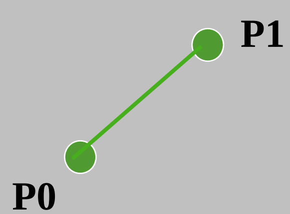
Parametric Line Equation

Mike Bailey

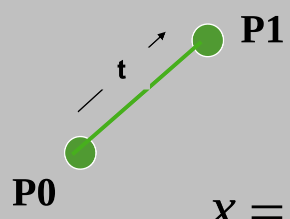
Oregon State University

The Line Equation: Why “y=mx+b” Isn't Good Enough for Graphics Applications Software



1. Cannot represent vertical lines ($m = \infty$)
2. Can only represent infinite lines, not finite line segments
3. Can only represent 2D lines, not 3D

Parametric Line Equation



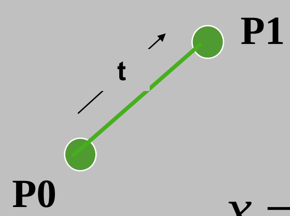
$$x = X_0 + t(X_1 - X_0)$$

$$y = Y_0 + t(Y_1 - Y_0)$$

$$z = Z_0 + t(Z_1 - Z_0)$$

$$0. \le t \le 1.$$

Can Also Be Thought of As a Blending Function



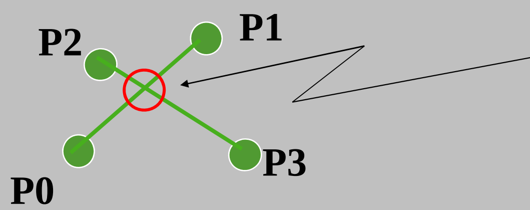
$$x = (1-t)X_0 + tX_1$$

$$y = (1-t)Y_0 + tY_1$$

$$z = (1-t)Z_0 + tZ_1$$

$$0. \le t \le 1.$$

Example Use: Intersection of two 2D Line *Segments*



$$x_{01} = x_{23}$$
$$y_{01} = y_{23}$$

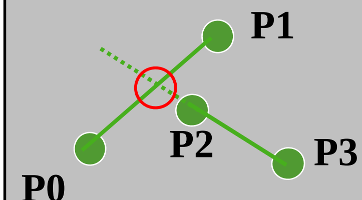
$$x_{01} = X_0 + t(X_1 - X_0) \qquad x_{23} = X_2 + u(X_3 - X_2)$$
$$y_{01} = Y_0 + t(Y_1 - Y_0) \qquad y_{23} = Y_2 + u(Y_3 - Y_2)$$

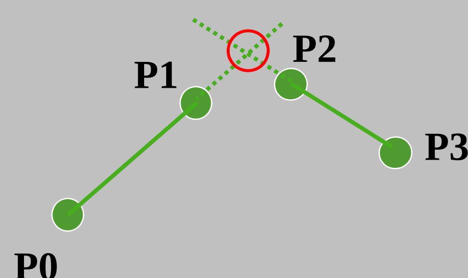
Since, at the point of intersection, $x_{01} = x_{23}$ **and** $y_{01} = y_{23}$

we have 2 equations in 2 unknowns (t and u)

Solve for t* and u*. If they are not each between 0. and 1., then the infinite lines intersect, but not the finite line segments. Plug t* and u* back into the equations above to find (x*,y*), the point of intersection.

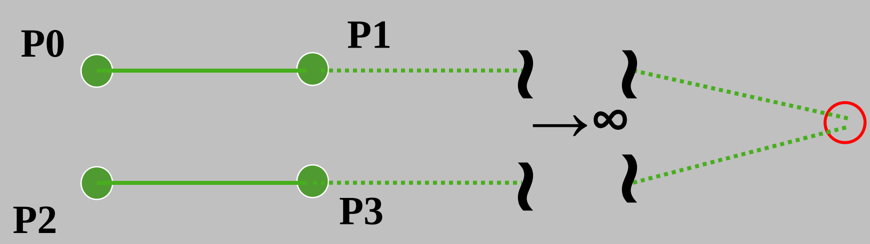
What if t* or u* is < 0. or > 1. ?





It means you have a situation like one of these. The infinite lines intersect, but not the finite line segments.

What if the lines are Parallel ?



Well, they *do* intersect, but at infinity. This shows up in the math by the expression for t* and u* becoming infinitely large, that is, there would be a divide by zero.

So, you need to check the denominator before doing the division.

Since t* and u* would both be infinity in this case, which is most certainly > 1, that means that the line segments do not intersect.

Linear Blending Shows Up in a Lot of Computer Graphics Applications

You can linearly blend *any* two quantities with:

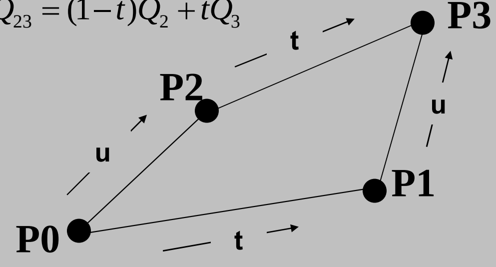
$$Q = (1-t)Q_0 + tQ_1$$

color, shape, location, angle, scale factors, •••

Linear Blending Shows Up in a Lot of Computer Graphics Applications as Well

You can bilinearly blend *any* 2D quantities within a quadrilateral by writing the line blending equation twice and then blending the two lines with:

$$Q_{23} = (1-t)Q_2 + tQ_3$$



$$Q_{01} = (1-t)Q_0 + tQ_1$$

$$\boxed{Q_{0123} = (1-u)Q_{01} + uQ_{23} = (1-t)(1-u)Q_0 + t(1-u)Q_1 + (1-t)uQ_2 + tuQ_3}$$

Linear Blending Shows Up in a Lot of Computer Graphics Applications as Well

Treating a quadrilateral as two triangles:

Treating a quadrilateral with bilinear interpolation: