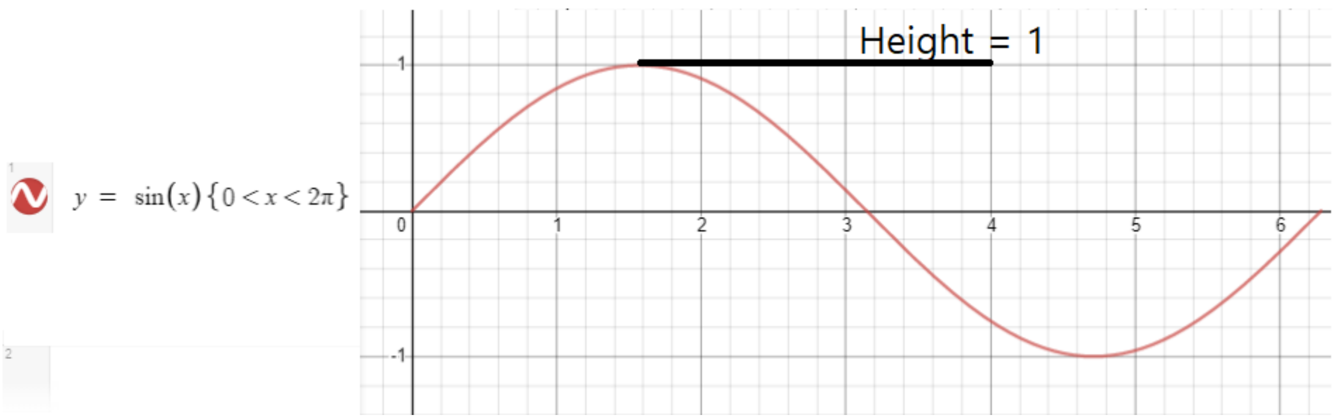
Waves

Wave

> y = sin(x) {0 < x < 2PI}

> Properties

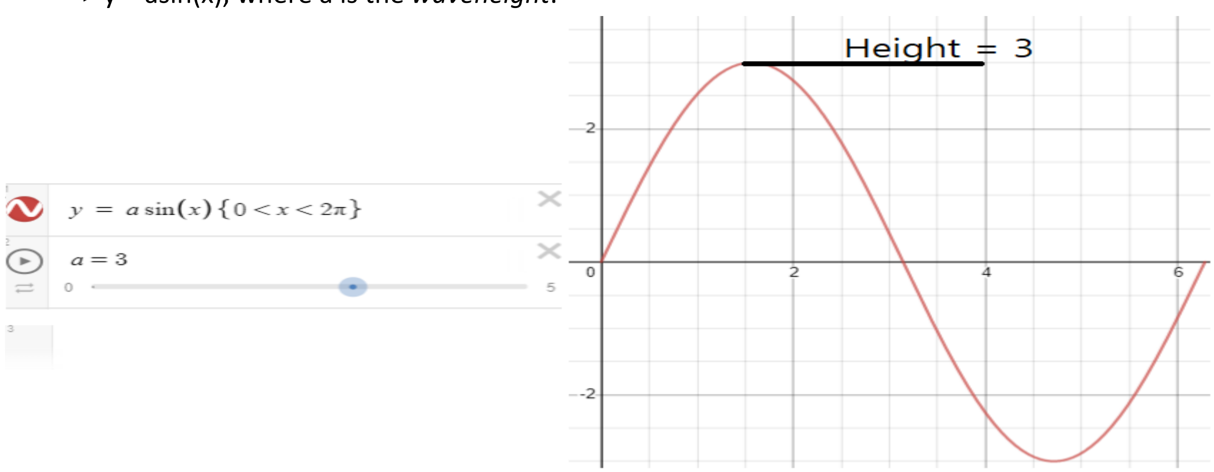
* A wave has a height of 1.
* A wave has a length of 2PI (or 6.28).



Amplitude

> A value that increases the waveheight.

