Super-Clothoids

Florence Bertails-Descoubes

INRIA, Grenoble, France

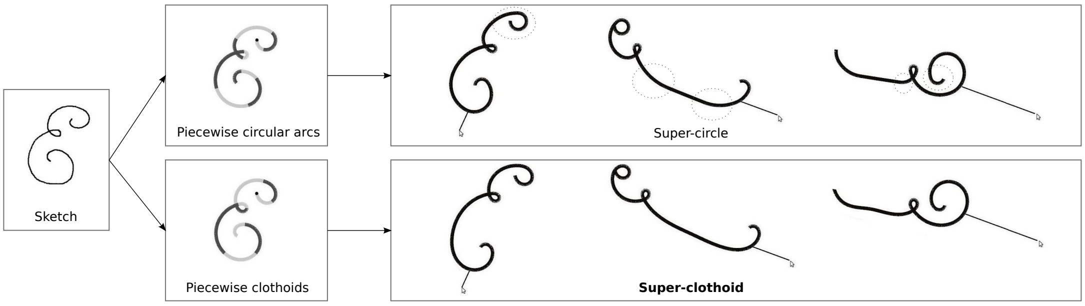


Figure 1: *A smooth sketch is first converted into either G^1 piecewise circular arcs or G^2 piecewise clothoids, using Baran et al.'s fitting method with a fixed geometric error [BLP10]. The two primitives are then converted into a super-circle [BAC*06] and into our new super-clothoid model, respectively. While the initial shape looks fair for both models, during deformation the super-circle exhibits displeasing discontinuities (dashed circled) whereas the super-clothoid always keeps a smooth shape.*

Abstract

Piecewise clothoids are 2D curves with continuous, piecewise linear curvature. Due to their smoothness properties, they have been extensively used in road design and robot path planning, as well as for the compact representation of hand-drawn curves. In this paper we present the Super-Clothoid model, a new mechanical model that for the first time allows for the computing of the dynamics of an elastic, inextensible piecewise clothoid. We first show that the kinematics of this model can be computed analytically depending on the Fresnel integrals, and precisely evaluated when required. Secondly, the discrete dynamics, naturally emerging from the Lagrange equations of motion, can be robustly and efficiently computed by performing and storing formal computations as far as possible, recoursing to numerical evaluation only when assembling the linear system to be solved at each time step. As a result, simulations turn out to be both interactive and stable, even for large displacements of the rod. Finally, we demonstrate the versatility of our model by handling various boundary conditions for the rod as well as complex external constraints such as frictional contact, and show that our model is perfectly adapted to inverse statics. Compared to lower-order models, the super-clothoid appears as a more natural and aesthetic primitive for bridging the gap between 2D geometric design and physics-based deformation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Deformable 1D structures are ubiquitous in our real environment as well as in imaginary, human-created worlds. Such long and thin objects can depict ropes, hair, grass, trees, and even the shape of animals or humans. In 2D, smooth curve-like shapes naturally emerge from the drawing of lines and serve as core primitives for many artistic techniques such as arabesques or calligraphy. Computing their deformation in a realistic way represents an important challenge, for both interactive design applications and 2D animation. In this paper we focus on the physics-based simulation of a widely used class of 2D geometric curves: *piecewise clothoids*.

1.1. Related Work

Geometric curve design In geometric design, much work has been devoted for more than half a century to the search for curve primitives adapted to the interactive and controllable design of objects. Spline primitives have been acknowledged as the ideal primitives for applications requiring some good and fast controllability under positional or tangential constraints. *Fairness*, characterizing the aesthetic of a curve or a surface, has also become an important criterion for quality in engineering design [Sap94]. Among all the different metrics that have been proposed to measure fairness, one popular criterion is to consider that a fair curve should possess a slowly varying curvature [MS92, HT10]. *Clothoids*, characterized by a linear curvature, are precisely curves that minimize the curvature variation, subject to endpoint constraints [Lev08]. They are thus renown for their visually pleasing appearance, and for this reason have recently found various applications such as shape completion in computer vision [KFP03] or the compact and automatic representation of artistic sketched lines in computer graphics [MS08, BLP10].

Physics-based animation of curves On the other hand, the animation of slender structures has become a very active field of research in computer graphics in recent years, and various mechanical models have been proposed for modeling the dynamics of 3D inextensible rods with elastic bending and twisting. While many authors have relied on an explicit 3D parameterization of the centerline [LMGC04, ST07, ST08, TGAB08, BWR*08, SLF08, BAV*10], others have preferred to design compact rod models parameterized by a minimal set of physical parameters that fully describe the kinematics of the rod, without the need for adding any extra constraints [BAC*06, Had06, Ber09]. As a result, the latter class of approaches can capture perfect inextensibility of the rod (and thus, "nervous" motions) while avoiding the recourse to some large stretching energy [ST07, ST08], which may ruin the stability of the model, or to projection schemes [BWR*08, BAV*10], which may cause some artificial energy loss.

In the vein of reduced-coordinates models, the super-helix model [BAC*06] has been proposed as a high-order primitive (helical elements in 3D, circular elements in 2D) able to accurately capture the motion of complex geometric curves - as curly as desired - with only a few elements. Moreover, as the model is parameterized by the curvature, the internal elastic forces - linear in curvature - can be computed using a fully implicit scheme, in contrast to other approaches. As a result, the model is guaranteed to remain very stable even under large displacements or when using a large time step. The degree of representation of this model is nevertheless limited to the first order, which is not sufficient for generating curves that are appealing to the eye, *i.e.*, that satisfy the fairness property. Note that some previous approaches do simulate the dynamics of at least G^2-smooth curves such as splines, see *e.g.* [LMGC04]. However, such models are parameterized using the maximal-coordinates formulation, which raises the stability and energy preservation issues mentioned above when attempting to enforce the kinematics of the rod. In this paper, we would like to go one step further in the formulation of high-order *reduced-coordinate* models, by proposing a G^2-smooth (instead of G^0 or G^1-smooth) dynamic curve primitive made of elements with linear curvature (clothoids) instead of constant curvature (segments or circular arcs).

From curve design to physics-based animation Recently, Derouet-Jourdan et al. [DJBDT10] showed that reduced-coordinates models such as the super-helix model are particularly suitable for solving the inverse statics problem: there always exists a simple and intuitive solution that identifies the physical parameters of the rod model such that any input curve corresponds to a stable configuration of the rod at rest under gravity. Applications include 2D physically-based animation in animated movies and video games, based on a "what you draw is what is simulated" concept. However, with this approach, input curves first need to be approximated by piecewise circular arcs in order to conform to the geometry of the 2D super-helix model. In contrast, piecewise clothoids nicely unify curve drawing and curve simulation while guaranteeing the fairness property from the design to the animation.

1.2. Contributions

- We introduce the super-clothoid model, a new dynamic primitive for simulating the motion of an inextensible and elastic piecewise clothoid. While being extremely compact in term of richness of representation and guaranteeing G^2-smoothness at any resolution, this model robustly and efficiently captures the deformations of rods under large displacements, with only a few elements.
- We handle various external forces, bilateral and unilateral constraints with or without friction, and all possible boundary conditions for this model. Such a versatility can be exploited to simulate a wide variety of physics-based objects, ranging from simple cantilever beams to free closed contours of deformable bodies colliding with the ground.
- We exactly and efficiently solve the inverse statics problem, in a similar spirit as in [DJBDT10]. This allows us to bridge the gap between, on the one hand, the geometric design of smooth curves, which often calls for the use of piecewise clothoids, and, on the other hand, the realistic deformation of the designed curves within a physics-based environment.

These contributions are described in Sections 2 and 3, Section 4, and Section 5, respectively. Section 6 validates our model against previous approaches and presents our results, before concluding.

