Simulation of Discrete Elastic Rods

Premkumar Sivakumar

I. Introduction

This report summarizes the results for a complete physically-based simulation of discrete elastic rods. Elastic rods, in addition to bending and stretching, can undergo twisting. So, the simulation of motion for a rod happens in three dimensions, in contrast to the planar motion for discrete elastic beams. Hence, the DOF of a rod with N nodes and (N-1) edges can be given by (4N-1) which includes three DOF (x, y, z coordinates) at each node and one DOF (twist angle, Θ) at each edge.

II. Discrete Elastic Rods

The elastic rod in consideration has 20 nodes, a total length l of 20 cm and is naturally curved with radius R_n = 2 cm. The location of the N nodes at time t = 0 is given by:

$$\mathrm{x}_k = [R_n \cos((k-1)\Delta\theta), \qquad R_n \sin((k-1)\Delta\theta), \qquad 0],$$

where $\Delta\Theta = (l / R_n) \times (1 / N\text{-}1)$. Fig 1 shows the initial configuration of the rod, with the red triangle marking the position of the first node. The twist angles at t = 0 are 0. The first two nodes and the first twist angle remain fixed throughout the simulation (i.e. one end is clamped). The physical parameters are: density ρ = 1000 kg/m^3, cross-sectional radius r_0 = 1 mm, Young's modulus E = 10 MPa, shear modulus $G = E/3$ (corresponding to an incompressible material), and gravitational acceleration $g = [0, 0, -9.81]^T$. The total simulation time is 5 s with a time step size Δt of 0.01 s. The rod is suspended under gravity from its initial position.

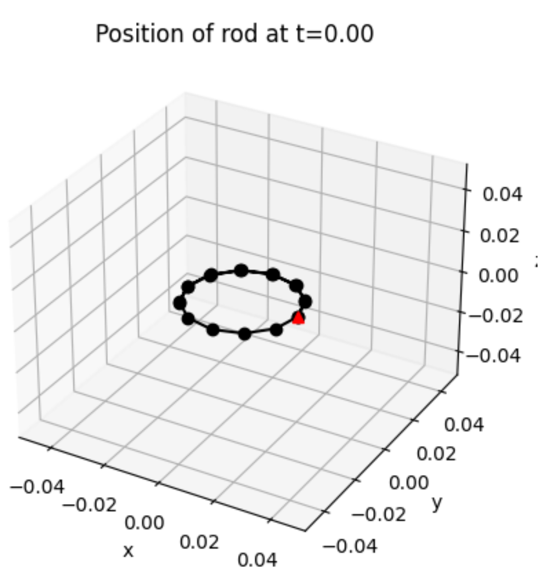

Figure 1

A. Simulation of Deformation of Rod Under Gravity

Fig 2 shows the final shape of the rod after 5 s of simulation.

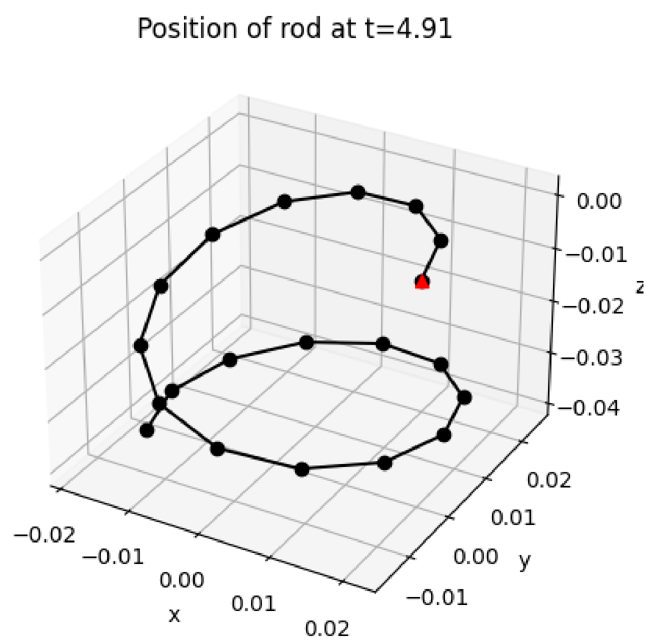

Figure 2

B. Tip Deflection of the Rod with Time

The rod, once suspended from the initial configuration under gravity, oscillates for a while before coming to rest. Fig 3 shows the tip deflection in terms of the z-coordinate of the last node with time.

