Implementation of Curved Strands II: Dynamics

J Tessendorf
Cinesite Digital Studios

D D Weston
Cinesite Digital Studios

December 3, 1998

1 Introduction

The Frenet-Serret Apparatus is an excellent method of describing space curves in our application. It neatly separates the intrinsic shape of the curve from spatial orientation. Numerical implementation of the necessary code is straightforward. By using curvature and torsion as the underlying variables, we are assured that the strands will maintain fixed length no matter what the shape. In fact, curvature and torsion are unconstrained variables which preserve the quality of the space curve representation.

The "cost" for this elegant representation of space curves is complexity of the dynamical equations. The cost is not as high as one might suppose, however, because of a variety of factors. First, formulating the dynamics directly in terms of the curvature and torsion guarantees that the strand will not stretch or shrink. Second, the dynamical equations follow directly from Newton's Laws, so there is a clear connection with other dynamical problems such as particle motion or rigid body dynamics.

This paper depends on the notation and results of its companion paper[1].

2 Strand Dynamics

The problem of subjecting strands to forces and computing the consequent motion has not been solved well in computer graphics. However, we take a new approach here, combining the Frenet-Serret Apparatus with Newton's Laws to construct a dynamical model for the evolution of curvature and torsion over time. We are able to convert the second order differential equations for position into equivalent second order differential equations for curvature and torsion, and reduce these into pairs of first order equations for curvature and torsion "amplitudes". These first order equations provide the starting point for the discretizing the dynamics and building a numerical solution.

We begin by formulating Newton's Laws in a way that is approapriate for strands. Beginning with $F = ma$, with create dynamical equations for the Frenet-Serret Appa-

[1] J. Tessendorf and D.D. Weston, "Implementation of Curved Strands I: Frenet-Serret Framework," Cinesite Digital Studios, 1998.

ratus. From these equations, we focus in specifically on the dynamics of curvature and torsion, then convert their second order equations into first order equations.

2.1 Newton's Law for Strands

The fundamental starting point for Newtonian dynamics is to express the evolution of a point particle at position $\vec{x}(t)$ at time t as

$$\frac{d^2\vec{x}(t)}{dt^2} = \vec{F}(\vec{x}(t)) \ , \tag{1}$$

where the (acceleration) force $\vec{F}$ is some time-dependent quantity. The extension to strands involves extending this equation to a series of points that are strung together and do not separate. In fact, for this problem, we will take the strand to be "incompressible", i.e. does not stretch or shrink, but stays a fixed arclength and fixed profile of thickness. Because the strand does not alter its cross-sectional structure, the equation for the motion of the piece of strand at arclength position $\vec{x}(s,t)$ as a function of time t is

$$\frac{d^2\vec{x}(s,t)}{dt^2} = \vec{F}_{strand}(s,t) \ . \tag{2}$$

Finally, strands subjected to a force on one end experience that force on the other via an elastic response that has some effective wave speed v. Hence the double time deriviative is two components:

$$\frac{d^2}{dt^2} = \frac{\partial^2}{\partial t^2} - v^2\frac{\partial^2}{\partial s^2} \ . \tag{3}$$

This ensures the proper restoring forces. In order to prevent us from confusing this with just a time deriviative, we will change notations to

$$\partial^2 \equiv \frac{\partial^2}{\partial t^2} - v^2\frac{\partial^2}{\partial s^2} \ . \tag{4}$$

There is an important difference between $\vec{F}_{strand}$ and $\vec{F}$: The strand force does not have a component in the direction of the tangent of the strand, since that would stretch is. The relationship between the two is

$$\vec{F}_{strand}(s,t) = \left(1 - \hat{T}(s)\hat{T}(s)\right) \cdot \vec{F}(\vec{x}(s,t)) \ . \tag{5}$$

This equation, along with initial conditions, is the complete description of the dynamics of strands. However, we know that the actual strand motion is constrained by the fact that it is a continuous strand: it does not break apart, and it can have phenomena such as waves and bending. This is because the underlying dynamically active quantities in a strand are the curvature and torsion. If equation 5 can be converted into corresponding equations for $\kappa(s,t)$ and $\tau(s,t)$, then their direct (numerical) solution will contain all of the constraints that are inherent in a strand. The remainder of the discussion on dynamics is focused on that goal.

