Project Numerical Analysis

COSSE

August 6, 2018

1 Introduction

This report discusses the project of modelling the deformations of a framework given an initial state produced by prescribed forces. The model allows for a user to specify a framework of beams with constraints such as restricted end-points (bearings), transition conditions (end-points of two beams are connected) and stiff angles (angle between connected beams is constant). Next, the user is allowed to prescribe forces to each beam. An initial stable state is calculated by solving the bending equation for an elastic Bernoulli beam via the finite element method (FEM). Finally, the prescribed forces are removed and the dynamic deformations of the framework are calculated using the Newmark method and displayed to the user.

1.1 Aim and Objectives

The Aim of this project was to develop a software tool which simulates the dynamic bending of complex structures when subjected to different loads and simulate the results with respective to time. The Objectives of this project were as follows:

- Implement finite element methods for solving partial differential equations corresponding to horizontal and vertical deflection
- Model and simulate the deflection of a single beam acted upon a point load
- Develop temporal models to simulate beam deflection as a function of time
- Understand beam coupling theory and develop the above simulation when applied to a pre-defined frameworks of beams
- Develop a custom GUI for user to draw his/her own beam network, define loads, moments and constraints at each node
- Simulate the beam deflection behaviour as a function of time for the above custom-created framework

  The team has successfully implemented all the above objectives with positive results. The approach and results are discussed in the following sections

1.2 Motivation

The motivation to work on this problem arises from the need to understand the behaviour of beams in everyday scenarios surrounding us. From cranes, to overpasses to bridges, beams are ubiquitous in all such structures. They form the backbone that supports almost any load-bearing structure. Hence, it is vital to understand how they behave under the influence of different forces, especially with regards to their vertical and horizontal deflections.

An understanding of the deflection behaviour provides engineers key insights to be factored in their designs. For example, knowing how much a beam deflects for a given load, allows us to further know when a beam would buckle or collapse. This is valuable in designing structures safely. Engineers make use of deflection data to understand the maximum load permissible for a given safety factor and choice of material.

In a nutshell, the project stems from the motive to model beam deflection behaviour which serves as a input into structural engineering designs.

2 Theory

2.1 Static Beam Bending

2.1.1 Bernoulli Beam Equation

The Euler–Bernoulli equation describes the relationship between the beam's deflection and the applied load:

$$(EIw'')''(x) = q(x), x \in [0, L] \quad \text{(piecewise, i.e } q \in V,\ (EIw'') \in V)$$

where,

$w(x)$	Height of the neutral axis (bending curve) at x
$E = E(x)$	Young's modulus
$I = I(x)$	Area moment of Inertia: $I(x) = \int_{crosss\ section\ at\ x} z^2\ dydz$
q = q(x)	Load (force density) at x

The set of peicewise twice differentiable functions on the interval $[0, L]$ is denoted by:

$$V := C^{2,p}(0, L) = \{\ \phi\colon [0, L] \to \mathbb{R} \mid\ \phi \in C(0, L),\ \phi' \in C(0, L),\ \phi'' \in C^p(0, L)\ \}$$

Further notation:

$M^x(w) = EIw''(x)$	Bending moment at x
$Q^x(w) = -(EIw'')'(x)$	Shear force at x

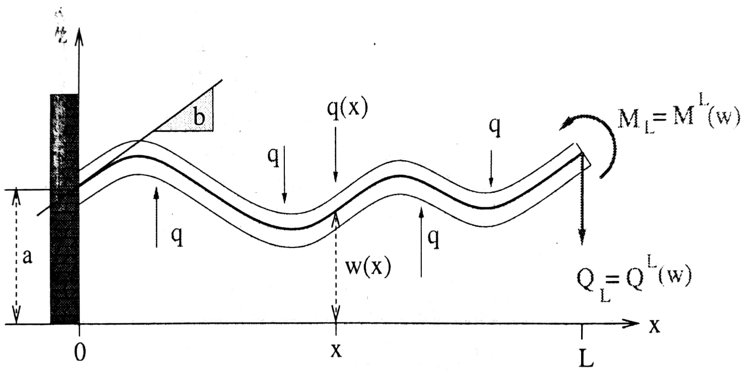

Figure 1: Clamped Beam

The general solution of the bending equation is

$$w = \left(\frac{q^{[2]} + p_1}{EI}\right)^{[2]} + p_2$$

where p_1, p_2 are arbitrary polynomials of degree ≤ 1. If EI is constant then the general solution of the bending equation is

$$w = \frac{q^{[4]}}{EI} + p$$

where p is a polynomial of degree ≤ 3. the general solution has 4 free parameters, the coefficients of p_1 and p_2. In order to specify a particular solution 4 boundary conditions are necessary. For a left sided clamped beam (cantilever)these boundary conditions are:

$$w(0) = a, \quad w'(0) = b, \quad Q^L(w) = Q_L, \quad M^L(w) = M_L,$$

where a, b, $Q_L, M_L \in \mathbb{R}$ are given numbers. The first two conditions, concerning $w(0)$ and $w(L)$, are called essential or Dirichlet Boundary conditions. The remaining two conditions, concerning the shear force $Q^L(w)$ and the bending moment $M^L(w)$, are called physical or natural boundary conditions.

2.1.2 Weak Formulation

Lemma. If $(EIw'') \in V$ then for all $\psi \in V$,

$$\int_0^L (EIw'')''\psi = \int_0^L EIw''\psi'' - \beta(w, \psi),$$

where

$$\beta(w, \psi) = Q^L(w)\psi(L) - Q_0(w)\psi(0) + M^L(w)\psi'(L) - M^0(w)\psi'(0)$$

.

