121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #9814

MATLAB-Based Finite Element Analysis in a Vibrations Class

Dr. John R. Baker, University of Kentucky

John R. Baker is an Associate Professor of Mechanical Engineering at the University of Kentucky Extended Campus Program in Paducah, KY. He received his B.S., M.S., and Ph.D. in Mechanical Engineering from the University of Kentucky in Lexington, KY. After obtaining his B.S., he spent three years working in the Plastics Division of Eastman Chemical Products, Inc. He entered his current position in July 2000.

MATLAB-Based Finite Element Analysis in a Vibrations Class

Abstract

This paper overviews MATLAB®-based assignments developed and implemented in a mechanical vibrations class which utilize finite element analysis (FEA) for structural vibration calculations. The course is dual level and includes upper-level undergraduates taking it as a technical elective, and graduate students taking it for graduate credit. In dual level courses, there are additional requirements for graduate credit as compared to the work required for undergraduate credit. The course is offered via ITV (Interactive Television).

A primary component of the paper is an overview of a graduate student project that requires the students to produce a flexible FEA program to analyze beam vibration using the MATLAB user function capability. This work is intended to solidify for the students the basics of structural vibration analysis, including calculation of natural frequencies and mode shapes, and also forced harmonic response analysis. It also provides an introduction to some students on FEA. Those who already have FEA experience gain more insight into development of a structural model, including mass, stiffness, and damping matrices, than they may obtain from use of standard FEA software, such as ANSYS®. Also, the students gain valuable programming experience, and better knowledge of the widely-used mathematical software tool, MATLAB.

While undergraduates are not required to develop an FEA program, they utilize the finite element method through MATLAB functions developed by the instructor and provided to them, and the programming approach is outlined in lecture material.

In some assignments, students compare finite element results to results based on other analysis methods to verify for them that the finite element analysis method is reliable. They also compare results from FEA implemented using MATLAB user functions to results from the commercial FEA software, ANSYS. The assignments provide some insight into important issues such as how results can be impacted by mesh density. This computational work complements a more traditional coverage of vibrations topics, which is the main basis of the course.

Background

As is the case with many engineering courses, there is much valuable information that could be covered in a single semester vibrations course, but the available time is limited. One challenge is prioritizing the topics to be covered. For engineering students to be well-prepared upon graduation, they need a good understanding of the underlying physics for the phenomena they study, and they also need up-to-date background on the tools used in industrial and research settings to analyze behavior of systems. FEA is currently the primary tool used in analyzing vibrations of structures. In any practical use, FEA is implemented in a computer program. It is a computer-based method of analyzing systems. It seems only reasonable to attempt to solve very small academic-type textbook FEA problems with a pencil, paper, and calculator. Commercial FEA software, such as ANSYS[1], has advanced to the point that a person with essentially no background in vibration theory or in the fundamentals of the finite element method can readily obtain solutions for vibration characteristics of complicated structures. It seems, however, that a competent engineer should have an understanding of the basis for the software calculations and enough knowledge of vibration theory to notice when software results are unreasonable. MATLAB[2], a widely-used software package, is a very useful tool in providing students opportunities to become more acquainted with FEA background, such as assembly of the global system mass and stiffness matrices, and the matrix algebra operations involved in solving large structural vibration problems. When a student writes his or her own finite element code in a MATLAB user function, it should solidify, at least to some extent, an understanding of what commercial FEA software is doing "behind the scenes". It is hoped that the exercise of writing FEA code with at least some level of complexity may eliminate any mystery for the students regarding how commercial codes arrive at results, such as natural frequencies and mode shapes, and harmonic response amplitudes and phase angles.

Inclusion of FEA in a vibrations course is certainly not a new idea. A quick review of five vibrations textbooks with a wide range of publication dates indicates that FEA has been included in standard vibrations textbooks for many years. Emphasis varies among authors. Of course, different textbooks have different target audiences and goals. The most recent edition of the textbook by Thomson and Dahleh[3] has a copyright date of 1998, and includes an introduction to finite element methods in Chapter 10. The text has a total of 14 chapters. The textbook by Meirovitch[4] includes a chapter on finite element methods, Chapter 8 of 12 total chapters, and its copyright date was 1986. A very recent edition of a vibration textbook by Inman[5], with a copyright date of 2014, discusses FEA in its final chapter, Chapter 8. However, the text by Balachandran and Magrab[6], with a copyright date of 2009, does not appear to cover FEA. Neither does the text by Tongue[7], with a copyright date of 2004. An exhaustive study of the treatment of FEA in vibrations textbooks is beyond the scope of this paper, but the differences in emphasis in these texts is noted here to illustrate the fact that, while FEA has been used for many years in vibration analysis in industry, it seems reasonable to conclude that, at this time, there is not universal agreement in the academic community on how much emphasis should be placed on FEA in a typical vibrations textbook.

In addition to coverage in textbooks, inclusion of FEA in vibrations courses is outlined in other work. For example, Hagigat[8] describes use of the ANSYS finite element analysis program to solve vibration problems in a vibrations class, and states that FEA in the course supplements classical vibration analysis. The current author has reported on application of ANSYS FEA software in teaching vibrations and other courses[9,10].

Use of MATLAB in the teaching of vibrations has also been reported multiple times. For example, a MATLAB script to animate beam vibration is reported in Jacquot, et al.[11] That work presents several examples and states that the script should be of use to those teaching a second course in vibrations. It does not discuss use of the script in a specific course.

Additional examples of works involving various levels of use of FEA and MATLAB in engineering education and in particular, in vibrations, can be found because FEA and MATLAB are widely-used tools in engineering. This paper primarily overviews some specific examples of use of the computer-based FEA in a mechanical vibrations course implemented through the use of the software package, MATLAB. This paper focuses on a specific graduate student project, and discusses four years of experience in implementing this project in class.

In the sections that follow, some details of the vibrations course are summarized. The FEA-based graduate student project is then described. Assessment of the graduate student project is then discussed, where the assessment is based on student performance on the project and student responses to questions on a survey. A discussion of possible modifications to the graduate student project is then provided. Additional homework assignments related to FEA are then described, and the final section summarizes the overall work.

Mechanical Vibrations Course Overview

Mechanical Vibrations (ME-513) at the University of Kentucky includes undergraduate students taking the course as an elective and graduate students taking the course for graduate credit. It is offered once a year, in the fall semester. It is taught via ITV (Interactive Television), with undergraduate students at the extended campus program in Paducah, KY taking the course simultaneously with undergraduate and graduate students on the main campus in Lexington, KY. The students taking the course in a given semester have a wide range of backgrounds. There is sometimes a significant gap between the math skills and computer programming skills of a graduate student as compared to an upper-level undergraduate.