2. Kinematics of a Super-Clothoid

Let $\mathbf{r}(s)$ be the centerline of the rod parameterized by the curvilinear abscissa s, L its length, $\theta(s)$ and $\kappa(s) = \frac{d\theta}{ds}$ its angular and curvature functions respectively, as depicted in Figure 2. For now, the rod is assumed to be clamped at the end $s = 0$ with clamping position $\mathbf{r}_0 = \mathbf{r}(0)$ and clamping angle $\theta_0 = \theta(0)$, and free at the other end $s = L$. In Section 4.1, we'll see how to relax this clamping constraint and, conversely, how to constrain both ends, so as to extend the range of modeled phenomena.

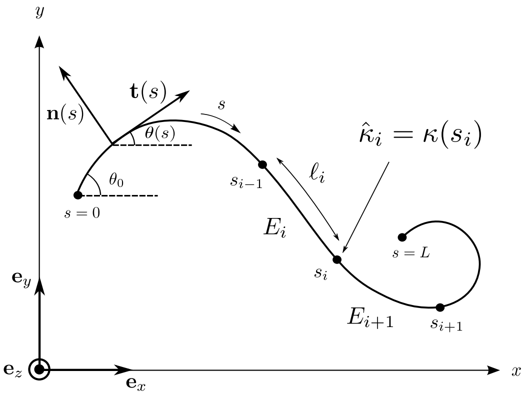


Figure 2: *Notations for a super-clothoid.*

Let N be the number of elements composing a super-clothoid, and E_i the i^{th} element of the rod, with length ℓ_i. We use the term *node* for the junction point between two consecutive elements E_i and E_{i+1}, located at the curvilinear abscissa s_i (see Figure 2). By convention we have $s_0 = 0$ and $s_N = \sum_{i=1}^{N} \ell_i = L$. Assuming that the curvature $\kappa(s)$ varies lineary over each element amounts to choosing $N+1$ scalar parameters $\hat{\kappa}_{i,i\in\{0,\ldots,N\}}$, located at the nodes s_i, for describing the geometry of the rod. A super-clothoid made of N elements is thus composed of $N+1$ discrete curvatures $\hat{\kappa}_i$, that are collected in vector $\boldsymbol{\kappa}$.

2.1. A Single Clothoidal Element

Let us consider the element E_i, characterized by its two extremal curvatures $\hat{\kappa}_{i-1}$ (left node) and $\hat{\kappa}_i$ (right node) and by its length ℓ_i. On the element E_i, the curvilinear abscissa s ranges between s_{i-1} and s_i. For the sake of clarity, we denote by u the local curvilinear abscissa $u = s - s_{i-1}$ ranging between 0 and ℓ_i on the element E_i. The *local* curvature and angular functions $\kappa_i(u)$ and $\theta_i(u) = \theta_i(0) + \int_0^u \kappa_i$ thus read

$$\kappa_i(u) = \frac{\ell_i - u}{\ell_i}\,\hat{\kappa}_{i-1} + \frac{u}{\ell_i}\,\hat{\kappa}_i \qquad (1)$$

$$\text{and} \qquad \theta_i(u) = \theta_i(0) + (u - \frac{u^2}{2\,\ell_i})\,\hat{\kappa}_{i-1} + \frac{u^2}{2\,\ell_i}\,\hat{\kappa}_i.$$

Let $P_i = a_i\,u^2 + b_i\,u + c_i$ be a second-order polynomial with $a_i = \frac{\hat{\kappa}_i - \hat{\kappa}_{i-1}}{2\,\ell_i}$, $b_i = \hat{\kappa}_{i-1}$ and $c_i = \theta_i(0)$. Then $\theta_i = P_i(u)$ and $\kappa_i(u) = b_i + 2\,a_i\,u$. The tangent and normal vectors $\mathbf{t}_i$ and $\mathbf{n}_i$ respectively read

$$\mathbf{t}_i(u) = \begin{pmatrix} cos\,(P_i(u)) \\ sin\,(P_i(u)) \end{pmatrix} \qquad \mathbf{n}_i(u) = \begin{pmatrix} -sin\,(P_i(u)) \\ cos\,(P_i(u)) \end{pmatrix}.$$

By integration of the tangent vector, we can compute the centerline vector as

$$\mathbf{r}_i(u) = \mathbf{r}_i(0) + \begin{pmatrix} I_i(u) \\ J_i(u) \end{pmatrix}$$

where the two integral functions $I_i(u) = \int_0^u cos(P_i(u'))\,\mathrm{d}u'$ and $J_i(u) = \int_0^u sin(P_i(u'))\,\mathrm{d}u'$ can be analytically computed as functions of the Fresnel integrals $\int_0^u cos(x^2)\,\mathrm{d}x$ and $\int_0^u sin(x^2)\,\mathrm{d}x$. In the following however, we shall directly use $I_i(u)$ and $J_i(u)$ as base functions for analytically deriving the kinematics of the super-clothoid. When required, these two integrals will be evaluated numerically using Romberg integration [PTVF07].

2.2. A G^2 Chain of N Clothoidal Elements

In this section we derive the kinematics of a full super-clothoid composed of N clothoidal elements with G^2-smooth junctions at nodes. The first element E_1 is clamped at the left end $s = 0$ with the clamping angle θ_0.

Curvature and angle We first aim at deriving the *global* expressions for the curvature and angular functions $\kappa(s)$ and $\theta(s)$. Let us start by decomposing the curvature function $\kappa(s)$ over each element E_i, as

$$\kappa(s) = \sum_{1 \le i \le N} \kappa_i(s - s_{i-1})\,\chi_i(s) \qquad (2)$$

where $\chi_i(s)$ equals 1 for $s \in [s_{i-1}, s_i]$ and 0 elsewhere.

Using expression (1) for the local function $\kappa_i(u)$, we can reformulate the global curvature function as

$$\kappa(s) = \sum_{0 \le i \le N} \hat{\kappa}_i\,F_i(s) \qquad (3)$$

where $F_i(s)$ is the C^0-smooth piecewise linear shape function

$$F_i(s) = \begin{cases} 0 & \text{if } s \le s_{i-1} \text{ or } s \ge s_{i+1} \\ \frac{s - s_{i-1}}{\ell_i} & \text{if } s_{i-1} \le s \le s_i \\ \frac{\ell_{i+1} - (s - s_i)}{\ell_{i+1}} & \text{if } s_i \le s \le s_{i+1}. \end{cases} \qquad (4)$$

For the completeness of Equation (4) we assume that $s_{-1} = s_0 + \varepsilon$, $s_{N+1} = s_N + \varepsilon$ and $\ell_0 = \ell_{N+1} = \varepsilon$ with $\varepsilon \to 0$. By integration of Equation (3), we have

$$\theta(s) = \theta_0 + \sum_{0 \le i \le N} \hat{\kappa}_i\,G_i(s) \qquad (5)$$

where $G_i(s) = \int_0^s F_i(s')\,\mathrm{d}s'$ is the C^1-smooth piecewise quadratic function

$$G_i(s) = \begin{cases} 0 & \text{if } s \le s_{i-1} \\ \frac{(s - s_{i-1})^2}{2\,\ell_i} & \text{if } s_{i-1} \le s \le s_i \\ \frac{\ell_i}{2} + (s - s_i) - \frac{(s - s_i)^2}{2\,\ell_{i+1}} & \text{if } s_i \le s \le s_{i+1} \\ \frac{1}{2}\,(\ell_i + \ell_{i+1}) & \text{if } s \ge s_{i+1}. \end{cases} \qquad (6)$$

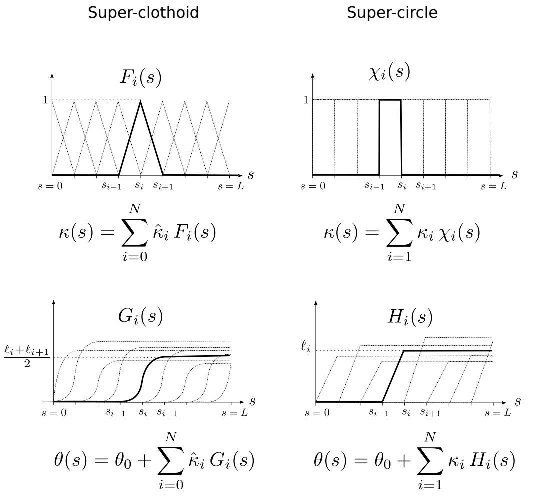


Figure 3: *Shape functions for the super-clothoid model (left) compared to those of the super-circle model (right).*

The curvature and angular shape functions $F_i(s)$ and $G_i(s)$ are represented in Figure 3 (left), and compared against the shape functions used in the super-circle model (right).

