Torrence Gue
Bilal Sharqi
Report Starting: 06-08-2020
Last Report: TBD

Weekly Reports Summer 2020

August 2, 2020: Weekly Report 9, Summary of Work from July 27th – July 31st 2020

Work completed this week:

- Properly matched the frequencies mode data from experimental data to numerical data (coord+mode)
- Implemented MAC function for beam data
- I.D.'d problematic modes and how their shapes affected MAC values
- Completed trade study of how Z translation affects change in MAC calculation

Current/Future Work:

- Currently working on verifying large batch MAC function, calculateMAC. The goal is to verify that it works so that I do not have to iterate through the singleMAC equation instead
- Need to do research into XOR and orthogonality testing, specifically finding the correct equation using the mass matrix.
- Need to simulate the torsional effects of the wing using given chord length and twist angle from numerical simulation.

Trade Study Analyzing How Z Translation of Experimental Modes Affects MAC Value:

Introduction: To compare numerical mode data (coord+mode) to experimental mode data(coord+mode), it is sometimes necessary to rotate a mode shape 180 degrees about an axis. In this case, rotating the experimental axis 180 degrees about the x-axis helps Modes 2, 3, 5, and 6 to create a mirror image that resembles a numerical mode (coord+mode). This creates problems however, when the mirrored experimental mode is not in the exact same position as the original experimental mode. MAC analysis of this beam in this case is predominately affected by changes in the z-axis. In order to counter this issue, I have analyzed how an additional Z translation to the experimental mode(s) can help reduce the impact of this rotation. The following is an analysis of how adding additional Z translations to experimental data have, if any, an effect on modal MAC calculations. If an additional Z translation is required, should it be made to a selected number of rotated modes, or all experimental modes? I have listed my conclusions below.

Conclusions:

Discovery 1:

Proximity of Z values matters. I made a test case where I essentially separated the experimental (mode + coord) data from the numerical (mode+coord) data and ran a full MAC. Results are below. Simply put, proximity of z value matters in this case because of MAC values came back very low for those values that were translated in the script. Mode 1 and Mode 5 were not altered in the script and appear normal because they did not require a rotation across the z axis. See **Table 1.0** below for the associated MAC with Z translation of -5.0 meters

	0	1	2	3	4	5
0	0.930631	0.295311	0.267309	0.101689	0.30925	0.297997
1	0.0876806	0.329082	0.426486	0.18406	0.425599	0.399824
2	0.00297491	0.29801	0.402398	0.0977148	0.267986	0.305352
3	0.0760216	0.374109	0.415393	0.85321	0.413286	0.394041
4	0.233398	0.298866	0.266254	0.0795008	0.376538	0.294405
5	0.0716786	0.375048	0.41671	0.251521	0.409119	0.315087

Table 1.0: MAC chart with additional z translation increased by -5.0 meters (moved further apart)

Discovery 2:

I simulated two cases, one where I analyze how an additional Z translation for Modes 2,3,5, and 6 of experimental position on similar mode average MAC value. These experimental modes were selected because they required a flip across the x-axis to correctly mirror the numerical data. This 'flip' however caused Z-position data (coord+mode) to decrease across the mode shape. The goal of this simulation was to decrease the average 'gap' between the experimental and numerical modes to close to zero. The best translation of these modes would be a Z translation of 0.88 meters. This created an average MAC between similar modes of 0.61. The MAC of the Z translation of 0.88 meters is listed below in **Table 2.a.** The additional **Figure 2.a** describes how Z translation effects average difference between numerical and experimental data as well as average MAC value between like numerical and experimental modes.

	0	1	2	3	4	5
0	0.930631	0.143818	0.336468	0.101689	0.13175	0.169841
1	0.0876806	0.931336	0.268515	0.18406	0.332359	0.422746
2	0.00297491	0.160135	0.0215141	0.0977148	0.449656	0.16699
3	0.0760216	0.424325	0.235663	0.85321	0.300266	0.358693
4	0.233398	0.125399	0.337636	0.0795008	0.00174706	0.180974
5	0.0716786	0.432417	0.240518	0.251521	0.339774	0.950282

Table 2.a MAC chart with additional Z translation increased by 0.88 meters, optimal configuration Mode 2, 3, 5, 6 Experimental Z position data (Coord+mode) on average are closest to the Num Z position data (Coord+Mode)

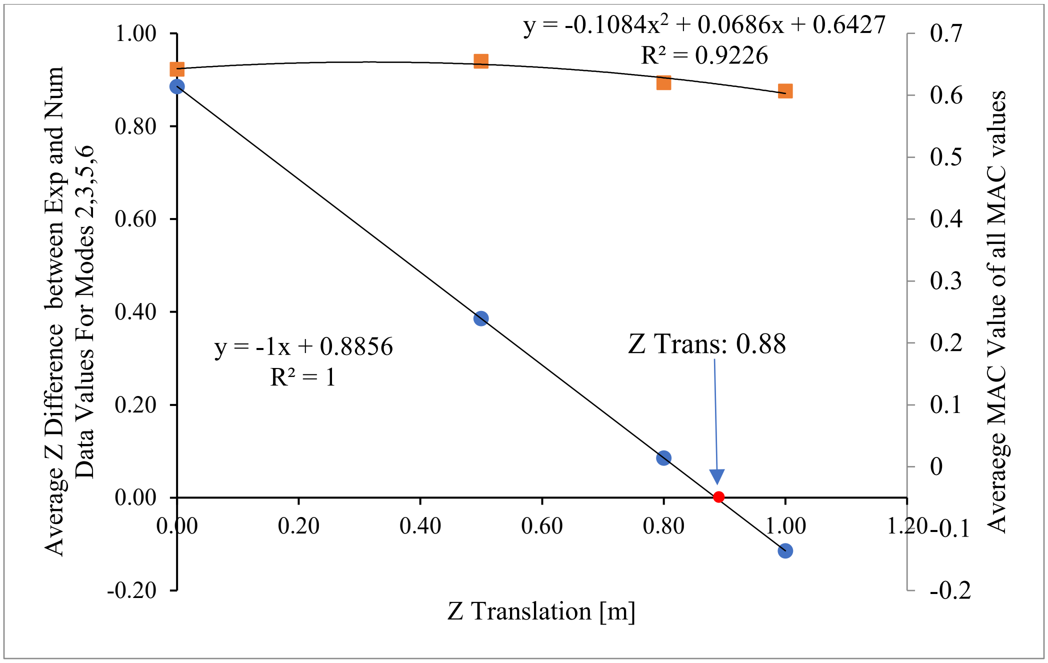


Figure 2.a Impact of Z translation on the average difference between Exp and Num (coord+mode) Z coordinate data

*Mode 2,3,5,6

Impact of Z translation on the Average MAC of complete mode data set

Z Translation of 0.8856 is best to create smallest average ‘gap’ between num and exp data (coord +mode) Modes 2,3,5,6

Conclusion:

Z Translation of 0.885 is best to create smallest average ‘gap’ between num and exp data (coord +mode) for Modes 2.3.5.6. Resulting average mac of each mode: MAC: 0.61

The second case analyzed how an additional Z translation for **all modes** of experimental position on similar mode average MAC value. The goal was to decrease the average ‘gap’ between the experimental and numerical modes to close to zero **for all modes**. The best translation of these modes would be a Z translation of 0.67 meters. The MAC of the Z translation of 0.67 meters is listed below in **Table 2.b.** The additional **Figure 2.b** describes how Z translation effects average difference between numerical and experimental data as well as average MAC value between like numerical and experimental modes.

This created an average MAC between similar modes of 0.59. This simulation seemed to be the ideal case because it creates a change on **all** MAC values, rather than a few selected modes.

	0	1	2	3	4	5
0	0.725507	0.0812263	0.271064	0.236864	0.0671117	0.115782
1	0.316998	0.875612	0.168473	0.376641	0.234554	0.350042
2	0.134614	0.0947985	0.122731	0.236154	0.408849	0.112021
3	0.296594	0.328603	0.14076	0.801624	0.205337	0.285306
4	0.351656	0.0654937	0.272576	0.215974	0.0593696	0.126914
5	0.293008	0.336388	0.144873	0.422659	0.246005	0.959678

Table 2.b: MAC chart with additional Z translation increased by 0.67 meters, optimal configuration so that all Experimental Z position data (Coord+mode) on average are closest to the Num Z position data (Coord+Mode)

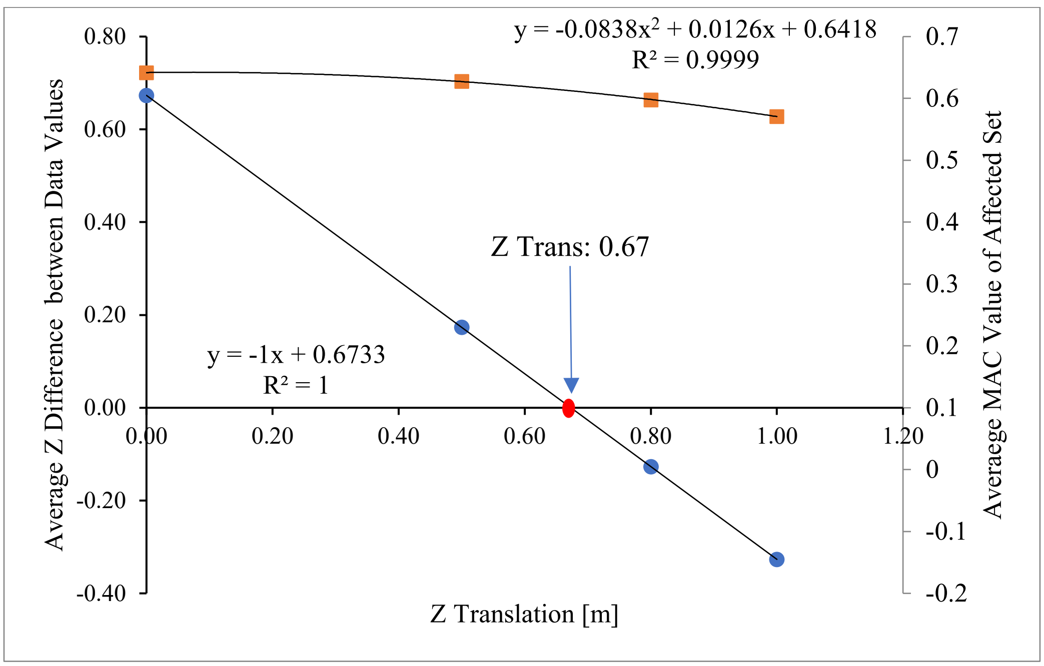


Figure 2.b Impact of Z translation on the average difference between Exp and Num (coord+mode) Z coordinate data

*All Modes (1,2,3,4,5,6)

Impact of Z translation on the Average MAC of complete mode data set

Conclusion:

Z Translation of 0.67 is best to create smallest average 'gap' between num and exp data (coord +mode) for All Modes. Resulting average mac of each mode: MAC: 0.590

<u>The effect increasing z translation has on MAC values of similar analytical and numerical modes</u>

*All Z Translations occur on the Z axis and are applied to the Experimental Data ONLY

