Tetrahedron Meshes and Hexahedron Metrics*

Michael Burgher, Qiuchi Shan, Shafieq Aryan

May 2019

*The distribution of work on this project is as follows. Shafieq determined the equations for aspect ratio and the centroid of a tetrahedron. Qiuchi determined equations and wrote code for taper and skew. She also is responsible for the code that plots tetrahedra and hexahedra, inputting the data and visualizing the meshes. Everything else was done by Michael.

1 Abstract

Meshes are partitioned into tetrahedra which are then partitioned into four hexahedra in a standard manner. The distributions of specific hexahedron metrics determine the quality of the mesh. If the quality is low, the question is could a different selection of hexahedron vertices affect the mesh positively? Different vertices could improve one hexahedron and make another worse, so optimizing is the ultimate goal. Helping to begin such a problem was the purpose of this project. Five of sixteen metrics were analyzed. Once the rest are added to the code the optimization can begin.

2 Introduction

The motive before Math Clinic of tetrahedron meshes is to be used as finite element methods to approximate numerical solutions to differential equations. The mesh volume is partitioned into tetrahedra. Each tetrahedron is partitioned into four hexahedra. The hexahedra are specifically cuboids (quadrilateral hexahedra) but for this paper will be described as hexahedra. If certain hexahedron metric distributions show a large amount of hexahedra are within specific ranges, this creates a good mesh. The word large in the previous sentence is intentionally chosen to be vague to represent the optimization problem that would be the eventual goal of continuing this project. For now, the goal of this project was to design a program that will input the coordinates of many tetrahedra, partition them into hexahedra, and analyze the metrics mentioned previously. The output of this program was histograms showing the distributions of the metric values. The impact of this code will be to give the sponsor the ability to see if a design will or will not provide a good mesh.

One way to check the accuracy of the code was to standardize the position, scale and orientation of each tetrahedron. The results will show such transformations do not effect the metric distributions as was expected.

3 Methods

The data for this project were for six meshes. Each mesh had a pair of files. The first file was a list of vertices (up to about 10,000) each with three real numbers representing cartesian coordinates. The second file was a list of tetrahedra (up to about 40,000) each with four vertices whose coordinates can be referenced in the first file. These data were extracted into an excel file and then read into python. Using a loop, each tetrahedron was partitioned into four hexahedra and the metrics were calculated.

The code written for this project begins by analyzing a regular tetrahedron. Then data for six meshes are visualized and analyzed. Histograms of their metric distributions are shown as well. The final part of the code shows examples of hexahedra which have good and bad metric values. The purpose of these visualizations is to further understand conceptually and visually what makes a good and bad hexahedron.

3.1 Tetrahedron Analysis

The process for analyzing a tetrahedron in the project is described in this subsection.

Given a tetrahedron, the vertices will be labeled $\vec{A}$, $\vec{B}$, $\vec{C}$ and $\vec{D}$. Then the two vertices incident with the shortest edge are labeled $\vec{A}$ and $\vec{B}$, with $\vec{A}$ being closer to the origin than $\vec{B}$. $\vec{C}$ and $\vec{D}$ are assigned to the remaining vertices such that the area of the face with vertices at $\vec{A}$, $\vec{B}$ and $\vec{C}$ is less than the area of the face with vertices at $\vec{A}$, $\vec{B}$ and $\vec{D}$. Heron's formula is used to calculate these areas.

Heron's Formula for the area of a triangle with side lengths p, q and r is

$$Area = \sqrt{s(s-p)(s-q)(s-r)}$$

where $s = \frac{p+q+r}{2}$.

Once the vertices have been labeled, a copy of the coordinates is made so the original can be kept while a new tetrahedron is transformed.

3.2 Tetrahedron Transformation

This will describe the manner in which each tetrahedron is transformed. The purpose of this is to show the metric distributions are independent of scale, position and orientation.

The first step of this transformation is to translate the tetrahedron so that $\vec{A}$ lies on the origin. Next, the tetrahedron is scaled so the magnitude of the vector pointing from $\vec{A}$ to $\vec{B}$ is one ($|\vec{AB}| = 1$). Then rotated so that $\vec{B}$ lies on the positive x-axis. Then rotated again so $\vec{C}$ lies within the xy-plane. Let the reader note it is necessary to scale before using the rotation functions listed below and derived in the appendix.

The goal of the first function is to rotate $\vec{B}$ onto the x-axis. This function then needs to be applied to all four vertices. Let the pre-rotated components of $\vec{B} = (b_x, b_y, b_z)$ and the components of the point to be rotated be (x_i, y_i, z_i). The first rotation function used is below.

$$f_1(x_i, y_i, z_i) = \begin{bmatrix} b_x x_i + b_y y_i + b_z z_i \\ \frac{-x_i b_y^2 - x_i b_z^2 + b_x b_y y_i + b_x b_z z_i}{\sqrt{b_y^2 + b_z^2}} \\ \frac{-b_z y_i + b_y z_i}{\sqrt{b_y^2 + b_z^2}} \end{bmatrix}$$

One can see $f_1(0,0,0) = (0,0,0)$ and $f_1(b_x, b_y, b_z) = (1,0,0)$ as is expected.

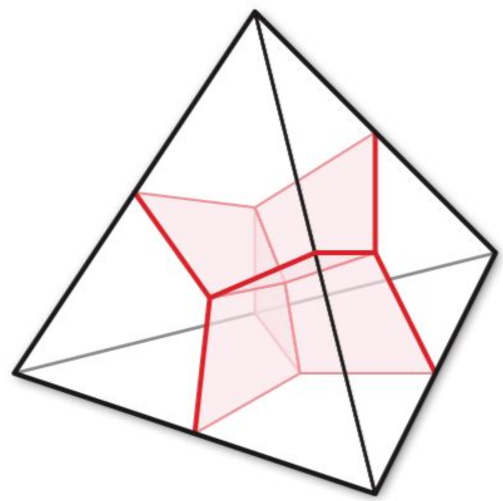
Once the second function has been applied to all four vertices, the z-component of $\vec{C}$ should be zero. Again, let the pre-rotated components of $\vec{C} = (c_x, c_y, c_z)$. Then the next rotation function used is below.

$$f_2(x_i, y_i, z_i) = \begin{bmatrix} x_i \\ \frac{c_y y_i + c_z z_i}{\sqrt{c_y^2 + c_z^2}} \\ \frac{c_y z_i - c_z y_i}{\sqrt{c_y^2 + c_z^2}} \end{bmatrix}$$

One can see $f_2(0,0,0) = (0,0,0)$, $f_2(1,0,0) = (1,0,0)$ and $f_2(c_x, c_y, c_z) = (c_x, \sqrt{c_y^2 + c_z^2}, 0)$ as is expected.

3.3 Hexahedra

This project's standard of partitioning a tetrahedron into four hexahedra follows. Each hexahedron has eight vertices. The 1^{st} vertex in each hexahedron is a vertex of the tetrahedron. Each hexahedron also has 3 vertices which are midpoints of the tetrahedron edges incident to the 1^{st} vertex. Each hexahedron also has 3 vertices at the centroids of the tetrahedron faces sharing a vertex with the 1^{st}. Finally, each hexahedron also has a vertex at the centroid of the tetrahedron.



Theoretical Advances in Hexahedral Mesh Generation Jeff Erickson. University of Illinois, Urbana-Champaign p. 20

Once the four hexahedra have been determined, the program analyzes the following five hex metrics: aspect ratio, skew, taper, stretch and diagonal ratio. While consistency has been attempted, the reader should allow the notation of each metric to be independent.