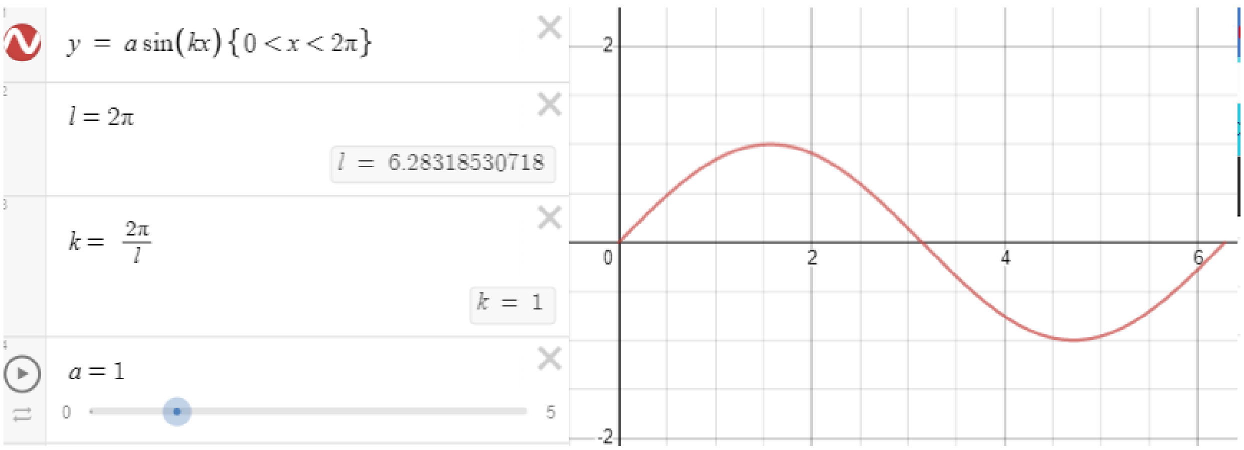
> y = asin(x), where a is the *waveheight*.



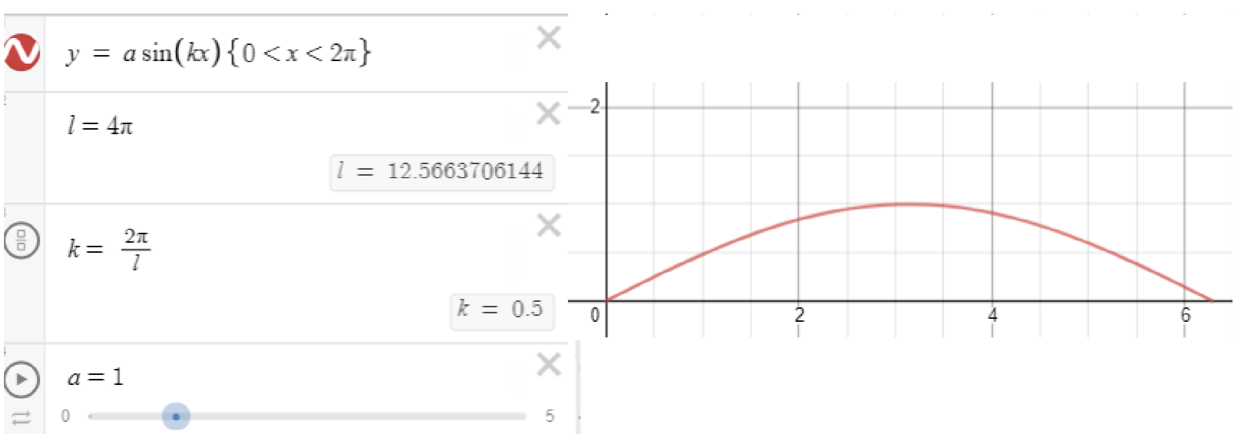
Wavelength & Wavenumber

> A value that increases/decreases the length of a wave.
> The standard length of a wave is 2PI.
> If we divide 2PI by some number, and multiply this result by the x inside sin(x), we would increase/decrease the wavelength.
> 2PI/λ, where λ is the *wavelength*.
> k = 2PI/λ , where k is called the *wave number.*
> y = asin(k*x)
> Properties