MAC Plots for Z Translations for Modes 2,3,5, and 6

	0	1	2	3	4	5
0	0.930631	0.0453449	0.000333034	0.101689	0.131783	0.0221556
1	0.0876806	0.128056	0.0621874	0.18406	0.0649222	0.00167112
2	0.00297491	0.0376747	0.861306	0.0977148	0.00157838	0.0259847
3	0.0760216	6.38752e-05	0.0747962	0.85321	0.0750477	0.000284904
4	0.233398	0.058798	0.000254212	0.0795008	0.814877	0.0165037
5	0.0716786	6.08853e-06	0.0726215	0.251521	0.0527896	0.277758

MAC chart with no additional z translations

	0	1	2	3	4	5
0	0.930631	0.0311242	0.184339	0.101689	0.0156081	0.0636517
1	0.0876806	0.743104	0.0752369	0.18406	0.12222	0.258974
2	0.00297491	0.0400834	0.297868	0.0977148	0.318701	0.0598765
3	0.0760216	0.218366	0.0567765	0.85321	0.100724	0.199659
4	0.233398	0.0209379	0.185922	0.0795008	0.209869	0.0732005
5	0.0716786	0.224963	0.059493	0.251521	0.134373	0.902964

MAC chart when additional z translation is increased by 0.5 meters

	0	1	2	3	4	5
0	0.930631	0.120483	0.316113	0.101689	0.108159	0.150476
1	0.0876806	0.920901	0.233047	0.18406	0.300003	0.398967
2	0.00297491	0.136061	0.0485173	0.0977148	0.440357	0.147206
3	0.0760216	0.392593	0.201526	0.85321	0.268433	0.334012
4	0.233398	0.102666	0.31743	0.0795008	0.0126248	0.16178
5	0.0716786	0.400681	0.206196	0.251521	0.309135	0.960757

MAC chart when additional z translation is increased by 0.8 meters

	0	1	2	3	4	5
0	0.930631	0.170837	0.354838	0.101689	0.158248	0.191562
1	0.0876806	0.929448	0.307054	0.18406	0.364835	0.44635
2	0.00297491	0.187626	0.00392619	0.0977148	0.453623	0.189308
3	0.0760216	0.455725	0.273405	0.85321	0.332766	0.384198
4	0.233398	0.152261	0.355806	0.0795008	0.00128306	0.202273
5	0.0716786	0.463671	0.27836	0.251521	0.370039	0.928902

MAC chart when additional z translation is increased by 1.0 meters

MAC Plots for Z Translations for Modes 1-6 (All Modes)

	0	1	2	3	4	5
0	0.930631	0.0453449	0.000333034	0.101689	0.131783	0.0221556
1	0.0876806	0.128056	0.0621874	0.18406	0.0649222	0.00167112
2	0.00297491	0.0376747	0.861306	0.0977148	0.00157838	0.0259847
3	0.0760216	6.38752e-05	0.0747962	0.85321	0.0750477	0.000284904
4	0.233398	0.058798	0.000254212	0.0795008	0.814877	0.0165037
5	0.0716786	6.08853e-06	0.0726215	0.251521	0.0527896	0.277758

MAC chart with no additional z translations

	0	1	2	3	4	5
0	0.785021	0.0311242	0.184339	0.217055	0.0156081	0.0636517
1	0.282988	0.743104	0.0752369	0.350472	0.12222	0.258974
2	0.105558	0.0400834	0.297868	0.215497	0.318701	0.0598765
3	0.263044	0.218366	0.0567765	0.834806	0.100724	0.199659
4	0.344818	0.0209379	0.185922	0.194559	0.209869	0.0732005
5	0.258852	0.224963	0.059493	0.403263	0.134373	0.902964

MAC chart when additional z translation is increased by 0.5 meters

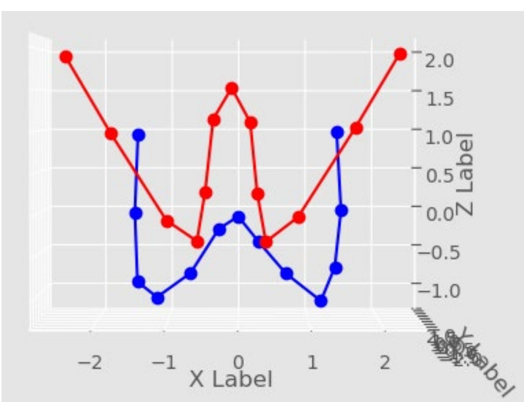
	0	1	2	3	4	5
0	0.686541	0.120483	0.316113	0.248102	0.108159	0.150476
1	0.336199	0.920901	0.233047	0.391024	0.300003	0.398967
2	0.152777	0.136061	0.0485173	0.247957	0.440357	0.147206
3	0.315689	0.392593	0.201526	0.777022	0.268433	0.334012
4	0.353542	0.102666	0.31743	0.228442	0.0126248	0.16178
5	0.312537	0.400681	0.206196	0.432381	0.309135	0.960757

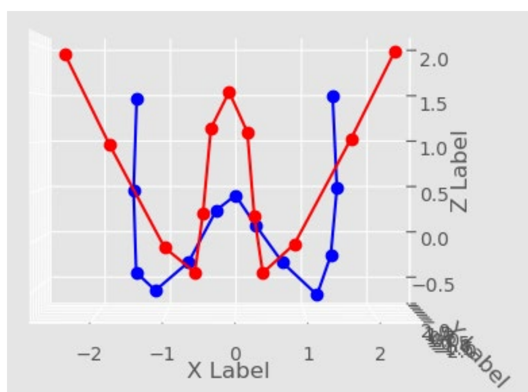
MAC chart when additional z translation is increased by 0.8 meters

	0	1	2	3	4	5
0	0.636729	0.170837	0.354838	0.260837	0.158248	0.191562
1	0.357946	0.929448	0.307054	0.406795	0.364835	0.44635
2	0.175209	0.187626	0.00392619	0.261427	0.453623	0.189308
3	0.337479	0.455725	0.273405	0.742525	0.332766	0.384198
4	0.353588	0.152261	0.355806	0.242938	0.00128306	0.202273
5	0.334914	0.463671	0.27836	0.441931	0.370039	0.928902

MAC chart when additional z translation is increased by 1.0 meters

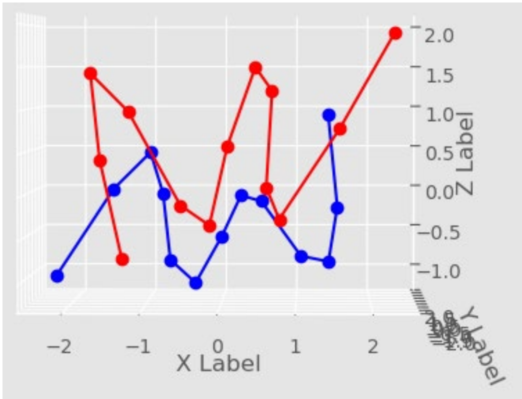
Visual Representation comparing MAC values of zero Z translation vs. Z translation of 0.5 m
For Modes 2,3,5,6



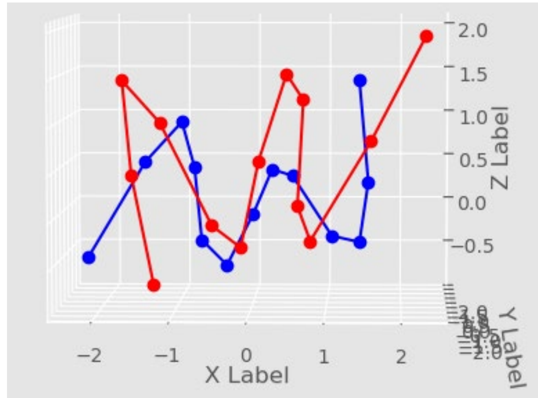


No Additional Z translation (MAC: 0.12) Additional Z Translation of 0.5 (MAC: 0.74)

Mode 2: Exp Frequency 5.91Hz (Blue) , Num Frequency 6.36 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

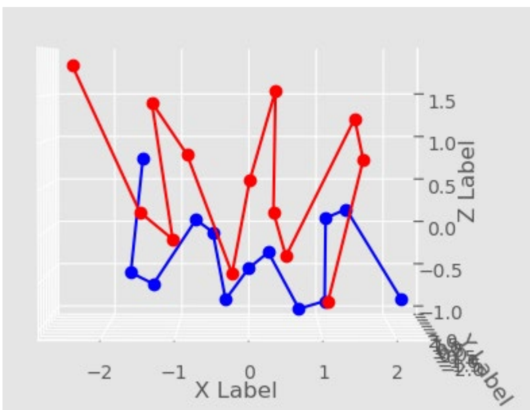


No Additional Z translation (MAC: 0.86)

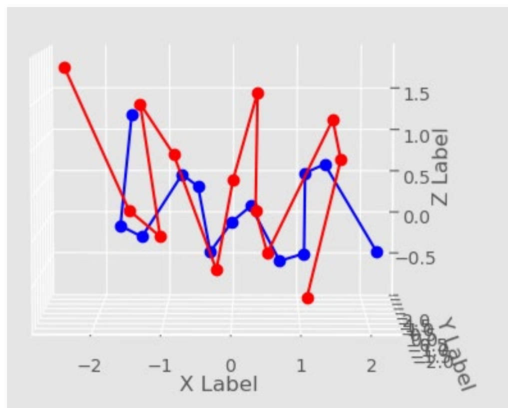


Additional Z Translation of 0.5 (MAC: 0.209)

Mode 3: Exp Frequency 9.29Hz (Blue) , Num Frequency 10.59 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

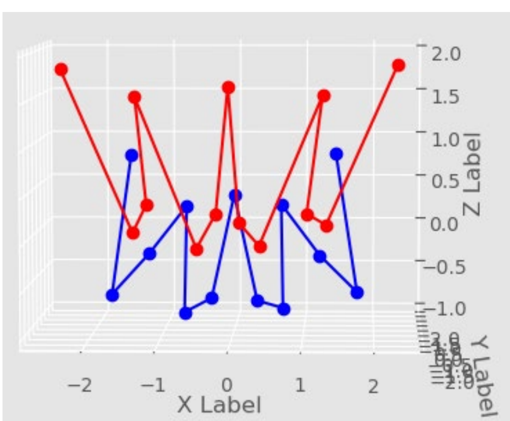


No Additional Z Translation (MAC: 0.81)

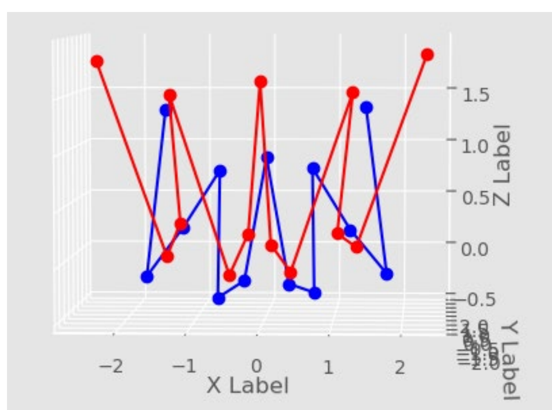


Additional Z Translation of 0.5 (MAC: 0.29)

Mode 5: Exp Frequency 18.91 Hz (Blue) , Num Frequency 22.07 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data



No Additional Z Translation (MAC: 0.27)



Additional Z Translation of 0.5 (MAC: 0.90)

Mode 6: Exp Frequency 27.65Hz (Blue) , Num Frequency 29.519 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

