ASEN 2001 Lab 1: Computer Analysis of Structures – Fall 2018

1. Summary

This lab is concerned with the development of a MATLAB code for analyzing the forces and moments acting on a 3-D structure. For a given set of external forces and moments as well as support conditions you will develop a MATLAB code that analyzes the support configuration and computes the reaction forces and moments. This lab provides a first opportunity to refine your program development skills and to translate an analysis method into a computer program. Particular emphasis is given to the ability to formulate, build and analyze the static equilibrium equations. The knowledge and skills reinforced through this lab are essential elements for several upcoming labs in ASEN 2001 and other sophomore and junior level courses.

2. Logistics

Group assignments will be provided by your lab section's TA at the beginning of your sections. The following schedule is meant to provide some guidelines regarding the expected progress. Note: Although this is a rather short lab making continuous progress is important.

Monday	**Tuesday**	**Thursday**
September 10 **Lab 1 starts;** development of program structure and I/O routines	September 11 Lecture	September 13 Lecture
September 17 Lab 1 continues; Formulating, analyzing & solving equilibrium equations	September 18 Lecture	September 20 **Exam 1**
September 24 Lab 1 continues; Code verification, lab report	September 25 Lecture	September 27 Lecture
October 1 **Lab 2 starts**	October 2 Lecture	October 4 Lecture

This lab involves theoretical work and code development. As no experimental work is involved the lab groups will be rather small. All team members need to contribute to this lab.

The deliverable of this lab is (a) a group report (see Section 6) and (b) the MATLAB code developed together with input files for the verification examples (see Section 5). Both, the report and the code + verification example input files need to be uploaded to Canvas by September 28, 7:00 am. You are strongly encouraged to upload the deliverables earlier such that you can accommodate Canvas downtimes and other unforeseen issues. **No late submission will be accepted**.

3. Static Analysis

The goal of this lab is to develop a MATLAB code that computes the reaction forces at the support of 3-D structures for given external loads and moments. A typical problem setting is shown in Figure 1.

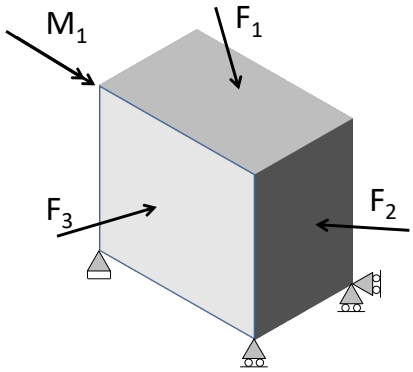

Figure 1: 3-D structure with external loads and moments and support conditions

Assume that an arbitrary number of external point loads $\mathbf{F}_i$ and moments $\mathbf{M}_i$ act on a 3-D structure. The magnitude and direction of the forces and moments as well as the location where they act on are known and should be treated as input to the code. The code should also allow for inputting the location and type of point supports. A support type is defined by the direction of the forces or moments a support resists.

For a given configuration of external loads and moments and support locations and types, the code should compute the magnitude of the reaction forces and moments at the support. This is done by formulating and solving the static equilibrium equations. For statically determinate 3-D structures, the number of equilibrium equations and the number of unknown support reactions is six. The code should check at appropriate steps in the analysis procedure that both conditions are satisfied. The correct number of unknown support reactions can be checked after reading the problem setup. The correct number of *linearly independent* equilibrium equations can be checked after building the equilibrium equations and before solving the set of linear equilibrium equations. Only if the problem is setup properly, the problem has a unique, mechanically meaningful solution and the reaction forces and moments can be determined. The code should output the problem setup and the support reactions.

4. Code structure

At the very minimum the code should consist of the following functions:

1. Main function that controls the overall execution of the analysis procedure.
2. Input routine that reads the input from an ASCII text file.
3. Routine for computing support reactions.
4. Output routine that writes problem setup as read from the input file and calculated support reactions to ASCII output file.

You are strongly encouraged to use functions to modularize your code, as well as other programming techniques to reduce the number of lines of your code, such as for-loops etc.

To run a particular problem, the main function should be called with the name of the input file as parameter, e.g.:

```
truss3d('inputfile_name')
```

where truss3d is the name of the file containing the main function and inputfile_name is the name of the input file. The code should not have to be modified to run a particular problem.

Your code needs to follow the requirements for documentation and commenting as defined in ASEN2012. In particular, define all input and output parameters of a function / subroutine. One should be able to understand your code based on the comments in the code.

