HAL
open science

Super Space Clothoids

Romain Casati, Florence Bertails-Descoubes

To cite this version:

Romain Casati, Florence Bertails-Descoubes. Super Space Clothoids. ACM Transactions on Graphics, 2013, Proceedings of SIGGRAPH 2013, 32 (4), pp.Article No. 48. 10.1145/2461912.2461962 . hal-00840335v2

HAL Id: hal-00840335

https://inria.hal.science/hal-00840335v2

Submitted on 3 Jul 2013 (v2), last revised 16 Jan 2014 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Super Space Clothoids

Romain Casati Florence Bertails-Descoubes

INRIA and Laboratoire Jean Kuntzmann (Grenoble University, CNRS), France*

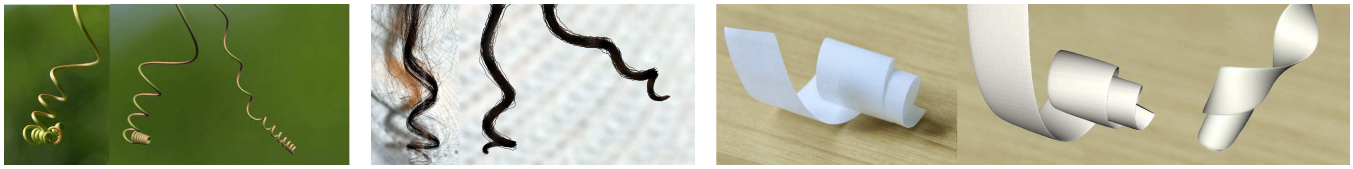


Figure 1: *Many physical strands exhibit a smooth curled geometry with affine-like curvature profile, which is captured and deformed accurately thanks to our new 3D dynamic primitive. From left to right, three examples of real strands whose shapes are synthesized and virtually deformed in real-time using a very low number of 3D clothoidal elements: a vine tendril (4 elements), a hair ringlet (2 elements), and a curled paper ribbon (1 single element). Left photograph courtesy of Jon Sullivan, pdphoto.org.*

Abstract

Thin elastic filaments in real world such as vine tendrils, hair ringlets or curled ribbons often depict a very smooth, curved shape that low-order rod models — e.g., segment-based rods — fail to reproduce accurately and compactly. In this paper, we push forward the investigation of high-order models for thin, inextensible elastic rods by building the dynamics of a G^2-continuous piecewise 3D clothoid: a smooth space curve with piecewise affine curvature. With the aim of precisely integrating the rod kinematic problem, for which no closed-form solution exists, we introduce a dedicated integration scheme based on power series expansions. It turns out that our algorithm reaches machine precision orders of magnitude faster compared to classical numerical integrators. This property, nicely preserved under simple algebraic and differential operations, allows us to compute all spatial terms of the rod kinematics and dynamics in both an efficient and accurate way. Combined with a semi-implicit time-stepping scheme, our method leads to the efficient and robust simulation of arbitrary curly filaments that exhibit rich, visually pleasing configurations and motion. Our approach was successfully applied to generate various scenarios such as the unwinding of a curled ribbon as well as the aesthetic animation of spiral-like hair or the fascinating growth of twining plants.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: 3D Euler spiral, thin elastic rod, power series

Links: DL PDF WEB VIDEO CODE

*e-mail:{romain.casati,florence.descoubes}@inria.fr

1 Introduction

A key motivation in Computer Graphics is the creation of digital shapes and motions which capture or even enhance the visual complexity and beauty of nature. Long and thin flexible structures, often called *strands* [Pai 2002], are well-spread in plants (foliage, stems), animals (hair, coral) and human-made objects (ropes, ribbons). Due to their smooth curved shape and complex way of deforming, characterized by many instabilities, strands largely participate to the world's visual richness and aesthetics. In this paper we aim at deriving an accurate, efficient and robust computational model to simulate the mechanics of strand-like structures, with a particular interest for *curled geometries*.

The nonlinear mechanical behavior of inextensible and unshearable strands is well-described by the Kirchhoff theory of thin elastic rods, set up more than a century ago [Dill 1992]. However, the governing equations of motion, consisting of stiff partial differential equations of fourth order in space, are known to be difficult to discretize and thus delicate to simulate in both a faithful and stable way. In particular, inextensibility and bending forces, which are the main sources of numerical stiffness, need to be treated carefully.

Most previous methods, relying on a nodal displacement formulation of strands, lead to sparse equations but require considerable refinement to account for *curved* geometries. Furthermore, handling the inextensibility constraint and discretizing the nonlinear bending forces in a *stable* way is challenging. In contrast, here we seek for *high-order* rod elements whose shape compactly and faithfully approximates large portions of real, arbitrarily bendy strands, with a reduced parametrization adapted to the kinematics of the rod. In that vein, the super-helix model, relying on deformable, perfectly inextensible helical elements, and yielding linear bending forces, was a first approach towards this goal [Bertails et al. 2006]. However, this model still lacks one order of continuity (only G^1-continuous junctions) for capturing visually pleasing smoothness properties (at least G^2 continuity). More generally, it turns out that in the real world, most strands exhibit a continuous curvature profile (see Figure 1), much closer to a piecewise *affine* profile rather than a piecewise constant one. Reinforced by this observation, we design a new rod element whose centerline takes the form of a 3D clothoid or 3D Euler spiral — a space curve characterized by linearly varying curvature and torsion (see, e.g., [Harary and Tal 2012])[1]. Our new *super space clothoid* rod model, stable and per-

[1]The centerline of our rod element is actually more general as it corresponds to linear *material* curvatures and twist — the entire class of so-called 3D Euler spirals being obtained by cancelling the first material curvature.

fectly inextensible, results from the G^2-continuous assemblage of such elements.

One major difficulty with a model based on linearly varying curvature and torsion, compared to lower-order geometries, lies in the numerical evaluation of the centerline (and as a consequence, of all kinematic terms), which does not have a closed-form anymore, but still needs to be precisely evaluated. In the 2D case, Bertails-Descoubes [2012] addressed this issue with the help of Romberg's quadrature rule to evaluate the kinematics of a 2D rod element (a clothoid). Although this approach remains practical and fast enough for real-time in 2D, due to simple relationships between curvature (one single variable), orientation (one angle), and position, it becomes totally unsuitable in the 3D case where the kinematics is governed by a linear differential equation operating on a nonlinear manifold of dimension 6. As for traditional numerical schemes, they turn out to be too prohibitive to discretize both kinematic and dynamic spatial terms at the requested precision, in a reasonable amount of time.

Our work is inspired by ideas from the symbolic computation community whose primary focus are *extremely accurate* computational methods with bounding guarantees, relying on advanced algebraic considerations combined with multi-precision evaluation algorithms (see, e.g., [Mezzarobba 2010]). However, if high computational accuracy is clearly part of our requirements, *efficiency* is also crucial to us, in order to achieve on a standard machine the millions of evaluations expected at each time step, in a reasonable amount of time. Our work is thus aimed at designing a method suitable for standard, *floating-point* arithmetic.

The chore of our approach is a new dedicated integration scheme based on *power series expansions*, which fully leverages the structure of the rod kinematic problem while carefully avoiding numerical issues due to floating-point arithmetic. Our method reaches machine precision *orders of magnitude faster* compared to classical numerical integrators. Furthermore, our algorithm naturally extends to the computation of sums, products and integrals, allowing us to evaluate all spatial terms of the rod dynamics in an efficient and accurate way. As a result, we are able to simulate a full dynamic rod made of 6 clothoidal elements in real-time. Compared to previous rod models, our approach provides a better order of spatial convergence and generates richer motion at a competitive computational cost.

2 Related Work

The scientific study of strands has a long history in various fields, tracing back to the first continuous mechanics theories a few centuries ago to their further analysis in physics and mathematics, and their recent numerical treatment in Mechanical Engineering and Computer Graphics. Motivation originates from a number of applications ranging from the understanding of DNA supercoiling [Benham and Mielke 2005] and climbing plants [Goriely and Neukirch 2006] to the simulation of submarine cables [Goyal et al. 2008], surgery threads and needles [Pai 2002; Chentanez et al. 2009], or hair [Ward et al. 2007].

Theories for thin elastic rods Various theories were proposed in mechanics to model the equilibria and the dynamics of strands, depending on the type of deformation considered. In this paper, our goal is to capture the geometric richness of typical strands deformations such as waving hair, coiling cables, curled ribbons or twining plants. These phenomena are largely *nonlinear*, dominated by *bending* and *twisting* elastic deformations, while stretching and shearing can be neglected. To properly account for this regime, we consider *inextensible* strands with a vanishing cross-section inertia, neglect shearing, and assume moment strains to remain small — making use of an *elastic* constitutive model — while large displacements, at the origin of the desired geometric nonlinearities, are allowed. The model is thus strictly subject to finite[2] rotations around the cross-section axes (bending) and around the tangent of the centerline (twisting). The corresponding governing equations — a set of partial differential equations together with boundary conditions — were first developed by Kirchhoff and Clebsch in their theory of thin elastic rods under finite displacements [Dill 1992]. Within a more general framework on shells, rods and points, the Cosserat brothers [1909] later on proposed a clever mathematical representation of the rod geometry, relying on a space curve (the *centerline*) together with a *material* frame attached to the rod cross-section and continuously rotating along the centerline around the so-called *Darboux* rotation vector. A modern description of these theories can be found in [Antman 1995; Audoly and Pomeau 2010]. Pai [2002] was the first to introduce them to the Computer Graphics community.

Discretizing material rods In Mechanical Engineering, both *finite differences* and *finite elements* approaches were developed to discretize material rods in space and time. Though finite differences schemes have in principle the advantage of being easy to set up, properly accounting for the rod boundary conditions (typically, a clamped rod with the other end free) generally requires the use of a shooting strategy, which implies the solving of multiple nonlinear problems. Moreover, the stiff nature of the Kirchhoff equations, stemming from the presence of fourth-order spatial derivatives, imposes the use of overly small steps in time and space, or sophisticated implicit integrators [Goyal et al. 2008]. In contrast, a finite elements strategy allows one to single out spatial terms from time-evolving quantities, and provides a vast choice of elements to approximate them together with the boundary conditions. A popular method is the so-called geometrically exact beam approach [Reissner 1973; Simo and Vu-Quoc 1986], which derives an exact weak formulation for a generalized Kirchhoff rod with stretching and shearing, and finally discretizes the displacement and rotation fields with interpolating shape functions. One important issue of this approach, which spurred many subsequent works in the finite elements community, deals with the proper interpolation of rotations for preserving *objectivity*, i.e., invariance of the strain measures under rigid motion [Crisfield and Jelenić 1998]. Moreover, regarding our specific needs here, this method is not directly applicable to the handling of *inextensible* and *unshearable* rods.