Matched Frequencies from Numerical and Experimental Beam Data (Coord + Mode)

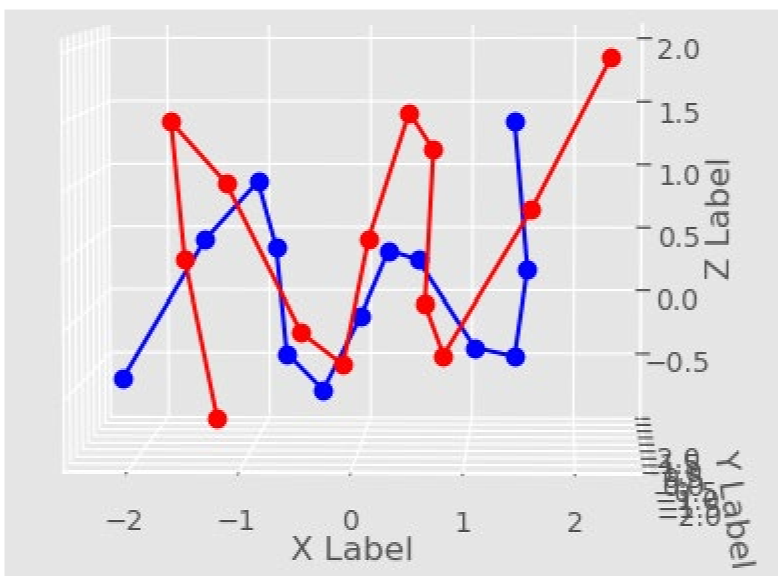
Exp Index	1	2	4	7	8	14
Num Index	7	8	9	10	12	13
Exp Freq.	3.02 Hz	5.91 Hz	9.29 Hz	15.29 Hz	18.91 Hz	27.6537 Hz
Num Freq	3.28 Hz	6.36 Hz	10.59 Hz	15.89 Hz	22.07 Hz	29.519 Hz

Mac diagram of beam data

	0	1	2	3	4	5
0	0.930631	0.0311242	0.184339	0.101689	0.0156081	0.0636517
1	0.0876806	0.743104	0.0752369	0.18406	0.12222	0.258974
2	0.00297491	0.0400834	0.297868	0.0977148	0.318701	0.0598765
3	0.0760216	0.218366	0.0567765	0.85321	0.100724	0.199659
4	0.233398	0.0209379	0.185922	0.0795008	0.209869	0.0732005
5	0.0716786	0.224963	0.059493	0.251521	0.134373	0.902964

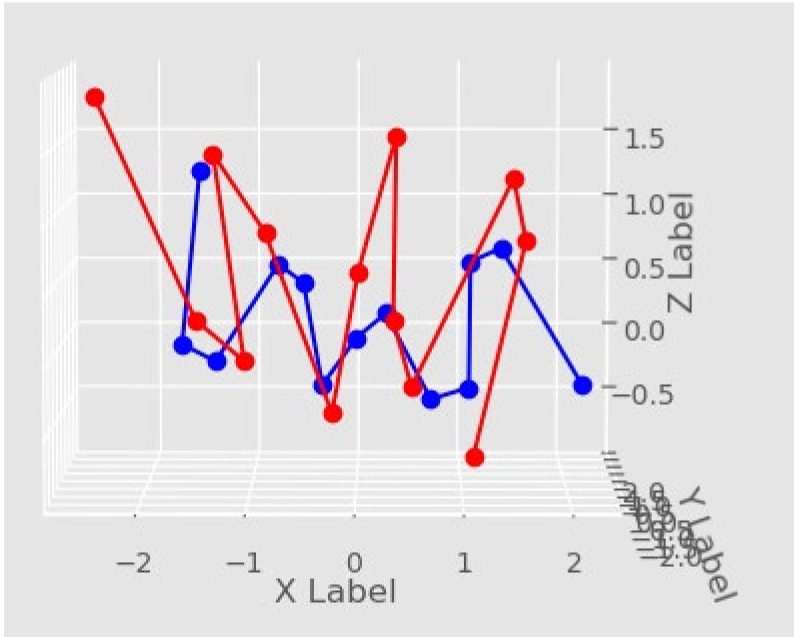
Modes 3 and 5 stand out:

Mode 3:



Mode 3: Exp Frequency 9.29Hz (Blue) , Num Frequency 10.59 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

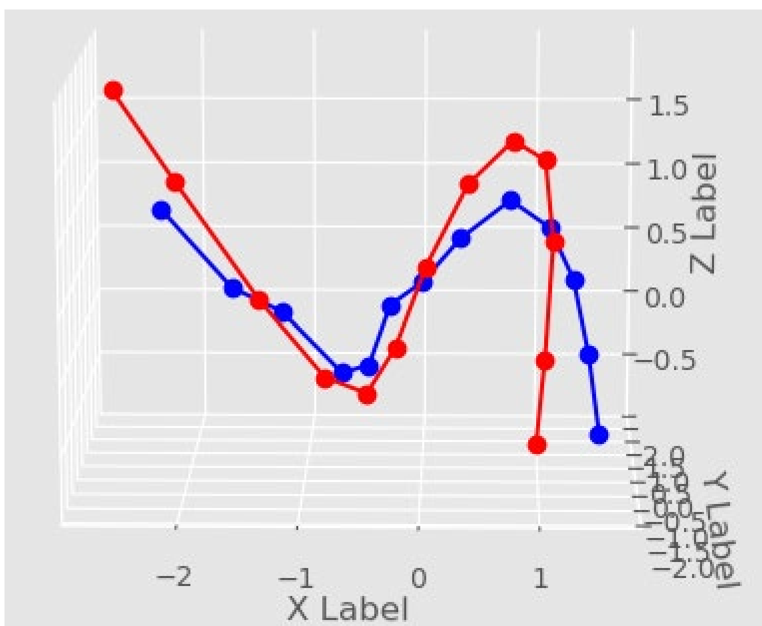
Mode 5:



Mode 5: Exp Frequency 18.91 Hz (Blue) , Num Frequency 22.07 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

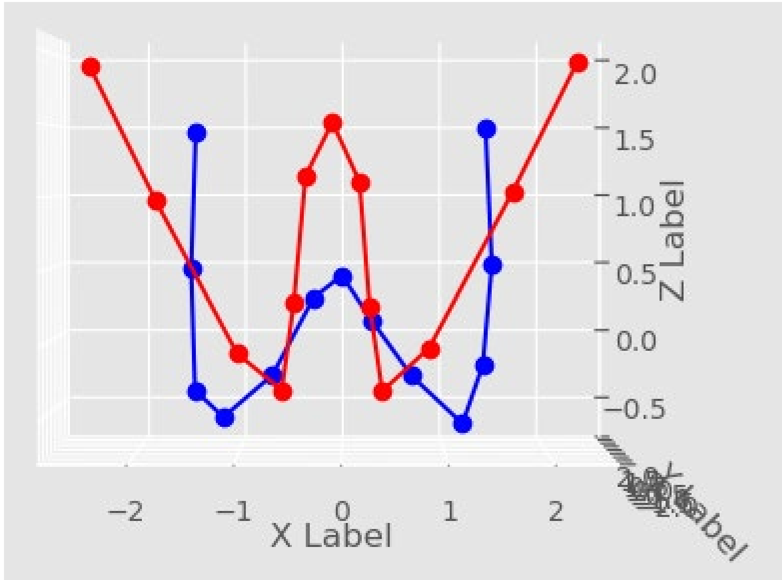
<u>Modes 1, 2, 4, and 6</u>

Mode 1:



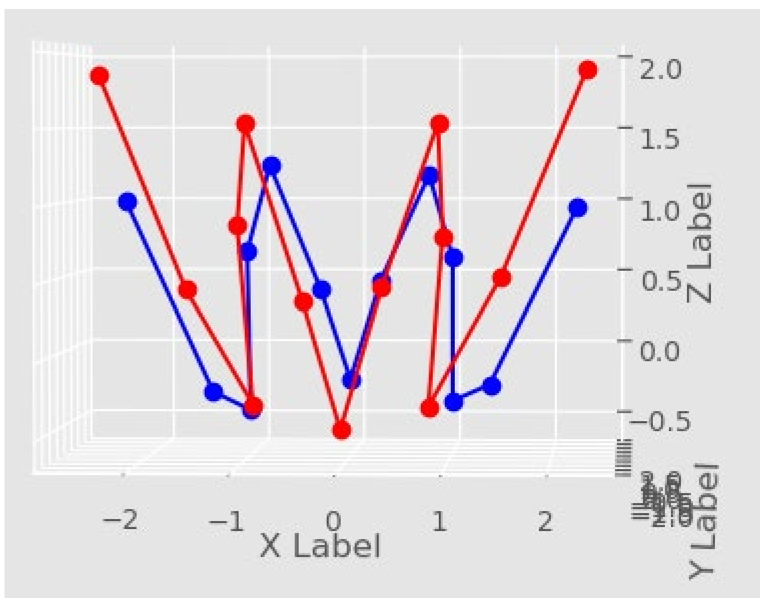
Mode 1: Exp Frequency 3.02Hz (Blue) , Num Frequency 3.28 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

Mode 2:



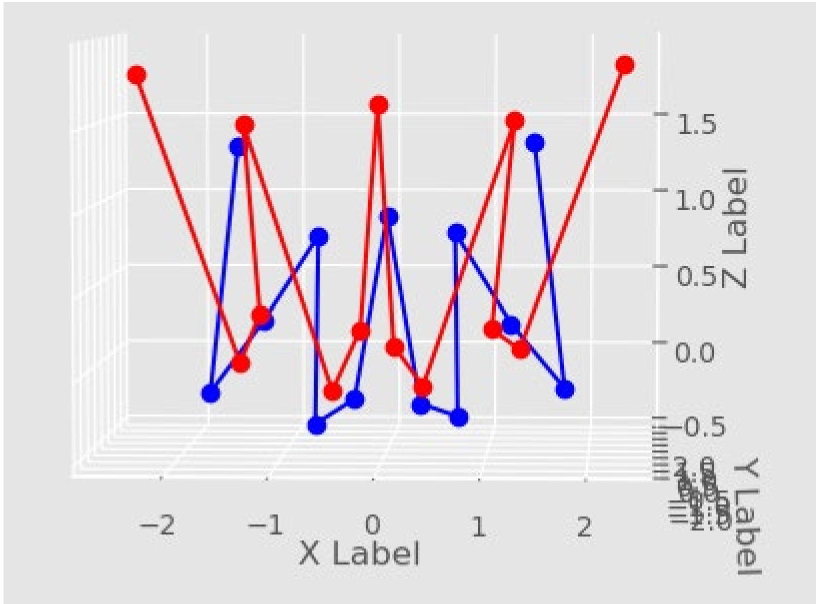
Mode 2: Exp Frequency 5.91Hz (Blue) , Num Frequency 6.36 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

Mode 4:



Mode 4: Exp Frequency 15.29Hz (Blue) , Num Frequency 15.89 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

Mode 6:



Mode 6: Exp Frequency 27.65Hz (Blue) , Num Frequency 29.519 Hz (Red), displaying Mode + Coord Data for Experimental and Numerical Data

*Special note: Should the X component of exp(mode+coord) data be accounted for here? It makes the data look odd

<u>Some additional Notes</u>

- Had to add an additional offset to experimental value of about 0.5. This was because of the flip in the horizontal greatly offput the experimental data. Made it more negative than it should have been
  - I could raise the minimum exp data (coord+mode) to make it flush with the numerical data (coord+mode). This could get us more favorable readings than using an arbitrary number of 0.5

<u>August 27, 2020: Weekly Report 8 - Summary of Work from July 20th to July 24th</u>

Summary of Work:

- A test version of the individual MAC was created to test a single beam mode comparing the experimental data and numerical data once properly processed. Numbers were deemed exceptional. It should be noted that we will not expect MAC values higher than 0.7 due to experimental error
- It should be noted that overall length of the simulated beam is not exactly the same as the experimental beam, and this causes issues when it comes to conservation of length in the experimental data (Pretty sure of this because of constraints put on numerical sim)
- An averaging function, beamAverage, has been made to average the given pairs of experimental data to a single set of coordinates instead of two sets of coordinates spanning each side of the

beam. (Bilal brought up concerns of this creating issues because the coordinates + the mode shape of each side would be averaged into one value.)