Weak formulation of the bending equation. Let $w \in V$ and $Q_0, Q_L, M_0, M_L \in \mathbb{R}$. Then the following statements are equivalent.

(a) w satisfies the bending equation and the boundary conditions

$$Q^0(w) = Q_0, \quad Q^L(w) = Q_L, \quad M^0(w) = M_0 \quad M^L(w) = M_L.$$

(b) for all $\psi \in V$,

$$\int_0^L EIw''\psi'' = \int_0^L q\psi + b(\psi)$$

where

$$b(\psi) = Q_L\psi(L) - Q_0\psi(0) + M_l\psi'(L) - M_0\psi'(0)$$

Proof. (a) $\Rightarrow$ (b). Multiply the bending equation with $\psi \in V$ and integrate to obtain

$$\int_0^L q\psi = \int_0^L (EIw'')''\psi = \int_0^L (EIw'')\psi'' + \beta(w, \psi) = \int_0^L EIw''\psi'' + b(\psi)$$

(b) $\Rightarrow$ (a). If $\psi(0) = \psi'(0) = \psi(L) = \psi'(L) = 0$ then $b(\psi) = 0$, and hence we *(b),*

$$\int_0^L EIw''\psi'' = \int_0^L q\psi$$

Thus, $(EIw'')'' = q$ by the lemma of Du Bois-Reymond. Thus, by (b), we have for all $\psi \in V$,

$$\int_0^L EIw''\psi'' = \int_0^L q\psi + b(\psi) = \int_0^L (EIw'')''\psi + b(\psi) = \int_0^L EIw''\psi'' - \beta(w, \psi) + b(\psi)$$

Thus, $\beta(w, \psi) = b(\psi)$. Now, choose $\psi \in V$ such that $\psi(0) = 1, \psi'(0) = \psi(L) = \psi'(L) = 0$. It then follows that $Q^0(w) = \beta(w, \psi) = b(\psi) = Q_0$. Analogously, one concludes the other identities.

2.1.3 Galerkin Method

In order to compute an approximate solution w_h of the bending equation choose a finite dimensional subspace V_h (Ansatzspace) of V with the basis $\phi_1, ...\phi_N : [0, L] \to \mathbb{R}$. Let

$$w_h(x) = \sum_{k=1}^{N} w_k\phi_k(x), \quad w_k \in \mathbb{R}.$$

Insert this Ansatz in the weak formulation, and let $\psi = \phi_j, \quad j = 1, ..., N$. This yields the equations

$$\int_0^L EIw_h''\phi_j'' = \int_0^L q\phi_j + b(\phi_j), \qquad j = 1, ..., N.$$

Observe that

$$\int_0^L EIw_h''\phi_j'' = \sum_{k=1}^{N} \left(\int_0^L EI\phi_k''\phi_J'' \right) w_k.$$

hence,

$$\sum_{k=1}^{N} \left(\int_0^L EI\phi_k''\phi_j'' \right) w_k = \int_0^L q\phi_j + Q_L\phi_j(L) - Q_0\phi_j(0) + M_L\phi_j'(L) - M_0\phi_j'(0), \quad j = 1, ..., N$$

These are N linear equations for the N unknowns $w_1, ..., w_N$. The equations can be written in matrix-vector form as follows.

$$S\underline{w} = \underline{q} + Q_L\underline{e}_L - Q_0\underline{e}_0 + M_L\underline{d}_L - M_0\underline{d}_0,$$

where,

$$\underline{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}, \qquad \underline{q} = \begin{bmatrix} \int q\phi_1 \\ \vdots \\ \int q\phi_N \end{bmatrix}, \qquad \underline{e_x} = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_N(x) \end{bmatrix}, \qquad \underline{d}_x = \begin{bmatrix} \phi_1'(x) \\ \vdots \\ \phi_N'(x) \end{bmatrix},$$

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & \ldots & \vdots \\ s_{N1} & \cdots & s_{NN} \end{bmatrix}, \qquad s_{jk} = \int_0^L EI\phi_k''\phi_j''$$

The symmetric matrix $S \in \mathbb{R}^{NXN}$ is called the stiffness matrix. Note that for all x,

$$w_h(x) = \underline{e}_x^T\underline{w}, \quad w_h'(x) = \underline{d}_x^T\underline{w},$$

Remark. The stiffness matrix is positive semidefinite since for any $\underline{w} \in \mathbb{R}^n$,

$$\underline{w}^T S\underline{w} = \sum_{j,k} \Big(\int_0^L EI\phi_k''\phi_j'' \Big) w_j w_k = \int_0^L EI\Big(\sum_j w_j\phi_j'' \Big)\Big(\sum_k w_j\phi_k'' \Big) = \int_0^L \underbrace{\Big(\sum_j w_j\phi_j'' \Big)^2}_{\geq 0} \geq 0.$$

However, in general S is not positive definite since there might be non zero vectors $\underline{w}$ such that $\sum\limits_j w_j\phi_j'' = \Big(\sum\limits_j w_j\phi_j \Big)'' = 0$.

Remark. The elastic energy stored in the bended beam is given by the formula,

$$E(w) = \frac{1}{2}\int_0^L EI(w'')^2.$$

If w is of the form $w = \sum\limits_k w_k\phi_k$ then

$$E(w) = \frac{1}{2}\underline{w}^T S\underline{w}.$$

2.1.4 Choice of Ansatz Space

In principle the Galerkin ansatz works with an arbitrary ansatz spaceV_h (for instance one could take aa space of trigonometric functions). However, not every ansatz space contains a close approximation w_h to the exact solution w. a good choice for V_H is a space of piecewise cubic polynomials as described below.

