UNIVERSITY OF CALIFORNIA

Los Angeles

An Implicit Contact Method for Tying Discrete Elastic Knots

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Andrew Sang-Jin Choi

2021

ABSTRACT OF THE THESIS

An Implicit Contact Method for Tying Discrete Elastic Knots

by

Andrew Sang-Jin Choi
Master of Science in Computer Science
University of California, Los Angeles, 2021
Professor Demetri Terzopoulos, Chair

Rod-rod contact is critical in simulating knots and tangles. To simulate contact, typically a contact force is applied to enforce the non-penetration condition. This force is often applied explicitly (forward Euler) such that at every time step of the dynamic simulation, the equations of motions are solved repeatedly until the right amount of contact force successfully imposes the condition. There are two drawbacks to this method: (1) the explicit implementation suffers from numerical convergence issues and (2) solving the equations of motion iteratively to find the right contact force slows down the simulation. In this thesis, we propose a simple, efficient, and fully-implicit contact model with good convergence properties. Compared to previous methods, ours is shown to be capable of taking large time steps without forfeiting accuracy during knot tying simulations. It involves a new contact potential, based on a smoothed segment-segment distance function, that is an analytic function of the four endpoints of the two contacting edges. Since our contact potential is differentiable, we can readily incorporate its force (negative gradient of the energy) and Jacobian (negative Hessian of the energy) into the elastic rod simulation.

The thesis of Andrew Sang-Jin Choi is approved.

Song-Chun Zhu

Jungseock Joo

Mohammad Khalid Jawed

Demetri Terzopoulos, Committee Chair

University of California, Los Angeles

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Following the pioneering computer graphics work on the physics-based modeling and simulation of deformable curves, surfaces, and solids (Terzopoulos et al., 1987; Terzopoulos and Fleischer, 1988b,a), discrete differential geometry-based simulations have shown surprisingly successful performance in simulating slender structures, e.g., rods (Bergou et al., 2008, 2010; Audoly and Pomeau, 2010; Jawed et al., 2018b), viscous threads (Audoly et al., 2013; Bergou et al., 2010), ribbons (Shen et al., 2015), and plates/shells (Baraff and Witkin, 1998; Grinspun et al., 2003; Batty et al., 2012).

The Discrete Elastic Rods (DER) algorithm (Bergou et al., 2010; Jawed et al., 2018b)—originally developed in the computer graphics community to simulate hair, fur, and other filamentary structures in movies—has been borrowed by the engineering community to solve a variety of problems involving the deployment of rods (Jawed et al., 2014; Jawed and Reis, 2014; Jawed et al., 2015b), propulsion of bacterial flagella (Jawed et al., 2015c; Jawed and Reis, 2016, 2017), elastic gridshells (Panetta et al., 2019; Baek et al., 2018; Baek and Reis, 2019), and self assembly of carbon nanotubes (Jawed et al., 2018a).

One of the most interesting and challenging problems related to these tasks is the simulation of knots, which introduces complex and intricate patterns of self-contact within an elastic rod. The mechanics of knots (Jawed et al., 2015a; Moulton et al., 2018; Patil et al., 2020; Audoly et al., 2007; Clauvelin et al., 2009; Durville, 2012; Przybyl and Pieranski, 2009) tied in elastic rods is an intricate interplay of the internal elastic forces and external friction. Therefore, when simulating knots, a fast, stable, and physically accurate contact model is desired.

1.1 Contributions

In this thesis, we introduce the Implicit Contact Model (IMC), a contact handling method for DER simulations (Bergou et al., 2008, 2010) in which the contact forces are handled implicitly. DER discretizes the elastic rod into a number of "nodes". Two consecutive nodes are connected by an "edge". To deal with contact, we define a contact energy, which will be used to derive the normal contact forces, responsible for enforcing non-penetration, and Coulombic frictional forces. Instead of formulating this contact energy as a function of the distance between nodes, we formulate it as a function of the minimum distance between two edges. This results in more visually and physically realistic results.

Figure 1.1 presents snapshots from our simulations of the knot tying process, where the two free ends of the "tails" of the knots are pulled . Throughout this thesis, we consider open overhand knots (Jawed et al., 2015a); such knots can be described by the unknotting number n, related to the number of turns in the "braid" of the knot. Figs. 1.1(a–d) show knots with $n = 1, \ldots, 4$. Interestingly, when the knots with $n = 3$ and $n = 4$ are sufficiently tight, they undergo snap-through buckling and the "loop" of the knot suddenly transitions from a near-circular shape to a distorted configuration. Our simulation can reliably capture this behavior.

1.2 Thesis Overview

The remainder of this thesis is organized as follows: Chapter 2 reviews related work on contact algorithms. In Chapter 3, we discuss the methodology of the proposed version of DER with IMC as the contact model. In Chapter 4, we undertake the theoretical validation of the IMC contact model and compare it to a well-established contact model in terms of pull force accuracy and run time. Finally, we present concluding remarks and discuss potential future research directions in Chapter 5.

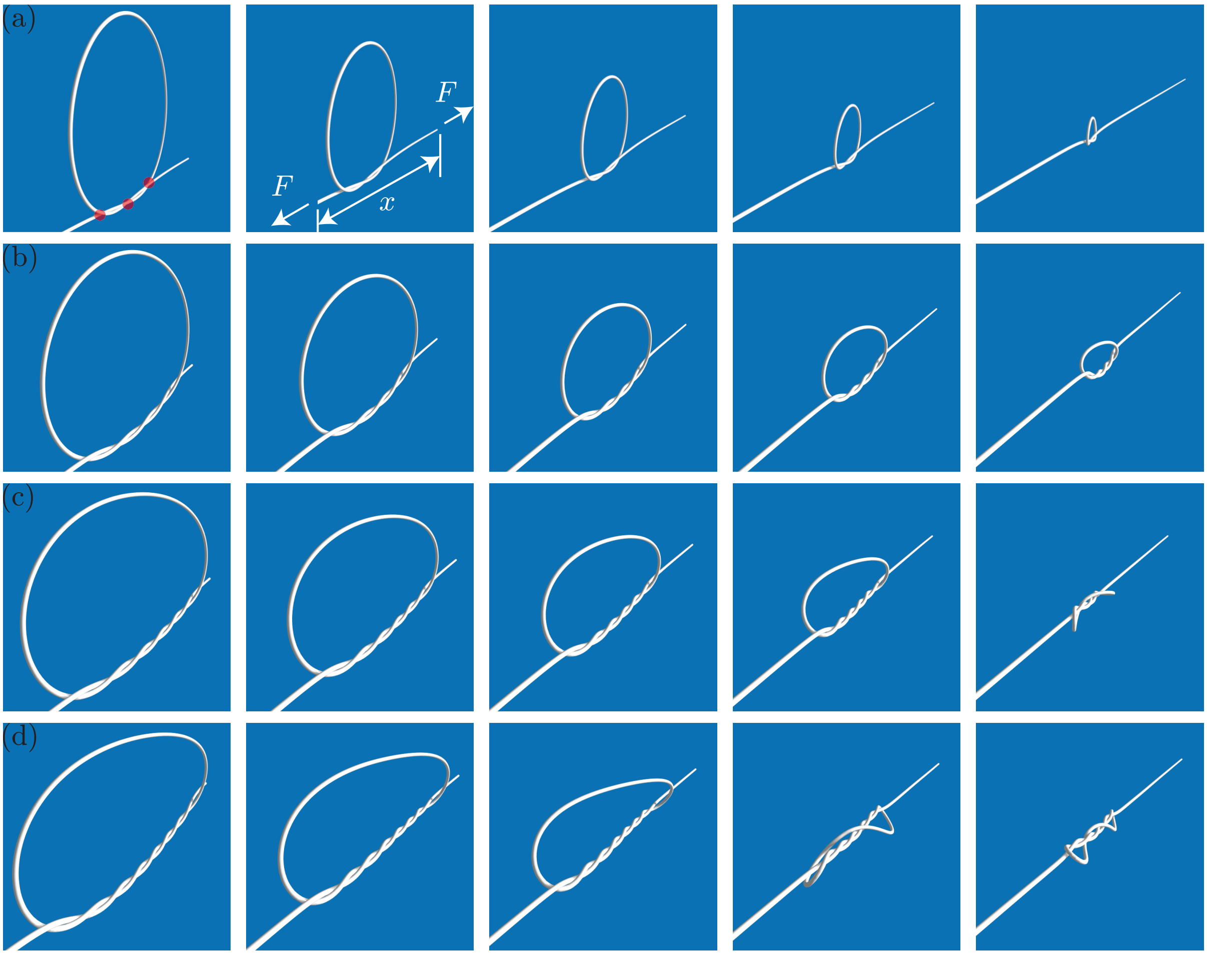

Figure 1.1: The knot tying process using DER and IMC for knots of (a) $n = 1$, (b) $n = 2$, (c) $n = 3$, and (d) $n = 4$. The red dots in (a) represent the crossing points of the braid. The unknotting number n is equal to $\frac{1}{2}$ times the number of crossing points minus 1. The end-to-end distance of a knot is $e = L - x$, where L is the total length of the rod and x is the distance between the two ends of the knot. The knot starts off in the configuration shown in the leftmost column and is gradually pulled tight from both ends leading to the configuration shown in the rightmost column. The physical parameters are detailed in Section 3.4.

CHAPTER 2

Related Work

Contact handling methods can generally be divided into three categories: impulse-based methods (Choe et al., 2005; Spillmann and Teschner, 2008), constraint-based methods (Bertails-Descoubes et al., 2011; Daviet et al., 2011; Kaufman et al., 2014), and penalty-based methods (Durville, 2012; Li et al., 2020; Patil et al., 2020).

DER, which implements a variation of Kirchhoff's elastic rod model (Kirchhoff, 1859), has been employed to handle contact during knot tying using a contact method proposed by Spillmann and Teschner (2008). This model resolves contact by computing the contact forces that will yield the desired, collision-free state. Although computationally efficient, due to its explicit nature, unrealistic visual jittering during knot tying occurs for insufficiently small time steps.

In addition to knot tying, several contact algorithms have been developed for simulating large assemblies of elastic rods, such as for the realistic simulation of hair. To do so, Choe et al. (2005) presented a framework that treated hair as a collection of wisps, greatly decreasing the number of collisions to process, albeit reducing accuracy. Wisp-wisp collisions are based on impulses while friction is modeled as a viscous damping force.

To capture exact Coulomb friction in fiber assemblies, Bertails-Descoubes et al. (2011) introduced a constraint-based method that models contact and Coulomb friction simultaneously as a zero finding problem of a nonsmooth function using the friction formulation of Alart and Curnier (1991). This problem is solved using a nonsmooth Newton algorithm along with Goldstein-Price line search.

