MAE 263F HOMEWORK 1

Yuan Hung Lo

I. RIGID SPHERE FALLING SIMULATION

A. Shape of Structure at Specified Time

As Shown in Fig. 1, the three nodes slowly drift down the fluid as time goes by. Since the middle node has the largest weight, it is less affected by the viscous force of the fluid. Therefore, sinking faster and drag the smaller and slower nodes behind it, leading to the bended shape.

B. Y-axis Position and Velocity of Middle Node

Fig. 2 plotted the vertical position and velocity of the node in the middle. From the position graph, you can see a slight curve at the beginning, and slowly change into a straight line. The velocity graph also shows the vertical velocity slowly reaches a steady value. These two phenomena indicate that the middle node is gradually reaching a steady state and falling at terminal velocity where its weight and viscous force equilibrate.

C. Terminal y-axis Velocity of Middle Node

As indicated in Fig. 2, the middle node reaches a state in around 9 seconds and has a terminal velocity of 0.005966 m/s.

D. Turning Angle of Identical Radius Nodes

From previous sections we know that due to the geometry of the spheres, the node in the middle bends. Reaching a maximum turning angle of 1.01 radiant according to MATLAB simulation. However, once we set the radius of the three spheres to identical value, in this case r1 = r2 = r3 = 0.025, all three nodes would experience the same amount of viscous force as well as weight. As a result, the three nodes would sink at the same speed and maintain horizontally inline. With the extra weight, all three nodes would also sink deeper in the same time frame compared to previous cases. The result can be seen in Fig. 3.

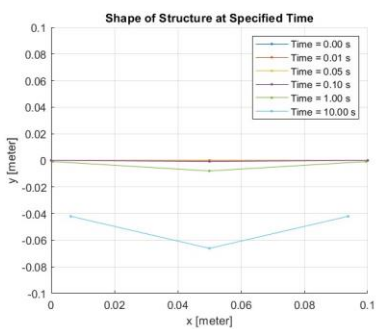

Figure 1. Structure shape at different time.

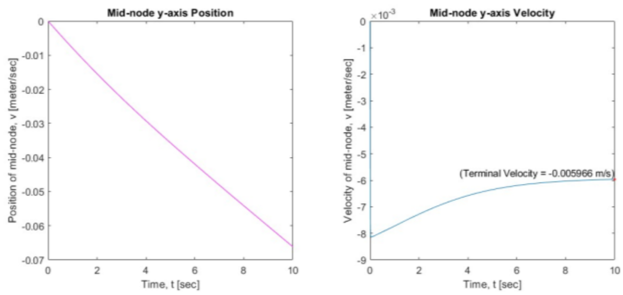

Figure 2. Vertical position and velocity of middle node.

E. Effect of Different Time Step Size

Generally, with smaller time steps the simulation can reach finer and more accurate results. With larger time step, while some details could be neglected causing errors in result, the runtime can be greatly reduced. By simulating the implicit method with time steps of 0.001, 0.01 and 0.1 second, it is observed that all of them produced identical results, with the same terminal velocity: 0.05966 m/s. The runtime of larger time step simulation also improved slightly. Time step dt of 0.001 second finished the simulation in 1.12 seconds, dt = 0.01 finished in 0.35 seconds, and dt = 0.1 finished in 0.22 seconds. Demonstrating that the model we create is robust and efficient.

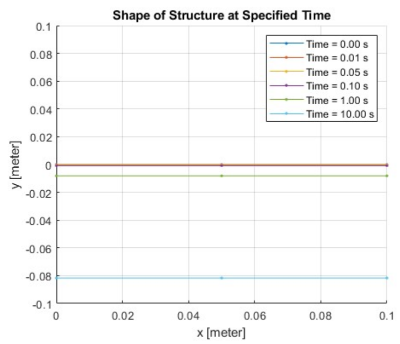

Figure 3. Shape of falling uniform sheres.