In Computer Graphics, finite differences schemes initially proposed by Pai [2002] to solve the statics of Kirchhoff rods were subsequently superseded with more robust schemes so as to deal with the full dynamic case, relying on variational formulations or discrete differential geometry. Two purely *reduced*-coordinates models, based on a minimal parametrization of the system, were proposed to account for the exact kinematics of the rod, and especially to preserve inextensibility: the articulated rigid body approach [Hadap and Magnenat-Thalmann 2001; Hadap 2006], parameterized by angular joints, and the super-helix model [Bertails et al. 2006], parameterized by curvatures and twist. In contrast, further work focused on *nodal* models in order to get an explicit, point based representation of the centerline leading to a sparse mass matrix, at the price of adding external constraints to preserve the true kinematics. In the CoRde model [Spillmann and Teschner 2007], both positions and orientations are considered as degrees of freedom. The Lagrange equations of motion are written for discrete approximations of kinetic and potential energies — including a stretch term — and orientations are coupled back to the centerline through soft constraints. Relying on the Bishop frame, Bergou et al. [2008] use a curve angle parameterization to reduce the number of redundant parameters and guarantee that the orientation frame naturally remains

[2] As opposed to infinitesimal.

adapted to the centerline. Discrete equations of motion are then established by leveraging principles from discrete differential geometry. Due to the choice of a nodal parameterization, inextensibility however needs to be explicitly enforced, e.g., through a fast projection scheme [Bergou et al. 2008] or a stiff stretch term [Bergou et al. 2010]. Finally, to ensure proper stability at an acceptable computational cost, a fully implicit scheme based on Newton's method is advocated to discretize the nonlinear stiff bending and stretching forces [Bergou et al. 2010].

Reduced Lagrangian dynamics One advantage of reduced dynamics is that, by directly considering moment strains as actual degrees of freedom (i.e., curvatures and twist instead of positions), the rod kinematics is exactly preserved, without redundancy and without adding any further constraint. Models parameterized by curvatures also benefit from an inexpensive implicit handling of bending forces, as those forces are linear in curvature. Finally, while multiple collision tricks — such as the position alteration technique [Baraff and Witkin 1998] — were specifically developed for nodal models in Computer Graphics, more sophisticated contact solvers including Coulomb friction naturally cope with reduced Lagrangian models [Daviet et al. 2011], without having to worry about getting intermingled with external kinematic constraints.

Super-helices Similarly to finite elements, the super-helix model relies upon a weak formulation of the dynamics of a Kirchhoff rod. The main difference is that the sparse discretization does not operate onto displacement and rotation fields, but at the curvature level. Curvatures and twist are approximated with piecewise constant functions, from which the geometry of the rod — which takes the simple form of a piecewise circular helix — is recursively derived. An important advantage of such a formulation is that the true kinematics of the rod, including perfect inextensibility, is intrinsically captured. The non-objectivity issue raised by finite elements methods that linearly interpolate the rotation field is also naturally circumvented. The price to pay is the loss of sparsity in the mass matrix. However, in practice a small number of elements — five to ten — is generally sufficient to capture rich shapes and deformations, while keeping computations reasonable.

Yet, using a low resolution may yield a lack in *fairness*, as illustrated in Figure 2. Like in the 2D case [Bertails-Descoubes 2012], G^1-smooth junctions between elements (continuity of tangents only, not of curvatures) are particularly visible in 3D and aesthetically disturbing. Moreover, a long helical element located at the clamped end of a strand may not possess enough degrees of freedom to correctly unwind when pulled downwards, and may remain "locked". Such issues are naturally alleviated with our new super space clothoid model.

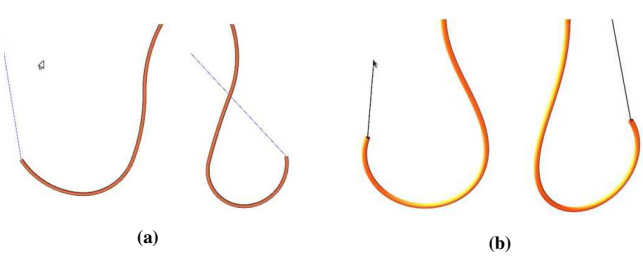


Figure 2: *Comparison of fairness between (a) the super-helix model and (b) our space clothoid model. Whereas junctions between elements are particularly visible for the super-helix (5 elements here), our model generates a very smooth, visually pleasing shape even at a very coarse resolution (2 elements here).*

3 Contributions and Overview

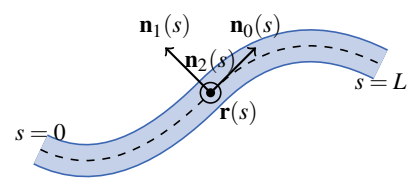
Our idea is to push forward the investigation of high-order rod models by considering elements whose material curvatures and twist vary *linearly* with arc length (Section 4). Compared to previous rod models, one important challenge when increasing the order of elements is the loss of a closed-form solution for the kinematics. While traditional integration schemes become excessively prohibitive at the requested precision in the 3D case, we discovered that the kinematic problem still possesses a lot of structure (Section 5), which can be leveraged so as to design a fast and highly accurate integration scheme (Section 6). Importantly, our algorithm naturally extends to the computation of all spatial terms of the dynamics (Section 7), allowing us to simulate rods with rich shape and motion efficiently. We carefully validate our new rod model against the most relevant models of the literature and demonstrate the effectiveness of our approach, especially for winding rods, through various examples ranging from the growth of twining plants to the animation of curly hair (Section 8).

4 Discrete Kirchhoff Rods

Notation In what follows, s denotes the space variable and t the time variable. Space derivatives are represented by the prime symbol, so that $\mathbf{a}'(s,t) = \frac{\partial \mathbf{a}}{\partial s}$ and time derivatives by the dot symbol, so that $\dot{\mathbf{a}}(s,t) = \frac{\partial \mathbf{a}}{\partial t}$. For the sake of clarity, we may omit the time variable when describing the geometry of the rod. The special orthogonal group of dimension 3, denoted $SO(3)$, collects finite rotations of $\mathbb{R}^3$ (represented as direct orthogonal matrices) and is a *non commutative* Lie group.

4.1 General Case

Let us consider an inextensible and unshearable material rod of length L, represented by a centerline $\mathbf{r}(s)$ together with a material frame $\mathscr{R}(s)$, both parameterized by arc length $s \in [0,L]$. At location s, the vector $\mathbf{r}(s) \in \mathbb{R}^3$ gives the 3D position of the centerline and the rotation $\mathscr{R}(s) \in SO(3)$ encodes the tangent vector $\mathbf{n}_0(s) = \mathbf{r}'(s)$ as well as the two normal vectors $\mathbf{n}_1(s)$ and $\mathbf{n}_2(s)$ attached to the cross section of the rod.



For simplicity, we assume the rod is clamped at $s = 0$ and its clamped position $\mathbf{r}(0) = \mathbf{r}_{\text{cl}}$ and orientation $\mathscr{R}(0) = \mathscr{R}_{\text{cl}}$ are given. Note that this assumption holds in most real strands we wish to model, e.g., plants and hair. Otherwise, it could easily be dropped out by releasing $\mathbf{r}_{\text{cl}}$ and $\mathscr{R}_{\text{cl}}$ as degrees of freedom.

Kinematics From $s = 0$ to $s = L$, the material frame $\mathscr{R}(s)$ continuously evolves along the centerline $\mathbf{r}(s)$ through infinitesimal rotations around the so-called *Darboux vector* $\Omega(s)$ which represents the instantaneous space rotation vector of the rod. This space evolution mathematically writes

$$\mathscr{R}'(s) = [\Omega(s)]_\times \mathscr{R}(s), \tag{1}$$

where $[\mathbf{u}]_\times$ denotes the skew symmetric matrix corresponding to the vector cross product operator, i.e., $[\mathbf{u}]_\times \mathbf{v} = \mathbf{u} \times \mathbf{v}$. It is noteworthy that the local coordinates of Ω in the material frame represent the *material twist* κ_0 and *curvatures* κ_1 and κ_2 of the rod, i.e., $\Omega(s) = \mathscr{R}(s)\,\kappa(s)$, where $\kappa(s) = [\kappa_0(s), \kappa_1(s), \kappa_2(s)]$ is called the

curvature vector in the remainder of the paper. By further using properties of rotation matrices, one can reformulate Equation (1) as

$$\mathscr{R}'(s) = \mathscr{R}(s)\,[\boldsymbol{\kappa}(s)]_\times\,. \tag{2}$$

Finally, by compacting the centerline and the material frame into one single variable $\mathscr{F}(s) = \{\mathbf{r}(s);\mathscr{R}(s)\}$ and assuming $\boldsymbol{\kappa}(s)$ is fixed, the full kinematics of the rod can be formulated as an explicit[3] linear first-order Cauchy-Lipschitz problem, referred to as the *Darboux problem* (see, e.g., [Ivanova 2000]),

$$\left\{\begin{array}{rcl} \mathscr{F}'(s) & = & \{\mathbf{n}_0(s)\,;\,\mathscr{R}(s)\,[\boldsymbol{\kappa}(s)]_\times\} \\ \text{with}\quad \mathscr{F}(0) & = & \{\mathbf{r}_{\text{cl}}\,;\,\mathscr{R}_{\text{cl}}\}\ \text{as initial conditions,} \end{array}\right. \tag{3}$$

which admits a unique solution. Note that the ambient space is not a vector space but rather a *nonlinear* differentiable manifold, since the kinematic relationship for the material frame operates onto the non commutative Lie group $SO(3)$. Due to non commutativity, the solution has no formal expression in the general case.

Dynamics Let ρ be the volumetric mass of the rod and S the surface area of its cross section. We assume the rod is subject to external forces such as gravity or contact forces. Expressing the balance of linear and angular momentums on an infinitesimal portion of the rod and neglecting inertial momentum due to the vanishing cross-section lead to the following dynamic equations for a Kirchhoff rod,

$$\left\{\begin{array}{rcl} \rho S\ddot{\mathbf{r}}(s) & = & \mathbf{T}'(s) + \mathbf{p}(s) \\ \mathbf{M}'(s) + \mathbf{n}_0(s)\times\mathbf{T}(s) & = & 0 \end{array}\right. \tag{4}$$

where $\mathbf{p}$ is the linear density of external forces and $\mathbf{T}(s)$ (resp. $\mathbf{M}(s)$) is the internal force (resp. internal moment) transmitted from the free part of the rod through its cross section at s. The free end condition at $s = L$ implies that $\mathbf{T}(L) = \mathbf{M}(L) = 0$.

