# Interactive simulation of one-dimensional flexible parts

Mireille Grégoire*
DaimlerChrysler Research and Technology

Elmar Schömer†
Johannes Gutenberg Universität Mainz, Germany

## Abstract

Computer simulations play an ever growing role for the development of automotive products. Assembly simulation, as well as many other processes, are used systematically even before the first physical prototype of a vehicle is built in order to check whether particular components can be assembled easily or whether another part is in the way. Usually, this kind of simulation is limited to rigid bodies. However, a vehicle contains a multitude of flexible parts of various types: cables, hoses, carpets, seat surfaces, insulations, weatherstrips... Since most of the problems using these simulations concern one-dimensional components and since an intuitive tool for cable routing is still needed, we have chosen to concentrate on this category, which includes cables, hoses and wiring harnesses.

This paper presents an interactive, real-time, numerically stable and physically accurate simulation tool for one-dimensional components. The modeling of bending and torsion follows the Cosserat model and is implemented with a generalized spring-mass system with a mixed coordinate system which features usual space coordinates for the positions of the points and quaternions for the orientation of the segments joining them. This structure allows us to formulate the springs based on the coordinates that are most appropriate for each type of interaction and leads to a banded system that is then solved iteratively with an energy minimizing algorithm.

**CR Categories:** I.6.5 [Simulation and Modeling]: Model Development

**Keywords:** Cable simulation, Cosserat model, Modelling of torsion

## 1 Introduction

The automotive industry aims to reduce development costs and time while meeting the demand for quality and for an increasing model range. In order to meet this challenge, more and more work is done digitally. Styling reviews, Digital Mock Ups and assembly simulations are used at an early stage in the development process. Thus, potential problems can be detected and solved much earlier, long before the first physical prototype is built. Assembly simulation is one of the applications used in the construction design process: the virtual prototype is tested for feasibility and ease of assembly, which can be optimized. Like in other Virtual Reality applications, the physical behavior of the components needs to be simulated: a collision detection system and a contact simulation that impede the interpenetration of the objects and allow them to slide one on the other are required. Our work will be based on the Virtual Reality software *veo*, developed by DaimlerChrysler Research and Technologies, which already has a real-time collision detection and interactive contact simulation [Buck and Schömer 1998] as well as a real-time multibody dynamics [Sauer and Schömer 1998]. A realistic way of dealing with flexible parts -which are currently treated as rigid bodies by the simulation - is however not yet included. This implies that deformations that can occur in the physical world cannot be simulated, which limits the possibilities of the tool. A typical example of this would be an assembly simulation in which a cable should be pushed slightly aside to permit the mounting of another part. Studies from the business units show that most of the problems concerning flexible parts are encountered with cables, wire harnesses and hoses. These parts are also concerned with another particular application: the routing. Wire harnesses are on the one hand growing more and more complex as the use of electrical and electronic components in vehicles grows; on the other hand they often need to be modified to accommodate for changes in surrounding parts or for optimization. To meet the interests of the business units, we have chosen to first concentrate on simulating these parts since they cover the most urgent need. For these bodies, one of their dimension (the length) is much bigger than the other two and their centerline contains most of the information needed to represent them. This one-dimensional nature leads to simplifications in the simulation compared to other objects such as flexible surfaces or bodies.

### 1.1 Previous work

Several approaches are used for simulating flexible bodies, and one-dimensional ones in particular. The spectrum of solution ranges from a purely graphical representation of oscillations without any physics and with very low computational requirements [Barzel 1997] to a complete finite element simulation of cloth to predict its mechanical properties [Finckh et al. 2004]. Hergenröther [Hergenroether and Daehne 2000] models a cable as a chain of cylinder segments connected by ball joints. The chain has at first two segments and is iteratively refined, doubling the number of segments at each iteration, thus giving an inexact but fast dynamic simulation and a more exact, slower static one. Loock [Loock and Schömer 2001] implements a cable as a spring-mass system with torsion springs for the bending forces that are proportional to the bending angle.

There are few approaches that deal with the torsion. This is unfortunate, since, as shown for example in [Goss et al. 2005], it plays a crucial part in the deformation of a cable. One of the most interesting approaches taking the torsion into account is the one from Pai [Pai 2002]. He uses a Cosserat model as a base for the simulation of suture strands during laparoscopic surgery. These strands are objects visually well approximated by a curve but nevertheless present three-dimensional body properties like twisting. In the typical use case in surgery simulation, the position and direction of the strand are defined at one end (corresponding to the end fixed in human tissues) and the forces and moments are defined at the other end, corresponding to the needle haptic device. The ordinary differential equation resulting from the Cosserat model is integrated in two passes to calculate all the variables.

*e-mail: mireille.gregoire@daimlerchrysler.com
†e-mail:schoemer@informatik.uni-mainz.de

### 1.2 Specific contributions

Being aware of the importance of torsion for the deformations of a cable, it became one of our main interests. After several attempts, we have finally chosen a generalized spring-mass system for modeling the cables. This system uses a mixed coordinate system that contains at the same time ordinary position coordinates and quaternions representing the orientation of segments on the basis of which the torsion is easy to calculate. For enforcing constraints like the conservation of length, we introduce an integral force analog to the action of a proportional-integral controller.

## 2 The Cosserat model for rods

The Cosserat model for rod-like solids (with one dimension - the length - much greater than the other two cross-section dimensions) is a model from continuum mechanics. It models such a three-dimensional body as a one-dimensional one while taking into account the properties of the cross-section. A rod is represented by its centerline (a curve in the usual three-dimensional space) associated to frames (whose vectors are the so-called directors) which represent material orientation and deformation. Such a model is well suited for real-time applications since it has a smaller number of variables (compared to finite elements models for example) and can nevertheless take into account a great number of physical properties. Large deformations are neither a problem since all the properties of the system can be defined relatively to the local frames.

