

Jakub EISENBERGER<sup>1</sup>, Piotr KUBALA<sup>2</sup>

Opiekun naukowy: Sławomir HERMA<sup>3</sup>, Dawid KOTRYS<sup>4</sup>

## OPROGRAMOWANIE WSPOMAGAJĄCE SYMULOWANIE KRZYWYCH INŻYNIERSKICH

**Streszczenie:** Artykuł ten opisuje działanie Programu, napisanego w języku C++, który przyjmuje parametry funkcji krzywych inżynierskich, wylicza określoną liczbę punktów łamanej przybliżającej krzywą. Następnie program renderuje łamaną za pomocą biblioteki SFML oraz oferuje możliwość zapisania jej i użycia w programie AutoCAD.

**Słowa kluczowe:** oprogramowanie, program, krzywe, inżynierskie, symulowanie, prosta, okrąg, stożkowa, cykloida, ewolwenta, łańcuchowa, epitrochoida, parabola, klotoida, śrubowa, SFML, C++, AutoCad

## SOFTWARE SUPPORTING THE SIMULATION OF ENGINEERING CURVES

**Summary:** This article explains a Program, written in the C++ programming language, which calculates the points of a polyline which approximates an engineering curve of the user's choice. Subsequently the Program renders the polyline using the SFML library and offers the user the ability of saving it to an AutoCAD script.

**Keywords:** software, program, engineering, curve, simulation, line, circle, conic, cycloid, involute, catenary, epitrochoid, parabola, clothoid, helix, SFML, C++, AutoCAD

### 1. Wstęp

Krzywe inżynierskie są przydatnym narzędziem przy tworzeniu rozmaitych projektów. Pomimo to, AutoCAD nie posiada wbudowanej funkcji ich generowania. Celem powstania Programu jest zapewnienie użytkownikom AutoCAD- a możliwości generowania krzywych inżynierskich oraz wstawiania ich do nowych, bądź też istniejących projektów. Program umożliwia generowanie: prostej, okręgu, krzywej

---

<sup>1</sup> V Liceum Ogólnokształcące w Bielsku-Białej, jakubeisenberger@icloud.com

<sup>2</sup> V Liceum Ogólnokształcące w Bielsku-Białej, piotr.kubala@o2.pl

<sup>3</sup> dr inż., Katedra Inżynierii Produkcji, WBMiI ATH, sherma@ath.bielsko.pl

<sup>4</sup> dr , Katedra Matematyki, WBMiI ATH, dkotrys@ath.bielsko.pl

stożkowej, cykloidy, ewolwenty, krzywej łańcuchowej, epitrochoidy, paraboli, klotoidy oraz linii śrubowej. Oprogramowanie generuje pliki .scr, zawierające współrzędne punktów łamanej możliwej do zaimportowania do projektu programu AutoCAD za pomocą komendy SCRIPT. Program oparty jest na prostym i intuicyjnym interfejsie tekstowym. Do renderowania krzywych używana jest biblioteka SFML<sup>5</sup>. Program napisany jest w języku C++.

## 2. Autorski opis interfejsu użytkownika

Po włączeniu programu, użytkownikowi zostaje przedstawiona lista dostępnych krzywych wraz z ich liczbami porządkowymi. Użytkownik wprowadza nazwę, wyświetlaną w kwadratowych nawiasach, lub liczbę odpowiadającą wybranej krzywej.

```
-----  
Dostępne krzywe:  
1. [prosta]  
2. [okrag]  
3. krzywa [stożkowa]  
4. [cykloida]  
5. [ewolwenta]  
6. krzywa [łańcuchowa]  
7. [epitrochoida]  
8. [parabola]  
9. [klotoida]  
10. linia [śrubowa]  
-----  
Podaj krzywa: klotoida
```

*Rysunek 1. Interfejs wyboru krzywej*

Następnie Program wyświetla wzór krzywej oraz zapytania o poszczególne parametry.

