**Animating Oscillatory Motion With Overlap: Wiggly Splines**

The spline generalizes traditional piecewise cubics when its resonance and damping are set to zero, but creates oscillatory animation when its resonance and damping are changed. The spline provides  
a combination of direct manipulation and physical realism. To create overlapped and propagating motion, we generate phase shifts of the Wiggly Spline, and use these to control appropriate degrees of freedom in a model. The phase shifts can be created directly by procedural techniques or through a paint-like interface. A further option is to derive the phase shifts statistically by analyzing a time-series of a simulation. In this case, the Wiggly Spline makes it possible to canonicalize a simulation, generalize it by providing frequency and damping controls and control it through direct manipulation.

Wiggly Splines re-create traditional spline behavior when their resonant frequency and damping are set to zero, but create oscillatory animation when their resonance and damping are set to be representative of an oscillatory physical system. The Splines provide a familiar interactive interface supporting direct manipulation while at the same time embedding the physical realism of an oscillatory differential equation into the spline itself. They provide the mix we seek between realism and control. With Wiggly Splines, we can address not only individual degrees of freedom in a model that oscillate, but also the more broadly important problem of oscillatory motion that involves propagating phenomena like waves. With a phenomenon like belly jiggle, the nature of this propagation will determine whether the belly looks stiff, “sloshy” or somewhere in-between.  
In particular, Wiggly Splines allow us to address a key issue known to animators by the term “overlap.” The term generally refers to the fact that different degrees of freedom of any given model need to accelerate and decelerate at different times. If all the degrees of freedom vary synchronously in phase, the result is motion that usually looks very stiff or “computery.” Breaking up the synchrony to the point that propagation effects are visible is vital to achieving an acceptably natural appearance.

To apply the Spacetime Constraints formalism, we take our physical system with a single mass and spring, add constraints provided by the animator, and subject to these constraints, minimize an objective function which penalizes non-physical and inefficient motion

In order to make the solution of equation 6 interesting, we need to add some constraints. The primary constraint, of course, is the ability to set a particular value at a particular time – an interpolation constraint

In addition to setting values, animators are used to establishing tangent constraints

In forward simulation, the only way to modify the curve is by changing the simulation parameters, but the relationship between the simulation parameters and the resulting curve can be very complex. Moreover, there is no guarantee that *any* settings of the parameters will yield the desired  
motion. Animators faced with such circumstances will frequently give up and accept an artistic compromise. With our system, by contrast, the animator can edit motion curves with direct manipulation. If the animator wants absolute control of a piece of the curve, he or she can set the resonant frequency and damping of the Wiggly Spline to zero on the corresponding interval and edit that segment like any other traditional piecewise cubic. With Wiggly Splines, the animator gets the default behavior of forward simulation, but can edit the output with the absolute control and  
directability of traditional splines.

Extending the Wiggly Spline to the complex domain turns out to be relatively simple. We begin as before by having the animator select the resonant frequency and damping