Homework 1
UCLA MAE 263F, Fall 2024
Mechanics of Flexible Structures and Soft Robots

He Kai Lim

Abstract— **This report strives to provide all deliverables for Homework 1 in partial fulfilment of the requirements for UCLA MAE 263F, Fall 2024 (Mechanics of Flexible Structures and Soft Robots).**

I. Introduction

Energy methods of node-based simulation are useful, and yield convergence in results when conditions are met. These conditions include using sufficiently small step sizes and sufficient density of nodes. Furthermore, we find that implicit methods solved numerically with Newton-Raphson methods are more computationally efficient than explicit methods for the same convergence criteria. Lastly, we note that even in small deformation, a discretized approach to modelling time-history deformation enables more accurate results since they capture nonlinear behavior that is not simply described with conventional theory (e.g. Euler Bernoulli or Timoshenko equations).

II. Assignment 1: from Section 4.2

Three rigid spheres of radii R_1, R_2, R_3, are constructed as discrete elements to simulate an elastic beam that is falling under gravity in a viscous fluid. The six equations of motion that fully describe their motion are:

$$m_1\ddot{x_1} = -(6\pi\mu R_1)\dot{x_1} - \frac{\partial E^{elastic}}{\partial x_1} \quad (1.1)$$

$$m_1\ddot{y_1} = -W_1 - (6\pi\mu R_1)\dot{y_1} - \frac{\partial E^{elastic}}{\partial y_1} \quad (1.2)$$

$$m_2\ddot{x_2} = -(6\pi\mu R_2)\dot{x_2} - \frac{\partial E^{elastic}}{\partial x_2} \quad (1.3)$$

$$m_2\ddot{y_2} = -W_2 - (6\pi\mu R_2)\dot{y_2} - \frac{\partial E^{elastic}}{\partial y_2} \quad (1.4)$$

$$m_3\ddot{x_3} = -(6\pi\mu R_3)\dot{x_3} - \frac{\partial E^{elastic}}{\partial x_3} \quad (1.5)$$

$$m_3\ddot{y_3} = -W_3 - (6\pi\mu R_3)\dot{y_3} - \frac{\partial E^{elastic}}{\partial y_3} \quad (1.6)$$

(equations 4.14 and 4.15 are taken from the course notes).

Correspondingly, the 6x6 Jacobian matrix used to solve the simulation step (under an implicit Newton-Raphson method) is as follows:

$$\frac{\partial^2 E^{elastic}}{\partial x\,\partial y} = \frac{\partial E_1^s}{\partial x_1\,\partial y_1} + \frac{\partial E_2^s}{\partial x_1\,\partial y_1} + \frac{\partial E^b}{\partial x_1\,\partial y_1} + \frac{\partial E_1^s}{\partial x_2\,\partial y_2} + \frac{\partial E_2^s}{\partial x_2\,\partial y_2} + \frac{\partial E^b}{\partial x_2\,\partial y_2} + \frac{\partial E_1^s}{\partial x_3\,\partial y_3} + \frac{\partial E_2^s}{\partial x_3\,\partial y_3} + \frac{\partial E^b}{\partial x_3\,\partial y_3}, \quad (1.7)$$

where:

$$E_1^s = \frac{1}{2}EA\left(1 - \frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}{\Delta l}\right)^2 \Delta l \quad (1.8)$$

$$E_2^s = \frac{1}{2}EA\left(1 - \frac{\sqrt{(x_3-x_2)^2+(y_3-y_2)^2}}{\Delta l}\right)^2 \Delta l \quad (1.9)$$

$$E^b = \frac{1}{2}\frac{EI}{\Delta l}\left(2\,tan\left(\frac{\theta}{2}\right)\right) \quad (1.10)$$

Derivation of this Jacobian comes from the partial derivatives of $E^{elastic}$ with respect to q the DOF vector, because $E^{elastic}$ represents the potential energy stored in the system, and so the rate of change of this energy with respect to every DOF is conceptually an indication of the deformed state of this system when the net sum of external forces yields a nonzero result (meaning energy is being stored or released from the compliant body).