Expanding upon the work of Bertails-Descoubes et al. (2011), Daviet et al. (2011) proposed a hybrid approach to contact solving by combining a zero-finding formulation of exact Coulomb friction along with an analytical solver of failure cases. This hybrid approach allows the solver to be scalable to fiber contact subject to tens of thousands of frictional contacts.

Kaufman et al. (2014) proposed a constraint-based method capable of simulating large assemblies of elastic rods by adaptively adjusting the degree of nonlinearity. This method formulates frictional contact using discrete Signorini-Fischera conditions (Kikuchi and Oden, 1988) and the maximal dissipation principle (Goyal et al., 1991). By adaptively incorporating sufficient nonlinearity into the collision response, more physically realistic results are produced compared to impulse methods.

Returning to the problem of knot tying, Durville (2012) introduced a penalty-based contact model that treated rods as elastic beams using a finite element approach. This method was successful in simulating rod deformation induced by knot tightening by using a 3D finite strain beam model.

In more recent work, Li et al. (2020) proposed an implicit time-stepping method that utilizes smooth barrier functions to simulate contact between deformable objects and induce deformations. Approximated functions are introduced to smooth the contact energy and frictional responses between two elements. This method robustly handles a variety of contact scenarios but remains relatively expensive compared to other penalty methods due to its use of continuous collision detection within minimization steps, contact-aware line search, and a customized Newton solver.

Finally, Patil et al. (2020) studied the topological mechanics of knots through a comparison of physical experiments with simulations in which the contact forces are based on strain potential and friction is formulated as a damping force.

Although robust, many of these state-of-the-art methods (Kaufman et al., 2014; Li et al., 2020) are difficult to implement, computationally expensive, and can often be overkill for simulating knots. This can be attributed to the fact that they are formulated in a way that robustly handles difficult contact scenarios such as high velocity impacts, which are usually absent in knot tying.

To remedy this, we formulate an implicit penalty-based contact algorithm that is efficient, intuitive, easy to implement, and specially catered to knot tying. We show that this algorithm is competitive with impulse methods in terms of computational efficiency while avoiding the inaccuracies of such methods. Furthermore, we do not resort to simulating friction using the simplified viscous damping formulation used by several penalty-based methods (Choe et al., 2005; Patil et al., 2020), but instead use Coulombic friction by limiting the problem space to non-static cases.

Finally, our algorithm does not contain expensive auxiliary components such as line searches (Bertails-Descoubes et al., 2011; Li et al., 2020) or heavily customized Newton-based solvers (Bertails-Descoubes et al., 2011; Daviet et al., 2011; Kaufman et al., 2014; Li et al., 2020). To demonstrate the effectiveness of our algorithm, we compare simulation results with the impulse-based contact method proposed by Spillmann and Teschner (2008) as it is the most comparable in implementation complexity and computation cost.

CHAPTER 3

Methodology

In this chapter, we first briefly discuss the DER model (Jawed et al. (2018b) provides a more in-depth introduction). Then, we formulate the IMC contact model as well as its intuitive integration into the existing DER algorithm. Finally, hyperparameters are explained in terms of their effects on algorithmic performance.

3.1 Discrete Elastic Rods

Under the DER framework, the centerline of a rod is discretized into N nodes, located at positions $\mathbf{x}_i$ for $1 \leq i \leq N$, and $N-1$ edges $\mathbf{e}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, for $1 \leq i \leq N-1$. Each edge has an orthonormal adapted reference frame $\{\mathbf{d}_1^i, \mathbf{d}_2^i, \mathbf{t}^i\}$ and a material frame $\{\mathbf{m}_1^i, \mathbf{m}_2^i, \mathbf{t}^i\}$, as shown in Figure 3.1(a). The reference frame is updated through parallel transport in time (Bergou et al., 2008, 2010) while the material frame can be obtained from the scalar twist angle θ^i with respect to the reference frame. Then, for a given rod, the Cartesian coordinates of each node as well as the twist angle for each edge results in $4N-1$ total degrees of freedom (DOFs), which can be represented by the vector $\mathbf{q} = [\mathbf{x}_1^T, \theta_1, \mathbf{x}_2^T, \theta_2, ..., \mathbf{x}_{N-1}^T, \theta_{N-1}, \mathbf{x}_N^T]^T$, where ^{T} denotes the transposition operator. In this representation, the location of node i is contained in entries $4i-3$, $4i-2$, and $4i-1$ of $\mathbf{q}$.

The total elastic energy of an elastic rod is

$$E = \sum_{i=1}^{N-1} E_i^s + \sum_{i=2}^{N-1} E_i^b + \sum_{i=2}^{N-1} E_i^t, \tag{3.1}$$

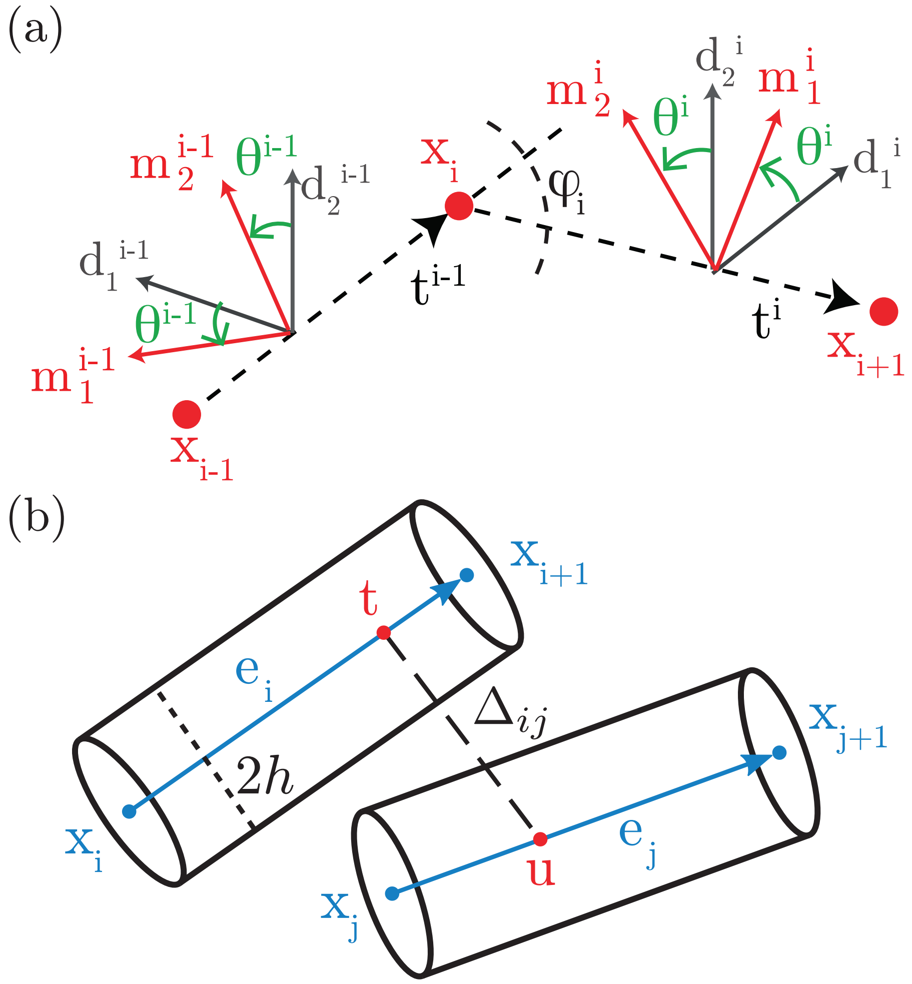

Figure 3.1: (a) Schematic diagram of a discrete rod. (b) The two edges (and four nodes) involved in a contact.

where E_i^s is the stretching energy of edge $\mathbf{e}_i$, E_i^b is the bending energy at node $\mathbf{x}_i$, and E_i^t is the twisting energy at node $\mathbf{x}_i$. For a rod with Young's modulus Y, shear modulus G, area moment of inertia I, polar moment of inertia J, and cross-sectional area A, the energies are formulated as

$$\begin{aligned} E_i^s &= \frac{1}{2} Y A \left(\frac{|\mathbf{x}_{i+1} - \mathbf{x}_i|}{|\bar{\mathbf{e}}_i|} - 1 \right)^2 |\bar{\mathbf{e}}_i|, \\ E_i^b &= \frac{1}{2} Y I (|\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i|)^2 \frac{1}{L_i}, \\ E_i^t &= \frac{1}{2} G J \boldsymbol{\tau}_i^2 \frac{1}{L_i}, \end{aligned} \tag{3.2}$$

where $|\bar{\mathbf{e}}_i|$ is the length of edge $\mathbf{e}_i$ in its undeformed state, $\boldsymbol{\kappa}_i$ is the curvature vector at node $\mathbf{x}_i$, $\bar{\boldsymbol{\kappa}}_i$ is the undeformed curvature for the same node, $\boldsymbol{\tau}_i$ is the integrated twist at node $\mathbf{x}_i$, and $L_i = (|\bar{\mathbf{e}}_{i-1}| + |\bar{\mathbf{e}}_i|)/2$ is the Voronoi length in the undeformed state. Typically the rod is uniformly discretized and the length of each edge is the same throughout the rod; i.e., $L_i = L$. All three energies are functions of the configuration of the rod represented by $\mathbf{q}$ (Jawed et al., 2018b). For each DOF q_i in $\mathbf{q}$, the internal elastic forces (for nodal positions) and elastic moments (for twist angles), which are elements of the internal elastic force vector $\mathbf{F}^{\text{int}}$, can be written as

$$F_i^{\text{int}} = -\frac{\partial E}{\partial q_i}. \tag{3.3}$$

In accordance with Newton's second law, the system of equations of motion is then

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{F}^{\text{int}} + \mathbf{F}^{\text{ext}}, \tag{3.4}$$

where $\mathbf{M}$ is the diagonal mass matrix, acceleration $\ddot{\mathbf{q}}$ is the second derivative of the DOFs with respect to time, and $\mathbf{F}^{\text{ext}}$ is the external force vector (contact forces, gravity, etc.). For the purposes of this thesis,

$$\mathbf{F}^{\text{ext}} = \mathbf{F}^{c} + \mathbf{F}^{f} + \mathbf{F}^{v}, \tag{3.5}$$

where $\mathbf{F}^{c}$ is the contact force, $\mathbf{F}^{f}$ is the friction force, and $\mathbf{F}^{v}$ is the viscous force. The formulation of these three forces will be discussed in the remainder of this chapter.

