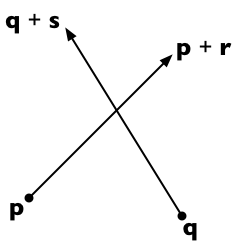
[**How do you detect where two line segments intersect?**](http://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-intersect)

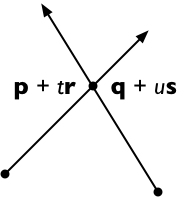
There’s a nice approach to this problem that uses vector cross products. Define the 2-dimensional vector cross product **v** × **w** to be **v**x**w**y − **v**y**w**x (this is the magnitude of the 3-dimensional cross product).

Suppose the two line segments run from **p** to **p** + **r** and from **q** to **q** + **s**. Then any point on the first line is representable as **p** + t **r** (for a scalar parameter t) and any point on the second line as **q** + u **s** (for a scalar parameter u).



The two lines intersect if we can find t and u such that:

**p** + t **r** = **q** + u **s**



Cross both sides with **s**, getting

(**p** + t **r**) × **s** = (**q** + u **s**) × **s**

And since **s** × **s** = **0**, this means

t(**r** × **s**) = (**q** − **p**) × **s**

And therefore, solving for t:

t = (**q** − **p**) × **s** / (**r** × **s**)

In the same way, we can solve for **u**:

u = (**q** − **p**) × **r** / (**r** × **s**)

Now if **r** × **s** = **0** then the two lines are parallel. (There are two cases: if (**q** − **p**) × **r** = **0** too, then the lines are collinear, otherwise they never intersect.)

Otherwise the intersection point is on the original pair of line segments if 0 ≤ t ≤ 1 and 0 ≤ u ≤ 1.

(Credit: this method is the 2-dimensional specialization of the 3D line intersection algorithm from the article "Intersection of two lines in three-space" by Ronald Goldman, published in Graphics Gems, page 304. In three dimensions, the usual case is that the lines are skew (neither parallel nor intersecting) in which case the method gives the points of closest approach of the two lines.)