Discrete Elastic Rod Simulation of Shoelace Knots and Strength Final Report

Mechanics of Flexible Structures and Soft Robots - MAE 263F

Esther Gérard, Evelyn Kim, Noah Shamsai

***Abstract*— In the field of flexible structure and soft robotics, analysing an elastic rod is an important fundamental subject with future potential and practical application. Recently, many innovative approaches have continued to emerge, such as [9], [6], and [1], enabling effective numerical interpretation and advanced simulation.**

Among the various topics, knot tying is the most representative target for analysis, which can serve as a foundational structure with the potential to extend to various forms of soft robotics.

In this project, we aim to advance our analysis beyond basic knot structures by conducting a comprehensive simulation study focusing on shoelace knots. Leveraging the Discrete Elastic Rod (D.E.R.) algorithm, we will simulate a shoelace knot structure with varying properties to compare knot strengths. We assume dynamic equilibrium under constraints for the implementation the Implicit Model Contact (IMC) model.

Through this analyses and simulation, we hope to not only acquire knowledge in the simulation of flexible structures, but also a deeper understanding of the the affects of model parameters and their impacts on both simulation accuracy and speed.

I. Introduction

Our group project goal is to simulate the tying and untying of a shoelace knot, and to study the impact of friction and elastic parameters on the strength of the knot. Our work will be based on the Discrete Elastic Rod algorithm and on the Implicit Contact Model, which are detailed below. We will implement our algorithm in C++ in order to take advantage of its computational efficiency.

The two following sections about our general simulation method and about the simulation of contact are put in this document as a reminder from our proposal report. Then we explained the assumptions we chose and our current progress. Finally, our plan for the following weeks is presented.

II. Discrete Elastic Rod algorithm

The Discrete Elastic Rod (D.E.R.) algorithm [3] is useful in 3D simulation of elastic rod. The rod discretized and made to be at dynamic equilibrium under applied constraints. My changing the constraints over time (i.e. end positions of the "shoelace"), we will be able to manipulate the rod into a tied and then subsequently untied shoelace knot. The D.E.R. algorithm is extremely useful to us because it models the rod with all elastic and external forces at equilibrium.

General implementation of the D.E.R. Algorithm can be followed in Figure 1 [8]. The first appearance of the

Algorithm 1 Discrete Elastic Rods

Require: $\mathbf{q}(t_j), \dot{\mathbf{q}}(t_j)$ ▷ DOFs and velocities at $t = t_j$
Require: $\left(\mathbf{a}_1^k(t_j), \mathbf{a}_2^k(t_j), \mathbf{t}^k(t_j)\right)$ ▷ Reference frame at $t = t_j$
Require: `free_index` ▷ Index of the free DOFs
Ensure: $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$ ▷ DOFs and velocities at $t = t_{j+1}$
Ensure: $\left(\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)$ ▷ Reference frame at $t = t_{j+1}$

function DISCRETE_ELASTIC_RODS($\mathbf{q}, \dot{\mathbf{q}}(t_j), \left(\mathbf{a}_1^k(t_j), \mathbf{a}_2^k(t_j), \mathbf{t}^k(t_j)\right)$)
  Guess: $\mathbf{q}^{(1)}(t_{j+1}) \leftarrow \mathbf{q}(t_j)$
  $n \leftarrow 1$
  while `error` > `tolerance` **do**
    Compute reference frame $\left(\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)^{(n)}$ using $\mathbf{q}^{(n)}(t_{j+1})$
    Compute reference twist $\Delta m_{k,\text{ref}}^{(n)}$ $(k = 2, \ldots, N-1)$
    Compute material frame $\left(\mathbf{m}_1^k(t_{j+1}), \mathbf{m}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)^{(n)}$
    Compute $\mathbf{f}$ and $\mathbb{J}$ ▷ Eqs. 7.1 and 7.2; see Algorithm 2
    $\mathbf{f}_{\text{free}} \leftarrow \mathbf{f}$ `(free_index)`
    $\mathbb{J}_{\text{free}} \leftarrow \mathbb{J}$ `(free_index, free_index)`
    $\Delta \mathbf{q}_{\text{free}} \leftarrow \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$
    $\mathbf{q}^{(n+1)}$ `(free_index)` $\leftarrow \mathbf{q}^{(n)}$ `(free_index)` $- \Delta \mathbf{q}_{\text{free}}$ ▷ Update free DOFs
    `error` $\leftarrow$ `sum ( abs (` $\mathbf{f}_{\text{free}}$ `) )`
    $n \leftarrow n + 1$
  end while