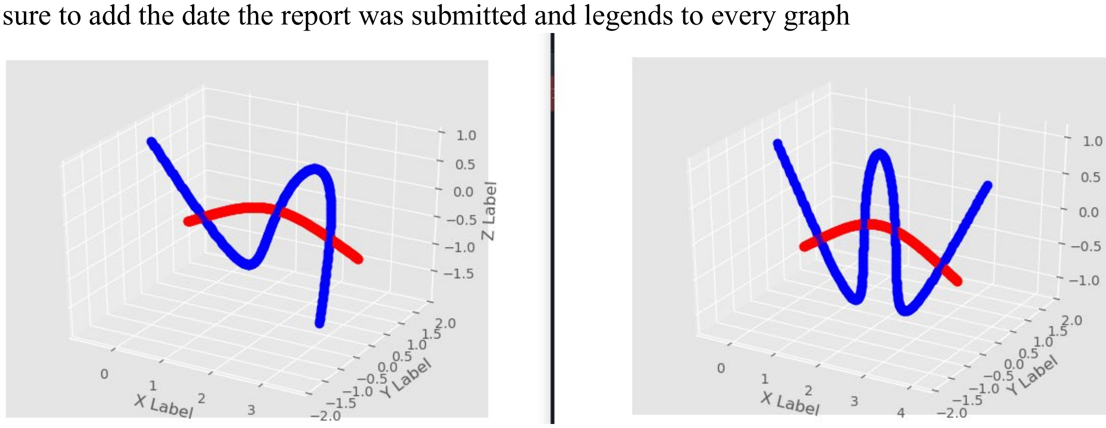
- Removed the contribution of static deformation from the numerical data set. It was formerly the sum of coordinates, static deformation, and mode shape. Bilal pointed out that the shape was over emphasized. Numerical data set is now the sum of coordinates of beam and its associated mode shape.
- Adjusted numerical data (coord+static def) to match the experimental data(coord) for mode shape data relatively centered relative to the experimental data set. I used the following translations:
    - (Added these to num coordinates) x: -1.92577, z : 0.532887
- Created new function called grad3D which takes the derivative of the first two most positive x values in the reduced numerical data (coord+mode shape). This function finds the derivative of z with respect to x of these two points and compares the sign with that of the two most positive x values in the averaged experimental data set (coord+mode shape). In theory the function will correct data sets where numerical mode data is “flipped” relative to the experimental data counterpart over the z or x axis. (Need to ask Bilal if he would prefer to edit the numerical to match the experimental data or vice versa. I was editing the numerical data to match the experimental data to keep consistent. Previously I moved the numerical data coordinate system to match the experimental data coordinate system because of the better placement. If Bilal wishes me to edit only the experimental data, I could also do that.
  (Note: Both data sets start from about x = 0 and the rest of the data has a positive x).

Work for this Week:

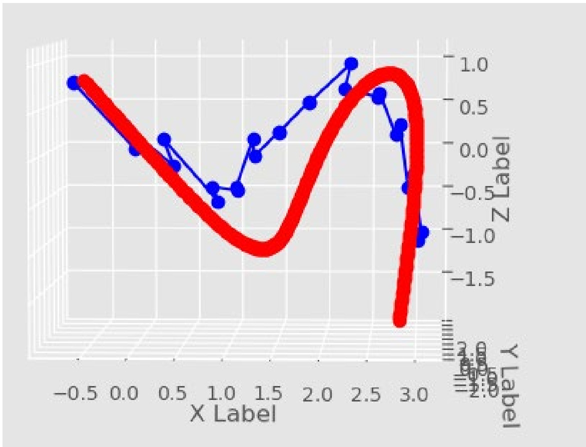
- I want to expand the grad3D function to handle rotations of data about the z axis as well. It is currently only making corrections about the x axis.
- Creating function to tell the torsional displacement using geometry and 3 rotational angles given from simulated data (coord+mode) from NASTRAN
- Form new or edit existing mac function to finish comparison of beam data
- Find the right modes to compare between modal exp beam data and num beam data to calculate full mac table

--------------------------Pictures generated for meeting on July 22nd 2020-----------------------------------

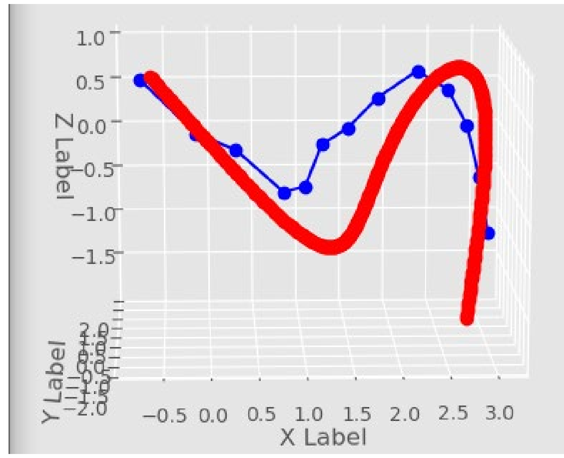
Make sure to add the date the report was submitted and legends to every graph



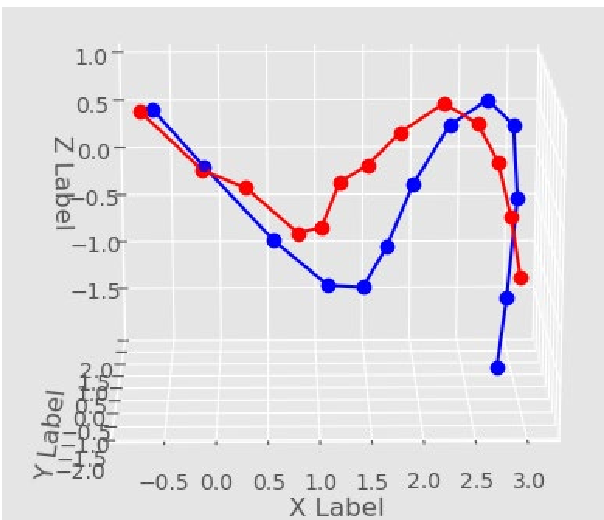
Numeric Frequencies (Blue) at 3.28 Hz (left) and 6.36 Hz Right compared to beam coordinates + static def shape(red)



Left: Full Experimental (Blue) and Numerical Data (Red) Set (Exp. 3.28 Hz, Num 3.28Hz)

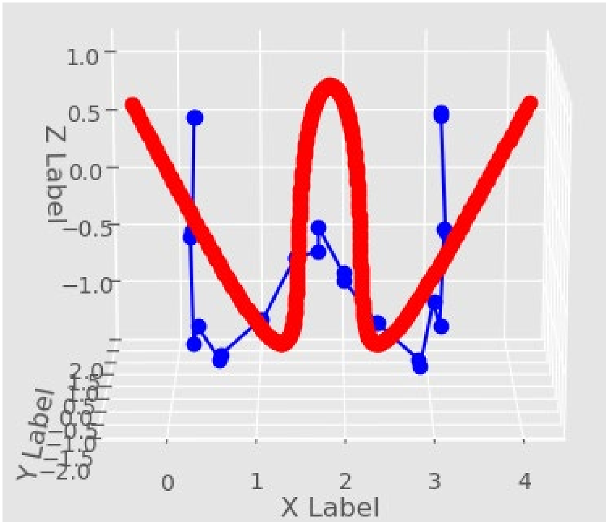


Right: Full Numerical Data Set(Red) with averaged Experimental Data Set(Blue)

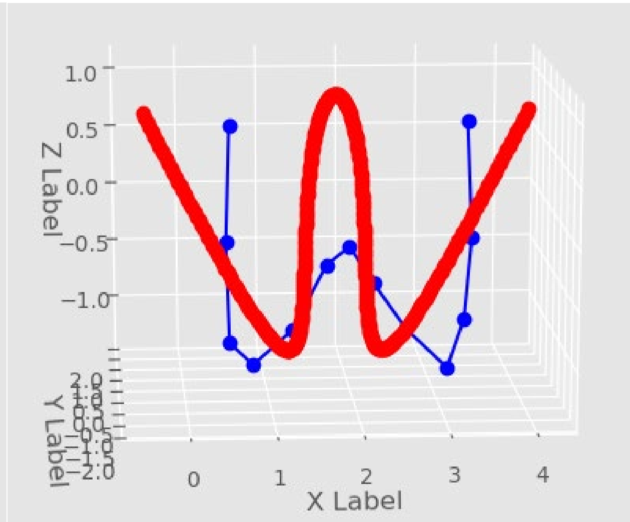


-add label here-

At frequency Exp 3.28 Hz, Num 3.28 Hz
MAC number = 0.744
Self MAC (num,num) = 1
Self MAC (exp,exp) = 1

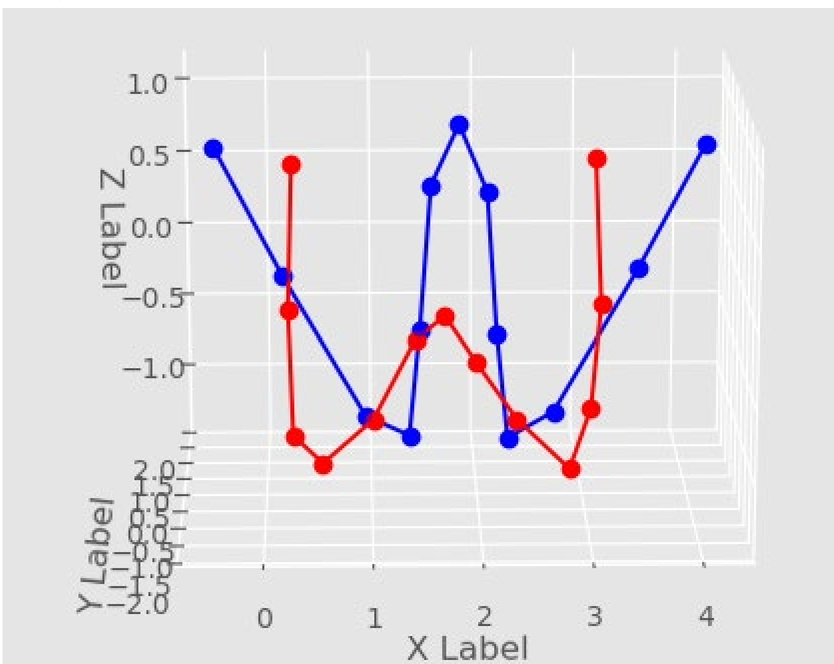


Full Experimental Data Set (Blue) with full Numerical Data Set (Red) (Num 6.36 Hz, Exp 5.91Hz)



