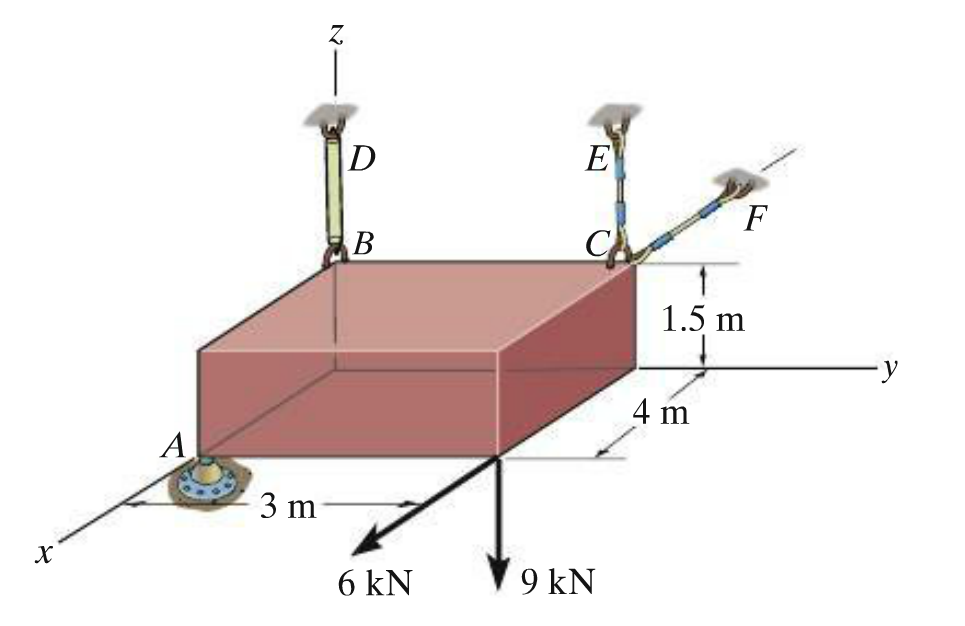
**Computer Analysis of Structures**

ASEN 2001 - 013

26 September 2016

Cody Goldman

Nicholas Renninger



**Theory Manual**

Theory

A rigid-body is an idealized object which does not deform under loads[[1]](#footnote-1). Rigid-bodies in three-dimensional space can be manipulated in six ways: translated or rotated along the three orthogonal axes representing the three Cartesian dimensions. A rigid-body is in equilibrium only if there is no translational or rotational movement. Equilibrium can be achieved by properly supporting an object with support structures that apply reaction forces and moments to balance external forces and moments and keep the object steady[[2]](#footnote-2).

In order to determine the magnitude of these reaction forces and moments, the system must first be assumed to be in equilibrium as described above, and must be assumed to be described by a linearly-independent system of equations, which is a system of equations that has the same number of unknown quantities as linearly independent equations[[3]](#footnote-3). The equilibrium equations simply state that the sum of all moments and forces (both vector quantities) about any point must be equal to the zero vector[[4]](#footnote-4):

Each of the statements above can be resolved into its three Cartesian components, resulting in six equations of equilibrium. With knowledge of the locations, magnitudes, and directions of the external forces; the magnitudes and directions of the external couple moments; the locations of the support structures; and the type and direction of the support reactions, the equilibrium equations can be used to calculate the magnitudes of the support reactions by creating and solving the linear system of equations:

where **A** is the 6x6 coefficient matrix of the 6x1 vector , which contains the magnitudes of the reaction forces and moments, and where is the 6x1 vector containing the external forces and moments summed in x, y, and z.

Procedure

There are many equations that can be used to get the necessary six in order to solve the system of equations, but we choose to calculate the sum of forces in each dimension and the components of the resultant moment about the origin. The first things done are the calculations for the external forces and moments to sum up their components and generate the vector used for the final calculation. Then, the x, y, and z directions of the unit vectors of the support forces are used for calculating the force coefficients in the **A** matrix. The x, y, and z components of the cross products between the positions of the force supports and the direction of the forces are used for the moment coefficients of the **A** matrix. The solution to the linear system can be found by multiplying both sides by the inverse of **A**, , with:

Where is the vector containing the magnitudes for the support reactions.

