Department of Computer Engineering
University of Peradeniya

CO328 Software Engineering Project

Finite Element Based Structural Analysis Software

Group 20:
E/14/158 Gihan Jayatilaka
E/14/339 Suren Sritaharan
E/14/379 Harshana Weligampola

Client:
Structures Laboratory,
Department of Civil Engineering,
University of Peradeniya.

Contents

1 Abstract

Finite Element Method (FEM) is widely used for structural analysis in civil engineering projects. FEM algorithms predict the deformations and failures of structures under different loading conditions. These algorithms are computationally intensive. The most popular method to perform nonlinear analysis is force based fiber beam column element. A newer (and better) formulation has been proposed by University of Peradeniya[4]. This algorithm has been experimentally proven to be giving stable and accurate solutions with respect to axial force - bending moment interaction of civil engineering structures.
A computationally efficient implementation of this algorithm with a user friendly interface is crucial in making this theoretical advance usable for practical applications.
This CO328 project tries to implement a usable version of the proposed algorithm and test it against manually calculated structural deformation predictions. The software program will be packaged in a way that it can act as the backend for the "CO328 project - Structural Analysis Project". Furthermore, the software will contain an API to enable it to be used as the building block of algorithms built on this proposed formulation.

2 Introduction

3 Related work

3.1 Related algorithms

3.1.1 Classical methods

These algorithms are usually studied in the first part of the undergraduate curriculum. They can solve simple structures. But they fail to analyze complicated structures.
Method of joints and method of sections cannot be used to analyze the deformations (linear elastic and plastic) or the failure for the structure since they work on the assumption that all elements of the structure are ideal. Elasticity based methods can analyze linear elastic deformations only.