The next step is to convert equation 5 into equations for the Frenet-Serret Apparatus. This can be accomplished by taking up to three derivatives with respect to arclength. The resulting equations are

$$\partial^2 \hat{T}(s,t) = \frac{d\vec{F}_{strand}(s,t)}{ds} \tag{6}$$

$$\partial^2 \kappa(s,t)\hat{N}(s,t) = \frac{d^2\vec{F}_{strand}(s,t)}{ds^2} \tag{7}$$

$$\begin{aligned}\partial^2 \kappa(s,t)\tau(s,t)\hat{B}(s,t) &= \frac{d^3\vec{F}_{strand}(s,t)}{ds^3} \\ &+ \partial^2 \left\{ \kappa^2(s,t)\hat{T}(s,t) - \frac{d\kappa(s,t)}{ds}\hat{N}(s,t) \right\}\end{aligned} \tag{8}$$

Keep in mind that these three equations are not independent of each other. They were all derived from equation 5, and they are merely ways of rephrasing it and bringing out some aspects of Newton's Laws. The original equation described dynamical evolution of up to three degrees of freedom. Because the object is a strand, there are really only two degrees of freedom, and from the properties of the Frenet-Serret apparatus, we know that $\kappa(s,t)$ and $\tau(s,t)$ are those two degrees of freedom. What we need to do is turn the equations above into more succinct equations for the dynamics of the curvature and torsion. This is done in the next section, but as a beginning, we note that, taking the inner product of the first equation and the tangent vector, we get the following constraint:

$$\left| \partial \hat{T}(s,t) \right|^2 = -\hat{T}(s,t) \cdot \frac{d\vec{F}_{strand}(s,t)}{ds} \ , \tag{9}$$

where the notation $\left| \partial \hat{T}(s,t) \right|^2$ is short for

$$\left| \partial \hat{T}(s,t) \right|^2 \equiv \left| \frac{\partial \hat{T}(s,t)}{\partial t} \right|^2 - v^2 \left| \frac{\partial \hat{T}(s,t)}{\partial s} \right|^2 \ , \tag{10}$$

so that the result can be restated as

$$\left| \frac{\partial \hat{T}(s,t)}{\partial t} \right|^2 = v^2\kappa^2(s,t) - \hat{T}(s,t) \cdot \frac{d\vec{F}_{strand}(s,t)}{ds} \ . \tag{11}$$

This equation is a constraint on the evolution of the curvature. In a numerical implementation, this constraint could serve as a quantitative test of the numerical validity and stability of the algorithms and code used, as discussed in later sections.

2.2 Dynamical Equations for Curvature and Torsion

Using a set of steps similar to those used to derive equation 9 from equation 6, we take the inner product of equation 7 with $\hat{N}$ to get

$$\left\{ \partial^2 - \Omega_N^2(s,t) \right\} \kappa(s,t) = \hat{N}(s,t) \cdot \frac{d^2\vec{F}_{strand}(s,t)}{ds^2} \tag{12}$$

with $\Omega_N^2(s,t) \equiv \left|\partial \hat{N}(s,t)\right|^2$ as the "relaxation coefficient".

The dynamical equation for τ, obtained from equation 8 in the same fashion, is actually an equation for the product $\kappa\tau$

$$\left\{\partial^2 - \Omega_B^2(s,t)\right\}(\kappa(s,t)\tau(s,t)) = F_B(s,t)\ , \tag{13}$$

where F_B is

$$F_B(s,t) = \hat{B}(s,t)\cdot\left\{\frac{d^3\vec{F}_{strand}(s,t)}{ds^3} + \partial^2\left(\kappa^2(s,t)\hat{T}(s,t) - \frac{d\kappa(s,t)}{ds}\hat{N}(s,t)\right)\right\} \tag{14}$$

and $\Omega_B^2(s,t) = \left|\partial \hat{B}(s,t)\right|^2$.

2.3 Solution Method for Curvature

The dynamical equations for κ and $\rho \equiv \kappa\tau$ are similar in structure, and so we concentrate first on building a solution for κ, and use the methodology later to build the solution for τ.

The quantities $\Omega_N^2(s,t)$ and

$$F_N(s,t) = \hat{N}(s,t)\cdot\frac{d^2\vec{F}_{strand}(s,t)}{ds^2} \tag{15}$$

are clearly dependent on κ and τ. However, the aproximation method we adopt here is to treat them as quasi-independent, solving iteratively for all of the quantities needed.