We note that our explicit algorithm is computationally expensive and required 99mins 35.7s to solve in python 3.10.

Algorithm 1 Explicit solution to time-varied deformation:

Given the equation of motion:

f = m * (q_new - q_old) / dt**2 - m * u_old / dt - (Fb + Fs + W + Fv)

Directly solve for q_new at a sufficiently small timestep dt

q_new = q_old + (dt**2 / m) * (Fb+Fs+W+Fv + m*(u_old/dt))

Algorithm 2 Implicit solution to time-varied deformation:

He Kai Lim is with the Department of Mechanical and Aerospace Engineering, University of California Los Angeles, Los Angeles, CA 90095 USA (e-mail: limhekai@ucla.edu).

This report has supplementary downloadable material available at https://github.com/he-is-kai/MAE263F_UCLA_Fall2024, provided by the authors.

Given the equation of motion:

f = m * (q_new - q_old) / dt**2 - m * u_old / dt - (Fb + Fs + W + Fv)

Apply Newton Raphson approach to solve for q_new

Find the Jacobian of the equation of motion

Guess q_new and compute the error in q_new
terminate loop when error in q_new is within tolerance

We now commence the following deliverables of the homework.

Write a solver that simulates the position and the velocity of the sphere as a function of time (between $0 \leqslant t \leqslant 10$ seconds) implicitly and explicitly. Use $\Delta t = 10^{-2}$ s for the implicit simulation and $\Delta t = 10^{-5}$ s for the explicit one.

1. *Show the shape of the structure at t = {0, 0.01, 0.05, 0.10, 1.0, 10.0} s. Plot the position and velocity (along y-axis) of R2 as a function of time*

First we examine the results of implicitly solved simulation.

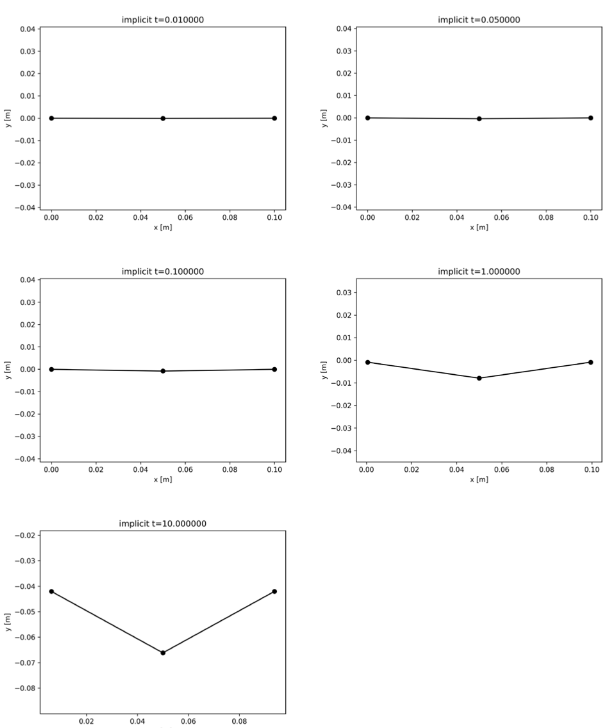


Fig. 1. Shape of the structure for implicitly solved simulation with 3 nodes, at t = {0.01, 0.05, 0.1, 1.0, 10}