- Method of joints

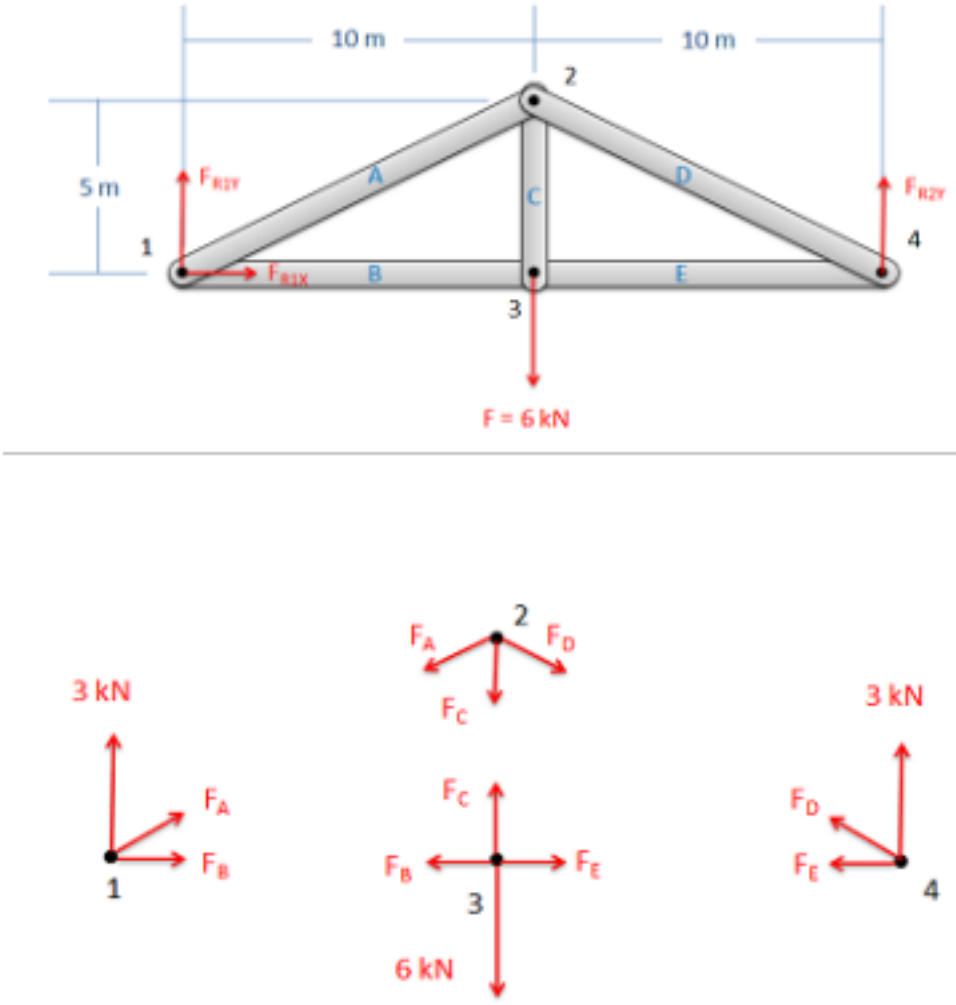

Figure 1: Method of joints

The method of joints is a process used to solve for the unknown forces acting on members of a truss. The method centers on the joints or connection points between the members, and it is usually the fastest and easiest way to solve for all the unknown forces in a truss structure.[2]

- Method of sections

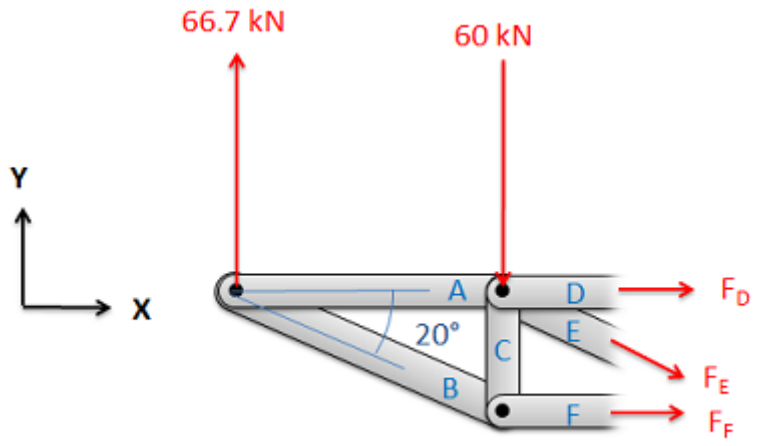

Figure 2: Method of sections

The method of sections is a process used to solve for the unknown forces acting on members of a truss. The method involves breaking the truss down into individual sections and analyzing each section as a separate rigid body. The method of sections is usually the fastest and easiest way to determine the unknown forces acting in a specific member of the truss.[2]

- Elasticity based methods
  Elasticity methods are available generally for an elastic solid of any shape. Individual members such as beams, columns, shafts, plates and shells may be modeled. The solutions are derived from the equations of linear elasticity. The equations of elasticity are a system of 15 partial differential equations. Due to the nature of the mathematics involved, analytical solutions may only be produced for relatively simple geometries.[1]

3.1.2 Numerical methods

It is common practice to use approximate solutions of differential equations as the basis for structural analysis. This is usually done using numerical approximation techniques.

- Finite difference methods
- Finite element methods

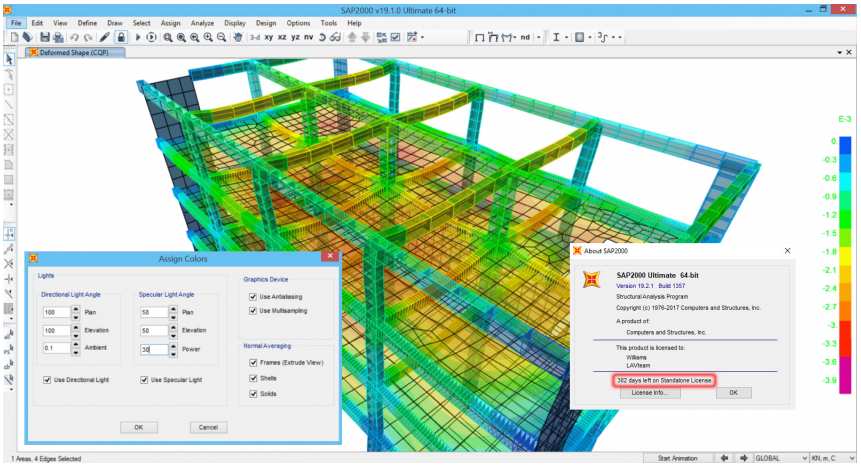

Figure 3: SAP2000 Structural Analysis Software

3.2 Related software packages

3.2.1 SAP 2000

SAP2000 is the industry standard structural analysis software. It has been used on every form of structural design ranging from simple structures for Burj Khalifa.

Pros: Industry standard
Cons: This does not allow non-linear analysis.