In DER, the backward Euler method is used to solve the $4N - 1$ equations of motion in order to update the DOF vector $\mathbf{q}$. Discretizing in time, with time step Δt (i.e., $t_{i+1} = t_i + \Delta t$), (3.4) can be approximated by the following system of discrete equations:

$$\frac{\mathbf{M}}{\Delta t}\left[\frac{\mathbf{q}(t_{i+1}) - \mathbf{q}(t_i)}{\Delta t} - \mathbf{v}(t_i)\right] - \mathbf{F}^{\text{int}}(t_{i+1}) - \mathbf{F}^{\text{ext}}(t_{i+1}) = 0, \tag{3.6}$$

where $\mathbf{q}(t_i)$ and $\mathbf{v}(t_i)$ are the DOFs and their velocities, respectively, at time step t_i. The current DOFs $\mathbf{q}(t_i)$ and velocities $\mathbf{v}(t_i)$ are known and the task at hand is to compute the new DOFs $\mathbf{q}(t_{i+1})$ and velocities

$$\mathbf{v}(t_{i+1}) = \frac{\mathbf{q}(t_{i+1}) - \mathbf{q}(t_i)}{\Delta t}. \tag{3.7}$$

As the $(4N-1) \times (4N-1)$ Jacobian matrix $\mathbf{J}$, whose elements are

$$J_{ij} = \frac{m_i}{\Delta t^2}\delta_{ij} + \frac{\partial^2 E}{\partial q_i \partial q_j} - \frac{\partial F_i^c}{\partial q_j} - \frac{\partial F_i^f}{\partial q_j} - \frac{\partial F_i^v}{\partial q_j}, \tag{3.8}$$

can be computed, the Newton-Raphson method is used to solve for $\mathbf{q}(t_{i+1})$ iteratively. Note that the last three terms on the RHS of (3.8) are square matrices $\mathbf{J}^c, \mathbf{J}^f$, and $\mathbf{J}^v$ representing the gradients of the three external forces—contact, friction, and viscous. The Newton-Raphson iteration is

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} - \alpha \mathbf{J}^{-1}\mathbf{F}, \tag{3.9}$$

where n is the iteration number, $\mathbf{F}$ is the sum of forces (LHS of Eq. 3.6), and α is the Newton damping coefficient (explained further in Section 3.4). Once $\mathbf{q}(t_{i+1})$ is known, $\mathbf{v}(t_{i+1})$ is computed using (3.7).

Normally, if the gradient of an external force, $\partial \mathbf{F}_i^{\text{ext}}/\partial q_j$, cannot be analytically evaluated, this term is omitted during Newton iterations and that external force is considered "explicitly" (forward Euler). This generally requires reducing the time step size Δt, resulting in increased computation time. Later, we will show that for IMC, the contact and friction Jacobians are analytically obtainable. The implicit treatment of contact and friction is a key contribution of this thesis.

3.2 Implicit Contact Model (IMC) Formulation

Referring to Figure 3.1(b), let us denote $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_j, \mathbf{x}_{j+1} \in \mathbb{R}^3$ as the Cartesian nodal coordinates of edges i and j in a discrete rod configuration. Next, we denote an edge "combination" as the 12-dimensional vector concatenation $\mathbf{x}_{ij} = [\mathbf{x}_i^T, \mathbf{x}_{i+1}^T, \mathbf{x}_j^T, \mathbf{x}_{j+1}^T]^T$ and the set of all valid edge combinations as $\mathcal{X}$, where two consecutive edges are always in contact and these combinations are excluded from $\mathcal{X}$.

Hence, an arbitrary edge combination is denoted simply as $\mathbf{x} \in \mathcal{X}$ and all edge combinations are assumed to be valid. The contact energy E^c is then expressed as a differentiable analytic expression involving the four nodes of the two contacting edges; i.e., $E^c(\mathbf{x}) : \mathbb{R}^{12} \to \mathbb{R}$. Under this formulation, the proposed contact energy is dependent only on the nodal coordinates of the discretized rod and not on the twist angles θ_i of the edges, because the contact forces are computed based on the minimum distance Δ between two contacting edges.

Then, in order to calculate Δ, we utilize an efficient and accurate algorithm proposed by Lumelsky (1985) to compute the minimum distance between two finite line segments in N dimensions. Originally a piecewise function, the algorithm is then modified to become a twice differentiable smooth approximation. Using this completed expression, we can then obtain the negative gradient of the energy $-\nabla E^c(\mathbf{x}_{ij})$ as well as the negative Hessian $-\nabla^2 E^c(\mathbf{x}_{ij})$, which are then used to evaluate $\mathbf{F}^c$ in (3.5), hence (3.6), and $\mathbf{J}^c$ in (3.8). The gradient of $E^c(\mathbf{x})$ produces contact forces that act in the direction of the contact normal and whose magnitude varies with Δ, which results in physically realistic forces when dealing with rod-rod contact. Finally, as this method is essentially a penalty method, a stiffness parameter k_c is used to scale $\mathbf{F}^c$ and $\mathbf{J}^c$ appropriately.

By producing the contact forces in this way, dynamic friction can be calculated according to Coulomb's friction law. Previous methods (Choe et al., 2005; Spillmann and Teschner, 2008) are unable to simulate Coulomb friction due to the inability to obtain its Jacobian. As we have access to the normal contact force Jacobian, we can calculate the Coulombic friction forces and Jacobian. Having access to the Jacobians of the normal and friction force, our contact model reliably converges even in complex contact states. As the cost of producing the Jacobian is relatively high and leads to solving a non-banded matrix, we introduce a hybrid approach that ensures computational speed by only calculating the Jacobian when necessary.

To model contact energy, we compute the minimum distance Δ between two edges and feed this value into a smooth inverse ReLU function whose origin is based on the contact distance $2h$, where h is the cross-sectional radius of the rod:

$$E^c = \frac{\log(1 + e^{k_e(2h-\Delta)})}{k_e}, \tag{3.10}$$

with stiffness term k_e determining how sharp the curve is. Intuitively, this function starts to gradually increase the contact energy between two edges as Δ decreases while $\Delta > 2h$ and then sharply increases as Δ approaches $2h$. Although the gradients in the region $\Delta > 2h$ are nonzero, the inclusion of these "cushioning" forces greatly aid convergence and reduce any unwanted oscillatory behavior that can often occur in penalty methods. The effects of these nonzero gradients are explained in detail in Sections 4.3 and 5.1. Subsequently, the key component of the contact model involves obtaining a differentiable analytical expression for Δ, which is difficult as computing the minimum distance between two line segments is a highly nonlinear and noncontinuous process. Next, we briefly describe Lumelsky's min-distance algorithm as well as a new modified smooth approximation that will be used as $\Delta(\mathbf{x})$.

3.2.1 Computing the Minimum Distance Between Two Line Segments

Lumelsky's algorithm produces the minimum distance between two line segments in $\mathbb{R}^N$ and contains three noncontinuous components. We can eliminate one of them by simply making the assumption that no edge can be reduced to a point. In other words, each edge must have a finite length greater than zero. With this condition eliminated, we now briefly lay out the simplified min-distance algorithm for an arbitrary edge combination $\mathbf{x}_{ij}$. Note that here, we simply go over the steps of the algorithm. For further intuition on how exactly the algorithm is computing the min-distance, refer to the original paper (Lumelsky, 1985).

To start off the min-distance algorithm, first, we add "edge" vectors $\mathbf{e}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ to the ones already previously formulated: $\mathbf{e}_i$ and $\mathbf{e}_j$. With these vectors, we can then calculate the necessary intermediate values as follows, where i and j subscripts are omitted for clarity:

$$
\begin{aligned}
D_1 &= \mathbf{e}_i \cdot \mathbf{e}_i \\
D_2 &= \mathbf{e}_j \cdot \mathbf{e}_j \\
S_1 &= \mathbf{e}_i \cdot \mathbf{e}_{ij} \\
S_2 &= \mathbf{e}_j \cdot \mathbf{e}_{ij} \\
R &= \mathbf{e}_i \cdot \mathbf{e}_j \\
\lambda &= D_1 D_2 - R^2.
\end{aligned}
\tag{3.11}
$$

Next, let

$$
F(x) = \begin{cases} 0 & x < 0; \\ x & 0 \leq x \leq 1; \\ 1 & x > 1. \end{cases}
\tag{3.12}
$$

This function is the first of two remaining noncontinuous components. Let t, $u \in [0, 1]$ be the ratios that determine at which point along the length of each edge the connecting min-distance vector lies, as shown in Figure 3.1(b). Then, the fix bound function $F(x)$ is used to ensure that these values do not go outside the valid range. As two edges become increasingly parallel, λ approaches 0, the value for which they are perfectly parallel. To prevent division by zero, a piecewise function is used to describe the assignments of t:

$$
\begin{aligned}
t &:= \begin{cases} (S_1 D_2 - S_2 R)/\lambda & \lambda \neq 0; \\ 0, & \lambda = 0; \end{cases} \\
t &:= F(t); \\
u &:= (tR - S_2)/D_2; \\
u_f &:= F(u).
\end{aligned}
\tag{3.13}
$$

The final noncontinuous component is a conditional assignment where if $u_f \neq u$ (i.e.,

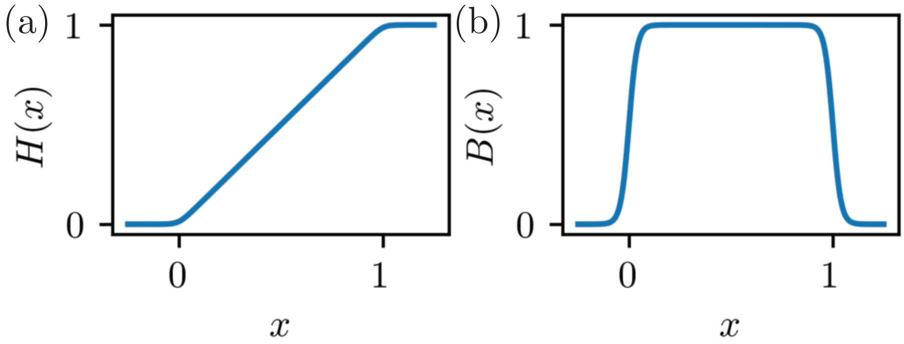

Figure 3.2: (a) Smooth approximated fixbound function $H(x)$, which models the piecewise function in (3.12). (b) Boxcar function $B(x)$, which allows for an analytical conditional reassignment. Both functions are plotted with a stiffness parameter of $k = 50$.

