Inextensible Elastic Rods with Torsional Friction based on Lagrange Multipliers

Jonas Spillmann Matthias Harders
Computer Vision Laboratory - ETH Zurich
Sternwartstrasse 7, 8092 Zurich
tel. (+41) 44 632 57 69
email: jonas.spillmann@vision.ee.ethz.ch

Abstract

Elastic rods are thin flexible objects typically undergoing large non-linear deformations that cannot be modeled with linear methods. They are used in a number of research fields, e.g., to represent hair or ropes in animations, or catheters or needles in medical simulations.

In this paper, we propose a deformation model for inextensible elastic rods. The method of Lagrange multipliers is employed to enforce the inextensibility of the rod, and to couple the material frames with the centerline. The resulting system is banded, allowing for an efficient linear time solution.

We also propose a manifold projection method to incorporate the non-penetration constraints resulting from contact handling into our constrained Lagrangian mechanics.
problem. We further augment the contact model by treating torsional friction. This allows to reproduce friction effects such as dynamic rolling and twisting of rods. Various examples underline the benefits and applicability of our model.

**Keywords:** Deformation modeling, Elastic rods, Constrained Lagrangian mechanics, Friction

**Introduction**

The modeling of thin flexible objects – commonly denoted as *elastic rods* – is an active field of research in computer graphics and animation [1, 2]. The modeling of elastic rods is particularly challenging due to their large non-linear deformations that cannot be reproduced plausibly with linear models. Moreover, if the twist about the centerline, *i.e.*, the material deformation, is considered, then concepts such as rotation minimizing frames and parallel transport come into play. In addition, the dynamic simulation of elastic rods is complicated by the fact that many threads or ropes do not stretch noticeably. Therefore, the explicit numerical integration of the corresponding differential equations requires small time steps.

Elastic rods reveal their complex deformation behavior when they are involved in collisions. For example, if an elastic rod is squeezed between two surfaces which are then shifted relatively to each other, then the rod forms a loop (so-called plectonemes [1]), as illustrated in Fig. 1 a). This effect is induced by the torsional friction between the rod and the surfaces. This behavior can be reproduced by, *e.g.*, rubbing an elastic strap between the thumb and the index finger. The torsional friction plays an important role in the context of interacting elastic rods. For example, the stability of many knots relies largely on frictional effects. In addition, torsional friction is important in surgical needle insertion procedures, as pointed
out by Chentanez et al. [3].

The deformation model of Spillmann et al. employs quaternions as degrees-of-freedom (DOFs) to express the material frames [4], therefore allowing to model torsional friction and non-uniform twist (Fig. 1 b) in a straightforward way. However, since they employ the penalty method to both adapt the material frames to the centerline, and to enforce the inextensibility, they require a small simulation time step.

**Contribution** We present a dynamic model for the simulation of elastic rods that overcomes the aforementioned limitations of [4]. First, we solve the constraints that couple the material frames and the centerline by employing the method of Lagrange multipliers, resulting in a novel constrained Lagrangian mechanics formulation. In this context, we show that by an intelligent choice of the corresponding constraint functions, the coupling of the material frames, and the centerline inextensibility can be enforced simultaneously. This allows to take significantly larger time steps for simulating rods at very low stretch, compared to previous approaches.

Second, we show how to incorporate contact constraints into our constrained Lagrangian mechanics formulation by employing a manifold projection method. We further augment the contact handling by modeling torsional friction. As a result, we can reproduce frictional effects such as non-uniform twist and dynamic rolling. We show the application of our deformation model in interactive and off-line animations, including a simulation of catheter navigation.

**Organization** We start by reviewing related work in the field of elastic rod modeling. Then, an overview on the existing rod model is given. Later, the constrained Lagrangian mechanics formulation is introduced, and the incorporation of contact constraints is dis-
cussed. Finally, we evaluate our model and show applications in the field of animation and medical simulation.

**Background**

The modeling of elastic rods has a long research history in the field of computer graphics. Spline-based methods [5–7] are widely used and can even be employed to model complex contact configurations [8]. However, these approaches do not model material torsion. To some extent, this limitation is overcome by the approach of Theetten et al. that additionally considers roll DOFs [9]. Another class of methods treats the rod as a chain of linked rigid bodies. This allows to employ methods from the field of articulated rigid body modeling to simulate the rod [10, 11].

In the field of medical simulations, elastic rods are frequently used to represent threads, sutures or catheters. To model the elastic rod deformations, non-linear finite element (FE) methods [12–15], quasi-static approaches [16–18] or mass-spring models [19] have been investigated. Although these approaches can handle twisting deformations and perform well in practice [20], they require either small time steps due to the stiff underlying differential equations, or they cannot reproduce dynamic effects. An finite element rod deformation model in combination with an implicit solver is proposed in [21].

The Cosserat theory of elastic rods treats the rod as a one-dimensional but oriented curve in space. Pioneering work on Cosserat models has been done by Pai [22]. The approach of Bertails et al. reproduces the dynamics of clamped Cosserat rods [23]. Recently, Bertails has proposed a linear-time approach to compute the dynamics of super-helices based on Featherstone’s method [2]. The deformation model of Bergou et al., in contrast, builds upon
the theory of parallel transport to compute the elastic deformation forces [1]. They defer the stiffness to a post-integration step, which makes the simulation more stable. Recently, Chentanez et al. have employed their deformation model in the simulation of surgical needle insertion [3].