Finally, dynamic equations are completed with a constitutive law that express the ability of the rod to elastically bend and twist,

$$\mathbf{M}(s) = \mathbb{K}_3\left(\boldsymbol{\kappa}(s) - \boldsymbol{\kappa}^0(s)\right) \quad \text{in the local basis } \mathscr{R}(s), \tag{5}$$

where $\mathbb{K}_3$ is a diagonal 3×3 matrix collecting the twisting and bending stiffness, and $\boldsymbol{\kappa}^0(s) \in \mathbb{R}^3$ collects the intrinsic curvatures and twist of the rod, used to model spontaneous curliness.

Numerical model Equations (3-5) together with the boundary conditions at $s = 0$ and $s = L$ form a nonlinear and stiff boundary value problem, which has no formal solution and is known to be difficult to solve numerically.

Realizing that *curvature* plays a key role in both the kinematics and the dynamics of the rod, an interesting idea consists in approximating the curvature vector with a simple, *polynomial* expression that is function of s. The coefficients of the polynomial are then taken as primary variables of the discrete model. One immediate consequence is that bending forces, which are linear in curvature, become linear in the discrete variables. Being stiff in nature, those forces can thus be treated implicitly in a straightforward manner, without having to solve a nonlinear problem. Furthermore, the kinematic (Darboux) problem becomes numerically *tractable*. In the simplest case when the curvature is assumed to be constant, the solution to Equation (3) is exactly a circular helix. Getting such a closed-form kinematics was the main strength of the super-helix model. However, as noted previously, a discrete rod with piecewise constant curvature may still represent a rather rough approximation of the continuous case, with an improper degree of continuity at the joints. Instead of using an excessively refined primitive, one may think it would be worth designing a richer, higher-order element with *affine* curvature, that would better stick to the actual curvature profile of real strands and guarantee visually pleasing smoothness of the centerline *at any resolution*. One becomes unfortunately faced with the loss of a formal expression for the kinematics. Yet, observing that the Darboux problem still possesses a lot of structure, we show in the following that such a *space clothoid* element can be conveniently derived. The key is to introduce a fast and accurate integration scheme based on power series expansions. This numerical algorithm is then used as a formal computation tool to evaluate the spatial terms of the dynamics at a high precision.

[3] Coefficient of the highest derivative is 1.

4.2 Affine Curvature: the Space Clothoid Element

Discrete kinematics Let us discretize the rod into $N+1$ nodes with arc lengths s_i, $i \in \{0..N\}$. Similarly to the 2D super clothoid model [Bertails-Descoubes 2012], discrete curvature variables $\hat{\boldsymbol{\kappa}}_i$ are located at nodes s_i. On each element between two successive nodes s_i and s_{i+1}, the curvature vector $\boldsymbol{\kappa}(s)$ is assumed to vary linearly with arc length, so that its expression on element i of length ℓ_i (with $L = \sum_i \ell_i$) reads

$$\boldsymbol{\kappa}(s) = \left(1 - \frac{s - s_i}{\ell_i}\right)\hat{\boldsymbol{\kappa}}_i + \frac{s - s_i}{\ell_i}\hat{\boldsymbol{\kappa}}_{i+1} \quad \forall s \in [s_i, s_{i+1}].$$

In the following, we shall denote $\mathbf{q} \in \mathbb{R}^{3(N+1)}$ our state variable collecting all the degrees of freedom $\hat{\boldsymbol{\kappa}}_i$, and $\mathbf{q}^0$ the constant vector of same size storing the discrete intrinsic curvatures and twists $\hat{\boldsymbol{\kappa}}_i^0$. For now, we assume the centerline of the rod $\mathbf{r}$ can be computed as a function of s, $\mathbf{q}$, $\mathbf{r}_{\text{cl}}$ and $\mathscr{R}_{\text{cl}}$, by solving the Darboux problem (3) with an accurate numerical method. This difficult point is specifically tackled in Sections 5 and 6. Formally differentiating the centerline twice with respect to time leads to the following expression for acceleration,

$$\ddot{\mathbf{r}}(s,t) = \ddot{\mathbf{r}}^*(s,t) + \dot{\mathbf{q}}(t)\frac{\partial^2\mathbf{r}}{\partial\mathbf{q}^2}(s,t)\dot{\mathbf{q}}(t) + \frac{\partial\mathbf{r}}{\partial\mathbf{q}}(s,t)\ddot{\mathbf{q}}(t), \tag{6}$$

where $\ddot{\mathbf{r}}^*(s,t)$ is the acceleration generated by the clamping motion, which can be dropped when the clamped end is static. Expression (6) puts in evidence the linear dependency of the centerline acceleration $\ddot{\mathbf{r}}$ with respect to the reduced acceleration $\ddot{\mathbf{q}}$.

Discrete dynamics Discrete equations of motion result from a weak formulation of the strong Kirchhoff equations (4), where the trial functions are deduced from the constrained, piecewise affine kinematics. Consider an infinitesimal virtual displacement $\delta\mathbf{q}$ of our discrete degrees of freedom. This translates into a perturbation $\delta\boldsymbol{\kappa}$ in curvature, which causes an infinitesimal rotation of the material frame around a virtual rotation vector $\delta\boldsymbol{\theta}$, such that $\delta\mathscr{R} = [\delta\boldsymbol{\theta}]_\times\mathscr{R}$, as well as an infinitesimal displacement $\delta\mathbf{r}$ of the centerline. Applying the principle of virtual work [Reissner 1973] while considering an inextensible rod, as well as the boundary conditions given above, leads to the following weak formulation,

$$\int_0^L \left(\mathbf{M}'(s) + \mathbf{n}_0(s)\times\mathbf{T}(s)\right)\cdot\delta\boldsymbol{\theta}(s)\,\mathrm{d}s = 0,$$

where $\mathbf{T}(s) = \int_s^L \left(\mathbf{p}(s') - \rho S\ddot{\mathbf{r}}(s')\right)\mathrm{d}s'$. Integrating by parts and noting that $\delta\boldsymbol{\kappa}(s) = \delta\boldsymbol{\theta}'(s)$ and $[\delta\boldsymbol{\theta}(s)]_\times\mathbf{n}_0(s) = (\delta\mathbf{r})'(s)$, we get

$$\int_0^L \mathbf{M}(s)\cdot\delta\boldsymbol{\kappa}(s)\,\mathrm{d}s + \int_0^L \mathbf{p}(s)\cdot\delta\mathbf{r}(s)\,\mathrm{d}s = \rho S\int_0^L \ddot{\mathbf{r}}(s)\cdot\delta\mathbf{r}(s)\,\mathrm{d}s.$$

Finally, relating perturbed quantities to the virtual displacement $\delta\mathbf{q}$ and using Equation (6) yields the discrete dynamic equations

$$\mathbb{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbb{K}\left(\mathbf{q} - \mathbf{q}^0\right) + \mathbf{G}(\mathbf{q}) + \mathbf{A}(\mathbf{q},\dot{\mathbf{q}}) = 0 \tag{7}$$

where

$$\begin{aligned}
\mathbb{M}(\mathbf{q}) &= \rho S \int_0^L \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \, \mathrm{d}s \\
\mathbf{G}(\mathbf{q}) &= -\rho S \mathbf{g} \cdot \int_0^L \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \, \mathrm{d}s \\
\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}) &= \rho S \int_0^L \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \cdot \left(\dot{\mathbf{q}} \frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}^2} \dot{\mathbf{q}} + \ddot{\mathbf{r}}^* \right) \mathrm{d}s,
\end{aligned} \tag{8}$$

and where the constant stiffness matrix is defined as

$$\mathbb{K} = \begin{pmatrix}
\frac{\ell_0}{3}\mathbb{K}_3 & \frac{\ell_0}{6}\mathbb{K}_3 & 0 & \cdots & 0 \\
\frac{\ell_0}{6}\mathbb{K}_3 & \frac{\ell_0+\ell_1}{3}\mathbb{K}_3 & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \frac{\ell_{N-2}+\ell_{N-1}}{3}\mathbb{K}_3 & \frac{\ell_{N-1}}{6}\mathbb{K}_3 \\
0 & \cdots & 0 & \frac{\ell_{N-1}}{6}\mathbb{K}_3 & \frac{\ell_{N-1}}{3}\mathbb{K}_3
\end{pmatrix}.$$

The main challenge consists in evaluating vectors $\mathbf{G}$ and $\mathbf{A}$ and matrix $\mathbb{M}$ in a both *accurate* and *fast* way. Section 7 addresses this issue by demonstrating how our power series computation algorithm naturally extends to the evaluation of these dynamic terms. An efficient and stable time-stepping scheme is then derived to discretize Equation (7).

5 Kinematics Integration with Power Series

We now focus on solving the Darboux problem (3) under the assumption of an *affine* curvature vector. For simplicity, we shall consider here a single clothoidal element of length ℓ along end curvatures $\hat{\kappa}_0$ and $\hat{\kappa}_1$. The handling of a kinematic chain of N smoothly connected elements will be addressed in Section 6.5.

5.1 Solution of the Darboux Problem

When the curvature vector is affine (and even polynomial), the key idea is to formulate the solution of (3) as a *power series expansion* (PSE). This is made possible thanks to the following theorem:

Theorem 1 *Let $R > 0$ (possibly $R = +\infty$). If κ is C^∞ and admits a power series expansion $\kappa(s) = \sum_{n=0}^{\infty} \kappa_n s^n$ on $]-R, R[$, then the solution $\mathscr{F}$ of (3) is also C^∞ and admits a power series expansion $\mathscr{F}(s) = \{\sum_{n=0}^{\infty} \mathbf{r}_n s^n; \sum_{n=0}^{\infty} \mathscr{R}_n s^n\}$ on $]-R, R[$, recursively defined as*[4]

$$\begin{cases}
\mathscr{R}_0 = \mathscr{R}_{\mathrm{cl}} \quad \text{and} \quad \forall n \in \mathbb{N}, \quad \mathscr{R}_{n+1} = \dfrac{1}{n+1} \displaystyle\sum_{k=0}^{n} \mathscr{R}_k \left[\kappa_{n-k}\right]_\times \\
\mathbf{r}_0 = \mathbf{r}_{\mathrm{cl}} \quad \text{and} \quad \forall n \in \mathbb{N}, \quad \mathbf{r}_{n+1} = \dfrac{1}{n+1} \mathscr{R}_n \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\top.
\end{cases}$$

This theorem ensues from Cauchy's theorem on analytic solutions of linear ODEs with analytic coefficients (see, e.g., [Poole 1936] §2). In the particular case where κ is a polynomial, the theorem applies with $R = +\infty$ and thus $\mathscr{F}$ admits a power series expansion on $\mathbb{R}$. We now derive such a solution when κ is a polynomial of degree 1, that is, when the element is a space clothoid.