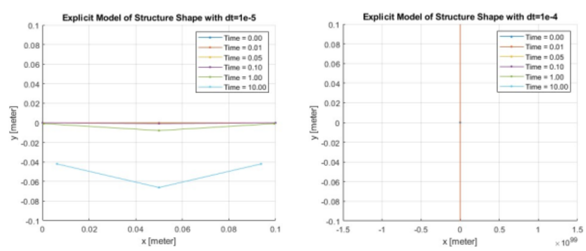

Figure 4. Shape of explicit model with different time step.

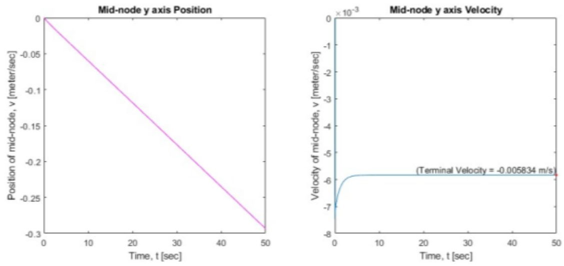

Figure 5. Position and velocity of general simulation.

On the other hand, the explicit method's performance is drastically affected by the change in time step. Originally with dt = 1e-5, the simulation finished in 30.09 seconds, which is already significantly slower than the provided implicit method. To lower the run time, it would seem intuitive to increase the time step dt to decrease the number of steps and speed up the calculation. Switching to dt = 1e-4, the run time decreased to 3.15 seconds. However, the simulation result is completely off, meaning that the solutions the model finds are incorrect. The comparison between the two configurations is shown in Fig. 4, clearly indicating the false result due to inaccurate solution.

F. *Comparison of Implicit and Explicit Method*

While the explicit method requires far less lines of code and is easy to compute, its rough estimation characteristics made it inconsistent. Requiring user to spend more time using finer time step to make up for the inaccurate equation to acquire the correct result. The implicit method on the other hand is far more reliable. Despite its relatively complex code, it can accurately complete the simulation in a short amount of time.

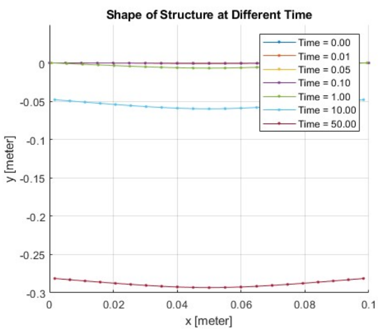

Figure 6. Shape and position of generalized falling spheres.

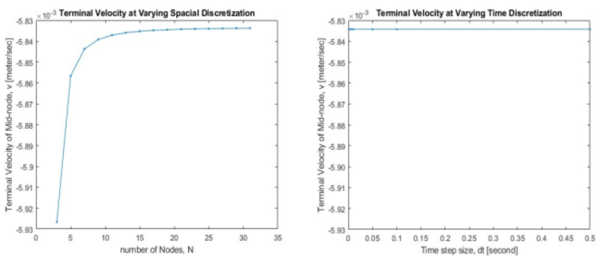

Figure 7. Example of a figure caption.

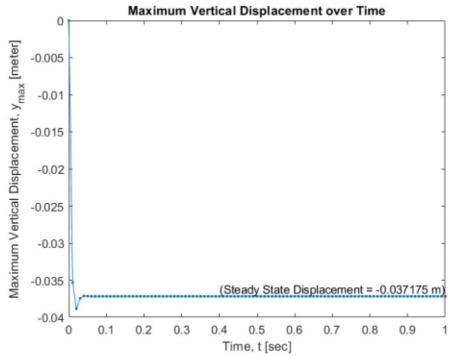

Figure 8. Maximum vertical displacement over time.

II. GENERALIZED SIMULATION

A. *Y-axis Position and Velocity of Middle Node*

With longer time frame, we can clearly see the motion reaching equilibrium. The position plot Fig. 5 has constant gradient meaning its moving at constant speed in most of the simulation. This is also observable in the velocity graph where it reached a steady value at around T = 7, where we can see the terminal velocity = 0.005834 m/s.