We base our deformation on the previous work of Spillmann and Teschner [4, 24, 25]. In contrast to their deformation model, we employ the method of Lagrange multipliers, which significantly improves the stability of the method. Thus, we can simulate elastic rods with very low stretch at significantly larger time steps.

One benefit of our deformation model is that it is based on a comparably simple theory. This is in contrast to [2, 23] which is considerably more difficult in terms of implementation, as mentioned in [26]. In addition, our approach can as well be employed to model rods with free boundaries while [2] is limited to clamped rods.

**Elastic rod deformation model**

We consider the radius of an elastic rod to be small compared to its length. Thus, we represent the rod centerline by a scalar function $r(\sigma) : [0, 1] \rightarrow \mathbb{R}^3$ where $\sigma$ is the line parameter. In contrast to the classical Frenet description, the Cosserat theory further considers the orientation of the rod cross-section, which is represented by an orthonormal frame $d_k(\sigma), k = 1, 2, 3$. The first two axes $d_1(\sigma)$ and $d_2(\sigma)$ give the orientation of the cross-section. The third axis $d_3(\sigma)$ is constrained to be parallel to the tangent $r'(\sigma) = \frac{\partial}{\partial \sigma} r(\sigma)$ of the centerline at $\sigma$. The orthonormal frame $d_k$ is also called the material frame. In the following text, we assume the rod to have a circular cross-section, since this type of rod is most often used in deformable modeling.
**Strains**  The deformation of elastic rods is described by the centerline strain $v$ and the material frame strain $u$ [27]. The centerline strain $v = (v_k)^T$ is related to shearing and stretching of the centerline. If we assume an unshearable and inextensible elastic rod, $v_1 = v_2 = 0$ and $v_3 = 1$. The strain $u = (u_k)^T$ in the material frame is related to the bending and twisting of the rod. Its components, expressed in the material frame, read as

$$u_k = \frac{1}{2} d_k \cdot \left( \sum_{k=1}^{3} d_k \times d'_k \right)$$

with $d'_k = \frac{\partial}{\partial \sigma} d_k$ the derivative of the material frame with respect to the line parameter $\sigma$. A derivation of this expression is given in [24].

**Elastic energy**  By assuming a linear stress-strain relationship, we can derive the elastic energy of the rod, based on the previously defined strain rates. Since we assume the rod to be inextensible, we do not need a stretching energy. If we further assume that the radius $r$ of the rod is small compared to its length, we obtain the bending energy $V_b$ as

$$V_b = \frac{1}{2} \int_0^1 \sum_{k=1}^{3} K_k (u_k - \hat{u}_k)^2 d\sigma$$

with $K_1 = K_2 = E \frac{r^2}{4}$ and $K_3 = G \frac{r^2}{2}$. Here, $E$ is the Young’s modulus governing the bending resistance, and $G$ is the shear modulus governing the torsional resistance. Moreover, the $\hat{u}_k$ are the intrinsic curvatures and twist of the rod. By setting $\hat{u}_k \neq 0$, rods with an intrinsic curvature or twist can be modeled.

**Parametrization of the centerline**  We discretize the centerline into $N$ spatial control points $r_i \in \mathbb{R}^3$ and $N - 1$ material frames $d_{k,i}$, each giving the orientation of the centerline element $(r_i, r_{i+1})$. 
Parametrization of the material frames  We employ unit quaternions $q_i = (q_1, q_2, q_3, q_4)^T$ to parameterize the material frames $d_{k,i}$. We use the convention that the $q_1, q_2$ and $q_3$ are the vector components of the quaternion, and $q_4$ is its scalar part. In order to express pure rotations, the quaternions must have unit length, i.e., $q_i^T q_i = 1$.

Having expressed the material frames in terms of the quaternion parameters, the elastic energy of the orientation element $(q_i, q_{i+1})$ is found to be

$$V_b[i] = \frac{l_i'}{2} \sum_{k=1}^{3} K_k \left( B_k (q_i + q_{i+1}) \cdot \frac{1}{l_i'} (q_{i+1} - q_i) - \hat{u}_k \right)^2$$

(3)

where the $B_k \in \mathbb{R}^{4 \times 4}$ are constant skew-symmetric matrices (see Appendix A). $l_i' = \frac{1}{2} (l_i + l_{i+1})$ is the Voronoi region of the point $r_i$ [1], with $l_i = \| r_{i+1} - r_i \|$. The elastic energy $V_b = \sum_i V_b[i]$ of the rod is obtained by summing the energy contributions of all elements. Of note is that adjacent elements with an in-between angle of more than $90^\circ$ tend to flip over. Thus, in order to represent high bending, the resolution must be chosen accordingly, which can be done by employing an adaptive scheme [28].

In addition, we compute internal and viscous damping forces acting on the centerline control points and material frames. We omit the formulas here and refer instead to [4].

Constrained model

Up to now, the elastic energy (3) governing bending and torsion depends exclusively on the configuration of the material frames. However, the resulting bending and twisting moments must be transferred to the centerline governing the spatial configuration of the rod. This mechanical coupling is established by constraining the third axis $d_3$ of the material frame to be parallel to the centerline element $(r_i, r_{i+1})$, thus neglecting shear in the model. This constraint must be implemented in a symmetric way such that a deformation of the centerline
imposes a change of the material frames which in turn transfer their elastic momenta back to the centerline points.