It is university policy that students taking a 500-level course for graduate credit must do some work beyond that required for undergraduate students taking the same course. The difference in required work is left to the instructor's discretion and must be described in the course syllabus. Of course, both undergraduates and graduate students attend the same course lectures. So, for both sets of students, the same basic course material is covered. In the mechanical vibrations course, the primary difference is that the graduate students complete a project that is not required of the undergraduates. The graduate student project grade is used in place of about half the homework credit for graduate students. So, for example, in fall 2013, homework accounted for 15% of an undergraduate's grade, but homework accounted for only 8% of a graduate student's grade, and the graduate student project accounted for 7%. The rest of the grade distribution involved exams and was the same for both groups. However, graduate students were still required to complete all of the homework assignments as undergraduates. A second difference in requirements for the two groups is that occasionally graduate students have some additional work required on a given homework assignment as compared to undergraduates.

While the assignments described in this paper show a significant emphasis on computer-based FEA, the course also includes the more traditional approach to teaching vibrations, such as analysis of lumped parameter models of simple systems with few numbers of degrees of freedom (DOF), and partial differential equation solutions of continuous systems.

Graduate Student Project

Overview

Various projects were assigned for graduate students from the time the current instructor (this paper's author) first taught the course as a dual-level course in 2005 until 2009. In 2010, a project requiring each graduate student to develop a somewhat flexible finite element analysis program for beam vibration analysis was implemented, and it has been used, with some modifications, for four consecutive offerings of the course, 2010-2013. The program is developed using the MATLAB user function capability (some may refer to MATLAB user function files as "m-files" because they have a ".m" extension to their filename).

The assignment sheet for the most recent course offering in fall, 2013, is provided as an appendix. The basic features are:

- The finite element analysis program must perform modal analysis and steady-state harmonic response analysis based on a 2D beam assumption.
- The modal analysis capability allows for calculation of undamped natural frequencies and the corresponding mode shapes.
- In the harmonic response analysis option, a sinusoidal load, or loads, is assumed to act on the structure, viscous damping is included in the model, and the steady-state response amplitudes and phase angles are calculated for every DOF in the model.
- The program must allow for four different sets of boundary conditions at the ends of the beam being analyzed.
- The program must conform to a specific set of user input guidelines and return results in a specific format.
- The students must perform analyses for five specific test cases, and submit a report on the results, along with their MATLAB user function file.

Project Goals

Some goals of the project are:

Goal 1: Students will gain some background in FEA applied to vibration problems.

Goal 2: Students will gain some understanding of the relationship between the mode shapes calculated in an undamped modal analysis and the steady-state response when a lightly-damped structure is excited at or near one of its natural frequencies.

Goal 3: Students will gain some understanding of the convergence of FEA results to those obtained from partial differential equation solutions for beams as the finite element mesh density is increased.

Goal 4: Students will gain some experience in structured computer programming.

Regarding Goal 1, the time available in the course does not allow for extensive coverage on details of the FEA. However, in the course, finite element mass and stiffness matrices are developed for simple spring and lumped mass systems, and also for rods, where the rod elements allow only for axial motion. The method of superposition for assembling global system mass and stiffness matrices from the individual element matrices is also presented, as well as implementation of external loads in the model and the application of boundary conditions. Regarding 2D beam elements, a derivation of the element mass and stiffness matrices is not included in the course material. But, the beam element matrices are provided to the students, and they are instructed that the assemblage of global system matrices and application of boundary conditions is the same for beam elements as it is for the simpler elements. The 2D beam elements used have two end nodes per element. Assuming the beam element x-direction to be along the beam element's axis, there are two degrees of freedom per node: UY (lateral displacement in the y-direction) and RZ (rotation about the z-direction).

In the course lectures and homework, a considerable amount of time is spent on simple lumped parameter spring-mass-damper systems. Coupled second-order differential equations are written in matrix form, and the system mass and stiffness matrices, and sometimes a damping matrix, are identified, and various types of calculations are performed. A theme repeated often to the students is that if they can understand the basics, such as calculation of natural frequencies and mode shapes, for a relatively simple 2-DOF case, then they really understand the basics for large system models, which are typically developed using computer-based FEA. The project is intended, in part, to clearly illustrate that the finite element method can be thought of as a systematic way of generating the system mass, stiffness, and damping matrices, and that the method is very amenable to implementation in a computer program. Although there is readily available finite element software on the market, it seems the details of how the mathematical models are created and how the calculations are performed by the software may be lost on some students if they only use commercial FEA software.

Regarding Goal 2, one of the test cases the students consider, Test Case 3 (see appendix), involves calculation of natural frequencies and mode shapes for a simply supported beam (a beam with pinned constraints at each end). Another test case they consider, Test Case 5, involves calculation of steady-state response amplitudes and phase angles for a very lightly-damped simply supported beam assuming a sinusoidal force acts on the beam in the lateral direction (perpendicular to the beam's axis). The required analysis and plots for Test Case 5 illustrate the fact that if a lightly-damped structure is excited at or near one of its natural frequencies, the response is dominated by the mode being excited. In other words, a snapshot of the beam when deformed in steady-state motion would reveal approximately the deformed shape defined by the mode shape of the corresponding mode.

To better explain how Test Cases 3 and 5 address Goal 2, consider the mode 2 mode shape of a simply supported beam. It is illustrated in Figure 1 and appears as a full sine wave from x=0 to x=L, where x is the coordinate along the beam's axis, and L is the total length of the beam. The students should see from Test Case 3 that the mode shape printout for mode 2 has the opposite sign for relative UY amplitudes on the left half of the beam as compared to the relative UY amplitudes on the right half. Test Case 5 includes calculation of steady-state responses when the beam is lightly-damped and excited with a sinusoidal force at the mode 2 natural frequency. Students plot the response amplitude along the beam. However, if their programs work correctly, they should calculate approximately a 180^{o} phase shift between steady-state UY responses on the left half of the beam as compared to those on the right half. Therefore, they are instructed to use a different sign in the plot of steady-state response amplitude depending on the phase angle for a given UY DOF response. The resulting steady-state response plot looks like that of the mode shape for the mode being excited. This is intended to solidify for the students the concept that when a lightly-damped structure is excited at or near one of its natural frequencies, the ensuing steady-state motion is approximately that indicated by the corresponding mode shape vector.

Regarding Goal 3, Test Case 1 in the project involves a mesh convergence study and the natural frequencies from the finite element method converge to the results from a partial differential equation solution. The beam partial differential equation solution is provided in the project assignment sheet and also covered in the class lecture material and the course textbook[3]. The students also compare results for node points (locations of zero motion for a given mode shape) for various modes and boundary conditions from their FEA solutions to those published from a partial differential equation solution in a textbook by Balachandran and Magrab[12].