Averaged Experimental data set (blue) with full numerical data set (red)

At exp frequency 6.36 Hz
MAC number = 0.4924



Averaged Experimental data set (red) with grid matched numerical data set (blue)

$$MAC(r,q) = \frac{\left| \{\varphi_A\}_r^T \{\varphi_X\}_q \right|^2}{\left(\{\varphi_A\}_r^T \{\varphi_A\}_r\right)\left(\{\varphi_X\}_q^T \{\varphi_X\}_q\right)},$$

Nomenclature	
$\{\varphi_X\}_q$	test modal vector, mode q
$\{\varphi_A\}_r$	compatible analytical modal vector, mode r
$\{\varphi_X\}_q^T$	transpose of $\{\varphi_X\}_q$
$\{\varphi_A\}_r^T$	transpose of $\{\varphi_A\}_r$
$\{\psi_X\}_q$	eigenvector defined the complex damped mode of vibration
$\{\psi_A\}_r$	eigenvector defined the complex damped mode of vibration
$\{\psi_X\}_q^*$	complex conjugate of $\{\psi_X\}_q$
$\{\psi_X\}_q^T$	transpose of $\{\psi_X\}_q$
$\{\psi_A\}_r^*$	complex conjugate of $\{\psi_A\}_r$

<u>July 20th 2020 : Weekly Report 7 - Synopsis for Week of July 13th – 17th</u>

- Primary focus: Address concerns showing incorrect plotting of numeric data which was the sum of static deformation data, coordinate position data, and additional modal data.
    - Upon further review two sources caused the misrespresented data
        1. The addition of static deformation to mode shape data in the processing file 'read_plot_beam_NASTRAN' located in the non-uniform beam→numerical data→ tests folder.
        2. The transposition of the imported data in MAC comparison script was done so incorrectly or manipulated the data incorrectly.
    - Solution: After diagnosing the problem and finding the proper way to add the data. I have decided to store the proper addition method in the MATLAB file and then directly import that into the MAC script. This should not impact the existing script significantly and will make handling the data much easier.
    - Below are a couple of modes of the numeric script based on my addition work (Figure 1). It should be noted that there are several modes that are odd and have endpoints perpendicular to the rest of the mode data sets, horizontally (See the green line in Figure 2)
- Goals this week:
    - Update MAC with new modal dataIsolate first mode (ideally a sin wave)
    - Compute MAC for this mode, adjust MAC equation as necessary

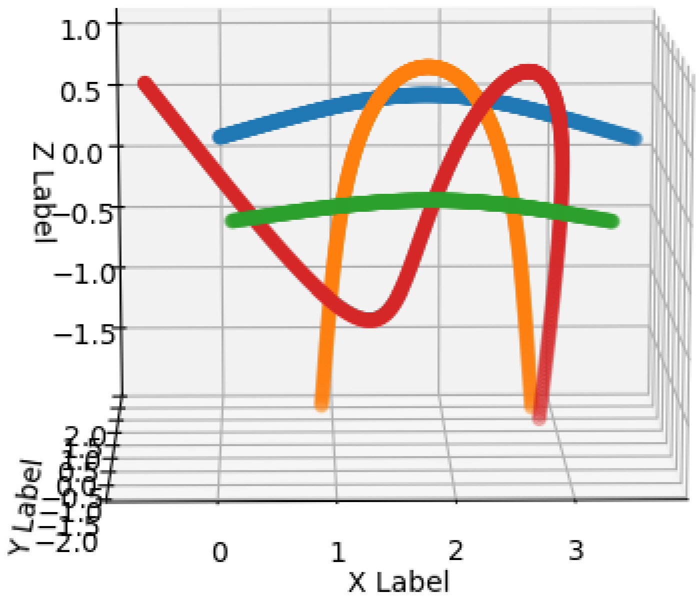


Figure 1: Modes 4,5,6,7 from numerical data set

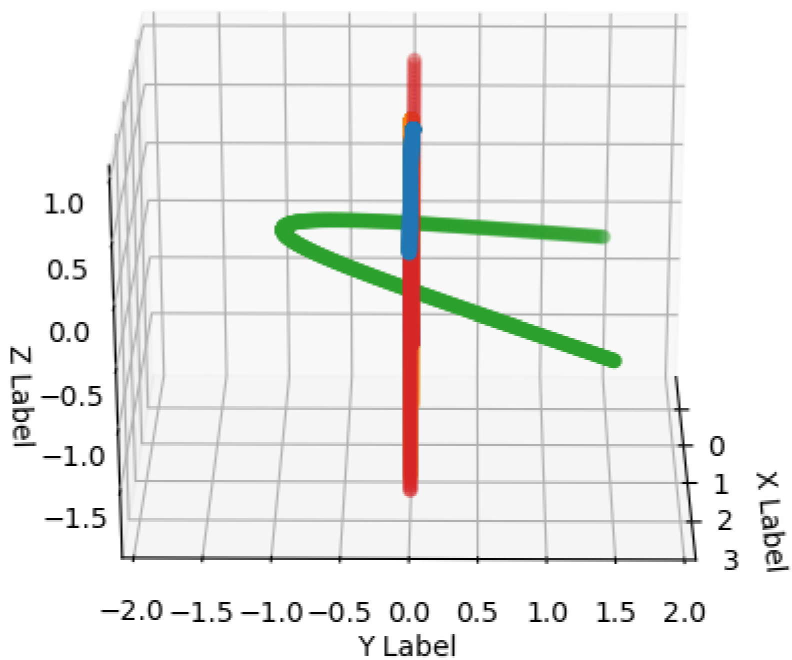


Figure 2 (Outlier mode in green, horizontal to the XY Plane)

Gupta paper has comparison modes etc
Use beam for perfecting mac for now

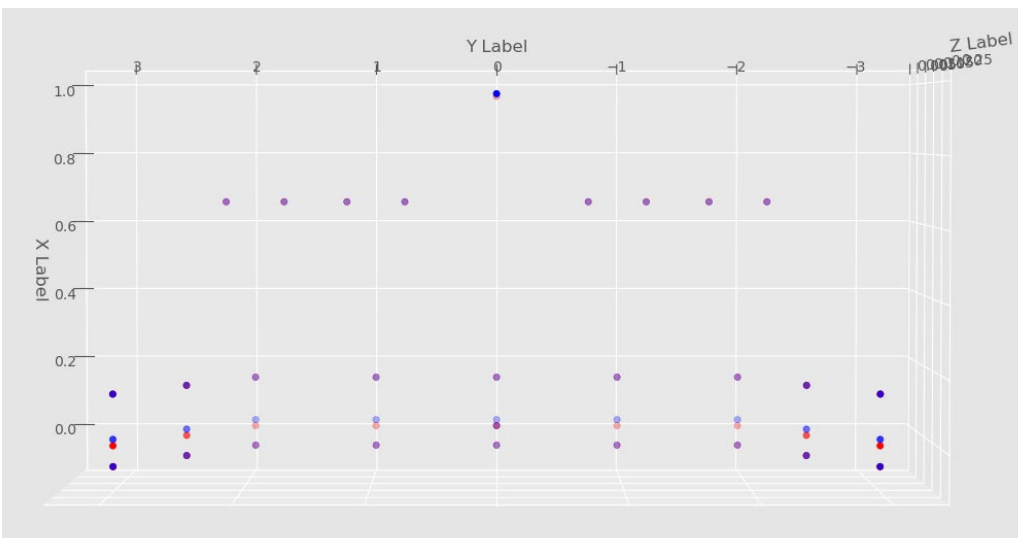
Concern; Mode shapes look worse for beam mode shapes
Tells me what phi means
Most likely contains

July 13th 2020: Weekly Report 6 - Synopsis of Week of July 6th, 2020 – July 15th, 2020

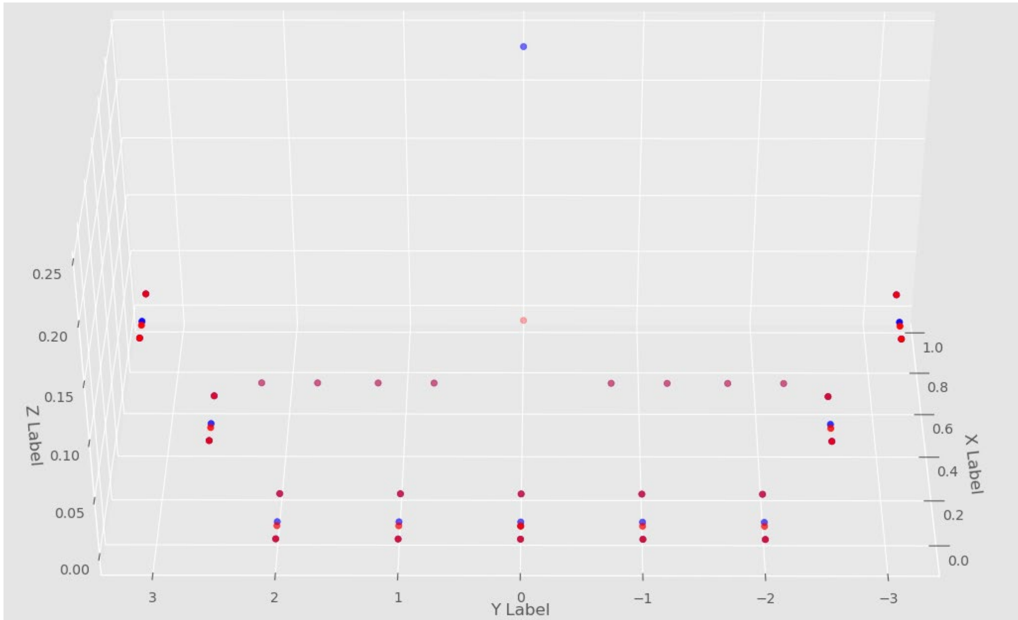
Special Section -- Bilal Requested Report:
The following is a proof of concept that the grid point algorithm matches correctly with a flat, undeformed aircraft as well as non-uniform beam data