Choose $n \geq 2$. Let $h = \frac{L}{n-1}, x_i = h(i-1), i = 1, ..., n-1$ and

The point x_i are called the nodes. Each function $\phi \in V_h$ is uniquely determined by the values $u_{2i-1} := \phi(x_i)$ and $u_{2i} := \phi'(x_i)$.

Each function $\phi \in V_h$ can be piecewise written as

$$\phi = \sum_{k=1}^{2n} u_k \phi_k = \sum_{i=1}^{n} (u_{2i-1}\phi_{2i-1} + u_{2i}\phi_{2i})$$

with the basis functions $\phi_1, \phi_2, ..., \phi_{2n-1}, \phi_{2n} \in V_h$ which are defined as follows. For $i = 1, 2, ..., n-1$,

$$\phi_{2i-1}(x) = \begin{cases} \bar{\phi}_3(\frac{x-x_{i-1}}{h}), & x \in [x_{i-1}, x_i] \\ \bar{\phi}_1(\frac{x-x_i}{h}), & x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise.} \end{cases} \qquad \phi_{2i}(x) = \begin{cases} h\bar{\phi}_4(\frac{x-x_{i-1}}{h}), & x \in [x_{i-1}, x_i] \\ h\bar{\phi}_2(\frac{x-x_i}{h}), & x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise.} \end{cases}$$

Additionally:

$$\phi_1(x) = \begin{cases} \bar{\phi}_1(\frac{x}{h}), & x \in [0, h] \\ 0 & \text{otherwise} \end{cases} \qquad \phi_2(x) = \begin{cases} h\bar{\phi}_2(\frac{x}{h}), & x \in [0, h] \\ 0 & \text{otherwise} \end{cases}$$

and

$$\phi_{2n-1}(x) = \begin{cases} \bar{\phi}_3(\frac{x-x_{n-1}}{h}), & x \in [x_{n-1}, L] \\ 0 & \text{otherwise} \end{cases} \qquad \phi_{2n}(x) = \begin{cases} h\bar{\phi}_4(\frac{x-x_{n-1}}{h}), & x \in [x_{n-1}, L] \\ 0 & \text{otherwise} \end{cases}$$

where,

$$\bar{\phi}_1(\xi) = 1 - 3\xi^2 + 2\xi^3, \qquad \bar{\phi}_3(\xi) = 3\xi^2 - 2\xi^3,$$
$$\bar{\phi}_2(\xi) = \xi(\xi-1)^2, \qquad \bar{\phi}_4(\xi) = \xi^2(\xi-1).$$

The functions $\bar{\phi}_i$ are called form functions.

Here are the graphs of the form functions:

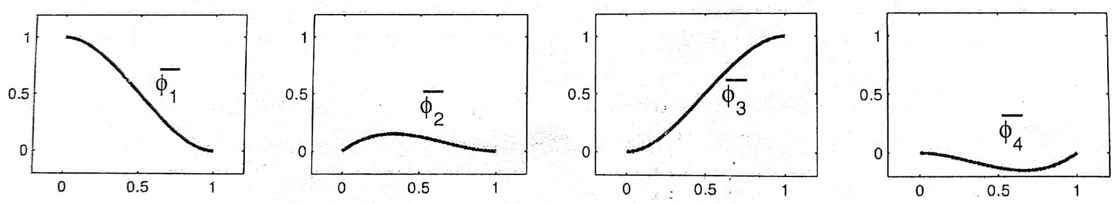

Figure 2: Form Functions:

The basis functions with odd index $2i-1$ have the form of a hump and attain the maximal value 1 at the node x_i. The basis functions with even index $2i$ have the form of a wave with slope 1 at the node x_i:

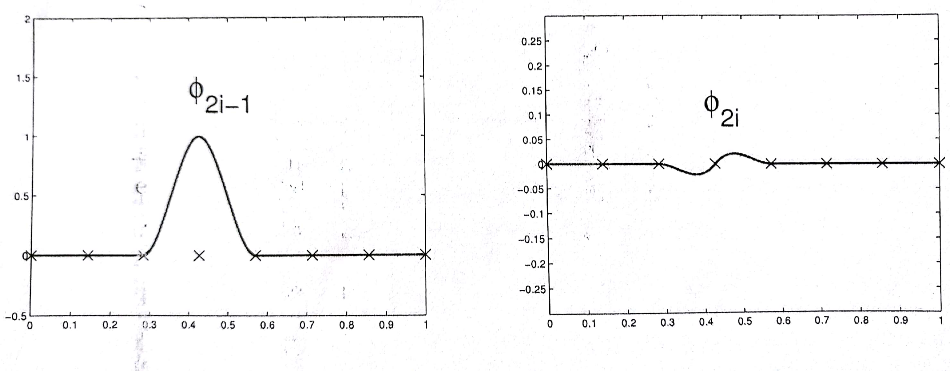

Figure 3: Basis Functions:

Remark. A one dimensional finite element is a finite interval with a set (vector space) of polynomials defined on it. So our ansatz space V_h is built from finite elements.