Position, velocity and acceleration For the sake of clarity, let us introduce the following 2D functions

$$\mathbf{\Phi}(s) = \int_0^s \mathbf{n}(s')\,\mathrm{d}s' \qquad (7)$$

$$\mathbf{\Phi}_i^G(s) = \int_0^s \mathbf{n}(s')\,G_i(s')\,\mathrm{d}s' \qquad (8)$$

$$\mathbf{\Psi}(s) = \int_0^s \mathbf{t}(s')\,\mathrm{d}s' \qquad (9)$$

$$\mathbf{\Psi}_i^G(s) = \int_0^s \mathbf{t}(s')\,G_i(s')\,\mathrm{d}s' \qquad (10)$$

$$\mathbf{\Psi}_{ij}^{GG}(s) = \int_0^s \mathbf{t}(s')\,G_i(s')\,G_j(s')\,\mathrm{d}s'. \qquad (11)$$

All these functions (nonlinearly) depend on θ_0 and $\boldsymbol{\kappa}$ only. Let us define $\mathbf{\Phi}^G(s)$ (resp. $\mathbf{\Psi}^G(s)$) the matrix collecting in a 2D row the $N+1$ column vectors $\mathbf{\Phi}_i^G(s)_{i=0..N}$ (resp. $\mathbf{\Psi}_i^G(s)_{i=0..N}$), and let $\mathbf{\Psi}^{GG}(s)$ be the third-order tensor with coordinate $\mathbf{\Psi}_{ijk}^{GG}(s)$ where k indicates the component index of the 2D vector $\mathbf{\Psi}_{ij}^{GG}(s)$. The global position $\mathbf{r}(s)$ then reads $\mathbf{r}(s) = \mathbf{r}_0 + \mathbf{\Psi}(s)$, and by applying time differentiation we get the global velocity and acceleration

$$\dot{\mathbf{r}}(s) = \dot{\mathbf{r}}_0 + \mathbf{\Phi}(s)\,\dot{\theta}_0 + \mathbf{\Phi}^G(s)\,\dot{\boldsymbol{\kappa}} \qquad (12)$$

$$\begin{aligned}\ddot{\mathbf{r}}(s) &= \ddot{\mathbf{r}}_0 + \ddot{\theta}_0\,\mathbf{\Phi}(s) + \mathbf{\Phi}^G(s)\,\ddot{\boldsymbol{\kappa}} \qquad (13)\\ &- \dot{\theta}_0{}^2\,\mathbf{\Psi}(s) - 2\,\dot{\theta}_0\,\mathbf{\Psi}^G(s)\,\dot{\boldsymbol{\kappa}} - \left(\mathbf{\Psi}^{GG}(s)\overline{\otimes}\dot{\boldsymbol{\kappa}}\right)\dot{\boldsymbol{\kappa}}.\end{aligned}$$

Note that $\ddot{\mathbf{r}}(s)$ linearly depends on $\ddot{\mathbf{r}}_0$, $\ddot{\theta}_0$ and $\ddot{\boldsymbol{\kappa}}$, and nonlinearly on the other terms $\dot{\theta}_0$, θ_0, $\dot{\boldsymbol{\kappa}}$ and $\boldsymbol{\kappa}$. Matrix $\mathbf{\Phi}^G(s)$ plays an important role as it represents the Jacobian matrix $\left(\frac{\partial \mathbf{r}}{\partial \hat{\kappa}_i}(s)\right)_{i\in\{0,\ldots,N\}}$ which linearly relates $\ddot{\boldsymbol{\kappa}}$ to $\ddot{\mathbf{r}}(s)$.

3. Dynamics of a Super-Clothoid

The dynamics of our new discrete rod model is given by the Lagrangian equations of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathscr{T}}{\partial \dot{\hat{\kappa}}_i} - \frac{\partial \mathscr{T}}{\partial \hat{\kappa}_i} + \frac{\partial \mathscr{U}}{\partial \hat{\kappa}_i} = 0 \qquad \forall i \in \{0,\ldots,N\} \qquad (14)$$

where $\mathscr{T}$ and $\mathscr{U}$ are respectively the kinetic and potential energies of our system, defined as [BAC*06]

$$\begin{aligned} \mathscr{T} &= \frac{\rho S}{2}\int_0^L \dot{\mathbf{r}}^2(s)\,\mathrm{d}s \\ \text{and} \qquad \mathscr{U} &= \mathscr{E}_g + \mathscr{E}_{el}, \end{aligned}$$

with $\mathscr{E}_g$ the weighting energy of the rod and $\mathscr{E}_{el}$ its internal elastic energy. Let g be the constant of gravity, ρS the lineic mass of the rod and EI its stiffness, assumed to remain constant along the centerline. The natural curvature of the rod is denoted $\kappa^0(s)$ and is supposed to vary linearly on each element, similarly as $\kappa(s)$. In the general continuous case, the potential energies $\mathscr{E}_g$ and $\mathscr{E}_{el}$ read [BAQ*05]

$$\begin{aligned} \mathscr{E}_g &= \rho S g \int_0^L (L-s)sin(\theta(s))\mathrm{d}s \\ \text{and} \qquad \mathscr{E}_{el} &= \frac{EI}{2}\int_0^L \left(\kappa(s) - \kappa^0(s)\right)^2 \mathrm{d}s. \end{aligned}$$

3.1. Discrete Equations of Motion

After analytically deriving the terms of Equation (14) in the piecewise linear case, we get the following discrete dynamic equations for a super-clothoid,

$$\rho S \int_0^L \ddot{\mathbf{r}}(s)^\top \mathbf{\Phi}_i^G(s)\,\mathrm{d}s + \frac{\partial \mathscr{E}_g}{\partial \hat{\kappa}_i} + \frac{\partial \mathscr{E}_{el}}{\partial \hat{\kappa}_i} = 0 \quad \forall i \in \{0,\ldots,N\} \qquad (15)$$

$$\begin{aligned} \text{where} \qquad \frac{\partial \mathscr{E}_g}{\partial \hat{\kappa}_i} &= \rho S g \int_0^L (L-s)\,cos\theta(s)\,G_i(s)\,\mathrm{d}s \\ \text{and} \qquad \frac{\partial \mathscr{E}_{el}}{\partial \hat{\kappa}_i} &= EI\sum_{j=0}^{N}\left(\hat{\kappa}_j - \hat{\kappa}_j^0\right)\int_0^L F_j(s)\,F_i(s)\,\mathrm{d}s. \end{aligned}$$

The linear dependence in $\ddot{\hat{\kappa}}_j$ explicitly appears in the acceleration term $\ddot{\mathbf{r}}$ of the left-side member of Equation (15). Furthermore, $\frac{\partial \mathscr{E}_{el}}{\partial \hat{\kappa}_i}$ linearly depends on the $\hat{\kappa}_j$. This term can thus be evaluated using an implicit time-stepping scheme. All other terms are nonlinear expressions of $\hat{\kappa}_j$ and $\dot{\hat{\kappa}}_j$ and will be computed explicitly. Finally, by decomposing the left-hand side term of Equation (15) using the expression (13) for $\ddot{\mathbf{r}}(s)$, we obtain

$$\mathbb{M}(\boldsymbol{\kappa})\,\ddot{\boldsymbol{\kappa}} + \mathbb{K}\left(\boldsymbol{\kappa} - \boldsymbol{\kappa}^0\right) = \mathbf{B}(\boldsymbol{\kappa}, \dot{\boldsymbol{\kappa}}). \qquad (16)$$