There is no general exact solution to equation 12 for arbitrary Ω_N and F_N. There has been a lot of work on approximate methods however, and they may at sometime in the future prove useful. At present however, our approach relies on a relatively simple pertubations and predictor-corrector technique.

To begin, κ is divided into homogeneous and inhomogeneous parts $\kappa(s,t) = \kappa_h(s,t) + \kappa_F(s,t)$ which satisfy the dynamics and initial conditions

$$\begin{aligned} \partial^2\kappa_h(s,t) &= 0 \qquad (16)\\ \kappa_h(s,t=0) &= \kappa_0(s)\\ \frac{d\kappa_h(s,t=0)}{dt} &= \dot{\kappa}_0(s)\\ \partial^2\kappa_F(s,t) &= F_N(s,t) + \Omega_N^2(s,t)\ \kappa(s,t) \qquad (17)\\ \kappa_F(s,t=0) &= 0\\ \frac{d\kappa_F(s,t=0)}{dt} &= 0 \end{aligned} \tag{18}$$

The homogeneous term represents the evolution of the curve do to initial and boundary conditions. The inhomogeneous term adds additional evolution from external forces and the "centripetal" behavior of the local coordinate system, as expressed in $\Omega_N^2(s,t)$. Even when no external force is present, the inhomogeneous computation is needed.

2.3.1 Exact Solution for the Homogeneous Term

By inspection, one can directly verify that the solution for κ_h is

$$\kappa_h(s,t) = \frac{1}{2}\left[\kappa_0(s-vt) + \kappa_0(s+vt)\right] + \frac{1}{2v}\int_{s-vt}^{s+vt} ds'\ \dot{\kappa}_0(s')\ . \tag{19}$$

2.3.2 Inhomogeneous Term

The inhomogeneous part κ_F can be created using a Green's function, in the form

$$\kappa_F(s,t) = \int_0^t dt' \int_0^L ds'\ G(s,s';t-t')\left[F_N(s,t') + \Omega_N^2(s',t')\ \kappa(s',t')\right]\ , \tag{20}$$

where L is the length of the curve and the Green's function satisfies:

$$\partial^2 G(s,s';t-t') = \delta(s-s')\ \delta(t-t') \tag{21}$$

$$G(s,s';t-t')|_{t=0} = 0 \tag{22}$$

Continuity conditions also give the properties

$$G(s,s';t-t')|_{t<t'} = 0 \tag{23}$$

$$G(s=0,s';t-t') = 0 \tag{24}$$

$$G(s=L,s';t-t') = 0 \tag{25}$$

These equations have the solution

$$G(s,s';t-t') = \theta(t-t')\ \frac{2}{\pi v}\sum_{n=1}^{\infty}\frac{1}{n}\sin\left(\frac{n\pi s}{L}\right)\sin\left(\frac{n\pi s'}{L}\right)\sin\left(\frac{n\pi v}{L}(t-t')\right) \tag{26}$$

with $\theta(\cdot)$ as the Heaviside step function.

There are two issues in the numerical implementation of the Green's function will should now be resolved: How many terms in the Fourier series sum of equation 26 must be kept; and what time steps should be used to get a smooth evolution? It may be that these questions must be answered through the behavior of the integrand $F_N + \Omega_N^2\kappa$, e.g. if they turn out to have sensitive responses to grid spacing and dynamics. Without addressing that aspect of the question, there are some bounds set by the form of the Green's function.

Because all quantities are sampled in arclength with a spacing Δs, the terms in the Green's function Fourier expansion with $n\pi\Delta s/L > 1$ account for spatial behavior on scales smaller than the spacing, so one criterion for cutting off the sum is to stop at n_{max} with

$$n_{max} = \frac{L}{\Delta s}\ , \tag{27}$$

i.e. the number of terms in the expansion is equal to the number of segments the curve is partitioned into.

Additionally, if every term in the sum changes only a little with each time step, then smooth evolution of the solution will be insured. This is accomplished if, for the deepest term, $n_{max}\pi v\Delta t/L < 1$, where Δt is the evolution time step. Using the above value of n_{max}, this becomes

$$\Delta t < \frac{\Delta s}{\pi v} \,. \tag{28}$$

2.4 Solution for Torsion

The solution for torsion follows analogously to that for curvature, except that the solution is for the product $(\kappa\tau)$. The final solution is then $\tau(s,t) = (\kappa\tau)(s,t)/\kappa(s,t)$.