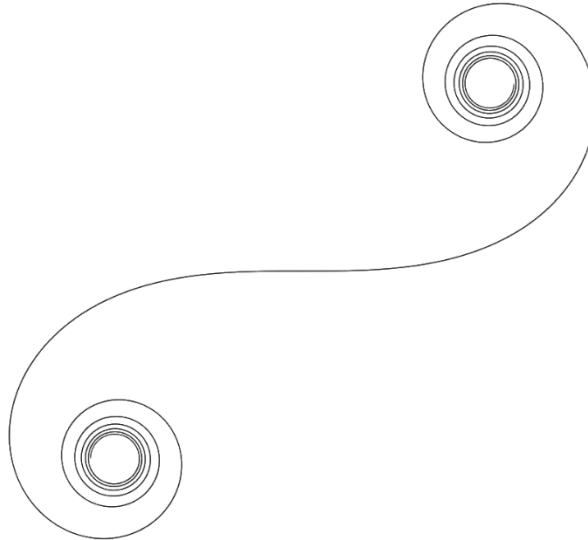
```
Podaj krzywa: klotoida  
(fresnel_cos(t) * a + xs, fresnel_sin(t) * a + ys)  
Podaj początkowe t: -5  
Podaj końcowe t: 5  
Podaj ilość punktów: 100000  
Podaj a: 1  
Podaj xs: 0  
Podaj ys: 0
```

*Rysunek 2. Wzór klotoidy*

Po podaniu wszystkich wymaganych parametrów Program wyświetla w oddzielnym oknie *render* krzywej.

---

<sup>5</sup> <https://www.sfm1-dev.org/>



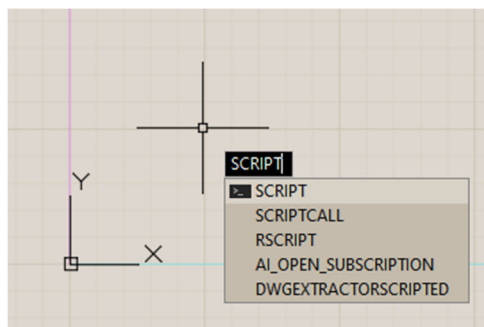
Rysunek 3. Render klotoidy

Po wyłączeniu podglądu prostej Program wyświetla zapytanie, czy zapisać punkty należące do łamanej przybliżającej krzywą do pliku. Użytkownik jest pytany o nazwę pod którą plik ma zostać zapisany w formacie .scr. Następnie użytkownikowi przedstawiona jest opcja zakończenia działania Programu lub wygenerowania kolejnej krzywej.

```
Zapisac do pliku? (t/n): t  
Podaj nazwe pliku: klotoida  
Zakonczyzc? (t/n): t
```