4.1. Input format

The problem setup should be specified via an ASCII input file using the following format (note the data provide here does not refer to a particular problem; the numbers are chosen arbitrary):

```
# number of external forces and moments
2  1
# coordinates of the points at which external forces are applied
# x    y    z
0.0  1.0  1.0
5.0  3.0  0.0
# magnitude and direction of external forces
# F   dx   dy    dz
11.0  3.0  2.0  -9.0
0.1  3.0  1.0   1.0
# location at which external couple moments are applied
# x    y    z
0.0  1.0  1.0
# magnitude and direction of external couple moments
# M   dx    dy   dz
10.0  4.0  -2.0  9.0
# location of supports
# x    y    z
1.0  1.0  1.0
1.0  1.0  1.0
1.0  1.0  1.0
0.0  1.0  0.0
0.0  1.0  1.0
1.0  1.0  0.0
# type (F/M) and direction of reaction
# type  dx    dy    dz
F      1.0   6.0  -7.0
F      4.0   1.0   1.1
F      1.0   8.0   1.0
F      6.0   1.0   0.0
M      0.0   9.0   1.0
M     -1.0   1.0   0.0
```

where the "#" symbol indicates a comment line.

Note:

- The numbers of lines will differ for different problem setups.
- The support reactions are defined via the direction of the reaction force (F) or moment (M).
- There will always be a total of 6 support reactions that need to be defined.
- Do not require that the direction information is provided via unit vectors; allow for vectors of arbitrary length.
- The problem defined by the sample input file given above does not necessarily represent a proper problem and should not be used to test your code.

MATLAB provides various functions to process and read information from ASCII files such as fgetl, dlmread and fopen. There are many options for you to research and then decide on which solution you want to use. Refer to the MATLAB documentation.

4.2. Computation of support reactions

The support reactions are computed by solving the static equilibrium equations. For statically determinate structures in 3-D, static equilibrium can be formulated by three linear force and three linear moment equilibrium equations. For a computational treatment, the static equilibrium equations can be conveniently written in matrix form:

$$\underbrace{\mathbf{A}}_{6\times6}\ \underbrace{\mathbf{x}}_{6\times1} = \underbrace{\mathbf{b}}_{6\times1}$$

where **A** is a coefficient matrix, **x** is the vector of unknown support reactions, and **b** is the vector of external forces and moments. Develop and implement in MATLAB a procedure to automatically formulate these equations. After building matrix **A** and vector **b** your code should provide a check that ensures that the system is solvable, that is that matrix **A** contains 6 linearly independent equations. MATLAB provides various functionalities to conveniently perform this check. Explore these options.

MATLAB provides a large number of options for solving systems of linear equations, including the constructions of the inverse of a matrix and the "\" operator. Familiarize yourself with these two options and discuss their advantages and disadvantages.

4.3. Output format

The problem setup and the computed support reactions should be written to an ASCII file. Develop an output format that facilitates reading the problem setup and the results.

Hint: MATLAB provides various options for writing ASCII output. Depending on the desired level of control over the output format you may chose different options. A function that allows for sophisticated output formats is `fprintf`. Also look into other options such as dlmwrite.

5. Code verification

A crucial part of the code development process is so-called code verification which ensures that the code produces the same results one would obtain by correctly solving a problem by hand. Verify your code by solving the following problems in Hibbeler by hand and with your code:

Example 4.11, F4-11, F4-12, Problems 4-28 and 4-34

Note: The setup of the boundary conditions in some of these examples needs to be augmented to obtain the proper number of boundary conditions and to suppress all rigid body modes.

Turn in the input and output files for the above problems (see Section 7). The manual solutions do not have to be turned in.

6. Group Report

This lab focuses on the development of the MATLAB code. Your report should be considered a theory and developer / user manual. The report should include the following elements:

Title Page. (1 page) Title of lab, lab section, team members, date.

Theory Manual. (1 page) Describe the theory and procedure analyzing the reaction forces for a 3-D structure in static equilibrium. Clearly define the theoretical limitations of your code and discuss the assumptions your analysis procedure is based on. Cite references as applicable.

Developer / User Manual. (3 pages) Describe the structure of your code, the functionality of the routines you have developed, and outline the usage of your code including the content and format of the input and output files. Describe clearly how the user can run the verification examples.

The MATLAB source code should not be included in the report; neither within the body of the text nor as appendix.

Format: The report should be written in MS Word using 12pt New Times Roman and 1" page margins. Larger fonts can be used for headings. No other formatting requirements need to be met.

7. Deliverables

The following items should be uploaded to Canvas by the due date. **No late submissions are accepted. No hardcopies are accepted.**

1. Group report (see Section 6)
2. MATLAB source code (.m files)
3. Input and output files of the verification examples (see Section 5).

All files should be combined into one zip file.

Addendum I - Report Grading

The score assigned to the lab report includes technical content (80%) and presentation (20%). This is a more detailed breakdown of the weights:

Category	Weight	Score	Contribution
Title page	0.05		
Theory Manual	0.35		
Code Structure	0.20		
Code Capabilities	0.20		
Organization	0.05		
Flow and style	0.05		
Grammar, spelling and typos	0.05		
Referencing	0.05		
Total	1.00		

"Flow" measures smoothness of reading from start to finish and correlation of material from section to section, as well as adherence to guidelines of technical writing.