Regarding Goal 4, the students are given specific instructions on the user input format and the results output format. For instance, the same program must be able to handle four different types of boundary conditions, where the boundary conditions for a given analysis are specified based on the value of a parameter in a user-defined matrix (see the instructions for user inputs "ui1", "ui2", and "ui3" in the appendix).

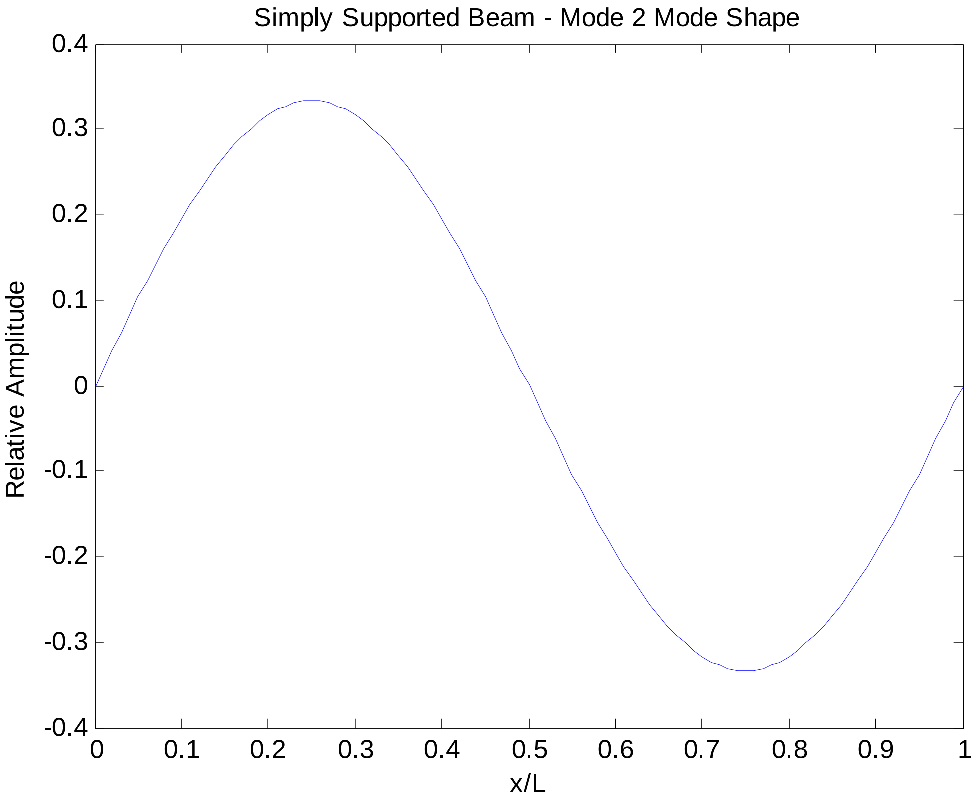

Figure 1: Beam Mode Shape.

From discussions with students regarding this assignment over the years, it has been noted that some are not well acclimated to the idea of writing flexible programs (or "functions" or "scripts"). Many seem much more comfortable using software to solve what they tend to view as separate, standalone problems. When they move from one problem to the next, many seem to view it as a time to start over from scratch. Of course, the students in the course are mechanical and civil engineers, and not students who are preparing for a computer programming career. However, it seems for them to be efficient in their careers, they need to consider methods for automating tasks when appropriate. It is hoped that the experience with this project will benefit some students by prompting them to think about how to automate repetitive tasks.

Graduate Student Project Assessment

Student Performance

In the past three course offerings, 2011-2013, 24 graduate students have completed the project. Instructions each year are altered to some extent in an attempt to prevent a student using a program submitted by another student in an earlier year. But, the same basic project has been assigned each of these three years. The first time a similar assignment was used was in 2010. But, there were fairly significant changes to the assignment between 2010 and 2011. So, this assessment section includes only the years 2011-2013.

All student submissions have been retained by the instructor since 2011. As a method for assessing and comparing student performance on the project over the years, the same assessment modal analysis test case was run on all 24 student programs based on a simply-supported (pinned-pinned) boundary condition assumption. Specific results were spot checked for accuracy and conformity with project instructions. Also, the same assessment steady-state harmonic response analysis test case was run on all 24 student programs based on a pinned-clamped boundary condition assumption. For this test case, also, specific results were spot checked for accuracy and conformity with project instructions. All submitted reports over the three year period were also reviewed. Summaries of the results of this assessment of student work are provided in Table 1, where the 24 students from the three class offerings are combined and listed in a random order. Each individual student is randomly assigned a label, such as "Student A". The two test cases used for assessment differ from the five test cases the students are required to run so that bugs may be found in the programs that weren't found by the students in running their required test cases.

As shown in Table 1, 19 of 24 students developed a flexible program that appears to successfully solve the assessment modal analysis test case, but the programs submitted by 9 of those 19 students had some problems related to conformity with the instructions. For example, the instructions state that the program should list natural frequencies in units of Hz, but some students had programs that list them in radians per second. Only 14 of 24 students developed a program that appears to successfully solve the more challenging assessment harmonic response analysis test case, and the programs submitted by 6 of those 14 students had some problems related to conformity with the instructions.

There was no attempt to diagnose problems with the students' programs, so it is quite possible that in some cases there were very minor bugs. It is noted that the far right column headed "Report" in Table 1 indicates that 19 of 24 students submitted a report deemed acceptable (OK) by the instructor. In this discussion, that is not intended to indicate that a report was flawless, but simply that the student basically covered all aspects required and submitted their work in a reasonable technical report format.

It should be noted that in some cases, a student may have "OK" listed for the report, but a significant problem is noted for the assessment modal analysis and/or harmonic response analysis test cases in Table 1. In some cases, it seems that students had significant difficulties writing a flexible program, but were able to perform the needed calculations for each of the five test cases required of them using some form of MATLAB script that perhaps did not conform to the project input guidelines. It was still considered acceptable for students to report results for the test cases from such a nonconforming script. Also, of course, their program may have worked well for the five test cases they were required to analyze, so they didn't identify a bug that was evident when the assessment test cases were run.

Development of a flexible program conforming to the instructions was only one goal of the project. It seems the students should still benefit from considering their five test case results, and they were encouraged to submit the best program and report they could develop, even if there were some flaws. So, for example, "Student F" did a nice job on a report. The submitted program handled all five required test cases. But, the program that was submitted was actually a series of separate scripts that successfully performed the calculations for the five test cases. While the student's MATLAB script did not conform to the instructions, it still took some significant programming effort to perform the global system matrix assembly for each case, calculate the required results, and generate required plots.

With 19 of 24 students submitting an acceptable report in the past three course offerings, it seems reasonable to conclude that overall, the project has been fairly successful in accomplishing its goals outlined earlier. The project is intended to be challenging and, it could be argued, if the student success were much higher, then perhaps the project difficulty level should be increased.

