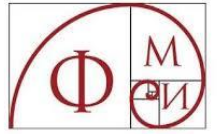




Софийски университет „Св. Кл. Охридски“
Факултет по математика и информатика



Курсов проект

Реализиране на алгоритъма crossplot за крива на Bezier

Курс: Компютърно геометрично моделиране,

Лектор: доц. Красимира Влъчкова

Зимен семестър, 2024/2025 год.

Изготвен от: Димитър Аврамов

Специалност: Софтуерно инженерство, IV курс

1. Задача

При зададени от потребителя точки, да се начертае кривата на Bezier, след което да се начертае crossplot-а на дадената крива.

2. Математическо описание и имплементация

- За получаване кривата на Bezier е използван алгоритъма на de Casteljau:

Algorithm 1 *de Casteljau*

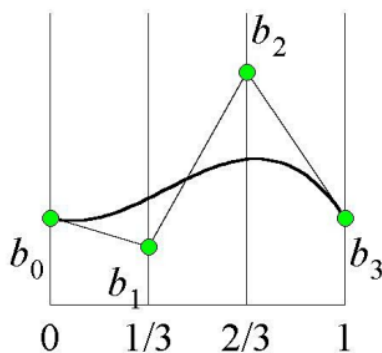
Вход: $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^3, t \in \mathbb{R}$

$$\begin{aligned}\mathbf{b}_i^r(t) &= (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t), \\ r &= 1, \dots, n; i = 0, \dots, n-r \\ \mathbf{b}_i^0 &= \mathbf{b}_i\end{aligned}$$

Изход: $\mathbf{b}_0^n(t)$ е точка от кривата \mathcal{B} , съответстваща на параметъра t .

Основните функции, които отговарят за намирането на кривата, са *calculate_bezier_curve* и *de_casteljau_algorithm*. Първата функция изчислява всички точки, които формират една крива на Bezier. За всяка стойност на параметъра t , който е в интервала $[0, 1]$ и се увеличава с 0.01 на всяка итерация, функцията извиква *de_casteljau_algorithm*. Функцията *de_casteljau_algorithm* реализира алгоритъма на de Casteljau за изчисляване на една точка от кривата при дадена стойност на t . Алгоритъмът започва с даден списък от контролни точки и ги редуцира итеративно чрез линейна интерполация между съседни точки, докато остане само една точка.

