Finite Element Formulation for an Extended Euler-Bernoulli Beam with Axial Forces

This document derives and implements a finite element formulation for a fixed-fixed Euler–Bernoulli beam incorporating both axial and bending dynamics. The governing equations are discretized using linear and Hermite shape functions, with consistent mass and stiffness matrices assembled for each element. Time integration is performed via both implicit (Newmark–Beta) and explicit (central difference) schemes, and control inputs are applied to simulate vibration suppression through axial forces and moments.

1 Nomenclature

E	[Pa]	Young's modulus (steel: 2.1×10^{11}).
A	[m^2]	Cross-sectional area (e.g. 0.01).
I	[m^4]	Second moment of area (e.g. 8.333×10^{-6}).
ρ	[kg/m^3]	Mass density (e.g. 7850).
L	[m]	Total beam length (e.g. 2).
L_e	[m]	Element length: $L_\mathrm{e} = L/n_\mathrm{elem}$.
x	[m]	Spatial coordinate.
t	[s]	Time coordinate.
$u(x,t)$	[m]	Axial displacement.
$w(x,t)$	[m]	Transverse displacement.
$\delta u,\ \delta w$	–	Virtual (test) displacements.
K_e	–	Full element stiffness matrix (6×6).
M_e	–	Full element mass matrix (6×6).

2 Governing Equations

The classical Euler–Bernoulli beam equation for transverse deflection $w(x,t)$ is

$$EI\, w''''(x,t) + \rho A\, \ddot{w}(x,t) = q(x,t). \tag{1}$$

With an added axial degree of freedom, the beam's strong form separates into two equations:

$$\rho A\, \ddot{u}(x,t) - \big(EA\, u'(x,t)\big)' = q(x,t), \tag{2}$$

$$\rho A\, \ddot{w}(x,t) - \big(EI\, w''(x,t)\big)'' = q(x,t). \tag{3}$$

An example of beam under analysis is shown in Figure 1. The beam is fixed at both ends and loaded transversely at its center by a point force P, resulting in internal axial forces N_A, N_B, bending moments M_A, M_B, and vertical reaction forces F_A, F_B.

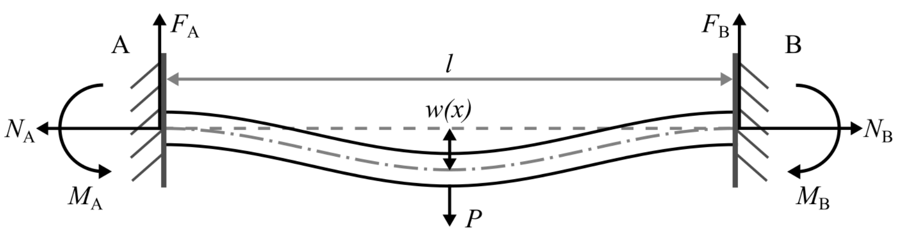

Figure 1: Free body diagram of a fixed-fixed beam subjected to a transverse point load.

3 Weak Formulation of Beam Dynamics

To reduce the continuity requirements on trial and test spaces and to impose natural boundary conditions, we derive the variational form by multiplying each strong equation by its virtual displacement and integrating by parts.

3.1 Axial Weak Form

Multiply Eq. (2) by δu, integrate by parts:

$$\int_0^L EA\, u'\, \delta u'\, dx - \big[EA\, u'\, \delta u\big]_0^L = 0 \implies \int_0^L EA\, u'\, \delta u'\, dx = 0. \tag{4}$$

3.2 Bending Weak Form

Multiply Eq. (3) by δw, integrate by parts twice:

$$\int_0^L EI\, w''\, \delta w''\, dx - \big[EI\, w''\, \delta w' - (EI\, w'')'\, \delta w\big]_0^L = 0 \implies \int_0^L EI\, w''\, \delta w''\, dx = 0. \tag{5}$$

4 Finite-Element Shape Functions and Interpolation

4.1 Linear Shape Functions for Axial Displacement

We approximate $u(x)$ over each element of length L_e by

$$N^{(u)}(x) = \begin{pmatrix} 1 - \frac{x}{L_\mathrm{e}} \\ \frac{x}{L_\mathrm{e}} \end{pmatrix}, \qquad u(x) = \big[N^{(u)}(x)\big]^\mathrm{T} (u_1, u_2)^\mathrm{T}. \tag{6}$$

4.2 Hermite Cubic Interpolation for Bending

To capture bending curvature with C^1 continuity, we use four cubic polynomials:

all of the subscript "e"s should not be italic, they are not a variable, they are a descriptor. I fixed those them, but just make sure to fix this in the future.