Of course, a realistic assessment of a project by an instructor should include consideration from a student's perspective of the relationship between effort and time expended and the expected impact on his or her final course grade. The project accounts for only 7% of a graduate student's grade. So, it is certainly possible that some who submitted flawed programs and/or reports would have fared better on the project if a larger proportion of the course grade depended on this project.

Table 1: Results of Graduate Student Projects 2011-2013 (24 Students)

Student	Modal Analysis	Harmonic Response Analysis	Report
A	OK	OK	OK
B	OK	Results Incorrect	OK
C	OK	OK	OK, but problems with phase angles on plots in Case 5.
D	OK	OK	OK
E	OK, but natural frequencies listed in rad/s instead of Hz. Also, node numbering problem in the printed mode shapes.	Results calculated over a freq. range that differs from user input.	OK
F	Program is actually a separate script for each test case; user input is ignored.	Program is actually a separate script for each test case; user input is ignored.	OK
G	OK, but natural frequencies listed in rad/s instead of Hz. Also, printed mode shapes not normalized according to the instructions.	Response magnitudes incorrect; phase angles correct. However, phase angles reported in radians, instead of degrees, and user has to specify the frequency range in rad/s, instead of Hz.	OK
H	OK	OK	OK, except mode shape plots included RZ DOF, but should only include UY. Case 5 harmonic response plots incorrect.
I	Results Incorrect	Program will not run this analysis.	No results
J	OK	OK	Results OK, but submitted test case results simply in tabs of a spreadsheet rather than in the form of a report.
K	OK, but natural frequencies listed in rad/s instead of Hz. Also, node numbering problem in the printed mode shapes.	Program will not run this analysis.	OK, except Test Case 5 harmonic response plots are incorrect.
L	OK	OK	OK
M	OK, but natural frequencies listed in rad/s instead of Hz, and the printed mode shapes are not normalized according to the instructions.	OK, except the user has to specify the frequency range in rad/s, and phase angles are reported in radians, instead of degrees.	OK
N	OK, but natural frequencies listed in rad/s instead of Hz; printed mode shapes not normalized according to instructions.	Program will not run this analysis.	Only submitted MATLAB figures, not an actual report. Test Case 5 not included.

(This table is continued on the following page.)

Table 1 - continued:

Student	Modal Analysis	Harmonic Response Analysis	Report
O	OK	OK, except the user has to specify the frequency range in rad/s, while the instructions specify Hz.	OK, except Test case 5 results are incorrect.
P	OK	OK	OK, but mode shape plots apparently included RZ DOF but should only include UY. Test Case 5 harmonic response plots are incorrect.
Q	OK, but natural frequencies listed in rad/s instead of Hz. Also, there is a node numbering problem in the printed mode shapes.	OK, except the user has to specify the freq. range in rad/s, while instructions specify Hz. Phase angles are reported in radians, instead of degrees, as specified in the instructions, and sign is wrong on the phase angles.	OK
R	Program will not run analysis.	Program will not run analysis.	OK
S	Program redefines input values at start; user input is ignored.	Program redefines input values at start; user input is ignored.	Not submitted
T	OK, but natural frequencies listed in rad/s instead of Hz.	OK, except phase angles are reported in radians, instead of degrees, as specified in the instructions, and sign is wrong on the phase angles.	OK, except problems with plots in Test Case 5.
U	OK, but natural frequencies listed in rad/s instead of Hz. Also, the printed mode shapes are not normalized according to the instructions.	OK, except phase angles are reported in radians, instead of degrees, as specified in the instructions.	OK
V	Program is actually a script with interactive input and cannot take the user input as outlined in the instructions. Script won't run.	Program is actually a script with interactive input and cannot take the user input as outlined in the instructions. Script won't run.	Not submitted
W	OK, but natural frequencies listed in rad/s instead of Hz. Also, node numbering problem in printed mode shapes and printed mode shapes not normalized according to the instructions.	OK, except user has to specify the frequency range in rad/s. Phase angles are reported in radians, and sign is wrong on the phase angles.	OK, except Test case 5 results are incorrect.
X	OK	OK	OK

Student Surveys

In each of the most recent three course offerings, the students were given the opportunity to anonymously submit questionnaires regarding the graduate student project. All 10 graduate students submitted surveys in 2011, all 9 submitted surveys in 2012, but only 2 of 5 submitted surveys in 2013. The questions are provided below. Student responses for Questions 1-11 are summarized in Table 2. In Table 3, student comments in response to Question 12 are listed. Discussion of the results follows the italicized list of questions.

Graduate Student Project Survey

1. For your graduate student project, did your results agree with those expected for the test cases?

a) All of them did.
b) Most of them did, but there were some unresolved discrepancies.
c) Some did and some did not.
d) None of them did. I could not get my program to work correctly.

2. Approximately how many total hours do you believe you worked on the graduate student project?

3. Rate your overall computer programming experience before taking this class:

a) No experience *c) Some experience / Intermediate*
b) Little experience / Beginner *d) Significant experience*

4. Rate your MATLAB programming experience before taking this class:

a) No experience *c) Some experience / Intermediate*
b) Little experience / Beginner *d) Significant experience*

5. Rate your Finite Element Analysis experience before taking this class:

a) No experience *c) Some experience / Intermediate*
b) Little experience / Beginner *d) Significant experience*

6. Did you review the supplemental document related to the project: "me513-gs-project-background-info.pdf"?

a) No *c) Yes, but it was not helpful*
b) No, but I did not need to. *d) Yes, and it was somewhat helpful.*
e) Yes, and it was very helpful.

7. Did you review the supplemental document related to the project: "matlab-notes.pdf"?

a) No *c) Yes, but it was not helpful*
b) No, but I did not need to. *d) Yes, and it was somewhat helpful.*
e) Yes, and it was very helpful.

8. On a scale of 1 (not helpful) to 10 (very helpful) rate the project on whether it was helpful to you in understanding computer-based vibration analysis of structures.

9. On a scale of 1 (not helpful) to 10 (very helpful) rate the project on whether it was helpful to you in understanding vibrations of beams, in general.

10. On a scale of 1(very easy) to 10 (very difficult) rate the project on the level of difficulty.

11. On a scale of 1 (not enough information provided) to 10 (plenty of information provided), provide a rating for whether you felt there was sufficient information provided for you to successfully complete the project.

12. Please provide any additional comments regarding the graduate student project.

From the results in Table 2, based on the responses to Question 1, it appears the large majority of students were confident in their results for the five assigned test cases. The responses for Question 2 indicate that possibly this project should account for more than 7% of a graduate student's grade. If a reasonable amount of work for a graduate student in a course is taken to be about 10 hours per week, then in a 16 week semester, a student would do about 160 hours of work. A project taking 28 hours of a student's time would account for about 17% of the work time required in the course. The responses to the subjective Questions 3, 4, and 5 indicate that the graduate students taking this course typically consider themselves to range from beginners to intermediate programmers, with at least some MATLAB and FEA background before taking the course. Considering the programming effort required in this project, along with the supporting material provided to the students on MATLAB and FEA, the project seems reasonable for students based on their self-reported background.

