Lauren Yates 10195969

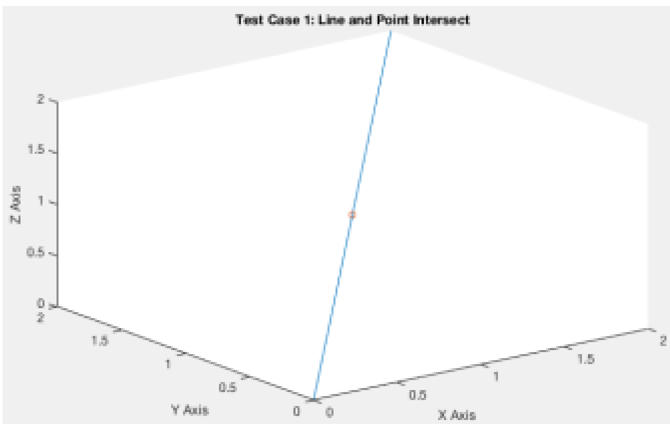
CISC 330 Assignment 1

Part 1- Distance of Line and Point

Description:
A function that calculates the distance between a line and a point.

Test Case 1: The Point and Line Intersect

Test Graph:



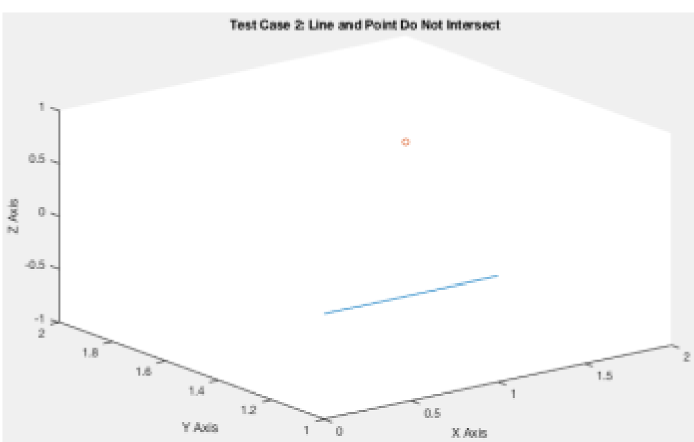
Test Output:

```
dist1 =

     0
```

As seen in the graph above, in test case 1, the distance between the line and point is zero because the line and point intersect.

Test Case 2: The Point and Line Do Not Intersect

Test Graph:



Test Output:

```
dist2 =

     1
```

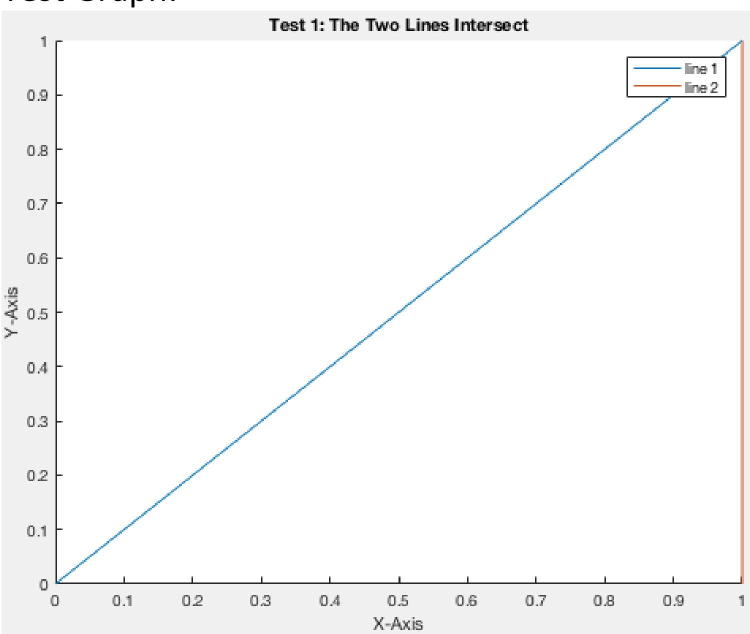
As seen in the graph above, in test case 2, the distance between the line and point is not zero because the line and point do not intersect.

Part 2- Intersect of Two Lines

Description:
A function that calculates the symbolic intersection of two lines. If the two lines are parallel the function returns an empty matrix and an error of NaN (not a number). If the two lines are not parallel and do not intersect, the function calculates a midpoint between the two lines and calculates the distance between the lines and the point as the error.

Test Case 1: The Two Lines Intersect

Test Graph:



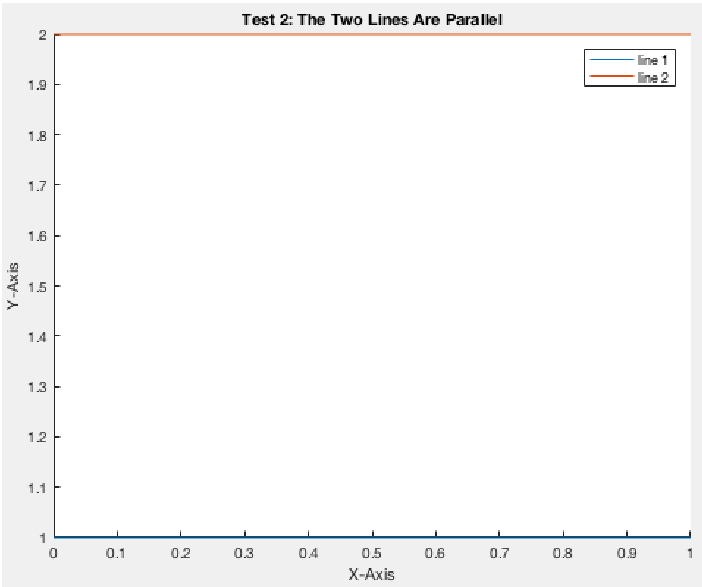
Test Output:

```
pt1 =

     1
     1
     1


error1 =

     0
```

As seen in the graph above, in test case 1, the two lines do intersect. The point pt1 given in the test output is the intersection point (see graph for visualization). The error of this test is zero because the distance between the two lines and the intersection point is zero.