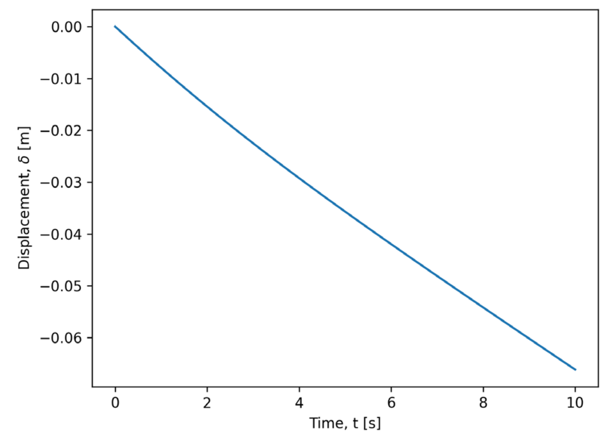


Fig. 2. Position of R2 along the y-axis for the implicitly solved simulation with 3 nodes.

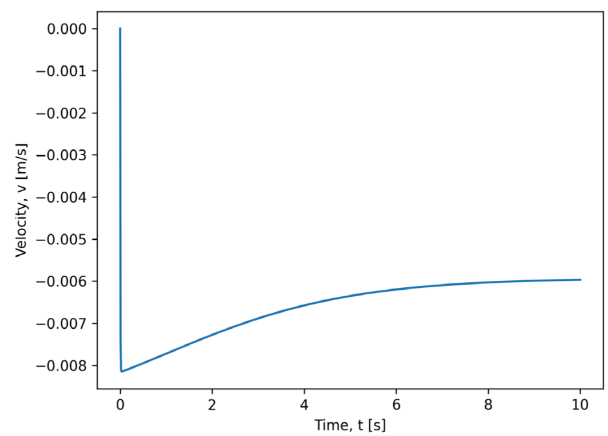
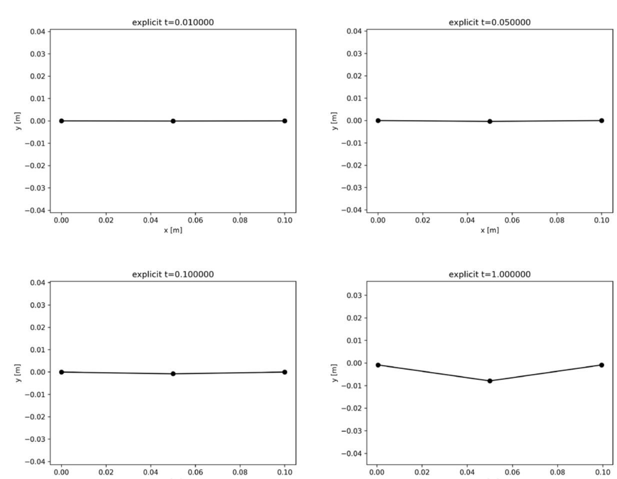


Fig. 3. Velocity of R2 along the y-axis for the implicitly solved simulation with 3 nodes.

Now, we examine the results of the explicitly solved simulation.



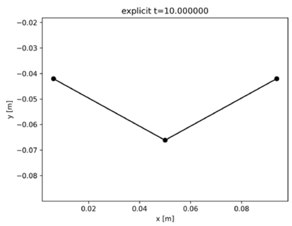


Fig. 4. Shape of the structure for explicitly solved simulation with 3 nodes, at t = {0.01, 0.05, 0.1, 1.0, 10}

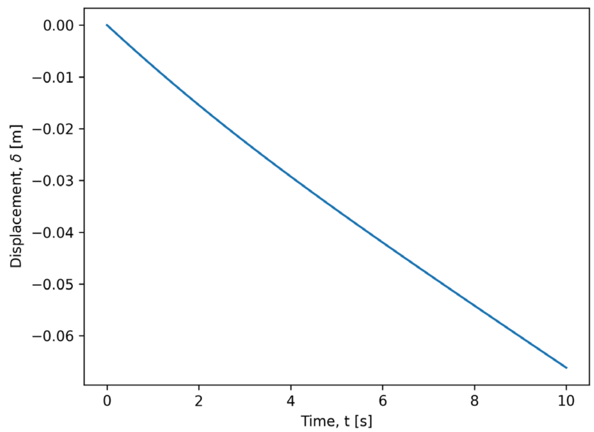


Fig. 5. Position of R2 along the y-axis for the explicitly solved simulation with 3 nodes.

