Abstract—This paper presents a new method for modelling the dynamic behaviour of developable ribbons, two dimensional strips with much smaller width than length. Instead of approximating such surface with a general triangle mesh, we characterize it by a set of creases and bending angles across them. This representation allows the developability to be satisfied everywhere while still leaves enough degree of freedom to represent salient global deformation. We show how the potential and kinetic energies can be properly discretized in this configuration space and time integrated in a fully implicit manner. The result is a dynamic simulator with several desirable features: We can model non-trivial deformation using much fewer elements than conventional FEM method. It is stable under extreme deformation, external force or large timestep size. And we can readily handle various user constraints in Euclidean space.

Index Terms—Developable Surface, Ribbon Simulation, Reduced Configuration

1 INTRODUCTION

Developable surfaces are ubiquitous in our daily life. Although their continuous properties have been well understood [1], their accurate modelling and discretization is still an open problem. Recently, methods have been proposed in [2], [3] to model these surfaces statically. In this paper, we takes a step further to model the dynamic properties of developable ribbons, a special type of developable surfaces that can be isometrically mapped to two-dimensional strips with much smaller width (latitude dimension) than length (longitude dimension). Developable ribbons have seen a lot of applications for modelling stylish hairs, satin bows or films.

FEM solver is clearly a competitive solution for modelling developable surfaces, using either conforming [4] or non-conforming triangle meshes [5]. However, none of these methods are geometrically accurate in that their configuration space is not a subset of the true developable shape space. As a result, on conforming meshes large stiffness energies have to be introduced to limit the stretch, which in turn leads to the locking phenomena. On the other hand, the hard length constraints in [6] on non-conforming meshes greatly limit the timestep size. Moreover, the reconstructed conforming meshes are again not exactly developable and usually suffer from noisy perturbation.

Key to our dynamic ribbon simulator is a novel configuration space that parameterizes a subset of the developable shape space which is large enough to cover most non-trivial deformations. Specifically, we describe the shape of ribbon by a set of creases along the centerline and bending angles across them, see Fig. 1. This method is in direct contrary to previous reduced models such as [7], where various constrains are introduced to pull the shape towards the true shape space. These constraints usually lead to stability issue or locking phenomena. Instead, our novel representation guarantees that the ribbon can be isometrically mapped to material space. As a result, no additional constraints are needed, making our method stable under large external force or timestep size.

Moreover, the whole timestepping scheme can be formulated as a single optimization, which greatly simplifies implementation. We noticed that a similar idea has been exploited in [8], [9] for modelling helical rod.

Under this configuration space, we present a proper discretization of the kinetic and potential energies. Our discretization scheme bears several desirable features: First, material space remeshing and world space deformations are modelled uniformly; Energy gradients can be analytically evaluated allowing quasi-newton method to converge efficiently; The $\mathbb{R}^3$ vertex positions $x_j, y_j$ are reintroduced as auxiliary variables so that conventional collision handlers can be trivially port to our new formulation. In conclusion, our contributions can be summarized as follows:

- A novel parameterization of salient global deformations of developable ribbon.
- A discrete timestepping scheme that can be efficiently integrated in a fully implicit manner.
- An optimization-based framework for multi-ribbon simulation allowing flexible user constraints and collision resolution.

The rest of the paper is organized as follows. After briefly reviewing the related works, we first describe the transfer function between $\mathbb{R}^3$ and our configuration space in section 3. We then present our discretization scheme for the kinetic and potential energies in section 4. Finally, in section 5, we go into some implementation details of our optimization strategy, constrain and collision handling before we conclude our discussion.

2 RELATED WORKS

Rod Modelling The theory of elasticity for 1D rod has been established in [10]. Later on, various discretization scheme for
Fig. 1. Illustration of our configuration space. The centerline of a ribbon is evenly segmented into \( n \) elements separated by creases. The angles between these creases and the centerline are \( \theta_i \) and the final shape of the ribbon in \( \mathbb{R}^3 \) is reconstructed by bending along these creases. The corresponding bending angles are \( \psi_i \). Given these parameters \( \theta_i \) and \( \psi_i \), the two ends of each crease, \( x_f \) and \( y_f \), can then be derived analytically.

Fig. 2. Compared with our method (top), extending a twisted rod (red) along binormal directions (top) doesn’t give isometric deformation (area error=4.7%).

