Perlin Noise to Gradient:

Consider 4D Perlin Noise projected into 3D space such that one axis serves as the time axis t which controls the noise's evolution through time. Imagine that the value returned by this 3D Perlin Noise is a heightmap for some hypothetical surface in this 3D space (think of a planet – the noise would represent the mountains and valleys on the surface):

$$h(x,y,z) = noise(x,y,z,t)$$

Where $noise(x,y,t)$ denotes the 3D Perlin Noise function at the given coordinates.

Using this approach, we can effectively create the graph of a surface that evolves over time. Thus, the mathematical operation of the gradient exists for this object. For continuous mathematical functions, the gradient is defined for a function $f(x,y,z)$ as follows:

$$\nabla f = \langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \rangle$$

In the case of this Perlin Noise surface example, the surface $\mathcal{S}$ is defined as:

$$\mathcal{S}(x,y,z,t) = noise(x,y,z,t)$$

Applying the gradient formula to this function results in the following vector field (gradient field) for the gradient:

$$\boldsymbol{g}(x,y,z,t) = \begin{pmatrix} \frac{\partial}{\partial x}\big(noise(x,y,z,t)\big) \\ \frac{\partial}{\partial y}\big(noise(x,y,z,t)\big) \\ \frac{\partial}{\partial z}\big(noise(x,y,z,t)\big) \end{pmatrix}$$

In order to evaluate this in Processing, we must convert this continuous definition to a discretized form. To do so, we do the following:

$$\boldsymbol{g}(x,y,z,t) = \begin{pmatrix} \frac{noise(x+\delta x,y,z,t) - noise(x-\delta x,y,z,t)}{2\delta x} \\ \frac{noise(x,y+\delta y,z,t) - noise(x,y-\delta y,z,t)}{2\delta y} \\ \frac{noise(x,y,z+\delta z,t) - noise(x,y,z-\delta z,t)}{2\delta z} \end{pmatrix}$$

So this is good! We like this; however, we're working on the surface of a sphere which means we can easily describe locations in terms of spherical coordinates. It's also required for the conversion to equirectangular coordinates, so we establish the following definition:

Spherical to Cartesian	**Cartesian to Spherical**
$x = \rho \sin\phi \cos\theta$	$\rho = \sqrt{x^2 + y^2 + z^2}$
$y = \rho \sin\phi \sin\theta$	$\tan\theta = \frac{y}{x}$
$z = \rho \cos\phi$	$\cos\phi = \frac{z}{\rho}$

So now that we know the spherical coordinates, we can define a conversion to equirectangular. Note that these are normalized coordinates. Equirectangular coordinates must be multiplied by the final image size and spherical coordinates must be added to a radius:

Spherical to Equirectangular	**Equirectangular to Spherical**
$x_{equi} = \frac{\theta}{2\pi} + .5$	$\theta = 2\pi\big(x_{equi} - .5\big)$
$y_{equi} = \frac{\phi}{pi}$	$\phi = \pi \cdot y_{equi}$

The next component is to take the curl of this new vector field, because that is in and of itself the velocity field that we're after. Note that this is the curl formula for 3D; however, we only have a 2D function, so the third component is 0.

$$curl(\boldsymbol{g}) = \begin{pmatrix} \frac{\partial}{\partial y}(g_3) - \frac{\partial}{\partial z}(g_2) \\ \frac{\partial}{\partial z}(g_1) - \frac{\partial}{\partial x}(g_3) \\ \frac{\partial}{\partial x}(g_2) - \frac{\partial}{\partial y}(g_1) \end{pmatrix}$$