[4]Note that the coefficients a_n of the power series expansion $\sum_n a_n s^n$ of an analytic function $A(s)$ do not share the same physical dimension. On the contrary, the general term of the series, $\tilde{a}_n(s) = a_n s^n$, which is used in Section 5.2, is physically homogeneous to the sum $A(s)$ for all $n \in \mathbb{N}$.

Affine curvature Let γ be the slope of $\kappa(s)$, $\gamma = \frac{\hat{\kappa}_1 - \hat{\kappa}_0}{\ell}$. Recursions of Theorem 1 turn into

$$\begin{cases}
\mathscr{R}_0 = \mathscr{R}_{\mathrm{cl}} \\
\mathscr{R}_1 = \mathscr{R}_0 \left[\hat{\kappa}_0\right]_\times \\
\mathscr{R}_{n+2} = \dfrac{1}{n+2} \left(\mathscr{R}_{n+1} \left[\hat{\kappa}_0\right]_\times + \mathscr{R}_n \left[\gamma\right]_\times \right) & \forall n \in \mathbb{N} \quad (9a) \\
\mathbf{r}_0 = \mathbf{r}_{\mathrm{cl}} \\
\mathbf{r}_{n+1} = \dfrac{1}{n+1} \mathscr{R}_n \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\top & \forall n \in \mathbb{N}. \quad (9b)
\end{cases}$$

Computing the centerline thus follows from that of the material frame, which involves the recursive sequence (9a) of second order.

5.2 Numerical Computation of the Solution

Consider the power series for the material frame $\sum_n \mathscr{R}_n s^n$, where $s > 0$ is fixed. The study developed here with the aim of computing the sum $\mathscr{R}(s)$ analogously transposes to the evaluation of the centerline $\mathbf{r}(s)$.

Summation Since the series $\sum_n \mathscr{R}_n s^n$ is convergent on $\mathbb{R}$, we have $\|\mathscr{R}_n s^n\| \underset{n \to +\infty}{\to} 0$ which is equivalent to $\|\mathscr{R}_n\| = o\left(\frac{1}{s^n}\right)$, where $\|.\|$ is an arbitrary norm on matrices. In the following, we shall take the max norm $\|.\|_\infty$ which gives the maximum absolute value over all entries of the matrix. The fast decreasing of $\mathscr{R}_n$ thus compensates for the fast increasing of s^n when $s > 1$. Instead of computing the sequence of coefficients $\mathscr{R}_n$ using Recursion (9a) and then performing multiplications by s^n to get each term of the sum, it is numerically wiser to directly compute the general term of the series $\tilde{\mathscr{R}}_n(s) = \mathscr{R}_n s^n$ for all $n \in \mathbb{N}$ and express $\mathscr{R}(s)$ as the sum $\sum_{n=0}^{\infty} \tilde{\mathscr{R}}_n(s)$. From (9a) it easily follows that $\tilde{\mathscr{R}}_n(s)$ is defined as

$$\begin{cases}
\tilde{\mathscr{R}}_0 = \mathscr{R}_{\mathrm{cl}} \\
\tilde{\mathscr{R}}_1(s) = s \tilde{\mathscr{R}}_0 \left[\hat{\kappa}_0\right]_\times \\
\tilde{\mathscr{R}}_{n+2}(s) = \dfrac{s}{n+2} \left(\tilde{\mathscr{R}}_{n+1}(s) \left[\hat{\kappa}_0\right]_\times + s \tilde{\mathscr{R}}_n(s) \left[\gamma\right]_\times \right) & \forall n \in \mathbb{N}.
\end{cases} \tag{10}$$

Truncated Series In practice, the sum of the series has to be truncated in order to be numerically computed. The question is whether the summation can be pruned down without dropping relevant terms and making a large approximation error; if so, where? Luckily enough, it turns out that only the very first terms of the series are relevant, the following ones rapidly decreasing in norm and falling below the machine precision. This is due to the simple structure of our kinematic problem (3), which formulates as an explicit linear ODE with polynomial coefficients. In this case indeed, we can prove that the general term of the series super-linearly decreases to zero at the limit when n tends to infinity [Neher 1999].

In our iterative summation algorithm, we stop adding a new term as soon as its norm falls below the machine precision. In practice, for all the series we computed, this led to about 100 terms to be added together. Actually, when using our piecewise power series computation algorithm described in Section 6, the number of relevant terms to be added was even lower (≈ 20).

Computation of the sum: some severe numerical issues All the theory developed so far seems to nicely go along with our initial goal consisting in integrating the kinematics efficiently and accurately. However, when numerically evaluating the sum of the relevant terms in finite precision, one is inevitably faced with round-off issues leading to huge approximation errors. This is not a surprise. Textbooks on numerical integration usually recommend *not* to use power series expansion to solve differential equations, because of the risk of having to sum and subtract small values together with

large ones [Press et al. 2007]. This numerical *catastrophic cancellation* problem precisely occurs in our case, and is described in details in next section.

Why then persist in such a foolish direction? Because, in our case, this numerical issue can be effectively *remedied*. In next section, we devise an adaptive piecewise summation algorithm which guarantees that the sum of the power series can be safely computed, without catastrophic cancellation. Besides, we show that on a certain class of integration problems including the Darboux problem, our new algorithm turns out to reach high accuracy, *orders of magnitude* faster than traditional integration schemes such as Euler or Runge-Kutta methods. Ultimately, power series expansions and applicability of our algorithm nicely extends towards all the spatial terms of the dynamics, as demonstrated in Section 7, leading to a powerful space discretization method for our high-order rod model.

6 Fast and Precise Power Series Summation

6.1 Numerical Cancellation Issue

Consider expression $(1+y)-y$ which should be equal to 1 whatever the value of y. In floating-point arithmetic, this equality only holds if y is close enough to 1. In double precision for example, take $y = 10^{16}$ and compute the expression above. The numerical result is 0.0, yielding a relative error of 100%. This error is the consequence of first, an *absorption* phenomenon when computing the sum $1+10^{16}$, which, due to machine overflow when aligning mantissa, is approximated as 10^{16}. Then, a *cancellation* phenomenon when subtracting 10^{16}. Such unfortunate combination of absorption and cancellation leads to erroneous results and for this reason is called *catastrophic cancellation*. Details on floating-point arithmetic can be found, e.g., in [Goldberg 1991].

Catastrophic cancellation revealed When naively computing the sum of our power series $\sum \tilde{\mathscr{R}}_n(s)$ for a long and/or curly rod, we observed a dramatic loss of precision leading to erroneous results, as illustrated in Figure 3a. Let us figure out why catastrophic cancellation occurs in this case, and how it can be efficiently remedied.

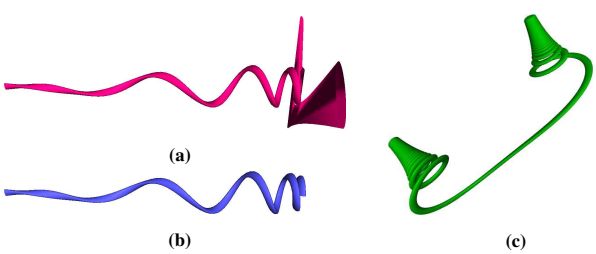


Figure 3: *(a): Dramatic loss of precision when naively summing power series of the kinematics. (b): In contrast, our piecewise summation algorithm guarantees high precision of the summation. (c): A long and highly curved space clothoid integrated with our piecewise computation method, using* 109 *subdivisions.*

In Figure 4 we have plotted in blue the norm of the general term $\tilde{\mathscr{R}}_n(s)$ function of n, for different values of s. The resulting "hillock"-like profile implies that when computing the sum of the series, one actually adds very small values together with very large ones in norm, the widest range being obtained when getting to the top of the hillock. Note that the larger s is, the higher the top of the hillock is. More precisely, in Section 6.2 we provide an explicit upper-bound $\mathscr{H}(s)$ for the top of the hillock (see Equation (12)). $\mathscr{H}(s)$ is shown to grow quasi-exponentially with the increasing function $\lambda(s)$, introduced in Equation (11). Moreover, as depicted in red by Figure 4, $\mathscr{H}(s)$ appears to match closely the increasing of the "top" of the hillock, function of s. Such a match helps one realize how fast and high the top of the hillock grows with s. Moreover, looking back to Recursion (9a), one notes that entries of the matrices to be added are of *alternating* sign, due to the product with skew symmetric matrices. This results in cancellation when computing the sum. All this combined together, it is then not surprising that we are faced with a catastrophic cancellation issue when $\lambda(s)$ (and thus s) becomes *too large*. As $\lambda(s)$ increases with s as well as with intrinsic curvatures (see Equation (11)), we now understand why numerical issues show up for a long and/or curly rod.

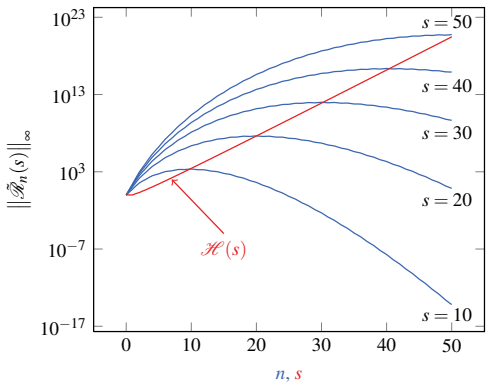


Figure 4: *Hillock-like profile of the general term $\tilde{\mathscr{R}}_n(s)$ in norm. In blue: Evolution of $\|\tilde{\mathscr{R}}_n(s)\|_\infty$ function of n (at fixed s), in log scale. As expected [Neher 1999], the decreasing towards 0 appears to be super-linear. In red: Evolution, function of s, of the upper-bound $\mathscr{H}(s)$ provided by Equation* (12)*, in log scale. Note that the plot of this upper-bound visually matches the maximum function* $\max_n \|\tilde{\mathscr{R}}_n(s)\|_\infty$*, meaning that the top of the hillock grows quasi-exponentially with s.*

Multi-precision schemes *vs.* floating-point arithmetic One common solution to avoid catastrophic cancellation is to work with algorithms allowing for arbitrary precision (see, e.g., [Neher 1999]). However, the required precision often needs to be fixed in advance, which only shifts the length/curvature's threshold beyond which catastrophic cancellation occurs. More importantly, as basic operations like addition are performed with a software library and not directly onto the machine processor, computations are considerably slowed down. We have tried to solve our kinematic problem with power series, using the `GMP` multi-precision library [Granlund and the GMP development team 2012], and observed a frame rate of 1 FPS for 100 sampling points, that is, about 1000 times as slow as with double precision (hardware) computations.