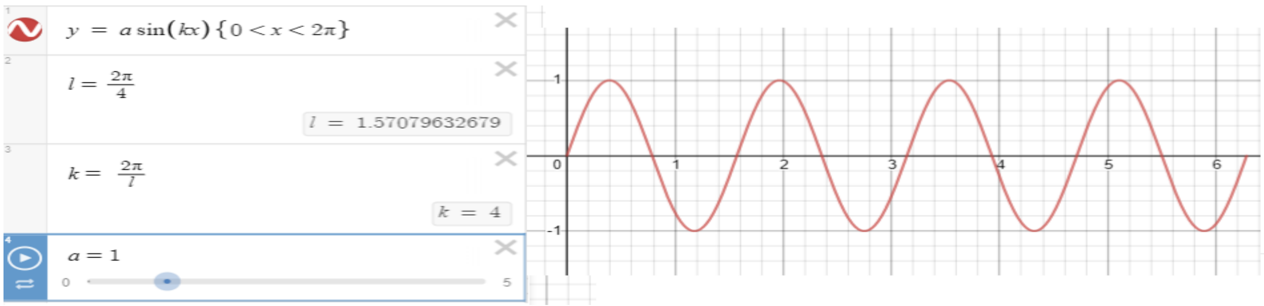
    * if λ = 2PI/c, where c is some constant, the wave will have c crests ("hills").
    This is shown in *Graphic 3.*



Graphic 1. Here λ (represented by l) = 2PI, so *k = 2PI/2PI, thus k = 1 and nothing happens.*



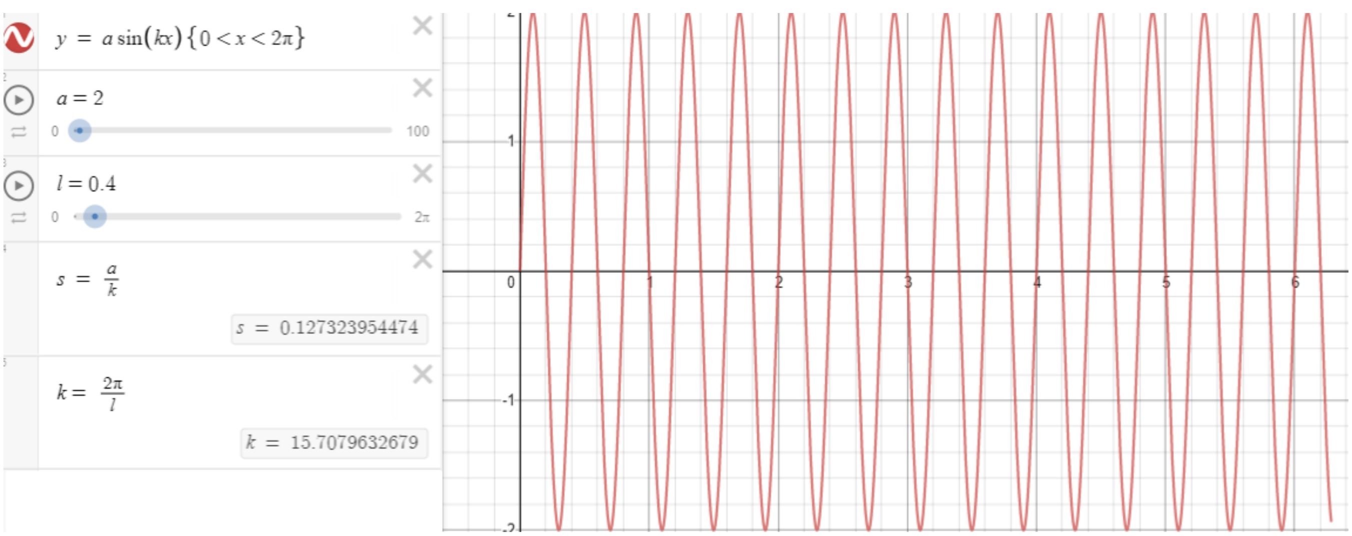
Graphic 2. Here λ = 4PI, so k = 2PI/4PI, thus k = 0.5, and the wavelength is halved.



