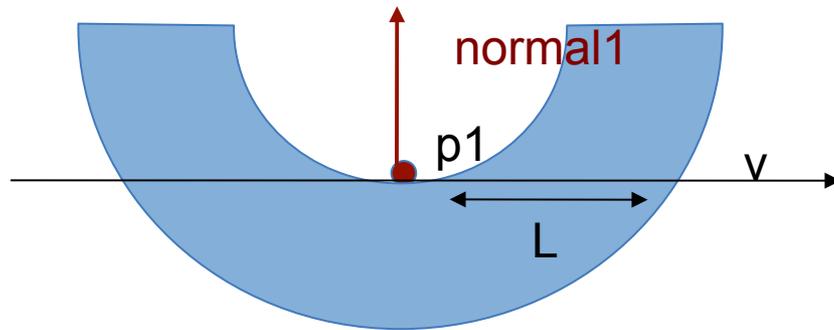
p3 : $\text{DistanceToIn}(p3,v) = \text{Infinity}, \quad (\text{normal2},v) > 0$
 $\text{DistanceToOut}(p3,v) = 0, \quad (\text{normal2},v) > 0$

Convex shape with curved surfaces, conventions for 'scratching' directions



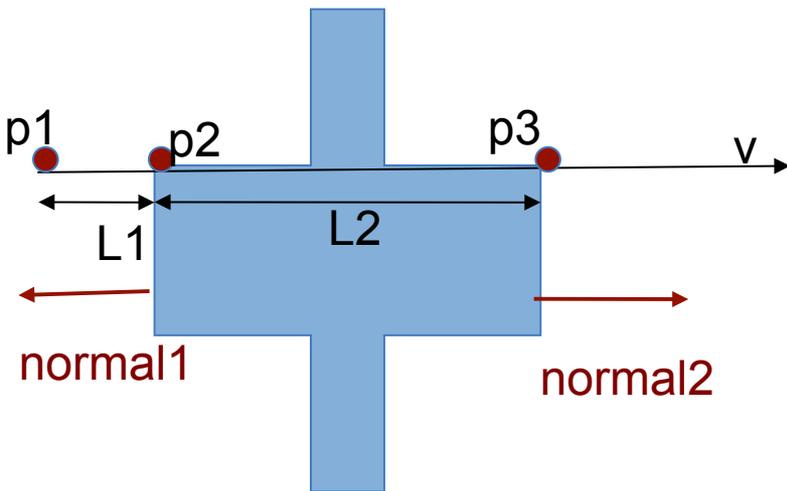
$p1$: DistanceToIn ($p1, v$) = Infinity, ($normal1, v$) = 0
DistanceToOut($p1, v$) = 0, ($normal1, v$) = 0

Concave shape with curved surfaces, 'scratching' directions



P1 : DistanceToIn (p1,v) = 0, (normal1,v) = 0
 DistanceToOut(p1,v) = L, (normal1,v) = 0

Concave shape with flat surfaces, 'scratching' directions

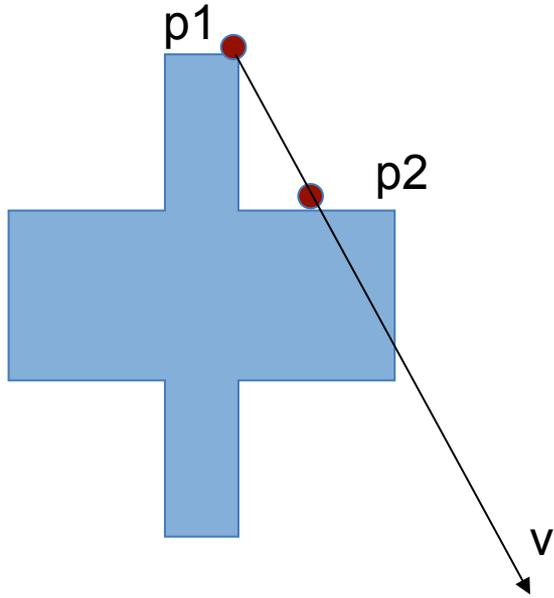


p1 : DistanceToIn(p1,v) = L1, not Infinity

p2 : DistanceToIn (p2,v) = 0, (normal1,v) < 0
 DistanceToOut(p2,v) = L2, (normal1,v) < 0

p3 : DistanceToIn (p3,v) = Infinity, (normal2,v) > 0
 DistanceToOut(p3,v) = 0, (normal2,v) > 0