For us it is critical to precisely evaluate kinematic terms so as to correctly compute all spatial coefficients of the dynamic problem. However, getting an extreme accuracy beyond machine precision is unnecessary. In contrast, computational cost *has to* keep very low so as to get an interactive dynamic scheme — involving millions of operations per time step — for our rod primitive. Thus we rather stick to standard floating-point arithmetic and look for an *efficient* way of guaranteeing high precision when computing power series.

Splitting strategy Our summation method relies on an automatic subdivision of the integration domain into subintervals, on which integration can be safely performed. This subdivision strategy is similar in spirit to the splitting approach proposed by Neher [1999] to integrate explicit linear ODEs with polynomial coefficients accurately. However, as Neher's goal is to reach extreme accuracy while precisely quantifying the loss of precision, the focus is primary put on the use of multi-precision arithmetic and on the correct propagation of error enclosure, rather than on computational efficiency.

Splitting is only performed as a fail-safe process when the multi-precision scheme fails to encompass the total range of the summands. Furthermore, splitting is recursively performed through dichotomy by computing at each step a (costly) *recess* condition indicating whether the summands fall in the appropriate precision range or not. In contrast, our approach automatically computes once and for all the *right* subintervals which guarantee high-precision computations over the entire integration domain. Our splitting is efficiently computed and proves to be close to optimal, thanks to a simple yet precise upper-bound.

6.2 Limiting the range of summands

Let us first transform the recursive sequence (10) of second order into a recursive sequence of first order by introducing the 3×6 matrices $\tilde{\mathscr{V}}_n(s) = \left(\tilde{\mathscr{R}}_n(s), \tilde{\mathscr{R}}_{n-1}(s)\right)$, with $\tilde{\mathscr{V}}_0 = (\mathscr{R}_{\mathrm{cl}},\ 0)$. At fixed s, $\tilde{\mathscr{V}}_n(s)$ can be easily upper-bounded in norm as

$$\left\|\tilde{\mathscr{V}}_n(s)\right\|_\infty \leqslant \frac{\lambda^n(s)}{n!} \left\|\tilde{\mathscr{V}}_0\right\|_\infty \text{ with } \lambda(s) = 2s(\|\hat{\kappa}_0\|_\infty + s\,\|\gamma\|_\infty). \quad (11)$$

This upper-bound may be identified, up to a constant factor, to the general term of the exponential series $\tilde{e}_n(x) = \frac{x^n}{n!}$ at point $x = \lambda(s)$.

Let us now upper-bound the top of the hillock of Figure 4 and see how it grows with the variable s assumed to be positive. The maximum of the general term $\tilde{e}_n(x)$ over $n \in \mathbb{N}$ is reached when $n = \lfloor x \rfloor$, where $\lfloor . \rfloor$ denotes the floor function. At the maximum, the term of the series thus reads

$$\max_n \tilde{e}_n(x) = \tilde{e}_{\lfloor x \rfloor}(x) = \frac{x^{\lfloor x \rfloor}}{\lfloor x \rfloor !} \underset{x \to +\infty}{\sim} \frac{e^{\lfloor x \rfloor \left(1 + \log \frac{x}{\lfloor x \rfloor}\right)}}{\sqrt{2 \lfloor x \rfloor \pi}}, \quad (12)$$

meaning that for large $x = \lambda(s)$, the upper-bound $\mathscr{H}(s) = \max_n \tilde{e}_n(\lambda(s))$, which actually closely approximates the top of the hillock (see Figure 4), grows quasi-exponentially with $\lambda(s)$.

To avoid catastrophic cancellation, a natural idea then consists in upper-bounding x by a value M depending on the machine precision, so that the top of the hillock remains within the range where additions between two numbers can be safely performed, i.e., with no absorption of their leading digit. More precisely, if the machine has a precision of 10^{-d} ($d = 7$ for a floating number encoded on 32 bits, $d = 16$ on 64 bits), then the top of the hillock should be bounded by $10^{\frac{d}{2}}$ so as to be able to safely cover additions on the range $[10^{-\frac{d}{2}}, 10^{\frac{d}{2}}]$. Using Equation (12), one can easily prove that a sufficient upper-bound for M is

$$M \leqslant \max\left\{n \in \mathbb{N} \text{ s.t. } (n+1)^n \leqslant 10^{\frac{d}{2}} n!\right\}. \quad (13)$$

One gets $M \leqslant 19$ for $d = 16$. In practice, we set M to 10 to maintain good precision across summation. This choice allowed us to reach high precision for all the summations we have computed.

6.3 Adaptive Piecewise Summation (APS)

Consider again the norm of the general term of our first-order recursive sequence $\left\|\tilde{\mathscr{V}}_n(s)\right\|_\infty \leqslant \tilde{e}_n(\lambda(s)) \left\|\tilde{\mathscr{V}}_0\right\|_\infty$. From previous section, a sufficient condition to avoid catastrophic cancellation when computing the sum $\tilde{\mathscr{V}}(s) = \sum \tilde{\mathscr{V}}_n(s)$ is to have $\lambda(s) \leqslant M$ with M provided by Equation (13). Since λ is a second order polynomial in $s \geqslant 0$, this implies $s \leqslant s_{\max}(0)$ with

$$s_{\max}(\sigma) = \begin{cases} \frac{\sqrt{\|\kappa(\sigma)\|_\infty^2 + 2M\|\gamma\|_\infty} - \|\kappa(\sigma)\|_\infty}{2\|\gamma\|_\infty} & \text{if } \gamma \neq 0 \\ \frac{M}{2\|\kappa(\sigma)\|_\infty} & \text{else if } \kappa(\sigma) \neq 0 \\ +\infty & \text{otherwise} \end{cases} \quad (14)$$

and recalling that $\kappa(0) = \hat{\kappa}_0$.

More generally, suppose we have already computed $\tilde{\mathscr{V}}$ at a given point $\sigma_i \geqslant 0$. Then $\tilde{\mathscr{V}}$ can be safely evaluated through Recursion (10) at any s satisfying $\sigma_i \leqslant s \leqslant \sigma_i + s_{\max}(\sigma_i)$.

The idea then consists in splitting the evaluation domain $[0, \ell]$ into p adaptive subintervals $[0, \sigma_1], [\sigma_1, \sigma_2], \cdots [\sigma_{p-1}, \ell]$ such that $\sigma_{i+1} = \sigma_i + s_{\max}(\sigma_i)$. On each subinterval, summation is thus guaranteed to be performed with good accuracy (see an illustration in Figure 5).

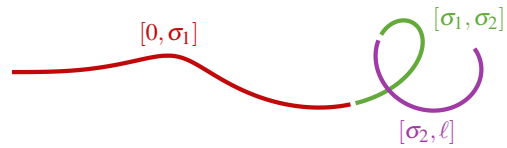


Figure 5: *Visual representation of our piecewise summation algorithm applied to the rod's kinematics (one clothoidal element). The length of each subinterval which guarantees a safe evaluation of the geometry is automatically provided by our method.*

From Equation (9b) it is clear that using the same subintervals, the centerline $\mathbf{r}(s)$ can be also safely computed, since the supplementary $\frac{1}{n+1}$ factor appearing in the power series coefficient only acts in favor of mitigating the norm of the general term of the series.

From Expression (14), more subdivisions are to be expected in curled parts than in straight ones, as depicted by Figure 5. In practice, the number of subdivisions used for our examples remained fairly low (around 10 to 20) and seldom reached more than one hundred. Moreover, we experimentally found out that our spatial upper-bound was close to *optimal*, as taking longer segments very often makes the computation algorithm fail. Figure 3b shows a typical example of clothoid integrated with our approach, and Figure 3c depicts an extreme case where the clothoidal element is lengthy and highly curved.

Summation algorithm Assume that we have a naïve routine `naiveSumDarboux(s, Rcl, kap0, gam)` for computing the sum $\mathscr{R}(s)$ at $s \geqslant 0$ (using Recursion (10)), with data `kap0` and `gam` as first end curvature and curvature slope, respectively. Then our new piecewise summation algorithm, which *safely* computes $\mathscr{R}(s)$ at any $s \geqslant 0$, simply reads

```
Matrix function adaptivePiecewiseSumDarboux(double s,
            Matrix Rcl, Vector kap0, Vector gam) {
  Matrix in = Rcl; // Current initial value
  sig0 = 0; sig1 = s_max(0); // Upper-bound given by (14)
  kap0c = kap0; // Curvature at first end point
  while( sig1 < s ) {
    // Compute the series at the end of the subinterval
    in = naiveSumDarboux(in, sig1-sig0, kap0c, gam);
    sig0 = sig1;
    sig1 = sig0 + s_max(sig0); // Updating upper-bound
    kap0c = kap0 + sig0*gam; // Updating kap0c
  }
  // Compute the series at s
  return naiveSumDarboux(in, s-sig0, kap0c, gam);
}
```

For the interested reader, we provide in supplemental material all the details for implementing our algorithm in the case of a simple 2D Cauchy problem on $SO(2)$.

6.4 Comparisons against classical integrators

We have applied our new integration method (APS) to the solving of the Darboux problem for a curly rod made of a single clothoidal element. The performance of APS was compared against 4 classical ODEs integrators: standard Euler (Euler), Euler on a Lie group

(Lie), Runge-Kutta of second (RK2) and forth (RK4) order. We first computed the geometry of the rod at a very high precision, with any of these integrators (all converged to the continuous solution). This high-precision geometry served as a reference for our comparisons. Then, we applied each integrator to our kinematic problem with a varying spatial step (corresponding to a varying truncature error for our method), and saved for each problem-solving the computational time as well as the numerical precision reached, measured as the L_2 distance to the reference.