This procedure only works if the system fits some assumptions. The system must be a rigid-body in equilibrium, with six support reactions, and with all necessary information about external forces and moments known. If there are less than 6 support reactions, then a linearly independent boundary condition must be introduced to the system, e.g. fictitious support reactions must be included in order to maintain a total of six support reactions to make sure the system is a determinant system. This also means that if there are more than six support reactions, the system is *indeterminate* and cannot be solved using equilibrium equations alone[[5]](#footnote-5). This represents a limitation on equilibrium analysis; solving systems with more than six unknown support reactions requires use of an alternate method of analysis with more augmenting equations.

**Developer / User Manual**

Code Structure

The main function, truss3d, calls three other functions: one to read the input file, another to perform the support reaction calculations, and a third to make an output file.

The function lab\_1\_fileRead reads the input file and takes the name of the input text file as an input argument. Eight output vectors are initialized as blank vectors to handle situations with no external forces and couple moments. The function then opens the file and begins recording the data with each line occupying a row in a cell. Since the input files are structured in an organized way with a predictable number of commented lines for separating different types of input data, lines which begin with the pound symbol, #, notify the function to begin inputting the next non-commented line of data as the next type of input. The input order is the same as the order of the input file: the number of forces and moments, the coordinates of external forces, the magnitudes and directions of external forces, the locations of external moments, the magnitudes and directions of external moments, the support locations, the direction of force reactions, and the direction of moment reactions. Each input type is saved in its own output matrix. Input types with no data, as in the case of no external forces or no external moments, return empty matrices.

The support reaction calculation function, calculate\_support\_reactions, takes the matrices returned by the first function to compute the magnitudes of the support reactions necessary for equilibrium. This function also utilizes a useful auxiliary function, lab\_1\_convertToUnitVector, which takes a vector as an argument and returns the equivalent unit vector. This function also includes a check to make sure that the zero vector does not create any division by zero errors. The support reaction calculation function then works to construct two 6x3 matrices used to solve the system of six linear equations. Three of the equations solve for the summation of forces in the three Cartesian dimensions. The other three equations solve for the moments about the origin in the three directions. These are then transposed and vertically concatenated into one 6x6 matrix, called **A**.

The matrix **A** contains the coefficients of the support reactions in each of the equilibrium equations. The vector contains the values of the external forces and moments acting on the object. The first three rows of the vector are simply the summation of the external forces in Cartesian coordinates with the three elements corresponding to the three dimensions. The computation is done one dimension at a time by summing all the external forces’ components before moving to the next dimension. The final three rows are the summation of external moments and moments created by the external forces. The moments created by external forces are computed using the cross product of the vector from the origin to the point at which the force acts, and the force vector itself. The force vector is calculated by multiplying the magnitude of the vector by the normalized direction components. The results of the cross products, in Cartesian coordinates, are summed together with the external moments to create the last three rows of .

Matrix **A** is a 6x6 matrix with the first three rows representing the force sum equations and the final three representing the moment equations. The six columns each correspond to one of the reactions, and the scalar values in each element in the matrix represent that reaction’s linear coefficient for that equation. For the force sum equations, the direction components of the force reactions are normalized to unit length, and those values are used as the coefficients. Moment reaction columns have values of zero for the first three rows. The last three rows use the normalized cross product vectors of the position of the support and the direction of the force support reaction for the force reactions. The normalized directions are used for the coefficients for the moment reactions.

After generating the matrices, a check is done to verify that matrix **A** is invertible. If the test passes, the inverse of matrix **A,** , is multiplied by -to result in a 6x1 column vector which contains the magnitudes of the six support reactions. Otherwise, if the **A** matrix is singular, meaning that it has no inverse and thus no unique solutions, an error message will be displayed. The function finishes, and the vector containing the magnitudes of support reactions is returned along with the number of support reactions that are forces.