In [4], this constraint has been enforced with the penalty method, i. e., spring forces act on the points \( r_i \) and quaternions \( q_i \), aligning the material frames to the centerline. However, the drawback of this approach is twofold: First, the constraint is never fulfilled exactly, and second, the stiff penalty forces require a small time step.

These problems can be alleviated by solving the corresponding constrained Lagrangian mechanics (CLM) problem [29], i. e., to compute accelerations that prevent the violation of the constraints. The first step is to define the constraint functions that couple the material frames to the centerline. In addition, with an appropriate choice of these constraint functions, the inextensibility of the centerline can be enforced simultaneously. Consider the centerline element \( i = (r_i, r_{i+1}) \) whose material direction \( d_3(q_i) \) is parameterized with the quaternion \( q_i \). Then the constraint function \( C_{p}[i] \in \mathbb{R}^3 \) which couples the material frames to the centerline is

\[
C_{p}[i] := r_{i+1} - r_i - l_i d_3(q_i) = 0
\]  

Again, \( l_i \) is the resting length of the element \((r_i, r_{i+1})\), and \( d_3(q) = (2(q_1 q_3 + q_2 q_4), 2(q_2 q_3 - q_1 q_4), -q_1^2 - q_2^2 + q_3^2 + q_4^2)^T \) is the third axis of the material frame \( q \). Note that by employing the resting length (instead of the current length), we do not only enforce the coupling, but also \( \|r_{i+1} - r_i\| = l_i \), i. e., the inextensibility of the rod. This is because \( d_3 \) has always unit length. To enforce the quaternion unity constraint, we define an additional scalar-valued constraint function,

\[
C_{q}[i] := 1 - q_i^T q_i = 0
\]  

Overall, we end up with four scalar-valued constraints per centerline element. We refer to these constraints as internal constraints, opposed to external constraints such as contact con-
straints. We stack all constraints in a global constraint vector \( \mathbf{C} = (\mathbf{C}_p[i] \mathbf{C}_q[i])^T \in \mathbb{R}^{N_c}, i = 1, \ldots, N - 1 \), where \( N_c = 4(N - 1) \) is the total number of constraints of the rod.

**Constrained Lagrangian mechanics**

Having defined the constraints, we solve for the feasible accelerations that lie on the constraint manifold. To accomplish this, we employ the so-called \( \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \)-approach [30], see also [31]. First, we have to compose the mass matrix \( \mathbf{M} \) governing the moments of inertia of the rod (see Appendix B). We further collect all degrees of freedom in a global coordinate vector

\[
\mathbf{g} = (\mathbf{r}_1^T \mathbf{q}_1^T \mathbf{r}_2^T \cdots \mathbf{q}_{N-1}^T \mathbf{r}_N^T)^T \in \mathbb{R}^{N_g}
\]

where \( N_g = 3N + 4(N - 1) \) is the total number of degrees of freedom of our discrete rod model. This allows us to write the evolution of the rod in the form

\[
\mathbf{M}\ddot{\mathbf{g}} = \mathbf{f} + \mathbf{J}^T\lambda
\]

where the internal torques \( \nabla\mathbf{V}_b \), damping torques [4] and external forces are collected in the vector \( \mathbf{f} \). The term \( \mathbf{J}^T\lambda \) denotes the constraint forces that annihilate the acceleration contributions that violate the constraints. The matrix \( \mathbf{J} := \nabla \mathbf{C} \in \mathbb{R}^{N_c \times N_g} \) is the Jacobian, and \( \lambda \) is the Lagrange multiplier vector. The \( \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \)-approach computes the constraint forces \( \mathbf{J}^T\lambda \) by solving the system

\[
\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\lambda = -\dot{\mathbf{J}}\mathbf{g} - \mathbf{J}\mathbf{M}^{-1}\mathbf{f} - \mathbf{b}
\]

for the Lagrange multipliers \( \lambda \), with \( \mathbf{b} = 2\alpha \mathbf{C} + \beta^2 \mathbf{C} \) the Baumgarte stabilization term [32] reducing the constraint drift caused by the numerical time-integration. A derivation of (8) is found in, e. g., [33].
The parameters of the Baumgarte stabilization term control the amount of numerical drift in the constraint enforcement. Thus, larger values for $\alpha$ and $\beta$ reduce the drift better, but in turn degrade the performance since smaller time steps are required. Giving an analytical relation between the Baumgarte parameters and the time step is a hard problem for elastic rods. As a rule of thumb, choosing values of $\alpha$ and $\beta$ in the same magnitude as the stiffness tensor entries $K_k$ allow to take the same time step as if the Baumgarte parameters were zero (usually about 1ms, see results), and therefore results in a stable simulation.

**Computational aspects**

Without any simplifications, the system (8) is a dense linear system of dimension $N_c \times N_c$. The expected running time would therefore be in $O(N^3)$, which would make the approach inappropriate for real-time simulations. Fortunately, we can easily make the system banded, allowing for an efficient linear solve. This is accomplished by assuming that the masses of the rod are lumped in the nodes. With this simplification, the mass matrix becomes block-diagonal,

$$
M = \begin{pmatrix}
M_r[1] & 0 & 0 & \ldots & 0 \\
0 & M_q[1] & 0 & \ldots & 0 \\
0 & 0 & M_r[2] & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & M_r[N]
\end{pmatrix}
$$

(9)

where the $M_r[i] \in \mathbb{R}^{3 \times 3}$ are the constant and diagonal centerline node mass matrices, and the $M_q[i] \in \mathbb{R}^{4 \times 4}$ are the time-variant and dense material frame mass matrices. Since the inverse of a block-diagonal matrix is again a block-diagonal matrix, we can directly assemble the inverse mass matrix $M^{-1}$ from the inverse nodal mass matrices $M_{r}^{-1}[i]$ and $M_{q}^{-1}[i]$. The
\( M^{-1}_r[i] \) can be pre-computed, and the \( M^{-1}_q[i] \) can be determined directly.