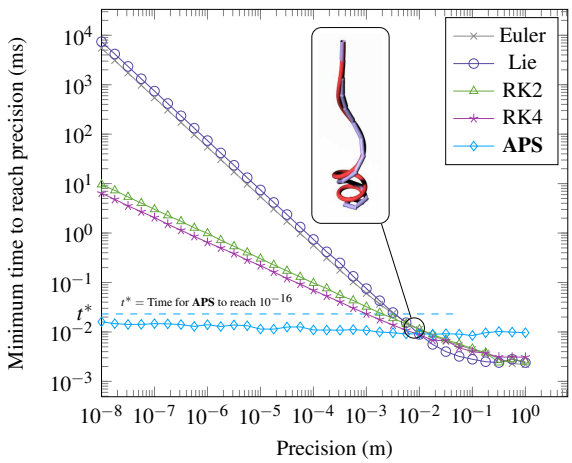


Figure 6: *Evaluation of our formal-like integrator (APS) compared to classical integrators, on the Darboux problem (reference rod in red). Our method largely outperforms all others.*

In Figure 6 we have plotted as a function of the precision the *minimum* computational time required to achieve the corresponding precision, in log scale. Apart from the bottom right region where precision is poor and corresponds to a visually large error, our method clearly and largely outperforms *all* other integrators. It even manages to reach machine precision with a very low timing $t^* = 2.6\ 10^{-2}$ ms, which appears to be out of reach for other integrators in a reasonable amount of time, whatever their order of convergence. Indeed, increasing the order of Runge Kutta from 2 to 4 does impact the number of performed time steps to reach a given precision, but *not* the total computational time required. The good performance of our method is explained by the fact that our computation algorithm leverages the particular structure of the problem consisting of an explicit linear ODE with polynomial coefficients.

Finally, although our summation method to compute the material frame is by no ways constrained to operate on $SO(3)$ (unlike the Lie integrator), the total sum $\mathscr{R}(s)$ appears to be, up to the machine precision, an *exact* rotation matrix, with no need for subsequent projection onto the $SO(3)$ manifold.

6.5 Computing a Kinematic Chain

Until now we have focused on a single clothoidal element. Our aim is now to build the kinematics of a full rod made of N connected elements with G^2-continuous junctions. This can be achieved by using a process very similar to our piecewise computation strategy for one element, presented above. Let $\mathbf{r}^i(s)$ and $\mathscr{R}^i(s)$ be respectively the local centerline and local material frame of element i. Denoting by $\mathscr{R}^i_n$ (resp. $\mathbf{r}^i_n$) the coefficient of degree n of the power series for $\mathscr{R}^i(s)$ (resp. $\mathbf{r}^i(s)$), continuity conditions between elements $i-1$ and i read

$$\begin{cases} \mathscr{R}^i_0 &= \mathscr{R}^{i-1}(\ell_{i-1}) \\ \mathscr{R}^i_1 &= \mathscr{R}^i_0\,[\hat{\kappa}_i]_\times \\ \mathscr{R}^i_{n+2} &= \frac{1}{n+2}\left(\mathscr{R}^i_{n+1}\,[\hat{\kappa}_i]_\times + \mathscr{R}^i_n \frac{1}{\ell_i}\,[\hat{\kappa}_{i+1}-\hat{\kappa}_i]_\times\right), \quad \forall n \in \mathbb{N} \\ \mathbf{r}^i_0 &= \mathbf{r}^{i-1}(\ell_{i-1}) \\ \mathbf{r}^i_{n+1} &= \frac{1}{n+1}\mathscr{R}^i_n(1\ 0\ 0)^\top, \quad \forall n \in \mathbb{N}. \end{cases}$$

7 Propagating Power Series to the Dynamics

7.1 Computing the coefficients of the temporal ODE

Our goal is now to accurately and efficiently compute vectors $\mathbf{G}$ and $\mathbf{A}$ and matrix $\mathbb{M}$ of the dynamic equation (7) for a super space clothoid. The expressions of these coefficients, given by (8), build upon the kinematics using 4 basic operations: (a) Linear combination; (b) Integration with respect to s; (c) Scalar product; and (d) Differentiation with respect to $\mathbf{q}$.

Invariance property A nice property is that, assuming the basic operands are convergent power series and that conditions for applying our piecewise summation are satisfied (which holds for $\mathscr{R}(s)$ and $\mathbf{r}(s)$, see Section 6), then the result of *each* operation (a), (b), (c), and (d) is also a convergent power series on $\mathbb{R}$ and our computation algorithm remains valid. Figure 7 sums up this nice invariance property, and provides the resulting power series as well as a suitable upper-bound for the corresponding general term, accounting for the validity of our summation algorithm.

Proving the convergence of a power series under linear combination, integration and product is straightforward. For the first two operations, the upper-bound given in Figure 7 is multiplied by a constant or a decreasing factor compared to the general term of the operand. Summation with our algorithm is thus guaranteed to operate within the range of high precision. Consider now the product of series, given by the Cauchy product. The $(n+1)$ factor in the upper-bound is actually not an issue as all scalar products appearing in Equation (8) are subsequently integrated, thus being multiplied by $\frac{1}{n+1}$. The product of maximums however raises up the top of the hillock, and theoretically, twice as much as subdivisions should be required to guarantee accurate summation of terms. In practice however, such a refinement proved to be unnecessary to maintain high precision.

Finally, power series and applicability of our summation algorithm also propagate through differentiation with respect to $\mathbf{q}$. This can be proved by considering the differential equation satisfied by the differentiated quantity – typically the material frame $\mathscr{R}(s)$ and the centerline $\mathbf{r}(s)$. The ODEs for $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ and $\frac{\partial \mathscr{R}}{\partial \mathbf{q}}$ read

$$\begin{aligned} \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}}\right)' &= \frac{\partial \mathscr{R}}{\partial \mathbf{q}}(1\ 0\ 0)^\top \\ \left(\frac{\partial \mathscr{R}}{\partial \mathbf{q}}\right)' &= \frac{\partial \mathscr{R}}{\partial \mathbf{q}}\,[\kappa]_\times + \mathscr{R}\frac{\partial\,[\kappa]_\times}{\partial \mathbf{q}} \end{aligned} \tag{15}$$

and are very similar to the kinematic equations. It is in particular easy to show that the supplementary term $\mathscr{R}\frac{\partial[\kappa]_\times}{\partial \mathbf{q}}$ in (15), which admits a power series expansion on $\mathbb{R}$, has no impact on the convergence of the solution nor on the profile of the general term of the corresponding power series.

Simple, fast and accurate computation From what precedes, all the terms given by Equation (8) are convergent power series on $\mathbb{R}$ and their sum can be computed with high precision thanks to our

Operation	Power series convergence on $\mathbb{R}$ and computation	Justification for applying our summation algorithm
$\mathbf{a}(s)+\alpha\,\mathbf{b}(s)$	$\sum_{n=0}^{\infty} a_n s^n + \alpha \sum_{n=0}^{\infty} b_n s^n = \sum_{n=0}^{\infty} v_n s^n = \sum_{n=0}^{\infty} (a_n + \alpha\, b_n)\, s^n$	$\lVert v_n s^n \rVert_\infty \leqslant \max_n \lVert a_n s^n \rVert_\infty + \lvert\alpha\rvert \max_n \lVert b_n s^n \rVert_\infty$
$\int_0^s \mathbf{a}(u)\,\mathrm{d}u$	$\int_0^s \sum_{n=0}^{\infty} a_n u^n du = \sum_{n=0}^{\infty} v_n s^n = \sum_{n=0}^{\infty} a_n \frac{s^{n+1}}{n+1}$	$\lVert v_n s^n \rVert_\infty \leqslant \frac{s}{n+1} \max_n \lVert a_n s^n \rVert_\infty$
$\mathbf{a}(s)\cdot\mathbf{b}(s)$	$\sum_{n=0}^{\infty} a_n s^n \cdot \sum_{n=0}^{\infty} b_n s^n = \sum_{n=0}^{\infty} v_n s^n = \sum_{n=0}^{\infty} \sum_{k=0}^{n} a_k \cdot b_{n-k} s^n$	$\lVert v_n s^n \rVert_\infty \leqslant (n+1) \max_n \lVert a_n s^n \rVert_\infty \cdot \max_n \lVert b_n s^n \rVert_\infty$
$\frac{\partial \mathbf{a}}{\partial \mathbf{q}}(s)$	$\mathbf{a}' = f(\mathbf{q},s)\,\mathbf{a} \Longrightarrow \left(\frac{\partial \mathbf{a}}{\partial \mathbf{q}}\right)' = \frac{\partial f}{\partial \mathbf{q}}\mathbf{a} + f(\mathbf{q},s)\frac{\partial \mathbf{a}}{\partial \mathbf{q}}$ Power series convergence and induction given by Cauchy's theorem	The ODE for $\frac{\partial \mathbf{a}}{\partial \mathbf{q}}$ fits the structure required (explicit, linear, with polynomial coefficients)

Figure 7: *Algebraic and differential operations preserve the structure of our problem and thus ensure the convergence of power series on* $\mathbb{R}$ *as well as the validity of our summation algorithm.*

adaptive piecewise computation algorithm. For instance, consider the mass matrix $\mathbb{M}(\mathbf{q})$. Its power series expression reads

$$\mathbb{M}(\mathbf{q}) \quad = \quad \sum_{i=0}^{N-1} \sum_{n=0}^{+\infty} \frac{(\ell_i)^{n+1}}{n+1} \sum_{k=0}^{n} \frac{\partial \mathbf{r}_k^i}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{r}_{n-k}^i}{\partial \mathbf{q}}.$$

We compute $\mathbb{M}$ by first, computing the general term of the power series of $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$. Then we compute the general term of the Cauchy product for this series, and finally sum across each element, using APS. Note how integration is simplified thanks to the use of power series. The most expensive operation actually appears to be the Cauchy product. In practice however, the overhead was not significant as the maximal number of terms involved in Cauchy products remained lower than 20.

7.2 Time-stepping scheme

For efficiency purposes, we discretized the dynamic equation (7) using a semi-implicit Euler scheme. Linear terms in $\mathbf{q}$ are handled in an implicit way while nonlinear terms are set explicit. As the stiffest terms (bending and twisting forces) are precisely linear in $\mathbf{q}$, such a simple scheme actually yields a fairly good stability while remaining very cheap. For scenarios involving fast and nervous motions (see Section 8.1), we found out that impliciting the term $\mathbf{A}$ at first-order in $\dot{\mathbf{q}}$ furthermore increased stability without adding too much overhead. In most of our demos, we have been using a large time step varying between 11 and 33 ms.