$u < 0$ or $u > 1$), the t value is reassigned as follows:

$$\begin{aligned} t &\coloneqq (u_f R + S_1)/D_1; \\ t &\coloneqq F(t); \\ u &\coloneqq u_f. \end{aligned} \tag{3.14}$$

Finally, Δ can be computed as

$$\Delta_{ij} = \left\| \mathbf{e}_i t - \mathbf{e}_j u - \mathbf{e}_{ij} \right\|. \tag{3.15}$$

3.2.2 Modifications Through Smooth Approximations

To obtain the gradient and Hessian of the contact energy $E^c(\Delta(\mathbf{x}))$, $\Delta(\mathbf{x})$ must be differentiable. In the previous section, we introduced an algorithm that can compute Δ. Now, we modify the min-distance algorithm into a differentiable analytical expression. As (3.11) is analytical, the only necessary modifications lie in (3.13) and (3.14). Firstly, the fixbound function $F(x)$ can be modeled with the smooth approximation

$$H(x) = \frac{\log\left(1 + e^{kx}\right) - \log\left(1 + e^{k(x-1)}\right)}{k}, \tag{3.16}$$

where k is a hyperparameter that determines how stiff the curves are. A larger k value will yield in a more accurate approximation but will result in “stiff” first and second derivatives leading to poorer convergence; thus, this value should be determined empirically. Next is the conditional reassignment in (3.14). As the reassignment depends only on whether or not $u_f \neq u$, this is equivalent to the reassignment only occurring when $u < 0$ or $u > 1$. To model this, we can use a boxcar function that consists of two compounded logistic functions:

$$B(x) = \frac{1}{1+e^{-kx}} - \frac{1}{1+e^{-k(x-1)}}. \tag{3.17}$$

Both functions $H(x)$ and $B(x)$ are plotted in Figure 3.2 for a value of $k = 50$, which is the value used to produce the simulation results. This k value is a good tradeoff between accuracy and reliable convergence.

The last noncontinuous component of the algorithm lies in the piecewise function in (3.13), which is actually left noncontinuous. Although this introduces a piecewise function into the expression, this does not hurt convergence for the following reasons. First, note that λ will almost never equal exactly 0 due to floating point arithmetic. Therefore, the piecewise function is only required to prevent simulation crashes for initial conditions with perfectly parallel rod configurations. Furthermore, the numeric stability of this algorithm is maintained as whenever λ approaches zero, the numerator $S_1D_2 - S_2R$ also approaches zero by a similar magnitude, effectively avoiding numeric overflow problems (Lumelsky, 1985).

With these two functions and the above prevention of division by zero, we can then replace (3.13) and (3.14) with the following expressions:

$$
\begin{aligned}
t_1 &= \begin{cases} (S_1 D_2 - S_2 R)/\lambda & \lambda \neq 0; \\ 0, & \lambda = 0; \end{cases} \\
t_2 &= H(t_1); \\
u_1 &= (t_2 R - S_2)/D_2; \\
u_2 &= H(u_1); \\
t_3 &= (1 - B(u_1))(u_2 R + S_1)/D_1 + B(u_1) t_2; \\
\Delta_{ij} &= \left\| \mathbf{e}_i t_3 - \mathbf{e}_j u_2 - \mathbf{e}_{ij} \right\|; \\
E^c_{ij} &= \frac{\log\left(1 + e^{k_c(2h - \Delta_{ij})}\right)}{k_c}.
\end{aligned}
\tag{3.18}
$$

3.2.3 Contact Energy, its Gradient, and its Hessian

As shown previously, the min-distance Δ is fed into (3.10), which then leads to a fully differentiable analytical expression $E^c(\mathbf{x})$. It should be noted that an end-to-end differentiation of $E^c(\mathbf{x})$ ends up with an extremely large and complex equation when using symbolic differentiation (Meurer et al., 2017). Therefore, to greatly simplify the expression and improve computational efficiency, the gradient and Hessian for several of the intermediary algorithmic values are computed and then chain ruled together. Effectively, we can define $E^c(\mathbf{x})$ by the following equivalent functional:

$$E^c(\mathbf{x}) = f(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_{ij}, D_1, D_2, S_1, S_2, R, t_2). \tag{3.19}$$

Since the arguments of f are all functions of $\mathbf{x}$, the chain rule tells us that we can obtain the gradient of the contact energy as follows:

$$\begin{aligned}\nabla E^c(\mathbf{x}) = &\frac{\partial f}{\partial \mathbf{e}_i}\frac{\partial \mathbf{e}_i}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{e}_j}\frac{\partial \mathbf{e}_j}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{e}_{ij}}\frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{x}} + \\ &\frac{\partial f}{\partial D_1}\frac{\partial D_1}{\partial \mathbf{x}} + \frac{\partial f}{\partial D_2}\frac{\partial D_2}{\partial \mathbf{x}} + \frac{\partial f}{\partial R}\frac{\partial R}{\partial \mathbf{x}} + \\ &\frac{\partial f}{\partial S_1}\frac{\partial S_1}{\partial \mathbf{x}} + \frac{\partial f}{\partial S_2}\frac{\partial S_2}{\partial \mathbf{x}} + \frac{\partial f}{\partial t}\frac{\partial t}{\partial \mathbf{x}}.\end{aligned} \tag{3.20}$$

Here, we see that for any arbitrary edge combination $\mathbf{x}$, the produced force $-\nabla E^c(\mathbf{x})$ will be a 12-dimensional vector consisting of four concatenated 3-dimensional contact force vectors for every node comprising $\mathbf{x}$. These 12 elements contribute to the 12 entries of the $4N-1$ dimensional $\mathbf{F}^c$ vector located at the following positions: $4i-3, 4i-2, 4i-1, 4(i+1)-3, 4(i+1)-2, 4(i+1)-1, 4j-3, 4j-2, 4j-1, 4(j+1)-3, 4(j+1)-2, 4(j+1)-1$. Once the contact forces are computed for every contacting edge combination during a time step, the force values are accumulated in $\mathbf{F}^c$ and then incorporated into DER.

To obtain the Hessian, we simply take (3.20) and differentiate once again using the product rule to obtain

$$\begin{aligned}\nabla^2 E^c(\mathbf{x}) = &\left(\frac{\partial^2 f}{\partial \mathbf{e}_i \partial \mathbf{x}}\frac{\partial \mathbf{e}_i}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{e}_i}\frac{\partial^2 \mathbf{e}_i}{\partial^2 \mathbf{x}}\right) + \left(\frac{\partial^2 f}{\partial \mathbf{e}_j \partial \mathbf{x}}\frac{\partial \mathbf{e}_j}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{e}_j}\frac{\partial^2 \mathbf{e}_j}{\partial^2 \mathbf{x}}\right) + \\ &\left(\frac{\partial^2 f}{\partial \mathbf{e}_{ij} \partial \mathbf{x}}\frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{e}_{ij}}\frac{\partial^2 \mathbf{e}_{ij}}{\partial^2 \mathbf{x}}\right) + \left(\frac{\partial^2 f}{\partial D_1 \partial \mathbf{x}}\frac{\partial D_1}{\partial \mathbf{x}} + \frac{\partial f}{\partial D_1}\frac{\partial^2 D_1}{\partial^2 \mathbf{x}}\right) + \\ &\left(\frac{\partial^2 f}{\partial D_2 \partial \mathbf{x}}\frac{\partial D_2}{\partial \mathbf{x}} + \frac{\partial f}{\partial D_2}\frac{\partial^2 D_2}{\partial^2 \mathbf{x}}\right) + \left(\frac{\partial^2 f}{\partial R \partial \mathbf{x}}\frac{\partial R}{\partial \mathbf{x}} + \frac{\partial f}{\partial R}\frac{\partial^2 R}{\partial^2 \mathbf{x}}\right) + \\ &\left(\frac{\partial^2 f}{\partial S_1 \partial \mathbf{x}}\frac{\partial S_1}{\partial \mathbf{x}} + \frac{\partial f}{\partial S_1}\frac{\partial^2 S_1}{\partial^2 \mathbf{x}}\right) + \left(\frac{\partial^2 f}{\partial S_2 \partial \mathbf{x}}\frac{\partial S_2}{\partial \mathbf{x}} + \frac{\partial f}{\partial S_2}\frac{\partial^2 S_2}{\partial^2 \mathbf{x}}\right) + \\ &\left(\frac{\partial^2 f}{\partial t \partial \mathbf{x}}\frac{\partial t}{\partial \mathbf{x}} + \frac{\partial f}{\partial t}\frac{\partial^2 t}{\partial^2 \mathbf{x}}\right).\end{aligned} \tag{3.21}$$

Once obtained, the Hessian is added to the $(4N-1) \times (4N-1)$ Jacobian matrix $\mathbf{J}^c$ in a similar manner.

3.2.4 Friction and its Jacobian

Just as we obtained contact force vectors of size 12, we produce frictional forces in a similar manner where given an edge combination $\mathbf{x}_{ij}$, we compute a friction force vector $\mathbf{F}^f_{ij} \in \mathbb{R}^{12}$ according to Coulomb's dynamic friction law. For each edge combination, this 12-dimensional vector is accumulated to the appropriate entries of $\mathbf{F}^f$ (3.6).

Coulomb's friction law states as follows:

1. Frictional force is independent of velocity, and
2. $\left\|\mathbf{F}^f\right\| = \mu_k F^n$ during sliding,

where μ_k is the dynamic friction coefficient and F^n is the normal force (additional details appear later in this section).

From the contact model, we are able to derive $-\nabla E^c(\mathbf{x})$, which yields the normal contact forces $\mathbf{F}^c$. As mentioned, the gradient of the contact energy will always produce forces that are along the contact normal. Therefore, we can obtain F^n at edge i of the contact pair $\mathbf{x}_{ij}$ by simply summing the contact forces on nodes i and $i+1$: $F^n = \|\mathbf{F}^c_i + \mathbf{F}^c_{i+1}\|$. We can also use these contact forces to obtain the contact normal $\mathbf{n} = (\mathbf{F}^c_i + \mathbf{F}^c_{i+1})/F^n$. The direction of friction is then determined by the tangential relative velocity $\mathbf{v}^{\mathrm{T}}_{\mathrm{rel}}$ of edge i with respect to edge j. This can be obtained using the nodal velocities $\mathbf{v}_i, \mathbf{v}_{i+1}, \mathbf{v}_j$, and $\mathbf{v}_{j+1}$ as follows:

$$\begin{aligned}
\mathbf{v}^e_i &= \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_{i+1}); \\
\mathbf{v}^e_j &= \frac{1}{2}(\mathbf{v}_j + \mathbf{v}_{j+1}); \\
\mathbf{v}_{\mathrm{rel}} &= \mathbf{v}^e_i - \mathbf{v}^e_j; \\
\mathbf{v}^{\mathrm{T}}_{\mathrm{rel}} &= \mathbf{v}_{\mathrm{rel}} - (\mathbf{v}_{\mathrm{rel}} \cdot \mathbf{n})\mathbf{n}; \\
\hat{\mathbf{v}}^{\mathrm{T}}_{\mathrm{rel}} &= \mathbf{v}^{\mathrm{T}}_{\mathrm{rel}}/||\mathbf{v}^{\mathrm{T}}_{\mathrm{rel}}||.
\end{aligned} \tag{3.22}$$

We formulate the friction force on edge i using a modified form of Coulomb's friction equation:

$$\mathbf{F}_i^f = -\mu_k \gamma \hat{\mathbf{v}}_{\text{rel}}^{\text{T}} F^n, \tag{3.23}$$

where weight $\gamma \in [0,1]$ eliminates frictional forces between edges with extremely small relative velocities that can cause unwanted behavior. To maintain differentiability, it is obtained using a smoothed Heaviside step function with the tangential relative velocity as input:

$$\gamma = \frac{1}{1 + e^{-k(\mathbf{v}_{\text{rel}}^{\text{T}} - c)}}, \tag{3.24}$$

where k is the stiffness and c determines the limit for the step transition, which must take into consideration the scaling of the model as explained in Section 3.4. In our experiments, $k = 50$ and $c = 0.15$. We can do the same for edge j to obtain $\mathbf{F}_j^f$. The computed frictional forces are then equally distributed to each node and concatenated four times to form the final 12-dimensional friction vector

$$\mathbf{F}_{ij}^f = \left[\frac{1}{2}\mathbf{F}_i^{f^T}, \ \frac{1}{2}\mathbf{F}_i^{f^T}, \ \frac{1}{2}\mathbf{F}_j^{f^T}, \ \frac{1}{2}\mathbf{F}_j^{f^T}\right]^T. \tag{3.25}$$

Once these friction force vectors are computed for every contacting edge combination during a time step, we can then compute the friction force Jacobian matrix $\mathbf{J}_{ij}^f$ as well. To greatly simplify the friction Jacobian, we treat friction semi-explicitly by using the known velocities from the previous time step when computing (3.22), which results in only F^n being dependent on $\mathbf{x}$. Additionally, this semi-explicit formulation allows the friction direction to remain constant during Newton iterations, which improves convergence considerably when the hybrid algorithm is used (additional details in Section 3.4.3).