Grid Matching

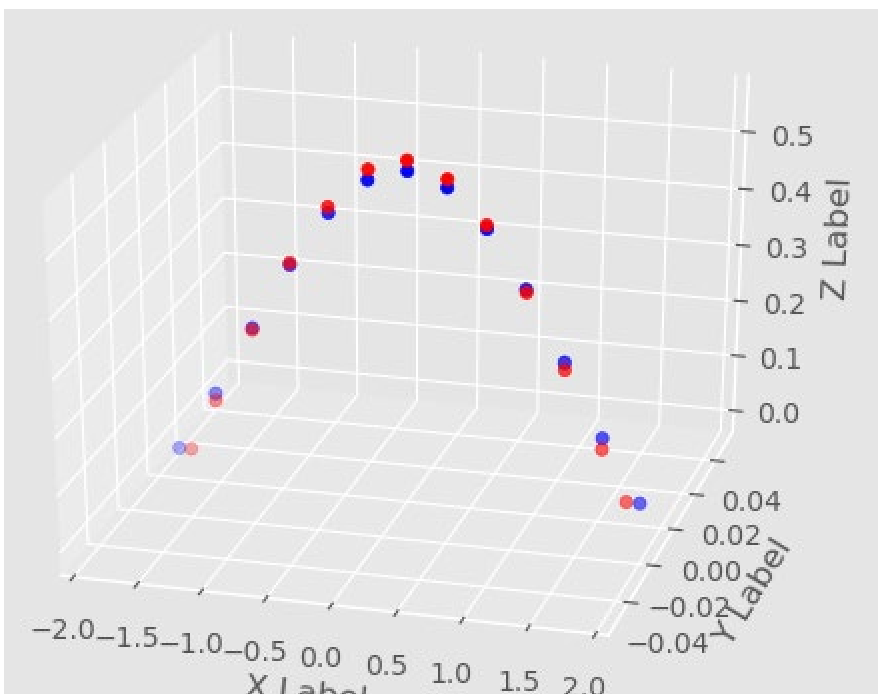
Below is a diagram demonstrating the capability of the organizeGrids function in the MAC script. The red markers are the experimental numerical matched grid points, while the blue markers are the original experimental grid points. The organizeGrids has proven capable of reducing and matching associated numerical grid points in the case of the undeformed aircraft, statically deformed aircraft, as well as a non-uniform beam, which required additional script editing. Images of matched gridpoints shown below:



Matched grid points of undeformed aircraft (Top-view)



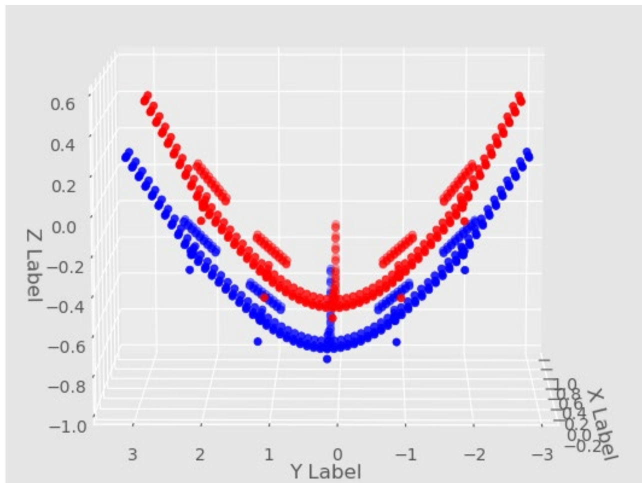
Matched grid points of undeformed aircraft (Front-Oblique View)



Matched grid points in static deformed state of non-uniform beam

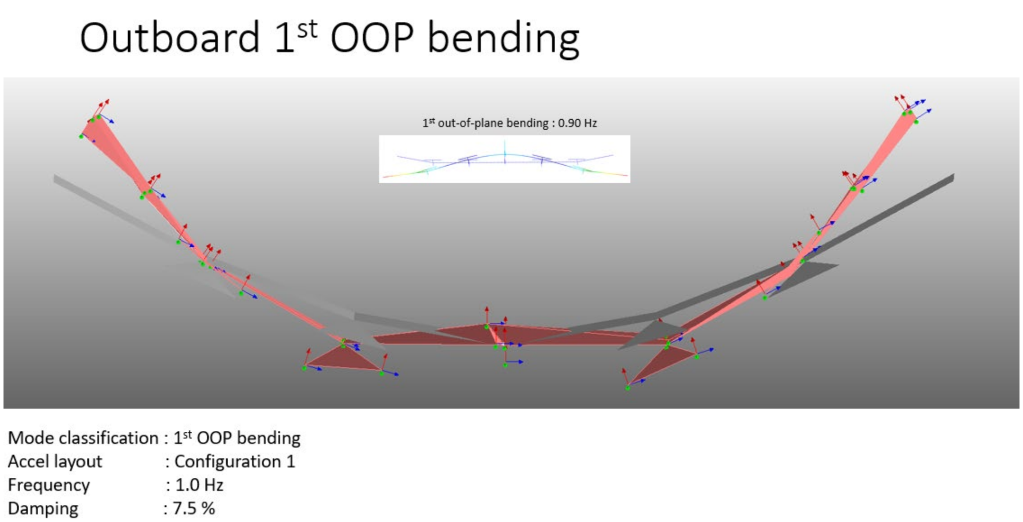
Concerns:

- MAC comparison of 1st Mode experimental to first Mode numerical results in MAC number of '1' despite experimental and numerical data being normalized in different formats. Additionally the visible graph (which may not be a credible test at this time) indicates very different modal shapes. The first scatter graph below depicts the change in deformed numerical coordinates based on the addition of the numerical coordinates to their associated eigen vector. The coordinates move from the deformed red state to the new modal shape in blue.



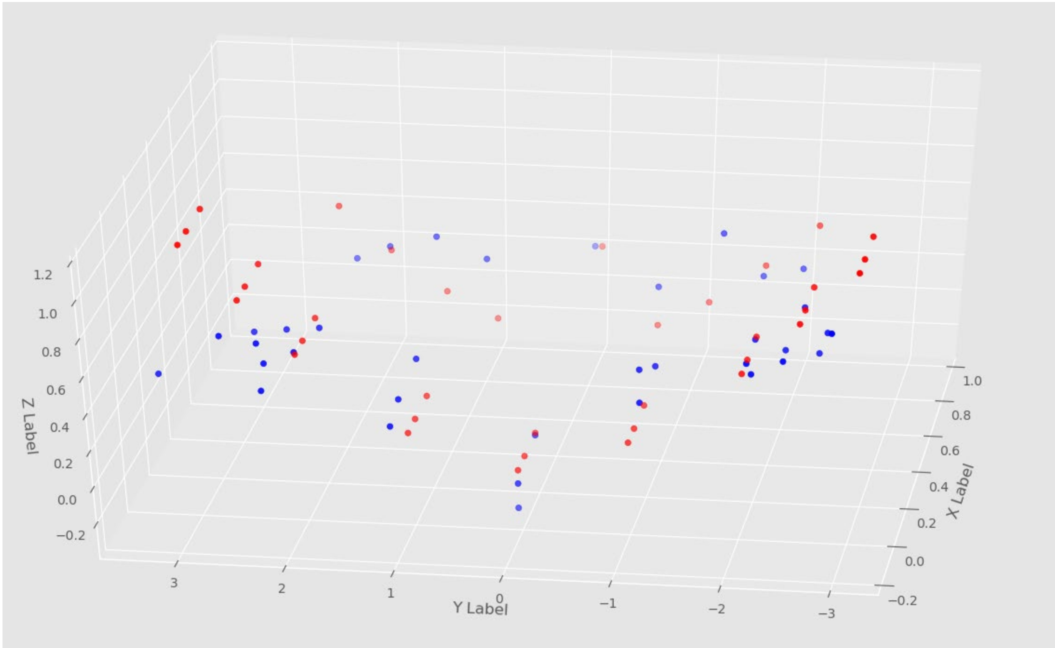
Numerical data 1st mode at 0.9Hz (blue) (Original coordinates in red)

The numerical modal aligns well with Bilal's simulated model. See image below (Small plot matches shape of figure above) (numerical does not have static def) (exp does)

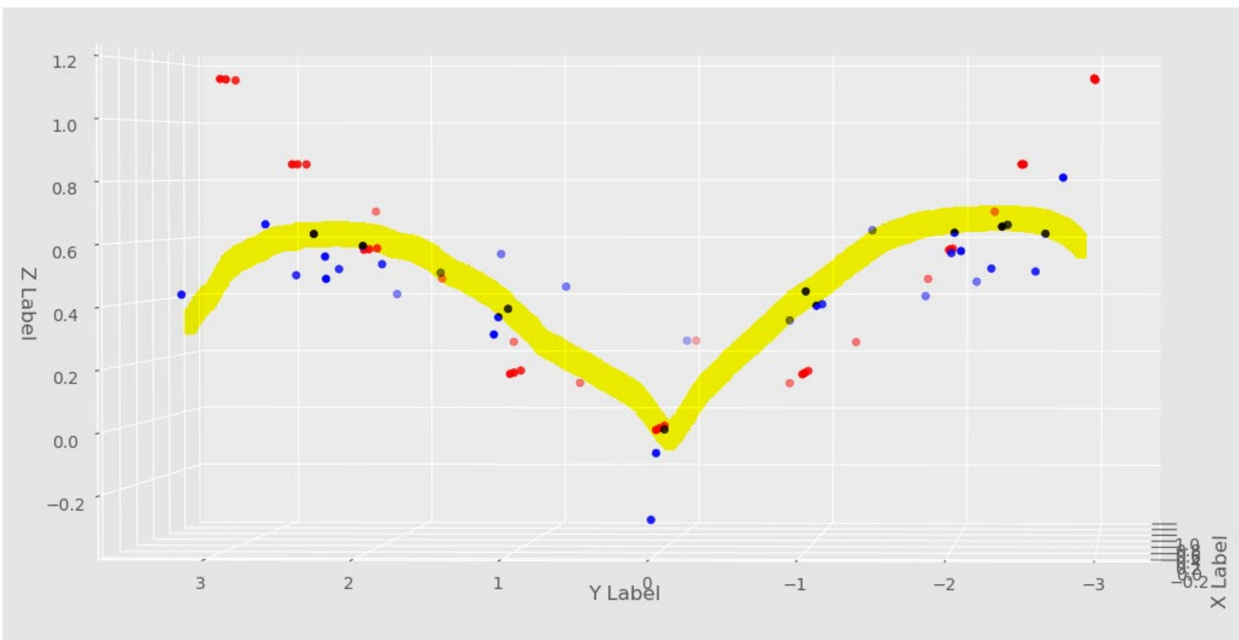


Bilal Prediction of First Mode

This however contrasts with the experimental data, mind you which is normalized differently. The experimental data shows a different shape, which might be an unreliable representation of the validity of the data given that the first mode is a sin wave. *See Bilal Modal document below

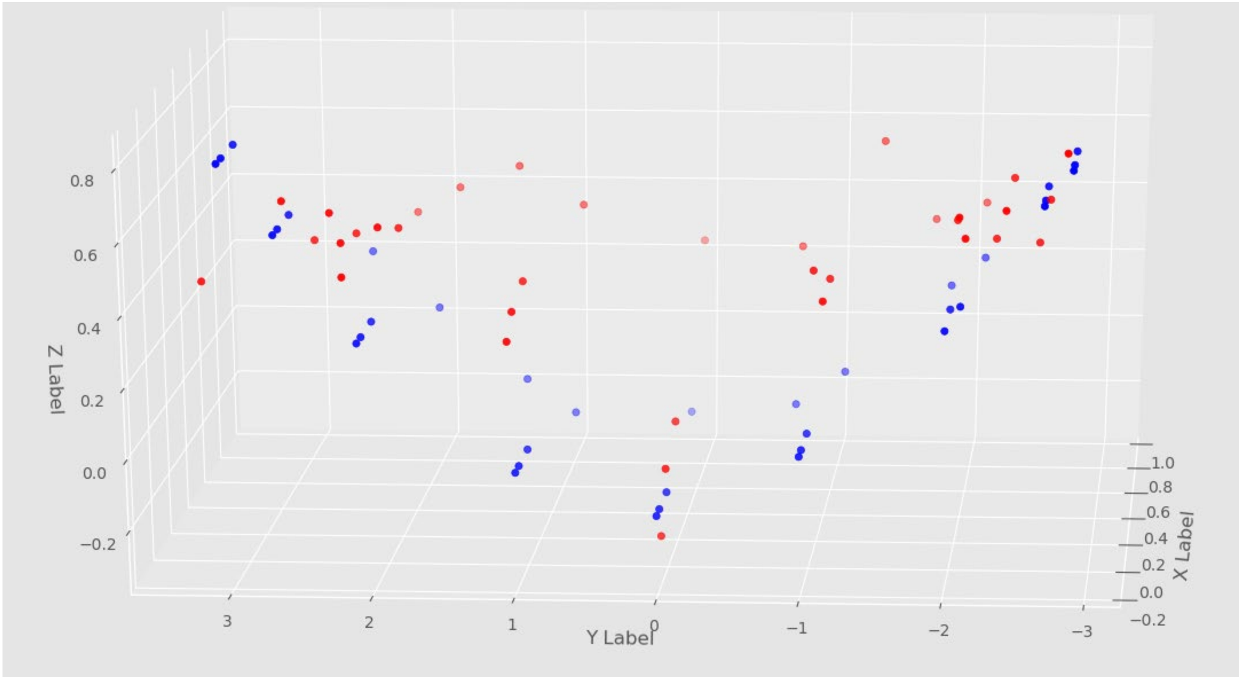


Front right oblique view of experimental coodinates moving from original position (red) to Fundamental Mode 1