$$curl(\boldsymbol{g}) = \begin{pmatrix} \frac{\frac{noise(x,y+\delta y,z+\delta z,t)-noise(x,y+\delta y,z-\delta z,t)}{2\delta z} - \frac{noise(x,y-\delta y,z+\delta z,t)-noise(x,y-\delta y,z-\delta z,t)}{2\delta z}}{2\delta y} - \frac{\frac{noise(x,y+\delta y,z+\delta z,t)-noise(x,y-\delta y,z+\delta z,t)}{2\delta y} - \frac{noise(x,y+\delta y,z-\delta z,t)-noise(x,y-\delta y,z-\delta z,t)}{2\delta y}}{2\delta z} \\ \frac{\frac{noise(x+\delta x,y,z+\delta z,t)-noise(x-\delta x,y,z+\delta z,t)}{2\delta x} - \frac{noise(x+\delta x,y,z-\delta z,t)-noise(x-\delta x,y,z-\delta z,t)}{2\delta x}}{2\delta z} - \frac{\frac{noise(x+\delta x,y,z+\delta z,t)-noise(x+\delta x,y,z-\delta z,t)}{2\delta z} - \frac{noise(x-\delta x,y,z+\delta z,t)-noise(x-\delta x,y,z-\delta z,t)}{2\delta x}}{2\delta x} \\ \frac{\frac{noise(x+\delta x,y+\delta y,z,t)-noise(x+\delta x,y-\delta y,z,t)}{2\delta y} - \frac{noise(x-\delta x,y+\delta y,z,t)-noise(x-\delta x,y-\delta y,z,t)}{2\delta y}}{2\delta x} - \frac{\frac{noise(x+\delta x,y+\delta y,z,t)-noise(x-\delta x,y+\delta y,z,t)}{2\delta x} - \frac{noise(x+\delta x,y-\delta y,z,t)-noise(x-\delta x,y-\delta y,z,t)}{2\delta x}}{2\delta y} \end{pmatrix}$$

Implemented in the code, this is much easier as it only requires subsequent calls to the method that returns the gradient.

In order to project the curl onto the surface of the sphere, we must define a plane on the surface of the sphere that has a normal vector parallel to the ρ direction. To do this, we must determine the unit vectors of spherical coordinates in terms of cartesian coordinates. To do this, we define the following:

$$\boldsymbol{r}(\rho, \theta, \phi) = \langle \rho \sin\phi \cos\theta\, , \rho \sin\phi \sin\theta\, , \rho \cos\phi \rangle$$

To calculate the unit vectors, we calculate the derivative with respect to each coordinate:

$$\boldsymbol{e}_\rho = \frac{\partial \boldsymbol{r}}{\partial \rho} = \langle \sin\phi \cos\theta\, , \sin\phi \sin\theta\, , \cos\phi \rangle$$

$$\boldsymbol{e}_\theta = \frac{\partial \boldsymbol{r}}{\partial \theta} = \langle -\sin\theta\, , \cos\theta\, , 0 \rangle$$

$$\boldsymbol{e}_\phi = \frac{\partial \boldsymbol{r}}{\partial \phi} = \langle \cos\phi \cos\theta\, , \cos\phi \sin\theta\, , -\sin\phi \rangle$$

NOTE: The vector for $\boldsymbol{e}_\theta$ upon derivation has a multiplicative factor $\frac{1}{\sqrt{\sin^2\phi}}$; however, since the domain of ϕ is $0 \le \phi \le \pi$, $\sin(\phi)$ is always positive and so this is not necessary.

Now that we have these, we can obtain the component of the curl vector $\boldsymbol{c}$ that is tangent to the sphere using the following:

$$\boldsymbol{c}_t = (\boldsymbol{c} \cdot \boldsymbol{e}_\theta)\boldsymbol{e}_\theta + (\boldsymbol{c} \cdot \boldsymbol{e}_\phi)\boldsymbol{e}_\phi$$

Additionally, there are some cases in which we must evaluate the divergence of the flow, especially to ensure that the flow remains incompressible. Divergence is given by the following:

$$div(\boldsymbol{c}_t) = \nabla \cdot \boldsymbol{c}_t$$

Now, this needs to be evaluated in spherical coordinates. I think. So let's do it! First, we must derive ∇ in spherical coordinates:

$$\nabla = \langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \rangle$$

SO NOW we simply have to redefine each in terms of the coordinates we want using the chain rule. Let's start with $\boldsymbol{x}$:

$$\frac{\partial}{\partial x}\boldsymbol{e}_x = \boldsymbol{e}_x\left(\frac{\partial}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial x} + \frac{\partial}{\partial \phi}\frac{\partial \phi}{\partial x}\right)$$

$$\equiv \boldsymbol{e}_x\left(\frac{\partial}{\partial \rho}\left(\frac{\partial}{\partial x}\left(\sqrt{x^2+y^2+z^2}\right)\right) + \frac{\partial}{\partial \theta}\left(\frac{\partial}{\partial x}\left(\tan^{-1}\left(\frac{y}{x}\right)\right)\right) + \frac{\partial}{\partial \phi}\left(\frac{\partial}{\partial x}\left(\cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)\right)\right)\right)$$