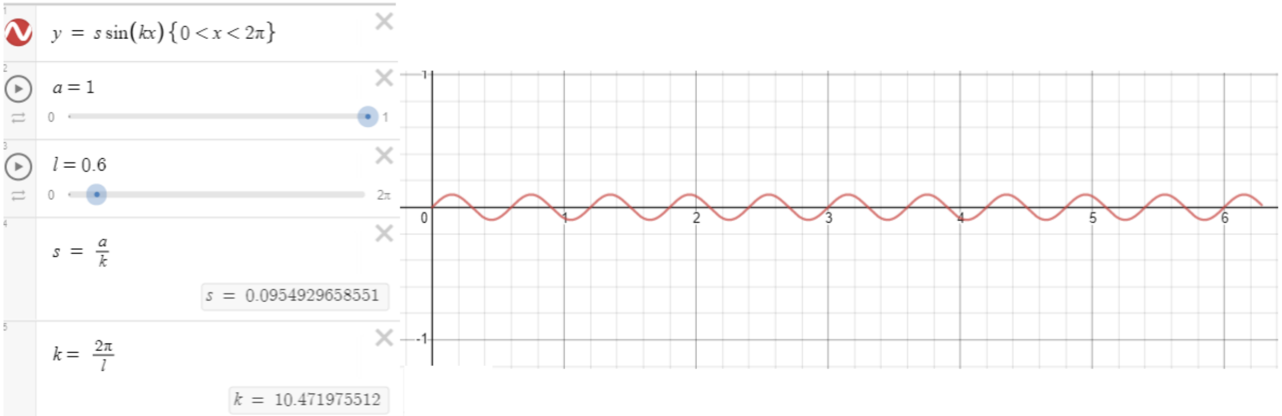
Graphic 3. Here λ = 2PI/4, so k = 2PI/(2PI/4) = 4, and the wavelength has been "shrunk" to have now 4 waves per length.

Steepness

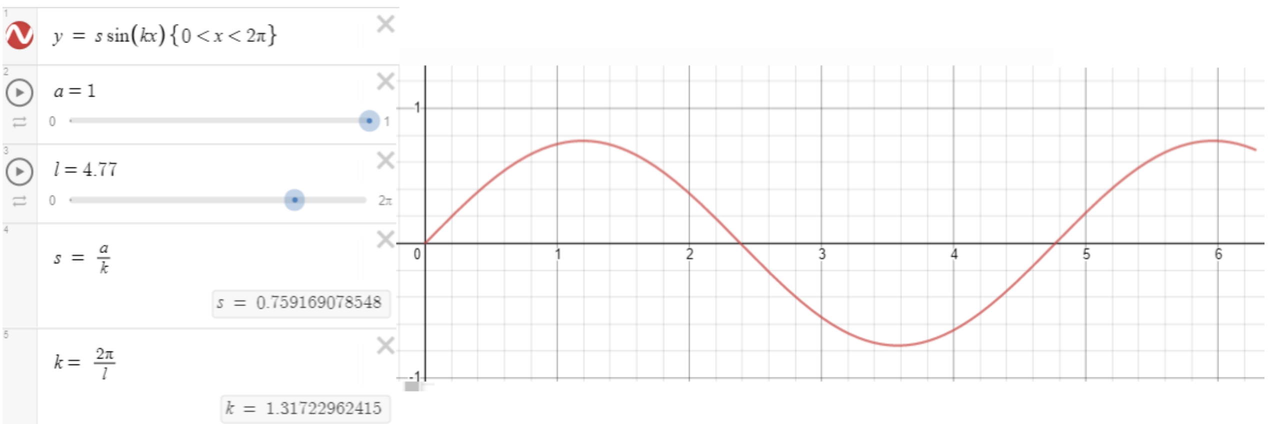
> A value that determines the ratio between waveheight and wavelength.
> s = a/k, where s is called steepness.
> y = s*sin(kx)
> The range of s = [0,1] if λ's domain = [0,2PI] and waveheights' domain = [0,1]
> So now height is not independent of its wavelength (which is good if we want to control the wave and not let any crazy values loose, i.e. *Graphic 4*