We can now see that the friction Jacobian for an arbitrary edge-edge contact can be calculated through a relatively simple chain rule procedure using the contact Hessian $\nabla^2 E^c(\mathbf{x})$ from (3.21). Noting that $\nabla^2 E^c(\mathbf{x})$ amounts to computing $\partial \mathbf{F}^c / \partial \mathbf{x}$, we can compute the friction Jacobian as

$$\nabla \mathbf{F}_i^f = \frac{\partial \mathbf{F}_i^f}{\partial \mathbf{F}_i^c}\frac{\partial \mathbf{F}_i^c}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}_i^f}{\partial \mathbf{F}_{i+1}^c}\frac{\partial \mathbf{F}_{i+1}^c}{\partial \mathbf{x}}. \tag{3.26}$$

These are then accumulated to $\mathbf{F}^f$ and $\mathbf{J}^f$ in exactly the same way as for the contact energy gradient and Hessian, as described in Section 3.2.3.

3.2.5 Friction Limitations

It should be noted that several simplifications were made for the friction model. Firstly, the relative velocities were computed using the midpoint of the edges rather than the contact points. Likewise, the friction forces were evenly distributed rather than being dependent on the contact points. This was done as it greatly simplifies the friction force Jacobian, leads to improved convergence, and does not have any noticeable effects so long as the rod is sufficiently discretized.

In terms of limitations, this method clearly does not enforce static friction and so should only be used for continuous sliding scenarios such as knot tying. Furthermore, friction occurring due to the rod twist θ is not modeled. When an edge undergoes enough twist and is in contact with a receiving edge, friction forces occur slightly off the centerline of this receiving edge. As our model assumes that all friction occurs precisely on the centerline and formulates all contact only using $\mathbf{x}$, such friction-twist coupling is neglected. Lastly, the produced friction forces can possibly overtake the pull forces when the pull speed is very low leading to unrealistic sliding in the opposite direction. This must be remedied by pulling at a sufficiently high speed.

3.3 Analysis of Approximation Consequences

With the contact energy gradient and hessian fully defined, we now briefly analyze the effects of the imposed approximations from (3.18). First, we can see that the approximations from Figure 3.2 only affect the barycentric coordinates of the pair of closest points on two segments, t and u, which can be seen in Figure 3.1(b). More specifically, the approximations smooth the piecewise transition that occurs when either t or u goes from being a point in between the edge ends to an edge end itself or vice versa. Given this, the area of contact we are most concerned about is this transition zone.

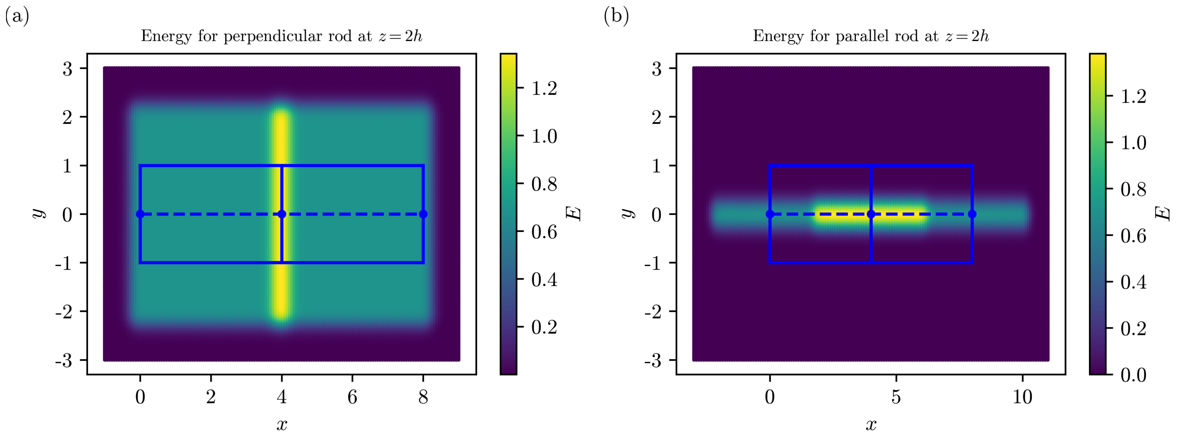

Figure 3.3: Contact potential heat map visualization. The blue rectangles indicate an edge i and edge $i+1$; they are adjacent edges that are aligned. The dash line indicates their centerlines and both have radii of 1. A heat map is then plotted of the energy intensities experienced by an edge j that moves translationally along the x and y-axes. Here, edge j is fixed in the z-direction at $2h$ from edges i and $i+1$ and the energy intensity is plotted with respect to the midpoint of edge j. The energy boundaries for when edge j is (a) perpendicular and (b) parallel to edges i and $i+1$ are shown.

To investigate whether any inconsistencies exist, we construct heat maps of the contact energy intensity experienced by an edge e when moved translationally along the x, y plane with respect to two other fixed adjacent aligned edges as shown in Figure 3.3. A heat map for when e is perpendicular and for when e is parallel with respect to the base edges are both shown. In a perfect system, we expect the energy experienced by edge e to smoothly taper off to zero once either t or u becomes an edge end and the distance between the edges increases.

By observing both Figures 3.3(a) and 3.3(b), we can qualitatively see that this is indeed the case. Note that the turqoise region represents the configurations for when a constant energy potential is experienced. In other words, it is the region of all configurations where t or u is an intermediary point along an edge. On the outer edge of this region, we then see that the turqoise transitions to blue indicating a drop in energy. Per our expectations, this change occurs a little after either t or u becomes an edge end. From the blue intensity region, increasing the distance between the edges results in the purple region indicating a near zero energy. From this examination, we can conclude that the approximation

Algorithm 1: Implicit Contact Method

```
Parameters: $k_c, \delta, \omega, S$
Input: $\mathbf{x}, \mathbf{v}, n$                                    // from DER
Output: $\mathbf{F}^c, \mathbf{J}^c, \mathbf{F}^f, \mathbf{J}^f, \alpha$
Function IMC($\mathbf{x}$, $\mathbf{v}$, $n$):
    scale $\mathbf{x}$ and $\mathbf{v}$ by $S$
    if $n == 0$ then                                   // run only on first iter
        $\mathcal{C}, m \leftarrow$ collisionDetection($\mathbf{x}, \delta$)
        $k_c \leftarrow$ updateConStiffness($k_c, m$)
    end
    if $n < \omega$ then                                  // compute only forces
        $\mathbf{F}^c \leftarrow$ genContact($\mathcal{C}$)
        $\mathbf{F}^f \leftarrow$ genFriction($\mathcal{C}, \mathbf{v}, \mathbf{F}^c$)
        $\mathbf{J}^c \leftarrow$ zero square matrix
        $\mathbf{J}^f \leftarrow$ zero square matrix
    end
    else                                // compute Jacobian for convergence
        $\mathbf{F}^c, \mathbf{J}^c \leftarrow$ genContact($\mathcal{C}$)
        $\mathbf{F}^f, \mathbf{J}^f \leftarrow$ genFriction($\mathcal{C}, \mathbf{v}, \mathbf{F}^c, \mathbf{J}^c$)
    end
    scale $\mathbf{F}^c, \mathbf{F}^f, \mathbf{J}^c, \mathbf{J}^f$ by $k_c$
    $\alpha \leftarrow$ newtonDamper($n$)
    return $\mathbf{F}^c, \mathbf{J}^c, \mathbf{F}^f, \mathbf{J}^f, \alpha$
```

functions do not affect the expected behavior between two contacting edges.

One thing to note is the energy doubling in both plots shown by the yellow region. For our task of knot tying, this energy doubling is not a noticeable issue as all contact cases occur along the braided region and are reasonably close to the parallel case. This allows the energy doubling to spread out more or less evenly. For cases concerning perpendicular configurations, this can possibly be an issue as an edge may experience an artificial bump in contact energy magnitude if the contact passes over a vertex of two connected edges. This can be handled by either filtering out such cases during collision detection or by assigning a regularization weight to the contact forces themselves.

Algorithm 2: Discrete Elastic Rods

```
Input: $\mathbf{q}(t_i), \mathbf{v}(t_i)$
Output: $\mathbf{q}(t_{i+1}), \mathbf{v}(t_{i+1})$
Require: boundary conditions $\rightarrow$ free
Function DER($\mathbf{q}(t_i), \mathbf{v}(t_i)$):
    Guess: $\mathbf{q}^{(1)} \leftarrow \mathbf{q}(t_i)$
    $n \leftarrow 0,\ \epsilon \leftarrow \infty$
    while $\epsilon > tolerance$ do
        $\mathbf{F}^{\text{int}} \leftarrow$ genForces($\cdot$)
        $\mathbf{J}^{\text{int}} \leftarrow$ genJacobian($\cdot$)                    // $\frac{\partial^2 E}{\partial q_i \partial q_j}$ in (3.8)
        $\mathbf{F}^c, \mathbf{J}^c, \mathbf{F}^f, \mathbf{J}^f, \alpha \leftarrow$ IMC($\mathbf{x}^{(n)}, \mathbf{v}(t_i), n$)          // Alg 1
        $\mathbf{F}^{\text{der}} \leftarrow$ left side of (3.6)
        $\mathbf{J}^{\text{der}} \leftarrow$ left side of (3.8)
        $\mathbf{F}^{\text{free}} \leftarrow \mathbf{F}^{\text{der}}$(free)        // downsize to only include free DOFs
        $\mathbf{J}^{\text{free}} \leftarrow \mathbf{J}^{\text{der}}$(free, free)
        $\Delta\mathbf{q}^{\text{free}} \leftarrow \mathbf{F}^{\text{free}}/\mathbf{J}^{\text{free}}$                  // solve $\mathbf{J}^{\text{free}}\Delta\mathbf{q}^{\text{free}} = \mathbf{F}^{\text{free}}$
        $\mathbf{q}^{(n+1)}$(free) $\leftarrow \mathbf{q}^{(n)}$(free) $- \alpha\Delta\mathbf{q}^{\text{free}}$
        $\epsilon \leftarrow \left\|\mathbf{F}^{\text{free}}\right\|$                              // update error
        $n \leftarrow n + 1$
    end
    $\mathbf{q}(t_{i+1}) \leftarrow \mathbf{x}^{(n)}$
    $\mathbf{v}(t_{i+1}) \leftarrow (\mathbf{q}(t_{i+1}) - \mathbf{q}(t_i))/\Delta t$
    return $\mathbf{q}(t_{i+1}), \mathbf{v}(t_{i+1})$
```

3.4 Full Algorithm

In addition to the force and Jacobian generation, there are several additional steps to the IMC algorithm that are explained in this section as well as several hyperparameters that must be properly tuned for optimal performance and convergence which are listed below.