3.3.1 Hex Metric 1: Aspect Ratio

Let $s = \max\{\text{distances from hex centroid to vertices}\}$

Let $t = \min\{\text{distances from hex centroid to face centroids}\}$

Then Aspect Ratio $= \frac{s}{t}$

3.3.2 Hex Metric 2: Skew

Let $\vec{c}$ be the centroid of a hexahedron with two distinct vertices $\vec{u}$ and $\vec{v}$. Let the components of these vectors be listed as below.

$$\vec{c} = (c_x, c_y.c_z)$$

$$\vec{u} = (u_x, u_y, u_z)$$

$$\vec{v} = (v_x, v_y, v_z)$$

Let the vectors beginning at the centroid and extending to $\vec{u}$ and $\vec{v}$ be denoted as $\vec{cu}$ and $\vec{cv}$.

Let $\alpha_{u,v}$ be the smaller angle separating $\vec{cu}$ and $\vec{cv}$.

Then Skew $= \max\{|\cos(\alpha_{u,v})|\}$

The program calculated the cosine by solving the following function for cosine.

$$\vec{cu} \cdot \vec{cv} = |\vec{cu}||\vec{cv}| \cos(\alpha_{u,v})$$

3.3.3 Hex Metric 3: Taper

Taper is the maximum ratio of the lengths of two edges of the hexahedron which share no common vertex.

3.3.4 Hex Metric 4: Stretch

Let $s = \min\{\text{edge lengths}\}$

Let $t = \max\{\text{body diagonal lengths}\}$

Then Stretch $= \frac{\sqrt{3}s}{t}$

3.3.5 Hex Metric 5: Diagonal Ratio

Let $s = \min\{\text{body diagonal lengths}\}$

Let $t = \max\{\text{body diagonal lengths}\}$

Then Diagonal Ratio $= \frac{s}{t}$

4 Results

This section is broken into 3 parts. No uncertainty analysis is done in this project. It seems it would only be necessary due to computer round-off error. So far, it has been assumed it is small enough to ignore.

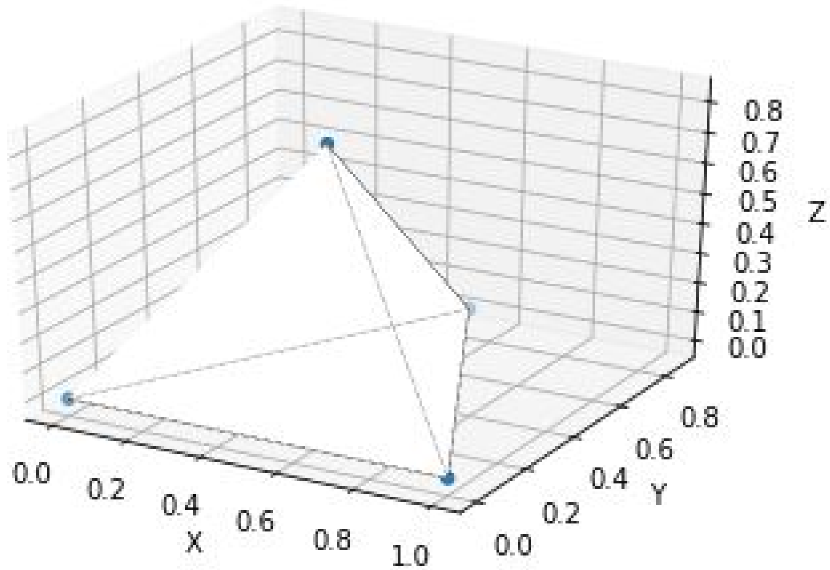
4.1 shows the metric results for hexahedra partitioned from a regular tetrahedron. All metric results besides taper are within acceptable ranges.

4.2 shows the six different meshes. These include pictures of the meshes as well as histograms of the five tested metrics showing their distributions. These histograms were the main goal of this project. They allow the sponsor to see how good a mesh is. The black dotted lines on the histograms represent the acceptable metric ranges.

It is surprising to see how few hexahedra are in acceptable ranges, especially skew and taper. This is discussed more in the next section but taper requires another note. Seeing these histograms, observe that not a single hexahedron has a good taper. As a result, the range of taper was chosen in two different manners. Meshes 1 and 2 show a range of 0-2 and emphasize that no hexahedra with good taper are produced. Meshes 3-6 show the full range of taper to show how far out of the acceptable range some of these are.

4.3 shows visualizations of some different hexahedra. Each metric gets one page. On each page is repeated the metric from mesh 1. Additionally, each page shows 5 hexahedra that produced good (when possible) results and 5 that produced bad. Each of these pictures have the same range and orientation. Given this range and orientation, graphs that could be clearly seen were chosen.

4.1 Regular Tetrahedron



When a regular tetrahedron is partitioned into 4 hexahedra in the manner described in this paper, all 5 metric values are the same for each hexahedron. These values are

Aspect Ratio: 2.847

Skew: 0.498

Taper: 2.449

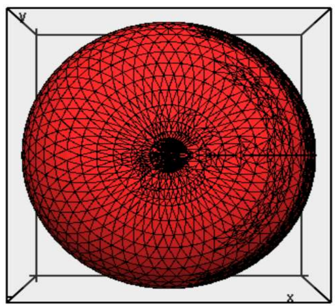
Stretch: 0.577

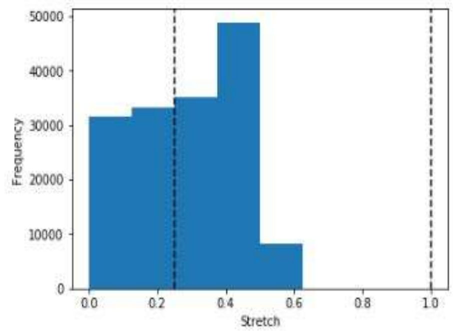
Diagonal Length: 0.816

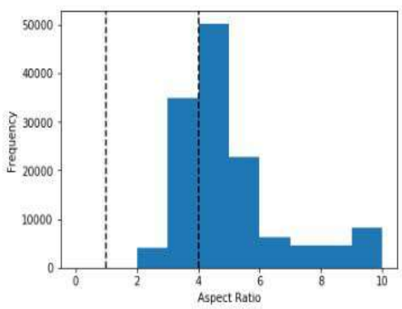
4.2 Mesh Visualization and Metric Distributions