Remark. It can be shown that for our choice of the ansatz functions (piecewise cubic polynomials) each exact solution w of the bending equations (+ boundary equations) coincides with its Galerkin approximation w_h at the nodes: $w_h(x_i) = w_h(x_i), \;\; w_h'(x_i) = w'(x_i), \quad i = 1, ..., N.$

2.1.5 Boundary Conditions

In order to get a unique solution of the beam equation one has to add at least two Dirichlet boundary conditions. For a left sided clamped beam (cantilever) these conditions are $w(0) = a, \;\; w'(L) = b$, where $a, b \in \mathbb{R}$ are given. The Galerkin approximation should satisfy these conditions, too, i.e.

$$a = w_h(0) = \underline{e}_0^T \underline{w}, \quad b = w_h'(0) = \underline{d}_0^T \underline{w}.$$

These equations combined with (12) are equivalent to

$$\begin{bmatrix} S & \underline{e}_0 & \underline{d}_0 \\ \underline{e}_0^T & 0 & 0 \\ \underline{d}_0^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ Q_0 \\ M_0 \end{bmatrix} = \begin{bmatrix} \underline{q} + Q_L \underline{e}_L + M_L \underline{d}_L \\ a \\ b \end{bmatrix}$$

If the quantities on the right hand side of this linear equations are given then $\underline{w}$, Q_0 and M_0 can be computed. If the beam is supported (but not clamped) at both ends, then the corresponding equation is

$$\begin{bmatrix} S & \underline{e}_0 & -\underline{e}_L \\ \underline{e}_0^T & 0 & 0 \\ -\underline{e}_L^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ Q_0 \\ Q_L \end{bmatrix} = \begin{bmatrix} \underline{q} - M_0 \underline{d}_0 + M_L \underline{d}_L \\ a_0 \\ -a_L \end{bmatrix}$$

These equations are both of the form

$$\begin{bmatrix} S & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{\mu} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{a} \end{bmatrix}$$

where S is symmetric and positive semi definite. The matrix S_e is called the extended stiffness matrix. The matrix C must be such that S_e is non-singular. Otherwise the above equation does not have a unique solution.

Remark. The vector $\underline{x}$ is the solution of a constrained energy minimization problem:

$$\varepsilon(\underline{x}) = min\{ \;\varepsilon(\underline{\xi}), | \;\; \underline{\xi} \in \mathbb{R}^N, C^T \underline{\xi} = \underline{a} \;\; \}, \quad \varepsilon(\underline{\xi}) := \frac{1}{2} \underline{\xi}^T S \underline{\xi} - \underline{f}^T \underline{\xi}.$$

The vector $\underline{\mu}$ is the associated Lagrangian multiplier (consisting of constraint forces and constraint moments.) The mulitplier can be eliminated from the equation in the following way. Let

$$\hat{P} = C(C^TC)^{-1}C^T, \quad P = I - \hat{P}.$$

Then P and $\hat{P}$ are complementary orthogonal projectors. In particular,

$$\hat{x} = P\underline{x} + \hat{P}\underline{x} \;\; for\ all\ \underline{x} \in \mathbb{R}^N.$$

Furthermore, $PC = 0, \;\; \hat{P}C = C.$ On multiplying the equations

$$S\underline{x} + C\underline{\mu} = \underline{f}, \quad C^T\underline{x} = \underline{a}$$

with P and $C(C^TC)^{-1}$, respectively, one obtains

$$PS\underline{x} = P\underline{f}, \;\; \hat{P}\underline{x} = \underline{\tilde{a}}, \quad where\ \ \tilde{a} := C(C^TC)^{-1}\underline{a}.$$

Finally we get,

$$PSP\underline{x} = P(\underline{f} - S\underline{\tilde{a}}).$$

In order to obtain an equation for $\underline{x}$ with an invertible matrix one can add a multiple of $\hat{P}\underline{x}$:

$$\underbrace{(\gamma\hat{P} + PSP)}_{:=S_{mod}}\underline{x} = \underbrace{P(\underline{f} - S\underline{\tilde{a}}) + \gamma\underline{\tilde{a}}}_{=:f_{mod}}, \quad \gamma \in \mathbb{R}\backslash\{0\}$$

2.2 Dynamic Beam Deformation

Building on from the static case above, this section models the bending of the beam for a dynamic case i.e where external forces applied to the beam cause the deformation to vary with respect to time.

When it comes to modelling dynamic beam deformation two approaches were explored - the Newmark Method and the Eigenvalue Method.

2.2.1 The Newmark Method

The Newmark method is a numerical integration method for solving partial differential equations of the form,

$$f(\ddot{u}(t), \dot{u}(t), u(t), t) = 0,$$

In this method, the values of $u(t_j), \dot{u}(t_j), \ddot{u}(t_j)$ are iteratively approximated over multiple time steps using $u_j, \dot{u}_j, \ddot{u}_j$, where j represents the time step at a given instant.

The Newmark method can be used for solving a beam system of the form,

$$f(\ddot{u}(t), \dot{u}(t), u(t), t) = 0 = M\ddot{u} + D\dot{u} + Su - p(t)$$

where in our case D = 0.
From Taylor-expansions of $u(t_{j+1})$ and $\dot{u}(t_{j+1})$, we get

$$u_{j+1} = u_j + \dot{u}_j h_j + ((1/2 - \beta)\ddot{u}_j + \beta\ddot{u}_{j+1})h_j^2$$
$$\dot{u}_{j+1} = \dot{u}_j + ((1 - \gamma)\ddot{u}_j + \gamma\ddot{u}_{j+1})h_j$$

where $h_j = t_{j+1} - t_j$ is the step size.