Test Case 2: The Two Lines are Parallel

Test Graph:



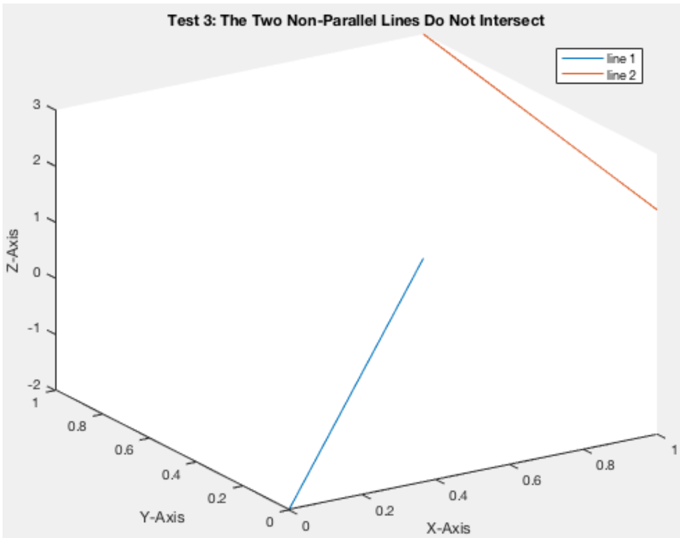
Test Output:

```
pt2 =

     []


error2 =

   NaN
```

As seen in the graph above, in test case 2, the two lines are parallel and therefore do not intersect. The function catches this in an if statement at the beginning of the function and returns an empty vector for the point of intersection, and an error of NaN because the point and therefore distance does not exist. If the function were to exist without this if statement there would be an infinite number of intersection points and would therefore be an intersection line instead.

Test Case 3: The Two Non-Parallel Lines Do Not Intersect

Test Graph:



Test Output:

```
pt3 =

     1
     0
     0


error3 =

    1.4142
```

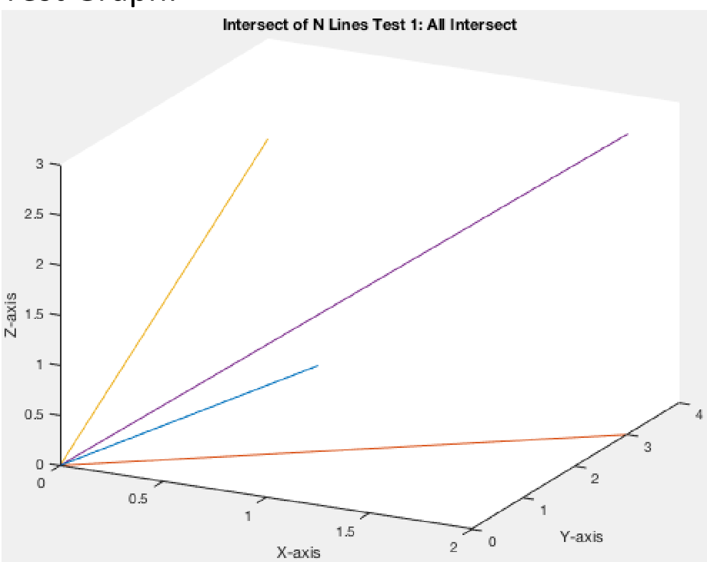
As seen in the graph above, in test case 3 the two non-parallel lines do not intersect. The intersection point calculated is the midpoint between the two lines, and the error is the distance from the lines to the point.

Part 3- Intersect of N Lines

Description:
A function that calculates the symbolic intersection of N lines, and the error metric for the intersection. If all of the N lines are parallel the function returns an empty matrix and a distance and error of NaN (not a number). If the N lines are not parallel and do not intersect, the function calculates a midpoint between the two lines, the average distance from the midpoint to all lines, and the standard deviation error for the distances.

Test 1: All N Lines Intersect at One Point

Test Graph:



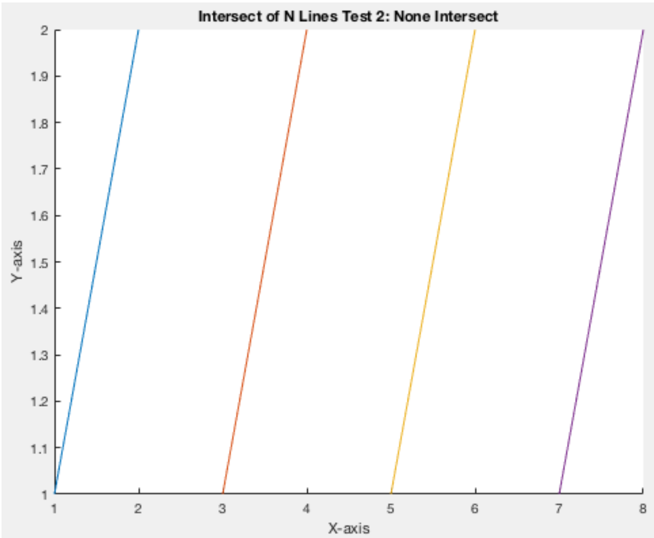
Test Output:

```
pt1 =

     0
     0
     0


dist1 =

     0


stdev1 =

     0
```

As seen in the graph and test output above, in test case 1 the four lines used for this ground truth all intersect at the point (0,0,0). The distance from the lines to the point is zero because all the points intersect here so there is no distance. The standard deviation is also zero for the same reason.