### 2.1 Description

We consider only unstretchable and unshearable bodies (which represent the vast majority of objects of interest in the context of automotive construction). We furthermore assume that the cross-section is homogeneous and undeformable, and that the mechanical properties are constant along the length of the cable. The general Cosserat model without the above mentioned restrictions is explained in [Antman 1995] or in [Rubin 2000]. A cable of length $L$ is parameterized by its arc length $s$. It is described by a function associating to each point of the centerline of a reference configuration (for example a state without tensions like a straight line without torsion) a vector $\mathbf{r}(s)$ describing the position of the point of the centerline and a directors frame, $(\mathbf{d_1}(s),\mathbf{d_2}(s),\mathbf{d_3}(s))$ representing material directions. Under the above mentioned restrictions the directors frame is a right-handed orthonormal basis and a member of the special orthogonal group $SO(3)$ (the group of rotations of $\mathbb{R}^3$).

$$\begin{array}{rcl} [0,L] & \rightarrow & \mathbb{R}^3 \times \mathrm{SO}(3) \\ s & \mapsto & (\mathbf{r},(\mathbf{d_1},\mathbf{d_2},\mathbf{d_3})) \end{array}$$

The basis of directors is adapted to the curve: the third director $\mathbf{d}_3$ points in the tangent direction of the curve. The vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ show the position of two material lines in the cross section of the rod. The evolution of the basis $(\mathbf{d}_k)_{1\leq k\leq 3}$ along the curve is represented by the Darboux vector $\omega$. Similar to the angular velocity vector (replacing the time derivative with a derivative along the arc length), this vector has the following property:

$$\frac{\mathrm{d}\,\mathbf{d}_k}{\mathrm{d}s} = \omega \times \mathbf{d}_k \text{ for } k = 1,2,3$$

where $\times$ represents the cross-product.

### 2.2 Forces and torques

At the point of arc length $s_0$, the rod has a tension $\mathbf{n}(s_0)$ and an inner torque $\mathbf{m}(s_0)$. The rod is also submitted to external distributed forces like the weight or the contact forces, with a linear density $f$. The static equilibrium leads to

$$\left\{ \begin{array}{rcl} \mathbf{0} & = & \frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s}(s_0) + \mathbf{f}(s_0) \\ \mathbf{0} & = & \frac{\mathrm{d}\mathbf{m}}{\mathrm{d}s}(s_0) + \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}(s_0) \times \mathbf{n}(s_0) \end{array} \right.$$

### 2.3 Material properties

We know from continuum mechanics that the torque at a point of a rod is

$$\mathbf{m}(s) = B_\tau \tau\, \mathbf{T} + B_\kappa \kappa\, \mathbf{B}$$

with $\kappa$ the curvature, $\tau$ the torsion, $\mathbf{T} = \mathbf{d_3}$ the tangent, $\mathbf{B}$ the binormal. The coefficients $B_\kappa$ and $B_\tau$ are defined in a similar way to the moments of inertia and depend on the material properties (Young's modulus $E$ and shear modulus $G$) and the geometry of the cross section. For a circular and homogeneous cross section with radius $R$, we have

- bending stiffness:

$$B_\kappa = EI = \iint_{crosssection} E\,x^2\,dA = E\frac{\pi R^4}{4}$$

- torsional stiffness:

$$B_\tau = GJ = \iint_{crosssection} G\,r^2\,dA = G\frac{\pi R^4}{2}$$

where $x$ and $r$ are the distances respectively to the bending axis (in the cross-section, passing by the centerline) and to the torsion axis (the tangent to the centerline). In the case of a non-homogeneous cross section, the stiffnesses can be calculated by considering $E$ and $G$ as a function of the positions. In particular, for a hollow body like a hose, the result can be expressed with the help of the inner radius $R_{inner}$: $B_\kappa = E\frac{\pi(R^4 - R_{inner}^4)}{4}$

### 2.4 Equations

The combination of these equations results in the following differential equations:

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}\,\mathbf{n}}{\mathrm{d}s} & = & -\mathbf{f} \\ \frac{\mathrm{d}\,\mathbf{m}}{\mathrm{d}s} & = & -\mathbf{d}_3 \times n \\ \frac{\mathrm{d}\,\mathbf{d}_k}{\mathrm{d}s} & = & u \times \mathbf{d}_k \\ \omega & = & \kappa_1\,\mathbf{d}_1 + \kappa_2\,\mathbf{d}_2 + \tau\,\mathbf{d}_3 \\ \mathbf{m} & = & B_\kappa(\kappa_1\,\mathbf{d}_1 + \kappa_2\,\mathbf{d}_2) + B_\tau\tau\,\mathbf{d}_3 \end{array} \right.$$

These equations have the remarkable property that the torsion is constant over the length of the rod.

$$\tau = \tau_0$$

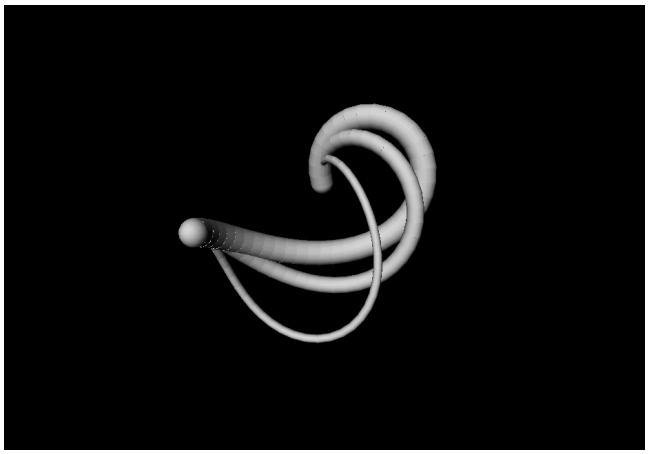

Figure 1: Influence of the radius : the different radii lead to different stiffnesses $B_\kappa$ and $B_\tau$. The influence on the form of the cable can be important. A change in material parameters would produce similar results.

### 2.5 Relation with the Frenet frame

Let us consider the Frenet frame (tangent $\mathbf{T}$, normal vector $\mathbf{N}$ and binormal vector $\mathbf{B}$) of the centerline. At any given point of the curve, the directors and the Frenet frames have at least a common vector, the tangent ($\mathbf{d}_3 = \mathbf{T}$). Therefore a rotation around $\mathbf{d}_3$ by an angle $\theta$ exists which transforms $(\mathbf{N},\mathbf{B},\mathbf{T})$ in $(\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)$. This angle $\theta$ shows the position of the material lines relatively to the Frenet frames: it is a "pure material torsion". The Cosserat torsion is then the Frenet torsion $\tau_f$ (geometrical torsion of the centerline) augmented with this material torsion.

$$\tau = \tau_f + \frac{\mathrm{d}\theta}{\mathrm{d}s}$$

The Darboux vector has the following expression:

$$\omega = \kappa_1 \mathbf{d}_1 + \kappa_2 \mathbf{d}_2 + \tau \mathbf{d}_3$$

where $\kappa_1$ and $\kappa_2$ are the components of the curvature on $\mathbf{d}_1$ and $\mathbf{d}_2$:

$$\kappa\mathbf{B} = \kappa_1 \mathbf{d}_1 + \kappa_2 \mathbf{d}_2$$

## 3 First attempts

### 3.1 Shooting method for ODE

The Cosserat model gives us a set of ordinary differential equations where the tensions and inner moments are coupled to the positions and orientations. In order to solve this, we need to know either the forces, moments, position and orientation at one end of the cable, or a combination of them distributed over the two ends. Since it is much easier for the user in the absence of force feedback to specify the position and orientation at one point, for example with a spacemouse, we wanted to have boundary conditions of geometrical type (position and orientation) at both ends of the cable. We therefore have to solve the following problem: knowing the positions and the directors frames at the two endpoints, we are looking for the deformation of the cable under the condition that the length of the cable should remain constant. This is a so-called two-points-boundary-value problem where the boundary conditions are spread on the two endpoints of the integration interval.