Experimetnal Fundamental Mode 1 Takes form of gull shape

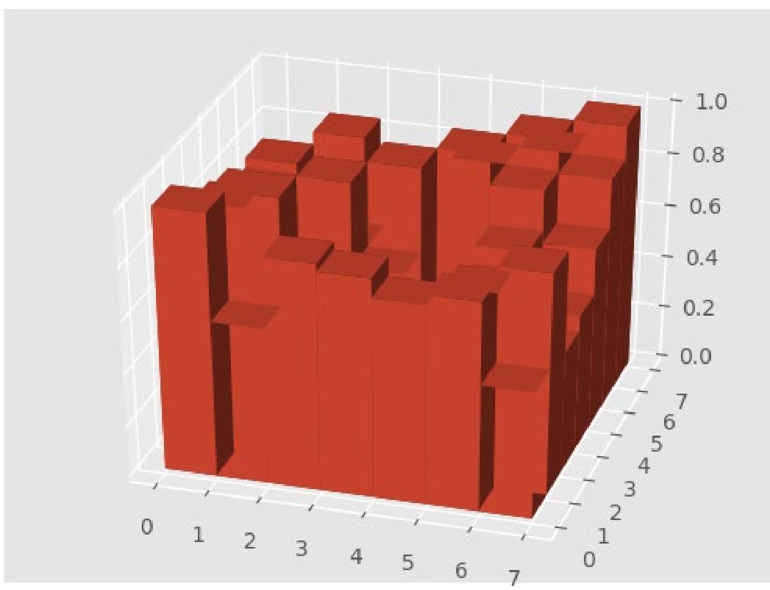
This 'gull shape' of the experimental mode might also resemble the first mode. See Bilal Presentation (simulated First Mode in white box) The visual representation may not be accurate though. Ask Bilal about this. Given the different shapes of the coordinate plus their respective eigenvector. Additional normalization may be necessary. Here are the two images overlaid on each other. Red is the experimental data, while blue is the numerical data.



Numerical(Blue) and Experimental(Red) Data of First Fundamental Mode

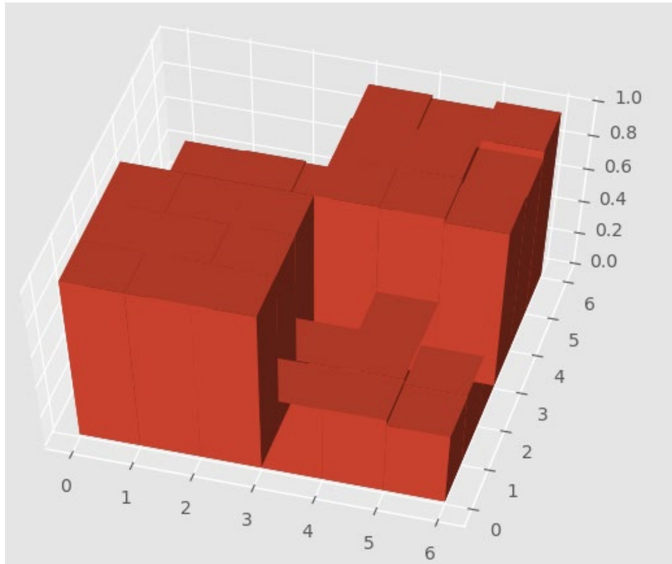
Concerns Continued – MAC Equation

- When completing a self mac with the experimental data. The following MAC result was calculated. The diagnol values corresponded to a value of '1,' which is expected when a data set is compared to itself



MAC of Experimental Mode Shape Fata (All Frequencies)

- Cause for concern arises though when looking at a self MAC comparing the numeric simulated data to itself.



Hand-picked grid points Numeric, Self MAC for Natural Frequencies (0.902838, 2.93288, 4.24894 , 4.47373, 7.14909, 8.30645)

Talk to Bilal about this. The shape is peculiar because so many values are close to one and may need to be further investigated. In addition to this a individual MAC was tested between the first three experimental modes and numerical modes. All test at a value of '1' despite being normalized differently. This warrants suspicion and requires further investigation.

Plans this Week

- Consult Bilal to see if further normalization is required of either numerical or experimental data sets.
- Use organizeGrids function to match a numeric grid point to an experimental grid point directly from a mode rather than the statically deformed state. This requires the correct normalization though and cannot likely be achieved prior
- Further investigate MAC equation accuracy

July 6th 2020: Weekly Report 5 - Synopsis of Week of June 29th, 2020

- Improved grid sort function using the following:
    - Added initial class definition to class compGrid to include storage of 'x' and 'y' tolerances. When sorting through the x and y grids, the code can now sort based upon a given tolerance
    - Revised X sorting process:
        - Once correct Y range of values are inputted into variable temp_list, algorithem uses nested if else statements to separate points into three categories
            1. Scenario where the experimental data x-coordinate is close to zero and numerical data x-coordinate is close to zero
                - Scenario 1 uses a essentially a small angle approximation to estimate percent difference. This equation is as follows:

                  Abs(Numerical X – Experimental X) * 10e-2

                  The values in this equation will be quite small, and will be always be stored as the smallest of possible x percent differences

2. Scenario where the experimental data x-coordinate is close to zero, but the numerical data point is not
    - Scenario 2 just prevents a divide by zero error by making the denominator and experimental data point in the percent difference equation very very small (~10e-3)
3. Scenario where the experimental data point is neither 1 or 2
    - Follows the typical percent difference formula relative to the experimental data point:

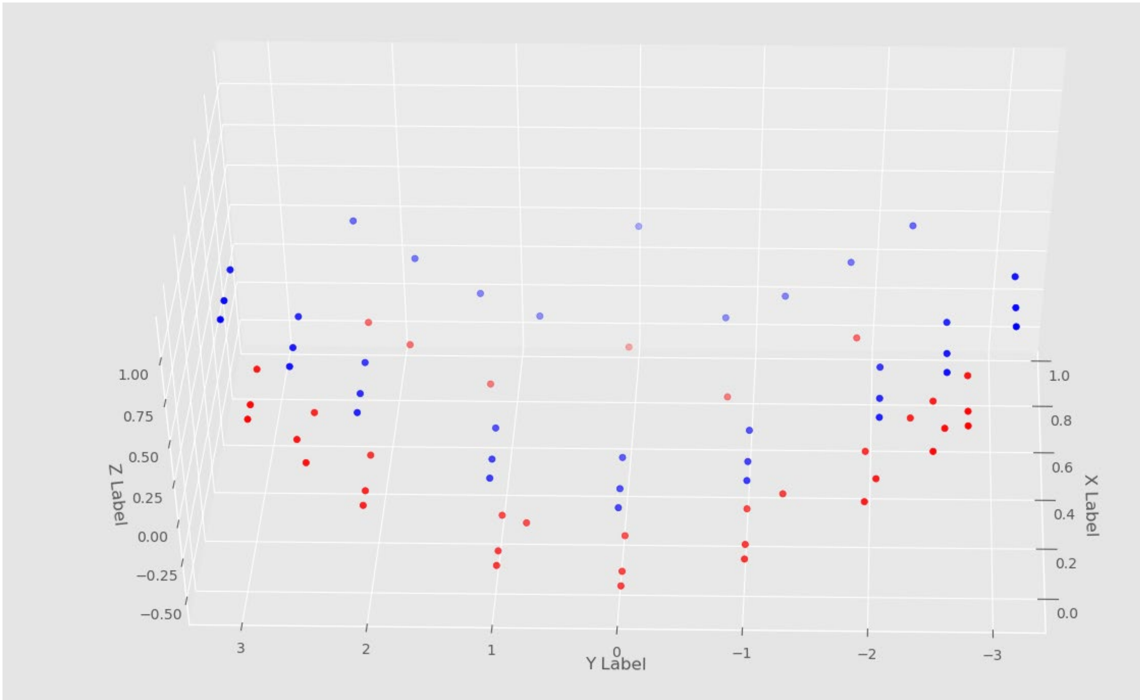
$$\frac{Num. x - Exp. x}{Exp. x}$$

      This creates a percentage and dimensionless number to implement tolerances

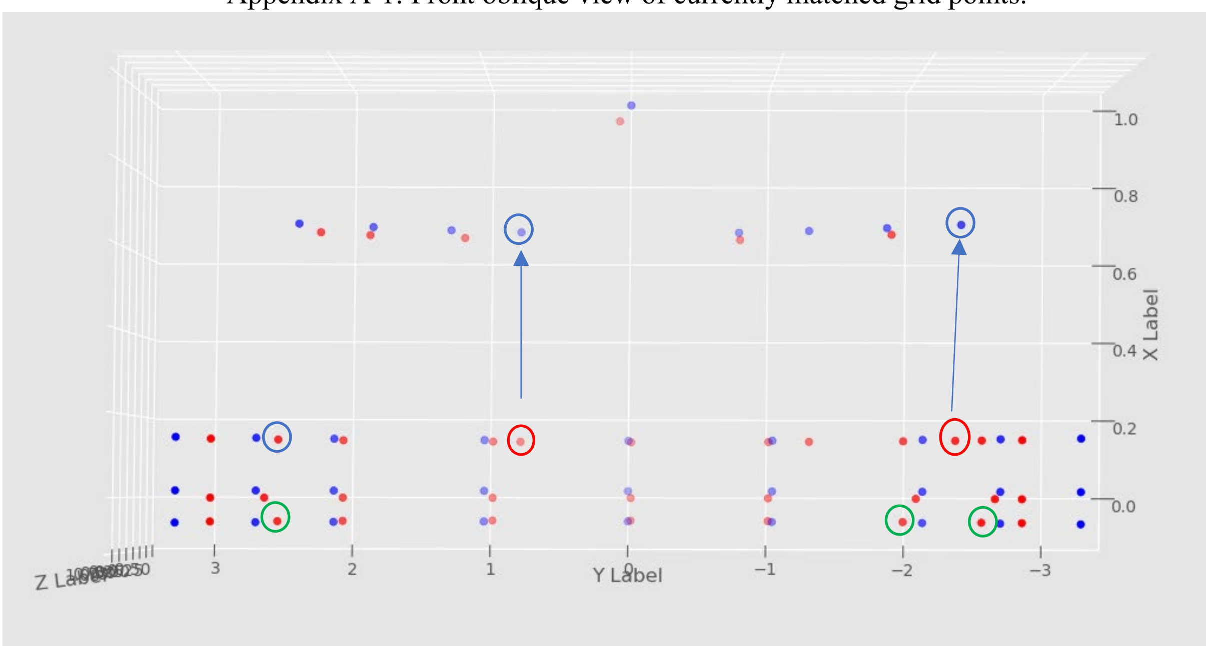
- After this sorting is complete. All points are filtered relative to being within 90% or 110% of the SMALLEST x tolerance. This new feature was created to get the smallest possible tolerances in x from a point. These grid points are added to a second list, sorted in order from highest to lowest tolerance, and the first index (the smallest x-tolerance) is used as the 'matched' grid point