Returning to the matrix system of the beams, in the dynamic case, a new time-dependant matrix component is introduced in the beam equation representing a mass matrix system of the form

$$\underbrace{\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}}_{M_e} \begin{bmatrix} \ddot{x} \\ \ddot{\mu} \end{bmatrix} + \underbrace{\begin{bmatrix} S & C \\ C^T & 0 \end{bmatrix}}_{S_e} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} f \\ a \end{bmatrix}$$

Applying the Newmark method to the above matrix system, it simplifies to solving the following linear system,

$$(M + \beta h_j^2 S)\ddot{u}_{j+1} = p(t_{j+1}) - Su_j$$

The obtained value of $\ddot{u}_j$ is used to update the values of u_{j+1} and $\dot{u}_{j+1}$ using the equations,

$$u_{j+1} = u_j + \beta \ddot{u}_{j+1} h_j^2 \; \dot{u}_{j+1} = \dot{u}_j + \gamma \ddot{u}_{j+1} h_j$$

The above iterations are repeated for each step of the entire time interval.

2.2.2 Eigenvalue Method

The Eigenvalue method is an alternative to model the dynamic behaviour for beam bending. It is applicable to the homogeneous case,

$$\underbrace{\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}}_{M_e} \begin{bmatrix} \ddot{x} \\ \ddot{\mu} \end{bmatrix} + \underbrace{\begin{bmatrix} S & C \\ C^T & 0 \end{bmatrix}}_{S_e} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The approach to this method is to solve the eigenvalue problem,

$$A \begin{bmatrix} w_k \\ \mu_k \end{bmatrix} = \lambda_k \begin{bmatrix} w_k \\ \mu_k \end{bmatrix} \text{ where } A = S_e^{-1} M_e$$

which yields $N - K$ independent eigenvectors $\begin{bmatrix} w_k \\ \mu_k \end{bmatrix}$
Solutions to the DAE method are then given by,

$$\begin{bmatrix} w(t) \\ \mu(t) \end{bmatrix} = \sum_{k=1}^{N-K} \left(\alpha_k cos(\omega t) + (\beta_k/\omega_k) sin(\omega_k t) \right) \begin{bmatrix} w_k \\ \mu_k \end{bmatrix}, \omega_k = 1/\sqrt{\lambda_k}$$

2.2.3 Implementation

The Eigenvalue method yielded an accurate result for the homogenous case. However due to its versatility and robustness, the Newmark method was opted for as the choice of method to model the dynamic case. The first part in implementing the Newmark method is to define the constants β and γ. While the parameters can take on a range of values the optimal parameters are $\beta = 1/4, \gamma = 1/2$.

The next step is to set the time steps for the discretization. In the implementation we have found that having a final time of 20s with 200 time steps resulted in an optimal simulation and visualisation of the solution.

Once these initial conditions have been set, the Newmark method was implemented as per the iterative algorithm detailed in section 2.3.1

2.3 Construction of Framework

In order to construct a framework of beams, the curvature of a bent beam must first be calculated in local coordinates and then converted to global coordinates. The curvature of a bent beam is described as follows in local coordinates, where v is the horizontal displacement and w is the vertical displacement:

$$\beta(x,t) = \begin{bmatrix} x + v(x,t) \\ x(x,t) \end{bmatrix} = \underbrace{\begin{bmatrix} x \\ 0 \end{bmatrix}}_{\text{Original position}} + \underbrace{\begin{bmatrix} v(x,t) \\ w(x,t) \end{bmatrix}}_{\text{Displacement vector}}$$

The curvature of the bent beam in global coordinates is then determined by displacing the beam to point $p = (a, b)$, where a and b are the x and y coordinates of the origin of the beam in the global coordinate system, and by using the following rotation matrix, $D_ø$, to rotate the beam by the correct angle, ø:

$$D_ø = \begin{bmatrix} \cos(ø) & -\sin(ø) \\ \sin(ø) & \cos(ø) \end{bmatrix}$$

This results in the following definition of the curve of the bent beam in global coordinates:

$$\beta_g(x, t) = p + D_ø \beta(x, t) = \begin{bmatrix} a + \cos(ø) \cdot (x + v(x, t)) - \sin(ø) \cdot w(x, t) \\ b + \sin(ø) \cdot (x + v(x, t)) + \cos(ø) \cdot w(x, t) \end{bmatrix}$$

2.3.1 Types of Constraints

There are three geometric constraints accounted for in the model. They are as follows:

1. *Bearings:* The motion of some beam end points is restricted. The bending angle of the fixed bearing may be zero.

2. *Transition conditions:* The end points of some beams should always be coincident.

3. *Stiffness of angles:* The angle between two connected beams should be constant in time. Note that this condition may not be satisfied for all beams.

2.3.2 Modelling of Constraints

In order to satisfy the geometric conditions, forces and moments must be present at the beam endpoints in order to constrain them. This was achieved by representing the constrains in terms of the entries of x and μ as a product of terms. This matrix, C, was then added to the stiffness matrix, resulting in an extended stiffness matrix, as shown below, where M_e ans S_e are the extended mass matrix and stiffness matrix respectively:

$$\underbrace{\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}}_{M_e} \begin{bmatrix} \ddot{x} \\ \ddot{\mu} \end{bmatrix} + \underbrace{\begin{bmatrix} S & C \\ C^T & 0 \end{bmatrix}}_{S_e} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} f \\ a \end{bmatrix}$$

2.3.3 Representation of Constraints as Inner Product

The different types constraints each had their own form of representation in the constraint matrix. They are as follows:

Bearings The end points of each beam could be fixed in x, y or both. This involves ensuring the node i has the following properties:

$$v_i(0, t) = 0$$
$$w_i(0, t) = 0$$
$$w_i'(0, t) = 0$$

Coupling: fixed position In order to connect two beams together, the end nodes of each beam, i and j, must have the following properties:

$$\cos(\phi_i)v_i(L_i, t) - \sin(\phi_i)w_i(L_i, t) = \cos(\phi_j)v_j(0, t) - \sin(\phi_j)w_j(0, t)$$
$$\sin(\phi_i)v_i(L_i, t) + \cos(\phi_i)w_i(L_i, t) = \sin(\phi_j)v_j(0, t) + \cos(\phi_j)w_j(0, t)$$

Coupling: stiff angle The angle of the connection between the beams can also be fixed using the following properties:

$$w_i'(L_i, t) = w_j'(0, t)$$

Each of these properties can be added to the model by adding the appropriate constraint to the constrains matrix C at the appropriate index.

3 Method

The approach to solving the given beam framework project was to break it down into more basic simpler components. The final solution was built bottom-up progressively in increasing complexity. During the first phase, a single beam fixed at one end was modelled. A point force acting on the other end was applied and the results were observed. For a basic scenario as this one, analytical solutions exist and therefore, the obtained results could be compared with the analytical solution.

This was followed by modelling dynamic case for the same single beam, where as an exercise both the Newmark Method and the Eigenvalue Method were implemented. As detailed previously, the Newmark method was chosen for its versatility and robustness.

Upon confirming the accuracy results for the static and dynamic case on the single beam, the team proceeded to tackle the approach of coupling, keeping in mind the need to design for complex structures where more than just two beams meet at any given point. This was found to be the most challenging aspect of the project. There were many "what-if" scenarios that arose and had to be accurately accounted for. Often, minor changes in the conditions would result in significant updates to our approach. Despite the challenges, the team has put together a sound working solution capable of analysing complex meshes of beams and producing positive results. Having tackled the design and simulation aspects, the final step was to make the project as user friendly as possible i.e allow the user complete flexibility in choosing his/her framework of choice, moments and forces to be applied. In short, using a GUI the user could draw and specify the conditions as necessary from scratch. This liberates the use case of the project from only presenting results of limited pre-defined use cases but instead allows the user complete freedom to customize for any structure.

4 Results

In order to verify the program, the static and dynamic cases were analysed. Ideally, real-world experiments could be tested against the model, however it was decided to be sufficient to compare the model to other, well-understood mathematical and analytical solutions. This would be determined to be a sufficient validation of the model.

The advantage of analysing the problem this way means that as long as the mathematical models are accurate, then the fidelity of the model can be accurately determined. A future analysis could be conducted with different test cases, such as the three point bending test, and using different materials. Comparing the data with simulated results would be a good way to validate the implementation and design choices.

4.1 Validation of Static Case

Figure 4 shows the initial beam deflection as calculated by the program, and compared to the analytically solved Bernoulli beam. It shows an accuracy approximately on the scale of machine precision. For this set-up, the relative error in the horizontal error was found to be $9.881\text{e}{-16}$, and the relative error in the vertical direction was found to be $2.5865\text{e}{-14}$.

This high accuracy shows that the implementation is accurately modelling the equations for Bernoulli beams for the initial static case.

4.2 Comparison of Dynamic Case Models

It is much more difficult to develop an analytical model for the dynamic case than it is for the static case. For this reason, the model was compared to another model for deflection of beams that uses a completely different approach. In this case the model was compared to a solution using the Eigenvalue method. The final state of both approaches are shown in Figure 5.

Here the numerical computation of both methods produced very similar results. Figure 6 shows how close the two approaches are for 400 time steps. This constant pattern likely indicates that the two models are slightly out of phase while still being very similar. The maximum relative error for this test was found to be 0.3244.