$$\equiv \boldsymbol{e}_x\left(\frac{\partial}{\partial\rho}\left(\frac{x}{\sqrt{x^2+y^2+z^2}}\right)+\frac{\partial}{\partial\theta}\left(-\frac{y}{x^2+y^2}\right)+\frac{\partial}{\partial\phi}\left(\frac{zx}{(x^2+y^2+z^2)^{\frac{3}{2}}*\sqrt{-\frac{z^2}{x^2+y^2+z^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_x\left(\frac{\partial}{\partial\rho}\left(\frac{\rho\sin\phi\cos\theta}{\rho}\right)+\frac{\partial}{\partial\theta}\left(-\frac{\rho\sin\phi\sin\theta}{(\rho\sin\phi\cos\theta)^2+(\rho\sin\phi\sin\theta)^2}\right)+\frac{\partial}{\partial\phi}\left(\frac{\rho\cos\phi*\rho\sin\phi\cos\theta}{\rho^3*\sqrt{-\frac{(\rho\cos\phi)^2}{\rho^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_x\left(\frac{\partial}{\partial\rho}(\sin\phi\cos\theta)+\frac{\partial}{\partial\theta}\left(-\frac{\sin(\theta)}{\rho\sin(\phi)}\right)+\frac{\partial}{\partial\phi}\left(\frac{\cos(\phi)\cos(\theta)}{\rho}\right)\right)$$

Now we do y:

$$\frac{\partial}{\partial y}\boldsymbol{e}_y=\boldsymbol{e}_y\left(\frac{\partial}{\partial r}\frac{\partial r}{\partial y}+\frac{\partial}{\partial\theta}\frac{\partial\theta}{\partial y}+\frac{\partial}{\partial\phi}\frac{\partial\phi}{\partial y}\right)$$

$$\equiv \boldsymbol{e}_y\left(\frac{\partial}{\partial\rho}\left(\frac{\partial}{\partial y}\left(\sqrt{x^2+y^2+z^2}\right)\right)+\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial y}\left(\tan^{-1}\left(\frac{y}{x}\right)\right)\right)+\frac{\partial}{\partial\phi}\left(\frac{\partial}{\partial y}\left(\cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)\right)\right)\right)$$

$$\equiv \boldsymbol{e}_y\left(\frac{\partial}{\partial\rho}\left(\frac{y}{\sqrt{x^2+y^2+z^2}}\right)+\frac{\partial}{\partial\theta}\left(\frac{1}{x\left(\frac{y^2}{x^2}+1\right)}\right)+\frac{\partial}{\partial\phi}\left(\frac{zy}{(x^2+y^2+z^2)^{\frac{3}{2}}*\sqrt{-\frac{z^2}{x^2+y^2+z^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_y\left(\frac{\partial}{\partial\rho}\left(\frac{\rho\sin\phi\sin\theta}{\rho}\right)+\frac{\partial}{\partial\theta}\left(\frac{1}{\rho\sin\phi\cos\theta*\left(\frac{(\rho\sin\phi\sin\theta)^2}{(\rho\sin\phi\cos\theta)^2}+1\right)}\right)+\frac{\partial}{\partial\phi}\left(\frac{\rho\cos\phi*\rho\sin\phi\sin\theta}{\rho^3*\sqrt{-\frac{(\rho\cos\phi)^2}{\rho^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_y\left(\frac{\partial}{\partial\rho}(\sin\phi\sin\theta)+\frac{\partial}{\partial\theta}\left(\frac{\cos(\theta)}{\rho\sin(\phi)}\right)+\frac{\partial}{\partial\phi}\left(\frac{\cos(\phi)\sin(\theta)}{\rho}\right)\right)$$

And finally, z:

$$\frac{\partial}{\partial z}\boldsymbol{e}_z=\boldsymbol{e}_z\left(\frac{\partial}{\partial r}\frac{\partial r}{\partial z}+\frac{\partial}{\partial\theta}\frac{\partial\theta}{\partial z}+\frac{\partial}{\partial\phi}\frac{\partial\phi}{\partial z}\right)$$

$$\equiv \boldsymbol{e}_z\left(\frac{\partial}{\partial\rho}\left(\frac{\partial}{\partial z}\left(\sqrt{x^2+y^2+z^2}\right)\right)+\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial z}\left(\tan^{-1}\left(\frac{y}{x}\right)\right)\right)+\frac{\partial}{\partial\phi}\left(\frac{\partial}{\partial z}\left(\cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)\right)\right)\right)$$

$$\equiv \boldsymbol{e}_z\left(\frac{\partial}{\partial\rho}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)+\frac{\partial}{\partial\phi}\left(\frac{\frac{1}{\sqrt{x^2+y^2+z^2}}-\frac{z^2}{(x^2+y^2+z^2)^{\frac{3}{2}}}}{\sqrt{-\frac{z^2}{x^2+y^2+z^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_z\left(\frac{\partial}{\partial\rho}\left(\frac{\rho\cos(\phi)}{\rho}\right)+\frac{\partial}{\partial\phi}\left(\frac{\frac{1}{\rho}-\frac{(\rho\cos(\phi))^2}{\rho^3}}{\sqrt{-\frac{(\rho\cos(\phi))^2}{\rho^2}+1}}\right)\right)$$

$$\equiv \boldsymbol{e}_z\left(\frac{\partial}{\partial\rho}(\cos(\phi))-\frac{\partial}{\partial\phi}\left(\frac{\sin(\phi)}{\rho}\right)\right)$$