8 Validation and Results

In this section we carefully validate our new rod primitive and compare it to the most relevant ones in Computer Graphics. We also demonstrate the ability of our primitive to efficiently and faithfully reproduce various challenging phenomena involving arbitrary curled rods, from the growth of twining plants to the animation of curly hair. Corresponding animations are presented in the accompanying video.

8.1 Comparisons with previous models

Framework We compare our Super Space Clothoid primitive (SSC, with $3\,(N+1)$ degrees of freedom) against two inextensible rod models based on radically different discretization approaches:

- The *Super-Helix* model (SH), based on a piecewise *constant* curvature discretization of the kinematics. Similarly to our model, inextensibility is intrinsically captured and a semi-implicit time-stepping scheme is used where the internal elastic forces, which are linear, are made fully implicit. This model possesses $3N$ degrees of freedom when the rod is split into N helical elements.
- The *Discrete Elastic Rod* model (DER), based upon a nodal discretization of the centerline [Bergou et al. 2008]. In the case of curled rods, to avoid severe numerical instabilities, we opted for the fully implicit time-stepping scheme derived in [Bergou et al. 2010]. In this latest version of the model, nodal positions $\mathbf{r}_i$ as well as discrete twist angles θ_i are released as degrees of freedom, and inextensibility is enforced through a stiff (implicit) stretch force. For a rod split into N straight elements, the total number of degrees of freedom is $3\,(N+1)+N = 4N+3$.

In both cases, we used the *reference* implementation provided by the authors for our comparisons.

Finally, we designed two different rod simulations as benchmarks: (a) a straight rod falling under gravity and swinging in the plane, and (b) a long curly rod unwinding under gravity (see Figure 8).

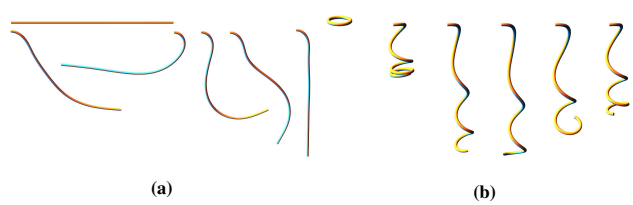


Figure 8: *Our two benchmarks (a) and (b), illustrated with our model* SSC *made of 5 elements.*

Validation and spatial convergence Our first experiment consists in computing the equilibrium position of the curled rod under gravity (b), using the three models SSC, SH, and DER, with a varying number of degrees of freedom (dof). At a high resolution, we notice that the three models converge *exactly* to the same configuration, which validates the consistency of our approach. We call this limit configuration the *reference* configuration. We then compare the geometric error (measured with the L_2 distance) between the reference configuration and the equilibrium configuration generated by each model[5], when the number of degrees of freedom varies. Results are shown in Figure 9, left, in log scale (similar results were obtained for experiment (a)). The higher order of convergence of our approach compared to others is clearly demonstrated.

[5] For DER, smoothing the centerline with a spline didn't improve results.

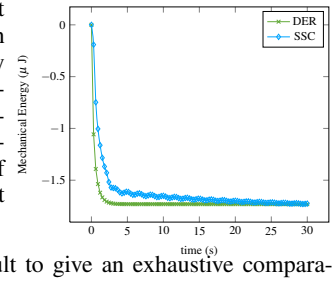
At the limit of the visual distinction to the reference, our model requires around 3 times less dofs than SH, and 50 times less than DER. Even with a low spatial resolution, our model provides good accuracy, whereas DER requires at least 200 dofs (≈ 50 elements) to generate a reasonable equilibrium configuration. To closely approximate the reference, 1000 dofs (≈ 250 elements) are required for DER compared to 70 dofs (≈ 23 elements) for SH and 24 dofs (≈ 7 elements) for our model. In order to free ourselves from variations in time-stepping schemes between the three models, as well as from their poor (only first) order of convergence — all this making comparisons to a *dynamic* reference very tricky — we have chosen to measure accuracy on the *static* rod configuration. Yet, one can reasonably imagine that such measured accuracy is a reliable indicator of the global spatial accuracy achieved *all along* motion.

Computational time for a nervous motion We now measure the computational time required to simulate our dynamic experiments (a) and (b). To enhance richness of motion, friction parameters are kept small. We then choose the same timestep ($dt = 11\,\text{ms}$) for all models, and plot the total computational time function of the number of dofs, for each model. Figure 9, middle, gives the resulting plots for the curly experiment (b). Very similar plots are obtained for the straight rod experiment (a). As expected, for a given number of dofs, our model requires more computations than SH which itself appears to be more costly than DER. Note also that the performance of DER better (linearly) scales up with dofs compared to the two other models. This is due to the sparse implementation of DER, compared to the dense structure of reduced models. However, as already mentioned, our model requires much less dofs than other approaches (especially DER) to generate accurate spatial configurations and to closely match reference static equilibria. This raises up the fundamental question of the *trade-off* between accuracy and computational time, which is addressed below.

Accuracy *vs.* computational time To get a hint of the accuracy *vs.* cost trade-off for the different rod models, a natural idea is to connect the accuracy plot (Figure 9, left) to the cost plot (Figure 9, middle). The intersection of both plots yields Figure 9, right. Interestingly, plots for SH and SSC cross each other above the visual limit for precision, and before the real-time limit. In the low-accuracy zone, the super helix model provides a better trade-off than our model. Note however that our accuracy measure does not take into account *fairness*, which, at such low resolutions, proves to be quite poor with SH compared to SSC (see Figure 2). When getting to the high accuracy zone, our model starts to behave better than SH. In this case indeed, the computational overhead for computing each element becomes compensated by the gain in accuracy.

Finally, on nervous motions such as those generated by our two benchmarks (a) and (b), DER clearly under-performs compared to the two reduced models. Indeed, DER is first penalized by the complex geometry of the rod, which requires a large resolution to be correctly represented and/or mechanically deformed. Second, due to the fast unwinding (resp. swinging) of the curly (resp. straight) rod, which causes a fast increase in stretching and bending terms, the Newton solver requires a large number of iterations to converge. Since DER is in practice almost always stable, one could increase its performance by raising up the timestep to a larger value, i.e., $dt = 33\,\text{ms}$. We re-run the two simulations (a) and (b) with this new timestep for all models. Our model coped with it and we again obtained the same plot profile.

Energy preservation To assess the richness of motion yielded by each model, we simulate the swinging motion (a) and plot the mechanical energy function of time. All friction parameters are set to 0 so that dissipation is caused by numerical damping only, and we use the same timestep ($dt = 11\,\text{ms}$) for all models. As SH is unstable at this timestep (while the two others remain stable even for $dt = 33\,\text{ms}$), we compare our model to DER only. We notice that the motion generated by DER is more damped than ours, as depicted by the substantial loss of energy right at the beginning of the motion. We observed the same phenomenon on experiment (b). Actually, such numerical damping is mainly due to the implicit treatment of stretch terms, which is necessary for DER to stably preserve inextensibility. In contrast, as our model is intrinsically inextensible, high frequencies due to a fast unwinding of the rod are much less filtered out by the time integrator.



Discussion It is quite difficult to give an exhaustive comparative study of such different rod models: each approach has its own strengths and weaknesses, and depending on the context of usage, the best trade-off may not be uniquely determined. In our study, we deliberately focused on the simulation of the *nervous* (as opposed to damped) motion of arbitrarily *curly* rods. In that context, we observed that our super space clothoid model offered the best trade-off in terms of spatial accuracy, stability and computational time. Indeed, although pushing upward the order of elements has a price to pay, this price is mitigated by the high resulting accuracy which especially outperforms that of the super helix model. Moreover, while our time-stepping is not fully implicit, it proves to be sufficiently stable in most scenarios thanks to the (inexpensive) implicit handling of linear stiff terms, in particular of elastic forces. In contrast, the main strength of the discrete elastic rod model is definitely its great stability, which proved to be essential for simulating fast and nervous motions of curly rods — an explicit time-stepping scheme requiring a way too small timestep to remain stable. However, stability has a price to pay, first in term of computational cost. This price is exacerbated in the case of a nervous motion, since the Newton solver needs much more iterations to converge. Second, some substantial numerical damping, mainly caused by implicit stretch terms, contributes to filter out high frequencies, thus impoverishing the original motion.

Note that our (modest) dynamic study relied for each model on the most stable time-stepping scheme implemented among available codes. Models could be coupled to more sophisticated integrators, with higher order of accuracy and better energetic properties [Hairer et al. 2006]. In the future we would like to push forward the dynamic comparisons between models in such settings.

8.2 Coupling with contact

Twining plant The growth of twining plants around a pole is a fascinating problem that raises several interesting questions such as the mechanical ability of a plant to climb depending on the width of the pole, or the topology of contact that is involved during growth [Goriely and Neukirch 2006]. In their paper, Goriely and Neukirch model this phenomenon as a spatial boundary value problem by considering a naturally curled Kirchhoff rod with constant intrinsic curvature $\hat{\kappa}$, contacting at both end points a cylinder of radius R, with $\hat{\kappa} > \frac{1}{R}$ (otherwise the plant could not twine around the pole). In the 2D frictionless case, by studying bifurcations of the underlying dynamical system (space playing the role of time) they show that there exists a unique critical value ρ_c for the curvatures ratio $\rho = \hat{\kappa} R$ below which the rod is able to climb, and numerically estimate (through a shooting strategy) that $\rho_c \approx 3.3$. In the favorable configuration ($\rho \leqslant \rho_c$), it is demonstrated that the contact topology remains stationary as the plant twines around the pole, and that it is characterized by an isolated contact point at the tip (the tip making a constant angle with the pole) followed by a region without contact, and finally a continuous contact zone between the pole

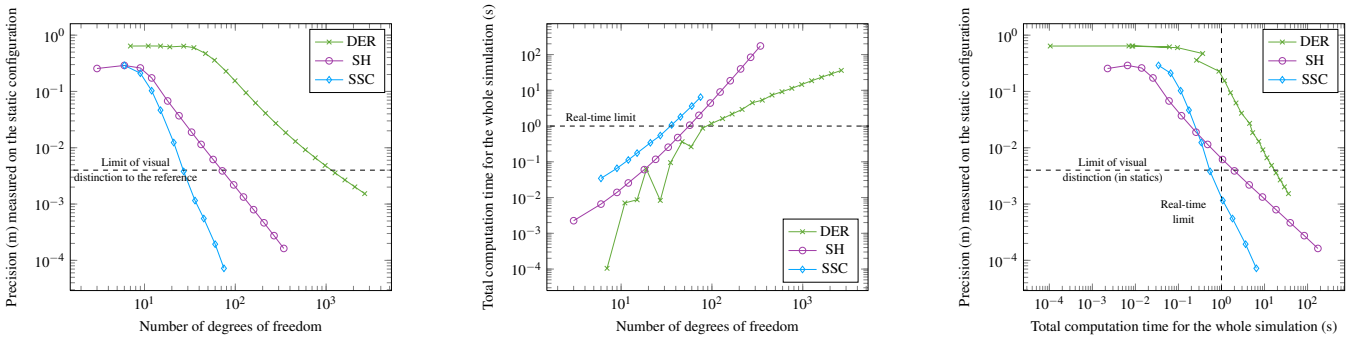


Figure 9: *Comparisons between our rod model (*SSC*) and previous models* SH *and* DER*, in terms of accuracy and efficiency.*

and the remainder of the rod. In the unfavorable case ($\rho > \rho_c$), the curve rolls on itself and loses its grip.