Finite Element Shells Finite element method is another promising alternative for modelling thin shells, see [16] for a description of their continuous model. The discrete counterpart has been introduced into the graphics community in [17], [18]. But these methods model elastic, instead of developable shells. Later, it is shown in [19] that isometric deformation can be approximated on a conforming mesh by enforcing hard length constraints in a post-projection step. Moreover, this method has the good property that bending energies become quadratic [20]. Although in this work we adopt the dihedral angle based formulations following [17], [18], our bending energies are also quadratic since we treat the bending angles as our generalized coordinates.

However, although a developable surface can be approximated using [19], a triangular conforming mesh has insufficient degrees of freedom to cover the developable shape space, leading to the so-called locking phenomena. This problem is resolved in [6] by enforcing the length constraints on a non-conforming mesh. However, [6] suffers from noisy vertex perturbation when a conforming mesh is reconstruction for rendering and collision resolution. Also, the fast-projection involved in [19], [6] greatly limits the timestep size and stability. Compared with these methods, our formulation allows the same or even higher accuracy with much less elements since no discretization along the latitude direction is needed.

Developable Surface Modelling Our method is also closely related to previous efforts towards static developable surfaces modelling. Among these works, [21] describes a rectifying developable surface by the envelop of its rectifying planes. However, their method cannot be directly used for dynamic modelling since the centerline is represented by an Bézier curve, on which inextensible and other consistency constraints are hard to formulate. Developable surfaces have also been known in the community of architectural design as PQ meshes [22]. Like [13], developability in their method depends on a set of nonlinear constraints to be satisfied. Finally, our representation is most closely related to [3], where a developable surface is characterized explicitly by creases and bending angles. But we emphasize that, since the correct dynamic behaviour of a developable ribbon heavily depends on the material space crease direction changes, [3] cannot be directly used to this end because their crease directions are hard to parameterize.

3 Ruling-Bending Coordinates

One unique property of a developable surface \( S \) is that the Gaussian Curvature is zero everywhere. As a result, each point on \( S \) is attached to a ruling line or crease along which normal is constant and the ribbon is bended by angle \( \psi \). As is noted in [3], the world space shape of \( S \) can be characterized by these crease directions and bending angles up to rigid transformation. In our formulation, these two sets of parameters define our configuration space. However, the crease directions for a general developable surface is hard to parameterize. Fortunately, we have observed that for \( S \) with much larger longitude then latitude dimension, a large subset
of salient deformations can be modelled with only creases that pass through the ribbon centerline, see Fig. 3 for an illustration. We can then parameterize our crease direction by 

\[ c = \tan(\theta), \]

where \( \theta \) is the angle between the crease and the centerline.

For \( T_0 \), if the first segment of the ribbon is fixed, we simply have \( T_0 = I_d \) as well. While if the ribbon is attached to a floating frame, \( T_0 \) is a global rigid transformation:

\[ T_0 = \begin{pmatrix} \exp^w & t \\ 0 & 1 \end{pmatrix}, \]

parameterized by rotation vector \( w \) and translation \( t \). If this is the case, our configuration space is parameterized by the set of variables \( <c, \psi, w, t> \). For numerical optimization, our dynamic simulator heavily depends on an analytical formula for \( \nabla x_j = \partial x_j / \partial <c, \psi, w, t> \). We leave their derivations to Appendix A. \( \nabla y_j \) can be found following the same procedure.

Fig. 3. Non-trivial global deformations (brown) can be modelled using only creases that cross the centerline. However, some local deformations (red) is excluded from our shape space.

Specifically, given \( S \) with longitude dimension \( l \) and latitude dimension \( w \), we first segment its centerline into \( n \) elements \( E_{0 \ldots n-1} \). Then, between any two consecutive elements we introduce creases with directions \( c_{1 \ldots n-1} \) and bending angles \( \psi_{1 \ldots n-1} \). Since we don’t allow singularity points, any two consecutive creases cannot intersection, leading to a set of crease constraints:

\[ |c_j - c_{j+1}| \leq \Delta c_{\text{max}}, \quad |c_0| \leq \Delta c_{\text{max}}, \quad |c_{n-1}| \leq \Delta c_{\text{max}}, \quad (1) \]

where \( 1 \leq j < n - 1 \) and \( \Delta c_{\text{max}} = \frac{\pi}{180} \). In practice, we set \( \Delta c_{\text{max}} = 0.95 \frac{\pi}{180} \) to avoid degenerate triangles in the reconstructed mesh for collision handling. When these constraints are satisfied, our configuration space is thus parameterized by \( <c, \psi_j> \).

