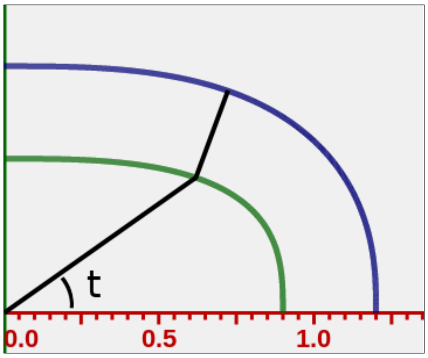
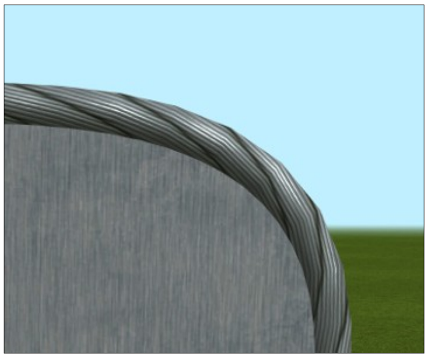
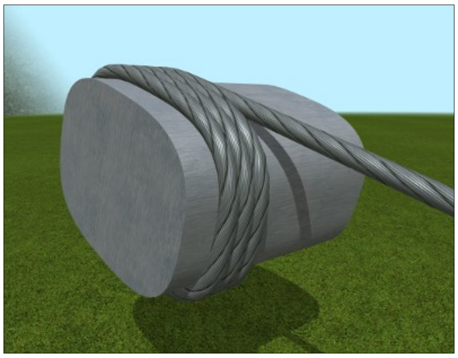
Perimeter of a Hyperellipse, Hyperellipse Coated With Layer



Because of a symmetry, ¼ perimeter is considered, $t \in [0, \pi/2]$.
Parametric equation of a hyperellipse:

$$x = \cos(t)\left(\frac{\cos^n t}{a^n} + \frac{\sin^n t}{b^n}\right)^{-1/n}, \quad y = \sin(t)\left(\frac{\cos^n t}{a^n} + \frac{\sin^n t}{b^n}\right)^{-1/n}, \quad n \geqslant 2 \,. \tag{1}$$

Parametric equation of a hyperellipse coated with layer of thickness r_1:

$$x = \cos(t)\left(\frac{\cos^n t}{a^n} + \frac{\sin^n t}{b^n}\right)^{-1/n} + r_1 \frac{\cos^{n-1} t}{a^n}\left(\frac{\cos^{2n-2} t}{a^{2n}} + \frac{\sin^{2n-2} t}{b^{2n}}\right)^{-1/2}, \tag{2}$$

$$y = \sin(t)\left(\frac{\cos^n t}{a^n} + \frac{\sin^n t}{b^n}\right)^{-1/n} + r_1 \frac{\sin^{n-1} t}{b^n}\left(\frac{\cos^{2n-2} t}{a^{2n}} + \frac{\sin^{2n-2} t}{b^{2n}}\right)^{-1/2}, \quad n \geqslant 2 \,.$$

<u>Tasks</u>:

- Find the length of ¼ perimeter.
- Given distance along the curve away from the intersection with axis X positive direction (X+), find point on the curve and normal vector at the point. This is performed every frame for every piece of wound rope on graphics processing unit (GPU). The aim is to reduce amount of computations required to perform this task.
- Find distance along the curve from given point on curve to the intersection with X+.

In case of n=2 (ellipse) there are extensive materials on the Internet, including elliptic integrals.

For parametric planar curves, arc length L between two points $t=0$ and $t=\varphi$ is expressed with following relation: $L(\varphi)=\int_0^{\varphi} \sqrt{(dx/dt)^2+(dy/dt)^2}\,dt$. Taking derivatives from (1), substituting, simplifying results in a formula for arc length of a hyperellipse:

$$L(\varphi)=\int_0^{\varphi}\left(\frac{\cos^n t}{a^n}+\frac{\sin^n t}{b^n}\right)^{-1/n-1}\sqrt{\frac{\cos^{2n-2} t}{a^{2n}}+\frac{\sin^{2n-2} t}{b^{2n}}}\;dt \,. \tag{3}$$

Taking derivatives from (2), substituting, simplifying results in a formula for arc length of
a hyperellipse coated with layer:

$$L(\varphi)=\int_0^{\varphi}\left(\frac{\cos^n t}{a^n}+\frac{\sin^n t}{b^n}\right)^{-1/n-1}\sqrt{\frac{\cos^{2n-2} t}{a^{2n}}+\frac{\sin^{2n-2} t}{b^{2n}}} + r_1(n-1)\frac{\cos^{n-2} t}{a^n}\frac{\sin^{n-2} t}{b^n}\left(\frac{\cos^{2n-2} t}{a^{2n}}+\frac{\sin^{2n-2} t}{b^{2n}}\right)^{-1} dt \,. \tag{4}$$

Closed-form antiderivatives of the above integrals were not found.

Functions (3), (4) are expanded into Taylor series:

$$L(t)=L(t_0) + \frac{L'(t_0)}{1!}(t-t_0) + \frac{L''(t_0)}{2!}(t-t_0)^2 + \ldots + \frac{L^{(n)}(t_0)}{n!}(t-t_0)^n + o(t^{n+1}) \,. \tag{5}$$

Here $L(t_0)$ is a length from X+ to a point about which Taylor expansion is performed. Derivative $L'(t_0)$ equals to the integrand from (3) or (4). Subsequent differentiation of $L'(t_0)$ yields higher-order derivatives.