So now that we have all of this disgust, we can sum everything and group it by the spherical differentials:

$$\nabla=\begin{pmatrix}\frac{\partial}{\partial\rho}\left(\sin\phi\cos\theta\,\boldsymbol{e}_x+\sin\phi\sin\theta\,\boldsymbol{e}_y+\cos(\phi)\,\boldsymbol{e}_z\right)\\ \frac{\partial}{\partial\theta}\left(-\frac{\sin(\theta)}{\rho\sin(\phi)}\boldsymbol{e}_x+\frac{\cos(\theta)}{\rho\sin(\phi)}\boldsymbol{e}_y\right)\\ \frac{\partial}{\partial\phi}\left(\frac{\cos(\phi)\cos(\theta)}{\rho}\boldsymbol{e}_x+\frac{\cos(\phi)\sin(\theta)}{\rho}\boldsymbol{e}_y-\frac{\sin(\phi)}{\rho}\boldsymbol{e}_z\right)\end{pmatrix}$$

We can extract unit vectors from this mess by writing it in the following form:

$$\nabla=\begin{pmatrix}\frac{\partial}{\partial\rho}\left(\sin\phi\cos\theta\,\boldsymbol{e}_x+\sin\phi\sin\theta\,\boldsymbol{e}_y+\cos(\phi)\,\boldsymbol{e}_z\right)\\ \frac{\partial}{\partial\theta}\left(\frac{1}{\rho\sin(\phi)}\left(-\sin(\theta)\,\boldsymbol{e}_x+\cos(\theta)\,\boldsymbol{e}_y\right)\right)\\ \frac{\partial}{\partial\phi}\left(\frac{1}{\rho}\left(\cos(\phi)\cos(\theta)\,\boldsymbol{e}_x+\cos(\phi)\sin(\theta)\,\boldsymbol{e}_y-\sin(\phi)\,\boldsymbol{e}_z\right)\right)\end{pmatrix}$$

This simplifies to:

$$\nabla=\begin{pmatrix}\frac{\partial}{\partial\rho}\boldsymbol{e}_\rho\\ \frac{\partial}{\partial\theta}\frac{1}{\rho\sin(\phi)}\boldsymbol{e}_\theta\\ \frac{\partial}{\partial\phi}\frac{1}{\rho}\boldsymbol{e}_\phi\end{pmatrix}=\left\langle\frac{\partial}{\partial\rho},\frac{1}{\rho\sin(\phi)}\frac{\partial}{\partial\theta},\frac{1}{\rho}\frac{\partial}{\partial\phi}\right\rangle$$

Nice that only took me 2.5 hours even WITH Maple.

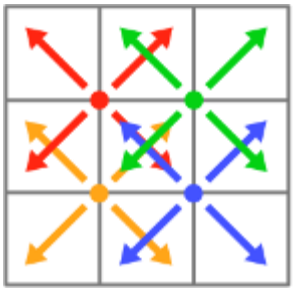
SO COOL we can now get the divergence (and curl and gradient if I'm being thorough) in spherical coordinates:

$$div\left(\boldsymbol{c}_t(\rho,\theta,\phi)\right)=\nabla\cdot\boldsymbol{c}_t=\frac{1}{\rho\sin(\phi)}\frac{\partial}{\partial\theta}\left(c_{t_\theta}(\rho,\theta,\phi)\right)+\frac{1}{\rho}\frac{\partial}{\partial\phi}\left(c_{t_\phi}(\rho,\theta,\phi)\right)$$

Written in discretized form:

$$\nabla\cdot\boldsymbol{c}_t=\frac{1}{\rho\sin(\phi)}*\frac{c_{t_\theta}(\rho,\theta+\delta\theta,\phi)-c_{t_\theta}(\rho,\theta-\delta\theta,\phi)}{2\delta\theta}+\frac{1}{\rho}*\frac{c_{t_\phi}(\rho,\theta,\phi+\delta\phi)-c_{t_\phi}(\rho,\theta,\phi-\delta\phi)}{2\delta\phi}$$