  $\mathbf{q}(t_{j+1}) \leftarrow \mathbf{q}^{(n)}(t_{j+1})$
  $\dot{\mathbf{q}}(t_{j+1}) \leftarrow \frac{\mathbf{q}(t_{j+1}) - \mathbf{q}(t_j)}{\Delta t}$
  $\left(\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right) \leftarrow \left(\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)^{(n)}$
  return $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1}), \left(\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)$
end function

Fig. 1. D.E.R. Algorithm Pseudo-code, from [8]

D.E.R. algorithm is in [2]. Since, it has been used in several research projects, especially to study knot tying [7]. The reason for that is the strength of this model on 1-D bodies in large deformation.

Kirchhoff-Love plate theory states that thin elastic plates undergo linear deformation with negligible transverse shear deformations when subjected to mechanical loads. When adapted to model rods by considering them as effectively 1-D structures, we can predict the deformation and stress distribution along the rod's length, allowing for the accurate analysis of slender structures under various loading conditions. Key elements of Kirchoff-Love plate theory include linear elastic behavior, linear strain-displacement relations, and the assumption that shear deformations are negligible.

Since the publication of [2], which relied principally an space-parallel transport frames to compute elastic rod twist, it was discovered that significant computational time advantages could be had if time-parallel transport was used. Hence this is the implementation that we will be using.

III. Implicit contact model

The D.E.R. algorithm is constructed to handle the elastic forces of the body. But in knot tying, another physical phenomenon is very important: friction. That is why a physical friction model was necessary to conduct a numerical study of knots. If a lot of numerical friction models exist, we chose the IMC model, for its physical relevancy and its computational efficiency.

The IMC model is presented in [10] and [5]. The main point of this method is to compute implicitly a forces equilibrium, including contact forces, by choosing a smoothed (and so differentiable) formulation of the contact potential. The contact potential is calculated based on an analytical segment-segment distance between the discrete edges of the rod, instead of on the distance between nodes, to obtain more realistic results. This differentiable formulation of the contact energy allows to compute its gradient and hessian, and therefore to include it in the implicit formulation. The contact potential in [10] is defined by:

$$E(\Delta, \delta) = (2h - \Delta)^2 \quad \text{if } \Delta \in [0, 2h - \delta]$$
$$E(\Delta, \delta) = (\frac{1}{K_1} log(1 + e^{K_1(2h-\Delta)}))^2 \quad \text{if } \Delta \in [2h - \delta, 2h + \delta]$$
$$E(\Delta, \delta) = 0 \quad \text{if } \Delta \geq 2h + \delta$$

In this formula, K_1 is the stiffness of the energy curve, h is the radius of the rod, Δ is the distance between the two edges in contact, and δ is a tolerance value on the distance. Using this expression, the curves shown in figure 2 are obtained.

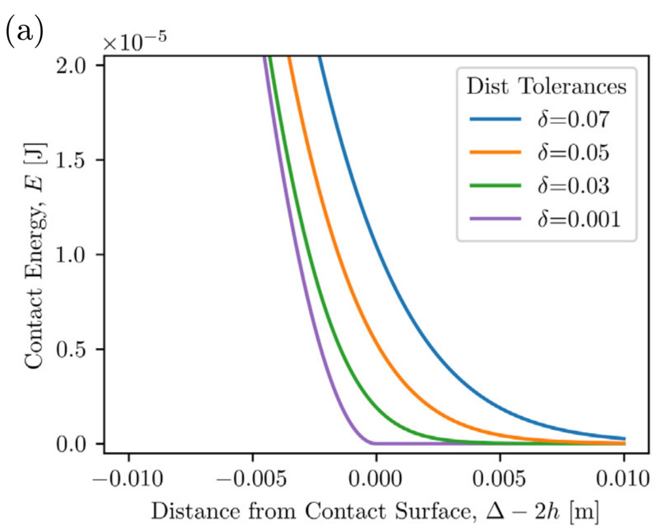

Fig. 2. Contact energy curve, this figure is extracted from [10]