3.2.2 S-FRAME Analysis

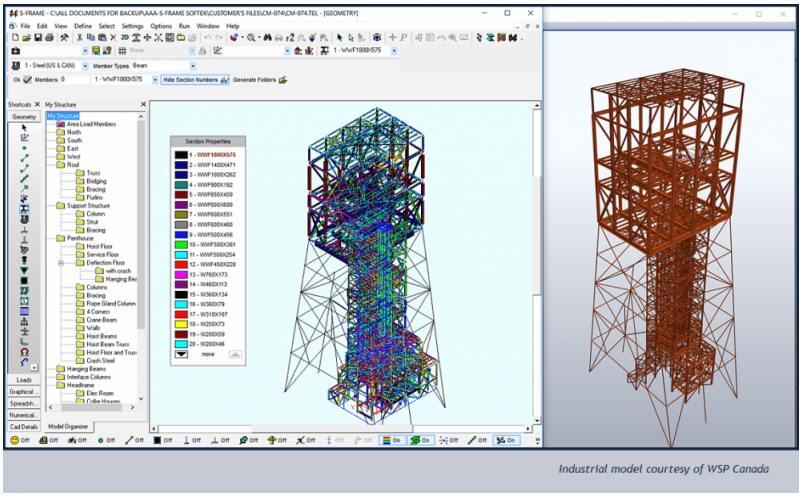

Figure 4: S-FRAME Structural Analysis Software

Pros: Can perform non linear analysis.
Limitations:

- Cannot analyze the bending moment, axial force interaction.
- Expensive

4 Proposed Solution

4.1 Introduction

The project aims to create a software capable of running a structural analysis on a given civil engineering structure. The program will be able to asses the response of the structure against different loading conditions.
The software will use industry standards from SAP2000 for structural element definitions and coordinate systems.[3]

4.2 Formulation

4.2.1 Coordinate system

The global coordinate system is a three-dimensional, right-handed, rectangular coordinate system. The three axes, denoted X, Y, and Z, are mutually perpendicular and satisfy the right-hand rule. The location and orientation of the global system are arbitrary.
Locations in the global coordinate system can be specified using the variables x, y, and z. A vector in the global coordinate system can be specified by giving the locations of two points, a pair of angles, or by specifying a coordinate direction. Coordinate directions are indicated using the values ±X, ±Y, and ±Z. For example, +X defines a vector parallel to and directed along the positive X axis. The sign is required.
All other coordinate systems in the model are defined with respect to the global coordinate system.

4.2.2 Frame element

The Frame element uses a general, three-dimensional, beam-column formulation which includes the effects of biaxial bending, torsion, axial deformation, and biaxial shear deformations. See Bathe and Wilson (1976).
Structures that can be modeled with this element include:

- Three-dimensional frames
- Three-dimensional trusses
- Planar frames
- Planar grillages
- Planar trusses

A Frame element is modeled as a straight line connecting two joints. Each element has its own local coordinate system for defining section properties and loads, and for interpreting output.
Each Frame element may be loaded by self-weight, multiple concentrated loads and multiple distributed loads.

4.2.3 Connectivity

Every frame element is connected at two end nodes. It may be connected to one or more elements at those ends.

4.2.4 Degrees of freedom

Every frame element has 6 degrees of freedom.

4.3 Algorithm

The algorithm consists of the following steps.

1. **Take inputs**
   The inputs are read from the standard format file.
2. **Create local stiffness matrices**
   The stiffness matrices for every element is created using it's properties (length, cross section area, young's modulus). The coordinate system at this stage is the local coordinate system.
   Given that the degrees of freedom per node is 6 and an element has 2 degrees of freedom
   $$K_{local12\times12}$$
3. **Global stiffness matrix**
   The global stiffness matrix is the characteristic function defining the load response of a structure. This is a square matrix of $no\_of\_nodes \times dof\_per\_node$. There are two steps required to generate this matrix.
   - Rotate the local stiffness matrices (according to the orientation of the element)
   - Add the resultant element stiffness matrix elements to the relevant elements in the global stiffness matrix.

   $$K_{global(N\times N)}$$
4. **Generate the external load matrix**
   This is done by placing the external torques and forces in a column matrix.
   $$F_{(N\times1)}$$