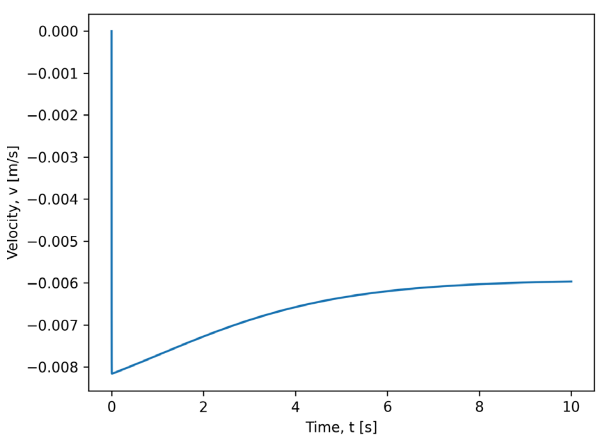


Fig. 6. Velocity of R2 along the y-axis for the explicitly solved simulation with 3 nodes.

2. *What is the terminal velocity (along y-axis) of this system?*

We obtain the terminal velocity as the last value of velocity found in the 10-second simulation.

Implicit Solution: terminal velocity = -0.00596573 m/s.

Explicit Solution: terminal velocity = -0.00596569m/s.

3. *What happens to the turning angle if all the radii (R1,R2,R3) are the same? Does your simulation agree with your intuition?*

Turning angle stays near zero. The single spike at t~= occurs at 5 e-19 which is still negligible. This is shown in figure X.

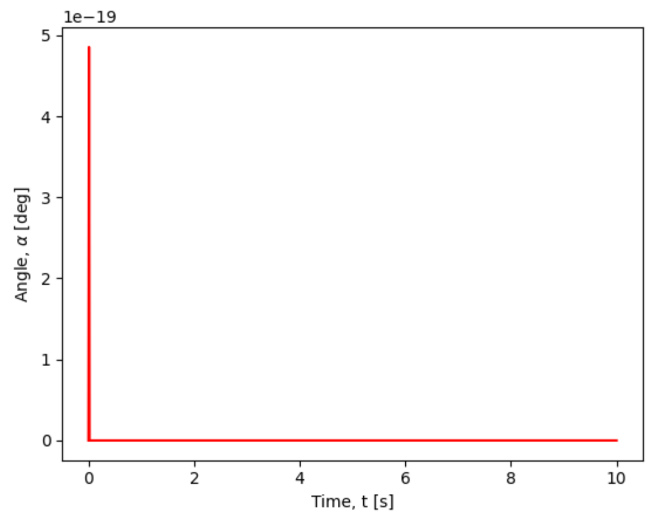


Fig. 7. The turning angle for all nodes having same radii, R1 = R2 = R3 = 0.005m.

Yes, this is expected with intuition.

Turning angle changes when there is a nonuniformly distributed mass, and if there are viscous effects on the boundary of the object that are asymmetric to the center.

In our simulation, we compute mass as a function of node size, and we calculate viscous effects are correlation to node size as well. Since each node is constant in size, we observe symmetry in forces across all 3 nodes, and so it is expected that the turning angle is 0, meaning that the beam maintains horizontal throughout its descent.

4. *Try changing the time step size (Δt), particularly for your explicit simulation, and use the observation to elaborate the benefits and drawbacks of the explicit and implicit approach?*

Starting with the easiest case of the implicit solution, we see that increasing (Δt) to 1.0 seconds (from original 1e-2) will marginally increase the error for each newton-raphson convergence loop. The final results, however, are pretty close to original.