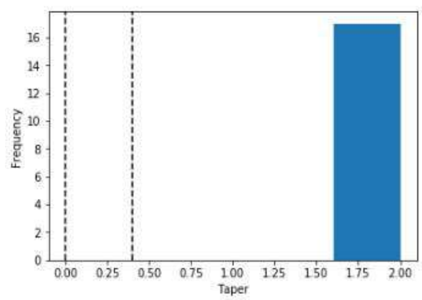
Object # 1

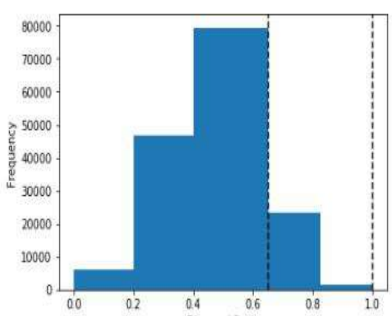
39,262 Tetrahedra
157,048 Hexahedra

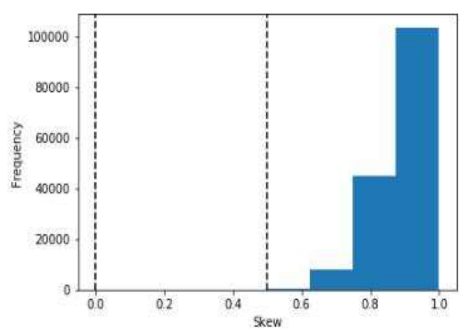






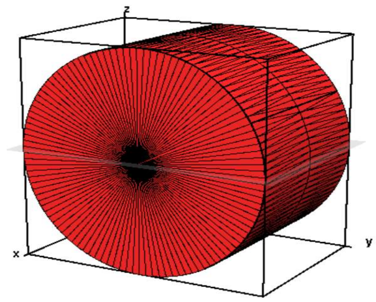


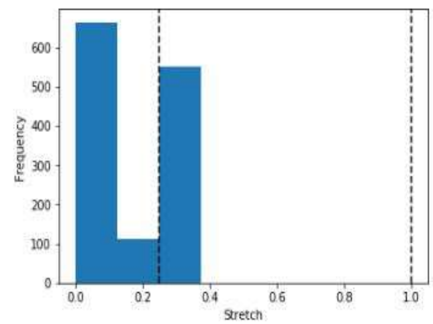


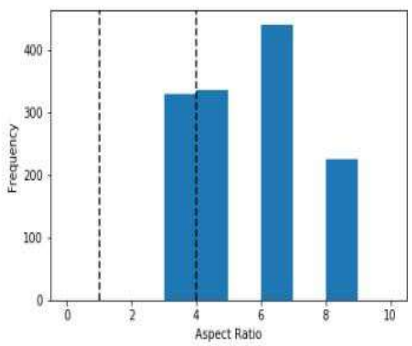


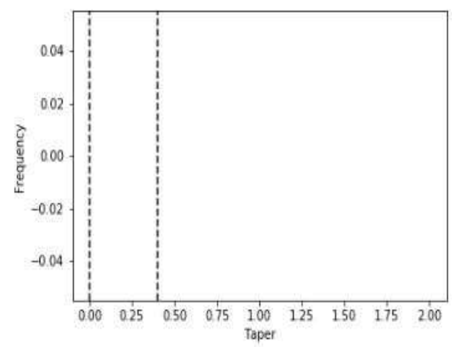
Object # 2

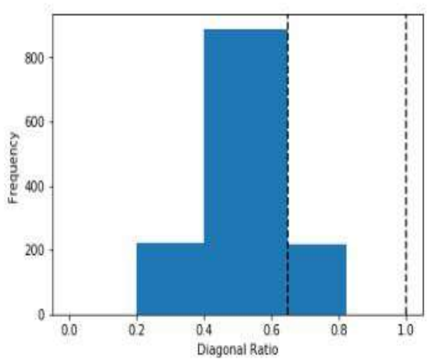
332 Tetrahedra
1,327 Hexahedra

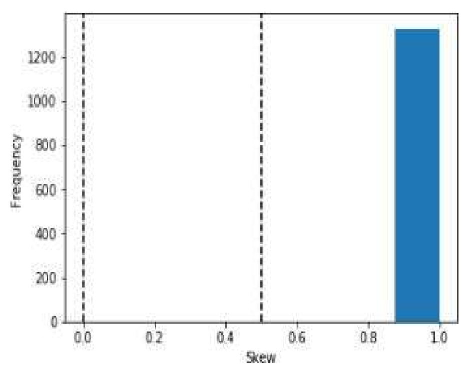






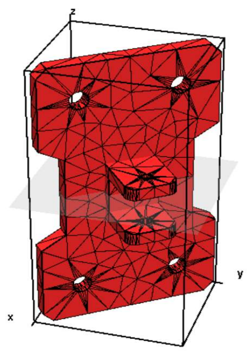


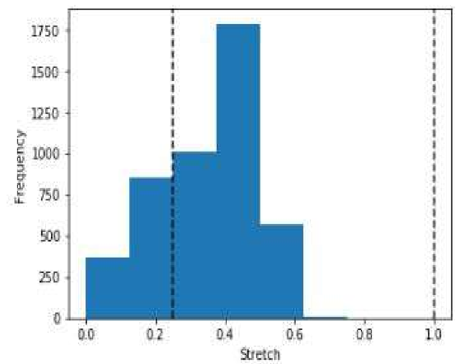


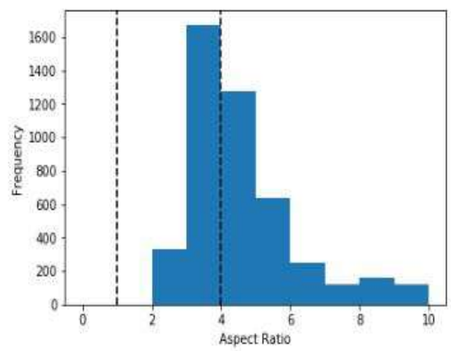


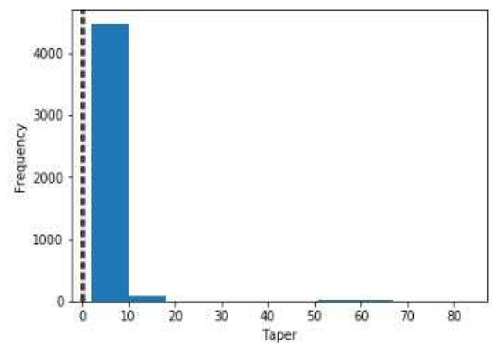
Object # 3

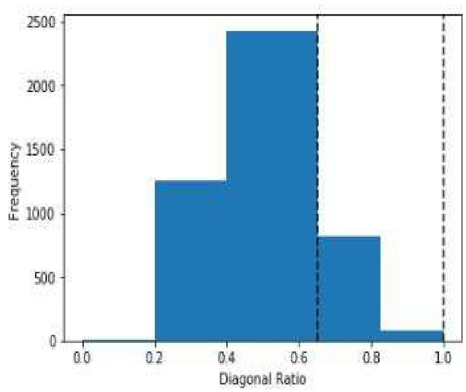
1,151 Tetrahedra
4,604 Hexahedra

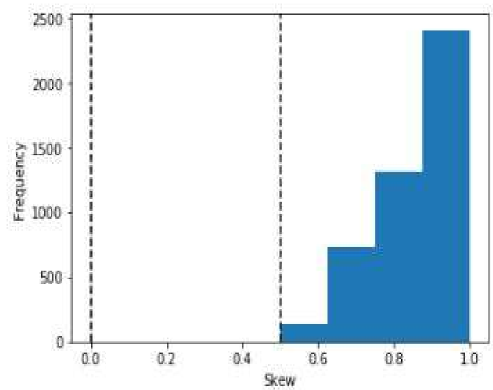






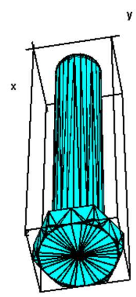