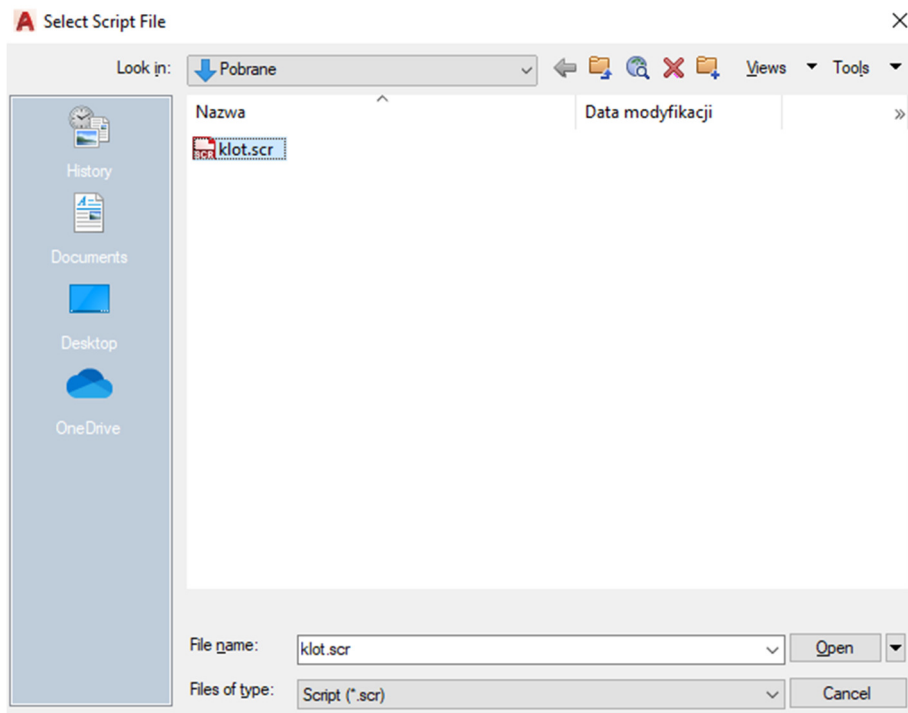
Rysunek 4. Interfejs zapisu pliku

Plik .scr wygenerowany przez Program może zostać wczytany do programu AutoCAD za pomocą funkcji SCRIPT.



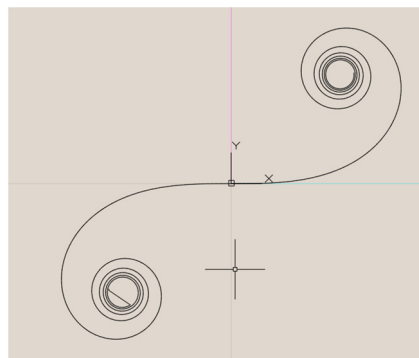
Rysunek 5. Komenda SCRIPT w programie AutoCAD

Użytkownik proszony jest o wybranie pliku .scr.



*Rysunek 6. Interfejs wczytywania pliku*

AutoCAD przyjmuje plik .scr i po chwili generuje odpowiednią krzywą. Czas importowania jest wprost proporcjonalny do ilości punktów krzywej.



*Rysunek 7. Krzywa zaimportowana z pliku .scr*

### 3. Opis działania programu

#### Zasada działania programu

Działanie programu opiera się o generowanie łamanej, która przybliży krzywą dla pewnego skończonego zbioru rosnących parametrów  $t$ , w taki sposób, że punkty będące wierzchołkami łamanej należą do krzywej. Użytkownik podejmuje decyzję na temat ilości punktów użytych do tej interpolacji; umożliwia to zachowanie odpowiedniej wydajności oraz precyzji w zależności od jego potrzeb.

#### 3.1. Sposób generowania łamanych

Program został napisany w języku C++, z wykorzystaniem prostej biblioteki SFML, umożliwiającej rysowanie prostych kształtów oraz udostępniającej interfejs obiektowy do wygodnej obsługi obiektów geometrycznych. Najważniejsze z nich będzie klasa `Vector2f` oraz `Vector3f`, umożliwiające tworzenie wektorów odpowiednio dwu i trójelementowy oraz wykonywanie na nich działań dodawania i mnożenia przez skalar.

W funkcji renderującej została wykorzystana funkcja przyjmująca jako argument tablicę obiektów klasy `Vertex`. Obiekty tej klasy posiadają pole przechowujące pozycję punktu i jego kolor, który w Programie pozostał domyślny.

Przykładowy kod generujący łamaną:

```
std::vector <sf::Vertex> okrag(
    double r,
    double xs,
    double ys,
    double t0,
    double tk,
    int n)
{
    std::vector <sf::Vertex> line;
    float dt = (tk - t0) / n, t;

    line.push_back(
        sf::Vertex(
            sf::Vector2f(cos(t0) * r + xs, sin(t0) * r + ys)));
    line.push_back(
        sf::Vertex(
            sf::Vector2f(
                line[0].position.x, line[0].position.y)));

    for(int i = 2; i <= n + 2; i++)
    {
        t = t0 + dt * i;
        line.push_back(
            sf::Vertex(
                sf::Vector2f(cos(t) * r + xs,
                    sin(t) * r + ys)));

        line[0].position.x=std::min(line[0].position.x,
            line[i].position.x);
    }
}
```

```

        line[0].position.y=std::min(line[0].position.y,
        line[i].position.y);
        line[1].position.x=std::max(line[1].position.x,
        line[i].position.x);
        line[1].position.y=std::max(line[1].position.y,
        line[i].position.y);
    }
    return line;
}

```

Funkcja jako argument przyjmuje parametry krzywej, w przypadku okręgu jest to jego promień oraz składowe wektora  $x_s$ ,  $y_s$ , o który krzywa zostanie przesunięta, początkowa i końcowa wartość parametru  $t$  oraz liczba punktów  $n$ . Jako wartość zwraca *vector* obiektów typu *Vertex*.

