// create a value of 0. or 1. from the value of x wrt edge:
float  t  =  step( float edge, float x );

// create a value in the range 0. to 1. from the value of x wrt edge0 and edge1:
float  t  =  smoothstep( float edge0, float edge1, float x );

// use the returned value from step( ) or smoothstep( ) to blend value0 to value1:
T  out  =  mix( T  value0,  T  value1,  float  t );
in float vX, vY;
in vec4 vColor;
in float vLightIntensity;

uniform float uA;
uniform float uP;
uniform float uTol;

const vec4 WHITE = vec4( 1., 1., 1., 1.);

void main( )
{
    float f = fract( uA*vX );
    float t = smoothstep( 0.5-uP-uTol, 0.5-uP+uTol, f ) - smoothstep( 0.5+uP-uTol, 0.5+uP+uTol, f );
    gl_FragColor = mix( WHITE, vColor, t );
ge federate light i ntensity; 
}

---

“SmoothPulse” in a Fragment Shader

Fun With One

Moral: There are many ways to turn \([ 0. - 1. ]\) into \([ 0. - 1. ]\)
Why Do These Two Curves Match So Closely?

The Taylor Series expansion of \( y = \sin^2 \left( \frac{\pi x}{2} \right) \) around \( x = 0.5 \) is:

\[
y = \left( \frac{1}{2} - \frac{\pi^2}{4} + \frac{\pi^4}{96} \right) + x \left( \frac{\pi^2}{2} - \frac{\pi^4}{16} \right) + x^2 \left( \frac{\pi^4}{2} - \frac{\pi^6}{8} \right) - x^3 \left( \frac{\pi^6}{12} \right)
\]

\[
= .038 - .37x + 3.88x^2 - 2.58x^3
\]

which is pretty close to: \( y = 3x^2 - 2x^3 \)

---

Cubic vs. Quintic

Both go from 0. to 1.
Both have initial and final slopes of 0.
The quintic has initial and final curvatures of 0.