The idea is to use Newton's Method to solve for the Eigenvectors of 2x2 matrices. This is done by using the equation from the definition of an Eigenvector, $(A - \lambda I)\vec{v} = \vec{0}$, as well as one to limit the length of the eigenvector, $|\vec{v}|^2 - 1 = 0$. Expressing the matrix A as $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $\vec{v}$ as $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$, the equation to find zeroes of is:

$$f \begin{pmatrix} v_1 \\ v_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} (a-\lambda)v_1 + bv_2 \\ cv_1 + (d-\lambda)v_2 \\ v_1^2 + v_2^2 - 1 \end{pmatrix}$$

The derivative of this matrix is:

$$[Df \begin{pmatrix} v_1 \\ v_2 \\ \lambda \end{pmatrix}] = \begin{bmatrix} a-\lambda & b & -v_1 \\ c & d-\lambda & -v_2 \\ 2v_1 & 2v_2 & 0 \end{bmatrix}$$

The way to test which eigenvalue Newton's Method converges to, is to first construct a matrix with known eigenvalues. This is done by taking a diagonal matrix, for example $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, and then multiplying it on the left by an invertible matrix, and on the right by that matrix's inverse. For the first test the diagonalized basis was arbitrarily determined to be $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, and the matrix to "shuffle" it was $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. In this case, the matrix to find eigenvalues of is $\begin{bmatrix} -1 & 1 \\ -6 & 4 \end{bmatrix}$ For each 2D image, one of the three variables (v_1, v_2, λ) was kept constant, and the values of the other two were varied through a multiple of -1 to 1. For example for the "xy" image with zoom 1, the initial guesses were varied from $\begin{bmatrix} -10 \\ 0 \\ -10 \end{bmatrix}$ to $\begin{bmatrix} 10 \\ 0 \\ 10 \end{bmatrix}$. Each guess is represented as a point on the image, and it is colored depending on which eigenvalue Newton's method converges to. I generated 15 images to start, five zooms (10^{-2} to 10^2)for the three different combinations of variable that were being varied.

Current Problems:
Current image generating code is extremely slow, perhaps having to do with keeping track of a very long string, takes about 4 hours on my laptop to generate 15 images.