How to find values of first N derivatives at a point, find absolute error bound on an interval using various methods is described in appendix [].

To accomplish tasks, data structure is created. It is a sequence of intervals ordered by parameter t. Each interval contains Taylor polynomial of order N approximating function value $L(t)$ with relative error not exceeding ε.

Data structure creation algorithm.

1. Take interval $t \in [0, \pi/2]$.
2. Compute values of first N derivatives at point t_0 in the center of the interval, find absolute error bound.
3. Create Taylor polynomial. Using formula (5) find arc length within the interval. Find relative error bound.
4. If relative error bound exceed given ε then remove interval. In place of removed interval create two intervals equal in t. For each of created intervals, go to step 2.
5. On every interval, record sum of lengths of previous intervals.

Number of intervals in data structure depends on parameters (n, a, b, r_1), order of Taylor polynomial N and relative error bound ε.

Hyperellipse

n	a	b	ε=10^{-6} N=5	ε=10^{-6} N=10	ε=10^{-10} N=7	ε=10^{-10} N=15
2.5*	1	1	16	8	42	18
2.5*	10	0.1	78	23	133	26
3	1	1	16	6	30	6
3	10	0.1	75	21	132	24
10	1	1	32	8	60	14
10	10	0.1	86	24	160	33
50	1	1	42	14	86	24

* reduced order Taylor polynomials (2 derivatives) on intervals containing 0 и π/2.

Hyperellipse coated with layer r_1=0.5

n	a	b	ε=10^{-6} N=5	ε=10^{-6} N=10	ε=10^{-10} N=7	ε=10^{-10} N=15
2	1	1	1	1	1	1
2	10	0.1	97	26	163	30
3	1	1	20	6	34	6
3	10	0.1	92	24	154	27
10	1	1	46	12	80	18
10	10	0.1	105	27	183	**
15	3	0.3	70	19	135	29
50	1	1	64	22	116	28

** exponent overflow in IEEE 754 double

If parameter n is non-integer then functions (3), (4) are not inifinitely differentiable on the interval $t \in [0, \pi/2]$.

In case of a hyperellipse, only $\text{floor}(n)+1$ derivatives are finite at points $t=0$ and $t=\pi/2$. The solution is to reduce order of Taylor polynomials on intervals containing that points, resulting in interval shrinking. Also in the neighborhoods of such points series convergency radii reduce and these and adjacent intervals additionally shrink. Other effect in the neighborhoods of abovementioned points is absolute values of polynomial coefficients grow substantially, require attention for possible overflow.

In case of a hyperellipse coated with layer, if $n \notin \mathbb{N}$ then at points $t=0$ and $t=\pi/2$ only $\text{floor}(n)-1$ derivatives are finite. Particularily, if $n \in (2,3)$ then only 1 derivative is finite while it requires no less than 2 derivatives to create Taylor polynomial (of order $\geqslant 1$) and to find absolute error bound. Approaches to overcome this issue were considered; implementation and description are left for the next versions.

Find point and normal given length L.

To find an interval where the point is located, binary search is performed.

On each interval it creates polynomial t(L) from Taylor polynomial L(t) using series reversion formula.

Point coordinates (x, y) depend on t by formula (1) or (2).
Components of normal vector (N_X, N_Y) are the same for both curves:

$$(N_X, N_Y) = \left[\frac{\cos^{n-1} t}{a^n}, \ \frac{\sin^{n-1} t}{b^n} \right] \Big/ \sqrt{\frac{\cos^{2n-2} t}{a^{2n}} + \frac{\sin^{2n-2} t}{b^{2n}}} \ . \quad (6)$$

Possible approach is: use t(L) to find t and then use formulae to find x, y, N_X, N_Y by t.

To reduce amount of computations, skip expensive raising to an arbitrary power it creates Taylor polynomials x(L), y(L), $N_X(L)$, $N_Y(L)$ on every interval where required.
For that, it computes values of first N derivatives of functions (1) or (2), and (6) at point t_0, in addition to t_0 it plugs derivatives from t(L) using Faà di Bruno formula.

Find curve length L between a point and X+.

This task can be performed applying one of numerical integration methods to formula (3) or (4).
The created data structure allows for less expensive method as follows.

Quarter perimeter is divided into 2 parts (“right” and “top”) by a point at which normal line forms angle 45° to X+. Parameter *t* for that point is:

$$t = \arctan\left[\left(b^n / a^n \cdot \tan\theta \right)^{1/(n-1)} \right], \quad \theta = \pi/4. \tag{7}$$

On the intervals in the “right” part it creates polynomials L(y), in the “top” part L(x) by applying series reversion formula to y(L) or x(L) accordingly. Also on every interval it records point (x,y) at the end of the interval.

To find L, first it runs binary search by x or y and finds interval where the point is located. Then Taylor polynomial L(x) or L(y) is evaluated. The evaluation requires approximately 2N multiplications and N additions where N is the order of Taylor polynomial.

Example of 3D image created using the above computations.

Hyperelliptic cylinder n=3, a=1.3, b=2 with partially wound chain r_1=0.067.