$\mathbb{M}$ is the dense, symmetric positive-definite mass matrix

$$\mathbb{M}_{i,j} = \rho S \int_0^L \left(\mathbf{\Phi}_i^G(s)\right)^\top \mathbf{\Phi}_j^G(s)\,\mathrm{d}s,$$

$\mathbb{K}$ is the tridiagonal, symmetric positive-definite stiffness matrix

$$\mathbb{K}_{i,j} = EI\int_0^L F_i(s)\,F_j(s)\,\mathrm{d}s$$

$$\begin{aligned} \text{with} \quad \mathbb{K}_{i,i} &= EI\int_0^L F_i^2(s)\,\mathrm{d}s = EI\left(\frac{\ell_i+\ell_{i+1}}{3}\right) \\ \text{and} \quad \mathbb{K}_{i-1,i} &= EI\int_0^L F_{i-1}(s)\,F_i(s)\,\mathrm{d}s = EI\,\frac{\ell_i}{6}, \end{aligned}$$

and $\mathbf{B} = \{B_i\}_{i\in\{0,\dots,N\}}$ collects all other (nonlinear) terms,

$$B_i \;=\; \tfrac{\partial \mathscr{E}_g}{\partial \hat{\kappa}_i} - \rho S\left(\int_0^L \ddot{\mathbf{r}}^*(s)^\top \mathbf{\Phi}_i^G(s)\,\mathrm{d}s\right)$$

where $\ddot{\mathbf{r}}^*(s) = \ddot{\mathbf{r}}(s, \ddot{\boldsymbol{\kappa}} = \mathbf{0})$ is the free acceleration of the rod.

3.2. Internal Damping

We use the same heuristics as in the super-helix model [BAC*06] for modeling internal damping, and simply add the supplementary implicit term $\mu\,\mathbb{K}\dot{\boldsymbol{\kappa}}$ to the left-hand side of Equation (16) where $\mu \geq 0$ is the internal damping coefficient. Intuitively, increasing the damping coefficient reduces the amounts of bending deformation of the rod. Note that using a small (but non-zero) value preserves the rod's "nervosity" while filtering out high-frequency oscillations, thus helping stabilize the simulation of the model.

3.3. External Forces

As in [BAC*06], external forces can be modelled in an unified way using a lineic density of distribution $\mathbf{p}(s)$. The contribution of this force density to the dynamic equations (16) is computed by projecting $\mathbf{p}(s)$ onto the $\mathbf{\Phi}_i$ vectors, leading to a generalized force vector $\mathbf{F} = \{F_i\}_i$ of size $N+1$ with

$$F_i = \int_0^L \mathbf{\Phi}_i^G(s)^\top\,\mathbf{p}(s)\,\mathrm{d}s. \tag{17}$$

To account for new external forces in the dynamic equations (16), one should only add the contribution F_i to the right-hand side term B_i given in Equation (17). Note that in our current dynamic equations, the derivative of the weighting energy $\mathscr{E}_g$ already accounts for the gravitational force, corresponding to a constant force density $\mathbf{p}_g = \rho\,S\,\mathbf{g}$. In the following, we derive the expression for a few other important external forces, such as air viscous friction and punctual forces, and explain how to model frictional contact.

Air viscous friction We model air viscous friction by the force density $\mathbf{p}_\nu(s) = -\nu\,\dot{\mathbf{r}}(s)$, where ν represents the air viscous coefficient. The corresponding generalized force contribution reads

$$F_i = -\nu\int_0^L \mathbf{\Phi}_i^G(s)^\top\,\dot{\mathbf{r}}(s)\,\mathrm{d}s. \tag{18}$$

Integration is performed numerically (see Section 3.5).

Punctual forces A punctual force applying to the location s_j is modeled using a Dirac force distribution $\mathbf{p}(s) = \mathbf{P}\,\delta(s-s_j)$, with the Dirac function defined such that $\int_E \delta(s-s_j)\,\mathrm{d}s = 1$ if $s_j \in E$, 0 else. The expression of the generalized force contribution is thus simply given by

$$F_i = \mathbf{\Phi}_i^G(s_j)^\top\,\mathbf{P}, \tag{19}$$

which only requires one evaluation of $\mathbf{\Phi}_i^G(s)$.

Frictional contact We follow the generic constraint-based approach proposed by Bertails et al. [BDCDA11]. A generalized contact force at location s_j is modeled as an implicit punctual force $\mathbf{F} = \mathbf{J}^\top\,\mathbf{r}$ with $\mathbf{r}$ the unknown (local) contact force (playing the role of a Lagrangian multiplier) and $\mathbf{J}$ the Jacobian matrix $\mathbf{\Phi}^G(s_j)$. Exact Coulomb friction is considered by adding a supplementary unknown, $\mathbf{u}$, which stands for the relative velocity between the two colliding objects, and relates to $\mathbf{r}$ through the simple equation $\boldsymbol{f}^{\mathrm{AC}}(\mathbf{u},\mathbf{r}) = \mathbf{0}$, where $\boldsymbol{f}^{\mathrm{AC}}$ is the (nonsmooth) Alart-Curnier function. The dynamic problem with frictional contact can then be formulated as a nonsmooth root-finding problem, and solved using a standard Newton's algorithm [BDCDA11].

3.4. Time Discretization

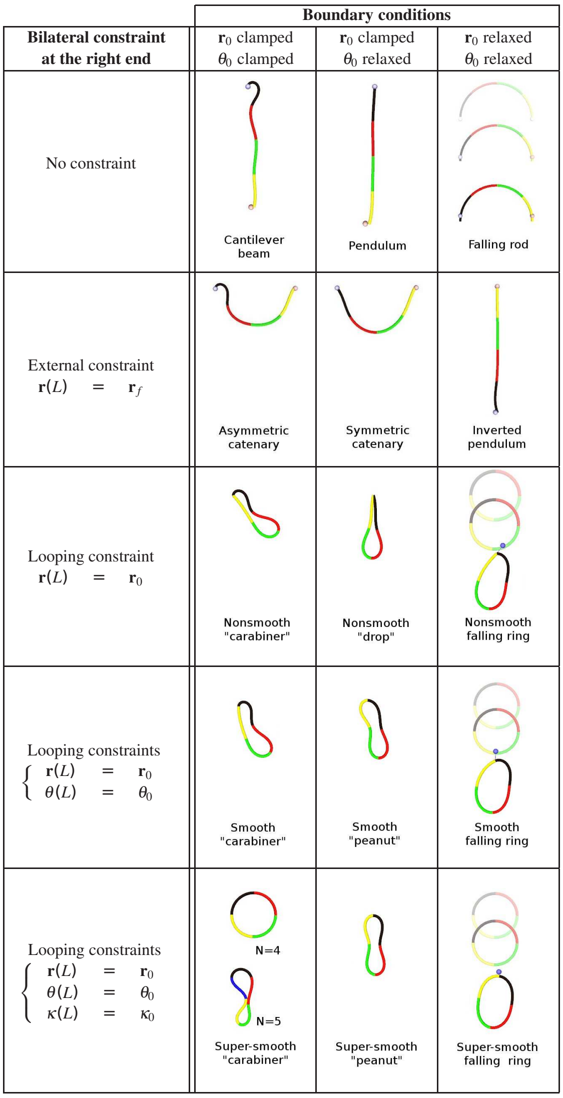
Implicit bending stress For the numerical solving of Equation (16) we used a semi-implicit Euler integration scheme. Linear terms such as the elasticity term and internal damping are computed implicitly, whereas all other (nonlinear) terms (gyroscopic terms and external forces - except contact forces) are evaluated explicitly. We observed that the implicit computation of the bending terms plays a major role in the stability of the simulation: compared to nodal models, for which the bending terms are nonlinear and have to be computed semi-explicitly [ST07, BWR*08], our model turns out to remain much more stable, even for high deformations and for a large time step (in practice, we have used a time step close to 10 ms for all our simulations).