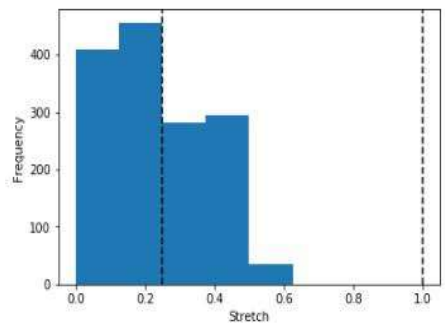


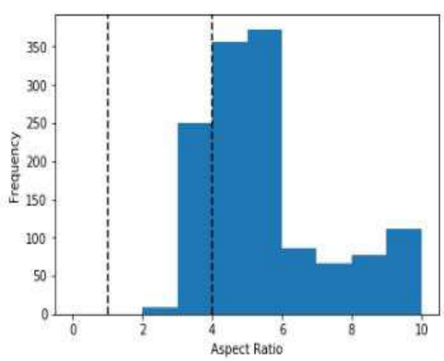


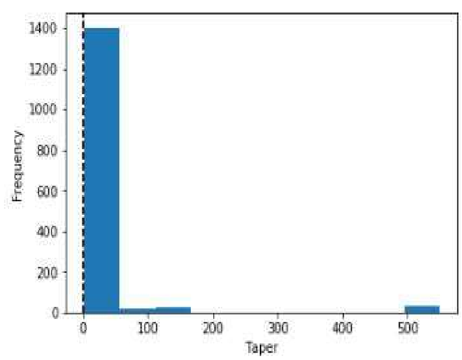
Object # 4

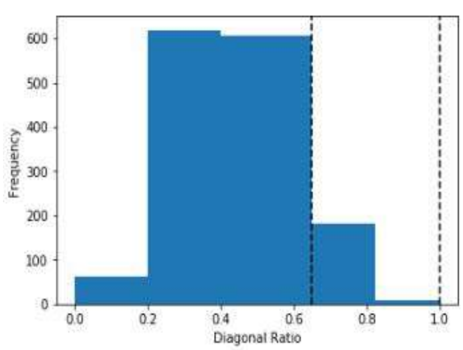
369 Tetrahedra
1,476 Hexahedra

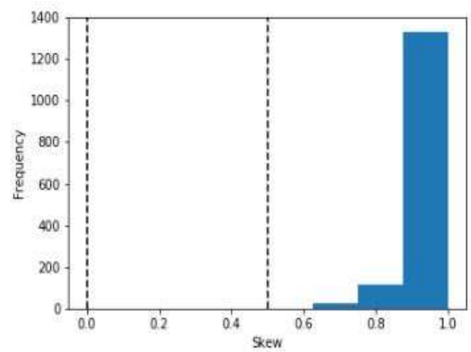






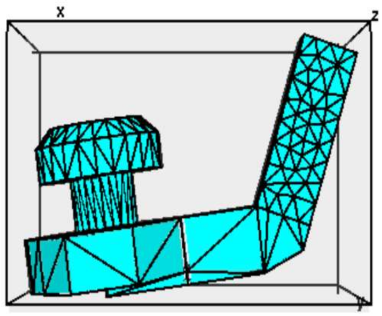


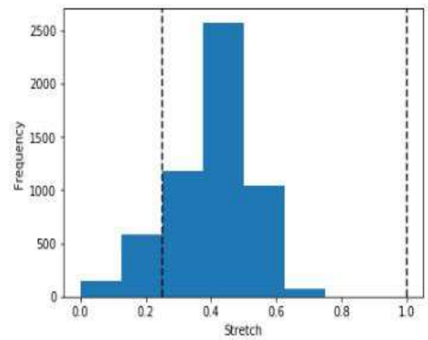


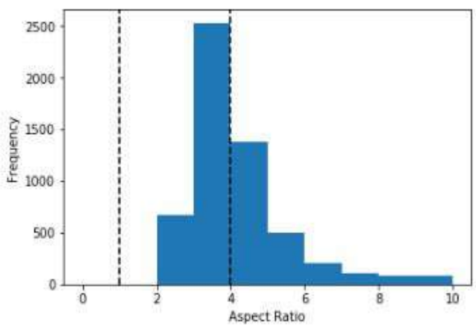


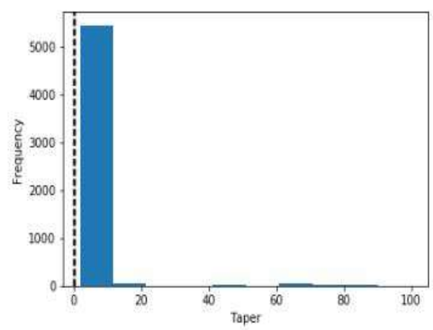
Object # 5

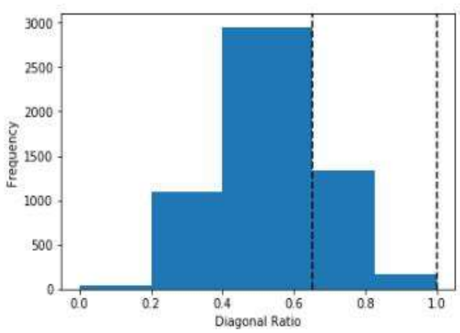
1,399 Tetrahedra
5,596 Hexahedra

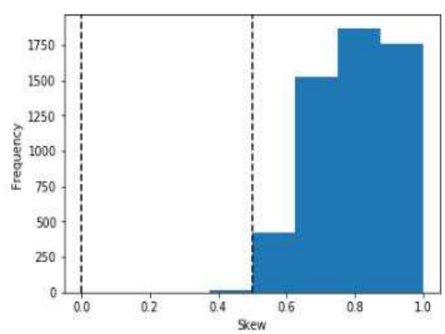






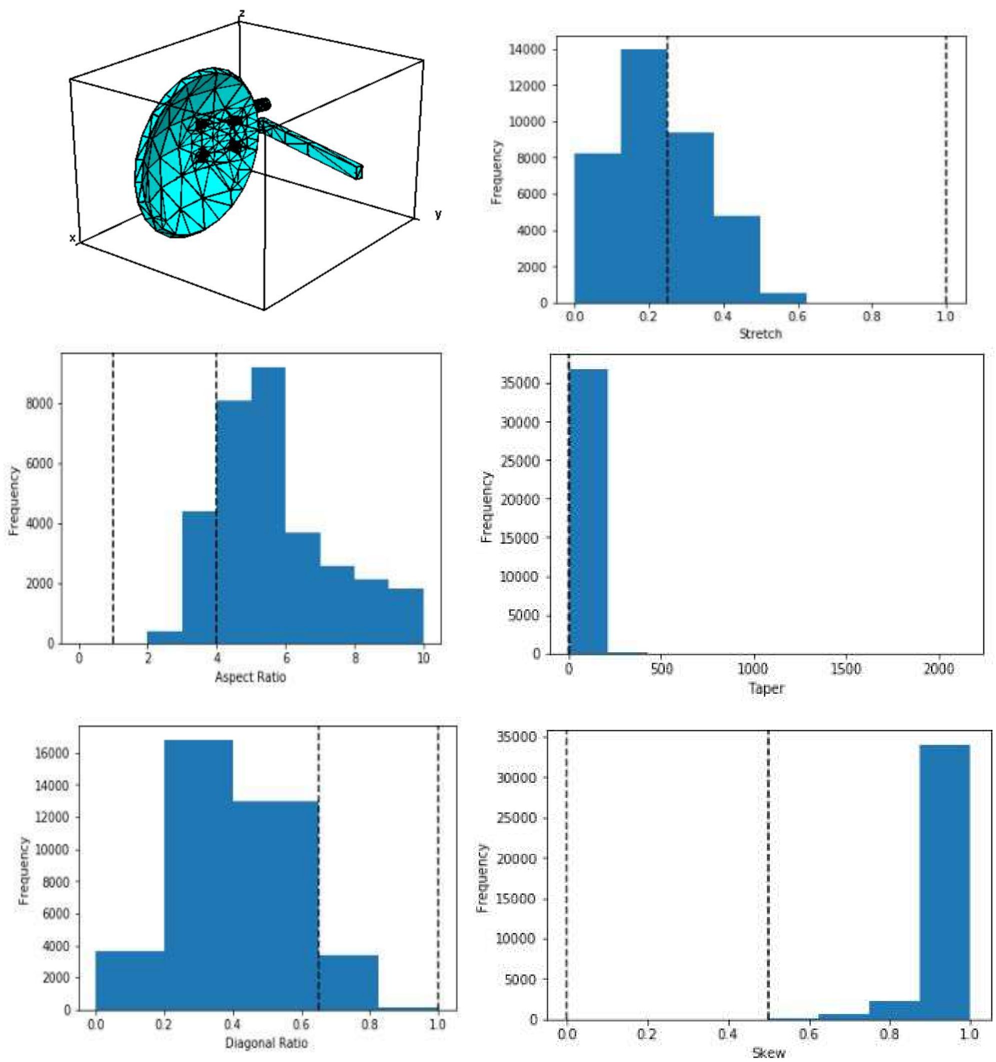






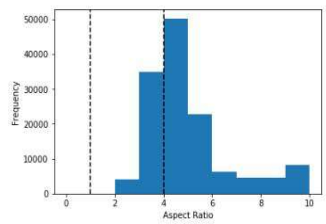
Object # 6

9,236 Tetrahedra
36,944 Hexahedra



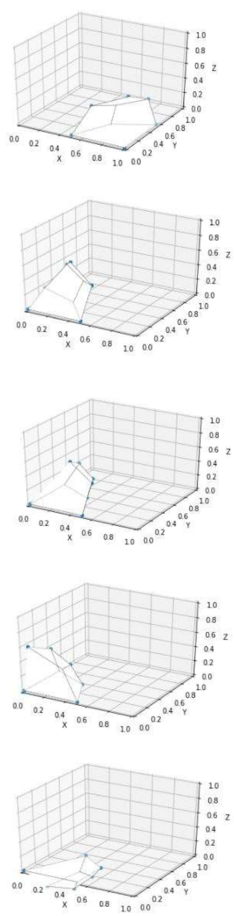
4.3 Visualizations of Good and Bad Hexahedra