Graphic 4. Here λ = 0.4, which means k = 2PI/0.4 = 15.7, meaning that the wavelength gonna pack a lot of jiggly jiggly. Nonetheless, the amplitude will still make the waveheight be 2 units tall. Generally, for creating ocean , waves or whatev this shit be to wildin' so its preferred that we find that smooth bad boy Young Steepness.



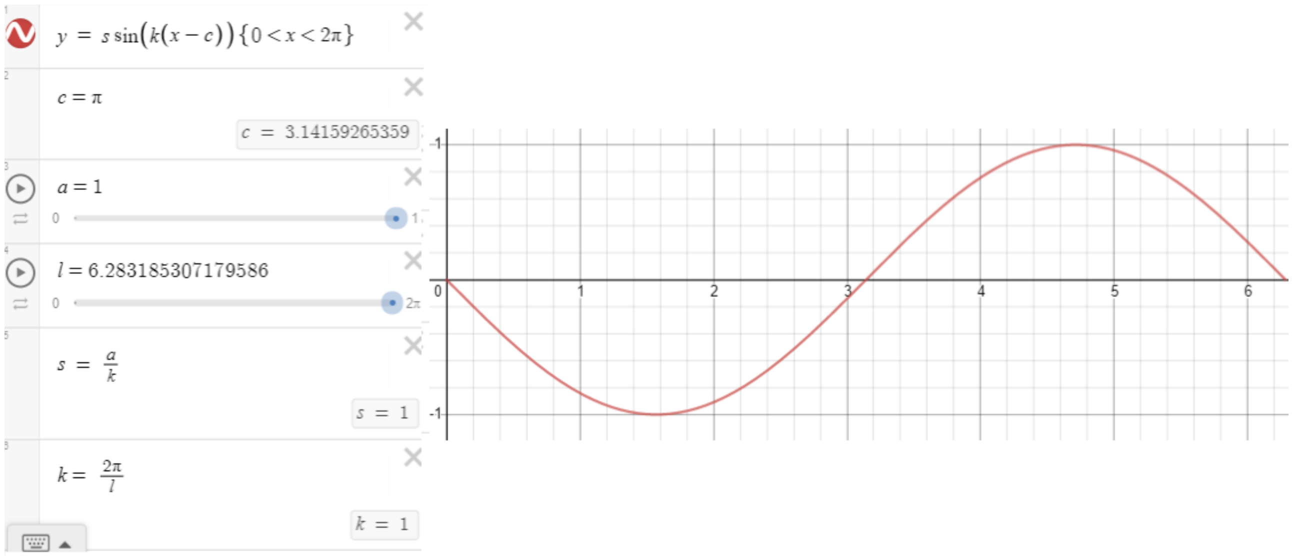
Graphic 5. Here λ is capped to 2PI, and the amplitude is capped to 1. Also, we're using steepness. That's the maximum height for a wavelength of 0.6. Because height is capped by the wavenumber (a/k).



Graphic 6. Here the λ increased, and so even though a = 1 (just like in Graphic 5), here the wave is taller, cause before it was capped by the wavelength, and now that λ grew up to be a real G.

Offset

> If we offset the x inside sin(x) by some number c, we would move the wave in relation to the c distance.
> If we we're to move the wave y = s * sin(k(x)), it would then be by offseting its original x position.
> y = s * sin(k(x-c))



Graphic 7. This isn't your typical sin(x), here the original sin(x) was offsetted by c = PI. So all values shifted and gave what was behind that zero when we had sin(x).