Adaptive time step To improve the stability of the simulator in the case of large variations of the motion (during the interactive manipulation of the model by a user, for instance), we have implemented an adaptive time step that automatically refines when the numerical integration error – evaluated as the norm of the difference between next generalized velocity and current generalized velocity of the rod – exceeds a certain threshold. In practice, we have observed that this adaptive scheme works well in most situations as it both improves the accuracy of computations and greatly limits possible simulator's failures in the case of rough motions.

3.5. Implementation

We have implemented the super-clothoid model in C/C++, and used the Numerical Recipes library [PTVF07] for numerically evaluating integrals having no closed form, by way of using Romberg integration. Note that any scientific computing library allowing for numerical integration could be used in replacement. For the sake of robustness and efficiency, we carefully perform formal computations as far as possible with the help of the Maple software [Map10]. For instance, all kinematics terms are stored in a formal format and are only evaluated when needed. We typically resort to numerical approximation when assembling the linear system to be solved at each time step.

4. New Boundary Conditions and Constraints



	Boundary conditions		
Bilateral constraint at the right end	$\mathbf{r}_0$ clamped θ_0 clamped	$\mathbf{r}_0$ clamped θ_0 relaxed	$\mathbf{r}_0$ relaxed θ_0 relaxed
No constraint	Cantilever beam	Pendulum	Falling rod
External constraint $\mathbf{r}(L) = \mathbf{r}_f$	Asymmetric catenary	Symmetric catenary	Inverted pendulum
Looping constraint $\mathbf{r}(L) = \mathbf{r}_0$	Nonsmooth "carabiner"	Nonsmooth "drop"	Nonsmooth falling ring
Looping constraints $\begin{cases} \mathbf{r}(L) = \mathbf{r}_0 \\ \theta(L) = \theta_0 \end{cases}$	Smooth "carabiner"	Smooth "peanut"	Smooth falling ring

Looping constraints
$\begin{cases} \mathbf{r}(L) = \mathbf{r}_0 \\ \theta(L) = \theta_0 \\ \kappa(L) = \kappa_0 \end{cases}$
	N=4
	N=5
	Super-smooth "carabiner"
		Super-smooth "peanut"	Super-smooth falling ring

Figure 4: *Various configurations of the super-clothoid (N = 4) subject to new boundary conditions and/or constraints.*

Until now we have assumed that the rod was clamped at the left end, in terms of both its position ($\mathbf{r}_0$) and its orientation (θ_0). A one-side clamped rod indeed turns out to be a suitable model for a large range of common thin objects such as hair, grass, trees, etc., for which the position and direction of clamping are meaningful. For other systems such as a pendulum, a rope or a designed path, it may however be useful to either fix or release both ending orientations and/or positions. In this section we show that with a small set of changes made to our super-clothoid model, we manage to capture the complete set of all these different configurations, and thus give the user the choice of modelling a wide range of phenomena, represented in Figure 4.

4.1. Relaxing Clamping Constraints at the Left End

The principle consists in adding the left end position $\mathbf{r}_0 = \{r_{0,x}, r_{0,y}\}$ and/or the angle θ_0 as new degrees of freedom of the system. Let $\boldsymbol{\xi} = \{\xi\}_i$ be the vector of size $n' > N+1$ collecting the unknowns of the system. Its $N+1$ first coordinates are composed of the $\hat{\kappa}_i$ and its last coordinates composed of either $r_{0,x}, r_{0,y}$ ($n' = N+3$) or θ_0 ($n' = N+2$), or both ($n' = N+4$). The new discrete equations of motion read

$$\tilde{\mathbb{M}}(\boldsymbol{\xi})\,\ddot{\boldsymbol{\xi}} + \tilde{\mathbb{K}}\left(\boldsymbol{\xi} - \boldsymbol{\xi}^0\right) = \tilde{\mathbf{B}}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \tag{20}$$

where the matrices $\tilde{\mathbb{M}}$, $\tilde{\mathbb{K}}$ and vector $\tilde{\mathbf{B}}$ of size n' are computed from the original ones with only a few modifications, as explained in Appendix A. We denote $\boldsymbol{\xi}^0$ the vector of size n' collecting the $N+1$ scalars $\hat{\kappa}_i^0$, and then filled up with 0.

Figure 4, top row, depicts the whole set of rods configurations we are now able to model. In the case when the angle θ_0 is released, we add some internal dissipation $\tau\,\dot{\theta}_0$, $\tau \geq 0$ to its dynamics, similarly as in Section 3.2, in order to stabilize the system and avoid brusque changes in orientation.

4.2. Adding Bilateral Constraints at the Right End

External constraints We define three different types of configuration constraints of the form $C(\boldsymbol{\zeta}) = 0$, applied onto the location s_c of the centerline (one may choose $s_c = L$ to constrain the right end of the rod):

- a nonlinear position constraint of dimension 2, denoted $\mathbf{C}_{\mathbf{r}}(\boldsymbol{\zeta}) = \mathbf{r}(s_c) - \mathbf{r}_f$;
- a linear, scalar angular constraint, $C_\theta(\boldsymbol{\zeta}) = \theta(s_c) - \theta_f$;
- a linear, scalar curvature constraint, $C_\kappa(\boldsymbol{\zeta}) = \kappa(s_c) - \kappa_f$,

where $\mathbf{r}_f$, θ_f and κ_f are fixed (*e.g.*, imposed by the user).

We add n_c scalar bilateral constraints of any type above (the position constraint splits into two scalar constraints that are linearized) to the original system (20) by augmenting the vector of unknowns $\boldsymbol{\zeta}$ with a Lagrangian multiplier $\boldsymbol{\lambda}$ of size n_c, and formulate the bilateral constraints at the velocity level. This leads to the following constrained system

$$\begin{cases} \tilde{\mathbb{M}}(\boldsymbol{\xi})\,\ddot{\boldsymbol{\xi}} + \tilde{\mathbb{K}}\left(\boldsymbol{\xi} - \boldsymbol{\xi}^0\right) & = & \tilde{\mathbf{B}}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) + \mathbb{H}^\top \boldsymbol{\lambda} \\ \mathbb{H}\dot{\boldsymbol{\xi}} & = & \mathbf{D} \end{cases} \tag{21}$$

where the matrix $\mathbb{H}$ of size (n_c, n') collects the Jacobian matrices of the constraints, and the vector $\mathbf{D}$ of size n_c collects the constant terms of the constraints. For each type of constraint mentioned above, these terms respectively read (assuming the left clamping end constraints are relaxed)

$$\begin{aligned} \mathbb{H}_{\mathbf{r}} = \left(\frac{\partial \mathbf{r}}{\partial \xi_i}(s_c)\right)_{0 \leq i \leq n'} &= \left[\boldsymbol{\Phi}_0^G(s_c), \ldots, \boldsymbol{\Phi}_N^G(s_c), \mathbf{e}_x, \mathbf{e}_y, \boldsymbol{\Phi}(s_c)\right] \\ \mathbb{H}_\theta = \left(\frac{\theta}{\partial \xi_i}(s_c)\right)_{0 \leq i \leq n'} &= \left[G_0(s_c), \ldots, G_N(s_c), 0, 0, 1\right] \\ \mathbb{H}_\kappa = \left(\frac{\kappa}{\partial \xi_i}(s_c)\right)_{0 \leq i \leq n'} &= \left[F_0(s_c), \ldots, F_N(s_c), 0, 0, 0\right] \end{aligned}$$

and $\quad \mathbf{D}_{\mathbf{r}} = \dot{\mathbf{r}}_f - \dot{\mathbf{r}}(s_c, \dot{\boldsymbol{\xi}} = 0) \qquad \mathbf{D}_\theta = \dot{\theta}_f \qquad \mathbf{D}_\kappa = \dot{\kappa}_f$

where $\dot{\mathbf{r}}_f$, $\dot{\theta}_f$ and $\dot{\kappa}_f$ are evaluated with a discrete Euler scheme using previous and current time steps.

