Fall 2025 ME751 Final Project Report

University of Wisconsin-Madison

Deformable Multibody System

Ganesh Arivoli
Department of Mechanical Engineering

December 10, 2025

Abstract

This project presents a finite element simulation of a clamped cantilever beam subjected to dynamic loading. The beam is modeled using B3-24 elements (2 nodes per element, 4 unknowns per node) and discretized into multiple elements to study convergence behavior. BDF-1 time integration with Newton-Raphson iteration is used to solve the nonlinear equations of motion. Results demonstrate mesh convergence, with tip deflection converging from -1.79 mm (1 element) to -2.92 mm (16 elements). Computational cost scales quadratically with element count, as expected for finite element methods.

Contents

1 General information

- Home department: Mechanical Engineering
- Current status: MS student

2 Problem statement

This project simulates the dynamic response of a clamped cantilever beam subjected to a time-dependent tip force. The beam is modeled using finite elements with B3-24 elements (2 nodes per element, 4 unknowns per node). The root end is clamped (fixed position and gradients), while a transient force is applied at the tip over a 10-second simulation period.

The beam and simulation parameters are as follows:

- Total length: $L = 0.5$ m
- Cross-section: square, $W = H = 0.003$ m
- Material: steel
  - Young's modulus: $E = 2.0 \times 10^{11}$ Pa
  - Poisson's ratio: $\nu = 0.3$
  - Density: $\rho = 7700$ kg/m^3
- Tip force: $F_z(t) = -1.0 + \cos(20\pi t)$ N for $t \in [0, 0.05]$ s, zero otherwise
- Material model: Saint-Venant-Kirchhoff (SVK) hyperelastic constitutive law
- Number of elements tested: $n \in \{1, 2, 3, 4, 8, 16\}$
- Mesh: Constant total length L, element size decreases as n increases
- Time step: $h = 5 \times 10^{-4}$ s

3 Numerical solution description

The equations of motion are solved using BDF-1 (Backward Differentiation Formula of order 1) time integration. At each time step, the position and velocity are updated as:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + h\mathbf{v}_{n-1} + h^2\mathbf{a}_n \tag{1}$$

$$\mathbf{v}_n = \mathbf{v}_{n-1} + h\mathbf{a}_n \tag{2}$$

where h is the time step and $\mathbf{a}_n$ is the acceleration at time step n.

The clamped boundary condition at the root end is enforced using Lagrange multipliers $\boldsymbol{\lambda}_n$, leading to an augmented system that solves for both acceleration $\mathbf{a}_n$ and constraint forces $\boldsymbol{\lambda}_n$:

$$\mathbf{M}\mathbf{a}_n + \mathbf{C}^T\boldsymbol{\lambda}_n + \mathbf{f}_{\text{int}}(\mathbf{x}_n) + \mathbf{f}_{\text{ext}}(t) + \mathbf{f}_{\text{gravity}} = \mathbf{0} \tag{3}$$

$$\mathbf{c}(\mathbf{x}_n) = \mathbf{0} \tag{4}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the constraint Jacobian matrix, $\mathbf{f}_{\text{int}}$ is the internal force vector, $\mathbf{f}_{\text{ext}}$ is the external force vector, and $\mathbf{f}_{\text{gravity}}$ is the gravity force vector.

The nonlinear system is solved using Newton-Raphson iteration. At each iteration k, the augmented residual vector $\mathbf{R}^{(k)}$ is computed as:

$$\mathbf{R}^{(k)}_{\text{dynamics}} = \mathbf{M}\mathbf{a}^{(k)}_n + \mathbf{C}^T\boldsymbol{\lambda}^{(k)}_n + \mathbf{f}_{\text{int}}(\mathbf{x}^{(k)}_n) - \mathbf{f}_{\text{gravity}} - \mathbf{f}_{\text{ext}}(t) \tag{5}$$

$$\mathbf{R}^{(k)}_{\text{constraint}} = \mathbf{c}(\mathbf{x}^{(k)}_n)/h^2 \tag{6}$$

$$\mathbf{R}^{(k)} = \begin{bmatrix} \mathbf{R}^{(k)}_{\text{dynamics}} \\ \mathbf{R}^{(k)}_{\text{constraint}} \end{bmatrix} \tag{7}$$