The Implicit Model Contact was used in [5] to study the three-foil knot-tying phenomena. To do so, the IMC was implemented in a D.E.R. algorithm. The simulation gave very good results when compared to theoretical results or to another accurate model for contact in knot tying, the Spillmann and Teschner model (SPT). By comparison, the IMC method gives smoother pull forces than SPT.

IV. Assumptions and Implementation

Our overall simulation flow follows the D.E.R. algorithm, the core settings are similar, and the setup values used for our simulation are shown in Table I. Here, we assume that no external force other than gravity and the force we set up is supposed to affect the system. Thus, the system modeling is assumed to be unaffected by other external forces.

We have implemented the D.E.R algorithm in C++ in order to improve speed and thus achievable quality of simulation. C++, do to its optimization advantages over MATLAB, can produce faster and more sophisticated results. Using Newton's method to simulate the system implicitly, we we able to achieve a high degree of accuracy while allowing larger time steps when compared to an explicit method.

We implemented our simulation based on DisMech [11, 4], a fully implicit penalty-based contact method for 3D elastic rod simulations. We incorporated IMC to simulate both contact and fabrication using the D.E.R. framework [3]. For external factors, we applied boundary conditions in the form of displacement instead of force. This is to ensure that the simulation can be controlled and carried out as intended. The approach is for the movement when a person holds onto a string and moves, where displacement restricts the structure. These conditions are sequentially applied as the structure of a shoelace changes into a certain form.

We faced a number of issues in implementation, however, and had to work through them via modified parameter testing and then subsequent changes in the code based on the data acquired the parameter testing. The specifics of the challenges and how we worked through them are address below in Section V.

Parameters	
Rod Radius	1.6[mm]
Young's Modulus, E	0.18 [MPa]
Poisson's Ratio	0.5
Density, ρ	10000 kg/m^3
Time Step, dt	0.001 [s]
Tolerance	0.0001 ‘
Total simulation time	30 [$minute$]

TABLE I

Parameters used in the simulation. Note lack of gravity & fluid friction

V. Code interventions

During the process of writing the boundary conditions that led to a tied reef knot, it was necessary to be able to release some of the previously fixed degrees of freedom. Previously, a variable containing the fixed DOF was computed at the beginning of the simulation, containing a 1 at the indexes of the fixed DOF and a 0 at the indexes of the free ones. Adding a new boundary condition implied adding 1 at all the lines of non-free indexes. All the non-zero values were considered in the Newton solving as fixed DOF. For this reason, it was possible to constraint DOFs that were previously free, but not possible to free the ones that were previously constrained. We implemented a function that resets the variable of fixed DOF and called it at the beginning of each new step of the knot tying.

Being able to release fixed DOF brought another problem: if any energy was stored in the fixed vertices, releasing them led to high dynamical answers, that caused divergences. Therefore, we also implemented a simple viscous force, proportional to the speed of the nodes. We added it to the implicit model (force and Jacobian), to slow the movements, and make them more realistic and easier to compute.

Even with this fluid viscous force activated when releasing some DOF, the convergence of the model was not easy to ensure, especially when the number of contacts was high at the end of the tying process. To solve this problem, we had to choose which parameters were better for the computation to converge.

```
helixRadius         0.01
helixPitch          0.05
numFlagella         1
rodRadius           0.0016
youngM              0.18e6
Poisson             0.5
deltaTime           1e-3
totalTime           35
tol                 1.0e-3
stol                1.0e-3
maxIter             3000
density             10000
gVector             0.0 0.0 0.0
viscosity           0
epsilon             1.02e-3
axisLengthInput     0.2
deltaLengthInput    5e-3
distance            0.03
render              1
saveData            1
omega               15
mu                  0
nu                  1e-4
delta               1e-5
colLimit            2e-3
lineSearch          1
ipc                 0
```

Listing 1. Code parameters

Listing 1 shows our optimal parameters for the simulation code. We focused on changing values for rodRadius, youmM, daltaTime, viscosity, and mu. Increasing the rod radius or time delta had a negative impact on the proper convergence of the simulation. We attempted various parameter sets to find the optimal values and experimentally confirmed that the parameters listed in Listing 1 have been optimized for our shoelace simulation.