1. δ, the collision limit,
2. k_c, the contact stiffness,
3. k_e, the contact energy stiffness, and
4. ω, the number of iterations before the hybrid algorithm computes the Jacobian

First, the nodal coordinates are scaled by a scaling factor $S = 1/h$ so that the adjusted rod radius equals a unit value of 1. To ensure that the distance at which two edges experience a force is very close to the rod surface, the following energy function is used:

$$E^c = \frac{\log\left(1 + e^{k_e\left(2 - \frac{\Delta}{h}\right)}\right)}{k_e}; \tag{3.27}$$

3.4.1 Choosing the Collision Limit

The collision limit δ is the threshold value used to determine when two edges are "in contact". This value is fed into a collision detection algorithm, which returns all edge combinations falling into this threshold, which is denoted by the set

$$\mathcal{C} = \left\{ \mathbf{x}_{ij} \in \mathcal{X} \mid \frac{\Delta_{ij}}{h} < 2 + \delta \right\}. \tag{3.28}$$

The minimum distance from this set is denoted by $m = \min_{\mathbf{x} \in \mathcal{C}} \Delta(\mathbf{x})$, which will be used later to adjust the contact stiffness k_c accordingly.

The collision limit δ must be chosen carefully as a higher δ value results in additional computation due to more qualifying edge combinations whereas a δ value that is too low will produce non-smooth gradients that hamper convergence.

A good way to determine a proper δ value is to observe the plotted contact energy function from (3.27) for a chosen k_e value. By observing the contact energy curve, the point at which the generated gradients are ≈ 0 can be found when the slope of the curve is nearly flattened out. Choosing a δ value that encompasses this region ensures that the generated gradients are sufficiently smooth. An example of this process can be found in Figure 3.4.

3.4.2 Stiffness Parameters

The stiffness of the contact forces is determined by the contact energy stiffness value k_e. As this value becomes higher, the region above the contact surface at which two edges experience a force decreases. This leads to stiff contact, which is more physically accurate as realistically the contact energy should be zero whenever $\Delta > 2h$ and shoot to ∞ as

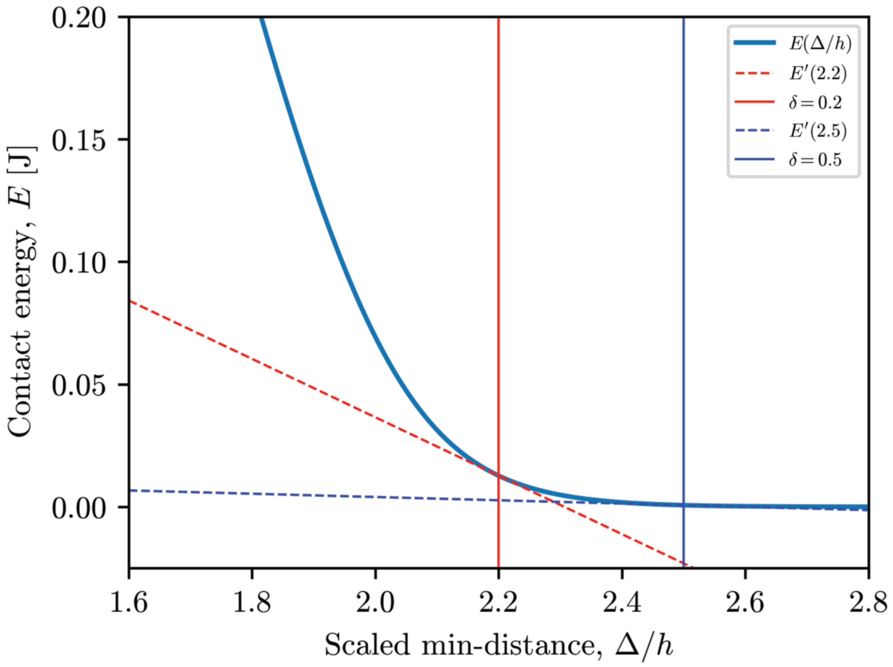

Figure 3.4: Contact energy curve of (3.27) for $k_e = 10$. Scaling by $S = 1/h$ results in the curve being centered at a collision length of $2h_s = 2.0$. As denoted by the solid blue line, the curve starts to flatten out to zero at $\Delta/h = 2.5$, which indicates that generated gradients are ≈ 0. Therefore, a collision limit $\delta = 0.5$ would be suitable. Conversely, $\delta = 0.2$ as denoted by the red line would result in the Newton solver potentially failing to converge due to non-smooth force generation.

soon as $\Delta = 2h$. On the other hand, a k_e value that is too large will result in convergence issues while a k_e value that is too low will have excessive oscillations between the contact bodies. A value that was found to be a good compromise between physical accuracy and convergence was $k_e = 50$.

The next hyperparameter that must be specified is the contact stiffness k_c. Not to be confused with the approximation stiffness k in (3.16) and (3.17), the contact stiffness k_c is a scalar value that is used to scale the contact force and Jacobian. This value is adaptively readjusted every time step to ensure that excessive hovering or penetration is minimized and so only the initial value must be specified, which can be found empirically. This initial k_c should be reasonably close to the value that would result in $m = 2h$. In our experiments, we employed a simple algorithm that decreased or increased k_c by a fraction of a percent depending on whether or not m was $< 2h - \epsilon_1$ or $> 2h + \epsilon_2$ where

ϵ_1 and ϵ_2 are limits indicating an acceptable contact range. This algorithm updates k_c only when m is deviating from the region defined by $[2h - \epsilon_1, 2h + \epsilon_2]$ and is otherwise left constant to prevent overshooting.

3.4.3 Hybrid Approach

As mentioned previously, this algorithm applies a hybrid approach in which the Jacobian of the contact and friction forces are only computed once the number of iterations passes a limit ω. This is done because often convergence can be quickly obtained even without the contact Jacobian and with the absence of the contact Jacobian, the overall Jacobian matrix remains a banded matrix, which can be solved significantly faster. Although the Jacobian results in a decrease in iteration count, the consequent increase in computational time outweighs this benefit as IMC is able to reliably converge rapidly without it for a majority of time steps.

With this in mind, the Jacobian is crucial for completing volatile contact states with high velocities and impacts that could otherwise end the simulation prematurely. Therefore, this hybrid approach maximizes computational speed while ensuring that the simulation can consistently reach the next time step during especially difficult contact scenarios such as inversion and the initialization phase where the rod rapidly reverts to its lowest energy state. The limit ω should be chosen empirically so that the Jacobian is only generated when necessary. In our experiments, we used an ω of 20.

3.4.4 Newton Damper and Algorithm Analysis

Finally, although not a hyperparameter, a damping coefficient α is used to reduce the step size of the Newton solver as the number of iterations increase for a particular time step. For this, a simple decaying algorithm is used which reduces α by a factor of 2 every other iteration.

Overall, aside from the collision detection algorithm (*which is only performed on the first iteration of every time step*), the time complexity of IMC with and without

Jacobian generation is $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ respectively, where n is the number of collisions detected. The full contact algorithm as well as its implementation in DER can be seen in Algorithm 1 and Algorithm 2, respectively. In Algorithm 2, the term "free" are the indices that correspond to the free degrees of freedom of the elastic rod. The remaining degrees of freedom are "fixed" and depends on the user defined boundary conditions.

CHAPTER 4

Results

In this section, we first validate the correctness of our contact model against theory. Then, to demonstrate the benefits of using IMC, we compare simulation results with the contact model SPT proposed by Spillmann and Teschner (2008) for knots of unknotting numbers $n \in \{1, 2, 3, 4\}$, which are shown in Figure 1.1. For both methods, the contact model is used to simulate knot tying until a mutual termination state, which is shown in the rightmost column of the figure. Finally, we compare the computational efficiency and convergence properties between the two methods. All simulations for IMC used a contact energy stiffness $k_e = 50$ and a collision limit $\delta = 0.15$, which was obtained using the method mentioned in Figure 3.4.

4.1 Damping for Stability

Since IMC is inherently a penalty method, the inclusion of damping forces greatly aid the stability of the contact model. A simple viscous damping force is applied to the elastic rod by applying on a node a force

$$\mathbf{F}^v = -\eta \left(\frac{\mathbf{q}(t_{i+1}) - \mathbf{q}(t_i)}{\Delta t} \right) dL, \tag{4.1}$$

where η (Pa $\cdot$ s) is a viscosity coefficient and dL is the node's Voronoi length.