Aspect Ratio

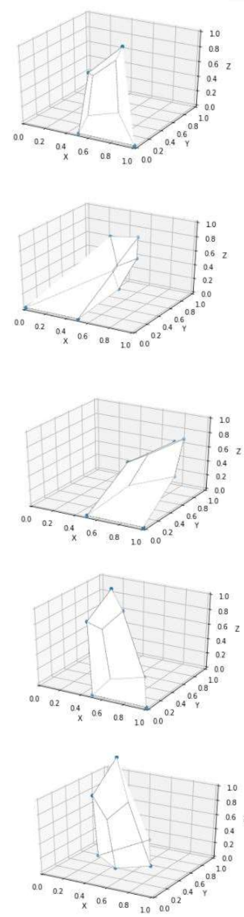


Acceptable Range: *1 - 4*

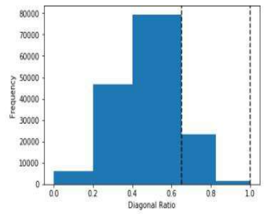
Good: *2.4 - 2.5*



Bad: < *1* or > *4*

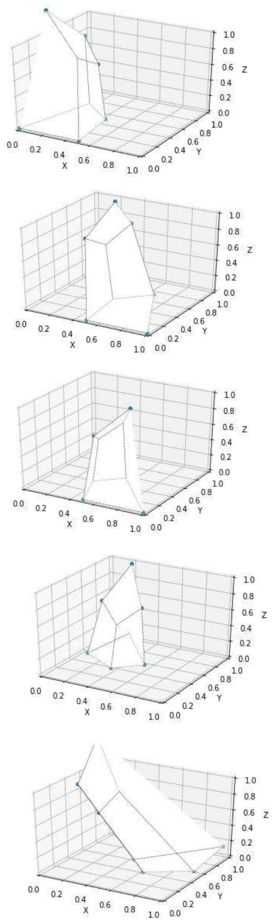


Diagonal Ratio

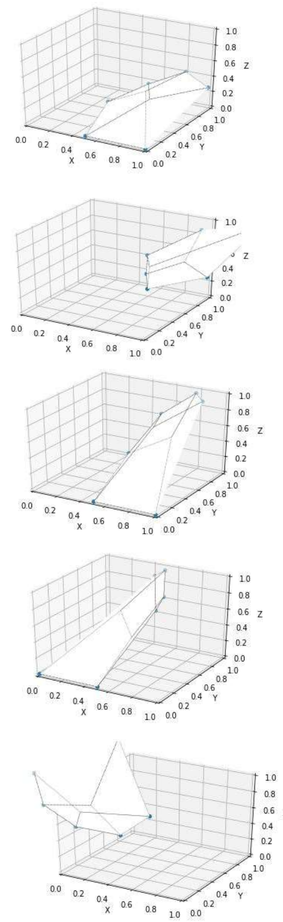


Acceptable Range: *0.65 - 1*

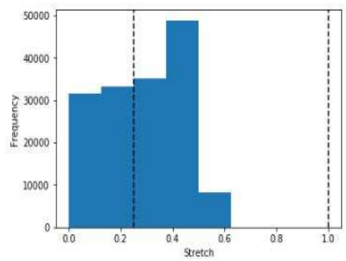
Good: 0.65 – 1



Bad: 0 – 0.3

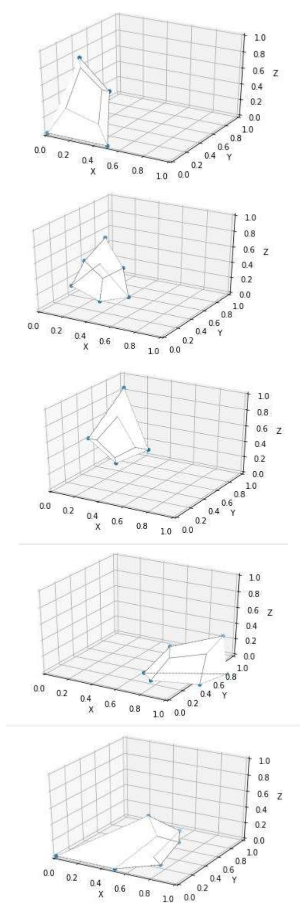


Stretch

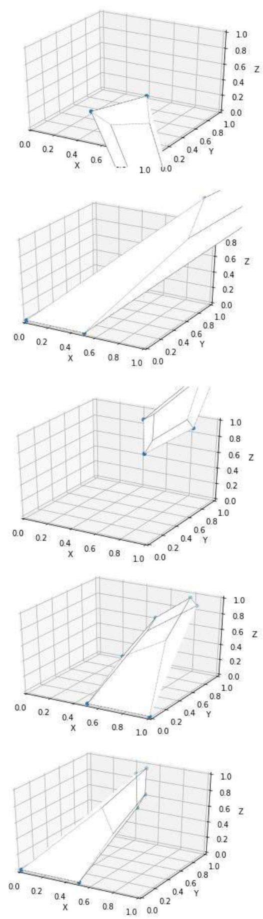


Acceptable Range: *0.25 - 1*

Good: 0.4 – 1

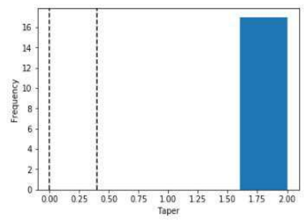


Bad: 0 – 0.25



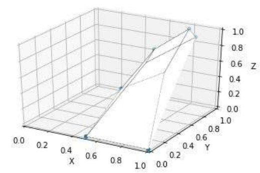
Taper

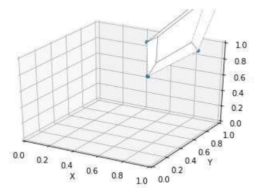
Acceptable Range: *0 - 0.4*

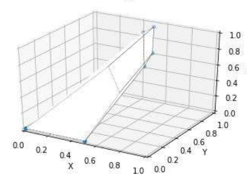


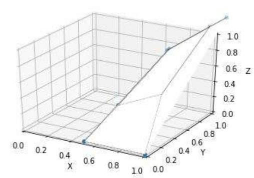
Good: 0 - 0.4 (no pictures available)

