

# CME211 HW4 Truss Analysis

Xuyi Long

November 4, 2022

## 1 Project Summary

This program consists of 2 python files. The `truss.py` file defines a `Truss` class for processing data from beam and joint files from a specified truss folder. The `main.py` file is used for program execution from the terminal. The `main.py` file take 2 required arguments, which are the beam data file and the joint data file, and an optional argument, which saves a plot of beam geometry in the same file location if the user specifies a file name as the third argument.

## 2 Truss Class Decomposition

### 2.1 Initialize Truss Object

The initialization methods takes arguments from the user and invoke the `beamforce` method to read input data files. If a third arguments is provided, the `Plot geometry` class will be invoked to generate a plot of beam structure.

### 2.2 Solve Matrix Equations

After reading input files, two dictionaries are create to store data. One dictionary stores joint numbers as keys and their coordinates, external forces, and rigid connections are stored as values. Another dictionary stores beam number as keys and their corresponding joints as values.

To solve for the force on each beam, we need to create a coefficient matrix `a` and result matrix `b`. The dimensions of matrices are determined by number of unknowns in the system. Given `n` unknowns for the structure system, the coefficient matrix `a` will have a dimension of `n` by `n` and the result matrix `b` of `n` by 1.

The key to solve the matrix equation is to determine coefficient values inside matrix `a` and external force values in matrix `b`. For matrix `a`, each even number row (row 0,2,4...) represents the equilibrium of `x` component of a joint, each odd number row (row 1,3,5...) represents `y` component a joint. In this program matrix `a` is defined as a spare matrix in COO format to store coefficients. By looping through each coordinates inside the `n` by `n` matrix `a`, we extract the coefficient value from the joint and beam dictionaries and associate the value to their corresponding coordinates. The external force values are stored into matrix `b`. In the end, matrix `a` and `b` are converted to CSR form and beam forces are solved by invoking sparse matrix solving method from SciPy.

### 2.3 Truss Geometry Plot

If the user provides a file name as a optional input in terminal, a plot of truss geometry will be created and saved as png file in the same directory. If the optional input is not provided, no plot will be created.

### 2.4 Error Handling

Two error cases are handled in this program. In the case that the truss structure is over or under defined (number of unknowns don't match number of equations) an error message "Truss geometry not suitable for static equilibrium analysis" will be raised. Another case is that when the matrix `a`

is singular (unstable truss geometry), an error message "The linear system is not solvable, unstable truss?" will be raised.

## **2.5 Terminal Output**

A repr method is created in the Truss class. It prints the instance to show calculated force on each beam when main.py file is executed. If no inputs are given, the program outputs usage information.