B. *Final Shape of Deformed Beam*

With more nodes in the model, we can see the structure bends with a curvature in Fig. 6. After reaching equilibrium, the shape is stable and the whole structure would fall at the same speed without further deforming. From T=10s to the end of the simulation when T = 50s, the shape of structure stays identical and falls at constant speed of terminal velocity.

C. *Significance of Spatial and Temporal Discretization*

The relationship between terminal velocity, special discretization and time discretization is shown in Fig. 7. It can be observed that with a higher number of nodes we can get closer to an actual beam with continuous structure. That said, with sufficient space discretization, N ≈ 20 in this case, we can obtain an accurate enough estimation. With time discretization, we can see that the terminal velocity does not vary much under different time step sizes. Showing that we can reach an accurate estimation even with small dt and save a lot of run time.

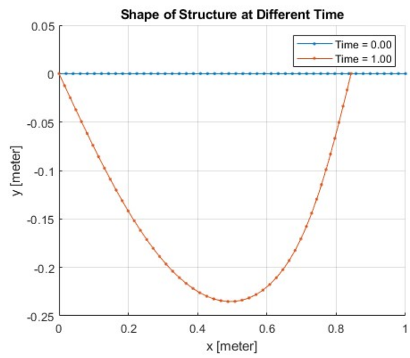

Figure 9. Significant deformation under heavy load.

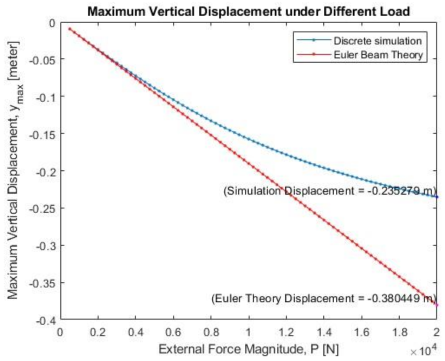

Figure 10. Comparison of simulation and theory under varying load.

III. Elastic Beam Bending

A. Maximum Displacement y_{max}

Fig 8. is the Maximum vertical displacement over the duration of the simulation. Since we assume that no viscous force is exerted, the beam can move freely without resistance and drag. This resulted in the structure reaching maximum displacement moments after the force is applied, leading to immediately reaching steady state with y_{max} = - 0.037175 meter. We can see the beam transformation overshoot and bounce back as well. Had we included viscous force in the simulator, we would observe the beam reaching steady state slower and may experience less overshooting. That said we would still obtain a similar steady state displacement, due to decreasing viscous force as velocity decreases.

B. Simulation Compared to Euler Beam Theory

To make a comparison with Euler Beam theory, we have its equation below:

$$y_{max} = \frac{Pc(L^2-c^2)^{1.5}}{9\sqrt{3}EIl} \quad where\ c = mid(d, l-d) \tag{1}$$

In which we calculated in theory y_{max} = 0.03804 meter. Compared to our simulation result we derive an error of:

$$\frac{y_{theory}-y_{simulation}}{y_{simulation}} = 0.0233 \tag{2}$$

As we know, Euler beam theory works best with small deformation as it neglects horizontal displacement. In our case, the beam experiences around 3.7% of vertical deformation and 0.35% of horizontal deformation. This relatively small shape change makes Euler theory adequate for finding an accurate solution. Our simulation result is very close to the theoretic solution in this case with only 2.33% error.

C. Accuracy under Large Deformation

Once we increase the external force to ten times the original value, the deformation is too big for the Euler theory to obtain the correct solution. Fig. 9 indicates the structure shape of our simulation under external force P = 20000 N. Where we can clearly see the significant deformation in both horizontal and vertical direction. Fig. 10 clearly shows at what P value the inconsistencies in Euler Beam theory starts to appear. It is observed that the two methods can obtain a relatively similar result withing P = 3000 N. Overall, with discrete simulation's attention to horizontal deformation, it can accurately calculate the solution under wide range of load magnitude. We can also see that the Euler Theory Displacement data travels in a straight line, which indicates that the relationship between displacement and force magnitude is linear. This can also be seen in the Euler formula where P is geometric proportional to maximum displacement. However actual beams in real world don't bend this way, thus supporting the observed inaccuracy of the Euler method.