*Note that this sorting method has also been implemented for the y functions as well

- Current problems
    - I have two grid point cases that correspond to a single outlier case in which a properly matched grid point lies outside of the initial sweep 5% tolerance (it has a tolerance of about 8% in this data set in the y coordinate, but has an extremely close x tolerance). It is excluded from the second more precise x coordinate sweep. Initially you could suggest increasing the y tolerance, but that would create a ripple effect down the line in which less precise points are selected. I have also tried adding additional inequality statements in which the first sweep could only pass in points that are within a certain percentage of percent difference, but this filter system was too stringent and grid points could not be matched.
- This week:
    - Find a way to match the last two grid points to their place in the tail.
        - Additionally find a way to tighten grid point selection to avoid anomalies in sorting (shown in green). They are relatively close to the point, but not quite the best match. See Appendix A-2
    - Test algorithm with beam data by Wednesday at the latest
    - Fix code to match with beam data in a more general form.



Appendix A-1: Front oblique view of currently matched grid points.



Correct positions (blue)

Appendix A-2: Top view of matched grid points
(Matching grid point with two tail outliers, circled in red. Additional anomalies in green)

<u>June 25th 2020: Weekly Report 4 - Synopsis of Week of June 21st, 2020</u>

- MAC function now has automatic 3D bar graph plot. Range is from 0 to 1.
- Plotted new 'captured' grids in 3D scatter plot for accurate representation of GVT node points
- Implemented new tolerance system (still needs to be debugged!!!!)

    - Automatically appends points between -0.1 and 0.1 for further inspection in 'y-axis' (span of the wing) (-0.1 < exp y < 0.1 && -0.1 < num y < 0.1)
        - This was meant to investigate numerical nodes on the line y = 0 line that seemed to have high percent difference between the numerical node and the experimental node (appx: y = 0.01)
        - If outside of the above bounds, it uses a 5% tolerance and then an 11% refined tolerance to look for points in the y span (Proven to work before except for cases in y = 0
        - This combined method does not work because of the new implemented state (first bullet point in series). Currently looking to refine it for further and more general use
    - A similar feature has been added to x search feature matching closest x value in numerical grids to those of the experimental grid points. Needed to capture points at x = 0
        - Unlike the two step refinement
        - Tolerance is between (-0.01 < exp x < 0.01 && -0.01 < num y < 0.01
            - Not capturing any points beyond x = -0.1 (estimate)
            - Needs to debugged for further use
- Tested theory of shifting origin of aircraft from (0,0) to point (4,4)
    - Increasing origin messed with tolerances which were based upon percent difference
        - This change would be too cumbersome and require even smaller tolerances. The larger the physical number is, the less of a dramatic percent difference between two points (separated by a small distance)
- **Work to do this week:**
    - Debug the issue of capturing points at y = 0 and x = 0
    - Create a more general tolerance system to replace the 5% and 11% tolerance.
        - I want to do this tolerance based on a percentage based on the undeformed distance between two nodes in span (y-direction) and a further refinement that is TBD
    - Create an alternative, more general solution to capture points near x = 0 and y = 0 lines
        - Perhaps if I combine the new proposed tolerance method with the origin shift method this could be feasible

June 20th 2020: Weekly Report 3 - Synopsis of Week of June 14th , 2020

- Fixed orderPhi function to include only comparisons of the 'z-component' of the evec-measurements (need to find out if the 'mode shapes' are actual evecs)
- Fixed the remove_Grid_Freq function to read in grid point I.D's rather than grid point I.D. indices
    - Still need to find an accurate method for frequencies to match proper mode shape with proper frequencies. (Numerical and experimental modes and frequencies do not match perfectly)
- Created organizeGrids function to create function. It will sift through undeformed grid points and find the closest comparison with the experimental equivalent undeformed state
    - Progress in function so far:
        - Created class with x component of num coordinates, y component of num coordinates, grid point ID

        - Based on this class idea it should be able to rearrange the y components in order of least to greatest (very difficult to do compared to MATLAB when using the 'sort' function and associated indices on MATLAB)
- **Plans for this week:** I plan to then match closest x and y value using a previous function I created before based upon tolerances. I am still having a few issues creating the class for all of the grid points. I also need to check the validity of the MAC matrix that was calculated. I will be working to plot that in 3D, but the matrix results have been calculated.

<u>June 15th 2020: Weekly Report 2 - Synopsis of Week of June 8th , 2020</u>

- Reviewed data files of sample MAC script in 'sample matlab code for manual comparison' folder.
- Consulted Bilal on how to complete MAC script on XHALE data.
- Mapped the 36 grid points in the experimental data set with 36 of the closest matching grid points in the numerical simulation. The numerical data had to be sorted via MATLAB due to its sheer volume, however the sorting methods used in MATLAB will likely be transferrable to the Python scripts.
- **Plans for this week**
    - Calculate the MAC matrix comparing numerical and experimental modes given the 36 experimental/numerical grid points (highest priority)
    - Set up MAC function to multiply eigenvectors for in the MAC equation in the Python MAC script
    - Using the matched grid points as the test case, create an automated function that sorts through numerical data points and matches them with their numerical simulation counterparts
    - Find way to match frequencies with their corresponding mode shapes. This presented a difficult problem previously because given an experimental frequency, the numerical result did not match in 4 cases.

<u>Matching Data Points (For reference)</u>

Experimental Data Points were made numbering from starting at most positive 'y' length value and moving to most negative 'y' length value. (Numbering 1 →36)

*Note Experimental Nodes 34,35,36 have large gaps in 'y' length when numerical grid value is compared to experimental value. This may cause issues farther down the road.

Exp. Node	Num Node	Exp. Coordinate	Num Coordinate
3	15420	(-0.05756,2.985,1.13)	(-0.05754,2,847,0.5326)
1	15020	(0.01762, 2.985,1.13)	(2.223e-5,2.847, 0.5326)
2	15220	(0.1424,2.985,1.13)	(0.1424,2.847,0.5326)
6	15416	(0.05756, 2.492,0.855)	(-0.05755, 2.508,0.3201)
4	15016	(0.01762, 2.492,0.855)	(1.129e-5, 2.508,0.3201)
5	15216	(0.1424, 2.492,0.855)	(0.1425, 2.508, 0.3201)
7	14004	(0.6572,2.24,0.6944)	(0.6586, 2.222, 0.08077)
10	10420	(-0.05756, 2, 0.58)	(-0.05756, 2, 0)

8	10002	(0.01762, 2, 0.58)	(0, 2, 0)
9	10220	(0.1424, 2, 0.58)	(0.1424, 2, 0)
11	13004	(0.6572, 1.76, 0.4656)	(0.6555, 1.76, -0.1025)
12	9003	(0.6572, 1.24,0.25)	(0.6579,1.238,-0.3167)
15	5419	(-0.05756, 1, 0.18)	(-0.05762, 0.9692, -0.3816)
13	5402	(0.01762, 1, 0.18)	(-4.12e-5, 1.046, -0.3601)
14	5220	(1.424, 1, 0.18)	(0.1424, 1.066, -0.3549)
16	8004	(0.6572,0.76,0.11)	(0.6557,0.836,-0.4372)
20	16411	(-0.05756, 0, 0)	(-0.05771, -0.01882, -0.5087)
17	16011	(0.01762, 0,0)	(-0.000152,-0.01883,-0.509)
19	16211	(0.1424, 0, 0)	(0.1423, -0.01887, -0.5098)
18	4005	(0.966, 0, 0.24)	(0.9741, 0.08117, -0.3066)
11	19003	(0.6572, -0.76, 0.11)	(0.6559, -0.7312, -0.4202)
24	21411	(-0.05756, -1, 0.18)	(-0.05759, -0.9984,-0.3241)
22	21011	(0.01762, -1, 0.18)	(-2.739e-5, -0.9985,-0.3243)
23	21211	(0.1424, -1, 0.18)	(0.1424, -0.9986,-0.3247)
21	19002	(0.6572, -0.76, 0.11)	(0.6562, -0.7887, -0.403)
29	26412	(-0.05756, -2, 0.58)	(-0.05756, -2.007, 0.107)
27	26012	(0.01762, -2, 0.58)	(2.213e-6, -2, 0.58)
28	26212	(0.1424, -2, 0.58)	(0.1424, -2.007, 0.107)
26	24001	(0.6572, -1.76, 0.4656)	(0.6566, -1.782, -0.03442)
33	26418	(-0.05756, -2.492, 0.855)	(-0.05754,-2.5148,0.4263)
31	26018	(0.01762, -2.492,0.855)	(1.592e-5, -2.515,0.4263)
32	26218	(0.1424, -2.492, 0.855)	(0.1425, -2.515, 0.4263
30	25004	(0.6572, -2.24,0.6944)	(0.6587,-2.06,0.08008)
36	26420	(-0.05756, -2.985, 1.13)	(-0.05754, -2.684, 0.5327)
34	26020	(0.01762, -2.985, 1.13)	(2.122e-5, -2.684, 0.5327)
35	26220	(0.1424, -2.985, 1.13)	(0.1425, -2.684, 0.5327)

June 8th, 2020: Weekly Report 1 - Synopsis of Week of June 1st , 2020

- NASTRAN: Simulated 1D and 2D beam in PATRAN. Was issued corrective measures on how to dimension 2D shell mesh and beam object

- Python: Interpreted original MAC script given to by Bilal. Made several key adjustments as follows
    - Imported numerical and experimental data directly from .MAT (MATLAB) files into MAC script
    - Transposed the 25 x 6 x 348 matrix to workable format of dimension 348 x 25 x 6. This took significant time because MATLAB and Python store and index three dimensional arrays differently. (MATLAB array → row, column, sheet; Python array → sheet, row, column)
    - Added the numerical "deformed state" + "static state" from cell array in MATLAB
    - Given a set number of grid location points (based upon index of imported numerical grid data, not actual node location point), reduced the number of grid points in node matrix from 348 x 25 x 6 to _grid points #_ x 25 x 6,
    - Given the indexes of the numerical frequencies that best match the experimental frequencies, reduced the node's matrix from 348 x 25 x 6 to 348 x __# of experimental frequencies__ x 6.
- **Plans for This Week:**
    - Create function that given experimental frequencies, creates an array with the indices of the numerical frequencies that most accurately match the given experimental frequencies. (This will be based upon a % tolerance that is TBD)
    - Create function that uses location of grid points to match experimental grid locations with their numerical counterparts.