To compute the matrix product \( JM^{-1}J^T \), we do not need to explicitly represent the banded Jacobian matrix \( J \). Instead, with some optimizations, the matrix product can be directly evaluated with only a small constant memory footprint. Since \( J \) is banded and \( M^{-1} \) is block-diagonal, the resulting matrix \( JM^{-1}J^T \in \mathbb{R}^{N_c \times N_c} \) is a quadratic banded matrix with a bandwidth of 9. To solve the linear system, we employ the band matrix solver of the WildMagic\(^1 \) library. The expected running time is \textit{bandwidth} times \textit{size} and therefore linear in the size of the rod.

**Numerical integration**

To evolve the rod in time, we have to time-discretize and solve the CLM problem (7). To accomplish this, we feed the product of the block-diagonal inverse mass matrix \( M^{-1} \) with the force sum \( \mathbf{f} + J^T \mathbf{\lambda} \) into an explicit Verlet scheme. Although the Verlet scheme becomes unstable for large time steps, we believe that explicit integrators have some advantages in the context of our elastic rod model. This is because the non-linear forces can be directly evaluated, which in turn allows to model the non-linear bending and twisting behavior. In contrast, by employing an implicit solver, the corresponding force gradients would need to be linearized, therefore multiple linearization steps per time steps would be necessary in order to account for non-linearities. Moreover, the numerical dissipation becomes dominant for larger time steps. Finally, our contact handling relies on small time steps, thus a cheap explicit solver is a good choice.

\(^1\)WildMagic 4.7, www.geometrictools.com
Contact constraints

Contact and friction play an important role in the simulation of elastic rods. In this section we briefly discuss the employed standard scheme for resolving collisions, followed by a description of the incorporation of the contact constraints into our constrained Lagrangian mechanics formulation.

Contact model

In order to detect the collisions and self-collisions, we follow a spatial subdivision approach. To compute the non-penetrating positions, we follow the approach described in [28], which is a Gauss-Seidel-like scheme in the spirit of [34, 35]. That is, we iterate over all pairs of colliding primitives and resolve each collision locally such that the resulting displacements preserve the momentum, which implies that the mass-weighted displacements sum to zero for each collision. By repeating this process until a given tolerance is reached, the displacements quickly propagate through the entire zone of impact [34]. In the end, the contact model returns the feasible, i.e., non-penetrating positions \( \mathbf{r}_{\text{coll}}^* \). This simple scheme allows to handle interactions between rods, rigid and deformable objects.

Modeling friction

Since we treat the material frames \( \mathbf{q}_i \) as dynamic DOFs of the rod elements, modeling the torsional friction is straightforward. For each pair of colliding primitives, we consider the one-dimensional contact space to be the minimum distance vector between the two segments.

Thereafter, the tangential velocities are computed. To this end, the sum of the centerline
and the material frame velocities is multiplied with the rod radius. The tangential velocities are then given by the components orthogonal to the contact space. Based on this, the relative velocity $v_r$ in the contact point is given as the difference of the two tangential velocity components.

The Coulomb sliding friction force is given as $f_s = \mu f_n \frac{v_r}{\|v_r\|}$, where $\mu$ is the sliding friction coefficient and $f_n$ is the normal force. If the relative velocity is above a threshold $\|v_{\text{roll}}\|$, then the sliding friction force $f_s$ is exerted on the centerline. If the relative velocity is below the threshold, then no friction is exerted on the centerline, since the rolling resistance is usually negligible [36]. In our experiments, we use $\|v_{\text{roll}}\| = 10^{-4} \text{m/s}$. Note that the threshold is necessary to prevent rods that have commenced rolling from decelerating by the friction. While this simple friction model is sufficient for plausible and efficient simulation of sliding and rolling contacts, a more elaborate model which solves the inequality of the Coulomb law for dry friction could be employed to improve the accuracy.

Moreover, a torsional friction momentum is exerted on the material frames, which is given by $\tau_t = 2Q_{\text{coll}} \begin{pmatrix} f_s \times c \\ 0 \end{pmatrix}$, where $Q_{\text{coll}}$ is the conjugate quaternion matrix [37] of the quaternion $q_{\text{coll}}$ and $c$ the radius vector of the rod at the contact point (see Fig. 2). Although no friction is exerted on the centerline for small relative velocities, the application of a torsional friction momentum on the material frames allows to reproduce static friction states.

This procedure is applied in each iteration of the search process for the non-penetrating positions. Since the contact model is position-based, we compute a sliding friction displacement by numerically integrating the corresponding sliding friction force $f_s$ by a time step $h$. Then, the friction displacements are added to the collision displacements, which allows for a consistent position-based contact handling. In contrast, the torsional friction torques $\tau_t$
are handled at the force-level, and therefore a separate treatment is required, which will be detailed in the following.