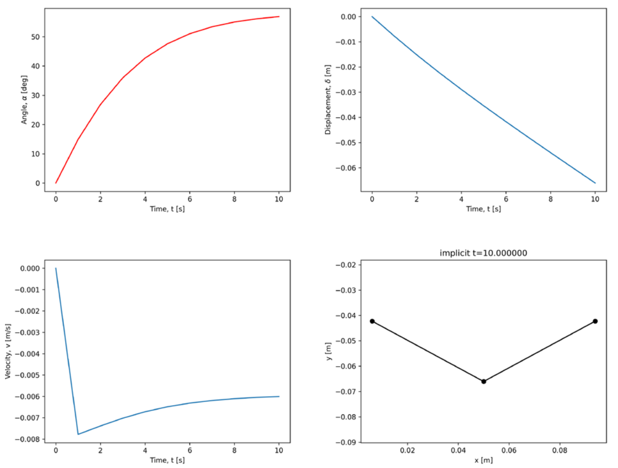


Fig. 8. The results of implicit simulation with large (Δt) = 1.0 seconds.

Explicit (Δt=1.0e0) terminal velocity = -0.006008498519768937 m/s.

Explicit (Δt=1.0e-2) terminal velocity = -0.005966069474258595m/s.

Explicit (Δt=1.0e-3) terminal velocity = - -0.005965730463602936m/s.

Now, we examine the results of the explicitly solved simulation.

Larger step sizes (Δt) yielded in non-convergence on my computer. We were thus only able to plot for slightly smaller step sizes considering the time constraints of the project.

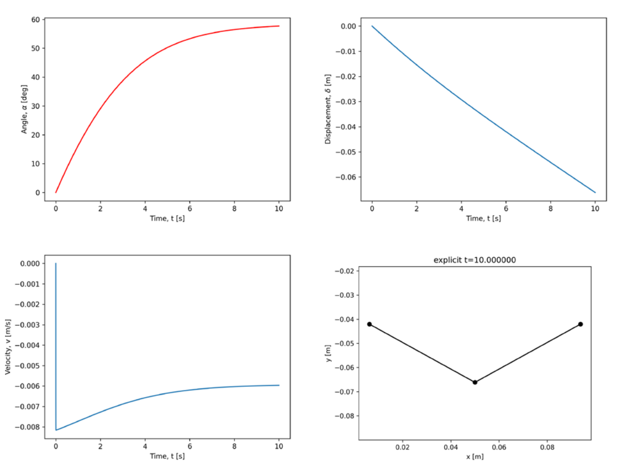


Fig. 9. The results of explicit simulation with smaller (Δt) = 0.8e-5 seconds.

Explicit (Δt=1.0e-5) terminal velocity = -0.005965693018750161 m/s.

Explicit (Δt=0.8e-5) terminal velocity = -0.005965693015974605 m/s.

The results match up to 11 decimal places.

Between the implicit and explicit approaches, we observe that the explicit method requires much more computing power to obtain any solution at all, because large dt values simply have no continuity between time steps in the explicit approach. However, the explicit approach does seem less sensitive to changes in dt, though it was not very discernible with available computing power.
In contrast, the implicit approach is more sensitive to dt closeness, but requires far less compute power.

On the whole, it is still faster to use the implicit approach, and validate it by converging to a sufficiently small dt for accurate results, as opposed to using the explicit approach.

.

III. Assignment 2: from Section 4.3

Each simulation case required 4min 20 seconds to complete computation. In total, this assignment required XX time for all cases to generate the plots in deliverable 3.

Write a solver that simulates the system of the sphere as a function of time (between $0 \leq t \leq 50$ seconds) implicitly with $\Delta t = 10{-}2$ s and $N = 21$.

1. *Include two plots showing the vertical position and velocity of the middle node with time. What is the terminal velocity?*

These plots are included thusly.