The Jacobian of this damping force is

$$\mathbf{J}^v = -\frac{\eta dL}{\Delta t} \mathbb{I}, \tag{4.2}$$

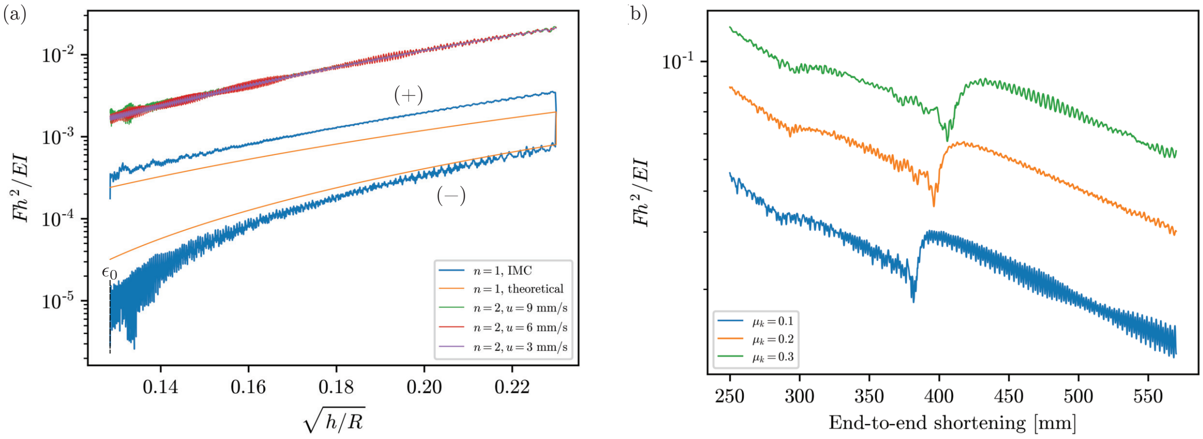

Figure 4.1: Model validation and comparison with theory. (a) In the top half is the pull force comparison for $n = 2$ with different pull speeds u and friction $\mu_k = 0.10$. As shown, the pull forces are identical in magnitude, indicating that friction is independent of velocity. In the bottom half is the $n = 1$ simulation data comparison with Audoly's predictive model shown in (4.3). Starting with an open trefoil knot, the knot is tightened and then loosened with $\mu_k = 0.10$, which is indicated by (+) and (−) respectively. A moving average of 200 steps is used for the n=1 pull forces to minimize visually large variations caused by the lower end of the log scale. (b) Pull force comparison for $n = 4$ and different μ_k values. There is a clear monotonically increasing relationship between the pull forces and μ_k. Additionally, inversion (indicated by the sudden drop in pull force) occurs earlier for higher μ_k values, as expected. A pull speed of 6 mm/s, 9 mm/s, and 12 mm/s was used for $\mu_k = 0.1$, 0.2, and 0.3, respectively. This was necessary as higher friction coefficients induced the limitation mentioned in Section 3.2.5 for higher pull speeds.

where $\mathbb{I}$ is the square identity matrix of size $(4N - 1)$. This damping force was also added to SPT for fair comparison. For all simulations, a viscosity value of $\eta = 0.01\,\text{Pa} \cdot \text{s}$ was used unless otherwise specified.

4.2 Theoretical Validation

Coulomb's friction law states that friction is independent of velocity. To show that our model abides by this, we plot the pull forces F in Newtons (N) when tightening a knot with unknotting number $n = 2$ with friction coefficient $\mu_k = 0.10$ for pull speeds u of 3, 6, and 9 mm/s (*pulled from both ends*). A 1 meter long rod with cross-sectional radius $h = 1.6$ mm, density $\rho = 1180$ kg/m^3, and Young's Modulus $E = 1.8e5$ Pa is used and is discretized into 301 nodes. We plot a regularized pull force Fh^2/EI against $\sqrt{h/R}$ where $EI = \frac{\pi E h^4}{4}$ is the flexural modulus and R is the radius of the knot loop. The radius R can be computed using the knot circumference from Figure 1.1 as $R = e/(2\pi)$. As Figure 4.1(a) shows, the magnitude of the $n = 2$ pull forces are approximately the same for all pull speeds. Dynamic friction force in Coulomb's model is independent of velocity and therefore this observation supports the physical correctness of the generated friction forces.

Next, the pull forces for a trefoil knot ($n = 1$) are compared with the predictive model of Audoly et al. (2007). This model states the theoretical equivalence

$$\frac{Fh^2}{EI} = \frac{\epsilon^4}{2} \pm \mu_k \sigma \epsilon^3, \tag{4.3}$$

where σ is a numerical constant ($\sigma = 0.492$ for trefoil knots), and $\epsilon = \sqrt{h/R}$. The ± term is the frictional component from tightening (+) and loosening (−). Using the same rod properties as mentioned, a trefoil knot is tightened and then loosened at 3 mm/s using IMC. Here we reduce η to 0.0005 as loosening can be sensitive to small forces. As shown in Figure 4.1(a), the recorded pull forces roughly follow the curves of the predictive model albeit with some displacements when tightening increases sufficiently. This can be attributed to imperfections in the predictive model as (4.3) is an elegant lightweight solution that does not perfectly model friction when the knot becomes sufficiently tight; i.e., when $\sqrt{h/R}$ becomes large. Still, aside from the displacement, the pull forces follow the rate of increase/decrease of the predictive model well, which is a good indicator of correctness.

Furthermore, in Figure 4.1(a), during the loosening, the trefoil knot is observed to be locked by friction at a point $\epsilon_0 = \sqrt{h/R_0} \approx 0.1285$. From the right side of (4.3), we can rearrange the terms to obtain the equivalence

$$\mu = \frac{1}{2\sigma}\epsilon_0 = 1.02\sqrt{h/R_0}. \tag{4.4}$$

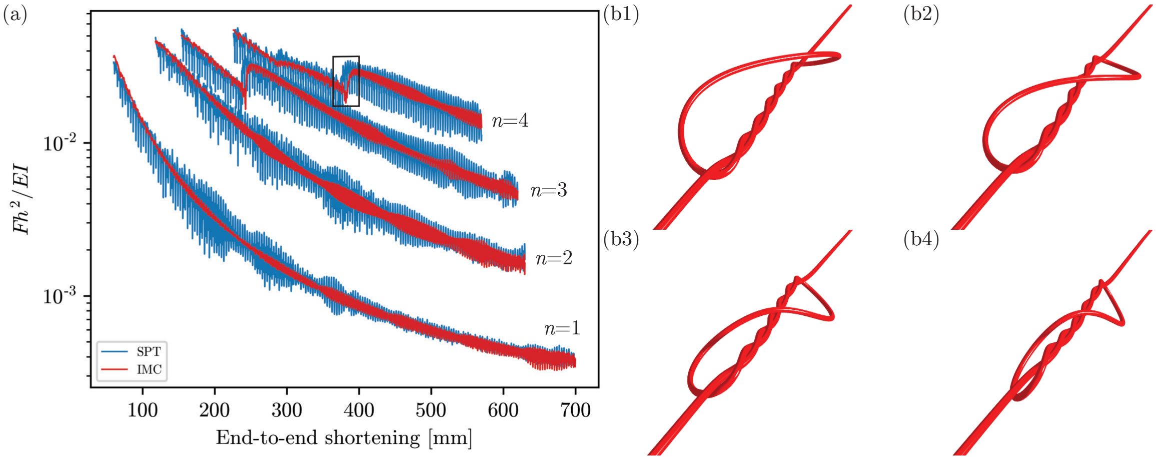

Figure 4.2: Simulation results comparison. (a) Pull force comparison for unknotting numbers 1 through 4 using SPT (Spillmann and Teschner, 2008) and IMC with $u =$ 6 mm/s, $\Delta t = 0.5$ ms, and $\mu_k = 0.10$. (b) Inversion occurring for $n = 4$ and the corresponding drop in pull forces shown by the border box in (a).

Plugging in our ϵ_0 value, we obtain $\mu_{\text{theory}} = 0.1306$, which is reasonably close to the friction coefficient $\mu_k = 0.10$ used in the simulation.

Finally, we show that the pull forces monotonically increase with μ_k in Figure 4.1(b). Here, we consider the $n = 4$ case. When recording the pull forces for μ_k of $0.1, 0.2,$ and 0.3, we can see that the rate of increase is held constant while the magnitudes monotonically increase. Furthermore, we can see that as friction increases, the inversion point occurs sooner, which indicates that the point of inversion is highly dependent on μ_k.

4.3 Pull Force Accuracy

Simulations of knot pulling were performed for both IMC and SPT using a pull speed $u = 6\,\text{mm/s}$, time step $\Delta t = 0.5$ ms, friction coefficient $\mu_k = 0.10$, and the same rod properties from Section 4.2. With these settings, knot tying was simulated for each method with the experienced pull forces being recorded at each time step. Figure 4.2(a) plots these pull forces with respect to the end-to-end shortening. Here we see that for both IMC and SPT, as the end-to-end shortening decreases (*knot is pulled tight*), the pull forces increase identically as expected. They also increase in magnitude as the unknotting number n goes up and for $n = 3$ and $n = 4$, inversion occurs, which is indicated by the sudden drop in pull force. This is shown visually in Figure 4.2(b) for $n = 4$.

The largest difference between the methods manifests in the considerable amount of force jittering by SPT, which occurs due to the too large time step. This leads to visually unrealistic results where the knot continuously "trembles" while being tied. On the other hand, IMC produces much smoother pull forces, which directly translate to visually smooth simulations for the same time step size. One caveat that should be noted is that SPT ensures exact non-penetration during contact whereas IMC allows small penetrations and hovering to occur, which is physically unrealistic. Still, through the adaptive contact stiffness and the inclusion of damping, IMC contains any hovering to stay within $20\,\mu\text{m}$ above the contact surface while penetrations rarely exceed $5\,\mu\text{m}$ (for comparison, the cross-sectional radius is $h = 1.6\,\text{mm}$). Thus, this minor error is largely indiscernible both visually and physically.

Overall, the enforcement of non-penetration at every time step limits the maximal time step that SPT can take without experiencing significant force jittering whereas IMC produces smooth results in exchange for contact varying within a small region.

4.4 Runtime Comparison

Next, we discuss run-time comparisons as well as convergence characteristics. Here, we show that in addition to IMC producing smoother results than SPT at sufficiently large time steps, IMC is also computationally competitive. Both models use the same DER implementation, which is written in C++. One discrepancy is that while SPT is directly implemented into DER in C++, IMC is written entirely in Python for ease of prototyping. To minimize any performance differences arising from using different languages, we employed a LLVM-based Python JIT compiler (Lam et al., 2015) for certain computational intensive portions of the code such as the chain rule procedures from (3.20), (3.21), and (3.26).

The computational time for each iteration for both contact methods is recorded using

Table 4.1: IMC vs. SPT (Spillmann and Teschner, 2008) run time data, $\mu_k = 0.10$, $\Delta t = 0.5\,ms$, pull speed $= 6\,\mathrm{mm/s}$. AIPT stands for average iterations per time step. ATPI stands for average time per iteration. Iters indicates the total number of Newton iterations that were necessary to complete the simulation. The time is the total computational time to completion.

Model	n	**AIPT**	**ATPI**[ms]	**Iters**	**Time**[s]
IMC	1	2.23	2.24	245824	551
	2	2.67	2.20	245830	542
	3	3.04	2.18	261707	570
	4	3.24	2.12	239987	509
SPT	1	8.25	2.30	907541	2085
	2	9.28	2.26	853935	1931
	3	10.25	2.26	881907	1990
	4	10.94	2.21	810160	1790

the ctime library. This timing was done so that any computational time expended in IO for recording the data and rendering the rod graphically were excluded. Note that the timings for IMC include all of the shared memory overhead between the C++ and Python programs. Therefore, a significant performance increase for IMC can be expected when it is fully implemented in C++ and compiled without this overhead. Finally, both methods used identical Newton tolerances for all simulations and all simulations were run in a single thread on an Intel Core i7-9700K 3.60GHz CPU.