- Crossplot алгоритъм за крива



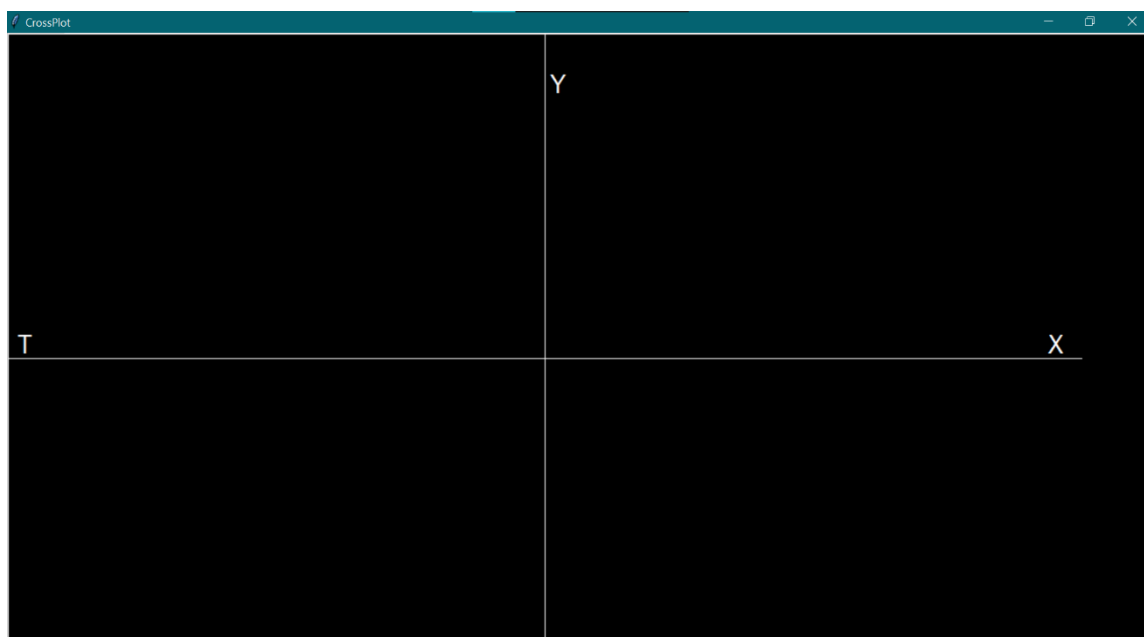
Фигура 33: Кубична функция на Bézier

откъдето следва, че контролния полигон на функцията $f(t)$ е с върхове точките $(j/n, b_j)$, $j = 0, \dots, n$. За да има пълна яснота, ще наричаме $f(t)$ **функция на Bézier**, а b_j , $j = 0, \dots, n$ - **ординати на Bézier**. На фиг. 33 е показана кубична функция на Bézier, като абсцисите ѝ са j/n , $j = 0, \dots, 3$, а ординатите ѝ са числата b_j , а $j = 0, \dots, 3$. Поради инвариантността при афинните преобразувания на кривите на Bézier може да разглеждаме произволен интервал $[a, b]$ вместо $[0, 1]$. Тогава абсцисите на контролните точки са $a + \frac{j(b-a)}{n}$, $j = 0, \dots, n$.

Основните функции, използвани за намиране на crossplot на кривата са *compute_x_functional_points* и *compute_y_functional_points*. При първата функция за всяка контролна точка на функционалната крива стойността на x координата е същата като тази на i-тата контролна точка от кривата на Bezier, а y координатата, принадлежаща на интервала $[h/2, h]$ намираме чрез формулата $h/2 + (i+1) * (h/2) / (n+1)$, където n е броят точки, h е височината на прозореца, а i е индекса на съответната точка. Логиката е аналогична и за другата функция *compute_y_functional_points*, но там за всяка контролна точка на функционалната крива стойността на y координата е същата като тази на i-тата контролна точка от кривата на Bezier, а x координатата, принадлежаща на интервала $[0, w/2]$ изчисляваме по следния начин: $w/2 - (i+1) * (w/2) / (n+1)$, където w е ширината на прозореца.

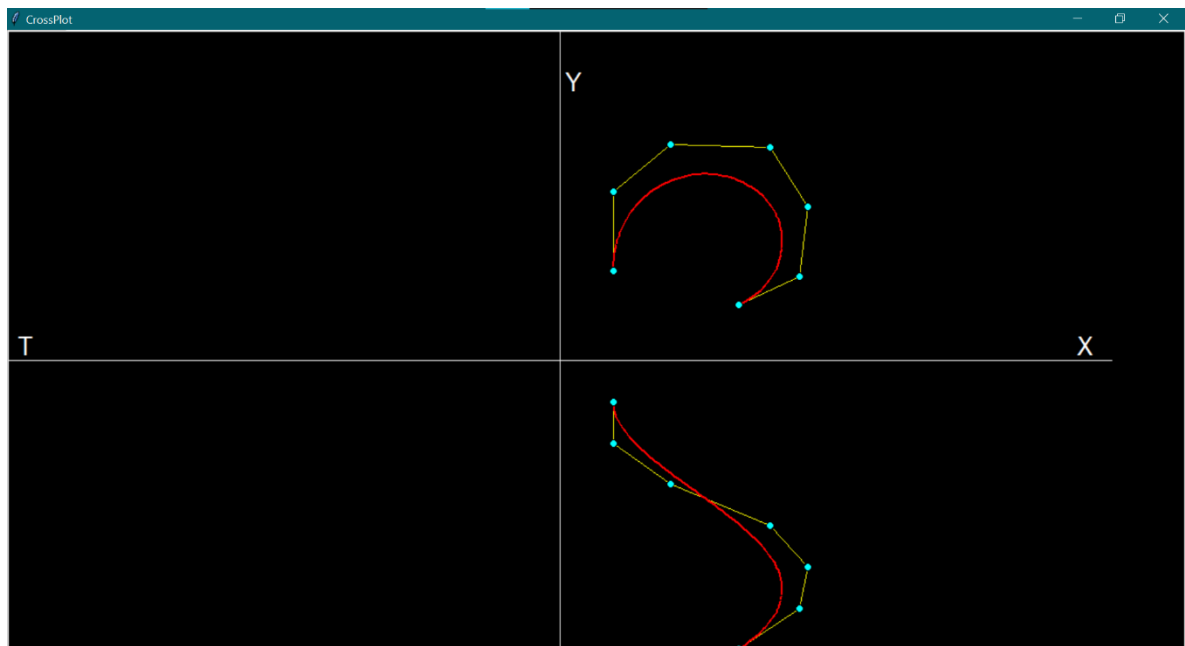
3. Потребителско упътване и функционалности