**Manifold projection**

To incorporate the feasible (i.e., non-penetrating) centerline control points \( \mathbf{r}^*_{\text{coll}} \) into our constrained Lagrangian mechanics formulation, we follow a class of numerical integration schemes that are commonly referred to as manifold projection methods [29]. A manifold projection method evolves the unconstrained mechanical system forward in time, and then enforces the constraints by projecting the intermediate result onto the constraint manifold. A prominent representative is the fast projection method of Goldenthal et al. [38].

We propose a modified version of manifold projection to incorporate the feasible positions into our formulation. We first solve the CLM formulation (7) and then evolve only the centerline control points forward in time to obtain intermediate control points \( \mathbf{r}(t+h) \). Then, we detect the collisions and feed the colliding points \( \mathbf{r}_{\text{coll}}(t+h) \) into the contact model. The computation of the non-penetrating points \( \mathbf{r}^*_{\text{coll}}(t+h) \) corresponds to projecting the centerline onto the constraint manifold imposed by the contact constraints. We finally update the collided points with \( \mathbf{r}_{\text{coll}}(t+h) \leftarrow \mathbf{r}^*_{\text{coll}}(t+h) \) and \( \mathbf{r}_{\text{coll}}(t+h) \leftarrow h^{-1}(\mathbf{r}^*_{\text{coll}}(t+h) - \mathbf{r}_{\text{coll}}(t)) \), and the non-collided points with \( \mathbf{r}_{\text{free}}(t+h) \leftarrow \mathbf{r}_{\text{free}}(t+h) \) and \( \mathbf{r}_{\text{free}}(t+h) \leftarrow h^{-1}(\mathbf{r}_{\text{free}}(t+h) - \mathbf{r}_{\text{free}}(t)) \).

To incorporate the friction torques acting on the material frames, we compute the material frame accelerations \( \mathbf{\ddot{q}} \) as

\[
\mathbf{\ddot{q}} = \mathbf{M}_q^{-1}(\mathbf{f}_q + \mathbf{J}_q^T \mathbf{\lambda}_q + \mathbf{\tau}_f)
\]  

(10)

with \( \mathbf{M}_q \) the composite quaternion mass-matrix, and \( \mathbf{\tau}_f \) the friction torques. With the syntax
[\cdot]_q$, we refer to the material frame components of the respective vector quantities. Notice that all quantities on the right hand side of (10) are known, therefore the computation of $\ddot{\mathbf{q}}$ is merely a matrix-vector product. To evolve the material frames in time, we again employ an explicit Verlet scheme.

Summarizing, our approach proceeds as follows: We first solve the CLM problem (7) and evolve only the centerline control points in time, resulting in the intermediate centerline $\mathbf{r}(t + h)$. Then, we compute the feasible positions $\mathbf{r}^*(t + h)$, i.e., we project the collided points onto the constraint manifold imposed by the non-penetration constraints. Based on this result, we update the centerline to obtain the feasible positions $\mathbf{r}(t + h)$. Then, we compute the material frame accelerations $\ddot{\mathbf{q}}(t)$ by evaluating (10), and finally evolve the material frames in time to obtain the subsequent rod configuration $\mathbf{g}(t + h)$.

**Discussion**

Our approach evaluates the direction $\mathbf{J}$ of the internal constraints at the beginning of the time step. In contrast, implicit constraint direction schemes such as the fast projection method [38] evaluate the constraint direction at the end of the time step, which has shown to be favorable in the context of velocity-level constraint enforcement. However, since the friction force according to the Coulomb friction model depends on the internal forces, a force-level handling of the constraints appears to be the more natural choice for elastic rod modeling. In addition, we avoid an iterative process to enforce the constraints.

In contrast to the internal constraints, the non-penetration constraints are evaluated at the end of the time step, and handled at the velocity-level. In doing so, we account for their discontinuous nature. However, our approach comes at the cost of drift in the internal constraints, resulting from two sources of inaccuracy: First, we evaluate the internal con-
straints at the beginning of the time step, thus neglecting the non-linearity of the constraint manifold. Second, we update the positions of the collided points at the end of the time step, which thus partially violate the internal constraints. Consequently, we must employ the Baumgarte stabilization term in order to avoid that the internal constraints diverge. Still, our two-stage manifold projection method keeps the divergence small, as illustrated by the examples below. Finding appropriate Baumgarte parameters is currently done manually. Formal guidelines for selecting these are currently under investigation.

Results

In this section, we evaluate the performance of our deformation model for elastic rods, and show the benefits compared to a penalty-based elastic rod model. Moreover, we present examples in interactive and off-line animations, and further show the integration of the model into a prototype interventional radiology simulator. All experiments have been carried out on an Intel Pentium Core 2 running at 3GHz.

Inextensibility One of the key benefits of the Lagrangian multiplier approach is that we can simulate rods with significantly less stretch at similar performance, compared to the previous models based on the penalty-method. To illustrate this, we simulate two physical seconds of a falling chain represented by an elastic rod discretized into 40 elements. We perform this experiment with different time steps, and record the maximum stretch during this period. The Baumgarte parameters are set to the maximum value that allow for stable simulation of the dynamics with the given time step (smaller Baumgarte parameters do not influence the stability of the simulation, but the rods become more stretchy). Of note is that the maximum stretch (see Tab. 1) depends quadratically on the time step, illustrating the
second order accuracy of our scheme. The measurements indicate that simulating the rod with a time step of $1 \cdot 10^{-3}$s results in less than $1.4 \cdot 10^{-4}$% maximum stretch. In contrast, in order to achieve less than $1.4 \cdot 10^{-4}$% stretch with a penalty-based model [4] requires a large stretching resistance and thus a smaller time step of $1.5 \cdot 10^{-5}$s. Thus, that simulation is almost 50 times slower, and therefore less viable for real time simulation.