$$\begin{aligned} N_1(x) &= 1 - 3(\tfrac{x}{L_\mathrm{e}})^2 + 2(\tfrac{x}{L_\mathrm{e}})^3, \\ N_2(x) &= x[1 - 2(\tfrac{x}{L_\mathrm{e}}) + (\tfrac{x}{L_\mathrm{e}})^2], \\ N_3(x) &= 3(\tfrac{x}{L_\mathrm{e}})^2 - 2(\tfrac{x}{L_\mathrm{e}})^3, \\ N_4(x) &= x[-(\tfrac{x}{L_\mathrm{e}}) + (\tfrac{x}{L_\mathrm{e}})^2]. \end{aligned} \tag{7}$$

Mapping $x \to \xi = x/L_\mathrm{e} \in [0,1]$ for integration:

$$\begin{aligned} &N_1(\xi) = 1 - 3\xi^2 + 2\xi^3, \quad N_2(\xi) = L_\mathrm{e}\, \xi(1 - 2\xi + \xi^2), \\ &N_3(\xi) = 3\xi^2 - 2\xi^3, \quad N_4(\xi) = L_\mathrm{e}\, \xi(-\xi + \xi^2). \end{aligned} \tag{8}$$

5 Derivation of Element Stiffness Matrices

Substituting the above interpolations into the weak forms (4)–(5) and evaluating the resulting integrals yields:

5.1 Axial

By inserting $u(x)$ from (6) into (4) and noting the constant derivative,

$$K_\mathrm{e}^{(\mathrm{axial})} = \int_0^{L_\mathrm{e}} EA\, \frac{dN^{(u)}}{dx} \frac{dN^{(u)}}{dx}^\mathrm{T} dx = \frac{EA}{L_\mathrm{e}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{9}$$

5.2 Bending

From the bending weak form (5), define the curvature–shape vector in ξ:

$$B(\xi) = \frac{d^2 N^{(w)}}{dx^2} = \frac{1}{L_\mathrm{e}^2}[6\xi - 6,\ 3\xi - 4,\ -6\xi + 6,\ 3\xi - 2], \tag{10}$$

then

$$K_\mathrm{e}^{(\mathrm{bending})} = EI \int_0^{L_\mathrm{e}} B^\mathrm{T} B\, dx = \frac{EI}{L_\mathrm{e}^3} \begin{pmatrix} 12 & 6L_\mathrm{e} & -12 & 6L_\mathrm{e} \\ 6L_\mathrm{e} & 4L_\mathrm{e}^2 & -6L_\mathrm{e} & 2L_\mathrm{e}^2 \\ -12 & -6L_\mathrm{e} & 12 & -6L_\mathrm{e} \\ 6L_\mathrm{e} & 2L_\mathrm{e}^2 & -6L_\mathrm{e} & 4L_\mathrm{e}^2 \end{pmatrix}. \tag{11}$$

5.3 Element-Level Assembly of Stiffness Matrix

Finally, the full 6×6 element stiffness matrix is

$$K_\mathrm{e} = \begin{pmatrix} K_\mathrm{e}^{(\mathrm{axial})} & 0 \\ 0 & K_\mathrm{e}^{(\mathrm{bending})} \end{pmatrix}. \tag{12}$$

6 Derivation of Element Mass Matrices

The element mass matrix comes from the beam's kinetic energy:

$$T = \tfrac{1}{2} \int_0^{L_\mathrm{e}} \rho A\, \dot{u}^2(x,t)\, dx + \tfrac{1}{2} \int_0^{L_\mathrm{e}} \rho A\, \dot{w}^2(x,t)\, dx. \tag{13}$$

6.1 Axial

We start with the axial kinetic energy

$$T_\mathrm{axial} = \tfrac{1}{2} \int_0^{L_\mathrm{e}} \rho A\, \dot{u}^2\, dx. \tag{14}$$

Substitute the linear interpolation (6):

$$\dot{u}(x,t) = \left[N^{(u)}(x)\right]^\mathrm{T} \dot{u}, \tag{15}$$

which gives

$$T_\mathrm{axial} = \tfrac{1}{2}\, \dot{u}^\mathrm{T} \left[\int_0^{L_\mathrm{e}} \rho A\, N^{(u)} N^{(u)T}\, dx\right] \dot{u} = \tfrac{1}{2}\, \dot{u}^\mathrm{T}\, M_\mathrm{e}^{(\mathrm{axial})}\, \dot{u}. \tag{16}$$