Test 2: All N Lines Are Parallel

Test Graph:



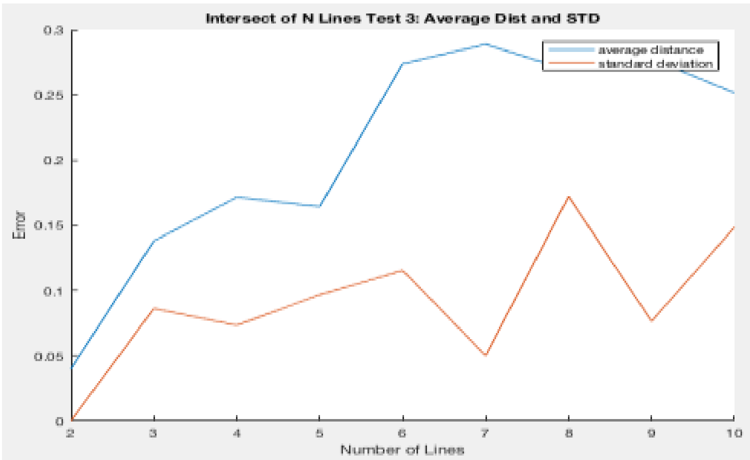
Test Output:

```
pt2 =

     []


dist2 =

   NaN


stdev2 =

   NaN
```

As seen in the graph and test output above, in test case 2 the four lines used for this ground truth are parallel to one another. Since all the lines are parallel, there is an infinite number of intersects between the lines and therefore is not possible to derive a single point (the intersect is a line). The function returns NaN for the distance and standard deviation because of this.

Test 3: Random N Lines Generated

Test Graph:



For this test, the average distance and standard deviation have been plotted for N lines from 2 to 10 lines. As N increases, the average distance and standard deviation also increases (see graph above). This may be due to the increase in randomly scattered lines, which would affect the intersection point and increase the average distance from the intersection point to all lines, therefore increasing the std as well.

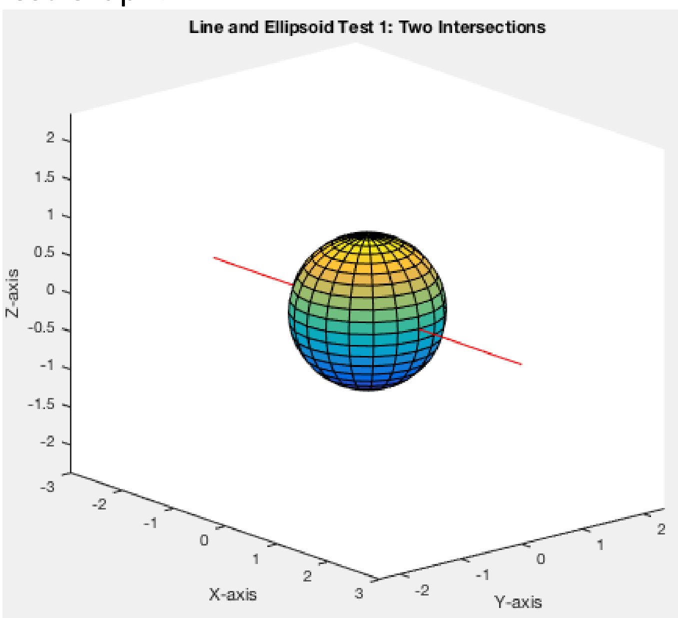
Part 4- Intersect of a Line and Ellipsoid

Description:

A function that calculates the intersection of a line and an ellipsoid using the standard equation of an ellipsoid and the calculations derived on paper.

Test 1: Line and Ellipsoid Intersect at Two Points

Test Graph:



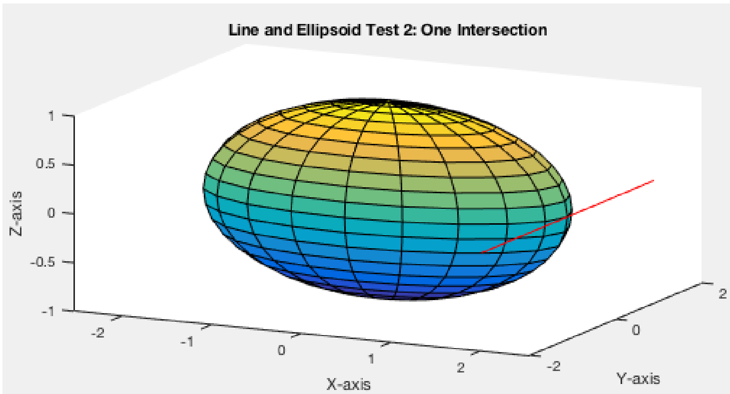
Test Output:

```
numInt1 =

     2


i11 =

     1
     0
     0


i12 =

    -1
     0
     0
```

The line and ellipsoid intersect at two different points. The graph visually confirms this as we can see the line passing through one side of the ellipsoid and exiting the other side. The test output confirms this as we are given the number 2 for number of intersections and two different intersecting points which correspond to the points on the graph.