5. **Handle fixed points**
   It is already known that the deformation at fixed points should be zero. Therefor all the row and column vectors corresponding to the fixed points in the global stiffness matrix are removed to obtain the reduced stiffness matrix.
6. **Solving**
   Now a linear system $F = K_{red}X$ is obtained where F is the load vector, K_{red} is the reduced global stiffness matrix and X is the deformation vector. The system should be solved for X as

   $$X = F^{-1}K_{red}$$

   This is the most computationally heavy operation of the program.
7. **Backtracking the deformation**
   The solution obtained up to now is for the displacement of nodes in global coordinate system. This is backtracked to the displacement of nodes in the local coordinate systems of elements. That can be traced back to the deformation and the forces acting on the elements.
8. **Output** The output is genearated in a standard format for the GUI program. It is algo dumped to a file on disk.

4.4 Software

The software product of the project is a easy to use shell application that can perform the above mentioned structural analysis. The core functionality will be released on two forms.

- A standalone application that can perform the analysis
- A python package that can be integrated into GUI software (Group 17) or other complex software taking use of this analysis technique.

5 Implementation

Since the algorithm executes in the back-end, it should be written in an efficient and reliable language. The I/O of the back-end was handled by Python. Then, the algorithm was developed in 3 parallel branches that used python, cuda and MATlab as the programming language. The MATlab and python solutions were released early

because they can be implemented fast. Cuda implementation was developed to a similar stage but further development came to a halt due to customer requirement to implement the algorithms using solely on python.

5.1 Design

The complexity lies in the algorithms and the data structures. Thus, the design of the software is simple. The software works as an interface for the front end. So the requirements are well defined and constrained.

In the context of this software, elements are the major classes of each structure. Almost all the elements have similar properties, thus they inherit their attributes from a generic element. The elements differ only in certain attributes known as the "degree of freedom" which varies between each element.

5.2 Technology stack

The software is made using python. The algorithmic implementation was also made in C++, cuda and MATLAB but due to the limited requirement of the project owner, these versions were not released (The development version of these implementations can be found in the development branch of the github repository).

The following packages were used:

- json: Read and write json files which is the agreed format of communication with the front end.
- numpy : Used for mathematical computation
- scipy : Used for efficient linear solving
- unittest : Used for testing
- cuBLAS : Used for basic linear algebraic operations

6 Agile development practices

A major reason for the success of this project was due to the agile development practices which were followed. These practices ensured a smooth development process. Few of the agile development practices which were followed have been explained in detail below.

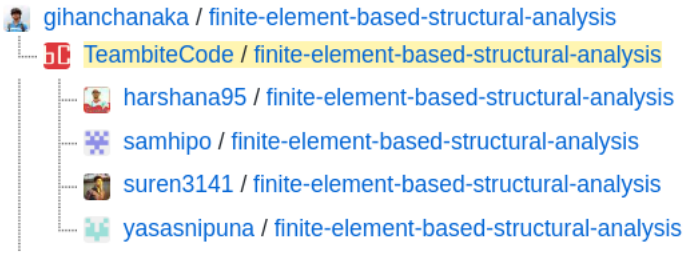

Figure 6: Github repo with both Group 20 and the Client

6.1 Collaboration

The project was done in collaboration with 2 other groups with the final objective of integrating the work of all 3 groups to form a fully working product. The work was divided between the three teams considering the specialization of each groups. As such, the task which was handed over to us was the formulation, design and optimization of the Finite Element based algorithm for Structural analysis.

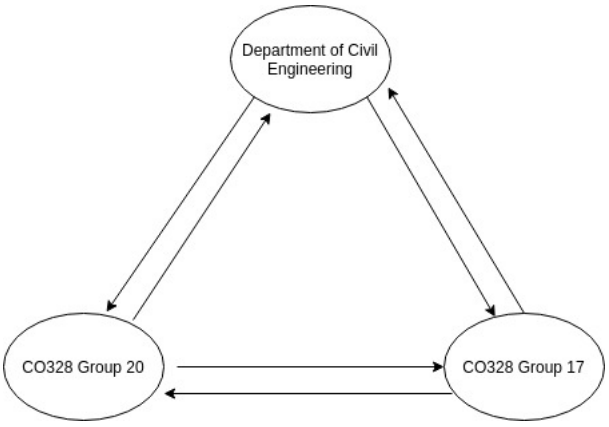

Figure 5: Group interactions

Stakeholder	Description
Department of Civil Engineering	Client
Group 17	GUI development
Group 20	**Backend developemnt**

6.1.1 Collaboration within Group 20

- Informal scrum meetings to plan, discuss.
- An email thread to set goals.
- Github to share code and collaborate work.

