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

Weekly Reports Summer 2020

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.

Synopsis of Week of June 8th , 2020

- 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.

Matching Data Points (For reference)
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)

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.