The solution of Pai [Pai 2002] is implemented for geometrical boundary conditions at one end of the strand and for dynamical boundary conditions at the other end, corresponding to the use of a haptic device. This particular configuration allows to solve the problem with two passes: the first one to calculate the forces and torques, and the second one in the opposite direction for the geometrical variables.

Unfortunately, an attempt to adapt this method to the desired type of boundary conditions was not successful. The usual method for solving such problems are so-called "shooting methods": there are two known and incomplete sets of boundary conditions, one at $s = 0$ and the other one at $s = L$. The method consists in completing the set at $s = 0$, integrating the differential equation with these initial conditions, and considering the obtained final conditions as a function of the initial ones, looking iteratively (for example with a Newton algorithm) for the appropriate initial conditions. Unfortunately, the initial conditions are difficult to determine, and the search does not always converge and sometimes aborts, for example when the curvature radius at a point is of the same order of magnitude as the integration step length. Furthermore, such an approach makes it difficult to integrate external forces such as contact forces.

### 3.2 Simple spring-mass-model

In order to avoid these difficulties, we experimented with a different approach using a spring-mass system. The rod is modeled as a sequence of mass points (lying on the centerline of the cable) which are connected with different kinds of springs: linear springs for the length conservation and torsion springs for the bending. The knowledge of the centerline leaves one degree of freedom unspecified, namely the material rotation around the centerline.

Since the Cosserat theory also considers the material direction, we need a new variable at each point to represent it, for example $\theta$, as the angle between the Frenet and the directors frames. The global torsion is the sum of both the Frenet torsion - calculated from the coordinates of the mass points - and the pure material torsion - calculated as the derivative of the fourth coordinate $\theta$-. The energy is then $E = \frac{1}{2}(B_\kappa \kappa^2 + B_\tau \tau^2)$ and the forces are calculated as the negative of the gradient of the energy.

We studied two ways of computing the Frenet torsion: one using the binormal vector and one using a function of the derivatives of the coordinates. For the first one, since $\mathbf{B}$ is a unit vector, the change of $\mathbf{B}$ is a rotation and the torsion is - in the discretized system - proportional to the angle between the two binormal vectors at the points $i$ and $i+1$:

$$\tau_f = \frac{\angle(\mathbf{B}_i, \mathbf{B}_{i+1})}{|s_{i+1} - s_i|}$$

where $s_i$ is the arc length at point $i$. Since the plane defined by the three points $i-1$, $i$, $i+1$ contains the tangent $\mathbf{T}$ and the normal $\mathbf{N}$, the binormal at point $i$ is orthogonal to both $\mathbf{u_{i-1}}$ and $\mathbf{u_i}$ and is calculated as:

$$\mathbf{B}_i = \frac{\mathbf{u_{i-1}} \times \mathbf{u_i}}{\|\mathbf{u_{i-1}} \times \mathbf{u_i}\|}$$

with $\mathbf{u_i}$ the unit vector between $\mathbf{r}(s_i)$ and $\mathbf{r}(s_{i+1})$:

$$\mathbf{u_i} = \frac{\mathbf{r}(s_{i+1}) - \mathbf{r}(s_i)}{\|\mathbf{r}(s_{i+1}) - \mathbf{r}(s_i)\|}$$

This scheme is - in particular in the case of a small curvature - extremely sensitive to noise in the position of the mass points. When for example $\mathbf{u_{i-1}}$ and $\mathbf{u_i}$ are fixed, and the angle between $\mathbf{u_i}$ and $\mathbf{u_{i+1}}$ is small, a very small change in $\mathbf{u_{i+1}}$ could mean a huge difference for $\tau_f$, just like at the Earth poles a small change in position can mean a large change in longitude. Accordingly, the forces in case of a small curvature are huge, which leads to numerical instability and impedes the convergence.

Inflexion points represent another problem: the binormals are not defined at such points and undergo a discontinuity (opposite directions). In a configuration with (almost) inflexion points, the direction of the binormal can change from one iteration to another, which is naturally impractical. An easy solution is to precalculate the binormal before each iteration and, when necessary, to multiply it by $-1$ in order to insure $\mathbf{B}_i^T \mathbf{B}_{i+1} \geq 0$, and to keep these places in memory to take into account the possible sign changes for the subsequent calculations.

Curvature and torsion can also be calculated as functions of the derivatives of $\mathbf{r}(s)$. After discretization and simplification, the expressions for the curvature and the torsion become

$$\begin{aligned} \kappa &= \frac{1}{2L} \left\| \mathbf{u_i} \times (\mathbf{u_{i+1}} - \mathbf{u_{i-1}}) \right\| \\ \tau_f &= \frac{1}{L^3 \kappa^2} \, \mathbf{u_i}^T (\mathbf{u_{i+1}} \times \mathbf{u_{i-1}}) \end{aligned}$$

with $L$ the length of a segment. This scheme is more stable and less noise-sensitive as the previous one, but it is still insufficient. The torsion remains in both cases the problem. Consequently, another way of calculating the torsion is needed.

# 4 Implementation

## 4.1 Mixed coordinates

We are looking for a coordinate system in which the torsion is easily expressed and not too noise-sensitive. The torsion depends on the difference of orientation along the tangent to the centerline of two segments. This orientation can be described as a rotation of $SO(3)$ from a reference orientation. From the several representations of rotations of $SO(3)$, we have chosen unit quaternions. (Quaternions are 4-tuples that can be seen as a generalization of complex numbers. They will be described in more detail in section 5.) Many properties speak in favor of unit quaternions: they only have 4 coordinates and one constraint (they must have a unit length), which is an advantage compared to rotation matrices (with 9 components and 6 constraints). Furthermore, the rotation of a vector is easily expressed and the composition of two rotations can also be easily calculated as the product of the two corresponding quaternions. This group structure is an advantage for the simulation. Quaternions also lack singular points and gimbal lock, contrarily to Euler or Cardan angles. So we used a system with seven coordinates for each point: the three usual space coordinates and a quaternion which represents the orientation of a segment between two points.