Test 2: Line and Ellipsoid Intersect at One Point

Test Graph:



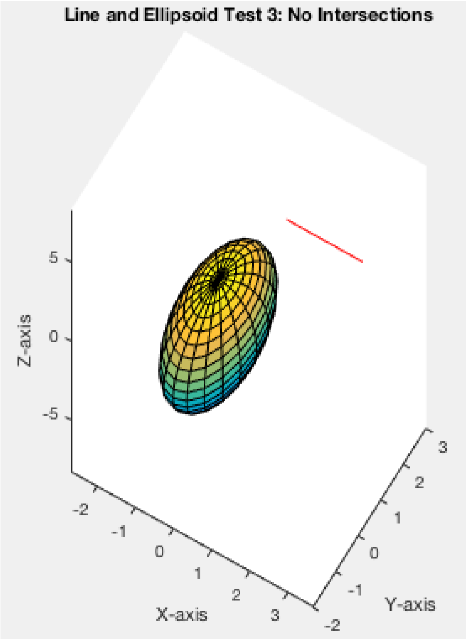
Test Output:

```
numInt2 =

     1


i21 =

     2
     0
     0


i22 =

     []
```

The line and ellipsoid intersect at one point. The generated graph visually confirms this as we can see the line is tangent to the ellipsoid and therefore intersects at one point only. The test output also confirms this as the number of intersections equals 1 and we are given one point for an intersection and an empty matrix for another intersection (intersection does not exist).

Test 3: Line and Ellipsoid Do Not Intersect

Test Graph:



Test Output:

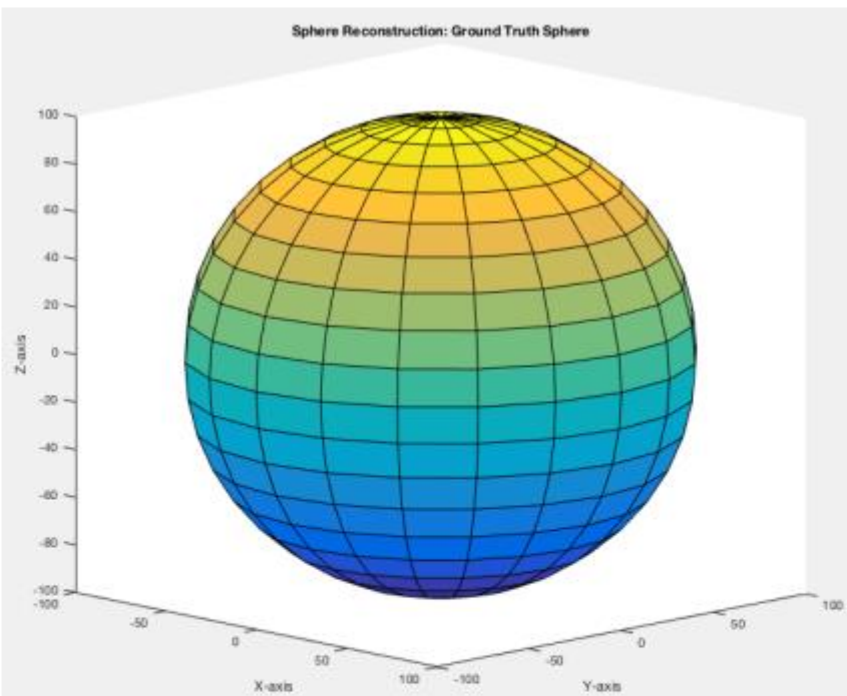
```
numInt3 =

     0


i31 =

     []


i32 =

     []
```

The line and ellipsoid do not intersect. The generated graph visually confirms this as we can see the line and ellipsoid are separate from each other, never intersecting. The test output confirms this as the number of intersections is equal to zero and both intersection points are empty matrices (the intersection does not exist).

Part 5- Sphere Reconstruction

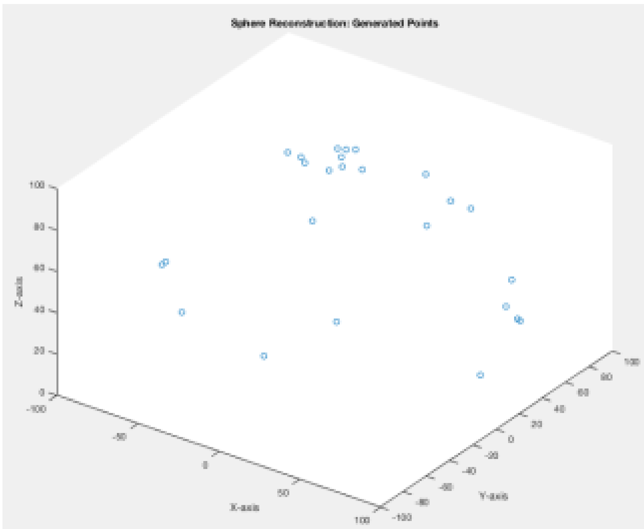
Description:
A function that accepts a sphere (as center point and radius), a value for the number of points to be generated (N), and a maxOff value used to translate the points by magnitude maxOff in any direction. This function generates N random points on the given sphere and reconstructs a new sphere using those points, returning the center point and radius of the new sphere, and the error metric of average distance from the surface of the sphere to the points and the standard deviation of those distances.