The main part of the simulation is how we apply boundary conditions. Other parts follow the typical processes of D.E.R. and the IMC model algorithm. Summarizing the process of updating boundary conditions, we iterate through each rod node to apply the boundary conditions. The overall process of updating boundary conditions is shown in Listing 2. Subsequently, at each time point that the boundary conditions are applied, we update the boundary condition u and apply it to the target node. We simulate the process of tying shoelaces with a total of 25 time points where boundary conditions are applied, ranging from t_{BC1} to t_{BC25}.

```
void world::updateBoundary(){

for (int i = 0; i < numRod; i++){
    Vector3d u;
    u < BOUNDARY CONDITION FOR THIS TIME POINT;

    if (currentTime <= t_BC1)
    {
        v_dumbviscoForce[i]->isReleasing = true;
        // Apply BCs to each target node
        rodsVector[i]->setVertexBoundaryCondition(
    rodsVector[i]->getVertex(0) - u * deltaTime, 0)
    ;
        ...
    }
    else if (currentTime > t_BC1 && currentTime <=
    t_BC2)
    {
        // Apply BCs at t_BC1, t_BC2, ..., t_BC25
        ...

}}
```

Listing 2. Implementation for Boundary Conditions

Our boundary conditions are applied in terms of displacement. When we apply the boundary condition u, we multiply it by the time delta to sequentially apply it over time. From these applied boundary conditions, we can analyze energy and forces within the rod during the shoelace tying and untying process.

VI. Achievements

Our main achievement was to propose a set of parameters and of hard-written boundary conditions that ensure the tying of a reef knot. The weakness of our proposition is nevertheless that we cannot use this set of boundary conditions if the material and geometrical parameters are different. The method can be easily adapted but it is for now not handled automatically.

A video of the knot being tied and untied is available at the following link: https://drive.google.com/file/d/1jrYWk0HMcBE7r-2BGL6L2ZscT2_xHoqB/view?usp=sharing.

Once the knot was tied, we implemented the untying. We were able to plot the force applied at the two ends of the rod as a function of the chosen displacement. What we obtained is shown in figures 3 and 4.

We were expecting a curve closer to a straight line, here we can see that almost no force is necessary to untie the

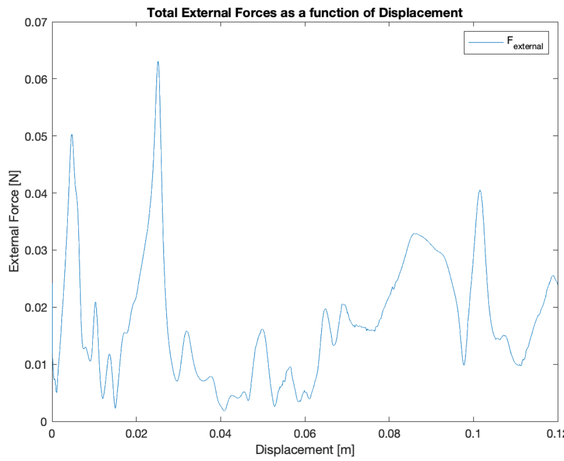

Fig. 3. Total External forces required to Untie Reef knot as a function of time

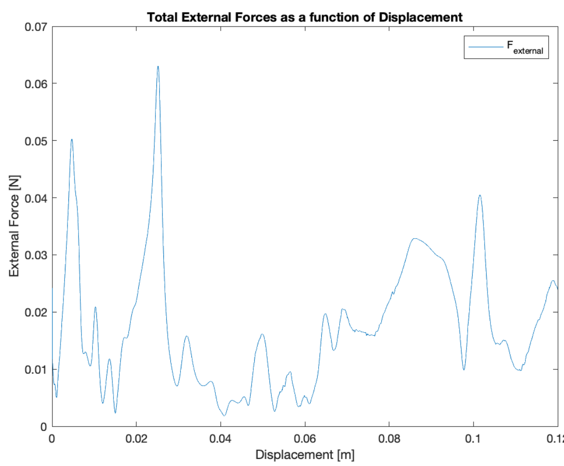

Fig. 4. X,Y,Z component's of External forces required to Untie Reef knot as a function of time