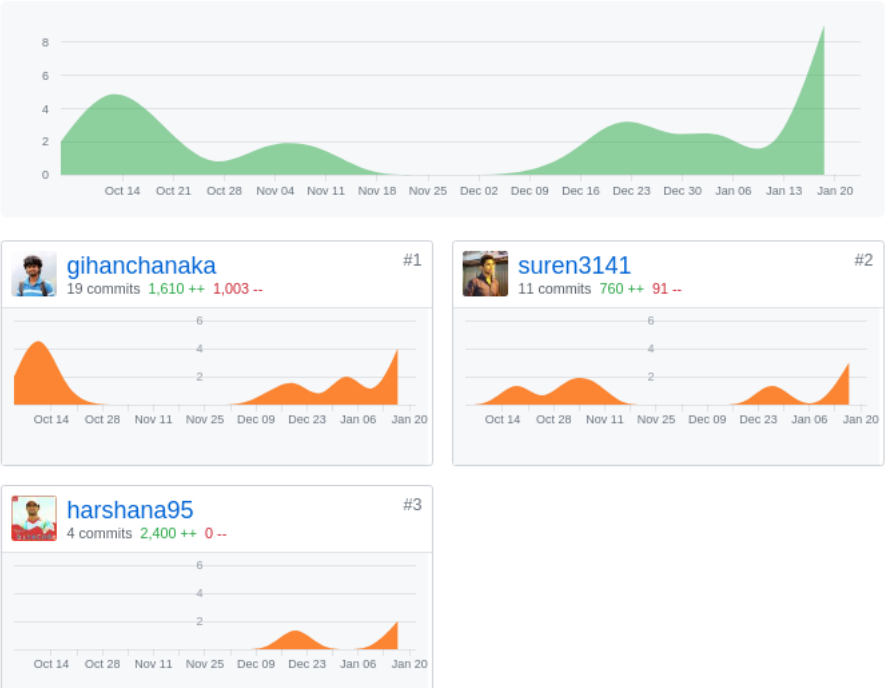

Figure 7: Contribution graph

6.1.2 Collaboration of Group 20 with Civil Engineering Department

- Weekly meetings (for first few weeks) for domain familiarization.
- Bi-weekly meetings (after first few weeks) to discuss the progress, clear doubts.
- Email thread

6.1.3 Collaboration of Group 20 with Group 17

A common interface (data saving format) had to be agreed upon for both groups to start work. This was critical for the project since every next step is dependent on

this. The discussions were done in meetings but the final decisions were always sent via email. The sample data saving formats were shared through github.

- Email thread to communicate the interface
- Github For code sharing

6.1.4 Collaboration of Group 17 with the Civil Engineering Department

This is out of the scope of this report.

6.2 Test driven development

Automated test cases play an important role in agile development. Constructing test cases together with the requirement engineering process not only ensures that the groups have significant clarity on their requirements; but it also helps to continuously test the code throughout the developments process. As such, tests were made based on 2 criteria:

- Use case testing
- Performance Testing

6.2.1 Use case testing

Use case test were constructed to ensure that the software satisfied the functional requirements of the user. In the context of this project the test cases which were generated at the beginning was the physical structures and their corresponding analytically values. These were used for integration / black box tests at the end of the project.
Apart from these, before the beginning of each task, specific test cases were created to ensure that each section of the code functioned properly. These acted as unit tests.
Python unittest module was used to test the code. Integration tests were done before each releases, but unit tests were done after each update / re-factoring.

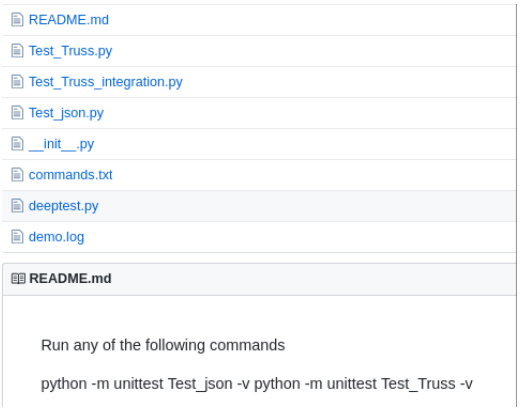

Figure 8: Use case testing scripts

6.2.2 Performance testing

This software is a CPU heavy computation. It is crucial to test it for performance to ensure that it will be usable for the internded purposes.
Usually the civil engineering structures have a large number of elements. The polynomial time algorithm may take longer time periods to give results. Therefor, the software was tested with different algorithms to make sure that it performs the computation in a time period which is not more than what the existing industrial solutions need.