Ground Truth Sphere Graph

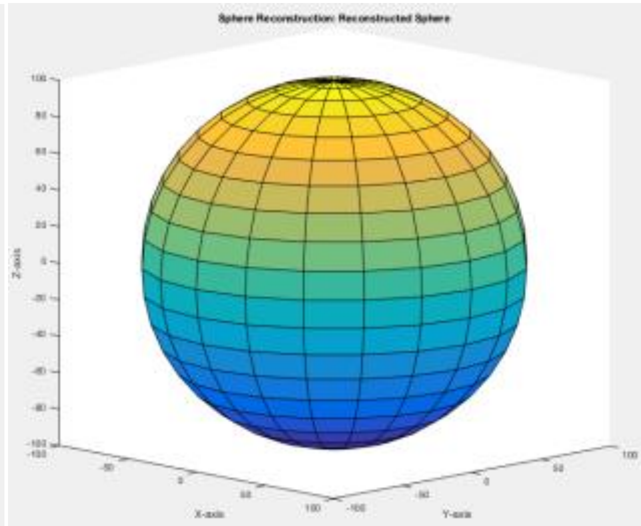


Test 1: MaxOff = 0 Sphere Reconstruction

Generated Points Graph:



Reconstructed Sphere Graph:



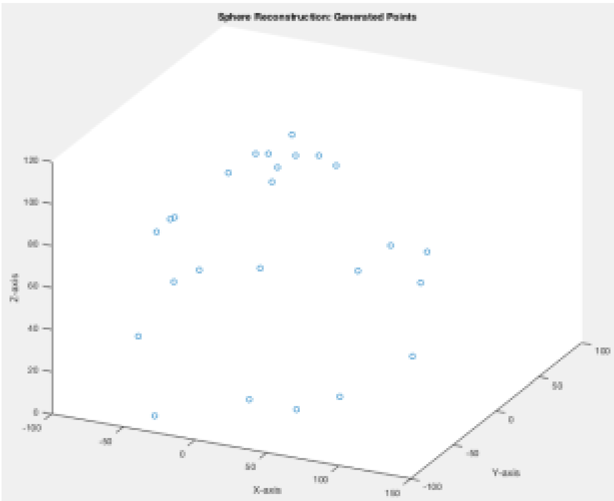
Test Output:

```
center =

     0  | radius =   | dist =   | stdev =
     0  |            |          |
     0  |   100.0000 |        0 |       0
```

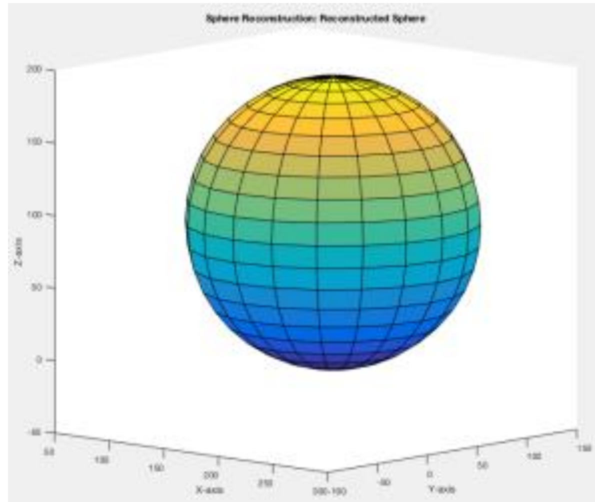
Observing the graphs and test output provided above, when the maximum distance is equal to zero (maxOff = 0), the function reconstructs an identical sphere to the ground truth sphere. The randomly generated points in the first graph for this test are shown to be within the bounds of the sphere and are all on the surface of the sphere. This is proven when examining the average distance (dist) in the output because it is equal to zero, meaning that the average distance from all the points to the surface of the reconstructed sphere is zero. This is further confirmed when examining the standard deviation as it is also zero. By observation of the sphere graphs (reconstructed and ground truth) and the center and radius outputs from this function the two spheres are the same because the center point and radius for both spheres are equal to one another.

Test 2: MaxOff = 10 Sphere Reconstruction

Generated Points Graph:



Reconstructed Sphere Graph:



Test Output:

```
center =

    1.9785
    0.1647
    0.9707
```

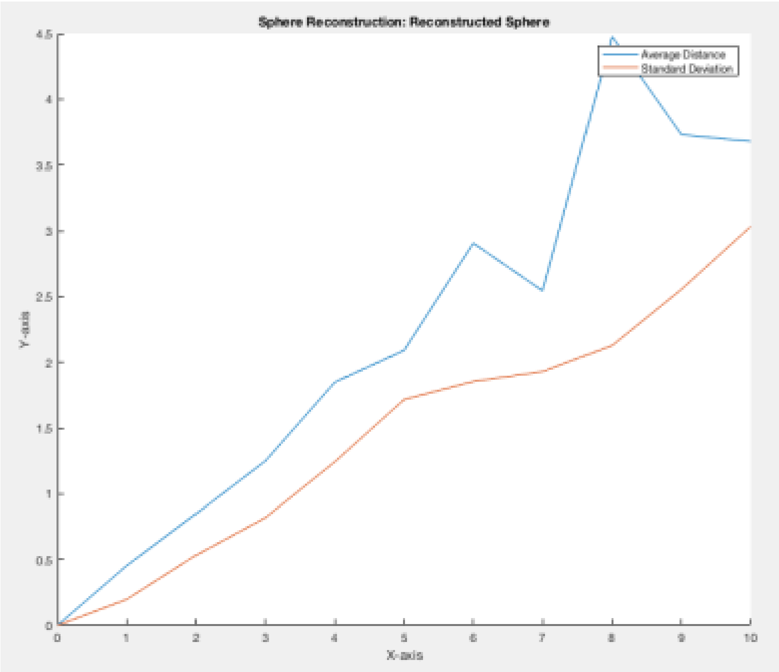
```
radius =

  100.4203
```

```
dist =

    3.6786
```

```
stdev =

    3.0328
```

Observing the graphs and test output provided above, when the maximum distance is set equal to 10 (maxOff = 10), the function is unable to reconstruct an identical sphere to the ground truth sphere, resulting in an error metric greater than zero. Having a maxOff greater than zero allows the points to be further away from the surface of the sphere (both in the positive and negative directions). This increase in distance and variety for the points results in a reconstructed sphere that is centered at a different point and has a different radius than the original sphere. This can be proven by observing the test output and comparing the reconstructed sphere graph to the original ground truth sphere graph. The average distance from all points to the reconstructed sphere surface is not zero, meaning that not all the points are on the line and the sphere radius is an average of those points. The standard deviation is also not zero meaning there is some error in this reconstruction.