The responses to subjective Questions 6 and 7 seem to indicate that the large majority of students did review the supporting material on FEA and MATLAB, and found that it was at least somewhat helpful.

The responses to Questions 8 and 9 seem to indicate that overall, the students view this project as a worthwhile learning experience. The responses to Question 10 seem to indicate that the project involves enough challenge for the level of credit they receive, and that they consider it neither too easy nor too difficult. The responses to Question 11 seem to support the conclusions stated above related to Questions 6 and 7, that there is sufficient background information provided to the students.

Regarding the comments in response to Question 12 listed in Table 3, some are quite positive, and some students had some criticisms and complaints, which may be expected when students are asked about assignments they were required to complete. Comments 2, 10, and 13 all suggest an ANSYS-based project as an alternative. That would not meet the instructor's goals for the graduate student project. However, implementation of an undergraduate project using

Table 2: Results of Graduate Student Project Surveys.

Question	2011 Results	2012 Results	2013 Results
1	a) 8 b) 1 c) 1 d) 0	a) 8 b) 1 c) 0 d) 0	a) 1 Left Blank: 1

2	Ave=28.2 Hours
	Min=8 Hours
	Max=40 Hours (reported 40+)
		Ave=20.75 Hours
		Min=10 Hours
		Max=40 Hours
			Ave=28.25 Hours
			Min=12.5 Hours
			Max=44 Hours

3	a) 1 b) 4 c) 5 d) 0	a) 1 b) 3 c) 4 d) 1	a) 0 b) 0 c) 1 d) 1
4	a) 2 b) 2 c) 5 d) 1	a) 2 b) 2 c) 5 d) 1	a) 1 b) 0 c) 0 d) 1
5	a) 0 b) 5 c) 4 d) 1	a) 1 b) 4 c) 3 d) 1	a) 0 b) 1 c) 0 d) 1
6	a) 0 b) 0 c) 1 d) 5 e) 4	a) 0 b) 1 c) 0 d) 5 e) 3	a) 0 b) 0 c) 0 d) 1 e) 1
7	a) 1 b) 1 c) 2 d) 4 e) 2	a) 0 b) 3 c) 0 d) 4 e) 2	a) 0 b) 0 c) 0 d) 1 e) 1
8	Ave=7.80	Ave=7.44	Ave=8.00
9	Ave=8.10	Ave=7.56	Ave=8.00
10	Ave=7.40	Ave=6.67	Ave=5.00
11	Ave=9.00	Ave=8.56	Ave=9.00

Table 3: Student comments in response to Question 12 of the Graduate Student Project Surveys.

No.	Year	Student Comments
1	2011	I have no additional comments other than to say I really enjoyed working on the graduate project and it was my favorite part of the class.
2	2011	Doing this project in ANSYS would take much less time and be more applicable to real world engineering. No one is going to make a FEM code every time they need to do a model if a company already has an OK GUI to do it in.
3	2011	The project was not so hard, just took a long time. Plenty of information was provided for the project by the end of the semester but need more on beam analysis earlier to finish on time (original due date).
4	2011	The project is really good. It helps me to understand the FEM better. And Dr. Baker is a nice and thoughtful teacher, since he always gives us useful and clear information.
5	2012	Most of the project went quickly, but I ran into a problem getting a couple of my numbers to match which took me a while to figure out.
6	2012	Only thing I did not like was that the program had to be altered to run the test cases (i.e. add plots)
7	2012	It was difficult to find what you wanted for each test case. I would have liked to see a bulleted list, summary, etc., so that I wouldn't have needed to dig through paragraphs.
8	2012	It's good.
9	2012	It was very tedious due to inexperience with MATLAB, but I learned a lot. I don't know if I will use the knowledge again, but it was a good project. VERY LONG though.
10	2012	Most of the stuff required for the project had already been done in the HW's. It was more a matter of MATLAB debugging in my case, at least, making the different programs from the HW play nicely together. I think maybe an ANSYS project varying beam geometry would be a good alternative. I probably wouldn't have spent as much time on the project if I didn't have MATLAB on my work PC.
11	2012	It was nice to get experience with finite elements and the first part of the project (modal analysis) was very straight-forward. But I ended up spending a lot more time on it than I planned. Also, grads did a considerable amount of homework that undergrads didn't as well.
12	2013	I understand that it is difficult to provide such a lot of information including explanatory videos, different softwares , .. that you did in vibration course. I learnt a lot in your class and I thank you for your great class.
13	2013	I think you can also give some student alternative choice to complete "Test Case". For instance, you can let some students who lack experience of using Matlab by using "Ansys workbench" to do simulation and plot the diagram as you demand. I use Matlab a lot, but some of students may lack the experience of using Matlab, but good at using Ansys to do simulation.

ANSYS is an idea under consideration. Comment 3 was probably a valid criticism for 2011, and in 2012 and 2013, course lecture information was rearranged to address this student's concern. In Comment 6, the student did not like having to alter the program for the test cases to generate plots. This was not a requirement. The submitted plots could be done outside the program the students were to submit. Comment 7, provided in 2012, is probably a legitimate criticism. It probably should have been addressed in 2013, but was not. The format of the assignment sheet will be reviewed before the project is assigned again.

Possible Project Modifications

In the next offering, it is likely that the graduate student project may account for more than 7% of the overall student grades. This would possibly provide more motivation for students to expend a higher level of effort. Also, based on the three years of student survey results, considering the average hours spent on the project, a higher proportion of credit is probably justified if the same project is assigned.

It may be worthwhile to also consider some added complexity in the project to better demonstrate for the students the flexibility and wide applicability of the finite element method. For example, possibly the students should be expected to account for applied boundary conditions along the beam and not only at the ends. Other possibilities are requiring them to incorporate the capability of adding lumped masses along the beam, or including rod elements perpendicular to the beam as flexible supports. A test case for which an analytical solution is not readily available might then be included in their required reported results. However, the primary focus of the course is not on finite element analysis, so it is not desirable to significantly increase the level of complexity of this particular project.

Other FEA Assignments

There is some variation from one semester to the next regarding homework assignments. However, the undergraduates get some background in vibration analysis through finite element methods. The following is a brief summary of homework questions posed to both undergraduate and graduate students related to FEA in fall, 2013. The problems typically involved use of MATLAB or ANSYS.