Karl Sims offers a method for evaluating and removing divergence using a 3x3 convolution kernel. Consider a 2D 3x3 grid with grid size ϵ of discretized velocity vectors (discretized such that $F[x,y]=\boldsymbol{V}(x,y)$) centered at (x_0,y_0) as shown below. The local divergence at each corner is defined as the normalized sum of components in the 4 neighboring cells along the diagonal unit vectors:



We will now evaluate this for each point:

Green Point (g):

$$\text{div}(g) = \frac{1}{4\sqrt{2}}(F[x_0+1, y_0+1] \cdot \langle 1,1\rangle + F[x_0, y_0+1] \cdot \langle -1,1\rangle + F[x_0, y_0] \cdot \langle -1,-1\rangle + F[x_0+1, y_0] \cdot \langle 1,-1\rangle)$$

Red Point (r):

$$\text{div}(r) = \frac{1}{4\sqrt{2}}(F[x_0, y_0+1] \cdot \langle 1,1\rangle + F[x_0-1, y_0+1] \cdot \langle -1,1\rangle + F[x_0-1, y_0] \cdot \langle -1,-1\rangle + F[x_0, y_0] \cdot \langle 1,-1\rangle)$$

Orange Point (o):

$$\text{div}(o) = \frac{1}{4\sqrt{2}}(F[x_0, y_0] \cdot \langle 1,1\rangle + F[x_0-1, y_0] \cdot \langle -1,1\rangle + F[x_0-1, y_0-1] \cdot \langle -1,-1\rangle + F[x_0, y_0-1] \cdot \langle 1,-1\rangle)$$

Blue Point (b):

$$\text{div}(b) = \frac{1}{4\sqrt{2}}(F[x_0+1, y_0] \cdot \langle 1,1\rangle + F[x_0, y_0] \cdot \langle -1,1\rangle + F[x_0, y_0-1] \cdot \langle -1,-1\rangle + F[x_0+1, y_0-1] \cdot \langle 1,-1\rangle)$$

Note that the scaling factor proceeding each expression is the result of normalization. We want each divergence to have an "effective magnitude" of 1, similarly to the dot product in the original divergence calculation. To achieve this, we first normalize the diagonal vectors (divide by a factor of $\sqrt{2}$) and then normalize the number of components evaluated (divide by a factor of 4).

We can use these points to evaluate the gradient of the divergence at the point of interest. Remember, the goal is to make the gradient in divergence 0, so we will increment the velocity at the point of interest by the calculated gradient. We evaluate the gradient as follows:

$$\nabla(\nabla \cdot \boldsymbol{V}) \approx \frac{1}{2\epsilon\sqrt{2}}\langle \text{div}(g) + \text{div}(b) - \big(\text{div}(r) + \text{div}(o)\big), \text{div}(g) + \text{div}(r) - \big(\text{div}(o) + \text{div}(b)\big)\rangle$$

Note that the scaling factor is once again included to normalize the result. Each divergence calculation has a "weight" of 1, and so we take the average of the two by dividing by a factor of 2. We then normalize the result (divide by the magnitude of $\sqrt{2}$) and then complete the numerical evaluation by dividing by the distance through which this difference occurs (divide by the factor of ϵ).

Written in terms of the discretized variables we have, we get:

$$\begin{aligned}\nabla(\nabla \cdot \boldsymbol{V}) = \frac{1}{16\epsilon}\big((F[x_0+1, y_0+1] + F[x_0-1, y_0-1]) \cdot \langle 1,1\rangle * \langle 1,-1\rangle + (F[x_0-1, y_0+1] + F[x_0+1, y_0-1]) \cdot \langle 1,-1\rangle * \langle 1,-1\rangle \\ + \langle 2,-2\rangle(F[x_0-1, y_0] + F[x_0+1, y_0] - F[x_0, y_0+1] - F[x_0, y_0-1]) - 4F[x_0, y_0]\big)\end{aligned}$$

Now ϵ isn't actually constant across the image because equirectangular coordinates have distortion. It is constant in the y direction, however, the x epsilon must be determined through angular displacement. We can calculate a pixel angular scale (which is constant in the y direction) using the following:

$$p = \frac{h}{\pi}$$

Where h is the height of the image.

We can then use this scale to determine the components of ϵ as follows:

$$\epsilon_y = r_{dyn} * p$$

$$\epsilon_x = r_{dyn} \sin\phi * p$$