**Inextensibility with collisions**  
We have proposed to compute the constraint forces of the internal constraints at $t$, but to update the collided points at time $t+h$. Consequently, the resulting configuration violates the internal constraints, and the Baumgarte term, serving as a spring-damper force, is necessary to prevent that the internal constraints diverge. However, the inaccuracy should be comparatively small, since we do not project the unconstrained points on the contact constraint manifold. Instead, we step the internal constraints forward in time, and feed the resulting intermediate centerline into the contact handling. To illustrate that the manifold projection keeps the stretch low in collisions, we simulate a rigid body wrapped in a clamped chain discretized into 30 elements (see Fig. 1 c). The rigid body falls under the influence of gravity, with about 200 collisions on average. We simulate one physical second (which is the time needed to uncoil the chain) and record the maximum stretch over this period. The Baumgarte parameters are again set to ensure stability. The measurements (see Tab. 2) indicate that the maximum stretch is larger, compared to the collision-free simulation (see above). Still, the simulation is about 10 times faster with our model, compared to the same experiment performed with a penalty-based approach.

**Performance**  
In order to assess the performance of the dynamic simulation, we again carry out the experiment (see Fig. 1 c). This experiment is particularly illustrative since it involves both dynamic movements, collisions with other objects, and self-collisions. We
simulate one physical second and average the time measurements. During the experiment, the average number of collisions is 200, with a peak of 268 collisions. In average, two iterations are sufficient to resolve the collisions, with a worst-case of 28 iterations. Summarizing, the computation of the elastic forces takes 0.025ms, solving for the Lagrange multipliers takes 0.49ms, the Verlet integration takes 0.002ms, detecting the collisions takes 2.36ms, and computing the feasible configuration takes 1.25ms. In total, one simulation step takes 4.15ms. From the measurements, it follows that the time to detect the collisions dominates the costs, which is due to the complexity of the scene that inhibits an early pruning of primitives. Since we employ an explicit integration scheme, the cost for the temporal evolution is almost negligible.

**Torsional friction** To illustrate the effects of torsional friction that can be reproduced, we simulate a U-shaped elastic rod on an inclined plane. The time step of the numerical integration is \( h = 5 \cdot 10^{-4} \)s. In Fig. 3 a), the rod is in a stable equilibrium, governed by a mild torsional deformation of its two arms, its torsional resistance, and the gravitational force. In Fig. 3 b), a heavy object is laid onto the rod, which induces a continued torsional deformation. In turn, the rod starts to curl, as depicted in Fig. 3 c) and d). Of note is the non-uniform twist in Fig. 3 d). There is no twist in the contact region between the rod and the object, but a significant twist in the curling part of the rod. This is because the friction resistance prevents that twist waves propagate instantaneously through the entire rod.

The deformation model can also be employed in interactive animations. In Fig. 4, an elastic rod discretized in 60 elements is curled by a user by navigating a virtual hand. The time step is \( h = 5 \cdot 10^{-3} \)s, and the simulation runs with an average of 20 frames/second.
Catheter navigation  We have integrated our elastic rod model into a prototype interventional radiology simulator. In this minimally-invasive approach, the physician navigates a catheter through the vessel tree. If a junction is reached, the appropriate branch to continue is selected by twisting the root of the tool and flexing the tip. The torsional momentum causes a rotation of the bent tip of the catheter. The goal of this experiment is to underline the employ of our deformation model in medical applications. Fig. 5 (a-b) shows snapshots of the catheter navigation simulation. The catheter is discretized into 80 elements, and the vessel into 400 vertices and 380 triangles. To compute one frame of the simulation, three simulation steps with a time step $h = 1 \cdot 10^{-2}$s are performed. The real-time simulation runs at 60 frames per second, including collision handling and generation of simulated X-ray images.

Coil embolization of brain aneurysms  Aneurysms are pathological dilations in the vessel walls that disturb the blood flow. They can be treated, for instance, by inserting flexible coils into the lumen, thus reducing the risk of rupture [21]. The coils are discretized into 40 elements each. The material parameters are taken from [21]. The blood vessel is a triangulated MRI scan with 430 vertices. An example of coil insertion in our simulation is shown in Fig. 5 (c-d). The computation of one frame takes about 44ms, with 7ms required for the simulated X-ray images. For each simulation step, 1.9ms is spent for computing the elastic forces, 0.3ms for solving the CLM (8) and (7), and 0.06ms for the numerical integration. The detection and handling of about 700 collisions takes 8.25ms. The remaining time is spent on managing the simulation and handling the user input. Therefore, the simulation runs at 22 frames per second, which is still interactive. Currently, we simulate only the catheter, although the model could be extended to represent a pair of catheter and
guidewire, as described in [20].