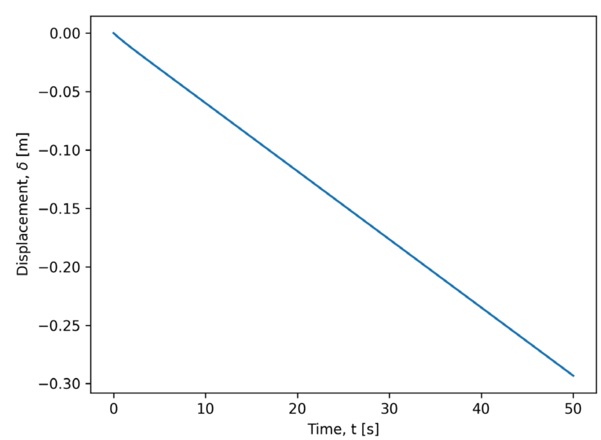


Fig. 10. The position of the middle node with time.

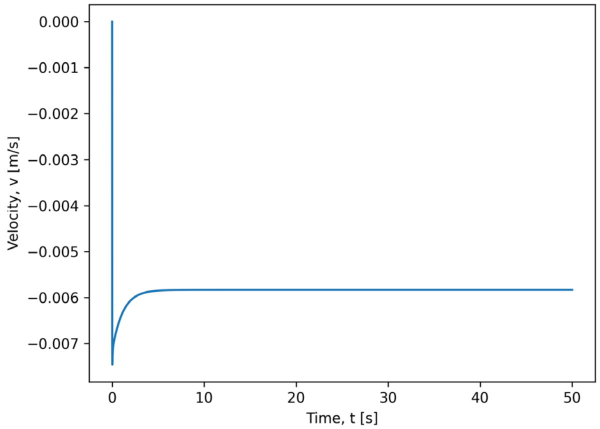


Fig. 11. The velocity of the middle node with time.

The terminal velocity is -0.005834 m/s.

2. *Include the final deformed shape of the beam*

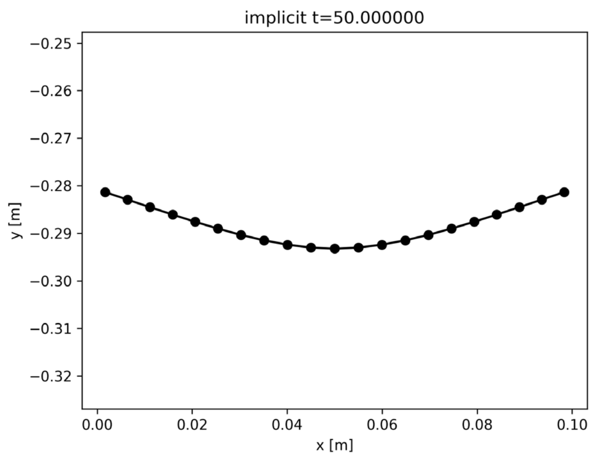


Fig. 12. The final deformed shape of the beam modelled with 21 nodes across a 50 second simulated fall time.

3. *Discuss the significance of spatial discretization (i.e. the number of nodes, N)and temporal discretization (i.e. time step size, Δt). Any simulation should be sufficiently discretized such that the quantifiable metrics, e.g. terminal velocity, do not vary much if N is increased and Δt is decreased. Include plots of terminal velocity vs. the number of nodes and terminal velocity vs. the time step size.*

[text]

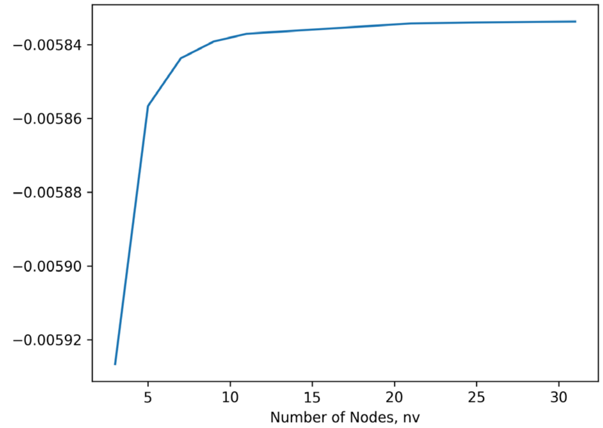


Fig. 13. Terminal Velocity vs Number of Nodes. Runtime to generate this data was 37min 27.3s.