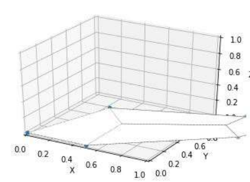
Bad: 0.9 – 1



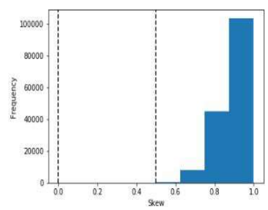






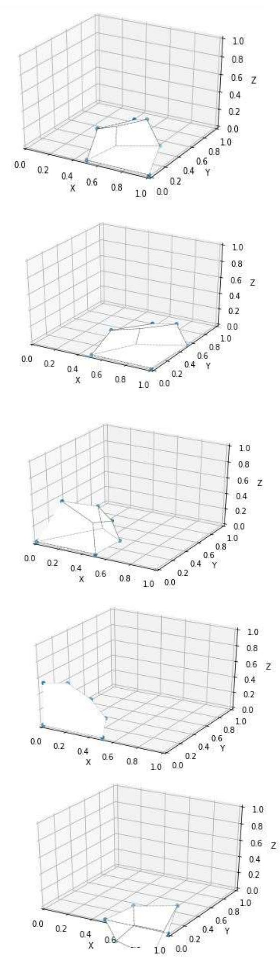


Skew

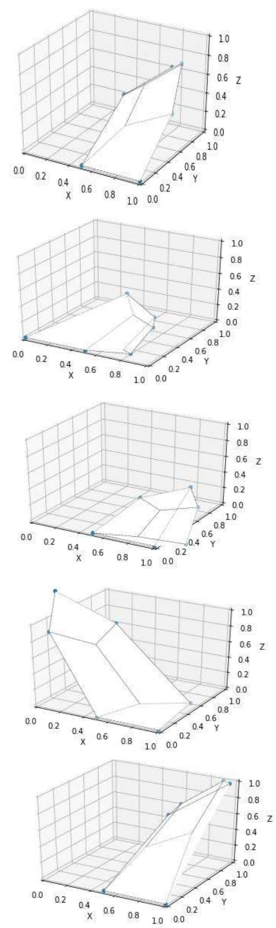


Acceptable Range: *0 - 0.5*

Good: 0 - *0.5*



Bad: *0.9 - 1*



5 Discussion

First, we would have guessed that a larger portion of hexahedra would have good results. Of course one possible explanation is that our code needs to be double checked. Another is that we are just measuring a binary result for each metric. Perhaps we should be measuring how good each hex is rather than if it is good enough. To do this, of course one would need to decide how to measure how good the entire hexahedron is. Still another explanation could be that tetrahedron meshing is difficult and the results are as good as one should expect. Whatever the explanation, taper and skew have particularly low results.

5.1 Metrics

In all six meshes, not a single hexahedron had a taper in the acceptable range. The reason for this is actually quite clear, but the solution is not. The definition of taper as listed in section 3.3.3 is the maximum ratio of the lengths of two edges of the hexahedron which share no common vertex. Any two edges in consideration have a ratio that is less than 1 and a ratio that is greater than 1. The code considers both and takes the max of all possibilities so no hex will ever produce a taper of less than 1 as it is currently calculated. This will of course mean it will never be in the acceptable range of $[0, 0.4]$.

One suggestion was that the code should only consider the ratios that are less than 1. The code is easily adaptable to do this. They python function $metricTaper$ has a line that has been commented but can be switched with the next to apply this suggestion. However, it is likely this is still not correct. If it were, the total range of taper would be $[0, 1]$, but it is listed as $[0, \infty)$.

Skew is the maximum absolute value of the cosine of the angle between the vectors extending from the hexahedron centroid to two distinct vertices. Because cosine is bounded between -1 and 1, Skew will be bounded between 0 and 1. It can be shown (and is in the appendix) that any hexahedron partitioned from a tetrahedron in the manner described in the this report will have skew of 1. This happens because the two vertices of the hexahedron that are the tetrahedron centroid and vertex are co-linear with the hexahedron centroid. This will give a cosine of -1, the absolute value of which is the maximum possibility for skew.

One suggestion was to ignore the absolute value sign. The histograms shown in the results section applied this suggestion. The code is easily adaptable to disregarding this suggestion. The python function $metricSkew$ contains a few lines that are commented which will add the absolute value back.

Another suggestion is to keep the absolute value but ignore the anticipated value of -1. This would allow a user to search for other values of -1. This would be easily done by adding a few lines of code to the same function. There is commented code which will exclude all values of -1 and can easily have another few lines of code to only exclude the anticipated value of -1. This anticipated case is when the 1^{st} and 8^{th} vertices are chosen (of course, this means their python indices are 0 and 7).

5.2 Regular Tetrahedron

The regular tetrahedron was specifically chosen to analyze because it should be the best tetrahedron in terms of hex metrics. We can see form section 4.1 it will have the following values

$$\text{Aspect Ratio of 2.847 within the acceptable range of } [1, 4]$$

$$\text{Skew of 0.498 barely within the acceptable range of } [0, 0.5]$$

$$\text{Taper of 2.449 which is not within the acceptable range of } [0, 0.4]$$

$$\text{Stretch of 0.577 within the acceptable range of } [0.25, 1]$$

$$\text{And diagonal ratio of 0.816 within the acceptable range of } [0.65, 1]$$

The fact that taper is so far off even in a regular tetrahedron is confirmation that the definition listed in this paper needs more understanding before one should draw any conclusions from the taper histograms. Skew is so close to the edge of the acceptable range it suggests there likely exists a better strategy than getting rid of the absolute value sign.

5.3 Tetrahedron Transformation

With few exceptions, the results of this project show translating, scaling, and rotating a tetrahedron will not affect the hexahedron metrics tested thus far. However, about 15 out of 40,000 tetrahedra are worth mentioning. Each one had a division by zero error and gave results of nan for all four hexahedra on every metric.

The most likely explanation is in the first rotation function. If $\vec{B}$ already lies on the x-axis, there will be a division by zero error. This should not be an issue with the second function because $\vec{C}$ should not be on the x-axis (that would mean three vertices of the tetrahedron are co-linear). This was discovered late in this project and efforts to confirm this have yet to be undertaken. Assuming this is correct, the code is easily adaptable. One would need only to add an if statement skipping the first rotation if $\vec{B}$ already lies on the x-axis. It was originally suggested that the error was caused when the four vertices of the tetrahedron were co-planar. This does not seem to be the case but could be double checked if the first explanation fails.