Like [8], a reconstruction procedure is needed to recover \( \mathbb{R}^3 \) positions of bottom rim vertices \( x_0, \ldots, n \) and top rim vertices \( y_0, \ldots, n \). The material space positions of these vertices are:

\[ \tilde{x}_j = <\frac{l_j}{n} - \frac{wc_j}{2}, \frac{w}{2}, 0, 1>, \quad \tilde{y}_j = <\frac{l_j}{n} + \frac{wc_j}{2}, \frac{w}{2}, 0, 1>, \]

where \( n \) need to introduce boundary crease directions \( c_0 = c_n = 0 \). Here we used homogeneous coordinates for convenience. Their world space positions \( x_j \) and \( y_j \) can then be derived by applying a series of crease transformations:

\[ x_j = \left[ \Pi_{i=0}^j T_j \right] \tilde{x}_j, \quad y_j = \left[ \Pi_{i=0}^j T_j \right] \tilde{y}_j, \quad (2) \]

where \( T_j \) is a rigid rotation by angle \( \psi_j \) along crease \( c_j \). In this equation, we again need to introduce boundary value \( T_n = I_d \).

In this work, since the configuration space is a subset of the true shape space, no stiffness energies are need to limit stretch and we are left with bending energies to be considered. Since we have the bending angles as an independent variable, bending energies based on dihedral angles [4, 18] become quadratic in \( \psi \) in our case. Specifically, for each crease between \( E_{i-1}, E_i \), we introduce:

\[ V_i^{\text{bend}} = \int_{D_i} H_i^2 dx \approx \frac{nw(1 + c_i^2) \psi_i^2}{l}, \]

where \( D_i \) is a diamond element between \( E_{i-1}, E_i \) with area \( lw/n \). This formula is derived in a similar way to [17]. The mean curvature measure of \( D_i \) is \( H_i = w \sqrt{1 + c_i^2} \psi_i \), which follows from the tube theory [24] and the mean curvature is then approximated in a area averaged manner. Although [3] adopted a more accurate form, taking the change of mean curvature along ruling into consideration, the difference is insignificant compared with our simplified form.

Other potential terms are discrete version of external forces or soft constraints. For example, the gravitational potential
energies are simply: $V^{\text{grav}} = -\rho g^T M \mathbf{x}^{n+1}$. In addition to these terms, we add a small regularization to resolve the ambiguity of crease directions for a flat ribbon, preferring ruling directions orthogonal to the centerline. The final $V$ is:

$$V = V^{\text{grav}} + \sum_{i=1}^{n-1} (\alpha V^\text{bend}_i + \beta c_i^2) + V^{\text{user}},$$

where $\alpha$ is the bending stiffness coefficient and $\beta = 0.1$ in all our examples. We also added an additional term $V^{\text{user}}$ for user controllability.

On the other hand, our compact configuration space poses great challenge on spatial discretization of the kinetic term. This is because $c, \psi$ encodes material space remeshing (by changing $c$) and world space deformation in a uniform manner. As a result, $\mathbb{R}^3$ positions $\mathbf{x}^{n+1}$ and $\mathbf{x}^n$ may not correspond to the same points $\tilde{\mathbf{x}}^{n+1}$ and $\tilde{\mathbf{x}}^n$ in material space and cannot be subtracted directly. A common practice here is to temporarily fix $c$ to ensure material space consistency and perform remeshing regularly. Although this strategy has been successfully adopted in conventional FEM shell solver such as [25], it fails to work with our method because our configuration space is so compact that bending angles along cannot cover a large enough subset of the true developable shape space, leading to severe locking artifact, see Fig. 4.

Fig. 4. A torsional force is applied to a ribbon with one end fixed. Due to our compact configuration space, bending angles alone cannot cover large enough subset of the developable shape space (top). The naive method of updating crease direction once every 5 frames (middle) still suffer from large deviation from the ground truth (bottom).

We thus need to introduce a resampling operation $\mathbf{R}$ satisfying:

$$\tilde{\mathbf{x}}^{n+1}_j = \mathbf{R}(\tilde{\mathbf{x}}^{n+1}_j, \mathbf{x}^n), \quad \tilde{\mathbf{y}}^{n+1}_j = \mathbf{R}(\tilde{\mathbf{y}}^{n+1}_j, \mathbf{y}^n),$$