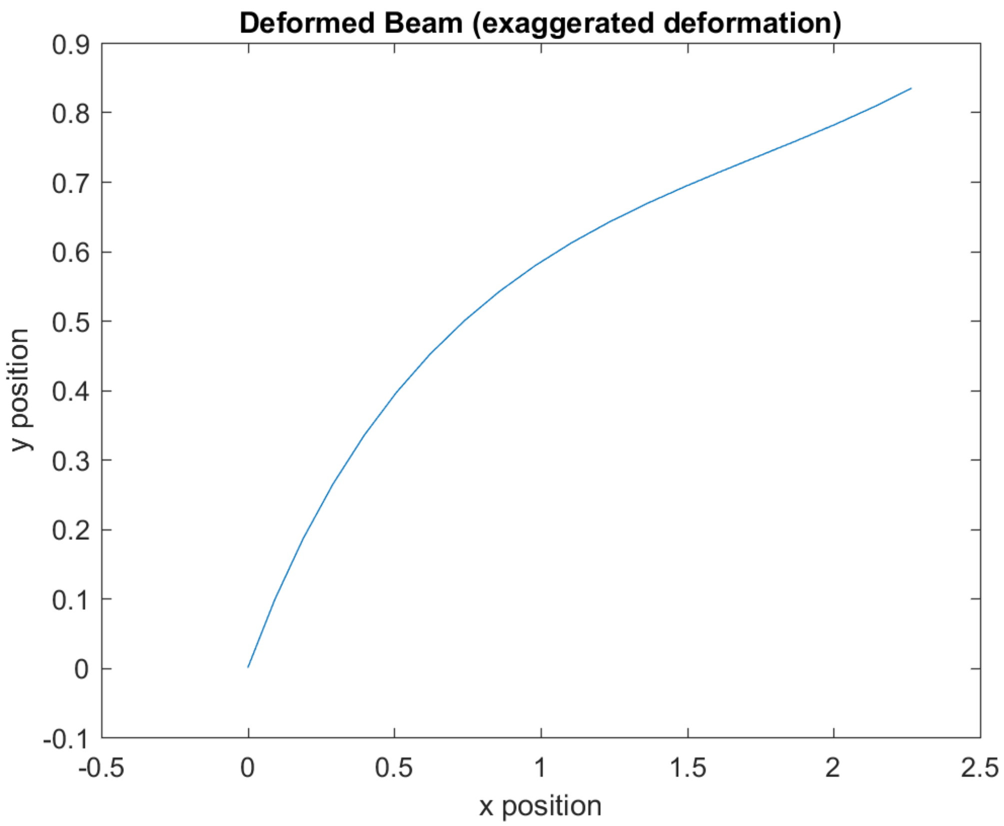

Figure 4: Finite element vs. analytical solution for static deflection

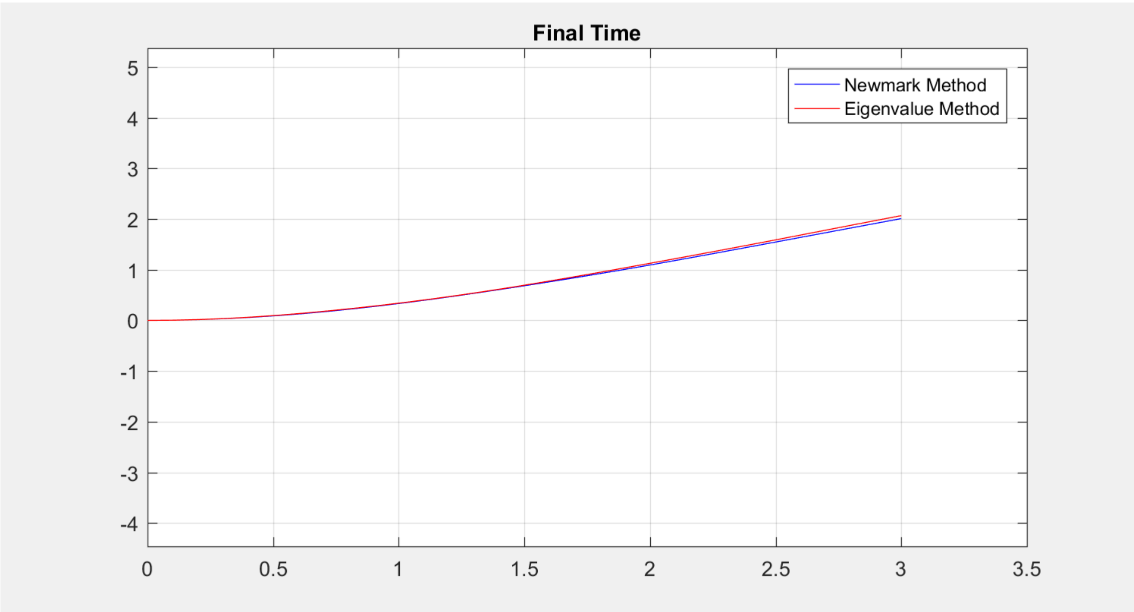

Figure 5: Newmark vs. Eigenvalue method for dynamic deflection

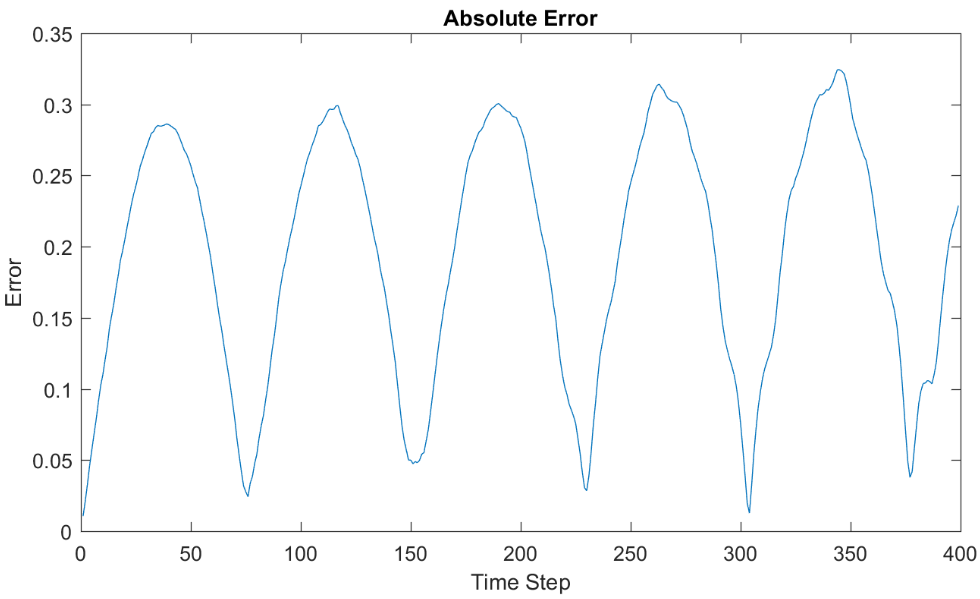

Figure 6: Newmark vs. Eigenvalue method for dynamic deflection

5 Discussion

The implementation of the theory from Section 2 lead to several interesting points. For example, it can be noted that an apparently regular structure appeared in the respective matrix elements for for different of beam connections. For example, the network shown below in Figure 7 produced a stiffness matrix with sparsity pattern shown in Figure 8. It can clearly be seen that this simple network with well defined constraints leads to the construction of a sparse stiffness matrix.