In Table 4.1, the run time, total number of iterations, average iterations per time step (AIPT), and average time per iteration (ATPI) are reported. For all knots, we can immediately see that IMC converges in a noticeably smaller number of iterations than SPT. While the ATPIs of both methods are about equal, the simulations for IMC finish approximately 4× faster than SPT for all knots. For both methods, the AIPT increases as the knot complexity n increases, as expected.

We observed that the number of iterations to complete a time step increased for IMC as the knot loop became extremely small and/or tightly inverted. Dynamically reducing the time step solves this issue but this was excluded such that all reported results were for a constant time step. For the same time step size, SPT can more reliably converge when the knot loop becomes very small, albeit with the large force jittering still present.

Therefore, it may be worth investigating performance differences between the two methods when SPT uses a time step size that is small enough to compete with the smoothness of IMC while IMC starts with a larger time step and dynamically reduces it as convergence becomes an issue.

Still, even for a constant time step size of 0.5 ms, IMC is seen to be more computationally efficient when pulling the knots close to taut and for the majority of the knot tying procedure, takes far fewer iterations to converge. This difference should only increase with IMC being implemented directly into DER in C++.

CHAPTER 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we introduced a novel Implicit Contact Method (IMC) for Discrete Elastic Rod simulations in which the contact forces are handled implicitly using a smooth penalty force formulation. It was shown to be capable of modeling dynamic friction and accurately simulating knot tying. We showcased comparisons with previous methods (Spillmann and Teschner, 2008) and concluded that our method produces visually smoother realistic results while also producing physically accurate data. Furthermore, our method is stable, computationally competitive, and requites a minimal number of iterations to converge.

Although our method has shown promising results for knot tying simulations, it is not without its drawbacks. First, IMC is largely unsuitable for contact scenarios that frequently involve sudden excessively large velocities and impacts as these will result in excessive penetration and possible overshoot of the contacting body without appropriate damping. IMC shines, however, in scenarios with constant sliding contact such as in knot tying, where contact forces may or may not gradually rise.

Additionally, to be absolutely physically realistic, contact forces should equal zero when $\Delta > 2h$, whereas the usage of (3.10) produces a force $F^c \neq 0$ when $\Delta > 2h$. It has been shown that employing a smooth approximation such as this can greatly improve convergence in penalty methods (Durville, 2012), which is one of the goals of this algorithm. Because the contact forces approach zero as k_e becomes sufficiently large while $\Delta > 2h$, this is not a significant problem, as was discussed in Section 4.3. Still, the fact that F^c is not exactly zero is worthy of consideration. Lastly, the large number of hyperparameters leaves something to be desired as tuning may be necessary when switching between rod properties, which can be time consuming.

5.2 Future Research Directions

Some possible future research directions involve modifications that further improve the realism of the contact model. One of these pertains to the stiffness parameter k_c. A simplification that was employed is the usage of a global stiffness parameter. More realistic contact can be simulated using local stiffness parameters as shown previously by Durville (2012), but at the expense of more computation. This addition may also fix the problem where friction forces overtake the pull force of the knot if the pull speed is too low, as was mentioned in Section 3.2.5.

Another research direction is to extend IMC to non-circular cross-sections. IMC is currently only capable of handling circular cross-sections because we to compute the minimum distance between edges using only the centerlines. Extending IMC to non-circular cross-sections while maintaining differentiability is nontrivial, but that will be necessary to simulate rods of more complex shapes. Furthermore, handling more complex knots will require IMC to take into consideration cross-section deformation as IMC currently assumes that the cross-section shape remains constant.

Finally, another challenging problem that remains is the proper modeling of static friction. Although dynamic friction is adequately modeled, ultimately, subtle frictional threshold events such as the transition from sticking to sliding and vice-versa are necessary to simulate realistic contact outside the realm of constant sliding.

Alart, P. and Curnier, A. (1991). A mixed formulation for frictional contact problems prone to newton like solution methods. *Computer Methods in Applied Mechanics and Engineering*, 92(3):353–375. 4

Audoly, B., Clauvelin, N., Brun, P.-T., Bergou, M., Grinspun, E., and Wardetzky, M. (2013). A discrete geometric approach for simulating the dynamics of thin viscous threads. *Journal of Computational Physics*, 253:18–49. 1

Audoly, B., Clauvelin, N., and Neukirch, S. (2007). Elastic knots. *Physical Review Letters*, 99(16):164301. 1, 30

Audoly, B. and Pomeau, Y. (2010). *Elasticity and geometry: from hair curls to the non-linear response of shells*. Oxford University Press. 1

Baek, C. and Reis, P. M. (2019). Rigidity of hemispherical elastic gridshells under point load indentation. *Journal of the Mechanics and Physics of Solids*, 124:411–426. 1

Baek, C., Sageman-Furnas, A. O., Jawed, M. K., and Reis, P. M. (2018). Form finding in elastic gridshells. *Proceedings of the National Academy of Sciences*, 115(1):75–80. 1

Baraff, D. and Witkin, A. (1998). Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM. 1

Batty, C., Uribe, A., Audoly, B., and Grinspun, E. (2012). Discrete viscous sheets. *ACM Transactions on Graphics (TOG)*, 31(4):113. 1

Bergou, M., Audoly, B., Vouga, E., Wardetzky, M., and Grinspun, E. (2010). Discrete viscous threads. *ACM Transactions on Graphics (TOG)*, 29(4):116. 1, 2, 7

Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E. (2008). Discrete elastic rods. *ACM transactions on graphics (TOG)*, 27(3):63. 1, 2, 7

Bertails-Descoubes, F., Cadoux, F., Daviet, G., and Acary, V. (2011). A nonsmooth newton solver for capturing exact coulomb friction in fiber assemblies. *ACM Trans. Graph.*, 30:6:1–6:14. 4, 6

Choe, B., Choi, M. G., and Ko, H.-S. (2005). Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–160. 4, 6, 11

Clauvelin, N., Audoly, B., and Neukirch, S. (2009). Matched asymptotic expansions for twisted elastic knots: a self-contact problem with non-trivial contact topology. *Journal of the Mechanics and Physics of Solids*, 57(9):1623–1656. 1

Daviet, G., Bertails-Descoubes, F., and Boissieux, L. (2011). A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12. 4, 6

Durville, D. (2012). Contact-friction modeling within elastic beam assemblies: an application to knot tightening. *Computational Mechanics*, 49(6):687–707. 1, 4, 5, 35, 36

Goyal, S., Ruina, A., and Papadopoulos, J. (1991). Planar sliding with dry friction part 2. dynamics of motion. *Wear*, 143(2):331–352. 5

Grinspun, E., Hirani, A. N., Desbrun, M., and Schröder, P. (2003). Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67. Eurographics Association. 1

Jawed, M., Dieleman, P., Audoly, B., and Reis, P. M. (2015a). Untangling the mechanics and topology in the frictional response of long overhand elastic knots. *Physical review letters*, 115(11):118302. 1, 2

Jawed, M. and Reis, P. M. (2017). Dynamics of a flexible helical filament rotating in a viscous fluid near a rigid boundary. *Physical Review Fluids*, 2(3):034101. 1

Jawed, M. K., Brun, P.-T., and Reis, P. M. (2015b). A geometric model for the coiling of an elastic rod deployed onto a moving substrate. *Journal of Applied Mechanics*, 82(12):121007. 1

Jawed, M. K., Da, F., Joo, J., Grinspun, E., and Reis, P. M. (2014). Coiling of elastic rods on rigid substrates. *Proceedings of the National Academy of Sciences*, 111(41):14663–14668. 1

Jawed, M. K., Hadjiconstantinou, N., Parks, D., and Reis, P. (2018a). Patterns of carbon nanotubes by flow-directed deposition on substrates with architectured topographies. *Nano Lett.* 1

Jawed, M. K., Khouri, N. K., Da, F., Grinspun, E., and Reis, P. M. (2015c). Propulsion and instability of a flexible helical rod rotating in a viscous fluid. *Physical review letters*, 115(16):168101. 1

Jawed, M. K., Novelia, A., and O'Reilly, O. M. (2018b). *A Primer on the Kinematics of Discrete Elastic Rods*. Springer. 1, 7, 9

Jawed, M. K. and Reis, P. M. (2014). Pattern morphology in the elastic sewing machine. *Extreme Mechanics Letters*, 1:76–82. 1

Jawed, M. K. and Reis, P. M. (2016). Deformation of a soft helical filament in an axial flow at low reynolds number. *Soft Matter*, 12(6):1898–1905. 1

Kaufman, D. M., Tamstorf, R., Smith, B., Aubry, J.-M., and Grinspun, E. (2014). Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Transactions on Graphics (TOG)*, 33(4):1–12. 4, 5, 6

Kikuchi, N. and Oden, J. T. (1988). *Contact problems in elasticity: a study of variational inequalities and finite element methods.* SIAM. 5

Kirchhoff, G. (1859). Uber das gleichgewicht und die bewegung eines unendlich dunnen elastischen stabes. *J. Reine Angew. Math.*, 56:285–313. 4

Lam, S. K., Pitrou, A., and Seibert, S. (2015). Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, New York, NY, USA. Association for Computing Machinery. 32

Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., and Kaufman, D. M. (2020). Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics (TOG)*, 39(4). 4, 5, 6

Lumelsky, V. J. (1985). On fast computation of distance between line segments. *Inf. Process. Lett.*, 21:55–61. 11, 12, 15

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., and Scopatz, A. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103. 16

Moulton, D. E., Grandgeorge, P., and Neukirch, S. (2018). Stable elastic knots with no self-contact. *Journal of the Mechanics and Physics of Solids*, 116:33–53. 1

Panetta, J., Konaković-Luković, M., Isvoranu, F., Bouleau, E., and Pauly, M. (2019). X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)*, 38(4):83. 1

Patil, V. P., Sandt, J. D., Kolle, M., and Dunkel, J. (2020). Topological mechanics of knots and tangles. *Science*, 367(6473):71–75. 1, 4, 5, 6

Przybyl, S. and Pieranski, P. (2009). Tightening of the elastic overhand knot. *Physical Review E*, 79(3):031801. 1

Shen, Z., Huang, J., Chen, W., and Bao, H. (2015). Geometrically exact simulation of inextensible ribbon. In *Computer Graphics Forum*, volume 34, pages 145–154. Wiley Online Library. 1

Spillmann, J. and Teschner, M. (2008). An adaptive contact model for the robust simulation of knots. In *Computer Graphics Forum*, volume 27, pages 497–506. Wiley Online Library. 4, 6, 11, 28, 31, 33, 35

Terzopoulos, D. and Fleischer, K. (1988a). Deformable models. *The Visual Computer*, 4(6):306–331. 1

Terzopoulos, D. and Fleischer, K. (1988b). Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 88)*, pages 269–278. 1

Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 87)*, pages 205–214. 1