Homework2: Discrete Elastic Rods Algorithm

Yuan-Hung Lo

I. Program Flow

A. Initialization

The program begins by initiating global variables such as mass, Young's modulus, cross section area, etc. to make our data transfer between functions easier. Then the number of nodes N, simulation time totalTime, are also determined.

B. Setup Physical Parameters

The properties of our elastic rod are then established by setting up the length of the rod, the curvature radius, cross section radius. Material parameters are then determined by assigning value to Young's modulus, Poisson's ratio, and shear modulus, allowing us to build up our physical characteristics. Stiffness properties such as moment of inertia and cross section area are also calculated based on our rod's geometry.

C. Setup Nodes and Geometry

Using geometry information provided in the question description, we defined the position of the 50 nodes in our rod by their x, y, and z axis coordinates. The twist and curve angle are also labeled as the theta of each node. We then set up the initial conditions of this rod by setting initial position: q0 as the current node position and curve angle and set initial velocity of all nodes to zero. We also set the reference frame, and compute material direction using that and bending angle.

D. Setup Simulation Environment

Our simulation only has gravity as external force F_g, and has bending, stretching, twisting elastic forces working between nodes. We also defined tolerance to help determine the convergence of our equation. To account for the fixed first two nodes of the rod, we established free index and fixed index. Only by doing Newton's method on free indexes can ensure our result follow the constraint of the simulation requirement.

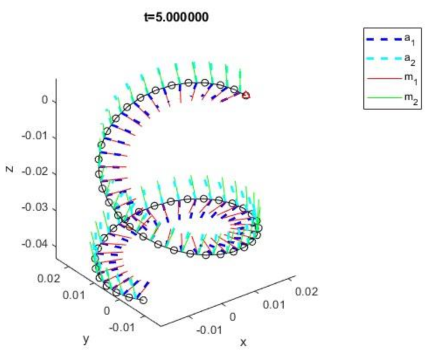

Figure 1. Final shape of rod.

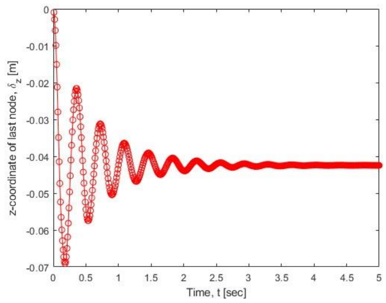

Figure 2. Total length of rod over time, with dt = 0.01.

E. Run Simulation

Running the object function, we use gradient and Hessian functions to get the Jacobian matrixes of our force equilibrium equation and utilize that result with Newton's method to update the next position of each rod node. These functions loop through all the nodes at every instance, eventually providing us an insight of the spring like rod bouncing freely with the effect of gravity and internal elastic forces.

F. Saving Result

Addition to saving the rod shape every 1 second, we also save the z coordinate of the end node. Giving us the total length of the rod at every gradient time of our simulation. This allows us to clearly see the dampening of the magnitude of the bouncing rod.

II. Result and Visualization

A. Shape of the Rod

As the simulation runs through 0 to 5 seconds, we can see the shape of the rod slowly changing as time progresses. Fig. 1 shows the end shape of our simulation. We can see that the shape of the rod follows the natural curvature of the restraint given by the question description. This natural curve makes the rod act like a helical spring, making our result very useful for future application. If we make the program to draw the position of the rod every 0.1 seconds, we will be able to clearly see how the rod starts off by bouncing in a high amplitude, and slowly progress to static due to damping of our simulation.

We can also see that the deformation of the final rod shape decreases gradually from node 1 to node 50. This is because the nodes in the front must withstand the weight or gravity of all the nodes and sections behind it, which is why we can see the last few sections experience little deformation. This is intuitively correct and can be seen in real world scenarios.

By inspection, the frequency of the rod is approximately 0.3 seconds, meaning that to see the proper amplitude change during the simulation, the rod shape figure should be plotted at least every 0.3 seconds.

B. *Total Length of the Rod over Time*

Fig. 2 shows the length of the rod for the duration of the simulation with time gradient of 0.01 seconds. We can see that the maximum length is 0.07 meters while the stable length is 0.0425 meters.

We can also see amplitude dampens and stabilizes at around 3.5 seconds. Although this seems intuitive at first, our simulation does not account for viscosity and friction in the system. Theoretically the rod should continue to bounce at the same amplitude according to conservation of energy. This effect is resulted from numerical damping that is purely fictional due to the calculation used in our simulation. Each step a certain amount of max value will be neglected causing the resulting amplitude to seem slowly stabling. Because of this characteristic, we can change the effect of numerical damping by changing the time gradient. Fig. 3 and Fig .4 shows rod length over time simulated in different time gradient, where we can see the finer the time gradient, the less potent the effect of numerical damping is. In order to achieve accurate result and lessen the effect of numerical damping, simulation should include drag or friction in systems or increase time discretization.

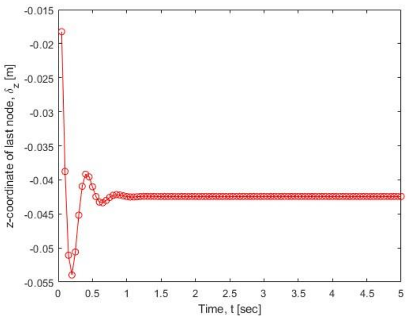

Figure 3. Total length of rod over time, with dt = 0.05.

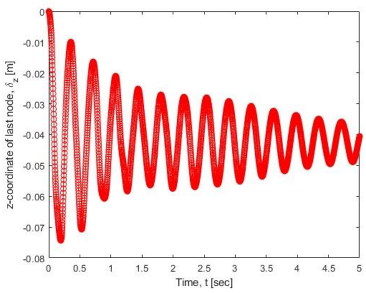

Figure 4. Total length of rod over time, with dt = 0.002.