Hence the consistent axial mass block is

$$M_\mathrm{e}^{(\mathrm{axial})} = \int_0^{L_\mathrm{e}} \rho A\, N^{(u)} N^{(u)T}\, dx = \frac{\rho A L_\mathrm{e}}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}. \tag{17}$$

6.2 Bending

Likewise, for bending

$$T_\mathrm{bending} = \tfrac{1}{2} \int_0^{L_\mathrm{e}} \rho A\, \dot{w}^2\, dx. \tag{18}$$

Using the Hermite-cubic interpolation (7):

$$\dot{w}(x,t) = \left[N^{(w)}(\xi)\right]^\mathrm{T} \dot{w}, \tag{19}$$

we obtain

$$T_\mathrm{bending} = \tfrac{1}{2}\, \dot{w}^\mathrm{T} \left[\int_0^{L_\mathrm{e}} \rho A\, N^{(w)}(\xi) N^{(w)T}(\xi)\, dx\right] \dot{w} = \tfrac{1}{2}\, \dot{w}^\mathrm{T}\, M_\mathrm{e}^{(\mathrm{bending})}\, \dot{w}. \tag{20}$$

Thus the bending mass block is

$$M_\mathrm{e}^{(\mathrm{bending})} = \int_0^{L_\mathrm{e}} \rho A\, N^{(w)}(\xi) N^{(w)T}(\xi)\, dx = \frac{\rho A L_\mathrm{e}}{420} \begin{pmatrix} 156 & 22L_\mathrm{e} & 54 & -13L_\mathrm{e} \\ 22L_\mathrm{e} & 4L_\mathrm{e}^2 & 13L_\mathrm{e} & -3L_\mathrm{e}^2 \\ 54 & 13L_\mathrm{e} & 156 & -22L_\mathrm{e} \\ -13L_\mathrm{e} & -3L_\mathrm{e}^2 & -22L_\mathrm{e} & 4L_\mathrm{e}^2 \end{pmatrix}. \tag{21}$$

6.3 Element-Level Assembly of Mass Matrix

Putting axial and bending together gives the full element mass matrix:

$$M_e = \begin{pmatrix} M_e^{(\text{axial})} & 0 \\ 0 & M_e^{(\text{bending})} \end{pmatrix}. \tag{22}$$

7 Global System and Time Integration Methods

The global equation of motion for the discretized beam—assembled from all element mass, damping, and stiffness contributions—can be written in matrix form as

$$\underbrace{\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} \ddot{u}_1 \\ \ddot{w}_1 \\ \ddot{\theta}_1 \\ \vdots \\ \ddot{\theta}_N \end{bmatrix}}_{\ddot{W}} + \underbrace{\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} \dot{u}_1 \\ \dot{w}_1 \\ \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_N \end{bmatrix}}_{\dot{W}} + \underbrace{\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} u_1 \\ w_1 \\ \theta_1 \\ \vdots \\ \theta_N \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} F_1(t) \\ F_2(t) \\ F_3(t) \\ \vdots \\ F_n(t) \end{bmatrix}}_{F(t)}. \tag{23}$$

Here:

- $W = [u_1, w_1, \theta_1, \ldots, u_N, w_N, \theta_N]^{\mathrm{T}}$ collects all $3N$ nodal degrees of freedom: u_i (axial displacement), w_i (transverse displacement), and θ_i (slope) at node i.
- $\ddot{W}$ and $\dot{W}$ are the corresponding acceleration and velocity vectors.
- $M = [m_{ij}]$ is the global mass matrix, assembled from each element's consistent mass blocks (Eqs. (17) and (21)).
- $C = [c_{ij}]$ is the global damping matrix (e.g. Rayleigh damping $C = \alpha M + \beta K$).
- $K = [k_{ij}]$ is the global stiffness matrix, assembled from each element's axial and bending stiffness (Eqs. (9)–(11)).
- $f(t) = [f_1(t), f_2(t), \ldots, f_n(t)]^{\mathrm{T}}$ is the vector of external (or control) forces and moments applied at each DOF.

Equation (23) succinctly represents the dynamic equilibrium of the entire beam structure in its finite-element discretization.

7.1 Implicit Newmark–Beta Time Integration

The Newmark–Beta method is a widely used implicit time integration scheme for solving second-order systems of the form:

$$M\,\ddot{w} + C\,\dot{w} + K\,w = f(t), \tag{24}$$

where $w, \dot{w}, \ddot{w}$ denote displacement, velocity, and acceleration vectors. The matrices M, C, K are the mass, damping, and stiffness matrices. The damping matrix is often modeled using Rayleigh damping.