knot. But it fits with what we see in the recording of the simulation: the stored stretching energy is sufficient to untie the knot when the extremities of the bunny ears loops are released. This effect might be a result of the very low Young's modulus we chose for the computation, and to the small diameter of the rod, which can lead to a non physical result in handling contacts.

VII. Conclusions and Further Ideas

In conclusion, our project focused on the simulation and analysis of shoelace knots using the Discrete Elastic Rod (D.E.R.) algorithm and the Implicit Model Contact (IMC) model. We successfully implemented the D.E.R. algorithm in C++ and incorporated the IMC model to account for frictional forces during knot tying. Our simulations aimed to understand the impact of various parameters such as rod radius, Young's modulus, and contact stiffness on the strength and untying behavior of the shoelace knot.

Through our efforts, we were able to successfully simulate the tying and untying of a reef knot through enforced and released boundary conditions that changed with time. We were able to work through challenges related to releasing fixed degrees of freedom, introducing a viscous force to manage dynamic responses, and fine-tuning parameters for convergence.

However, our simulation results showed unexpected behavior in the untying process, with minimal external forces required. This outcome may be attributed to the choice of low Young's modulus and small rod diameter, leading to non-physical results in contact handling. Further refinement and exploration of material and geometrical parameters are needed to improve the realism of the simulation.

The work we realised in this project is a first step towards a complete study of the reef knots. The modifications we add to the existing code and the boundary conditions we wrote can be use for a further work on making the simulation fits the reality, after obtaining experimental data about the strength of reef knots. It can also be slightly modify to obtain different types of knots, as the granny knot for instance.

References

[1] Costanza Armanini et al. "Soft robots modeling: A structured overview". In: *IEEE Transactions on Robotics* (2023).

[2] Miklos Bergou et al. "Discrete Elastic Rods". In: *ACM Transactions on Graphics* 27 (2008).

[3] Miklós Bergou et al. "Discrete elastic rods". In: *ACM SIGGRAPH 2008 papers*. 2008, pp. 1–12.

[4] Andrew Choi et al. "Implicit Contact Model for Discrete Elastic Rods in Knot Tying". In: *Journal of Applied Mechanics* 88.5 (Mar. 2021). ISSN: 0021-8936. DOI: 10.1115/1.4050238. URL: https://doi.org/10.1115/1.4050238.

[5] Andrew Choi et al. "Implicit contact model for discrete elastic rods in knot tying". In: *Journal of Applied Mechanics* 88.5 (2021).

[6] Xiaonan Huang et al. "Design and Closed-Loop Motion Planning of an Untethered Swimming Soft Robot Using 2D Discrete Elastic Rods Simulations". In: *Advanced Intelligent Systems* 4.10 (2022), p. 2200163.

[7] M Khalid Jawed et al. "Untangling the Mechanics and Topology in the Frictional Response of Long Overhand Elastic Knots". In: *American Physical Society* 115 (2015), p. 118302.

[8] M. Khalid Jawed and Sangmin Lim. *Discrete Simulation of Slender Structures*. Los Angeles: University of California, Los Angeles, 2022.

[9] Longhui Qin et al. "Modeling and simulation of dynamics in soft robotics: a review of numerical approaches". In: *Current Robotics Reports* (2023).

[10] Dezhong Tong et al. "A fully implicit method for robust frictional contact handling in elastic rods". In: *Extreme Mechanics Letters* 58 (2023), p. 101924.

[11] Dezhong Tong et al. "A fully implicit method for robust frictional contact handling in elastic rods". In: *Extreme Mechanics Letters* 58 (2023), p. 101924. ISSN: 2352-4316. DOI: https://doi.org/10.1016/j.eml.2022.101924. URL: https://www.sciencedirect.com/science/article/pii/S2352431622002000.