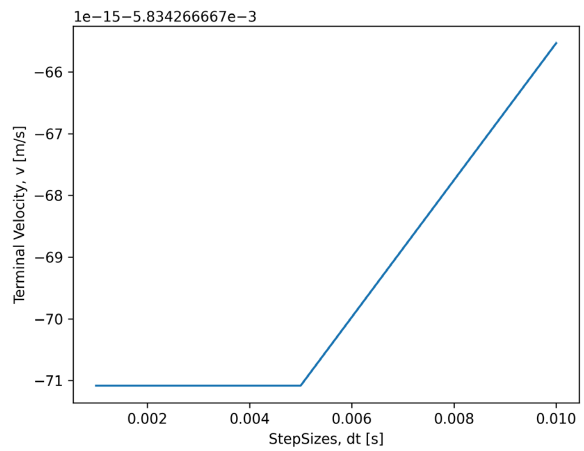


Fig. 14. Terminal Velocity vs Step Sizes. Runtime to generate this data was 52min 57.7s.

We observe that increasing N above (eyeballed) 21 yields convergence in terminal velocity calculations. We observe that step size smaller than 0.005s yields convergence as well.

The significance is that spatial discretization is more important to yield convergence. For N=21, the marginal overhead of adding 2 more nodes will yield 2/21 = ~ 10% more computing time, whereas at dt = 0.005, a smaller step size of 0.004 will yield ~ 20% more compute time.

All this assumes that steady state convergence is equivalent to accuracy, which is not true. In modelling a continuous object with discrete nodes, naturally, our intuition also confirms that having more nodes is important that having small time steps to find the dynamic (transient) deformation accuracy.

IV. ASSIGNMENT 3: FROM SECTION 4.4

Write a solver that simulates the beam as a function of time (between $0 \le t \le 1$ seconds) implicitly. Use $\Delta t = 10{-}2$ s for the implicit simulation and N = 50.

1. *Plot the maximum vertical displacement, ymax, of the beam as a function of time. Depending on your coordinate system, ymax may be negative. Does ymax eventually reach a steady value? Examine the accuracy of your simulation against the theoretical prediction from Euler beam theory:*

$$y_{max} = \frac{Pc\left(L^2-c^2\right)^{1.5}}{y\sqrt{3}EIl} \qquad \text{where } c = \min(d, l-d) \quad (4.21)$$

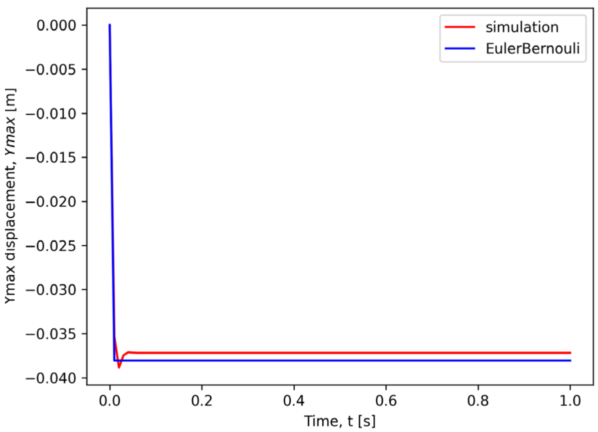


Fig. 15. Ymax vs time for the elastic beam under smaller 2000N load.