where $\mathbf{x}^{(k)}_n = \mathbf{x}_{n-1} + h\mathbf{v}_{n-1} + h^2\mathbf{a}^{(k)}_n$ is the position computed from the current acceleration guess. The constraint residual is scaled by h^2 to maintain consistent units in the augmented system. The augmented Jacobian matrix has the block structure:

$$\mathbf{J}^{(k)} = \begin{bmatrix} \mathbf{M} + h^2\mathbf{K}^{(k)} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \tag{8}$$

where $\mathbf{K}^{(k)} = \partial\mathbf{f}_{\text{int}}/\partial\mathbf{x}$ evaluated at $\mathbf{x}^{(k)}_n$ is the tangent stiffness matrix. The tangent stiffness is computed using forward finite differences with perturbation $\epsilon = 10^{-6}$. At each iteration, the linear system $\mathbf{J}^{(k)}\boldsymbol{\Delta}^{(k)} = -\mathbf{R}^{(k)}$ is solved to obtain the update $\boldsymbol{\Delta}^{(k)} = [\boldsymbol{\Delta}^{(k)}_{\mathbf{a}}; \boldsymbol{\Delta}^{(k)}_{\boldsymbol{\lambda}}]$, and the solution is updated as $\mathbf{a}^{(k+1)}_n = \mathbf{a}^{(k)}_n + \boldsymbol{\Delta}^{(k)}_{\mathbf{a}}$ and $\boldsymbol{\lambda}^{(k+1)}_n = \boldsymbol{\lambda}^{(k)}_n + \boldsymbol{\Delta}^{(k)}_{\boldsymbol{\lambda}}$. The iteration continues until $||\mathbf{R}^{(k)}|| < 10^{-6}$ or a maximum of 100 iterations is reached. Internal forces are computed via Gauss-Legendre quadrature with 16 integration points per element (4 points along the beam axis, 2 points in each transverse direction).

4 Overview of results

Simulations were performed for element configurations ranging from 1 to 16 elements. The tip trajectory was computed for each case, and the results demonstrate convergence behavior as the mesh is refined.

Key findings from the simulations:

- **Convergence:** Tip z-position converges from -1.79 mm (1 element) to -2.92 mm (16 elements). The difference between 8 and 16 elements is only 0.012 mm, indicating good mesh convergence.