where $\mathbf{R}$ is assumed to be piecewise linear in its first parameter. In fact, since all vertices lie on the top or bottom rim of the ribbon, $\mathbf{R}$ is a simple 1D-interpolation. We can then define a valid subtraction: $(\tilde{\mathbf{x}}^{n+1}_j - \mathbf{R}(\tilde{\mathbf{x}}^{n+1}_j, \mathbf{x}^n))/h - \mathbf{R}(\mathbf{x}^{n+1}, \mathbf{y}^n)$, where $\mathbf{R}(\mathbf{X}, \bullet)$ means apply resampling on each $3 \times 1$ block of $\mathbf{X}$. Note that in this way we have essentially encoded remeshing and deformation in a single optimization, and our final form of optimization becomes:

$$\arg\min_{c, \psi, \mathbf{w}, t, \mathbf{x}_r, \mathbf{y}_r} f \quad \text{s.t. constraints 1}$$

$$f = \frac{1}{2} \left\| \frac{\mathbf{x}^{n+1} - \mathbf{R}(\tilde{\mathbf{x}}^{n+1}, \mathbf{x}^n)}{h} - \mathbf{R}(\mathbf{x}^{n+1}, \mathbf{y}^n) \right\|^2 + V.$$  

We want to emphasize that $\mathbb{R}^3$ positions $\mathbf{x}^n$ here is not required to be reconstructed from our generalized coordinates, allowing the ribbon to temporarily deviate from the developable shape space. This flexibility enables conventional collision handlers to be easily integrated into our framework.

Our solver for this nonlinear optimization is detailed in section 5, which heavily relies on an analytical formula for the energy gradient. A naive way of gradient evaluation may follow from Appendix A and chain rule. But we show in Appendix C that it could be greatly accelerated by evaluating in an adjoint mode.

5 The Solver Framework

In this section, we present our full-featured multi-ribbon solver framework, including various user constraints or external forces handling and collision resolution. Algorithm 1 provides an outline of our two-step pipeline. In the first substep, the optimization problem is solved for each ribbon in parallel to predicate a desired new configuration where the $\mathbb{R}^3$ vertex positions are reconstructed. The collision handler then finds a corrected collision free $\mathbb{R}^3$ positions in the second substep, temporarily leaving the developable shape space.

Algorithm 1 one timestep of ribbon solver

Input: a mesh with vertices $< \mathbf{x}^n, \mathbf{y}^n >$

Input: a set of user constraints $C(\mathbf{x}) \geq 0$

1: for each ribbon $r$ do
2: \hspace{0.5cm} $< c, \psi, \mathbf{w}, t, n^r > \rightarrow \text{optimize}(< \mathbf{x}^n_r, \mathbf{y}^n_r >, C_r)$
3: \hspace{0.5cm} $< \mathbf{x}^{n+1}_r, \mathbf{y}^{n+1}_r > \rightarrow \text{reconstruct}(< c, \psi, \mathbf{w}, t, n^r >)$
4: end for
5: $< \mathbf{x}^{n+1}, \mathbf{y}^{n+1} > \rightarrow \text{resolveCollision}(< \mathbf{x}^{n+1}, \mathbf{y}^{n+1} >)$

For the optimization in our first substep, since the analytical Hessian matrix of equation 2 is too costly to evaluate, Newton-type solvers becomes largely unavailable. We thus choose the L-BFGS-B algorithm [26] as our underlying solver, which is wrapped into an Augmented-Lagrangian framework [27] to handle nonlinear constraints.

5.1 Constraints

Now we discuss several types of constraints supported by our framework. The most important is the non-intersecting ruling constraints. Since these constraints are large in number, we transform them into box constraints by a variable substitution as:

$$\Delta c_i = c_i - c_{i-1} \quad 2 \leq i < n - 1,$$

so that they can be handled by the L-BFGS-B algorithm. We are thus left with only two general linear constraints:

$$|c_1 + \sum_{j=2}^{n-1} \Delta c_j| \leq \Delta c_{\text{max}},$$
to be handled by the Augmented-Lagrangian framework.

Another common constraint is the loop constraint that requires the two ends of a ribbon to be connected:

$$x_0 = x_n, \quad y_0 = y_n, \quad n_0 = n_{n-1}$$

for the orientable case or:

$$x_0 = y_n, \quad y_0 = x_n, \quad n_0 = -n_{n-1}$$

for the non-orientable case. These constraints are useful for modelling a ribbon chain or the Mobius band, see Fig. 5.

Finally, a lot of interesting deformations are resulted from torsional forces which is introduced into our framework as additional normal guiding energies: \(V^\text{user} = K/2 ||n - n_0||^2\). Other kinds of common user constraints can be added as in conventional FEM methods. Now we can summarize our optimization substep in algorithm 2.