In one homework problem, the students were required to use finite elements to model a steel bar, fixed at the one end and free at the other, and perform a finite element mesh convergence study in which they calculated undamped natural frequencies for axial motion vibration. For a three element case, they were required to show their final global system mass and stiffness matrices after application of the fixed boundary condition. Undergraduates were only required to handle cases of 1, 2, and 3 element models. Graduate students were required to handle 1, 2, 3, 5, 10, and 20 element models, requiring some programming to assemble the matrices. However, all students had access to a MATLAB script provided by the instructor that, although not suited exactly to this assignment, could be readily adapted to this assignment. Undergraduates could choose to complete the graduate student portion for bonus homework points.

In another problem, the students were required to use finite elements to model a rectangular cross-section simply supported aluminum beam. For a two element case, they were required to show their final global system mass and stiffness matrices after application of the boundary conditions. Undergraduates were only required to handle cases of 1, 2, and 3 element models. Graduate students were required to handle 1, 2, 3, 6, 12, and 20 element models, requiring some programming to assemble the matrices. However, again, all students had access to a MATLAB script that was not suited exactly to this assignment, but could be readily adapted to this assignment. Undergraduates could choose to complete the graduate student portion for bonus homework points. In another beam FEA homework problem, students calculated steady-state response for a damped two element beam finite element model with a lateral sinusoidal forcing function.

In other homework problems, the students used the ANSYS/Workbench FEA software to perform modal analysis and harmonic response analysis for a beam modeled in other homework using FEA in MATLAB. A video with step-by-step instructions on modeling the beam in ANSYS/Workbench was created by the instructor and made available to the students on Blackboard, as there is no course prerequisite that ensures students have experience with this software.

Another ANSYS/Workbench homework problem involved a simple 3D solid model of a compressor blade and illustrated the effects of stresses on natural frequencies for plate-like structures. The stresses were due to centrifugal forces produced by an imposed rotational velocity of the blade. This stress-stiffening effect is difficult to illustrate for students, so the software solutions seem helpful in discussing this important topic.

Partial differential equation solutions for simple structures such as rods and beams are covered after finite element methods in the course. In some assignments, partial differential equation solutions were compared to finite element analysis results from earlier assignments.

Summary and Conclusions

Use of FEA in a mechanical vibrations course has been overviewed. The primary focus of the paper has been the description of a MATLAB programming project assigned to graduate students in the course for the past four course offerings. An assessment of the project has been provided based on the past three times the course has been offered. The assessment of the graduate student project was based on running uniform tests of all graduate student programs submitted over the past three years and review of student reports. Assessment was also based on student responses to questions on a survey.

Based on testing of the student programs, it appears that most students can successfully create a flexible program to perform the required tasks. The fact that some student programs fail in solving the assessment test cases does not necessarily indicate that the student performance is poor, or that the project is not reasonable. It could be viewed as an indication that the level of difficulty is reasonable. If all student programs passed the assessment tests, that may indicate the project is too easy.

Student responses on the survey indicate that they feel they are provided enough background material, that the level of difficulty is reasonable for their FEA and programming background, and that, in general, they consider the project to provide a good learning experience.

Based on the students' reported number of hours required to complete the project, possibly the amount of credit for the project should be increased above 7% of their total grade. Or else, the project requirements should be pared down somewhat. One issue of concern about the relatively low proportion of the course grade is that perhaps some students may have their motivation to successfully complete the project dampened. They have other obligations on which to utilize their time. The fact that a lower grade on this assignment may not have a significant impact on their final course grades could result in reduced effort expended on the project.

This paper has also briefly summarized some additional FEA homework incorporated in the class, some of which involved MATLAB, and some of which involved ANSYS/Workbench.

References

[1]ANSYS®, Copyright 2013, SAS IP, Inc., http://www.ansys.com

[2]MATLAB®, Copyright 1984-2013, The MathWorks, Inc., http://www.mathworks.com/

[3]Thomson, W., and Dahleh, M., *Theory of Vibration with Applications, Fifth Edition*, Prentice-Hall, Upper Saddle River, NJ (1998).

[4]Meirovitch, L., *Elements of Vibration Analysis*, McGraw-Hill, Inc., New York, NY (1986).

[5]Inman, D., *Engineering Vibration, Fourth Edition*, Pearson Education, Inc., Upper Saddle River, NJ (2014).

[6]Balachandran, B., and Magrab, E., *Vibrations, 2nd Edition*, Cengage Learning, Toronto, ON, Canada (2009).

[7]Tongue, B., *Principles of Vibration, Second Edition*, Oxford University Press, Inc., New York, NY (2002).

[8] Hagigat, C. "Using Finite Element Software for Enhancing A Vibration Analysis Curriculum", Proceedings of the 2004 ASEE Annual Conference & Exposition, American Society for Engineering Education (2004).

[9] Baker, J.R., "ANSYS Macros for Illustrating Concepts in Mechanical Engineering Courses", Computers in Education Journal, Vol. XVI, No. 2, pp. 6-21 (2006)

[10] Baker, J.R., "Effective Use of Experimentation and Finite Element Analysis in a Vibrations Course", Proceedings of the 2005 Annual Meeting of the American Society for Engineering Education, American Society for Engineering Education, June 2005, Portland, OR.

[11] Jacquot, R., Anderson, J., and Walrath, D. "Damped Beams: A Versatile MATLAB Script for the Animation of a Variety of Beam Vibration Problems", Proceedings of the 2009 ASEE Annual Conference & Exposition, American Society for Engineering Education (2009).

[12] Balachandran, B. and Magrab, E.B., *Vibrations*, Brooks/Cole, Belmont, CA (2004)

Appendix: Graduate Student Project Assignment Sheet – Fall, 2013

Overview

The finite element method is a widely-used computer-based method for vibration analysis. Your project is to write a program using the MATLAB m-file capability that will perform vibration analysis based on a two-dimensional beam finite element model. Your program will allow for various geometric properties and boundary conditions. However, the programming effort is not too complex because of limitations discussed below on the types of boundary conditions for which your program must allow. This project is intended to familiarize you with how vibration behavior of structures is estimated using the finite element method, and to also familiarize you with some aspects of beam vibration through test cases you will perform to verify your program.

There is a separate document available to you named "me513-beam-fem.pdf" which provides the necessary background on the finite element method applied to the two dimensional modeling beams. You should review this document before reading the project instructions below. Finite element modeling using 2D beam elements is also discussed in the course notes and covered in class meetings. There is also another separate document named "matlab-notes.pdf" which includes some needed background on MATLAB. While that document is somewhat general and includes information not relevant to this project, the section in it on MATLAB m-files may be helpful. Some examples in the document illustrate use of loops and "if-else" sequences in MATLAB that you may need if you have not used these features before in MATLAB.