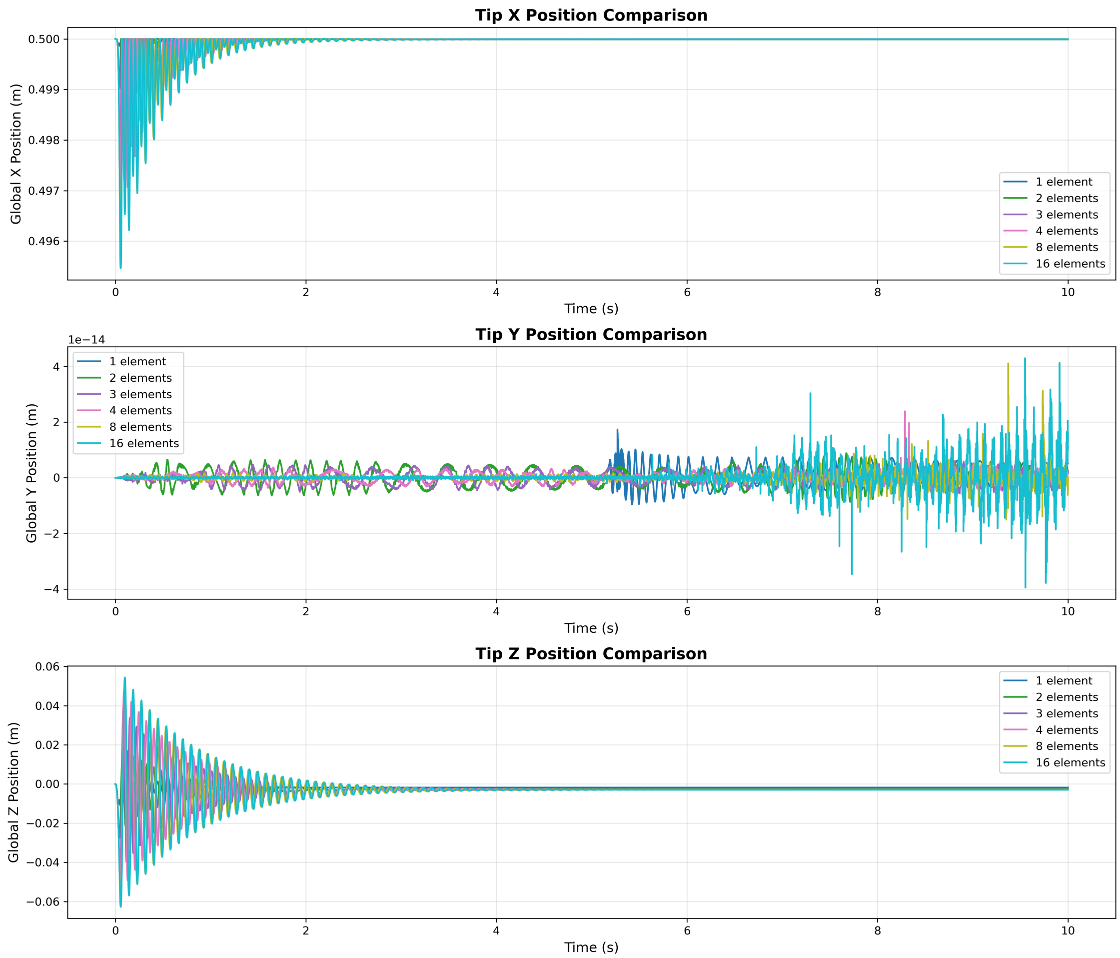

Figure 1: Tip trajectory comparison for different element configurations showing X, Y, and Z positions over time.

No. of Elements	Time (s)	Time (min)	Tip x (mm)	Tip y (mm)	Tip z (mm)
1	29.31	0.49	499.9959	0.0000	-1.7919
2	145.73	2.43	499.9920	0.0000	-2.6017
3	360.05	6.00	499.9909	0.0000	-2.7945
4	637.36	10.62	499.9906	-0.0000	-2.8553
8	2499.51	41.66	499.9903	-0.0000	-2.9069
16	7459.29	124.32	499.9903	0.0000	-2.9190

Table 1: Simulation timing and tip positions at $t = 10$ s for different element configurations.

- **Trajectory behavior:** Tip trajectories are smooth and physically reasonable (see Figure 1). The beam deflects downward under gravity and the transient tip force, then oscillates and settles. The X and Z positions show clear convergence with mesh refinement, while Y position remains near zero as expected.
- **Damping:** The simulation includes only numerical damping from the BDF-1 time