Time

> If we we're to continually offset c in small increments that continue on indefinately, we'd emulate a flowing wave.

> say t = some value that's starts from zero and slowly continues growing indefinately.

> y = s * sin(k (x - (c * t)))

> So here the c would also be the *wave speed* of its flow.

Gersnter Waves

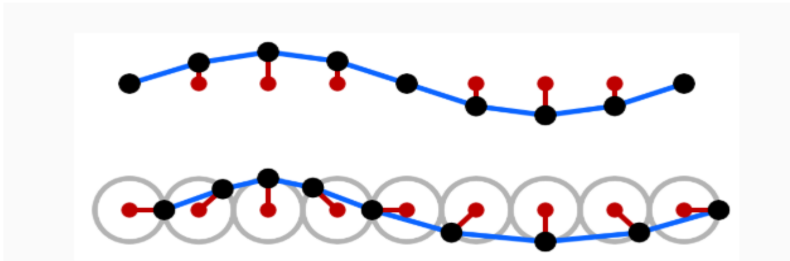
> Up until now, we've been moving only been plotting points P = (x, sin(x)), and learned about features we could give the sin(x). In order to create a realistic wave, we could use the Gersnter Wave which also takes the x point into consideration.

> lets change y = s * sin (k (x - (c * t))) to be incapsulated in g(x).

$$g(x) = k(x - (c \cdot t))$$

$$y = s \sin(g(x)) \{0 < x < 2\pi\}$$

> Now, Gernster Waves take x into consideration how? Well first lets look at this pic.



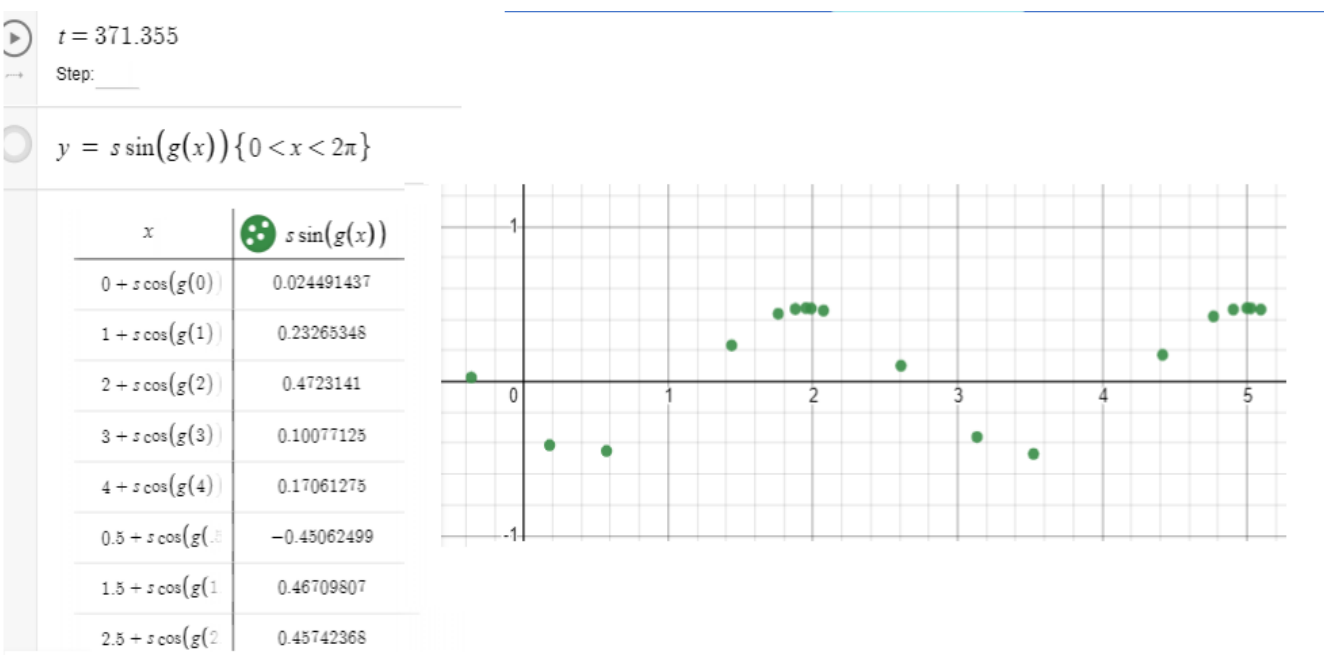
Sine vs. Gerstner wave.