**Algorithm 2** optimize(\(<X^0, V^n, C>\))

1. \(<c, \psi, w, t >_n= <c, \psi, w, t >_n, \lambda = 0, \mu = 1e^3\)
2. for \(k = 1, 2, 3, \cdots \)
3. \(g = \frac{d}{dt}\min(C(X) + \lambda/\mu, 0)^2\)
4. initial guess \(<c_1, \Delta c, \psi, w, t >_0 = \frac{P}{c_1, \Delta c, \psi, w, t >_k - 1}\)
5. \(<c_1, \Delta c, \psi, w, t >_k = L_{\text{BFGS}}(f + g_1, |\Delta c| \leq \text{max})\)
6. \(<c, \psi, w, t >_k = P^{-1} c_1, \Delta c, \psi, w, t >_k\)
7. if \(|| c, \psi, w, t >_k - c, \psi, w, t >_k - 1 < 1e^{-4}\) then
8. return \(<c, \psi, w, t >_k\)
9. else
10. \(\lambda = \min(\lambda + \mu C(X, 0))\)
11. end if
12. end for

5.2 Collision Resolution

One advantage of our formulation is that existing collision detection and resolution methods can be trivially plugged into our framework as a post processor. After a new configuration \(<c, \psi, w, t >_n+1\) is returned by our optimizer, a triangle mesh with vertices \(X_{n+1}\) is reconstructed and passed to the collision handler, which in turn finds a closest collision free mesh with vertices: \(X^n\). Although this mesh may not lie in the developable shape space, its distance to our configuration space is very close in our experiments. Moreover, this developability error will not accumulate because a valid configuration is always recovered by our first substep at next frame.

To work the best with our method, one need to be careful in choosing of underlying continuous collision handler. There are generally two methods for resolving a large amount of continuous collisions: local methods based on randomized impulses [28] and globally coupled methods such as non-rigid impact zone [29]. Since the closeness to a valid configuration is important in our case, we choose to use globally coupled handlers. Specifically, we solve a quadratic programming of the following form to resolve a set of collision constraints \(C_{col}\) are returned by the detector:

\[
\begin{align*}
\operatorname{argmin} & \quad E(X) \\
\text{s.t.} & \quad C_{col}(X) \geq 0,
\end{align*}
\]

where \(E\) is some closeness measure. In the original work [29], \(E\) is simply \(E = \frac{1}{2}||X_{n+1} - X^n||^2\). This measure leads to diagonal Hessian so that the QP problem can be solved efficiently in its dual form for a very large mesh, especially when the number of constraints are much smaller than number of vertices. Although this measure totally ignores the stiffness between vertices, this choice is appropriate for most cloth animation setups where a small timestep size is used.

But in our case, the situation is reversed. Due to our compact representation, the reconstructed mesh is orders of magnitude smaller for comparable results than FEM method. As a result, the number of potential constraints are usually comparable to the number of vertices. On the other hand, since we used fully implicit method, our solver is stable under large timestep. Unfortunately, such large timestep size also makes the collision force stiff. As a result, ignoring internal stiffness in \(E(X)\) would lead to large discrepancy between \(X_{n+1}\) and \(X^n\). This may introduce large developability error and finally cause collision failure due to degenerated or flipped triangles which is illustrated in Fig. 6.

Fig. 6. A frame from the same animation as bottom Fig. 12. Compared with conventional method (left), the collision handler can be greatly stabilized with additional \(E_{\text{Stiff}}\) terms (right) under large timestep size.

Out of these considerations, we add an quadratic artificial stiffness term to \(E(X)\). Specifically, for each edge \(e\) of the reconstructed triangle mesh with vertices \(x_1, x_2\), we introduce additional energy terms: \(E_{\text{stiff}}(x_1, x_2) = \frac{K}{2}||x_1 - x_2||^2\), where \(K\) is an artificial stiffness coefficient set to \(1e^2\) in all our examples. The new QP problem:

\[
\begin{align*}
\operatorname{argmin} & \quad E(X) + \sum_e E_{\text{stiff}}(X) \\
\text{s.t.} & \quad C_{col}(X) \geq 0,
\end{align*}
\]

is again solved in its dual form using the active set method, where the dual Hessian \(CH^{-1}C^T\) is calculated by factor-factoring the sparse Hessian \(H\) and solve the sparse right hand side \(C^T\). Since our mesh is rather small, the overhead of this solve is negligible.