Na początku obliczany jest przyrost parametru  $t$ , zapisywany do zmiennej  $dt$ , można go obliczyć z równania

$$dt = \frac{t_k - t_0}{n}. \quad (1)$$

Dwa pierwsze elementy zwracanego wektora będą przechowywane dwa wektory, których składowe będą odpowiednio najmniejszą współrzędną  $x$  i  $y$  oraz największą wartością  $x$  i  $y$  spośród współrzędnych punktów należących do łamanej. Wartości te zostaną wykorzystane w czasie renderowania.

Następnie nastąpi  $n$  iteracji pętli *for*, w której do wektora zostanie dodane  $n$  punktów należących do krzywej będących wierzchołkami łamanej, których współrzędne zostały obliczone z równania parametrycznego krzywej oraz nastąpi aktualizacja wartości minimalnych i maksymalnych przechowywanych pod indeksami 0 i 1 wektora.

### 3.2. Renderowanie

Główna funkcja renderująca jako argument przyjmuje *vector* zwracany przez funkcję generującą łamaną, oraz łańcuch znaków będący jej nazwą, która zostanie wyświetlona na górnym pasku okna.

Następnie rozpoczyna się iteracja, której początkiem jest skopiowanie wektora a następnie poddanie go transformacjom skalowania i translacji wykonywanych przez funkcję *transform*.

```

#define WSPOLCZYNNIK_SKALOWANIA 0.9f

void transform(std::vector<sf::Vertex> &fig, float x, float y)
{
    sf::Vector2f skale(x / 2, y / 2);

    sf::Vector2f srodek(
        (fig[0].position.x+fig[1].position.x)/2,
        (fig[0].position.y+fig[1].position.y)/2);

    float min_wsp=std::min(
        skale.x/(fig[1].position.x- srodek.x),
        skale.y/(fig[1].position.y-srodek.y));

    for(int i = 2; i < fig.size(); i++)

```

```

    {
        fig[i].position-=srodek;
        fig[i].position*=-
WSPOLCZYNNIK_SKALOWANIA*min_wsp;
        fig[i].position.x*=-1;
    }
}

```

Funkcja *transform* przyjmuje jako argumenty *vector* reprezentujący łamaną oraz dwie zmienne typu *float* *x* i *y* przechowujące wymiary obszaru renderowania. Funkcja przekształca krzywą w taki sposób, żeby w czasie renderowania została narysowana w całości, jest odpowiednio przesunięta pomniejszona lub powiększona w taki sposób, żeby szerokość krzywej była równa części szerokości ekranu określonej w *WSPOLCZYNNIK\_SKALOWANIA* lub wysokość była równa takiej wartości, w taki sposób, żeby cała łamana była widoczna.

Na końcu przekształcona łamana jest renderowana.

### 3.3. Interfejs użytkownika i zapisywanie do pliku

Interfejs użytkownika wyświetlany jest przez funkcję *main* za pomocą standardowych strumieni *cin* oraz *cout*. Dane użytkownika przechodzą przez rząd instrukcji warunkowych, których celem jest sprawdzenie, która krzywa została wybrana. Następnie użytkownikowi wyświetlane są zapytania o poszczególne parametry funkcji, co opisane zostało w rozdziale 2.

```

std::cout << "-----\n\tDostepne
krzywe:\n\t1.      [prosta]\n\t2.      [okrag]\n\t3.      krzywa
[stozkowa]\n\t4.  [cykloida]\n\t5.  [ewolwenta]\n\t6.  krzywa
[lancuchowa]\n\t7.  [epitrochoida]\n\t8.  [parabola]\n\t9.
[klotoida]\n\t10. linia [srubowa]\n-----\n";
std::cout << "Podaj krzywa: ";
std::cin >> krzywa;

if(krzywa == "prosta" || krzywa == "1")
{
...
}