## 4.2 Notation

The following notations will be used in the rest of the paper:

$n$ is the number of discretization points of the cable. The cable begins and ends with a discretization point and the centerline is linearly interpolated between two points. $\mathbf{X} \in \mathbb{R}^{7n-4}$ is the vector of all coordinates of the cable and therefore of all scalar unknowns. $X_i \in \mathbb{R}$ is the $i$-th component of $\mathbf{X}$. $\mathbf{x_i} \in \mathbb{R}^3 = (X_{7i+1}, X_{7i+2}, X_{7i+3})$ is the position of the point $i$. $\mathbf{q}_i \in \mathbb{R}^4 = (X_{7i+4}, X_{7i+5}, X_{7i+6}, X_{7i+7})$ is a quaternion representing the orientation of the segment between the points $i$ and $i+1$.

For representing the orientations, we can arbitrarily choose an orientation reference - in our case a right-handed orthonormal basis whose third vector is in the direction $\mathbf{ref} = (0,1,0)$. This corresponds to the default orientation of the axis of cylinders in OpenInventor/OpenGL, which will simplify the graphical representation afterwards.

Additionally, $\mathbf{F} \in \mathbb{R}^{7n-4}$ is the vector of all forces; $F_i \in \mathbb{R}$ is the i-th component of $\mathbf{F}$. $L_{ref}$ is the reference length for each segment. It is equal to the total length of the cable divided by the number of segments $(n-1)$. $L_i = \sqrt{(\mathbf{x_{i+1}} - \mathbf{x_i})^2}$ is the actual length of segment $i$.

# 5 Forces

The interactions taken into account are bending and torsion, weight, stretch forces (unextensibility), normalization of quaternions and coherence between the quaternions and the space coordinates. Handles (fixed points) can be defined optionally: each one defines the additional constraint that the cable should pass by a particular point with a particular orientation. By default, the two endpoints are considered to be fixed in position and orientation: their coordinates are excluded from the solver range.

The use of space coordinates and quaternions allows to apply the forces to the most appropriate kind of coordinates:

- On the space coordinates:
  - conservation of length
  - handles (position)
  - weight
- On the quaternions:
  - bending and torsion forces
  - normalization of the quaternions
  - handles (orientation)
- On both the quaternions and the position:
  - coherence between position and quaternions

The main disadvantage of this method is the increased number of variables. But since every type of interaction is relatively easy to calculate, while at the same time the system matrix is strongly banded, the system as a whole is easy to solve.

## 5.1 Derivation of the forces

For each kind of interaction, we first define an energy function and then derive the forces from this function, following the use in mechanics:

$$F_i = -\frac{\partial E}{\partial X_i}$$

We also define the symmetric Hessian matrix $\mathbf{H} \in \mathbb{R}^{7n-4 \times 7n-4}$, such that

$$H_{i,j} = -\frac{\partial^2 E}{\partial X_i \partial X_j} = \frac{\partial F_i}{\partial X_j}$$

Since each type of interaction can be decomposed as a sum of interactions between either two points, two quaternions or two points and a quaternion, we calculate the energy, the forces and the Hessian as a sum over element groups. In order to enhance the calculation speed, it is important to calculate only once the partial terms that appear several times in the three functions.

For a global position $\mathbf{X}$, we can formulate the energy as the sum of the energies over all points and segments and over all interactions.

$$\begin{aligned} E &= \sum_{Interactions} \sum_{i} E_{Interaction,i} \\ &= \sum_{i=1}^{n-1} E_{Length,i} + \sum_{i=1}^{n-1} E_{QuatNorm,i} \\ &+ \sum_{i=1}^{n} E_{Weight,i} + \sum_{i=1}^{n-1} E_{Coh,i} \\ &+ \sum_{i=1}^{n-2} E_{Bending,i} + \sum_{i=1}^{n-2} E_{Torsion,i} \end{aligned}$$

The forces and the Hessian can be calculated in a similar way. In the following, $k_{Interaction}$ is the constant relative to the interaction *Interaction* that will be used for defining the relative weights of the different interactions.

## 5.2 Stretch forces and conservation of the length

We first use strong linear springs for the length conservation. The energy is defined classically as

$$E_{Length,\,P,\,i} = \frac{1}{2} k_{Length} (L_i - L_{ref})^2$$

where $k_{Length}$ is the constant of the spring. If we consider only this spring, the minimum of the energy is clearly at $L_i = L_{ref}$. But the points $i$ and $i+1$ are also submitted to other interactions that result in a disturbing force that stretches (or compresses) the spring. Since the spring can only exert a force when it is not at equilibrium, it alone cannot enforce exactly the constraint of a constant length: we use an additional force to achieve this. If we look at control theory, a spring alone is a proportional controller: the actuating variable (the force) is proportional (with the proportionality constant $k_{Length}$) to the difference between the desired value ($L_{ref}$) and the actual one ($L_i$). The forces for segment $i$ between point $i$ and $i+1$ due to the stretch forces are

$$\mathbf{f}_{P,\,i} = -\mathbf{f}_{P,i+1} = k_{Length}(L_i - L_{ref})\mathbf{u_i}$$

We want to use the equivalent of a PI-(proportional-integral) controller, which has the property of being able to enforce a constraint exactly so that the steady-state error is null. The PI-controller combines a proportional part (the spring) with an integral one. Its actuating variable is proportional to the integral over the time of the difference between the desired and actual value. When the error is null, the proportional force is also null, but the integral one counteracts the perturbations. The proportional force is nonetheless essential to the stability of the system. As a consequence of the presence of the integral force, the proportional constant can be reduced (in comparison with a spring-only system), which makes the system less stiff as a whole. To implement this, we have added constant forces $f_{I,\,i}\,\mathbf{u_i}$ on point $i$ and $-f_{I,\,i}\mathbf{u_i}$ on point $i+1$. This force must have the same direction as the proportional one. The total force on point $i$ due to the stretch of segment $i$ is then

$$\mathbf{f}_i = (f_{I,\,i} + k_{Length}(L_i - L_{ref}))\mathbf{u_i}$$

This corresponds to an energy of

$$E_{Length,\,i} = \frac{1}{2} k_{Length}(L_i - L_{ref})^2 + f_{I,\,i}(L_i - L_{ref})$$

We have now $n-1$ additional parameters $f_{I,\,i}$. For solving the system, we use an iterative approach: we first solve the system holding the $f_{I,\,i}$ constant, then adjust their values in function of the result and solve anew the system until an equilibrium is reached (The constraints are enforced within an $\varepsilon$ tolerance that should be chosen slightly bigger than the numerical precision of the system solver).