При стартиране на програмата се зарежда декартовата координатна система в следния вид:

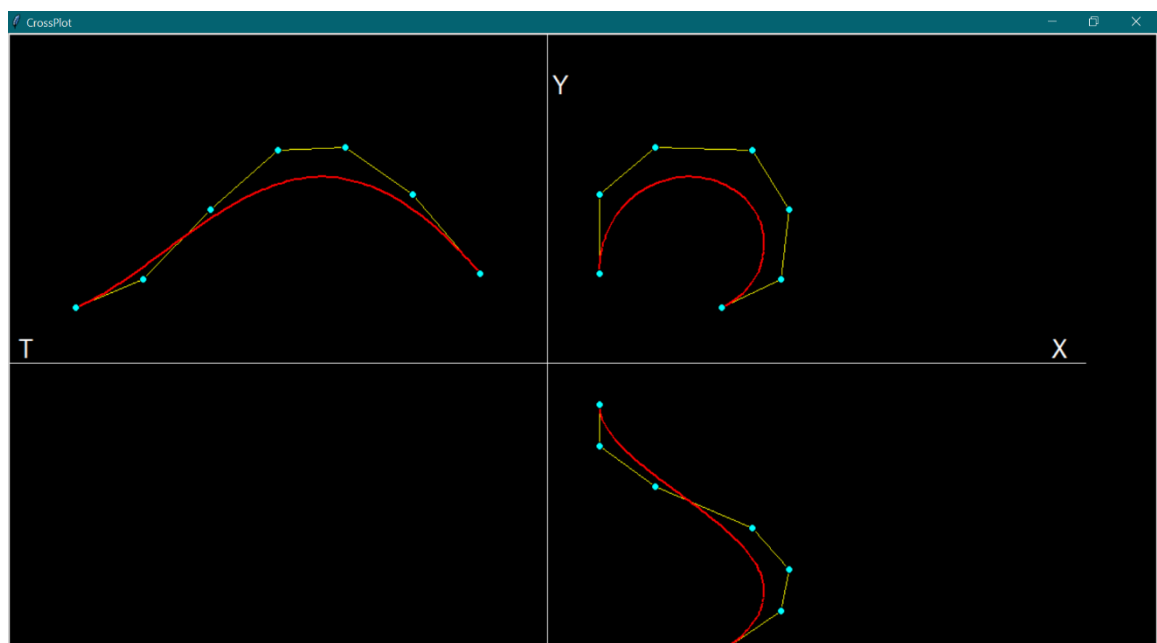


Потребителят въвежда нови точки чрез кликания с ляв бутон на мишката в първи квадрант, а във втори и четвърти, при желание, могат да се визуализират съответно $y(t)$ функцията и $x(t)$ функцията. Трети квадрант стои празен през цялото време. Програмата предоставя следните възможности на потребителя:

- **Добавяне на нова контролна точка** чрез натискане на левия бутон на мишката върху първи квадрант. Новата точка се добавя на мястото на курсора в момента на натискането, като след всяко добавяне на точка се изчертава и обновената крива на Bezier.
- **Изтриване на последната добавена точка** чрез натискане на десния бутон на мишката върху първи квадрант, след което отново се изчертава кривата на Bezier с обновените точки.
- **Визуализиране и скриване на $x(t)$ функцията** чрез натискане на бутона 'x' от клавиатурата. При първо натискане функцията се визуализира, а при повторно се скрива, след което при последващи натискания процеса на визуализиране/скриване се повтаря. При едно примерно изпълнение на програмата, при натискане на бутона 'x', екрана би изглеждал по следния начин:



- **Визуализиране и скриване на $y(t)$ функцията** чрез натискане на бутона ,у‘ от клавиатурата. При първо натискане функцията се визуализира, а при повторно се скрива, след което при последващи натискания процеса на визуализиране/скриване се повтаря. При едно примерно изпълнение на програмата, при натискане на бутона ,у‘, екрана би изглеждал по следния начин:



4. Използвани технологии

- Програмен език Python
- Библиотека Tkinter

5. Използвана литература

- Записки от лекциите към курса „Компютърно геометрично моделиране“
- Tkinter документация: <https://docs.python.org/3/library/tk.html>