It does reach a steady state value of:

Simulated: -0.03717 m

Euler Bernouli: -0.03804 m

The simulation results match very closely. The divergence lies in the simulation being able to account for history-dependent pathways for how energy dissipates and is stored in the elastic beam, to be manifested as deformation. We expect simulation to be more accurate, though the deformations are all small to begin with

2. *What is the benefit of your simulation over the predictions from beam theory? To address this, consider a higher load P = 20000 such that the beam undergoes large deformation. Compare the simulated result against the prediction from beam theory in Eq. 4.21. Euler beam theory is only valid for small deformation whereas your simulation, if done correctly, should be able to handle large deformation. Optionally, you can make a plot of P vs. ymax using data from both simulation and beam theory and quantify the value of P where the two solutions begin to diverge*

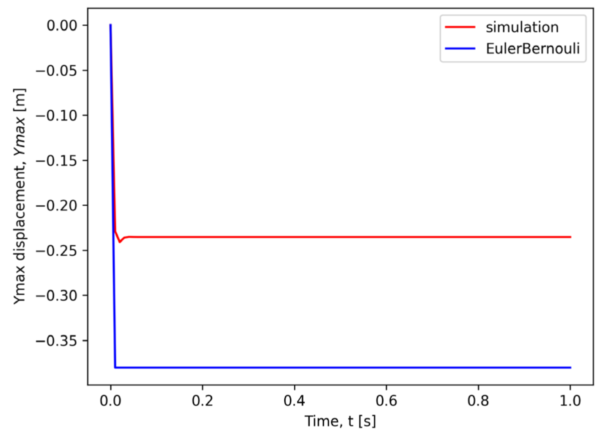


Fig. 16. Ymax vs time for the elastic beam under larger 20000N load.

Simulated: -0.23427917 m

Euler Bernouli: -0.38044916 m

The simulated result shows a smaller deformation compared to Euler Bernouli.

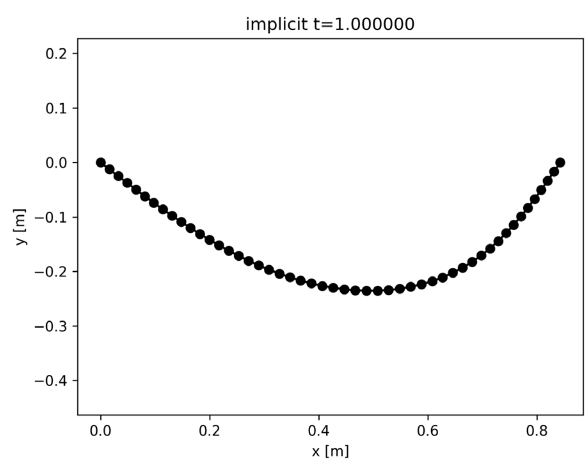


Fig. 17. Deformation of elastic beam under larger 20000N load.

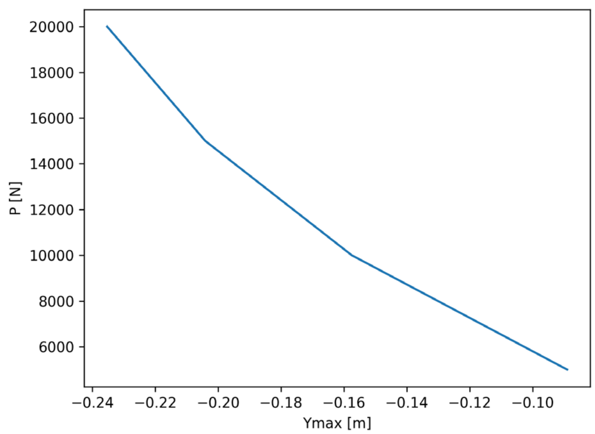


Fig. 18. Simulated P vs Ymax of the elastic beam deformation.

ACKNOWLEDGMENT

Materials used in this report are taken from the course UCLA MAE 263F, Fall 2024.

REFERENCES

[1] M. Khalid Jawed, Singmin Lim, "Discrete Simulation of Slender Structures," UCLA Fall 2024 MAE 263F