Self-constraints The three above constraints can of course be similarly derived for handling self-constraints, *e.g.*, for attaching the right end of the rod to its left end (forming a loop) or for glueing two different rods at a specific location. Let A and B be two rods (possibly representing the same rod) and s_c^A, s_c^B two locations where the constraint applies, on rods A and B respectively (if $A = B$ then we assume $s_c^A \neq s_c^B$). Compared to the external constraint, the gradient matrix is simply replaced with

$$\mathbb{H}_x = \mathbb{H}^A_{x^A} - \mathbb{H}^B_{x^B}$$

where $\mathbb{H}^A_{x^A}$ (resp. $\mathbb{H}^B_{x^B}$) is the gradient of the quantity x (representing $\mathbf{r}$, θ or κ) relative to rod A (resp. rod B), evaluated at s_c^A (resp. s_c^B). The vector $\mathbf{D}$ reads

$$\mathbf{D}_\mathbf{r} = \dot{\mathbf{r}}^B(s_c^B, \dot{\boldsymbol{\xi}}^B = 0) - \dot{\mathbf{r}}^A(s_c^A, \dot{\boldsymbol{\xi}}^A = 0) \qquad \mathbf{D}_\theta = 0 \qquad \mathbf{D}_\kappa = 0.$$

Figure 4, second row, depicts simple rod systems with various left end boundary conditions, subject to an external position constraint at the right end. In the case when the left clamped angle is relaxed, one gets a rod system with symmetric ends. When the left clamped position is additionally relaxed, one retrieves a pendulum system where the two ends are inverted compared to the one depicted in the first row, second column. The three last bottom rows of the figure illustrate the effect of applying self-constraints of various types (position, position and angle, position, angle and curvature) joining the right end to the left end of the rod. Note that adding an angular (resp. curvature) constraint to lower-order constraints enforces the G^1 (resp. G^2) smoothness of the resulting looping curve.

5. Inverse Statics: from Curve Design to Animation

Resolving the inverse statics problem with our super-clothoid model allows us to bridge the gap between the geometric design of smooth curves and their physically-based deformation.

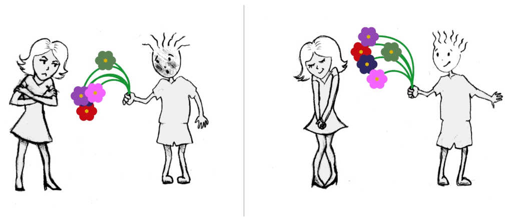


Figure 5: *Flowers sketched as piecewise clothoids by the user and converted into super-clothoids ($N = 2$). Without performing stable inverse statics (left) the flowers fall under gravity, whereas with stable inverse statics (right) their original shape remains at stable equilibrium.*

5.1. Sketching a Piecewise Clothoid

We use the accurate curve fitting algorithm by Baran et al. [BLP10] that allows the user to choose between either a piecewise clothoid approximation, or a piecewise circular arcs fit. This way, we are able to couple their fitting algorithm either to the super-circle model, or to our super-clothoid model, and perform thorough comparisons between the two approaches (see Section 6).

5.2. Inverse Statics

Similarly as in [DJBDT10], our goal is to find the parameters EI, ρS and $\boldsymbol{\kappa}^0$ such that the current configuration of the rod is a stable rest position, *i.e.*, its potential energy $\mathscr{U}$ is at a local minimum. The nice thing is that all the results regarding the computation of an equilibrium and its stabilization, derived in the super-circle case [DJBDT10], are directly applicable to the super-clothoid case.

Finding an equilibrium under gravity The equilibrium problem $\boldsymbol{\nabla}\mathscr{U} = \mathbf{0}$ always admits a solution for $\boldsymbol{\kappa}^0$,

$$\boldsymbol{\kappa}^0 = \boldsymbol{\kappa}^{fit} - (\mathbb{K}(EI))^{-1}\,\mathbf{B}(\rho S, \theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}, \dot{\boldsymbol{\kappa}} = \mathbf{0}), \quad (22)$$

where the parameters EI and ρS can freely span the entire positive real space. In Equation (22), we have explicitly mentioned the dependence of the matrix $\mathbb{K}$ (resp. of the vector $\mathbf{B}$) on the parameter EI (resp. ρS). The vector $\mathbf{B}$ is computed for a static configuration ($\dot{\boldsymbol{\kappa}} = \mathbf{0}$), using the fitting values $\theta_0^{fit}, \boldsymbol{\kappa}^{fit}, \boldsymbol{\ell}^{fit}$ output from the geometric fitting step. Finally, compared to [DJBDT10] where finding an equilibrium amounts to solving N independent scalar equations (diagonal system), solving Equation (22) here requires the inversion of a positive-definite tridiagonal system of size $N+1$, which can still be done very efficiently.

A sufficient condition for stability The Hessian of the potential energy reads $\nabla^2 E_p = EI\,\mathbb{T} + \rho S\,\mathbb{S}$ where $\mathbb{T} = \frac{1}{EI}\mathbb{K}$ is a symmetric, positive-definite tridiagonal matrix and $\mathbb{S}$ a real symmetric matrix, with

$$\mathbb{S}_{ij} = -g\int_0^L (L-s)\,sin\theta(s)\,G_i(s)\,G_j(s)\mathrm{d}s.$$

Note that both $\mathbb{T}$ and $\mathbb{S}$ are independent of EI, ρS, and $\boldsymbol{\kappa}^0$. With a similar reasoning as in [DJBDT10], we find a sufficient condition such that $\nabla^2 E_p$ is positive-definite,

$$\frac{EI}{\rho S} > -\frac{\lambda_N}{\gamma_N}, \quad (23)$$

where λ_N is the minimal eigenvalue of $\mathbb{S}$ and γ_N the minimal (strictly positive) eigenvalue of $\mathbb{T}$. Compared to [DJBDT10], evaluating condition (23) requires some extra computations for extracting the eigenvalues γ_i of the matrix $\mathbb{T}$. However, computing the eigenvalues of a real symmetric tridiagonal matrix can be efficiently achieved, for instance using a QL decomposition algorithm [PTVF07].

Finally, based on Equation (23), by either increasing EI or decreasing ρS, the stability of the rest shape under gravity can always be guaranteed, whatever the input curve is.

6. Results

In this section we provide some validation of the super-clothoid model against lower-order models, show some application results, and discuss the limitations of our approach. All our simulations are part of the accompanying video.

6.1. Validation

We have compared our super-clothoid model to two lower-order models, the articulated chain of rigid bodies [Fea83, Had06] and the super-circle model [BAC*06], in terms of geometric and motion accuracy, numerical robustness, and computational efficiency.

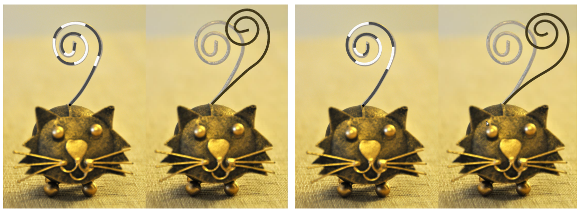


Figure 6: *Bringing dynamics to the tail of a photographed cat figurine, using (left) a super-circle and (right) our super-clothoid model. Compared to the super-circle model, the super-clothoid requires less elements for fitting the initial shape of the tail (9 instead of 11) and, more importantly, offers more visually pleasing (smoother) animation results.*

Geometric accuracy and smoothness Figure 6 shows the geometric fitting and the subsequent animation of the 2D tail of a cat figurine, by using either the super-circle primitive or our new super-clothoid model. Beyond its superior richness of representation – only 9 clothoid arcs are sufficient to accurately capture the shape of the tail, compared to 11 circular arcs, the piecewise clothoid primitive provides a more aesthetic (G^2-smooth) depiction of slender objects, and this level of smoothness is preserved all along animation. For this experiment the chain of articulated bodies was too unstable (see below) for representing a worthy model for comparison.