**Conclusion**

In this paper, we have presented a deformation model for elastic rods which overcomes the limitations of the previous model [4]. Instead of using the penalty method, we employed a constrained Lagrangian mechanics formulation to couple the centerline and the material frames. Moreover, the inextensibility of the rod is handled simultaneously by the same mechanism. To evolve the rod in time, we have employed a constrained Verlet scheme. This allows to simulate rods with very low stretch at significantly larger time steps.

Further, we have suggested how to make a position-based contact handling approach compatible with the Lagrangian formulation. The resulting approach is related to manifold projection methods [29, 38]. Moreover, we model torsional friction, which allows to reproduce complex deformation patterns. We have further underlined the wide applicability of our deformation model by showing an example application in the field of interventional radiology.

As future work, we plan to extend our deformation model to rods with non-circular cross-sections. This imposes further challenges in the context of contact handling, especially when dynamic rolling is considered. In addition, we investigate into an automatized process to choose optimal Baumgarte parameters. Further, it would be interesting to study deformation along the cross-section in order to generalize the model.
Acknowledgements

This work has been performed within the frame of the Swiss National Center of Competence in Research on Computer Aided and Image Guided Medical Interventions (NCCR CO-ME) supported by the Swiss National Science Foundation. The authors would like to thank Evren Samur of the LSR, EPFL for providing the X-ray simulation.

References


Assembling the mass matrix

To assemble the global mass matrix, we first compute the mass matrices per element. This can be accomplished in many ways such as with the Lagrange formalism. The Lagrange
formalism [39] first sets up the Lagrangian of a dynamic system, which is defined as the difference between kinetic energy $T$ and potential energy $V$. The equations of motion including the mass matrix of the system are obtained by substituting the Lagrangian into Lagrange equations of motion,

$$\frac{d}{dt} \left( \frac{\partial (T - V)}{\partial \dot{g}} \right) - \frac{\partial (T - V)}{\partial g} = 0$$

(11)

where the vector $g$ collects the generalized coordinates. Since we are only interested in obtaining the dynamic values, the potential energy $V$ can be neglected.

Since the material frames and the centerline are not coupled directly but by constraints, we can derive the mass matrices of the centerline elements and the quaternions separately. The mass matrix $M$ of the whole system is then obtained by summing up the contributions of the centerline elements and quaternions.

**Centerline elemental mass matrix** To obtain the mass matrices of the centerline elements, we first derive their kinetic energy. Following the finite element (FE) approach, we obtain the kinetic energy per element as

$$T_r[i] = \frac{1}{2} \rho \pi r^2 \dot{r}[i] \cdot \left( \int_0^{l_i} N^T N d\xi \right) \dot{r}[i]$$

(12)

with $r[i] = (r_i^T, r_{i+1}^T)^T \in \mathbb{R}^6$ the six-dimensional node vector of the element $i$, and $N = (I(l_i - \xi), I\xi) \in \mathbb{R}^{3\times6}$ the linear interpolation matrix. The centerline mass contributions are obtained by first symbolically differentiating $T_r[i]$ with respect to $\dot{r}[i]$, then with respect to time, and then collecting the terms that linearly depend on $\ddot{r}[i]$,

$$\frac{d}{dt} \left( \frac{\partial T_r[i]}{\partial \dot{r}[i]} \right) = M_r[i]\dot{r}[i]$$

(13)
The symbolic evaluation can be done by hand, or, more convenient, by employing a symbolic computer algebra program such as Maple\textsuperscript{2}. The mass matrices of the centerline elements are constant and can be pre-computed. Since the elemental mass matrices depend on both \( \mathbf{r}_i \) and \( \mathbf{r}_{i+1} \), their composition will not be diagonal.

**Quaternion mass matrix** To obtain the kinetic energy of the quaternion \( \mathbf{q}_i \), we first have to obtain its angular velocity \( \omega_i = (\omega_{k,i})^T \in \mathbb{R}^4 \). With some quaternion algebra, \( \omega_{k,i} = 2\mathbf{B}_k \mathbf{q}_i^T \dot{\mathbf{q}}_i, k = 1, 2, 3, 4 \). Details are found in [4]. We further define the inertia tensor of the material frames as \( \mathbf{J} = \text{diag}(J_1, J_2, J_3, J_4) \), with \( J_1 = J_2 = \frac{\rho \pi r^2}{4}, J_3 = J_1 + J_2 \) and the (virtual) parameter \( J_4 = J_1 + J_2 + J_3 \) (see [40]). Then the kinetic energy of the \( i \)th quaternion is simply

\[
T_q[i] = \frac{1}{2} l_i \omega_i^T \mathbf{J} \omega_i
\]

Although the numeric value of \( J_4 \) has no influence on the dynamics, it must be carried along the symbolic evaluation of the Lagrange formalism in order to avoid a singular quaternion mass matrix. Again, the quaternion mass matrices are obtained by first symbolically differentiating \( T_q[i] \) with respect to \( \dot{\mathbf{q}}_i \), then with respect to time, and then collecting the terms that linearly depend on \( \ddot{\mathbf{q}}_i \),

\[
\frac{d}{dt} \left( \frac{\partial T_q[i]}{\partial \dot{\mathbf{q}}_i} \right) = \mathbf{M}_q[i] \dddot{\mathbf{q}}_i
\]

Unlike the masses, the moments of inertia are time variant. Therefore, the quaternion mass matrices must be assembled in each time step.