For updating the integral forces, we adjust each one individually: the new force is the old one augmented with the difference of lengths multiplied by a constant

$$f_{I,\,i,\,new} = f_{I,\,i,\,old} + k_{I,\,Length}(L_i - L)$$

This constant $k_{I,\,Length}$should not have a value too big compared to $k_{Length}$ for ensuring stability.

## 5.3 Weight

We consider the cable as a chain of mass points with equal masses $m$. The total mass of the cable is $n\,m$. The energy is $E_{Weight,i} = -m\mathbf{g}^T\mathbf{x_i}$ where $\mathbf{g} \in \mathbb{R}^3$ is the acceleration of gravity.

## 5.4 Coherence between quaternions and positions

It is important to ensure that the bending and torsion forces are correctly coupled to the positions. The unit vector $\mathbf{u}_i = \frac{\mathbf{x_{i+1}} - \mathbf{x_i}}{\|\mathbf{x_{i+1}} - \mathbf{x_i}\|} \in \mathbb{R}^3$ in the direction between the two points $i$ and $i+1$ should be equal the reference direction **ref** rotated by the quaternion $\mathfrak{q}_\mathbf{i}$. Let us consider a rotation of an angle $\theta$ around an axis $\mathbf{v} \in \mathbb{R}^3$ (with $\|\mathbf{v}\| = 1$). The corresponding quaternion is then

$$\mathfrak{q} = (q_0, q_1, q_2, q_3) = (q_0, \mathbf{q}) = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v})$$

The image $\mathbf{b} \in \mathbb{R}^3$ of a vector $\mathbf{a} \in \mathbb{R}^3$ by a rotation represented by a quaternion $\mathfrak{q}$ can be calculated by: $(0, \mathbf{b}) = \mathfrak{q} \cdot (0, \mathbf{a}) \cdot \overline{\mathfrak{q}}$ where $\overline{\mathfrak{q}} = (q_0, -\mathbf{q})$ is the conjugated of $\mathfrak{q}$ and where the quaternion product

$$\mathfrak{p} \cdot \mathfrak{q} = (p_0 q_0 - \mathbf{p}^T\mathbf{q}, p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q})$$

is used. For a constant $\mathbf{ref} = (0,1,0)$, the image can be directly calculated as

$$Rot(\mathbf{ref}) = (2(q_1q_2 - q_0q_3), q_0^2 - q_1^2 + q_2^2 - q_3^2, 2(q_2q_3 - q_0q_1)).$$

The energy is defined as

$$E_{Coh,i} = k_{Coh}\|(0, \mathbf{u_i}) - \mathfrak{q}_\mathbf{i} \cdot (0, \mathbf{ref}) \cdot \overline{\mathfrak{q}_\mathbf{i}}\|^2$$

In the same manner as for the length conservation, we introduce an integral force for each of the three components of the coherence between quaternions and positions.

## 5.5 Quaternion norm

Quaternions represent a pure rotation only when they have a unit norm, otherwise a scaling effect is introduced. The energy is simply defined as

$$E_{QuatNorm,i} = k_{QuatNorm}(\|\mathfrak{q}_\mathbf{i}\| - 1)^2$$

for $i \in 1...n-1$ with $\|\mathfrak{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$. This term is easy to enforce and does not pose any difficulty.

## 5.6 Bending and torsion

The bending and torsion forces are calculated jointly. They are determined by the two consecutive quaternions $\mathbf{q_i}$ and $\mathbf{q_{i+1}}$. (Observe that the previous formulations needed three segments and that the Frenet torsion is not needed anymore.) The relative rotation from segment $i$ to segment $i+1$ is represented by

$$\mathbf{q_{i \to i+1}} = \mathbf{q_{i+1}} \overline{\mathbf{q_i}}.$$

(The inverse of a quaternion $\mathbf{q}$ is $\frac{\overline{\mathbf{q}}}{\|\mathbf{q}\|}$ and for unitary quaternions simply $\overline{\mathbf{q}}$.). This rotation of an angle $\theta$ around an axis $\mathbf{v}$ is nothing else but the integral of the Darboux vector over the length of a segment:

$$\theta \mathbf{v} = \int_{s_i}^{s_{i+1}} \omega(s)\, \mathrm{d}s.$$

We also have

$$\mathbf{q_{i \to i+1}} = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v}).$$

Since $\frac{\theta}{2}$ can be expected to be small, it follows the approximation

$$\omega = \frac{2}{L}\mathbf{q_{i \to i+1}}.$$

On the other hand, the general properties of the Darboux vector give

$$\omega = \kappa \mathbf{B} + \tau \mathbf{T}.$$

The curvature $\kappa$ and the torsion $\tau$ can then be calculated. The tangent vector $\mathbf{T}$ is defined (symmetrically in $\mathbf{u_i}$ and $\mathbf{u_{i-1}}$) as

$$\mathbf{T}_i = \frac{\mathbf{u_{i-1}} + \mathbf{u_i}}{\|\mathbf{u_{i-1}} + \mathbf{u_i}\|}.$$

A decomposition of $\omega$ on the basis formed by $\mathbf{T}$ and $\mathbf{B}$ (which is defined as the direction of the residual component $\omega - \tau\mathbf{T} = \kappa\mathbf{B}$) gives us:

$$\tau = \omega \cdot \mathbf{T}$$

and

$$\kappa = \|\omega - \tau \mathbf{T}\|.$$

The energy is

$$E_{BendingTorsion,i} = E_{Bending,i} + E_{torsion,i} = \frac{1}{2} B_\kappa \kappa^2 + \frac{1}{2} B_\tau \tau^2$$

A possibility would be to use only the quaternions as coordinates and calculate iteratively the positions. The disadvantage is that the system matrix would be a full matrix (the position of point $i$ depends on all the quaternions between 1 and $i-1$), which makes the calculation much slower. External forces like contact forces are also difficult to implement.