Project Instructions

Your m-file, which contains your beam finite element program, must be named: "xxxxbeam.m", where "xxxx" are the first four letters of your last name. The program must be able to calculate:

a) Natural frequencies and mode shapes (modal analysis). Damping will not be included in this type of analysis.

b) Steady-state response to a sinusoidal force or moment applied at some node along the beam over a user-defined range of excitation frequencies at user-defined frequency increments (harmonic response analysis). You need to allow for forces and/or moments input at all unconstrained DOF in the model. The user will define the excitation amplitude and specify damping in the model. The only possibility you must allow for in the specification of damping is that the damping matrix, [C], is calculated according to: [C] = α [M] + β [K], where [M] is the global system mass matrix, and [K] is the global system damping matrix. The user will specify values for α and β.

The possible boundary conditions for which your program must allow are:

1. Beam clamped at both ends.
2. Beam pinned at both ends (simply supported).
3. Beam clamped at one end and free at the other (cantilevered).
4. Beam clamped at one end and pinned at the other.

At a clamped end, both the lateral displacement (UY) and the rotation (RZ) are constrained to zero. At a pinned end, only UY is fixed to zero.

The program only needs to allow for a maximum of 100 elements (101 nodes), and it only needs to allow for uniform element lengths. Also, it only needs to handle cases with uniform cross-section and material properties along the beam. These limitations significantly reduce the programming effort required.

The program (m-file) must be set up to be executed by placing the m-file in the user's MATLAB working directory, or "Current Folder", and then typing, at the MATLAB prompt:

```
[res1,res2]=xxxxbeam(ui1,ui2,ui3)
```

The user must first define the user input (ui1, ui2, ui3) before executing your program, and the program returns results in the matrices "res1" and "res2".

The user input will be defined as follows:

ui1: This is a matrix with two rows and four columns, so there are four entries in each row.

Row 1 of ui1: The first entry is the type of analysis. It should be set to "1" for a modal analysis, and it should be set to "2" for a harmonic response analysis. The second entry is the total beam length, L. The third entry is the number of beam finite elements in the model. The fourth entry is a number that specifies the boundary conditions:

Fourth entry is "1" for a beam clamped at both ends.
Fourth entry is "2" for a beam pinned at both ends (simply supported).
Fourth entry is "3" for a beam clamped at one end and free at the other (cantilevered).
Fourth entry is "4" for a beam clamped at one end and pinned at the other.

Row 2 of ui1: The first entry is the beam cross-sectional area, A. The second entry is the area moment of inertia, I, for the beam cross-section. The third entry is the material density, ρ (mass per unit volume). The fourth entry is Young's modulus, E.

It will be assumed that the model is set up so that, for discussion purposes, "node 1" is at the "left end", which is at x=0. For a beam clamped at one end (cantilevered), it will be the end where x=0 that is clamped. For a beam clamped at one end and pinned at the other, it will be the end where x=0 that is clamped and the end where x=L that is pinned. This only matters in that the printed mode shape will be assumed to have the left end (x=0) DOF amplitudes listed first, and the right end (x=L) DOF amplitudes listed last, so it will be helpful, when your program is checked, if all students use the same convention for the boundary conditions.

ui2: This is an input row vector that is used only for harmonic response analysis. It does not need to be input for a modal analysis. This vector has 6 entries:

Entry 1 of ui2: Minimum excitation frequency for the analysis (in Hz)
Entry 2 of ui2: Maximum excitation frequency for the analysis (in Hz)
Entry 3 of ui2: Frequency increment (in Hz).
Entry 4 of ui2: The value of α, the mass matrix multiplier, used to define damping in the model.
Entry 5 of ui2: The value of β, the stiffness matrix multiplier, used to define damping in the model.
Entry 6 of ui2: The number of loads on the beam (forces and moments).

ui3: This is an input matrix that has m rows and three columns, where m is the number of sinusoidal loads (forces and moments) acting on the beam (Entry 6 of ui2). All loads are at the same frequency and are all in phase with one another. In each row:

- The first column is the node number of a sinusoidal input load.
- The second column is a "1" if the input is an applied lateral force (UY direction), or it's a "2" if the input is an applied moment about the Z-direction.
- The third column is the magnitude of the force or moment.

Care should be taken in defining the node number for the load. It will be assumed that the nodes in the model are numbered sequentially from 1 to k, where k is the highest node number. The nodes are equally spaced. And, for example, in a 5 element model of length L, node 1 is assumed to be at x=0, node 2 is at x=0.2L, node 3 is at x=0.4L, node 4 is at x=0.6L, node 5 is at x=0.8L, and node 6 is at x=L.

The following information will be calculated by the program and returned to the user in the MATLAB session in the results matrices, "res1" and "res2".

Modal Analysis Results:

res1: This is a matrix with two columns. The first column contains a mode number and the second column contains the natural frequency for that mode, in units of Hz. Use the convention that mode 1 has the lowest natural frequency, mode 2 has the next to lowest, etc. The number of rows equals the number of unconstrained degrees of freedom in the model. Your program will report all of the calculated natural frequencies, although accuracy may be diminished in the higher numbered modes with your finite element model.

res2: This is a matrix that contains the mode shape vectors for the modes. At the appropriate locations in the vector, a value of zero should be inserted into the mode shape vectors for each constrained degree of freedom. So, if the model contains 5 nodes and 8 unconstrained DOF, then there are 2 imposed constraints. But, each mode shape vector stored in "res2" should contain 10 entries, which is 2 entries per node: one for the UY relative amplitude and one for the RZ relative amplitude at that node. The first entry in a stored mode shape vector pertains to node 1, UY, the second entry pertains to node 1, RZ, the third entry pertains to node 2, UY, etc. The mode shape vectors calculated by the MATLAB "eig" command will need to be modified to insert zeros for constrained DOF at the appropriate locations. Also, the mode shapes should be normalized so that the largest absolute value for any entry is one. The first column of "res2" should be the mode 1 mode shape vector, the second column should be the mode 2 mode shape vector, etc.

Harmonic Response Analysis Results:

res1: This is a matrix with the same number of rows as excitation frequencies considered in the analysis, as defined in the user input "ui2". It has "2n+1" columns, where n is the number of nodes the model. The first column will list the excitation frequencies, in units of Hz. The second column will be the steady-state response amplitude as a function of excitation frequency for DOF#1, which must be the node 1 UY degree of freedom. The third column will be the steady-state response amplitude as a function of excitation frequency for DOF#2, which must be the node 1 RZ degree of freedom. The fourth column will be the steady-state response amplitude as a function of excitation frequency for DOF#3, which must be the node 2 UY degree of freedom. The remaining columns show response amplitudes for the remaining DOF in the model in order. Columns should be included for DOF that are constrained to zero displacement. In these columns, enter all zeroes.

res2: This will be a matrix set up the same as res1, except Columns 2 thru "2n+1" will contain the steady-state response phase angles, in units of degrees. Again, print a zero for the phase angle at a constrained degree of freedom.