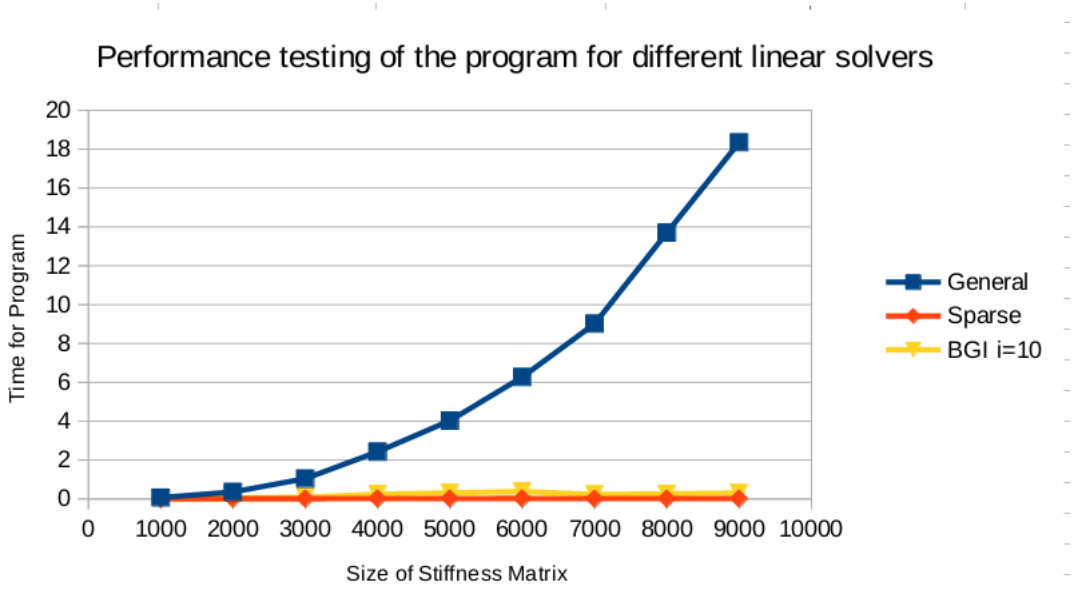

Figure 9: Performance testing

6.3 Continuous integration

Agile development places significant importance on working software and it's codes over planning and documentation: The argument being that a working software could be easily evaluated by the customers and other team members as opposed to an abstract design.

To facilitate continuous development and integration github was used as the online platform for collaborative coding. The figure shows the contributions of each member of the group.

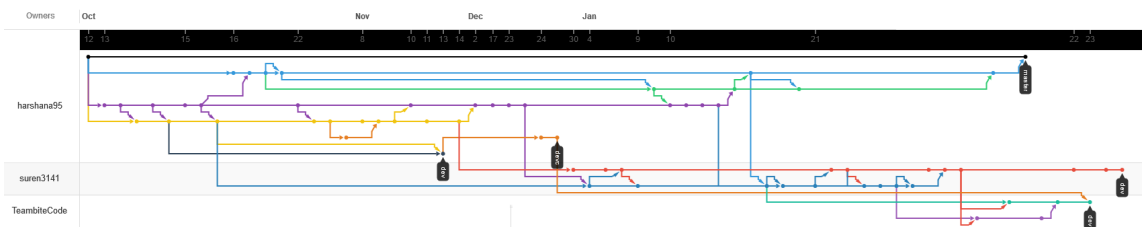

Figure 10: Network of contributions

6.3.1 Branches

In order to have a clear understanding the repository was divided into 2 branches: development and master.

Dev branch: All the group members push to their dev branch. They may pull from any other member's dev branch. Finally, once a milestone is reached, a member sends a pull request to the Team dev branch. After discussion, the scrum master pulls to the team dev branch.