6 Results and Validations

The stability, accuracy and efficiency of our method is evaluated using several benchmark tests, the performance of our
Fig. 5. Our algorithm can robustly handle nonlinear loop constraints. In this example, an non-orientable loop constraint is assigned to each of the 9 ribbons, forming a Mobius chain.

solver on all examples is summarized in table 1. In a first set of tests illustrated in Fig. 7, we apply large torsional or dragging forces on a ribbon with or without loop constraint. Fortunately, our optimizer presents no performance degradation in both cases. The stability is also validated in temporal domain.

Fig. 7. Stability test on a ribbon with length=1m and width=0.05m segmented into 50 elements. Top row: twisting the spherical ribbon by $4 \pi$ with strong normal constraints. Bottom row: a looped ribbon under large dragging force.

In this subsequent test, we compared two simulated helix unrolling sequences under different timestep size. Our solver finds a reasonably consistent result with $h$ up to 0.05s, see Fig. 8.

To demonstrate our advantage over previous methods, we compared our solver with conventional FEM methods using (non-)conforming mesh. For the conforming solver, we have to introduce an artificial stiffness term which is set to $1e8$. According to Fig. 9 and the accompanying video, the conforming solver would suffer from noisy spatial error due to locking phenomena and the non-conforming solver instead suffers from noisy temporal error due to the conforming reconstruction procedure and additional boundary constraints. By contrast, our method on a much smaller mesh faithfully regenerates the zigzag ruling pattern without any instability in both spatial and temporal domain.

In addition, we investigated the accuracy of our kinetic energy formulation by showing the consistency with conventional FEM methods again. However, as is shown in Fig. 9, FEM solver cannot serve as a valid groundtruth. To resolve this contradiction, we observe that our method is equivalent to an conventional FEM solver with dynamic remeshing respecting the ruling direction. In view of this, we work on an animation sequence generated using our method, and for any two consecutive frames $i$ and $i+1$, we insert all vertices to get a combined mesh with material space vertex positions $\bar{Z}' = (\bar{X}_{i}, \bar{X}_{i+1})$. From this mesh with world space vertex positions $Z' = (X_{i}, R(\bar{X}_{i+1}, X'))$, we perform time integration.

| Scene                   | Ribbon | Opt./[Coll.](sec) | Inner/Outer |
|-------------------------|--------|-------------------|-------------|
| Extreme Torsion         | 50 x 1 | 0.015             | 270/1       |
| Extreme Dragging        | 50 x 1 | 0.03              | 432/4       |
| Torsion (ours)          | 50 x 1 | 0.06              | 110/1       |
| Torsion (C-FEM)         | 50 x 10| 0.36              | N/A         |
| Torsion (NC-FEM)        | 50 x 10| 3.0               | N/A         |
| Helix Unrolling (h=0.01)| 50 x 1 | 0.045             | 414/1       |
| Helix Unrolling (h=0.05)| 50 x 1 | 0.12              | 439/1       |
| Ribbon Falling          | 150 x 1| 2.12/1.2          | 981/1       |
| Double Chain            | (50 x 1) x 18 | 2.18/2.5       | 321/5       |
| Mobius Chain            | (50 x 1) x 9  | 2.24/2.1         | 3417/5      |

TABLE 1

Time cost for one step of our solver. From left to right: number of ribbon segments in longitude/latitude dimension, average time cost for optimization/collision detection if applicable and number of inner/outer iterations taken by the optimizer. All tests are done on a single desktop computer with dual Intel Xeon E5-2630 CPU and 128Gb memory.
Fig. 10. The animation of free helix unrolling with $h = 0.01s$. Visualization of discrepancy between $Z_{FEM}^{i+1}$ (red) and $Z_i^{i+1}$ (brown) at $t = 0s, 0.25s, 0.5s, 0.75s, 1s$, from left to right. For reference, $Z_i$ is also shown in green.