## 5.7 Handles

Each handle (fixed point) is attached either to a point or to a segment. The two possibilities are offered for an easy interface with the graphical representation: the user can attach it either to a sphere or a cylinder. The handle can fix either only the position or both the position and orientation at a point. In the case of a sphere, the orientation is taken as the (normed) mean value of the two quaternions surrounding it; in the case of a cylinder, the position is the mean value of the positions of the two surrounding points. The energy, if

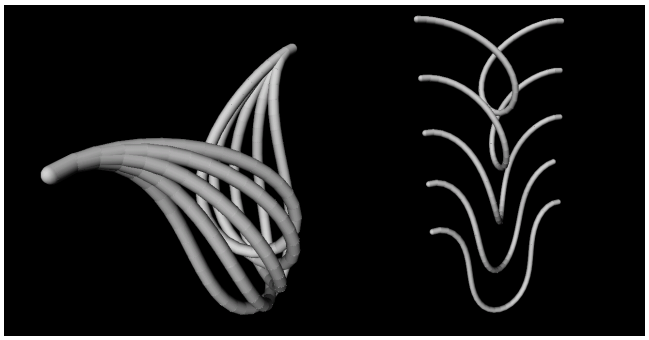

Figure 2: Influence of the torsion : the same cable is submitted to different values of the torsion by rotating one of its extremities

a handle is associated with the point $\mathbf{x_i}$ and with the quaternion $\mathbf{q_i}$, is

$$E_{HandleSphere,i} = k_{Handle}((\mathbf{x_i} - \mathbf{x_{Handle}})^2 + (\frac{\mathbf{q_{i-1}} + \mathbf{q_i}}{\|\mathbf{q_{i-1}} + \mathbf{q_i}\|} - \mathbf{q_{Handle}})^2)$$

or

$$E_{HandleCylinder,i} = k_{Handle}(\frac{\mathbf{x_i} - \mathbf{x_{i+1}}}{2} - \mathbf{x_{Handle}})^2 + (\mathbf{q_i} - \mathbf{q_{Handle}})^2)$$

## 5.8 Contact forces

When a collision is detected (the mass point $i$ has penetrated inside of another body), we apply a spring and an integral force at this point to keep it on the surface. Let $\mathbf{y} \in \mathbb{R}^3$ be the orthogonal projection of $\mathbf{x_i}$ on the surface and $\mathbf{n} \in \mathbb{R}^3$ the normal at $\mathbf{y}$, oriented towards the outside and normed. The penetration distance is then $d = (\mathbf{y} - \mathbf{x_i})^T \mathbf{n}$ and the force is

$$\mathbf{f_{Collision,\,i}} = k_{Collision}\, d\mathbf{n} + f_{I,\,Collision,\,i}\mathbf{n}$$

and the energy

$$E_{Collision,\,i} = \frac{1}{2} k_{Collision}\, d^2 + f_{I,\,Collision,\,i}\, d$$

When the penetration distance of a point is 0 (within the tolerance), if its integral force $f_{I,\,Collision,\,i}$ is negative, it means that the point needs to be attracted towards the object in order to be on the surface: in this case,it is not a collision point anymore and the spring and the integral forces are removed.

$f_{I,Collision,i}$ can be updated with two different methods: an individual or a global one. The first method is similar to the one we used for the length:

$$f_{I,i,new} = f_{I,i,old} + k_{I,Collision} d$$

The second method uses the derivative of the forces as a predicator for the behavior of the system upon a small change of the values of the forces. The Taylor series for the forces is $\mathbf{F}(\mathbf{X}+\mathbf{h}) = \mathbf{F}(\mathbf{X}) + \mathbf{H}\mathbf{h}$ with $\mathbf{h} \in \mathbb{R}^{7n-4}$ the difference between the old and the new position. Since the old position is an equilibrium, $\mathbf{F}(\mathbf{X}) = \mathbf{0}$. In order to fulfill the non-penetration constraint, we need that

$$\mathbf{h}^T \tilde{\mathbf{n}} = d$$

with $\tilde{\mathbf{n}} \in \mathbb{R}^{7n-4}$ the "direction" of the constraint: $\tilde{\mathbf{n}}_k = 0$ except for $\tilde{\mathbf{n}}_{7i+1}$, $\tilde{\mathbf{n}}_{7i+2}$, $\tilde{\mathbf{n}}_{7i+3}$ that take the values of $\mathbf{n}$. Additionally, the new force is $\mathbf{F}(\mathbf{X}+\mathbf{h}) = \Delta f_I \tilde{\mathbf{n}}$ (the variation of the integral force

$\Delta f_I = f_{I,mew} - f_{I,old}$ is not included in the Hessian). Combining the two equations leads to

$$\Delta f_I \tilde{\mathbf{n}} = \mathbf{H}\mathbf{h}$$

and further to

$$\Delta f_I = \frac{d}{\tilde{\mathbf{n}}^T \mathbf{H}^{-1} \tilde{\mathbf{n}}}$$

In the case of several contact points, the new force is $\mathbf{F}(\mathbf{X}+\mathbf{h}) = \sum_{i \in Contact points} \Delta f_{I,i} \tilde{\mathbf{n}}_\mathbf{i}$ and the condition to fulfill $\tilde{\mathbf{n}}_\mathbf{i}^T \mathbf{h} = d_i$. The combination of these relations results in a matrix equation:

$$\mathbf{M}(\Delta f_{I,1}, ..., \Delta f_{I,c})^T = (d_1, ..., d_c)^T$$

with $c$ the number of contact points and $\mathbf{M} \in \mathbb{R}^{c \times c}$ a matrix such that $\mathbf{M}_{i,j} = \tilde{\mathbf{n}}_\mathbf{i}^T \mathbf{H}^{-1} \tilde{\mathbf{n}}_\mathbf{j}$. Although we have to solve the above system, this method converges experimentally much faster than the other: it only needs one or two iterations to find the correct values for the $f_I$. It cannot be used as such in the case of the length conservation or of the coherence between positions and quaternions since the direction $\tilde{\mathbf{n}}$ would not be constant.