7.1.1 Rayleigh Damping Coefficients

Rayleigh damping is modeled by a linear combination of the mass and stiffness matrices:

$$C = \alpha_R M + \beta_R K, \tag{25}$$

where α_R and β_R are the mass- and stiffness-proportional damping coefficients, respectively.

To determine α_R and β_R, we specify the desired damping ratios ζ_1 and ζ_2 for two selected natural modes with angular frequencies ω_1 and ω_2. The damping ratio for each mode is defined as:

$$\zeta_i = \frac{\alpha_R}{2\omega_i} + \frac{\beta_R \omega_i}{2}, \quad i = 1, 2. \tag{26}$$

This yields the following system of linear equations:

$$\begin{cases} \zeta_1 = \dfrac{\alpha_R}{2\omega_1} + \dfrac{\beta_R \omega_1}{2}, \\[2ex] \zeta_2 = \dfrac{\alpha_R}{2\omega_2} + \dfrac{\beta_R \omega_2}{2}. \end{cases}$$

This system can be rewritten in matrix form:

$$\begin{bmatrix} \dfrac{1}{2\omega_1} & \dfrac{\omega_1}{2} \\[2ex] \dfrac{1}{2\omega_2} & \dfrac{\omega_2}{2} \end{bmatrix} \begin{bmatrix} \alpha_R \\[2ex] \beta_R \end{bmatrix} = \begin{bmatrix} \zeta_1 \\[2ex] \zeta_2 \end{bmatrix}. \tag{27}$$

Let this system be denoted as $Ax = b$, where:

$$A = \begin{bmatrix} \dfrac{1}{2\omega_1} & \dfrac{\omega_1}{2} \\[2ex] \dfrac{1}{2\omega_2} & \dfrac{\omega_2}{2} \end{bmatrix}, \quad x = \begin{bmatrix} \alpha_R \\[2ex] \beta_R \end{bmatrix}, \quad b = \begin{bmatrix} \zeta_1 \\[2ex] \zeta_2 \end{bmatrix}.$$

The damping coefficients are obtained by solving:

$$\begin{bmatrix} \alpha_R \\[2ex] \beta_R \end{bmatrix} = A^{-1} b. \tag{28}$$

This procedure ensures that the resulting damping matrix C produces the desired damping ratios at the specified frequencies. The coefficient α_R primarily affects low-frequency (inertial) damping, while β_R dominates in high-frequency (stiffness-driven) modes.

7.1.2 Implicit Time Integration

At each time step, the Newmark–Beta scheme computes the unknown displacement w_{n+1}, then updates acceleration and velocity using known values from the previous step. The method uses the following interpolation formulas:

$$w_{n+1} = w_n + \Delta t\, \dot{w}_n + \frac{1}{2}\Delta t^2 \left[(1-2\beta)\ddot{w}_n + 2\beta \ddot{w}_{n+1}\right], \tag{29}$$

$$\dot{w}_{n+1} = \dot{w}_n + \Delta t \left[(1-\gamma)\ddot{w}_n + \gamma \ddot{w}_{n+1}\right], \tag{30}$$

where β and γ are method parameters. The common choice $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$ gives unconditional stability and second-order accuracy.

To initiate the time integration, the initial acceleration is computed using:

$$\ddot{w}_0 = M^{-1}\left[f_0 - C\, \dot{w}_0 - K\, w_0\right]. \tag{31}$$

The effective stiffness matrix is defined by:

$$K_{\text{eff}} = K + \frac{\gamma}{\beta \Delta t} C + \frac{1}{\beta \Delta t^2} M. \tag{32}$$

The effective force vector is assembled from known values:

$$F_{\text{eff}} = f_{n+1} + M\left(\frac{1}{\beta\Delta t^2} w_n + \frac{1}{\beta \Delta t}\dot{w}_n + \frac{1-2\beta}{2\beta}\ddot{w}_n\right) \tag{33}$$

$$+ C\left(\frac{\gamma}{\beta\Delta t} w_n + \left(\frac{\gamma}{\beta} - 1\right)\dot{w}_n + \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\ddot{w}_n\right). \tag{34}$$

The displacement is computed by solving:

$$K_{\text{eff}}\, w_{n+1} = F_{\text{eff}}. \tag{35}$$

The acceleration and velocity at the new time step are updated as:

$$\ddot{w}_{n+1} = \frac{1}{\beta \Delta t^2} \left(w_{n+1} - w_n - \Delta t \dot{w}_n \right) - \frac{1 - 2\beta}{2\beta} \ddot{w}_n, \tag{36}$$

$$\dot{w}_{n+1} = \dot{w}_n + \Delta t \left[(1 - \gamma) \ddot{w}_n + \gamma \ddot{w}_{n+1} \right]. \tag{37}$$

The influence of Rayleigh damping is embedded directly in the matrix solve. The coefficient α_R provides damping for lower-frequency (mass-dominated) modes, while β_R suppresses higher-frequency (stiffness-dominated) vibrations. This results in a stable and physically realistic solution, particularly in systems with broad frequency content.