Program Verification

In order to verify that your program works, you will need to perform test cases as described below. Results for the beam natural frequencies from your finite element model can be compared to results from a partial differential equation solution based on beam theory. From the textbook[3] by W.T. Thomson and M.D. Dahleh, Theory of Vibration with Applications, 5th Edition, Prentice-Hall Inc., 1998, the following solution is available for beam natural frequencies:

$$\omega_n = F_n \sqrt{\frac{E\,I}{\rho\,A L^4}} \qquad \text{rad/s}$$

Where E, I, ρ, A, and L have the same meanings as earlier in this document. The equation has been altered slightly in order to conform to the notation in this document. In the above equation, the factor, F_n, depends on the boundary conditions. The textbook provides values for n=1,2,3. This allows for calculation of the first three natural frequencies: ω_1, ω_2, and ω_3. The values of F_n for four sets of boundary conditions are as follows:

Beam pinned at both ends (simply supported): F_1=9.87, F_2=39.5, F_3=88.9
Beam clamped at one end (cantilevered): F_1=3.52, F_2=22.0, F_3=61.7
Beam clamped at both ends: F_1=22.4, F_2=61.7, F_3=121.0
Beam clamped at one end and pinned at the other: F_1=15.4, F_2=50.0, F_3=104.0

Also, mode shapes of beams include "nodes" or "node points". In this context, "nodes" has a different meaning from the "nodes" of a finite element model. The node points in a mode shape of a beam are locations where the UY amplitude is zero. The following results for the node points for the beam configurations above are provided in the textbook[12] by B. Balachandran and E.B. Magrab, Vibrations, Brooks/Cole, 2004:

Beam pinned at both ends (simply supported), node points for the modes are:

Mode 1: x=0, L
Mode 2: x=0, L/2, L
Mode 3: x=0, L/3, 2L/3, L

Beam clamped at one end, at x=0 (cantilevered), node points for the modes are:

Mode 1: x=0
Mode 2: x=0, 0.783L
Mode 3: x=0, 0.504L, 0.868L

Beam clamped at both ends, node points for the modes are:

Mode 1: x=0, L
Mode 2: x=0, 0.5L, L
Mode 3: x=0, 0.358L, 0.642L, L

Beam clamped at one end and pinned at the other, the node points for the modes are:

Mode 1: x=0, L
Mode 2: x=0, 0.557L, L
Mode 3: x=0, 0.386L, 0.692L, L

Perform the following test cases with your program. Select realistic beam properties (length, modulus, etc.), and use the same beam properties for all cases.

Test Case 1: Model a beam clamped at both ends. Compare natural frequency results from your program to those from the equation provided above in a "mesh convergence" study. Model the beam using your program in at least 11 different analyses, using at least the following numbers of uniform length beam elements: 2,3,4,5,6,8,10,15,20,50,100. Create a table showing the natural frequencies calculated for each of the first three modes for each number of elements. Note that you will only be able to calculate two natural frequencies for the 2 element case, as there will only be 2 unconstrained DOF.

From the mode shape printouts for the first three modes for the 100 element model, estimate the "node points" from your model (again, here "node points" this means x-locations where the relative UY amplitude is zero), which may require some linear interpolation between your finite element nodes for some modes. Compare the node points from your model to the appropriate results provided above.

Generate a mode shape plot for each of the first three modes from your calculated mode shapes. Use only the entries pertaining to UY DOF. Plot relative UY amplitude as a function of x-location, with x location plotted along the horizontal axis. For the plots, re-normalize the mode shapes so that the maximum absolute value of any amplitude is one third the length of the beam.

Test Case 2: Model a beam clamped at one end, at x=0, with 100 uniform length beam elements. Compare the first three natural frequencies from your model to those calculated from the equation above. Also, from the mode shape printouts, estimate the "node points" from your model for the first three modes. Again, this may require linear interpolation between your finite element nodes for some modes. Compare the node points from your model to the appropriate results provided above.

Test Case 3: Repeat Test Case 2 for the case of a beam pinned at both ends

Test Case 4: Repeat Test Case 2 for the case of a beam clamped at one end and pinned at the other.

Test Case 5: Perform a harmonic response analysis on your 100 element beam model from Test Case 3. Assume a sinusoidal force acts lateral to the beam (UY direction) at x=0.5L. So, the force is applied at node 51 in the 101 node model. Assume very light damping, so you might use a small value for β and a value of zero for α in your input. For an excitation frequency equal to the beam's first natural frequency, calculate the steady-state response. Select β small enough so that the phase angles of the steady-state responses for all UY DOF are all very nearly the same for excitation at the beam's first natural frequency. For the UY DOF only, plot response amplitude as a function of x location along the beam.

Specify the force amplitude so that the maximum UY response amplitude calculated is about one-third the length of the beam. Similar to the mode shape plots, plot x-location along the horizontal axis. While this is not a practical level of response for a real beam, this force amplitude is selected so that the response amplitude plot generated here can be compared to the mode 1 mode shape plot. Note that this is a linear model, so the responses simply scale up or down linearly with the selected force amplitude.

Repeat the procedure just described for modes 2 and 3. However, for the mode 2 case, apply the force at node 76. You should not be able to excite response in mode 2 with a force at x=L/2 because x=L/2 is a "node point" for mode 2 for this beam configuration. For the mode 3 case, place the force at node 51. In each case, compare the "Amplitude vs. x-location" plot to the appropriate mode shape plot. For light damping, at each UY degree of freedom, the phase angles should all be very nearly equal to either one of two values, and the two values should differ by about 180 degrees. We can refer to these two possible angles as φ_a and φ_b, and if these two angles do not differ by about 180 degrees (π radians), the damping should be reduced. For any UY response that has a phase angle of approximately φ_a, plot the response amplitude as a positive value. For any UY response that has a phase angle of approximately φ_b, plot the response amplitude as a negative value. For light damping, the response at a specific mode's frequency should be in approximately the same deformation pattern as that indicated by the mode shape for a given mode. Of course, the two plots may be "flipped" about the horizontal axis with respect to one-another.

In the testing of your program, keep in mind that the finite element method is an approximate method. The results typically improve as more elements are used. If you don't get exact agreement of results with the equations provided, it doesn't necessarily mean your program is wrong. But you should get very good agreement, especially for the 100 element cases. If you do not, you may have a programming error.

Project Deliverables

You will need to submit your m-file, "xxxxbeam.m", as an email attachment to the instructor, along with a report summarizing your test case results. The report should include appropriate plots and descriptive captions, axis labels, and plot titles. The project is due by 5 pm eastern time on Friday, December 12.

Your program will be tested, and the input for your program must conform to the instructions provided above.