Master branch: Individual group members do not push to their own master branch. The team discusses and pushes to the team's master branch from it's dev branch. The individual members may pull from the team maaster branch.
Master branch is also used as a reference for other group members (Civil Engineering Department and Group 17.

6.4 Timeline

The time line agreed with the project owner is shown below.

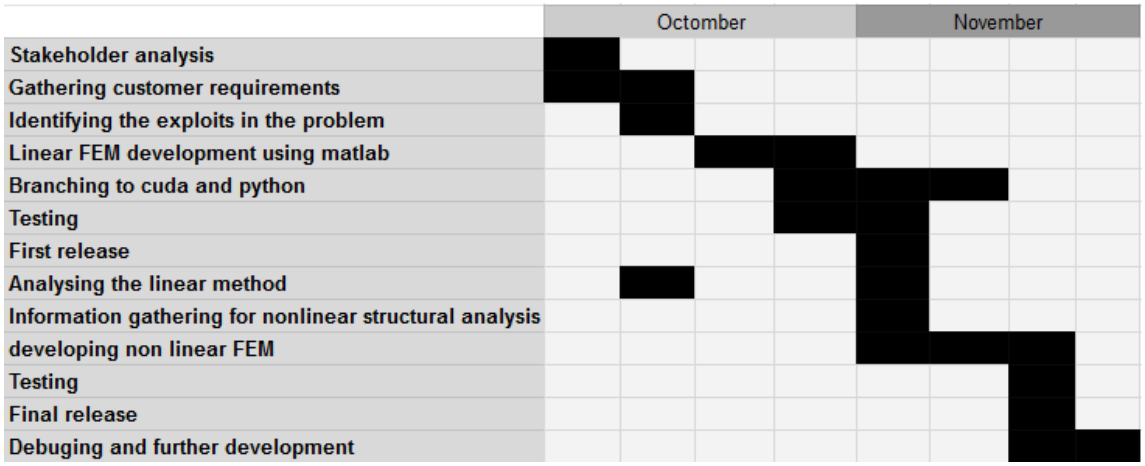

Figure 11: Planned time-line for the project

As shown, the project is expected to be completed by the end of the year. Within the current academic semester only part of it was expected to be completed. As per

this requirement, the linear solver has been developed and the interface has been completed for deployment .Therefore, for the CO328 : Software engineering course only this section has been completed and presented.
Additionally, the non-linear solvers have been developed, but requires testing to ensure that it works according to the customer requirements.

6.5 Releases

The software is released by the name **fem-struct** (Finite Element Method based STRUCTURAL analysis software). There are two versions given for different use cases.

- **Standalone software**
  This is released on github releases.
  https://github.com/TeambiteCode/finite-element-based-structural-analysis/releases
  This can be installed by running the script **install.sh** Then the program can be called by the shell keyword **fem-struct**.
- **Python package**
  https://test.pypi.org/project/fem-struct/
  This can be installed by the command
  pip install -i https://test.pypi.org/simple/fem-struct

7 Conclusion

The major objective of this software was to optimize the algorithm as well as it's implementation to achieve better performance considering both the time as well as the memory. Algorithmic optimization was the priority requirement of the project owner, and the given implementation was able to provide a 100x speedup. Although code optimization was a minor concern a significant improvement was achieved using good coding practices and re-factoring.

The entire algorithm is expected to be completed by the end of the year and would be integrated with the components made by the two other groups (front-end, manuals, etc) to form the final product.

8 Future work

The software developed until now concentrates only on linear solvers. Since there are many other alternative software which provide similar performances, the major part of the software would be the non-linear solver.

Although significant speedup was achieved through the modified implementation, the efficiency could be further improved. Especially parallel and low level (assembly) implementations could be considered as alternatives for this issue. Furthermore other techniques which could exploit the structure of the matrices (sparsity) were not analyzed in detail when improving the time efficiency of the algorithm. Having a deeper understanding of this structure could be vital in improving the efficiency.

References

[1] Wikimedia Foundation. Structural analysis - wikipedia.

[2] Dr. Jacob Moore. *Adaptive Map Digital Textbook for Engineering Statics.*

[3] Dr. Jacob Moore. *SAP2000 : BASIC ANALYSIS REFERENCE.*

[4] R.Nascimbene C.S.Bandara R.Dissanayakea R.M.C.M.Rajapakse, K.K.Wijesundara. Accounting axial-moment-shear interaction for force-based fiber modeling of rc frames.