7.1.3 Central Difference Explicit Integration

The central difference method is a second-order explicit time integration technique suitable for wave propagation and dynamic response problems. The update of displacement is given by:

$$w_{n+1} = \Delta t^2 M^{-1} \left[f_n - K w_n - C \left(\frac{w_n - w_{n-1}}{\Delta t} \right) \right] + 2 w_n - w_{n-1}, \tag{38}$$

where $C = \alpha_R M + \beta_R K$ is the Rayleigh damping matrix. This formulation includes both stiffness and mass proportional damping contributions.

We proceed as follows:

Assume initial displacement w_0, and velocity $\dot{w}_0$. Compute the initial acceleration from the equation of motion:

$$\ddot{w}_0 = M^{-1} \left(f_0 - C \dot{w}_0 - K w_0 \right). \tag{39}$$

Using a Taylor expansion for a special start (explicit scheme has no w_{-1}):

$$w_1 = w_0 + \Delta t \dot{w}_0 + \frac{1}{2} \Delta t^2 \ddot{w}_0. \tag{40}$$

If $\dot{w}_0 = 0$, this simplifies to:

$$w_1 = w_0 + \frac{1}{2} \Delta t^2 \ddot{w}_0. \tag{41}$$

For $n \geq 1$, we use the central difference recurrence:

$$w_{n+1} = \Delta t^2 M^{-1} \left[f_n - K w_n - C \left(\frac{w_n - w_{n-1}}{\Delta t} \right) \right] + 2 w_n - w_{n-1}. \tag{42}$$

Velocity and acceleration can be recovered at each time step using finite differences:

$$\dot{w}_n = \frac{w_{n+1} - w_{n-1}}{2 \Delta t}, \tag{43}$$

$$\ddot{w}_n = \frac{w_{n+1} - 2 w_n + w_{n-1}}{\Delta t^2}. \tag{44}$$

It is important to note that explicit time integration methods are conditionally stable and require careful selection of the time step to maintain numerical stability. The stability criterion is typically governed by the following inequality: :

$$\Delta t < \frac{2}{\omega_{\max}}, \tag{45}$$

where $\omega_{\max}$ represents the highest natural frequency of the system. This value is directly associated with the largest eigenvalue of the matrix $M^{-1} K$, where M and K are the global mass and stiffness matrices, respectively. If this condition is not met, the numerical solution may become unstable and diverge.

8 Control Force Implementation

Control forces are introduced at selected nodes along the beam to actively influence both axial and bending responses. As shown in Figure 2, each actuator applies a concentrated axial force F_{control} along the beam axis and induces a moment M_{control} by acting off-center relative to the beam's neutral axis. These control actions are designed to counteract undesirable vibrations or deflections and are computed based on real-time deviations from a desired trajectory or shape.

The control force is defined as

$$F_{\text{control}} = K_{\text{ctrl}} \cdot e(t), \tag{46}$$

where K_{ctrl} represents a control gain and $e(t)$ is the error between the measured and desired beam states. The corresponding moment is defined as

$$M_{\text{control}} = F_{\text{control}} \cdot \frac{h}{2}, \tag{47}$$

where h is the beam thickness, and the factor $h/2$ accounts for the distance from the neutral axis to the surface where the actuator applies the force.

In the finite element framework, F_{control} is applied to the axial degree of freedom u at the control node, while M_{control} contributes to the rotational degree of freedom θ at the same node. These contributions are inserted directly into the global force vector in the positions corresponding to those degrees of freedom.

The modified equation of motion, including the control forces and moments, becomes

$$M\ddot{W} + C\dot{W} + KW = F_{\text{control}}(t), \tag{48}$$

where M, C, and K are the assembled global mass, damping, and stiffness matrices, and $F_{\text{control}}(t)$ contains the time-varying control inputs mapped to the appropriate DOFs.

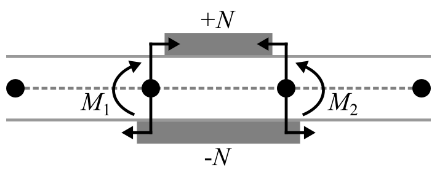

Figure 2: Control force F_{control} and generated moment M_{control} applied at the control node.