Drawing Shapes with Continuous Functions

D. Demidov

26.04.2025

1 Non-continuous-to-Continuous Approach

The following approach is actually straightforward and works well, although it does not provide a differentiable function in the end.

1.1 Overview of the Algorithm

(i) We take a sequence of points $p_1, ..., p_n$ that we want to be connected such that $p_n = (x_n, y_n)$ and any two points like (p_k, p_{k+1}) as well as (p_n, p_1) are connected.

(ii) In order to construct a function $f : [0,1] \to \mathbb{R}^2$, such that $f\left(\frac{k-1}{n}\right) = p_k$ for all $k \in \{1, ..., n\}$ as well as $f(1) = p_1$, we linearly interpolate between each pair of points, such that $f|_{\left[\frac{k-1}{n}, \frac{k}{n}\right]}$ is a linear function for all $k \in \{1, ..., n\}$. For that sake, we need functions
$\iota_k : [0,1] \to \{0,1\}, t \mapsto \begin{cases} 0 \text{ if } t \in \left[\frac{k-1}{n}, \frac{k}{n}\right] \\ 1 \text{ otherwise} \end{cases}$
in order to define the resulting function as one closed math expression. Assuming we have a step function $s : \mathbb{R} \to \mathbb{R}$ with $s(t) = \begin{cases} 0 \text{ if } t \leq 0 \\ 1 \text{ otherwise} \end{cases}$, we clearly have $\iota_k(t) \coloneqq s\left(t - \frac{k-1}{n}\right) s\left(\frac{k}{n} - t\right)$

1.2 Defining Parts

$$\begin{aligned} &\text{We want } f_k(0) = p_k, f_k(1) = p_{k+1} \\ &\text{so we set } f_k(t) \coloneqq \begin{pmatrix} x_k + t(x_{k+1} - x_k) \\ y_k + t(y_{k+1} - y_k) \end{pmatrix}, k \in \{1, ..., n\} \\ &\quad \text{with } p_{n+1} \coloneqq p_1 \end{aligned} \tag{1}$$

Now we combine these:

$$f(t) \coloneqq \sum_{k=1}^{n} \iota_k(t) f_k(nt - (k-1)) \tag{2}$$

Rewriting results in

$$f(t) = \sum_{k=1}^{n} s\left(t - \frac{k-1}{n}\right) s\left(\frac{k}{n} - t\right) \begin{pmatrix} x_k + (nt - (k-1))(x_{k+1} - x_k) \\ y_k + (nt - (k-1))(y_{k+1} - y_k) \end{pmatrix}. \tag{3}$$

One of the ways to define a step function is $s(t) \coloneqq \frac{x\sqrt{\frac{1}{x^2}}}{2}$, but this expression is undefined for $x = 0$. So we approximate it by setting $s_m(t) \coloneqq \frac{1}{1+e^{-mx}}$, for s_m converges pointwise to s for $x \in \mathbb{R} \setminus \{0\}$ and $s_m(0) = \frac{1}{2}$ for all $m \in \mathbb{N}$. So, for numerical purpuses, it makes sense to choose some $m \in \mathbb{N}$ and set $s \coloneqq s_m$. Then, m would also control the perceived smoothness of the resulting figure. Further, one could define m_k as a parameter for each interval to more precisely control the look of the figure.

1.3 Final Formula

$$f(t) = \sum_{k=1}^{n} \frac{1}{1 + e^{-m\left(t - \frac{k-1}{n}\right)}} \cdot \frac{1}{1 + e^{-m\left(\frac{k}{n} - t\right)}} \begin{pmatrix} x_k + (nt - (k-1))(x_{k+1} - x_k) \\ y_k + (nt - (k-1))(y_{k+1} - y_k) \end{pmatrix}. \tag{4}$$