Concave shapes , 'reentering' directions



$p1$: $\text{DistanceToIn}(p1, v) = L1$, not Infinity

Conventions. DistanceToIn(p,v)

DistanceToIn(p,v) return the exact distance (double) to the surface of the shape for given point p and direction v. Normal is pointing outwards shape.

	Geant4	USolids	VecGeom	ROOT
Point p is Outside No intersection Between shape and ray(p +v*t)	Infinity	Infinity	Infinity	Infinity
Point p is Outside Intersection Between shape and ray(p +v*t)	Distance	Distance	Distance	Distance
Point p is on Surface	<p>0, if 'entering shape', normal.dot(v) < 0 for convex shapes</p> <p>Infinity for convex shapes or Distance to next Intersection for concave shapes, if 'leaving shape', normal.dot(v) >= 0 for convex shapes</p>	<p>0, if 'entering shape', normal.dot(v) < 0 for convex shapes</p> <p>Infinity for convex shapes or Distance to next Intersection for concave shapes, if 'leaving shape' normal.dot(v) >= 0, for convex shapes</p>	<p>0, if 'entering shape'</p> <p>Infinity for convex shapes or Distance to next Intersection for concave shapes, if 'leaving shape'</p>	<p>0, if 'entering shape', normal.dot(v) < 0 for convex shapes</p> <p>Infinity for convex shapes or Distance to next Intersection for concave shapes, if 'leaving shape', normal.dot(v) >= 0 for convex shapes</p>
Point p is Inside "Wrong side"	No general rule(?), depends on shape	0	Negative number or -1	
If Distance(p,v)<halfTolerance	0	0	Distance	

Conventions. DistanceToOut(p,v)

DistanceToOut(p,v) return the exact distance (double) to the surface of the shape for given point p and direction v. Normal is pointing outwards shape.

	Geant4	USolids	VecGeom	ROOT
Point p is Inside No intersection Between shape and ray(p+v*t), “wrong result”	Infinity, as default value	Infinity, as default value	Infinity, as default value	Infinity, as default value
Point p is Inside Intersection Between shape and ray(p+v*t)	Distance	Distance	Distance	Distance
Point p is on Surface	0 , if 'leaving' shape, normal.dot(v) >= 0 for convex shape DistanceTo next boundary , if 'entering' shape, normal.dot(v) < 0 for convex shape	0 , if 'leaving' shape, normal.dot(v) >= 0, for convex shape DistanceTo next boundary , if 'entering' shape, normal.dot(v) < 0 for convex shaper	0 , if 'leaving shape' DistanceTo next boundary , if 'entering shape'	0 , if 'leaving' shape, normal.dot(v) >= 0 for convex shape DistanceTo next boundary , if 'entering' shape ,normal.dot(v) < 0 for convex shape
Point p is Outside “Wrong side”	No general rule(?), depends on shape	0	Negative number or -1	
If Distance(p,v)<halfTolerance	0	0	Distance	

Conventions. *SafetyFromOutside(p)*

SafetyFromOutside estimates isotropic distance to the surface of the shape from Outside. This must be either accurate or an underestimate.

	Geant4	USolids	VecGeom	ROOT
Point p is Outside	Safety	Safety	Safety	Safety
Point p is on Surface	0	0	0	0
Point p is Inside "Wrong side"	0	0	Negative number or -1	Negative number or -1

Conventions. *SafetyFromInside(p)*

SafetyFromInside(p) estimates isotropic distance to the surface of the shape from Inside point *p*. This must be either accurate or an underestimate.

	Geant4	USolids	VecGeom	ROOT
Point <i>p</i> is Inside	Safety	Safety	Safety	Safety
Point <i>p</i> is on Surface	0	0	0	0
Point <i>p</i> is Outside "Wrong side"	0	0	Negative number or -1	Negative number or -1