> Before we used y and only plotted points going up and down. With Gersnter Waves, we're also moving x in a circular fashion, which will add a layer of realism to our waves.

> The circle can be plotted via P = (s*cos(g(x)), s*sin(g(x))). That's great. We got the circle. But the problem is, this circle is stuck in the origin, no matter how many points we plot, because points using (cos(x), sin(x)) will remain at origin unless we do a lil push. We didn't need this before because we had P = (x, s*sin(g(x))) so our points moved along nicely in the x position.

> To solve this simply add x distance to the x point we're focusing on.

P = (x + s*cos(g(x)), s*sin(g(x))) where g(x) = k(x-ct)



Graphic 8. Gernstsnnsssrr Wavers baby. As you can see, the points have a strong bias towards the tip, which makes it more oceany.

Phase Speed

> Waves don't really have a number c that determines how fast they're going, unlike our g(x) = s * sin(k (x - (c * t))). What actually determines wave speed is gravity

(g = 9.82m/s^2) for Earth and the wave number. So we could use thiWs value instead of our arbitrary c.

> Make c then be equal to sqrt(g/k) instead of an arbitrary value.

> Now g(x) = s * sin(k (x - (c * t))) where c = sqrt(g/k).

Unit Tangent Vectors

> If the points are to be used as 3D vertices, then we'd need to define both the tangent vector and the normal vector of this wave mesh, because vertices hold a lot of stuff, like position, normal vector, uvs, tangent vectors, color, etc.

> The *Unit Tangent Vector (T)* is a unit vector (aka has a magnitude or length of one) that has a direction parallel to the surface at point P. (this is my definition, it might not be exactly true but you'll get what I mean from *Graphic 9*.

> The formula for the tangent line of a 2D curve with a parametric equation r(t) = (f(t), g(t)) at some point t requires you to find the derivatives f'(t) and g'(t), obtaining

T(x) = (g'(t) / f'(t)) * (x-t) + g(t)

> Now for a function in space, say r(t) = (f(t), g(t)), to find its unit tangent vector (length of one), we'd be require to derive r'(t), dividing that by the magnitude of r'(t).

T(t) = r'(t)/|r'(t)|

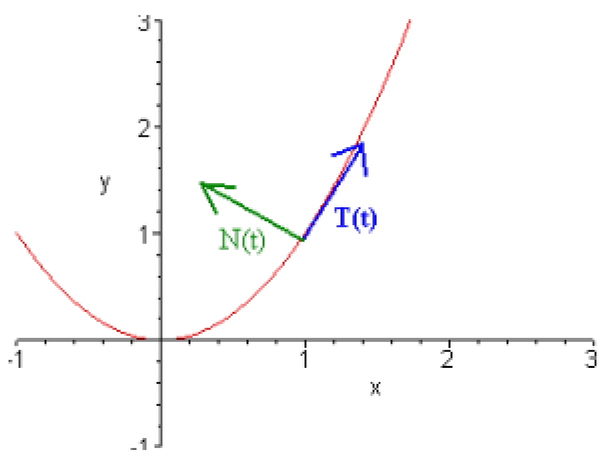
> The reasoning behind this division is because if its the "Unit" tangent vector, then our vector needs a length of one, so dividing the r'(t) by its magnitude will give us that.

