Polygonal Knot Simulation System

Overview

A knot is an oriented closed curve in three-space. A polygonal knot can be understood as a piecewise linear approximation to a knot. Knot theory has applications in exploring the behavior of DNA strands and may have applications in other domains as well. This project will develop a system for modeling polygonal knots and for simulating polygonal knot behavior. The project diverges from the web-based systems that have been typical of our program in the past and may even use C programming for a good majority of the project.

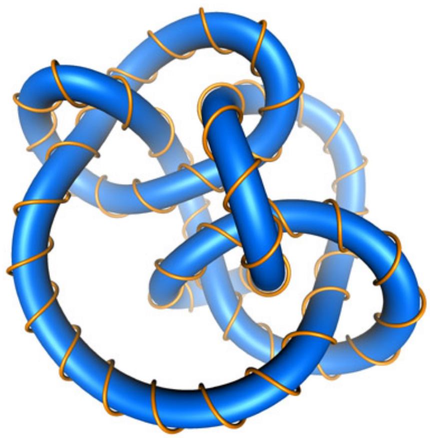

Figure 1: Knot Image (http://www.colab.sfu.ca/KnotPlot/13may02a.jpg)

What is a Polygonal Knot

Consider a polygonal knot to be a sequence of n connected edges K = {e1, e2, e3, ..., en}. Let len(e) be the length of edge e and md(e1, e2) be the minimum distance between edges e1 and e2. An energy function, E_{md}, is then defined as:

$$E_{MD}(K) = \sum_{i,j\leq 1}^{n} \frac{len(e_i)len(e_j)}{md(e_i, e_j)^2}.$$

A common problem in polygonal knot topology is to find a configuration (a positioning of knot vertices) that minimizes this energy while not allowing edges to 'pass through' each other. Another way of saying this is to consider the knot as a flexible but real thing where the nodes of the knot can be moved around into different positions.

The movement of vertices can be simulated using a physically based approach that applies a 'stretching' force to each edge in addition to an electrostatic repulsive force to each node. These forces then apply a force to each node of the knot that serves to move the knot in a well-defined manner assuming that edges have a mass that is dependent upon their length.

Another problem is this. Consider placing a knot into a "field" of "knots" . The field is such that the knots are not movable and not necessarily closed. If we apply a gravitational field to the knot we might wonder what path it would take through this field and we might attempt to characterize the path in terms of its overall velocity based on the complexity of the knot and the density of the field. This type of simulation is being used by Biologists to simulate gel electrophoresis.

Consider placing 500 knots each with 500 edges into a field of 5000 knots. Simulation of this system will be computational complex and a real-time visualization of the results would be very useful.

System Components

The system will be composed of several software components. The main components of the system are the computational engine, a visualization engine and an application.

Computational engine

- ***Description***: The computational engine provides services that efficiently compute properties and dynamic behavior of a large set of high resolution knots. The computational engine must be able to take some canonical representation of a set of knots, internalize this representation for efficient large-scale computation, and return the result in some canonical form.
- ***Challenges***: The computational engine should leverage parallel computation and make heavy use of GPU technology. Could also explore physical simulation engines such as PhysX.

Rendering engine

- ***Description***: An OpenGL application that is able to visualize a collection of polygonal knots. Various parameters will include color schemes, level of detail, geometric properties (size/shape of knot rendering) and other properties such as 'sphere of influence', focal points, and transparencies.
- ***Challenges***: This engine may also may use of GPU processing and could be highly parallelized. Most likely be written in OpenGL.

Application Development

- ***Description***: This is not a component so much as an application (or set of applications) that uses the system to perform some simulation(s). The simulations will have a small set of controlling parameters and will require some understanding of the expected results. Managing large data sets, generating charts and properties of the datasets will be important.
- ***Challenges***: This is a high-level layer that does not actually require a GUI but could benefit from a GUI. It uses the other components to create a simulation, store the results, and report on various aspects of the simulation.