\textsuperscript{2}Maple v9.5, www.maplesoft.com
**Jonas Spillmann** received his PhD degree in Computer Science from University of Freiburg in 2008, where he was a member of Professor Matthias Teschner’s research group. Dr. Spillmann is currently a Post-doctoral fellow at Computer Vision Laboratory at the Swiss Federal Institute of Technology, where he is working with Dr. Harders on interactive surgical simulation.

**Matthias Harders** studied Computer Science with focus on Medical Informatics at the University of Hildesheim, Technical University of Braunschweig, and University of Houston. He completed his doctoral thesis in 2002 and his habilitation in 2007 at ETH Zurich. Currently, he is lecturer and senior researcher at the Computer Vision Lab of ETH, and leader of the Virtual Reality in Medicine Group. His current research concerns surgical simulation and computer haptics. He is a founder of the IEEE RAS/CS Haptics Technical Committee, the IEEE Transactions on Haptics, the EuroHaptics conference and society. He is also the co-founder of the ETH spin-off company VirtaMed.
Figure 1: These images illustrate the benefits and applications of our elastic rod deformation model. a) By modeling torsional friction, we can reproduce the twisting deformation that results from squeezing an elastic rod between obstacles and shifting them relative to each other. b) A twisted rope is attached to poles with two knots. While the freely hanging ends untwist, the knots prevent the middle part of the rope to do so. This results in a non-uniform distribution of twist. c) By combining a position-based contact handling approach with our Lagrangian rod model, we can simulate two-way coupling between rods and rigid bodies. The rigid body falls down under the influence of gravity, thereby uncoiling the chain. In turn, the friction between the chain and the object induces a spinning motion of the object.
Figure 2: The torsional friction momentum is $f_t \times c$, where $c$ is the radius vector of the rod at the contact point, and $f_t$ is the Coulomb sliding friction working against the centerline velocity $v_{\text{centerline}}$. In this case, this results in a clockwise rolling velocity $v_{\text{roll}}$. 
<table>
<thead>
<tr>
<th>Model</th>
<th>Time step [s]</th>
<th>Time [s]</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>Max. stretch [%]</th>
<th>Real time</th>
</tr>
</thead>
<tbody>
<tr>
<td>New model</td>
<td>$1 \cdot 10^{-2}$</td>
<td>0.093</td>
<td>40</td>
<td>150</td>
<td>$1.37 \cdot 10^{-2}$</td>
<td>yes</td>
</tr>
<tr>
<td>&quot;</td>
<td>$1 \cdot 10^{-3}$</td>
<td>0.892</td>
<td>400</td>
<td>1500</td>
<td>$1.37 \cdot 10^{-4}$</td>
<td>yes</td>
</tr>
<tr>
<td>&quot;</td>
<td>$1 \cdot 10^{-4}$</td>
<td>9.008</td>
<td>4000</td>
<td>15000</td>
<td>$1.36 \cdot 10^{-6}$</td>
<td>no</td>
</tr>
<tr>
<td>Penalty</td>
<td>$1.5 \cdot 10^{-5}$</td>
<td>43.39</td>
<td>-</td>
<td>-</td>
<td>$1.36 \cdot 10^{-4}$</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 1: This table summarizes the time to simulate two physical seconds of a falling chain discretized into 40 elements. $\alpha$ and $\beta$ are the Baumgarte parameters that are set to the maximum value that allow for the stable simulation of the rod dynamics with the given time step. Further, the maximum stretch over this period is shown. Since the accuracy of the scheme is of order 2, the maximum stretch decreases quadratically for smaller time steps. The measurements indicate that a rod with $1.37 \cdot 10^{-4}\%$ stretch can still be simulated in real time. In contrast, simulating the rod with less than $1.37 \cdot 10^{-4}\%$ stretch by employing a penalty-based model requires a large stretching resistance of $E_s = 50$ GPa, and therefore a small time step of $1.5 \cdot 10^{-5}$ s. Thus, the simulation is almost 50 times slower, as the bottom row indicates.
Table 2: This table summarizes the time to simulate one physical seconds of a falling rigid body wrapped in a chain (Fig. 1 c) represented with our deformation model. If a time step of $h = 1 \cdot 10^{-3}$ s is employed, then a maximum stretch of 0.48% is measured, i.e., significantly more stretch than in the collision-free setting. In contrast, if we employ the penalty method to enforce the inextensibility, then as much as 7.37% is measured, with the stretching resistance set to the maximum value allowing for stable simulation. To reduce the stretch to 0.5%, a larger stretching resistance and thus a smaller time step has to be chosen, and the simulation is about ten times slower (due to the contact handling, the time does not scale linearly with the time step).
Figure 3: This experiment illustrates the interplay between torsional friction and curling deformation. In a), the rod is in a resting state. A heavy object is laid onto it which starts to slide b). This induces a curling deformation c)-d). This result is due to the torsional friction the object exerts.
Figure 4: An interactive animation with torsional friction. The user controls the virtual hand, and induces a torsional deformation of the rod. The simulation runs at 20 frames/second.
Figure 5: (a-b) In interventional radiology, the physician drives a catheter through the vessel system by pushing and twisting its root. The catheter is discretized into 80 elements, and the real-time simulation runs at 60 frames per second. (c-d) Our deformation model can also be employed to represent the catheter and coils in a coil embolization procedure. First, the catheter is navigated to the aneurysm location (c). Then, the coils are driven through the catheter and placed within the aneurysm (d). The simulation runs at 22 frames per second, including the handling of 700 collisions.