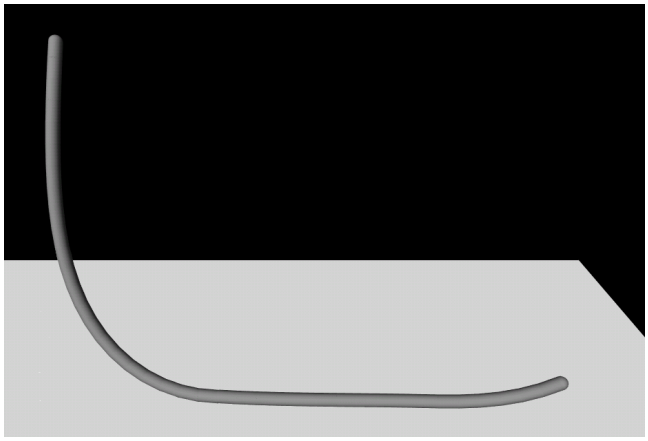

Figure 3: Deformation of a cable due to the contact to a plane

# 6 Solver

## 6.1 General principles

Since the cables and hoses do not have a high dynamic range, considering a static solution at each time step is sufficient for most of the applications we are concerned with, like wire routing or assembly simulation. Dynamic phenomena like fast oscillations are excluded by the quasi-static system: the system is after each time step at equilibrium. Numerical oscillations, which are often a problem for stiff spring-mass systems, are also excluded by the absence of speed as variables. We use an energy minimizing algorithm for solving the system. Note that the system could be easily modified to become a dynamic one, introducing the velocities as supplementary variables and solving the system for example with the algorithm proposed by Baraff [Baraff 1996]. It would be necessary to add damping forces for each kind of interaction, following the formulation of Baraff: if the constraint is $C(\mathbf{X}) = 0$, the energy is $E = \frac{1}{2} k C(\mathbf{X})^T C(\mathbf{X})$, the forces are of the form $-k \frac{\partial C(\mathbf{X})}{\partial \mathbf{X}} C(\mathbf{X})$ and the damping forces of the form $-k \frac{\partial C(\mathbf{X})}{\partial \mathbf{X}} \dot{C}(\mathbf{X})$ .

If the norm of the forces vector $\mathbf{F}$ is null (in practice small enough $\|\mathbf{F}\| < \varepsilon$ in order to account for numerical error), the system is at equilibrium and we have the solution we were looking for. If not, we minimize the energy until we find an equilibrium. The basic hypothesis is that the new solution (for slightly modified conditions: a point has been moved between the two frames for example) should be near to the old one, and the old solution vector $\mathbf{X_{old}}$ can thus be used as a good starting point for the search of the new solution. Our algorithm is iterative. In each loop, a sequence of different algorithms is used until a smaller value for the energy is found. If a particular algorithm gives a solution, $\mathbf{X_{old}}$ is replaced by the new solution and a new loop begins until the equilibrium is reached. If it does not find a better solution, the next algorithm is used. When the difference either in position $\|\mathbf{X_{new}} - \mathbf{X_{old}}\|$ or in energy $\|E_{new} - E_{old}\|$ is smaller than a predetermined value, the loop is stopped. The different algorithms that we use are in order: Newton's Method applied to the forces, non-linear Conjugate Gradient Method, Linear Conjugate Gradient Method, Steepest Descent Method and if all else fails, a linear search along the forces. It is important to note that in the immense majority of the cases, only the first or the first two are used; the other algorithms serve as a security for particular stiff cases or for tuning the different constants for the interactions. This structure allows us to have at the same time a fast response in usual cases while retaining the robustness necessary to deal with stiff cases.

## 6.2 Individual algorithms

The first algorithm is the Newton Method applied to the forces. It is particularly efficient near the equilibrium. However, it is well-known that it only converges if the starting solution is near enough to the equilibrium. If the energy in a point $\mathbf{X}+\mathbf{h}$ is approximated by the Taylor series in $\mathbf{X}$, $E_{approx} = E(\mathbf{X}) - \mathbf{F}^T\mathbf{h} - \frac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}$. This approximation is minimum when its gradient is null, i.e. for $\mathbf{F} + \mathbf{H}\mathbf{h} = 0$ which leads to $\mathbf{h} = -\mathbf{H}^{-1}\mathbf{F}$ and $\mathbf{X_{new}} = \mathbf{X_{old}} - \mathbf{H}^{-1}\mathbf{F}$. For inverting the Hessian, which is a strongly banded matrix, we use a simple Gauss algorithm slightly modified for taking into account all the zero-elements of the matrix.

The conjugate gradient method uses a family of $\mathbf{H}$-conjugated vectors (two vectors $\mathbf{u}$ and $\mathbf{v}$ are $\mathbf{H}$-conjugated if $\mathbf{u}^T\mathbf{H}\mathbf{v} = 0$ with $\mathbf{H}$ a symmetric positive definite matrix) to look iteratively for the solution to the system $\mathbf{H}\mathbf{h} + \mathbf{F} = 0$. In the non-linear method, the matrix $\mathbf{H}$ is recalculated during the process, while the linear method keeps it constant.

The steepest descent and the line search use the forces as a search direction (The forces are the opposite of the gradient, and thus indicate the direction of the steepest descent). Both methods look for a coefficient $\alpha$ such that $\mathbf{X_{new}} = \mathbf{X_{old}} + \alpha\mathbf{F}$ has a minimum energy. In the case of the steepest descent, the Hessian is used to approximate $E(\alpha) = E(\mathbf{X_{new}}(\alpha))$ by a parabola and finding its minimum, which is found for $\alpha = -\frac{\mathbf{F}^T\mathbf{F}}{\mathbf{F}^T\mathbf{H}\mathbf{F}}$. The linear search uses decreasing powers of 2 ($\alpha = \frac{1}{2^k}$) until it finds a solution with a lower energy.

# 7 Integration in a Virtual Reality environment

The simulation is integrated in the Virtual Reality software *veo* of DaimlerChrysler Research and Technology. It provides the whole environment for the simulation, such as graphics, scenes and objects handling... The cable is represented as a sequence of spheres and cylinders. The spheres are set at the discretization points. The

| Number of Points $n$ | Mean Calculation Time in ms |
|---|---|
| 10 | 6,25 |
| 20 | 12,95 |
| 30 | 19,49 |
| 40 | 21,43 |
| 50 | 25,55 |
| 60 | 27,11 |
| 80 | 29,89 |
| 100 | 38,86 |
| 120 | 45,94 |
| 140 | 54,97 |
| 160 | 60,92 |
| 180 | 80,04 |
| 200 | 77,09 |
| 250 | 104,45 |
| 300 | 128,31 |

Table 1: Influence of the number of points on calculation time. The calculation time is proportional to the number of discretization points. The banded structure of the Hessian allow a good scalability of the system. Performance of simulation on a Pentium M 1.5GHz.

cylinders are set along the segments joining two consecutive points. The cylinders and spheres can be moved (in the geometrical limits permitted by the simulation) easily by selecting them and moving them with the spacemouse. At each frame, the simulation checks whether the eventually selected objects in the scene correspond to a part of a cable. If yes, its new position and orientation are taken into account, the simulation calculates the new solution and the new positions and orientations of all the spheres and cylinders are passed to the Virtual Reality software for representation.