We simulated such a 2D frictionless growth process with our own rod model coupled to a (penalty-based) contact solver, and were able to capture all these phenomena accurately (see Figure 10a). We especially retrieved the critical value $\rho_c = 3.3$ up to a 1% precision, using only $N = 5$ clothoidal elements. In the 3D case, the configuration of a twining plant is modeled in [Goriely and Neukirch 2006] as a generalized helix with non-uniform pitch. We simulated the growth of our rod model in 3D (see Figure 10b) and indeed observed at any stage of the growing process a variation in pitch along the centerline, especially emphasized near the tip. Note that we have been using only $N = 10$ clothoidal elements for the full simulation generating 5 loops around the pole.

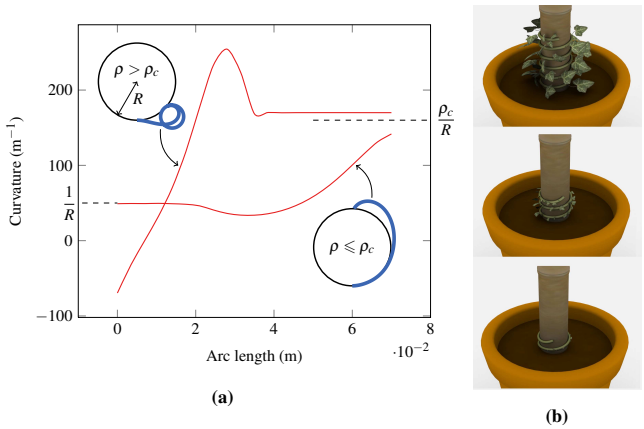


Figure 10: *Twining plant. (a): 2D experiment capturing the threshold for twining, and corresponding curvature profiles. (b): 3D experiment, with $N = 10$ clothoidal elements. Observe the variation in pitch near the tip.*

Realistic and stylized hair animation Our rod model was seamlessly coupled with the frictional contact solver presented in [Daviet et al. 2011]. To illustrate the versatility of our model, we simulated various hair scenarios involving hair-body and hair-hair contacts with Coulomb friction ($\mu = 0.2$). We first animated two hairstyles made of 662 rods with 4 elements per rod: one smooth, and the other highly curly. Note that in the second case, due to the strong entangling between fibers — correctly taken into account by the frictional solver —, we reproduced the bulk, quasi-rigid motion characteristic of fuzzy hair. We also designed and animated an aesthetic curly hair inspired from SIGGRAPH's Animation Mother [CGSociety 2008] using 100 rods and 1 up to 4 clothoidal elements per rod (only 1.55 element per rod on average). Results are shown in Figure 11 and in the accompanying video. This example captures well each strand's own nonlinear dynamics, as well as the strong, frictional entangling between strands.

Figure 11: *Aesthetic simulation of a stylized hair, with 100 super space clothoids made of 1.55 clothoidal elements on average.*

8.3 Towards the animation of curled surfaces

Finally, though assumptions for a thin elastic rod do not strictly hold anymore, we found that with an exaggerate flat and long cross-section, our primitive was able to generate rich shapes and motion for a small surface in a very efficient way, by using only two or even one single element. We used this interesting feature to simulate in real-time the geometry and motion of a curled ribbon, a poster and a rolled parchment (see Figures 1 and 12). These simple but striking examples illustrate the particular richness of our space clothoid element, both in terms of geometry and mechanical deformation.

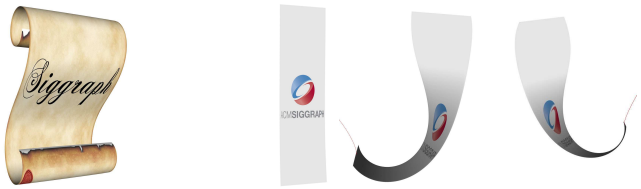


Figure 12: *Rolled parchment and poster, simulated as one single clothoidal element with an exaggerate flat and long cross-section.*

Conclusion

We have introduced a high-order rod primitive based on 3D clothoidal elements. Accurate and fast spatial discretization is achieved thanks to a new formal-like integrator based on power series and adapted to floating-point arithmetic. Our model is successfully applied to the simulation of straight as well as highly curly strands like curled ribbons, plants and hair, with a very low number of elements. We also thoroughly compare our new rod model to the most relevant ones in graphics, and show that our model offers

a better trade-off in terms of spatial accuracy, richness of motion, and efficiency. Such a study is particularly difficult in the dynamic settings as time-stepping schemes are of low order, and greatly vary from one model to the other. In the future, we would like to refine this study by increasing the order of accuracy of the time-stepping schemes as well as better preserving energy. Finally, we believe that extending our new high-order rod primitive to plates and shells would allow one to efficiently and accurately simulate the deformations of detailed geometric surfaces like cloth folds and wrinkles.

Acknowledgments

We would like to thank Laurence Boissieux for her artistic contribution to the paper, Gilles Daviet for producing the hair demos, Alexandre Derouet-Jourdan and Martin Guay for proofreading the paper, and Marc Mezzarobba and Sébastien Neukirch for many fruitful discussions. We are also very grateful to Eitan Grinspun's group for sharing their code on Discrete Elastic Rods. Finally we wish to thank the anonymous reviewers for their helpful comments.

References

ANTMAN, S. 1995. *Nonlinear Problems of Elasticity*. Springer Verlag.

AUDOLY, B., AND POMEAU, Y. 2010. *Elasticity and Geometry: from hair curls to the nonlinear response of shells*. Oxford University Press.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'98)*, 43–54.

BENHAM, C., AND MIELKE, S. 2005. DNA mechanics. *Annual Review of Biomedical Engineering 7*, 21–53.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'08) 27*, 3, 1–12.

BERGOU, M., AUDOLY, B., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2010. Discrete viscous threads. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'10) 29*, 4.

BERTAILS-DESCOUBES, F. 2012. Super-clothoids. *Computer Graphics Forum (Proc. Eurographics'12) 31*, 2pt2, 509–518.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'06) 25*, 1180–1187.

CGSOCIETY, 2008. SIGGRAPH's Animation Mother. http://www.cgsociety.org/index.php/CGSFeatures/CGSFeatureSpecial/siggraph_animation_mother.

CHENTANEZ, N., ALTEROVITZ, R., RITCHIE, D., CHO, L., HAUSER, K., GOLDBERG, K., SHEWCHUK, J., AND O'BRIEN, J. 2009. Interactive simulation of surgical needle insertion and steering. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'09)*, 88:1–10.

COSSERAT, E., AND COSSERAT, F. 1909. *Théorie des corps déformables*. Hermann.

CRISFIELD, M. A., AND JELENIĆ, G. 1998. Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation. *Proc. Royal Society of London, Series A 455*, 1983, 1125–1147.

DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia'11) 30*, 139:1–139:12.

DILL, E. 1992. Kirchhoff's theory of rods. *Archive for History of Exact Sciences 44*, 1, 1–23.

GOLDBERG, D. 1991. What every computer scientist should know about floating-point arithmetic. *ACM Comp. Surveys 23*, 5–48.

GORIELY, A., AND NEUKIRCH, S. 2006. Mechanics of climbing and attachment in twining plants. *Physical Review Letters 97* (Nov), 184302.

GOYAL, S., PERKINS, N., AND LEE, C. 2008. Non-linear dynamic intertwining of rods with self-contact. *International Journal of Non-Linear Mechanics 43*, 1, 65 – 73.

GRANLUND, T., AND THE GMP DEVELOPMENT TEAM. 2012. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 5.0.5 ed. http://gmplib.org/.

HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. *Computer Graphics Forum (Proc. Eurographics'01) 20*, 3, 329–338.

HADAP, S. 2006. Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'06)*, ACM-EG SCA, 91–100.

HAIRER, E., LUBICH, C., AND WANNER, G. 2006. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, vol. 31. Springer Series in Comput. Mathematics.

HARARY, G., AND TAL, A. 2012. 3D Euler spirals for 3d shape completion. *Computational Geometry 45*, 3 (April), 115–126.

IVANOVA, E. 2000. On one approach to solving the Darboux problem. *Mechanics of Solids 35*, 36–43.

MEZZAROBBA, M. 2010. NumGfun: a package for numerical and analytic computation with D-finite functions. In *ISSAC '10*.

NEHER, M. 1999. An enclosure method for the solution of linear ODEs with polynomial coefficients. *Numerical Functional Analysis and Optimization 20*, 779–803.

PAI, D. 2002. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum (Proc. Eurographics'02) 21*, 3, 347–352.

POOLE, E. 1936. *Introduction to the theory of linear differential equations*. Clarendon Press.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 2007. *Numerical Recipes: The Art of Scientific Computing (Third Edition)*. Cambridge University Press.

REISSNER, E. 1973. One one-dimensional large-displacement finite-strain beam theory. *Studies in App. Math. 52*, 2, 87–95.

SIMO, J., AND VU-QUOC, L. 1986. A three-dimensional finite-strain rod model. part ii: Computational aspects. *Computer Methods in Applied Mechanics and Engineering 58*, 1, 79 – 116.

SPILLMANN, J., AND TESCHNER, M. 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'07)*, ACM-EG SCA, 63–72.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S., CANI, M.-P., AND LIN, M. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 13*, 2 (Mar-Apr), 213–34.