5.4 Other Partitions Considered

One vertex of each hexahedron is at the centroid of the tetrahedron, three are centroids of faces and three are midpoints of edges. It should be noted that triangles and tetrahedra have different centers besides the centroids. The centroids were chosen because unlike some other centers, they will always be within a tetrahedron or triangle and because they are simple formulas (the arithmetic mean of the vertex components). The midpoints seemed like a logical choice.

In fact, for the optimization portion of this project which would come farther down the road, the choices do not need to be centers or midpoints at all. They can be any point in the volume of the tetrahedron and any point in the plane of a face and any point along an edge. Determining the best choice of partition would be difficult. Among other reasons, one could easily change a vertex to improve the quality of one hexahedron, but it could (and I believe likely would) decrease the quality of another hexahedron. This optimization would be the ultimate goal this project attempts to begin.

6 Conclusion

If more time were given on this project, several more tasks would be accomplished. First, more metrics would be taken into consideration. The rest, save dimension, are all dependent on the Jacobian matrix so understanding that would be a top priority.

Also, confirmation on current metrics measured should be attained. Skew and taper clearly have work. Multiple suggestions have been made but it seems that efforts to ensure accuracy should be made rather that picking whichever one seems best. Aspect ratio seems clear but it never hurts to double check anything. Stretch and diagonal ratio have diagonal lengths in their definitions. It seems that these should be body diagonals but there are also face diagonals one could consider. We do not believe they should be, but this again could be confirmed.

At the beginning of this project there was discussion of not only analyzing current meshes, but also providing advice on how to create a good mesh in the future. Obviously this task would be dependent upon analyzing more metrics. However, even with only the five that were analyzed it would be difficult.

The difficulty in creating a good mesh is threefold. First, a single metric (like taper) could be the max of a set with dozens of values. There would be a very large amount of conditions necessary to describe a tetrahedron which produces 4 hexahedra meeting all metrics. This leads to the next difficulty which is the fact that a mesh which partitions a volume must be created. These are not single tetrahedra being created and are being measured on a range. This would seem to make creating better meshes even more difficult to analyze. This leads to the final reason such advise would be difficult. Any such advise would probably have to be purely mathematical. It does not seem (at first glance) to be easy to visualize when a hexahedron would have a good or bad metric value.

Thus, it seems the optimization problem would not have simple analytic solutions. I do not know much on the subject of optimization, only the basics. If it were a finite function (which it is not) one could simply check all the function values. If it were a differentiable function of one variable (which it is not) one could set the first derivative equal to zero and find roots. At first glance, it seems this would be a function of 17 variables (3 for the tetrahedron centroid, 8 for the face centroids, 6 for the edges). With this many variables it would be easy to get stuck in a local minimum and not find the optimal solutions. We could probably outline a set of 17 loops which would check all the discrete values of each variable, but it seems like computational power would soon be a problem. In any event, it seems such optimization would be fascinating and challenging.

We end this report by saying how much we appreciate Tech-X for being a sponsor for this class and Dr. Fournier for all the help and advise. We enjoyed this project and learned much. Thank you.

7 Appendix

7.1 Regarding Skew

The purpose of this section explain why according to the definition listed in the paper, any tetrahedron partitioned in the manner this project does, will yield four hexahedra, each with Skew of 1. Recall, from the definition of skew, it is the maximum absolute value of the cosine of the angle formed by the vectors extending from the centroid of the hexahedron to any two distinct vertices.

Because the absolute value of cosine is bounded between 0 and 1, we need to show that any potential hexahedron will have two vertices which will yield an angle of π and therefore a cosine of -1. The absolute value of which will be the max value used for a skew of 1. This will happen because two of the vertices of the hexahedron will always be co-linear with the centroid of the hexahedron.

These two vertices are the tetrahedron centroid and the tetrahedron vertex. We need to show these two points are always co-liner with the hexahedron centroid. Let an arbitrary tetrahedron have vertices A, B, C, D with components $A = (a_x, a_y, a_z), B = (b_x, b_y, b_z), C = (c_x, c_y.c_z), D = (d_x, d_y, d_z)$. Without loss of generality, consider the hexahedron sharing vertex A with the tetrahedron. This hexahedron has the following vertices:

$$v_1 = (a_x, a_y, a_z)$$

$$v_2 = (\frac{a_x+b_x}{2}, \frac{a_y+b_y}{2}, \frac{a_z+b_z}{2})$$

$$v_3 = (\frac{a_x+c_x}{2}, \frac{a_y+c_y}{2}, \frac{a_z+c_z}{2})$$

$$v_4 = (\frac{a_x+d_x}{2}, \frac{a_y+d_y}{2}, \frac{a_z+d_z}{2})$$

$$v_5 = (\frac{a_x+b_x+c_x}{3}, \frac{a_y+b_y+c_y}{3}, \frac{a_z+b_z+c_z}{3})$$

$$v_6 = (\frac{a_x+b_x+d_x}{3}, \frac{a_y+b_y+d_y}{3}, \frac{a_z+b_z+d_z}{3})$$

$$v_7 = (\frac{a_x+c_x+d_x}{3}, \frac{a_y+c_y+d_y}{3}, \frac{a_z+c_z+d_z}{3})$$

$$v_8 = (\frac{a_x+b_x+c_x+d_x}{4}, \frac{a_y+b_y+c_y+d_y}{4}, \frac{a_z+b_z+c_z+d_z}{4})$$

The centroid of a hexahedron is the arithmetic mean of the vertex components. This hexahedron has a the following centroid.

$$h_c = \tfrac{17}{96}(\tfrac{45}{17}a_x + b_x + c_x + d_x, \tfrac{45}{17}a_x + b_x + c_x + d_x, \tfrac{45}{17}a_x + b_x + c_x + d_x)$$

The vector originating at h_c and extending to v_1 is below.

$$\vec{h_c v_1} = \tfrac{17}{96}(-3a_x + b_x + c_x + d_x).$$

The vector originating at h_c and extending to v_8 is below.

$$\vec{h_c v_8} = -\tfrac{7}{96}(-3a_x + b_x + c_x + d_x).$$

So, one can see that $\vec{h_c v_8} = -\frac{7}{17}\vec{h_c v_1}$. From this, one can see the points h_c, v_1 and v_8 will all be co-linear and taper will be 1.

7.2 Rotation Functions

This section contains the derivations for the rotation functions used. Please allow for independent notation in each subsection.

7.2.1 Rotation Function 1: Rotation about an arbitrary axis onto the x-axis

This function should rotate a vector in $\mathbb{R}^3$ whose norm is 1 (recall the tetrahedron had been scaled such) onto the x-axis. Call the components of this vector $\vec{r} = (x, y, z)$. We will do this using two rotation matrices. Here, $\vec{r}$ is playing the role of $\vec{B}$ on the tetrahedron.

The first matrix will rotate $\vec{r}$ about the x-axis into the xy-plane so that so that $z = 0$. The rotation matrix about the x-axis is by a an angle whose tangent is $\frac{z}{y}$ is shown below.

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{y}{\sqrt{y^2+z^2}} & \frac{z}{\sqrt{y^2+z^2}} \\ 0 & -\frac{z}{\sqrt{y^2+z^2}} & \frac{y}{\sqrt{y^2+z^2}} \end{bmatrix}$$

There are two angles in $[0, 2\pi)$ whose tangent is $\frac{z}{y}$. When this matrix is in terms of the components, it naturally makes the correct selection.