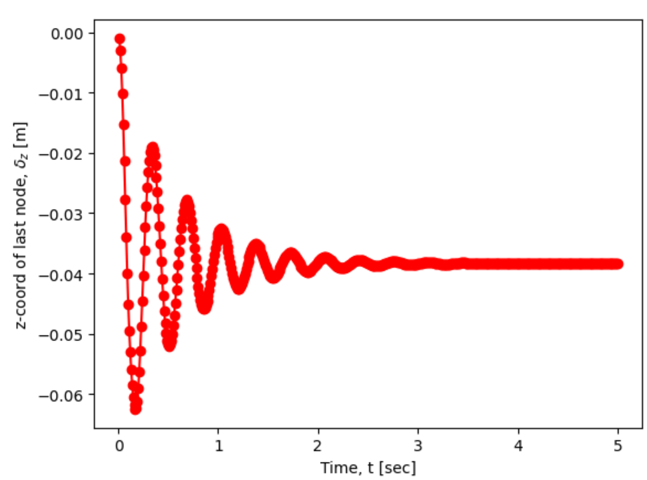

Figure 3

As it can be seen, the *z*-coordinate of the last node reaches a steady state value of ~ -0.04 m for a time step size Δt of 0.01 s and 20 nodes.

III. Implementation of Discrete Elastic Rods Algorithm

Following is the pseudocode of discreate elastic rods:

Algorithm 1: Discrete Elastic Rods

Require: DOFs and velocities at $t = t_j$
Require: Reference frame at $t = t_j$
Require: Index of the free DOFs
Ensure: DOFs and velocities at $t = t_{j+1}$
Ensure: Reference frame at $t = t_{j+1}$

```
function Discrete_Elastic_Rods(DOF_old, Velocity_old,
                               Reference frame_old)
    Guess DOFs at t_{j+1} as DOF at t_j
    n = 1
    while error > tolerance do
        Compute reference frame
        Compute reference twist
        Compute material frame
        Compute force and Jacobian of free indices
        Update DOFs of free indices using Newton-Raphson
        Calculate error
        n = n+1
    end while

    Update DOF at t = t_{j+1}
    Calculate velocity at t = t_{j+1}
    Update reference frame at t = t_{j+1}
    return DOF, velocity, ref frame, tangent at t = t_{j+1}
end function
```

To compute the elastic forces and Jacobian of the elastic forces as shown in the algorithm above, the gradient and hessian of elastic energies are needed. Following is the pseudocode for calculating gradient and hessian of elastic energies in a rod.

Algorithm 2: Gradient & Hessian of Elastic Energies in Rod

Require: DOFs
Require: Elastic stiffness
Require: Undeformed Voronoi length
Require: Undeformed edge length
Require: Natural Curvature
Require: Reference frame
Require: Reference twist
Require: Material frame
Ensure: Force
Ensure: Jacobian

```
function Grad_Hess_Elastic_Rod(DOF)
    Initialize Force as a vector of zeros of size 4N-1
    Initialize Jacobian as a zero matrix of size (4N-1, 4N-1)
    for k from 1 to N-1 do
        Save the DOFs x_k and x_{k+1}
        Calculate gradient and hessian for stretching energies
        Store the force in the vector
        Store the Jacobian in the matrix
    end for

    for k from 2 to N-1 do
        Save the locations of DOFs
        Calculate gradient and hessian of bending energies
        Store the force value
        Store the Jacobian value
        Calculate gradient and hessian of twisting energies
        Store the force value in the vector
        Store the Jacobian value in the matrix
    end for
    return Force and Jacobian
end function
```

Following is the algorithm for the simulation or the time stepping loop.

Algorithm 3: Simulation

Require: Physical parameters
Require: Material properties
Require: Simulation time and time step size
Require: Mass matrix
Require: External force
Require: Initial DOF vector
Require: Boundary conditions
Require: Natural curvature

```
Calculate total number of steps (Nsteps)
Initialize current time as zero
for timeStep=1, timeStep < Nsteps, timeStep++ do
    Assign the initial DOF as the guess value
    Call the function Discrete_Elastic_Rods()
    Update the DOFs
    Update current time
    Store the z-coordinate of the last node
end for

Plot the z-coordinate of last node with time
```

References

[1] Professor M. Khalid Jawed, MechAE 263F Course modules Link: https://bruinlearn.ucla.edu/courses/193842/modules Portion of this code and the helper functions or files have been referenced from the class resources on Bruinlearn.