Fig. 9. Twisting a ribbon with length=$1m$ and width=$0.1m$ segmented into 50 longitude elements and 10 latitude elements. From top to bottom: our method, FEM solver on a conforming mesh and FEM solver on a non-conforming mesh.

using FEM method on conforming mesh to predicate a new combined mesh with world space vertex positions $Z_{FEM}^{i+1}$, which is then compared with $Z_i^{i+1} \triangleq (R(\bar{X}_i, X_{i+1}), X_{i+1})$. This essentially compares the per-frame discrepancy between our method and FEM method with remeshing factored out. The result is visualized in Fig. 10, which validates the accuracy of our formulation. Unfortunately, such accuracy is achieved at the cost of an additional resampling operator $R$. There is however a naive simplification to our kinetic term by lumping all the mass to the centerline, giving a simplified objective energy:

$$f \triangleq \sum_{i=0}^{n} \frac{M_i}{2} \left\| (\bar{x}^{i+1}_i + y^{i+1}_i - x^{i}_i - y^{i}_i) / 2h - (\bar{x}^{n}_i + y^{n}_i) / 2 \right\|^2 + V,$$

which just takes the average of velocities and positions along the ruling direction for the centerline. Here $M_i$ is the constant lumped mass for the $i$th centerline vertex. Such approximation works well in some cases such as Fig. 10 but may lead to severe artifact elsewhere. An extreme example is shown in Fig. 11.

One major disadvantage of our algorithm is that the performance degenerates with the number of segments according to Fig. 13. The bottleneck of our method is the evaluation of $\nabla x_j$ and $\nabla y_j$, which is quadratic in $n$, which can be largely removed using the adjoint method. But longer ribbon won’t affect the stability our method. In the falling ribbon example of Fig. 12, we used a long ribbon with 150 segments. In this case the overhead of optimization dominates our solver pipeline.

The stability of collision and contact handling is illustrated in Fig. 12 as well. For both of these examples we used $h = 0.01$. Under such large timestep size, our new closeness metric $E$ is an indispensable component especially for the double ribbon chain example. Due to the stiffness of collision forces, conventional non-rigid impact zone solver generates extremely distorted triangles and quickly fails after the first few segments of the chain get in contact.

7 Conclusion and Discussion

In conclusion, the paper presents an configuration space that covers a large subset of the shape space of developable ribbon. Based on this configuration space, we develop an optimization based ribbon simulator with flexible user controllability and robust collision handling. Thanks to the compactness of the configuration space, the solver is locking free compared with previous FEM based methods. This enables the solver high fidelity on a very small mesh. Moreover, it presents better stability in both temporal and spatial domain over conventional methods [17], [6].
We also noticed several drawbacks of the method. A clear problem is that our configuration parameters are densely related to vertex positions, which limits the scalability in terms of both time and memory for extremely long chain. The optimizer would also require more gradient evaluations for longer ribbon. To alleviate this problem, it is worth exploring acceleration techniques such as Newton-type optimizer or exploring acceleration techniques such as adjoint method (blue) energy gradient evaluation.

**Fig. 12.** Our method handles collision and contact robustly. Top: A long ribbon falling on the ground. Bottom: impact of two ribbon chains with 9 ribbons each.

**Fig. 13.** Time cost of finding the rest shape of a mobius band for ribbon of different length, using simple chain rule (red) or the adjoint method (blue) energy gradient evaluation.

**REFERENCES**

[1] M. P. Do Carmo and M. P. Do Carmo, *Differential geometry of curves and surfaces*. Prentice-hall Englewood Cliffs, 1976, vol. 2.

[2] B. Bo and W. Wang, “Geodesic-controlled developable surfaces for modeling paper bending,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 365–374, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2007.01059.x

[3] J. Solomon, E. Vouga, M. Wardetzky, and E. Grinspun, “Flexible developable surfaces,” *Computer Graphics Forum*, vol. 31, no. 5, pp. 1567–1576, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2012.03162.x

[4] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, “Discrete shells,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 62–67.

[5] E. English and R. Bridson, “Animating developable surfaces using nonconforming elements,” *ACM Transactions on Graphics*, 2008.

[6] ——, “Animating developable surfaces using nonconforming elements,” in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 66.

[7] J. Spillmann and M. Teschner, “Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects,” in *In Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007, pp. 63–72.

[8] F. Bertails, B. Audoly, M.-P. Cani, B. Querleux, F. Leroy, and J.-L. Lévéque, “Super-helices for predicting the dynamics of natural hair,” in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1180–1187.

[9] R. Casati and F. Bertails-Descoubes, “Super space clothoids,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 48, 2013.

[10] E. Cosserat and F. Cosserat, “Théorie des corps déformables,” Paris, 1909.

[11] S. Javdani, S. Tandon, J. Tang, J. F. O’Brien, and P. Abbeel, “Modeling and perception of deformable one-dimensional objects,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1607–1614.

[12] D. K. Pai, “Strands: Interactive simulation of thin solids using cosserat models,” *Computer Graphics Forum*, vol. 21, no. 3, pp. 347–352, 2002. [Online]. Available: http://dx.doi.org/10.1111/1467-8659.00594

[13] J. Spillmann and M. Teschner, “C o r d e: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007, pp. 62–67.