> To find the tangent vector of a standard sine wave, say

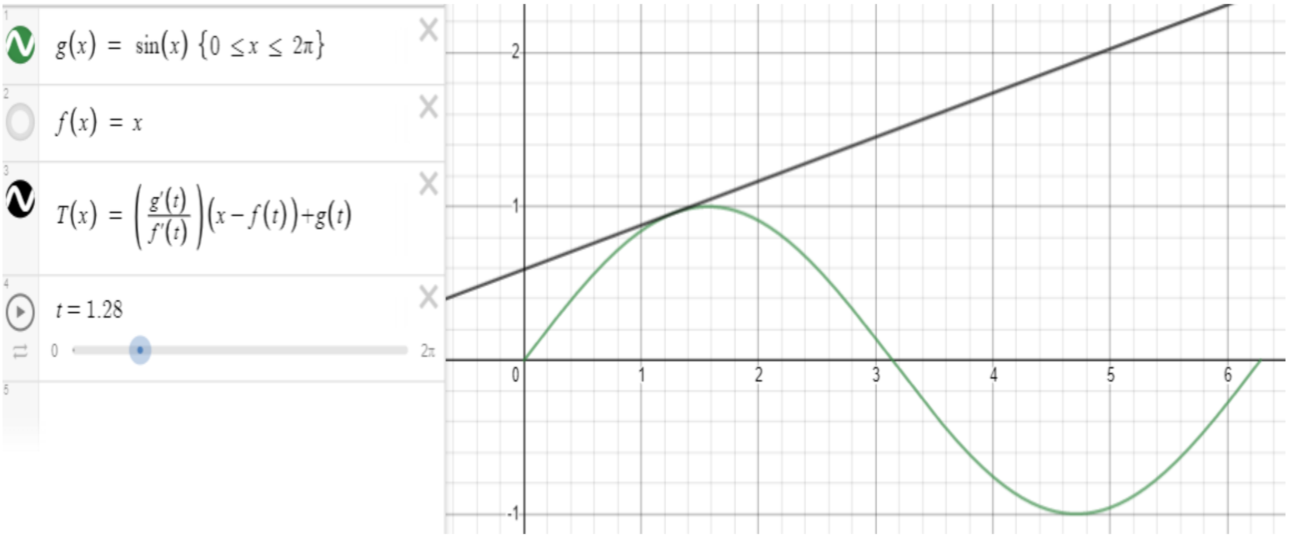
r(x) = (x, asin(f))

where f = k(x-ct)

we'd need r'(x) = (1, kacos(f)) and then divide by |r'(x)| or if you have a "normalize" function in whatever software you use, then use that.



Graphic 9. The Tangent Vector and Normal Vector at point (1,1) of this function .

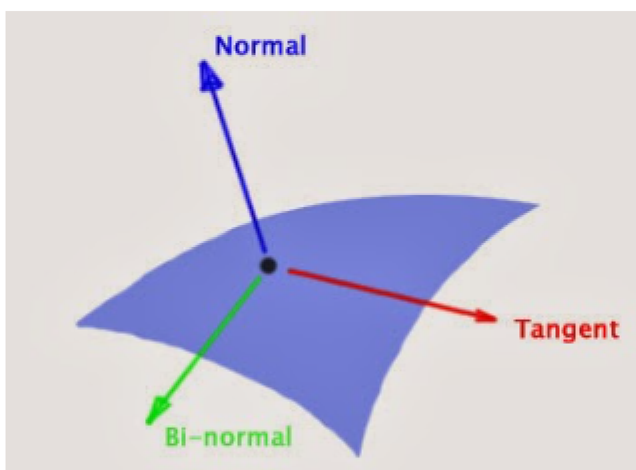


Graphic 10. Using the sin wave as g(x) and x as f(x) we can see the tangent line at point 1.28

Normal Vectors

> The vector that is perpendicular to a surface at point P.

> The normal vector is the cross product of both tangent vectors (tangent and bitangent). Look at *Graph 11.*



Graph 11