Motion accuracy Figure 7 illustrates, on a rod swinging motion under gravity, the gain in precision and smoothness brought by the super-clothoid model compared to the two other approaches. When using a very high resolution ($N = 50$ elements), the three models converge towards the same motion (close to the continuous case), which serves as a reference basis for our comparisons.

With only 4 elements, we note that the super-clothoid model produces a motion that is very close to the reference. In contrast, with the super-circle model, the circular arcs remain quite visible up to $N = 4$ elements, and highly curved parts, especially near the clamped end, are unsatisfactorily approximated even for a large number of elements ($N \leq 20$). This is even more critical for the articulated rigid body model, where at least 20 elements are necessary to reproduce the motion with a sufficient resolution. For further accuracy, these two models would require an adaptive, non-uniform distribution of elements along its centerline with short elements located at the highly deformed parts. We found that such an adaptive algorithm is not necessary for our super-clothoid model to be accurate all along the rod.

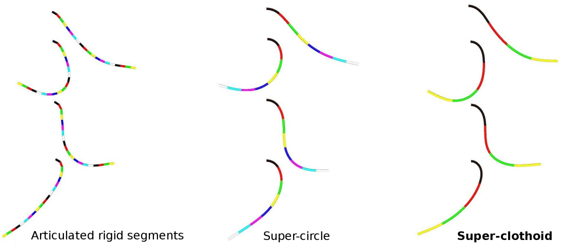


Figure 7: *To get an equivalent visual result for a swinging motion, 20 elements have been used for the chain of rigid segments (left), 8 for the super-circle (middle), and only 4 for the super-clothoid (right).*

Numerical robustness While the articulated chain of rigid bodies suffers from stability issues when simulating the motion of undamped rods, we have noted that the numerical robustness is equivalent for the two other models. With a fixed time step (dt = 11 ms), the super-circle model diverges for large and fast motions, similarly as for the super-clothoid model. We thus have also incorporated our adaptive time step scheme to the super-circle model so as to perform fair comparisons between the two. With the adaptive time step, we could only observe that a high, similar level of stability was achieved by both models.

Computational efficiency All our experiments were run on a single threaded application running on a laptop equipped with an Intel Core 2 Duo CPU at 2 GHz. The super-clothoid model turns out to be fully interactive up to 15 elements, which actually represents a wider range than the one we have been using for our applications in practice. As expected however, due to the recourse to accurate numerical integration, this model cannot, from a mere animation viewpoint, compete with the two other, analytical models, and especially with the extremely fast (up to 300x faster) super-circle model, for which all integrals computations have been worked out by hand and hard-coded. Nevertheless, in the context of unifying curve design and animation, the super-clothoid model offers the best deal, as explained in the discussion Section 6.3.

6.2. Examples of Applications

Bringing life to pictures If you look around you, you will see how many objects are made of planar, long and thin parts. These range from chair bars, curtain rods, decorative objects, to pattern designs on paper walls, cloth, and household linens. Thanks to our new dynamic primitive, it becomes straightforward to animate such 2D objects, thus bringing some dynamics to still images. Figure 6 provides an example of this picture animation process.

Secondary motion in 2D animation Thanks to the wide range of configurations offered by setting multiple boundary conditions and bilateral constraints to our model (see Section 4), we are able to animate various 2D dynamic objects. As an example, the video presents a ball composed of a smooth closed super-clothoid that falls under gravity before colliding with the ground. Due to Coulomb friction, this deforming "ball" may slide or roll on the ground.

Table 1: *Important features for our example demos.*

Example (N elements/rod)	Left end relaxed ?	Contact & constraints ?	Inverse statics ?
Cat tail (9)	No	No	Yes
Statuette leg (2)	No	No	Yes
Falling ball (4)	Yes	Yes	No
Flowers (2)	No	No	Yes

6.3. Limitations and Discussion

Unlike lower-order rod models (articulated rigid bodies and super-helix), the super-clothoid model requires spatial numerical integration for computing the discrete equations of motion. This is due to the non-closed form of the geometry of a clothoid, which relies on the Fresnel integrals. As mentioned above, accurate numerical approximation does not alter the stability of the system, but it does affect its computational cost. This supplementary cost is the price to pay for gaining one order of precision and smoothness in the geometry as well as in the mechanical behavior of the system. As illustrated above, this gain of smoothness is relevant in all our simulations. Moreover, a low number of elements (typically, N ranging between 5 and 10) proves sufficient for capturing complex deformations at a high resolution.

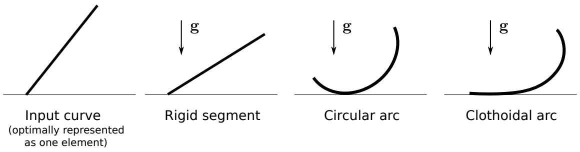


Figure 8: *An initially straight curve is optimally represented with one element (left), whatever the geometric primitive used. After conversion into the corresponding dynamic model, gravity is applied (right). With a rigid segment, the curve cannot deform and with a circular arc, motion looks unnaturally constrained. In contrast, with a clothoidal arc, the curve nicely bends and unfolds on the ground.*

One could still argue that by using lower-order models, it would be affordable to artificially refine input curves so as to get a sufficient resolution for animation without degrading performance too much. For that purpose, the circular arcs reconstruction method proposed in [DJBDT10] lets the user choose the resolution of the reconstruction, depending on what he/she desires as for the animation precision. Since animation comes at the end of the pipeline, such a process may require many tests and trials before the user finds a suitable resolution. Moreover, the authors mention algorithmic issues when trying to excessively refine the curve with very small arcs. In contrast, here we are concerned with a much closer unification between geometric curve design and animation. We claim that the resolution of the piecewise arcs primitive should only depend upon the *geometric precision*, should be *optimal*, and should be *automatically* computed from the sketched curve, as done in most approximation methods [MS08, BLP10]. In this context the super-clothoid model offers the best deal compared to low-order models. Figure 8 shows that even an extremely low resolution will yield fairly rich deformations. Moreover, as illustrated in Figures 1 and 6, a low resolution will still ensure the fairness property all along animation. In other words, the required input geometric resolution is *compatible* with the needs of animation. This is not the case for the two lower-order models for which there exists a huge gap between the required geometric resolution and the required animation resolution.

7. Conclusion

We have introduced the super-clothoid model, a new compact and robust physics-based primitive for simulating the dynamics of a piecewise clothoid. With only a few clothoidal elements, this model is able to interactively capture the elastic deformations of a wide range of 2D systems such as cantilever beams, pendulums, smooth closed curves, possibly subject to contact and friction. Combined with inverse statics, this approach bridges the gap between the geometric design of smooth curves and their subsequent animation, opening the way for new interactive and dynamic design interfaces. In the future, we would like to extend the class of created objects to complex combinations of curves and to the handling of 2D objects defined by an extensible contour. Designing a 3D super-clothoid model is also part of our motivation, given the numerous applications in 3D animation as well as the recent interest towards 3D piecewise clothoids in computer vision and geometric design [GXH01, BHHT10, HT10].

Acknowledgments

The author would like to thank Romain Casati and Alexandre Derouet-Jourdan for sharing their code on articulated rigid bodies and static fitting, respectively, and all the anonymous reviewers for their very helpful comments.