The implementation used in this project made use of symbolic variables to construct the stiffness matrix as a dense matrix, which was then evaluated. This method of implementation was selected as it is very flexible and the same method can be used to build a wide variety of networks. These networks could contain any of the constraints mentioned above, and any configuration of beams. The disadvantage of this technique is that it takes longer to construct the matrix.

The implementation used in this project made use of symbolic variables to construct the stiffness matrix as a dense matrix, which was then evaluated. This method of implementation was selected as it is very flexible and the same method can be used to build a wide variety of networks. These networks could contain any of the constraints mentioned above, and any configuration of beams.

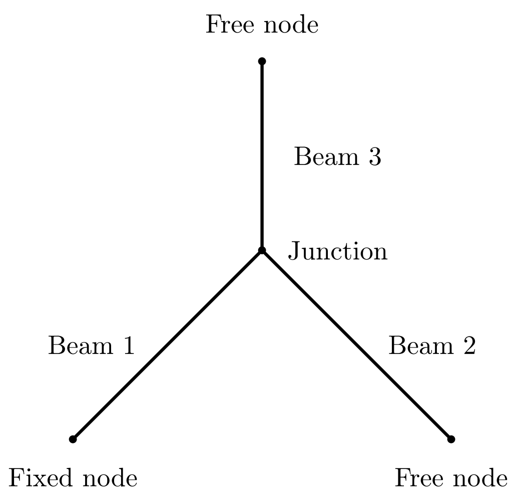

Figure 7: Example network

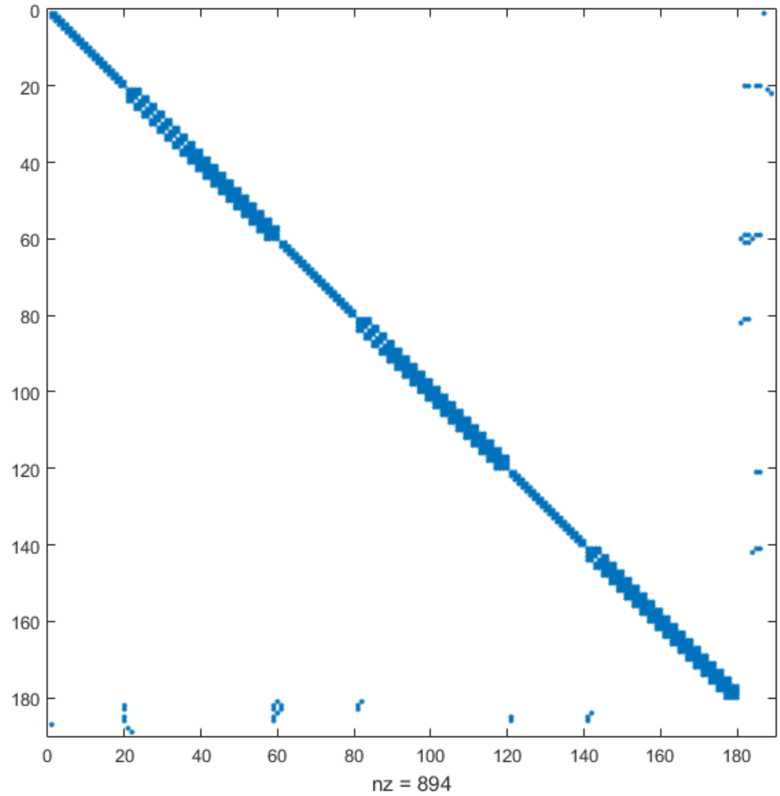

Figure 8: Extended stiffness matrix

This was necessary as the stiffness matrix made use of third degree polynomials as basis functions, and the patterns in the matrix of the lattice formed by these basis functions with their overlaps had not been derived as it had been for using hat functions, for example.

The disadvantage of this technique is that it takes longer to construct the matrix. A potential improvement that could be made on this model is that the underlying pattern relating the matrix elements and the lattice spacing could be derived, in the same way as for hat functions. This would likely significantly improve the performance of matrix assembly as only the non-zero elements of the stiffness matrix would be computed, instead of all of the elements, as is the case for a dense matrix.

On a more technical side, issues such as over-determination of the system were encountered. This occurred when the initial boundary conditions were set up in such a way that no simple solution could be solved for, such as with many fixed angles that cannot be resolved. A detection for this kind of overdetermined system with a warning for the user could be beneficial.

Overall, the team met the objectives that were set out and achieved the aim of developing a software tool capable of simulating the dynamic bending of complex structures subjected to various loads.

6 Conclusion

The project was a learning experience across an array of topics related to solving partial differential equations ranging from finite element element methods, coupling of structures, time-based discretization, modelling of boundary conditions. While it was important to gain a sound grasp of the theory, the learning from the project stretched beyond. Implementing all the theoretical knowledge into a numerical algorithm using MATLAB code was a deep dive into programming in MATLAB.

The team is extremely content with the outcomes achieved during the course of the project. There were multiple challenges faced along the way as the implementation of some of the concepts became progressively more complex. However, despite the steep learning curve, it was satisfying to overcome the challenges. The team hopes that this project can be used as a reference example for future projects.

References

Supplementary notes provided by Prof Karow