Standard Deviation and Distance Graph for MaxOff = 0, 1, …, 10:



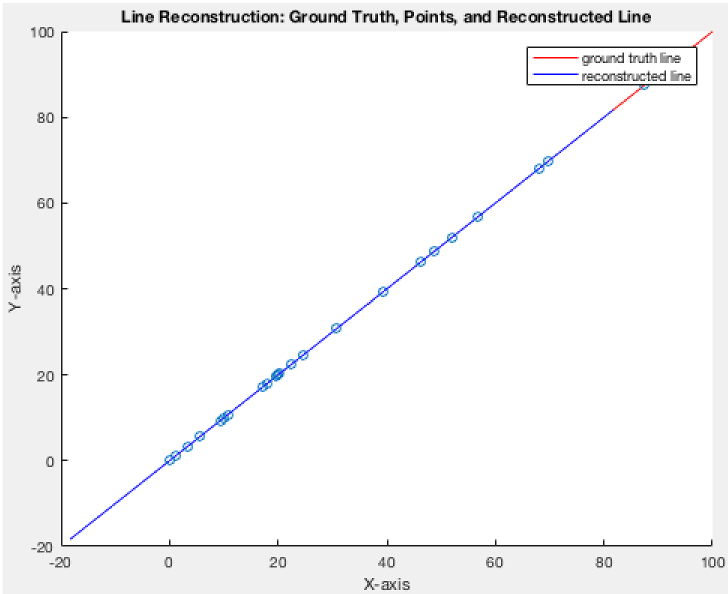
As the maxOff increases, the average distance and standard deviation increases as well. When maxOff is equal to zero, the average distance and standard deviation is also equal to zero. This is because when maxOff is zero, there is no translation from the generated points on the sphere to be moved off the sphere. Therefore when the sphere is reconstructed all of the points are on the sphere and the distance and std is zero. As the maxOff increases, the points generated on the sphere are translated futher away from the surface of the sphere in any random direction, thus making the average distance and standard deviation increase as well.

Part 6- Line Reconstruction

Description:
A function that accepts a line as two points, a value for the number of points to be generated (N), and a maxOff value used to translate the points by magnitude maxOff in any direction. This function generates N random points on the given line and reconstructs a new line using those points, returning a point on the line and the direction vector of the new line, as well as the error metric of average distance from the line to the points and the standard deviation of those distances.

Test 1: MaxOff = 0 Line Reconstruction

Test Graph:



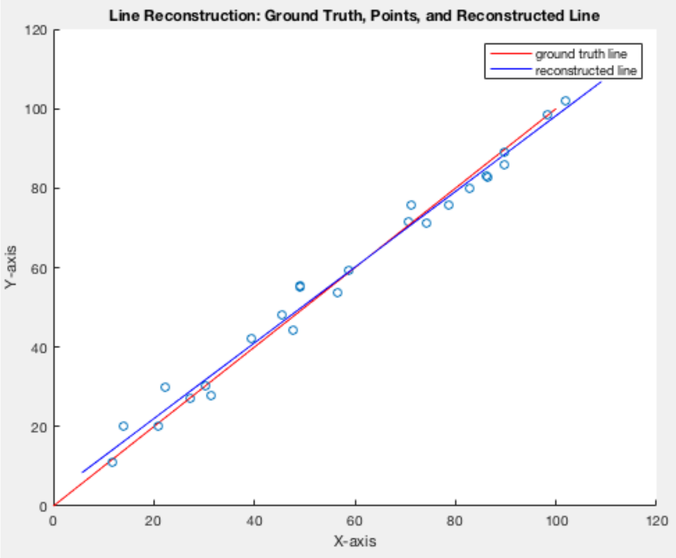
Test Output:

```
lnPt =

   31.6886
   31.6886
   31.6886

lnDir =

   -0.5774
   -0.5774
   -0.5774

dist =

     0

stdev =

     0
```

When maxOff is equal to zero, the line reconstruction function produces an identical line to the ground truth line (see graph and output data). Looking at the direction vector and line point returned, the point of the reconstructed line is on the ground truth line because x, y and z are all equal and the ground truth line goes from point (0,0,0) to (100,100,100). The direction vector is negative but is still the same direction vector as the ground truth if the absolute value were taken. The error metrics average distance and standard deviation are also equal to zero which means all the randomly generated points are on the regenerated line and therefore are also on the ground truth line because the points were distributed off the line by a random vector of magnitude zero (meaning they were never moved off the ground truth line).