The result of multiplying (x, y, z) with $R_x(\psi)$ is listed below.

$$R_x(\psi)\begin{bmatrix}x\\y\\z\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & \frac{y}{\sqrt{y^2+z^2}} & \frac{z}{\sqrt{y^2+z^2}}\\ 0 & -\frac{z}{\sqrt{y^2+z^2}} & \frac{y}{\sqrt{y^2+z^2}}\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} = \begin{bmatrix}x\\ \sqrt{y^2+z^2}\\ 0\end{bmatrix}$$

Now that $\vec{r}$ has been rotated into the xy-plane, it should be rotated about the z-axis onto the x-axis. The degree of this rotation is an angle whose tangent is $\frac{\sqrt{y^2+z^2}}{x}$

The rotation matrix about the z-axis, recalling the scaling of $\vec{r}$ is shown below.

$$R_z(\theta) = \begin{bmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}\frac{x}{\sqrt{x^2+y^2+z^2}} & \sqrt{\frac{y^2+z^2}{x^2+y^2+z^2}} & 0\\ -\sqrt{\frac{y^2+z^2}{x^2+y^2+z^2}} & \frac{x}{\sqrt{x^2+y^2+z^2}} & 0\\ 0 & 0 & 1\end{bmatrix} =$$

$$\begin{bmatrix}x & \sqrt{y^2+z^2} & 0\\ -\sqrt{y^2+z^2} & x & 0\\ 0 & 0 & 1\end{bmatrix}$$

We can see below that putting $\vec{r}$ through this rotation will give us $(1, 0, 0)$ as is desired.

$$R_z(\theta)\begin{bmatrix}x\\ \sqrt{y^2+z^2}\\ 0\end{bmatrix} = \begin{bmatrix}x & \sqrt{y^2+z^2} & 0\\ -\sqrt{y^2+z^2} & x & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ \sqrt{y^2+z^2}\\ z\end{bmatrix} = \begin{bmatrix}1\\0\\0\end{bmatrix}$$

So the rotation matrices to rotate $\vec{r}$ onto the x-axis have been derived. The next step is to rotate any point in $\mathbb{R}^3$ in the same manner. Let the components of this point be (x_i, y_i, z_i). The function which will input these components and output the rotated components is below.

$$f_1(x_i, y_i, z_i) = R_z(\theta)R_x(\psi)\begin{bmatrix}x_i\\y_i\\z_i\end{bmatrix} =$$

$$\begin{bmatrix}x & \sqrt{y^2+z^2} & 0\\ -\sqrt{y^2+z^2} & x & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \frac{y}{\sqrt{y^2+z^2}} & \frac{z}{\sqrt{y^2+z^2}}\\ 0 & -\frac{z}{\sqrt{y^2+z^2}} & \frac{y}{\sqrt{y^2+z^2}}\end{bmatrix}\begin{bmatrix}x_i\\y_i\\z_i\end{bmatrix} =$$

$$\begin{bmatrix} xx_i + yy_i + zz_i \\ \frac{-x_i y^2 - x_i z^2 + xyy_i + xzz_i}{\sqrt{y^2+z^2}} \\ \frac{-zy_i + yz_i}{\sqrt{y^2+z^2}} \end{bmatrix}$$

One can see that substituting (x_i, y_i, z_i) with (0, 0, 0) will give us a point at the origin as it should. Also, if one substitutes (x_i, y_i, z_i) with (x, y, z) and recalling scaling will end with (1, 0, 0) as it should.

7.2.2 Rotation Function 2: Rotation about the x-axis into the xy-plane

This function should rotate a vector in $\mathbb{R}^3$ onto the xy-plane. Call the components of this vector $\vec{r} = (x, y, z)$. This will be done with a single rotation matrix. Here, $\vec{r}$ is playing the role of $\vec{C}$ on the tetrahedron.

This matrix will rotate $\vec{r}$ about the x-axis into the-xy plane so that so that $z = 0$. The rotation matrix about the x-axis is by an angle whose tangent is $\frac{z}{y}$ is shown below.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{y}{\sqrt{y^2+z^2}} & \frac{z}{\sqrt{y^2+z^2}} \\ 0 & -\frac{z}{\sqrt{y^2+z^2}} & \frac{y}{\sqrt{y^2+z^2}} \end{bmatrix}$$

So the rotation matrices to rotate $\vec{r}$ into the xy-axis have been derived. The next step is to rotate any point in $\mathbb{R}^3$ in the same manner. Let the components of this point be (x_i, y_i, z_i). The function which will input these components and output the rotated components is below.

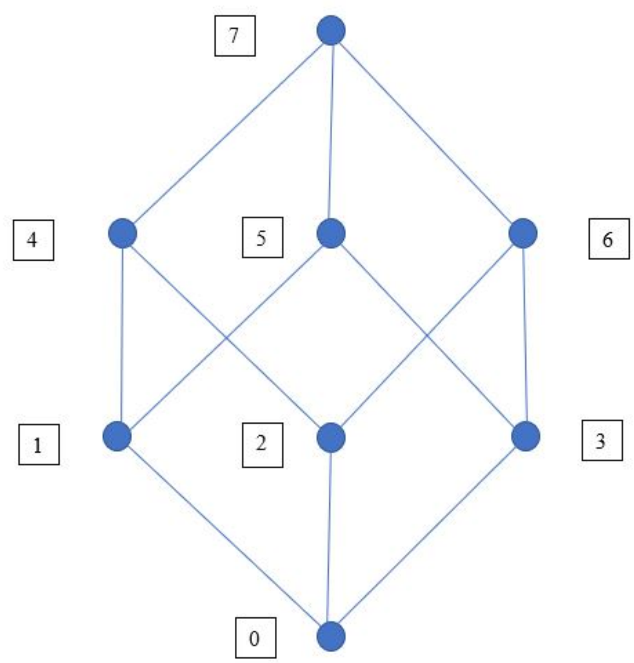
$$f_2(x_i, y_i, z_i) = R_x(\phi) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{y}{\sqrt{y^2+z^2}} & \frac{z^2}{\sqrt{y^2+z^2}} \\ 0 & -\frac{z}{\sqrt{y^2+z^2}} & \frac{y}{\sqrt{y^2+z^2}} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \implies = \begin{bmatrix} x_i \\ \frac{yy_i + zz_i}{\sqrt{y^2+z^2}} \\ \frac{yz_i - zy_i}{\sqrt{y^2+z^2}} \end{bmatrix}$$

And as before, we can see a point at the origin will remain there, a point at

(1, 0, 0) will remain there, and (x, y, z) will rotate to $(x, \sqrt{y^2 + z^2}, 0)$ all as expected.

7.3 Hexahedron Vertex Adjacency

When the code for this project calculates a hexahedron metric, it passes an array of eight arrays of size three to represent the eight cartesian coordinates of the vertices. If the reader needs to know, to continue to use the code, what the order of vertices are, see the picture below. This could help with determining adjacency, diagonals, faces, etc.



7.4 Required Files

a. Main Program: Tech-X.ipynb

b. Input Files

1. Coordinate Location.xlsx
2. Tet Verts.xlsx
3. ele2.xslx
4. ele3.xslx
5. ele4.xlsx
6. ele5.xlsx
7. ele6.xlsx
8. node2.xlsx
9. node3.xlsx
10. node4.xlsx
11. node5.xlsx
12. node6.xlsx

The input files were originally in a different format and data were transferred into excel.