[14] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, “Discrete elastic rods,” in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 63.

[15] F. Bertails, “Linear time super-helices,” in *Computer graphics forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 417–426.

[16] P. G. Ciarlet, *Theory of shells*. Elsevier, 2000.

[17] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, “Discrete shells,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 62–67.

[18] R. Bridson, S. Marino, and R. Fedkiw, “Simulation of clothing with folds and wrinkles,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 28–36.

[19] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun, “Efficient simulation of inextensible cloth,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 49, 2007.

[20] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, “Discrete quadratic curvature energies,” *Computer Aided Geometric Design*, vol. 24, no. 8, pp. 499–518, 2007.

[21] B. Bo and W. Wang, “Geodesic-controlled developable surfaces for modeling paper bending,” in *Computer Graphics Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 365–374.

[22] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang, “Geometric modeling with conical meshes and developable surfaces,” in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 681–689.

[23] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross, “Example-based elastic materials,” in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 72.

[24] D. Cohen-Steiner and J.-M. Morvan, “Restricted delaunay triangulations and normal cycle,” in *Proceedings of the nineteenth annual symposium on Computational geometry*. ACM, 2003, pp. 312–321.

[25] R. Nairn, A. Samii, and J. F. O’Brien, “Adaptive anisotropic remeshing for cloth simulation,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 152, 2012.
APPENDIX A

EXPLICIT FORMULA FOR $x_j$ AND $\nabla x_j$

Since $T_k$ for $1 \leq k < n$ is just rigid rotation, they have simple analytical form:

$$T_k = \begin{pmatrix} \cos(y_k) + c_k & -\sin(y_k) c_k - c_k \cos(y_k) & \cdots & -\sin(y_k) c_k + \cdots & \sqrt{c_k^2 + 1} \\ -\sin(y_k) c_k - c_k \cos(y_k) & \cos(y_k) & \cdots & \sqrt{c_k^2 + 1} \\ \sqrt{c_k^2 + 1} & \sqrt{c_k^2 + 1} & \cdots & \sqrt{c_k^2 + 1} \\ 0 & 0 & \cdots & n(c_k^2 + 1) \end{pmatrix},$$

whose partial derivatives $\partial T_k / \partial c_k$ and $\partial T_k / \partial w_k$ can be found using a symbolic software. We omit these here to save space. And for the global rigid transformation, we have:

$$\frac{\partial T_0}{\partial w_k} = \begin{pmatrix} \frac{\partial \text{exp}^w}{\partial w_k} & 0 \\ 0 & 1 \end{pmatrix}, \quad \frac{\partial T_0}{\partial T_k} = \begin{pmatrix} 0 & e_k \\ 0 & 1 \end{pmatrix},$$

where $\frac{\partial \text{exp}^w}{\partial w_k}$ can be found using the Rodriguez’s formula. Now $\nabla x_j$ can be found using simply chain rule:

$$\frac{\partial x_j}{\partial c_k} = \left[ \Pi_{i=0}^{k-1} T_i \right] \frac{\partial T_k}{\partial c_k} \left[ \Pi_{i=k+1}^n T_i \right] \tilde{x}_j + \left[ \Pi_{i=0}^k T_i \right] \frac{\partial \tilde{x}_j}{\partial c_k},$$

$$\frac{\partial x_j}{\partial \psi_k} = \left[ \Pi_{i=0}^{k-1} T_i \right] \frac{\partial T_k}{\partial \psi_k} \left[ \Pi_{i=k+1}^n T_i \right] \tilde{x}_j,$$

$$\frac{\partial x_j}{\partial w_k} = \frac{\partial T_0}{\partial w_k} \left[ \Pi_{i=1}^k T_i \right] \tilde{x}_j,$$

$$\frac{\partial x_j}{\partial e_k} = \frac{\partial x_j}{\partial T_k} \left[ \Pi_{i=1}^k T_i \right] \tilde{x}_j.$$

APPENDIX B

THE MASS MATRIX

Without loss of generality, we use linear shape function for each quad element. In this case, element $E_i$ would contribute a $12 \times 12$ mass block of the following form:

$$M_i = \begin{pmatrix} \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \\ \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \\ \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \\ \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \\ \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \\ \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c \\ \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 & \Delta c & \Delta c^2 \end{pmatrix} \otimes \text{Id}^{3 \times 3},$$

where $\Delta c = c_{i+1} - c_i.$