References

[BAC*06] BERTAILS F., AUDOLY B., CANI M.-P., QUERLEUX B., LEROY F., LÉVÊQUE J.-L.: Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph. 25* (2006), 1180–1187. 1, 2, 4, 5, 8

[BAQ*05] BERTAILS F., AUDOLY B., QUERLEUX B., LEROY F., LÉVÊQUE J.-L., CANI M.-P.: Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics short papers* (2005). 4

[BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete viscous threads. *ACM Trans. Graph.* (2010). 2

[BDCDA11] BERTAILS-DESCOUBES F., CADOUX F., DAVIET G., ACARY V.: A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Trans. Graph. 30* (2011), 6:1–6:14. 5

[Ber09] BERTAILS F.: Linear time super-helices. *Comp. Graph. Forum 28*, 2 (2009). 2

[BHHT10] BEN-HAIM D., HARARY G., TAL A.: Piecewise 3D Euler spirals. In *ACM Symp. Sol. and Phys. Modeling* (2010), SPM '10, pp. 201–206. 9

[BLP10] BARAN I., LEHTINEN J., POPOVIĆ J.: Sketching clothoid splines using shortest paths. *Comp. Graph. Forum* (2010). 1, 2, 7, 9

[BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete elastic rods. *ACM Trans. Graph. 27*, 3 (2008), 1–12. 2, 5

[DJBDT10] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Stable inverse dynamic curves. *ACM Trans. Graph. 29* (2010), 137:1–137:10. 2, 7, 9

[Fea83] FEATHERSTONE R.: The calculation of robot dynamics using articulated-body inertias. *Int. J. Robotics Research 2*, 1 (1983), 13–30. 8

[GXH01] GUIQING L., XIANMIN L., HUA L.: 3D discrete clothoid splines. In *Proc. Int. Conf. Comp. Graphics* (2001), CGI '01, IEEE Computer Society, pp. 321–. 9

[Had06] HADAP S.: Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symp. Comp. Animation (SCA'06)* (2006), pp. 91–100. 2, 8

[HT10] HARARY G., TAL A.: 3D Euler spirals for 3D curve completion. In *Symp. Comp. Geometry* (2010), SoCG '10, ACM, pp. 393–402. 2, 9

[KFP03] KIMIA B., FRANKEL I., POPESCU A.: Euler spiral for shape completion. *Int. J. Comput. Vision 54* (2003), 157–180. 2

[Lev08] LEVIEN R.: *The Euler spiral: a mathematical history*. Tech. Rep. UCB/EECS-2008-111, EECS Department, University of California, Berkeley, 2008. 2

[LMGC04] LENOIR J., MESEURE P., GRISONI L., CHAILLOU C.: A suture model for surgical simulation. *2nd Int. Symp. Medical Simulation (ISMS'04)* (2004). 2

[Map10] MAPLESOFT: Maple 14, 2010. 5

[MS92] MORETON H., SÉQUIN C.: Functional optimization for fair surface design. In *Comp. Graph. Proceedings* (1992), vol. 26, ACM, pp. 167–176. 2

[MS08] MCCRAE J., SINGH K.: Sketching piecewise clothoid curves. *Computers & Graphics 33*, 4 (2008), 452–461. 2, 9

[PTVF07] PRESS W., TEUKOLSKY S., VETTERLING W., FLANNERY B.: *Numerical Recipes: The Art of Scientific Computing (Third Edition)*. Cambridge University Press, 2007. 3, 5, 7

[Sap94] SAPIDIS N. S. (Ed.): *Designing fair curves and surfaces*. Society for Industrial and Applied Mathematics, Philadephia, PA, 1994. 2

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM Trans. Graph 27*, 3 (2008), 1–11. 2

[ST07] SPILLMANN J., TESCHNER M.: CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *ACM SIGGRAPH - EG Symp. Comp. Animation* (2007), pp. 63–72. 2, 5

[ST08] SPILLMANN J., TESCHNER M.: An adaptive contact model for the robust simulation of knots. *Comp. Graph. Forum 27*, 2 (2008). 2

[TGAB08] THEETTEN A., GRISONI L., ANDRIOT C., BARSKY B.: Geometrically exact splines. *J. Comp. Aided Design 40*, 1 (2008), 35–48. 2

Appendix A: Relaxed Clamping End

Once the clamping end constraint has been relaxed, the dynamics of the system is described by the new equations of motion

$$\tilde{\mathbb{M}}(\boldsymbol{\xi})\,\ddot{\boldsymbol{\xi}} + \tilde{\mathbb{K}}\,(\boldsymbol{\xi} - \boldsymbol{\xi}^0) = \tilde{\mathbf{B}}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$$

with the modified matrices $\tilde{\mathbb{M}}$, $\tilde{\mathbb{K}}$ and vector $\tilde{\mathbf{B}}$ of size n'.

$\tilde{\mathbb{M}}$ is the symmetric positive definite matrix of size n' with

- the previous square block related to the degrees of freedom $\hat{\kappa}_i$,

$$\tilde{\mathbb{M}}_{i,j} = \mathbb{M}_{ij} \quad \forall i, j \in \{0, \ldots, N\},$$

- the supplementary elements if $\mathbf{r}_0$ is a degree of freedom,

$$\begin{aligned}
\tilde{\mathbb{M}}_{i,N+1} &= \rho S \left(\mathbf{e}_x^\top \int_0^L \boldsymbol{\Phi}_i^G(s) \right) \mathrm{d}s \qquad \forall i \in \{0, \ldots, N\} \\
\tilde{\mathbb{M}}_{i,N+2} &= \rho S \left(\mathbf{e}_y^\top \int_0^L \boldsymbol{\Phi}_i^G(s)\, \mathrm{d}s \right) \quad \forall i \in \{0, \ldots, N\} \\
\tilde{\mathbb{M}}_{N+1,N+1} &= \tilde{\mathbb{M}}_{N+2,N+2} = \rho S L \\
\tilde{\mathbb{M}}_{N+1,N+2} &= 0,
\end{aligned}$$

- the supplementary elements if θ_0 is a degree of freedom,

$$\begin{aligned}
\tilde{\mathbb{M}}_{i,n'-1} &= \rho S \int_0^L \boldsymbol{\Phi}(s)^\top \boldsymbol{\Phi}_i^G(s)\, \mathrm{d}s \quad \forall i \in \{0, \ldots, N\} \\
\tilde{\mathbb{M}}_{n'-1,n'-1} &= \rho S \int_0^L \boldsymbol{\Phi}(s)^\top \boldsymbol{\Phi}(s)\, \mathrm{d}s,
\end{aligned}$$

- the supplementary elements if both $\mathbf{r}_0$ and θ_0 are free,

$$\begin{aligned}
\tilde{\mathbb{M}}_{N+1,N+3} &= \rho S \left(\mathbf{e}_x^\top \int_0^L \boldsymbol{\Phi}(s) \right) \\
\tilde{\mathbb{M}}_{N+2,N+3} &= \rho S \left(\mathbf{e}_y^\top \int_0^L \boldsymbol{\Phi}(s) \right).
\end{aligned}$$

$\tilde{\mathbb{K}}$ is the symmetric matrix of size n' with

- the previous square block related to the degrees of freedom $\hat{\kappa}_i$,

$$\tilde{\mathbb{K}}_{i,j} = \mathbb{K}_{ij} \quad \forall i, j \in \{0, \ldots, N\},$$

- the supplementary elements,

$$\tilde{\mathbb{K}}_{i,j} = 0 \quad \forall i, j \in \{N+1, \ldots, n'-1\}.$$

$\tilde{\mathbf{B}}$ is the vector of size n' with

- the previous vector block, modified as

$$\tilde{\mathbf{B}}_{i \in \{0,\ldots,N\}} = \mathbf{B}(\ddot{\mathbf{r}}_0 = 0) \quad \text{if } \mathbf{r}_0 \text{ is a degree of freedom}$$

$$\tilde{\mathbf{B}}_{i \in \{0,\ldots,N\}} = \mathbf{B}(\ddot{\theta}_0 = 0) \quad \text{if } \theta_0 \text{ is a degree of freedom}$$

- the supplementary elements if $\mathbf{r}_0$ is a degree of freedom,

$$\begin{aligned}
\tilde{\mathbf{B}}_{N+1} &= 0 \\
\tilde{\mathbf{B}}_{N+2} &= -\rho S g L,
\end{aligned}$$

- and the supplementary element if θ_0 is a degree of freedom,

$$\tilde{\mathbf{B}}_{n'-1} = -\rho S g \left(\mathbf{e}_y^\top \int_0^L \boldsymbol{\Phi}(s) \right).$$