CS-446 - Digital 3D Geometry Processing
Assignment 4

Maxime Pisa, Lucas Ramirez, Philippe Weier

October 18, 2018

1 Theory Exercise

1.1 Curvature of Curves

- $k_1(s) = \frac{s^2-1}{s^2+1}$

  This function corresponds to curve **d**. We see that the function is symmetric around 0, meaning that $\forall s \in \mathbb{R}^+,\ k_1(s) = k_1(-s)$. Hence the representation must also be symmetric around the line $x = 0$. Secondly, we observe that $\lim_{s\to\infty} k_1(s) = \lim_{s\to-\infty} k_1(s) = 1$, so on both ends the curve must shape like a non-shrinking unit circle, which is only the case for curve **d**.

- $k_2(s) = s$

  This function corresponds to curve **a**. Again the function is symmetric around 0, so the representation must also be symmetric around the line $x = 0$. The curvature linearly tends to $\pm\infty$ when $s \to \pm\infty$, so on both ends the curve must shape like a shrinking unit circle, which is only the case for curve **a**.

- $k_3(s) = s^3 - 4s$

  This function corresponds to curve **c**. The funcion is not symmetric around 0, so the representation must also not be symmetric around the line $x = 0$. This is only the case for curve **c**.

- $k_4(s) = \sin(s) \cdot s$

  This function corresponds to curve **b** (by default). The curvature is divergent at both infinities because of the sin function. It also makes the curvature equal 0 periodically (every 2π), changing its sign every period. This is only the case for curve **b**.

1.2 Surfaces Area

We directly consider the general problem with N slices. We denote by S_i the slice of index $i \in \{1, 2, \ldots, N\}$, starting from the half-dome's top (this means that slice S_1 is the upper slice).

We need to compute the area $f(i)$ associated to each slice S_i. We denote by R the half-dome's radius.

$$f(i) = \int_{\frac{i-1}{R}}^{\frac{i}{R}} (2\pi \cdot R)\, dr = \pi \left[\left(\frac{i}{R}\right)^2 - \left(\frac{i-1}{R}\right)^2 \right] = \pi \cdot \frac{2i+1}{R^2}$$

The most profitable slice is the one with the lowest area:

$$\min_{i\in\{1,\ldots,N\}} f(i) = f(1) = \frac{3\pi}{R^2}$$

Hence the upper slice S_1 is always the most profitable one. In the $N = 2$ case one must pick *Part A* to make the most profit.

2 Coding Exercise

Some notes about our code:

- We added a field to the `MainWindow` structure to store the polyline's length when it's created. Otherwise, if we tried to re-compute the length at each iteration the polyline would eventually end up shrinking because of the accumulation of floating-point errors.
- We use a different ϵ for each smoothing method. $\epsilon = 0.1$ for **Laplacian smoothing** whereas $\epsilon = 0.00003$ for **osculating circle smoothing**. Because the epsilon is really small for the second method, we would recommend increasing the number of iterations executed each time the "C" key is pressed in order to speedup smoothing (20 iterations per key press seems reasonable).
- We find that, to obtain a good end result, it is beneficial to start with **osculating circle smoothing** and to switch to **Laplacian smoothing** once the shape is more or less circle-shaped.