A seamless integration in the processes already in place is necessary for the users. The data from the construction - usually in CATIA - are tessellated and converted automatically to the OpenInventor format, used for the Virtual Reality application. We also have an interface that allows to convert all or part of a tessellated wire harness or hose to a flexible one having the same characteristics (like length, radius, position...). A partial conversion is useful since some of the construction parts represent a huge harness extended in the whole vehicle. The connectivity information of the tessellation triangles is parsed to create a graph representing the interconnections of the vertices. Geometrical considerations allow to find cross-sections on the surface at a given point that we use for cutting the tree at the appropriate vertices and for calculating the coordinates of points on the centerline, from which we construct the new flexible object. A copy of the tessellation with the remaining points is also created for the graphical representation. A complete new cable can also be created between two points of the scene.

A graphical user interface is available: it groups the commands that are useful for the simulation. The simulation can be turned on or off, handles can be added or removed at any selected point or can be set equidistant automatically along the cable, geometrical parameters like the radius or the length and material parameters like the Young Module and the Poisson coefficient can be changed interactively, the new centerline of the cables can be exported to CATIA via the VDA-FS (*Verband der Automobileindustrie - Flaechen-Schnittstelle* )format and can then be used as a draft for the construction. Our tool has already been used and tested by pilot users who have reacted positively to the possibilities offered to them. It is used in various contexts: on the one hand during assembly simulation and the other hand for cable routing. In assembly simulation, flexible parts like cables or hoses are often in the way for mounting another part of the vehicle. The idea is to make the problematic part flexible (which is represented as a rigid body), push it apart out

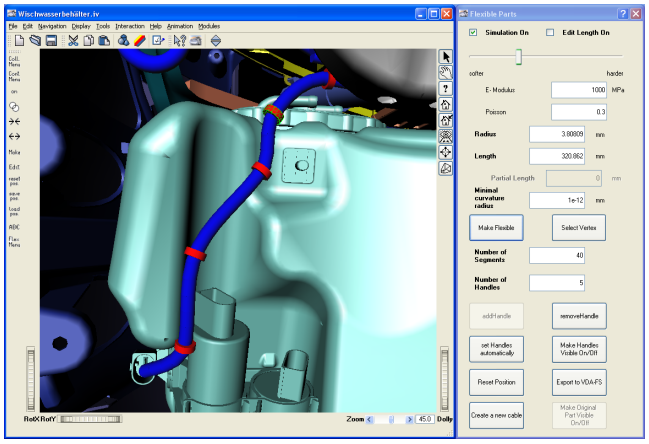


Figure 4: The Graphical User Interface

of the way as it could be done with the physical parts and continue with the assembly simulation. For wire routing, a new path for the cable needs to be defined and constructed or adapted to changes in the construction of other parts. The length of the cable may need to be adapted to the new form. We have two special modes for this, changing the handling of the stretch forces. In both cases, the integral forces relative to the length are removed. In the first mode, the user chooses two points on the cable between which the segments can be elongated either by pulling them or by typing the new length in a text field in the GUI. The reference length is then replaced either by the actual length of the segment or by the value of the text input. The length can then be changed smoothly from one iteration to another until the user is satisfied with the result. In the second mode, the stretch energy is set to

$$E_{Length,i} = \frac{k_{Length}}{2} L_i^2$$

with a very soft value for $k_{Length}$. The cable tends to stretch or to become shorter, finding automatically an optimal length for given endpoints and handles, thus allowing the user to route intuitively without worrying about the length. Playing with the ratio of $k_{Length}$ and the bending and torsion stiffnesses make the form of the cable vary form “rather straight” to “rather bend”.

## 8 Conclusion and future work

In this paper, we presented a virtual environment suitable for the simulation of cables. Our approach modeled the cable with an extended spring-mass system that was solved with an energy minimizing algorithm. The cable was modeled using the Cosserat theory, taking into account the conservation of length, the weight, the bending and the torsion. In order to easily formulate the bending and torsion interaction, we used a mixed coordinate system where each mass point had three space coordinates and the orientation of each segment was represented by a normalized quaternion. Each type of interaction was then calculated on the base of the coordinate type that is best suited: on the one hand, the length conservation and the contact forces with the positions, on the other hand the bending and torsion with the quaternions. Additionally, forces for the coherence between positions and quaternions and for the normalization of quaternions were employed.

For constraints that must be exactly enforced, like the length conservation, the coherence between positions and quaternions or the

collision, we added an integral force at each point in the same direction as the proportional spring force. The system was solved for a constant value of those forces which were then updated as a function of the result. This mechanism allowed softer springs (and thus a less stiff system) while at the same time exactly enforcing the constraints.

Future work will include the extension of the functionalities of the user interface to better meet user needs, a better embedding of the collision response and an extension of the tool to also simulate other kinds of flexible parts.

## References


ANTMAN, S. S. 1995. *Nonlinear Problems of Elasticity*, vol. 107 of *Applied Mathematical Sciences*. Springer Verlag.

BARAFF, D. 1996. Linear-time dynamics using Lagrange multipliers. *Computer Graphics 30*, Annual Conference Series, 137–146.

BARZEL, R. 1997. Faking dynamics of ropes and springs. *IEEE Comput. Graph. Appl. 17*, 3, 31–39.

BUCK, M., AND SCHÖMER, E. 1998. Interactive rigid body manipulation with obstacle contacts. *Journal of Visualization and Computer Animation 9*, 243–257.

FINCKH, H., STEGMAIER, T., AND PLANCK, P. H., 2004. Numerische Simulation der mechanischen Eigenschaften textiler Flächengebilde - Gewebeherstellung.

GOSS, V. G. A., VAN DER HEIJDEN, G. H. M., THOMPSON, J. M. T., AND NEUKIRCH, S., 2005. Experiments on snap buckling, hysteresis and loop formation in twisted rods.

HERGENROETHER, E., AND DAEHNE, P., 2000. Real-time virtual cables based on kinematic simulation.

LOOCK, A., AND SCHÖMER, E. 2001. A virtual environment for interactive assembly simulation: From rigid bodies to deformable cables. In $5^{th}$ *World Multiconference on Systemics, Cybernetics and Informatics (SCI'01)*, vol. 3, 325–332.

PAI, D. K. 2002. Strands: Interactive simulation of thin solids using cosserat models. *Comput. Graph. Forum 21*, 3, 347–352.

RUBIN, M. B. 2000. *Cosserat Theories: Shells, Rods and Points*. Kluwer Academic Publ., Dordrecht,.

SAUER, J., AND SCHÖMER, E. 1998. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, 153–161.