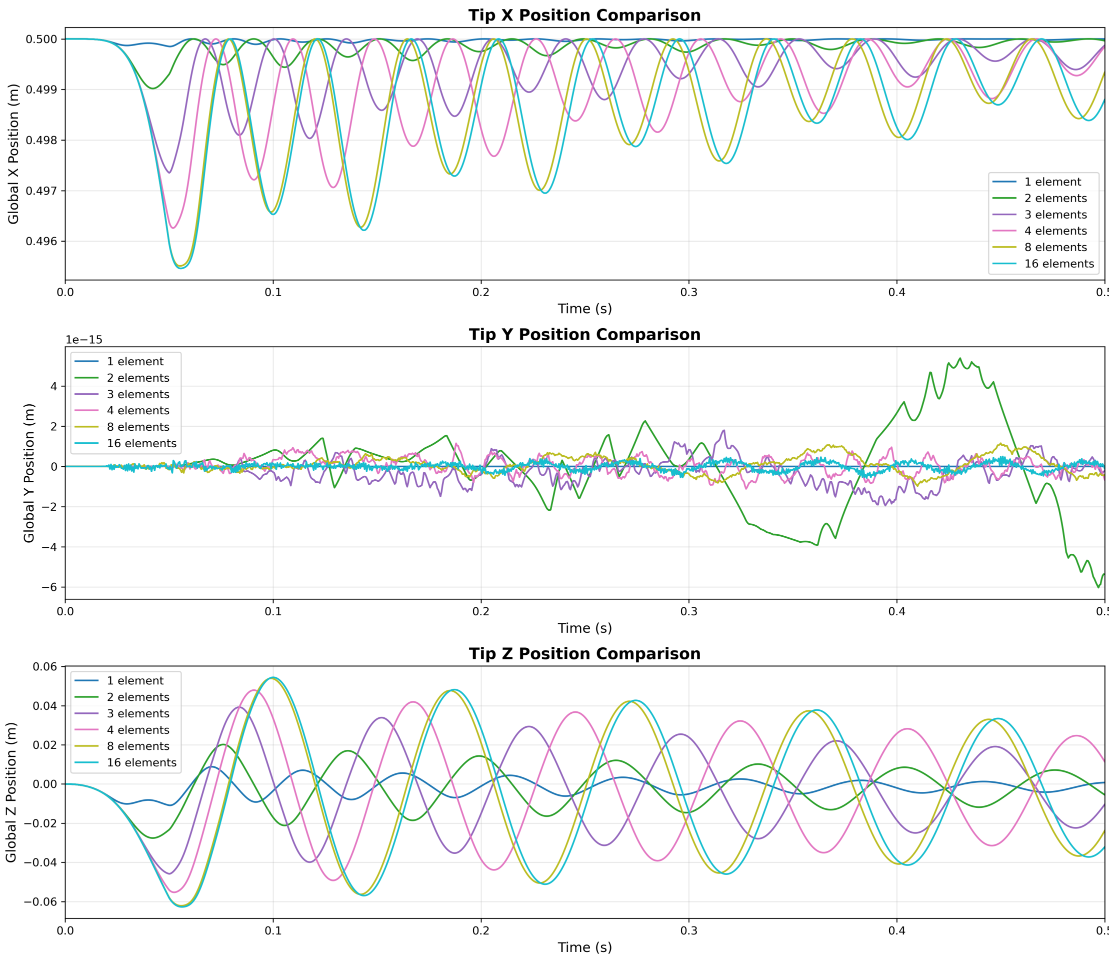

Figure 2: Tip trajectory comparison for different element configurations showing X, Y, and Z positions over time, zoomed to $t \leq 0.5$ s to show initial transient response.

integration scheme, with no structural or material damping included in the model.

- **Computational cost:** Simulation time scales approximately quadratically with element count, from 29 seconds (1 element) to 124 minutes (16 elements). This is expected due to increasing DOFs and Newton-Raphson iteration cost.
- **Physical consistency:** Results show expected behavior: downward deflection due to gravity, response to transient force, and proper constraint enforcement at the clamped end.

5 Deliverables

The following deliverables are provided:

- **Main code:** `main-mbd-deformable.py` - Complete simulation code for the deformable beam system

- **Output files:** CSV files with tip positions and PNG plots for each element configuration

5.1 Running the Code

The code requires Python 3 with NumPy and Matplotlib. To run a simulation:

```
python3 main-mbd-deformable.py <n_elements>
```

where <n_elements> is the number of elements. The script automatically generates a CSV file with tip positions and a PNG plot showing the tip trajectory.

5.2 Code Optimizations

The implementation includes several performance optimizations to enable efficient simulation of multi-element systems:

- **Precomputed matrices:** The B-matrix inverse (`B_INV`) and shape function evaluations at all 16 Gauss points (`GAUSS_16`) are precomputed once during element initialization, avoiding repeated computation during force evaluation.
- **Constant matrix precomputation:** The mass matrix and gravity force vector are computed once before time stepping, as they remain constant throughout.
- **Vectorized operations:** Internal force computation uses NumPy's `einsum` for efficient vectorized tensor operations over all Gauss points simultaneously, eliminating element-wise loops in the critical force evaluation path.
- **Initial guess strategy:** Newton-Raphson iterations are seeded with converged values from the previous time step, reducing the number of iterations required for convergence.

These optimizations enable the simulation of 16-element systems to complete in reasonable time despite the quadratic scaling of computational cost with element count.

6 Conclusions and Future Work

The deformable beam simulation successfully demonstrates convergence behavior with mesh refinement, and future work could explore more complex geometry like nets, expanding to shell elements and more advanced adaptive mesh refinement strategies.

References

1. ME 751 Course Slides and Notes, Fall 2025, University of Wisconsin-Madison

Acknowledgments

Code development assistance was provided by Claude AI Assistant (Anthropic).