Test 2: MaxOff = 10 Line Reconstruction

Test Graph:

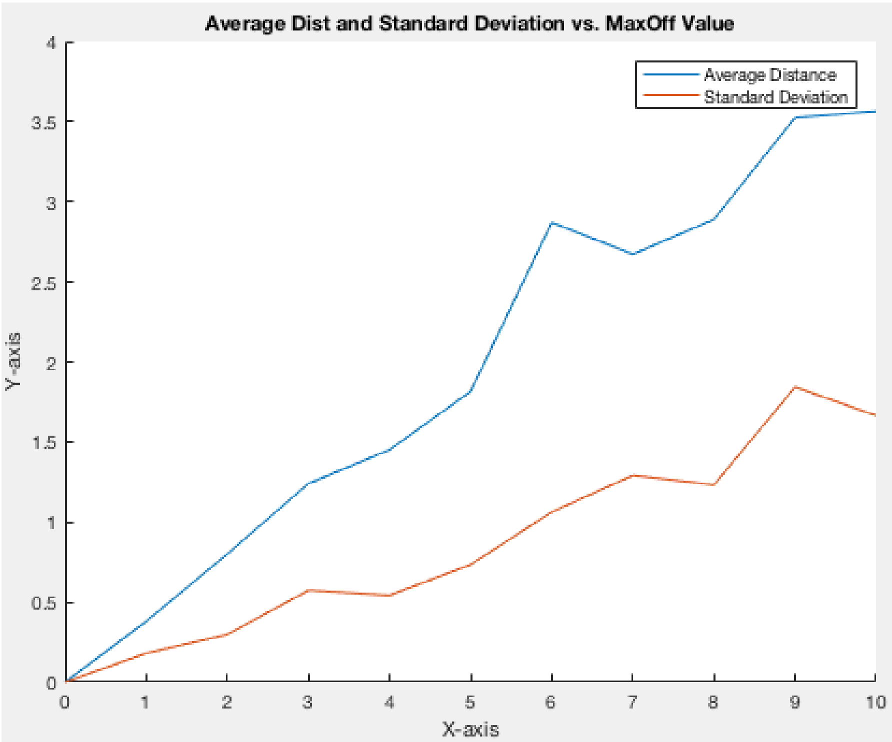


Test Output:

```
lnPt =

   57.3670
   57.6111
   58.0466


lnDir =

    0.5967
    0.5685
    0.5664

dist =

    3.5661


stdev =

    1.6651
```

When maxOff is equal to 10, the line reconstruction function is unable to construct an identical line to the ground truth line. This is because the randomly generated points are being translated in a random direction of magnitude maxOff, meaning the points are no longer on the line and the new reconstructed line is the average between those randomly offset distributed points. This can be seen in the graph as the majority of the points are not on either line but are scattered around the ground truth line instead. Looking at the test output it can also be noted that the line point calculated for the reconstructed line is not on the ground truth line (x, y and z coordinates are not equal), and the unit direction vector calculated is not equal to the direction vector of the ground truth line (x, y, and z coordinates are not equal). This is further proved by observing the error metrics for average distance and standard deviation as they are not equal to zero and therefore there is an error in this line reconstruction.

Distance and Standard Deviation vs. MaxOff Value Graph



As the maxOff value increases, the average distance and standard deviation also increases. This makes sense because the maxOff value is the magnitude of the randomly generated translation vector used to move each point in a different random direction. The greater the magnitude for points to be randomly moved, the greater the average distance from the points to the reconstructed line is and therefore the greater the standard deviation.

Part 7- Orthonormal Coordinate System

Description:
This function takes three points from a triangle and calculates the orthonormal coordinate system for that triangle (3 base vectors and the center point of the coordinate system).

Test 1: points (0,0,0), (1,0,0), (0,1,0)

Test Output:

```
x1 =

     1
     0
     0


y1 =

     0
     1
     0


z1 =           ctr1 =

     0             0.3333
     0             0.3333
     1                  0
```

The x, y and z coordinates calculated on paper (ground truth) are equal to the x, y and z coordinates calculated in the orthonormal coordinate system function. The center points calculated are also equal. Therefore, this function produces this ground truth answer.

Test 2: points (-2,-2,0), (2,-2,0), (-2,4,0)

```
x2 =

     1
     0
     0


y2 =

     0
     1
     0


z2 =

     0
     0
     1
```

```
ctr2 =

   -0.6667
         0
         0
```

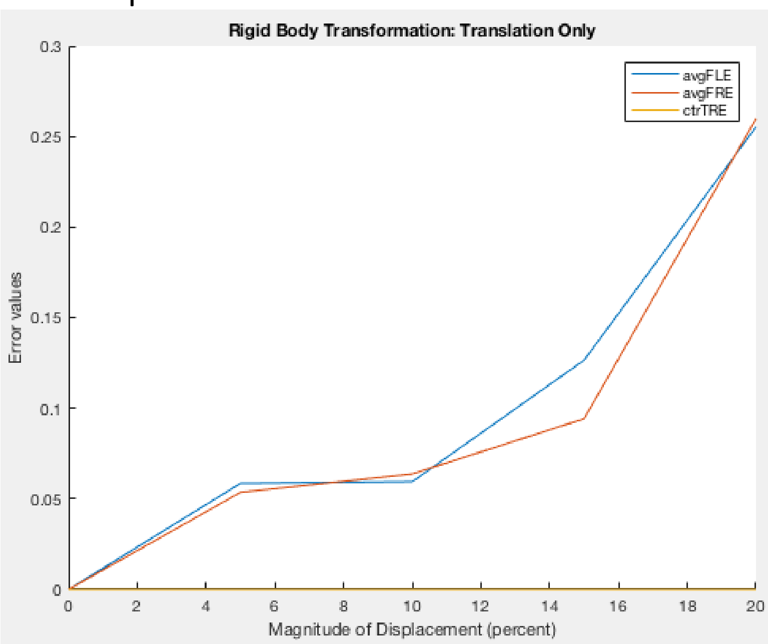
The x, y and z coordinates calculated on paper (ground truth) are equal to the x, y and z coordinates calculated in the orthonormal coordinate system function. The center points calculated are also equal. Therefore, this function produces this ground truth answer.