3 Numerical Implementation of Dynamics

The final solutions for κ and τ are complex because $\Omega_{\{NB\}}(s,t)$, $F_{\{NB\}}(s,t)$, and $\varphi_{\{NB\}}(s,t)$ are dependent on all of the components of the Frenet-Serret Apparatus, including dependence on $\kappa(s,t)$ and $\tau(s,t)$. Even for simple forces, such as gravity or wind drag, the direct solution is not easy to obtain analytically or numerically. Fundamentally, what has been done here is that the differential equations for the dynamics have been partially integrated, converting them into integro-differential equations for the curvature and torsion. This form, however, is more convenient, numerically more stable, and potentially more accurate than a direct assault on the full differential equations. An important ingredient of the numerical implementation will be the choice of predictor-corrector scheme to speed convergence and contain instability.

The numerical method that we will try initially is a predictor-corrector approach, using the solutions above to iteratively adjust the predictor until (hopefully) the iteration converges to a solution.

For a time series of curves, with time interval Δt between computations, the steps are as follows:

1. Initialize predictors $\kappa^-(s,t+\Delta t)$, $\tau^-(s,t+\Delta t)$ to simple initial guesses:
$$\begin{aligned} \kappa^-(s,t+\Delta t) &= \kappa(s,t) \\ \tau^-(s,t+\Delta t) &= \tau(s,t) \end{aligned}$$
2. Compute the Frenet-Serret Apparatus and appropriate derivatives, to obtain $\Omega_{\{BN\}}$, $F_{\{BN\}}$, and $\varphi_{\{BN\}}$.
3. Compute correctors $\kappa^+(s,t+\Delta t)$ and $\tau^+(s,t+\Delta t)$ using the solutions of section 2.
4. Check convergence criteria.
5. If iteration has not converged, set
$$\begin{aligned} \kappa^-(s,t+\Delta t) &\to \kappa^+(s,t+\Delta t) \\ \tau^-(s,t+\Delta t) &\to \tau^+(s,t+\Delta t) \end{aligned}$$
and repeat beginning at step 2.

6. If iteration has converged, set

$$\begin{aligned}\kappa(s,t+\Delta t) &= \kappa^{+}(s,t+\Delta t)\\ \tau(s,t+\Delta t) &= \tau^{+}(s,t+\Delta t)\end{aligned}$$

and clean up.

Obviously, two of the important issues in implementing this approach are (1) what should the convergence criteria be, and (2) how fast is the convergence to an acceptably accurate answer. At present, the answers to both questions needs to be obtained by experimenting with a working code.

A "natural" convergence criterion is the magnitude of the difference between prediction and correction. Clearly, a zero difference is a fully converged solution. However, in the absence of full convergence, it is not clear how sensitive the difference magnitude is. Also, the difference is a function of position on the strand, so some strand-averaged difference might be more beneficial. Also, if there are dramatic oscillations in the predictor-corrector loop, a better choice of updating might be a kalman-like weighting process:

$$\begin{aligned}\kappa^{-}(s,t+\Delta t) &\rightarrow (1-\epsilon)\,\kappa^{+}(s,t+\Delta t)+\epsilon\,\kappa^{-}(s,t+\Delta t)\\ \tau^{-}(s,t+\Delta t) &\rightarrow (1-\epsilon)\,\tau^{+}(s,t+\Delta t)+\epsilon\,\tau^{-}(s,t+\Delta t)\end{aligned}$$

This kind of weighting will dampen oscillations.

A second convergence criterion that could be used derives from equation 9. Using $\kappa^{+}(s,t+\Delta t)$ and $\tau^{+}(s,t+\Delta t)$, we can compute the dimensionless quantity

$$\gamma(s,t+\Delta t) = 2\frac{\left|\dot{\hat{T}}(s,t+\Delta t)\right|^{2}+\hat{T}(s,t+\Delta t)\cdot\left(d\vec{F}_{strand}(s,t+\Delta t)/ds\right)-v^{2}\kappa^{2}(s,t)}{\left\{\left|\dot{\hat{T}}(s,t+\Delta t)\right|^{4}+\left|\hat{T}(s,t+\Delta t)\cdot\left(d\vec{F}_{strand}(s,t+\Delta t)/ds\right)\right|^{2}+v^{4}\kappa^{4}(s,t)\right\}^{1/2}} \tag{29}$$

When convergence is good, $\gamma \rightarrow 0$. This is a second method of testing convergence behavior.