The final function produces the output file and takes the name of the input file, the number of support reactions that are forces, and the vector containing the solved support reaction magnitudes as its input parameters. This output file is first created by taking the name of the input file and appending “\_output” to the input filename. The function copies all of the information from the input file into this new output file, and appends another six lines to display the results of the function. Each support reaction is identified as either a force or a moment and for each support reaction the magnitude is listed. The support reactions are printed in the order they were given in the input text file, with each force identified as the 1st - *n*th reaction force, where *n* is the number of reaction forces. Each moment is identified as the 1st - *k*th reaction moment, where *k* is the number of reaction moments.

Code Capabilities

The code was built with very specific problems in mind. Only the magnitudes of support reactions may be unknown, and there must be six or less support reactions. Systems with more than six unknown support reactions are not determinate and *cannot* be computed without additional information. The input file must be structured properly, and there can be no contradictions. Systems with indeterminate solutions such as multiple support reactions which are identical, or insufficient supports to keep the system in equilibrium, will result in errors in the process of calculating support reactions. Done correctly, the code is useful for quickly computing support reactions of known systems with six or less support reactions.

There are five functions used in these calculations. The main function, truss3d, has an input argument of the name of a text file. Do not include the file type extension (e.g. omit ‘.txt’ from the filename used) in the function call. This function does not return anything. When running the function on an input file, make sure the text files are in the correct directory, as they were in the zip file. If the text files are not in the folder ‘text\_files’, the code will not be able to find the input files, and will throw an error. The read function, lab\_1\_fileRead, also uses the file name as an input argument, the argument passed to truss3d. This function returns eight matrices which correspond to the eight different matrices returned by the lab\_1\_fileRead function. The support reaction calculation function, calculate\_support\_reactions, does most of the work to rearrange and solve the matrices read in from the input text file. It takes the return parameters from the function used to read the input file and itself returns a 6x1 column vector containing the magnitudes of the support reactions and also the number of the support reactions that are forces. Within this function is lab\_1\_convertToUnitVector, which takes a row vector of length 3 or 4 and, assuming the last three elements are directional components, converts them to unit length and returns those three elements normed. In the case where the input argument is length 4, typically of the case where the magnitude of the vector was included as the first element of the vector, the magnitude is ignored and the norm of the remaining vector elements is found. The last function, lab\_1\_filewrite, takes the column vector given by the calculation function, the file name originally passed to truss3d, and the number of force support reactions and creates a new file with all of the information of the input file with the results of the analysis appended. This function does not return anything.

Contained in the zip file are multiple input files based on problems from the textbook, Statics and Mechanics of Materials Fifth Edition by R.C. Hibbeler. Calling the function truss3d with the names of these files will solve the problems and write the solution to output text files. The output text files can be found in the same folder as the input files, with “\_output” appended to the original filename to indicate that it contains the program output and problem solution.

1. Hibbeler, R.C. "Equilibirum of a Rigid Body." Statics and Mechanics of Materials. 5th ed. London: Pearson Education, 2014. 158. Print. [↑](#footnote-ref-1)
2. Hibbeler, R.C. "Equilibirum of a Rigid Body." Statics and Mechanics of Materials. 5th ed. London: Pearson Education, 2014. 159. Print. [↑](#footnote-ref-2)
3. "Solving Linear Systems." Wolfram Language & System Documentation Center. Wolfram, n.d. Web. 23 Sept. 2016. [↑](#footnote-ref-3)
4. (same as 1) [↑](#footnote-ref-4)
5. "Energy Methods in Structural Analysis." Indian Institute of Technology (n.d.): 11. Indian Institute of Technology. Web. <http://www.facweb.iitkgp.ernet.in/~baidurya/CE21004/online\_lecture\_notes/m1l1.pdf>. [↑](#footnote-ref-5)