Part 8- Rigid Body Transformation

Description:
This function takes two triangles (indicated by 3 fiducial markers each) as input and calculates the transformation matrix to translate triangle 1 to triangle 2. It also calculates the average fiducial localization error (avgFLE, difference between the line lengths of the transformed triangle and the inputted triangle 2), the average fiducial registration error (avgFRE, the distance between the fiducial markers of the transformed triangle and triangle 2), and the target registration error in the center point (ctrTRE, the difference between the center point for the transformed triangle and triangle 2).

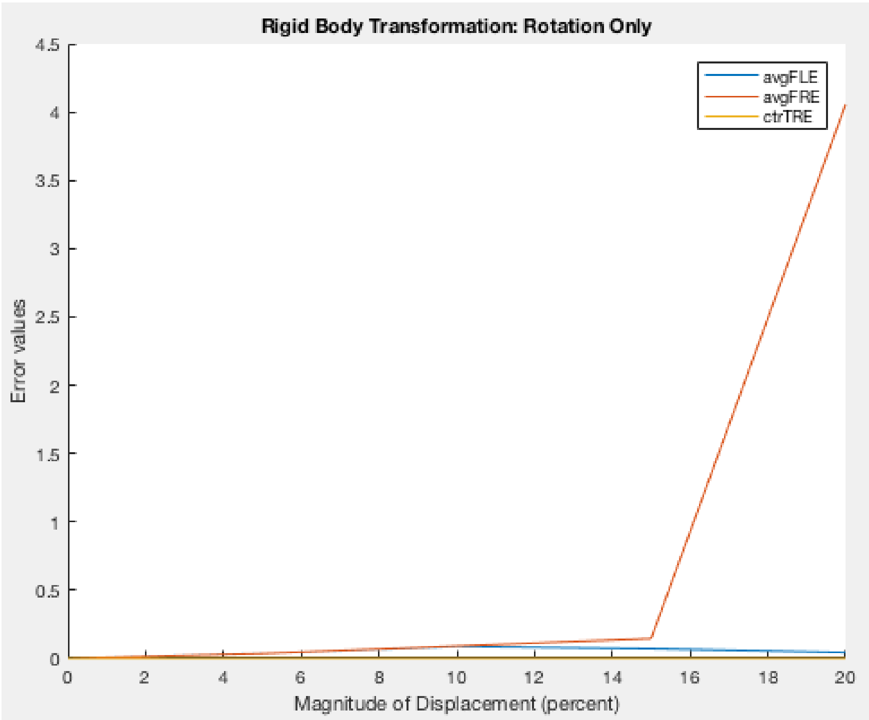
Test 1: Translation Only

Test Graph:



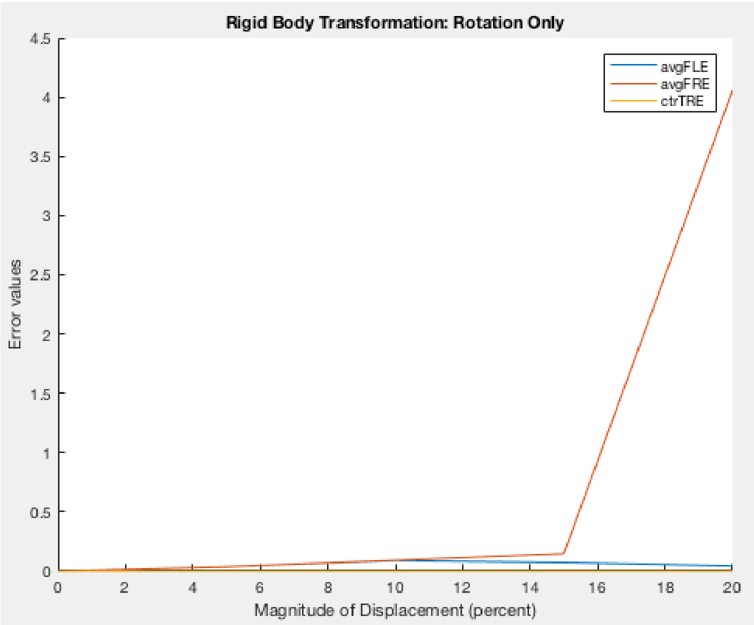
As the magnitude of the displacement increases, the avgFLE and avgFRE increase as well. This is seen in the graph and makes sense because as the fle increases the fiducial markers are displaced more and therefore the errors returned increase too. The ctrTRE does not increase because it is the same point for both triangle 2 and the transformed triangle (center point is the most accurate point even with error).

Test 2: Rotation Only



As the magnitude of the displacement increases, the avgFLE and avgFRE increase as well. This is seen in the graph and makes sense because as the fle increases the fiducial markers are displaced more and therefore the errors returned increase too. The ctrTRE does not increase because it is the same point for both triangle 2 and the transformed triangle (center point is the most accurate point even with error).

Test 3: Rotation and Translation



As the magnitude of the displacement increases, the avgFLE and avgFRE increase as well. This is seen in the graph and makes sense because as the fle increases the fiducial markers are displaced more and therefore the errors returned increase too. The avgFRE increases exponentially for the rotation because the displaced fiducial points are moved more when rotated and are more random so when actual fle is introduced with rotation it spikes. The ctrTRE does not increase because it is the same point for both triangle 2 and the transformed triangle (center point is the most accurate point even with error).