```

Zapisywanie do pliku odbywa się za pomocą strumienia *fstream*. Metoda zapisywania jest różna dla krzywych dwuwymiarowych i trójwymiarowych.

```

if(rodzaj != 0)
{
    std::cout << "Zapisac do pliku? (t/n): ";
    std::cin >> zapisac;
    if (zapisac == 't')
    {
        std::cout << "Podaj nazwe pliku: ";
        std::cin >> nazwa;
        nazwa += ".scr";
        plik.open(nazwa, std::ios::out);
        if(rodzaj == 1)

```

```
{
    plik << "LINE\n";
    for (int i = 2; i < punkty.size(); i++)
    {
        plik << punkty[i].position.x << ',' <<
        punkty[i].position.y << '\n';
    }
}
else
{
    plik << "LINE\n";
    for (int i = 0; i < punkty3D.size(); i++)
    {
        plik << punkty3D[i].x << ',' << punkty3D[i].y << ','
        << punkty3D[i].z << '\n';
    }
}
plik.close();}
```

Plik, tworzony przez Program, jest plikiem .scr. Jest to plik poleceń programu AutoCAD. Plik zawiera w sobie polecenie LINE oraz współrzędne punktów, przez które ta linia ma być poprowadzone. W przypadku krzywych dwuwymiarowych, są to dwie współrzędne na punkt, w przypadku trójwymiarowych – trzy.

## 4. Podsumowanie

Zaprezentowane algorytmy generowania krzywych inżynierskich można wykorzystać w sposób praktyczny w wielu obszarach, zbieżnych z obszarami zastosowania opisanych tu krzywych.

### 4.1. Klotoida

Klotoida znajduje zastosowanie w projektowaniu dróg i linii kolejowych. W układzie ciała poruszającego się ruchem jednostajnym wzdłuż tej krzywej istnieje siła odśrodkowa, której wartość jest wprost proporcjonalna do drogi przebytej wzdłuż krzywej. Umożliwia to łączenie fragmentów dróg o różnych krzywiznach.

### 4.2. Krzywa łańcuchowa

Jest to krzywa, której kształt przyjmuje lina zaczepiona w dwóch punktach w jednorodnym polu grawitacyjnym. Ma ona zastosowanie przy badaniu wiszących lin, takich jak przewody elektryczne czy liny metalowe, jest również używana przy projektowaniu stropów budynków.

### 4.3. Linia śrubowa

Używana jest w inżynierii przy projektowaniu śrub, gwintów, sprężyn. Ponadto, cząstka naładowana elektrycznie w jednorodnym polu magnetycznym porusza się po tej krzywej.



#### 4.4. Ewolwenta

Krzywa ta jest powszechnie stosowana przy projektowaniu kół zębatych, zazębien, przekładni, układów transmisji mocy.

#### 4.5. Cykloida

Jest ona stosowana przy projektowaniu mostów i charakteryzujących się najwyższą odpornością na obciążenia dynamiczne. Ponadto skocznie narciarskie projektowane są w taki sposób aby ich kształt był fragmentem tej krzywej, będącym brachistochroną.

### LITERATURA

1. HERMA S.: Krzywe? -To proste!, tyle że krzywe. Bielsko-Biała 2020.
2. Wikipedia : wolna encyklopedia,  
*pl.wikipedia.org/w/index.php?title=Prosta&oldid=60830161*  
2020-11-09 14:32Z
3. Okrąg [online]. Wikipedia : wolna encyklopedia, 2020-03-16 08:52Z  
[dostęp: 2020-11-09 14:33Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Okr%C4%85g&oldid=59085841*
4. Krzywa stożkowa [online]. Wikipedia : wolna encyklopedia, 2019-06-30 18:55Z  
[dostęp: 2020-11-09 14:34Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Krzywa\_sto%C5%BCkowa&oldid=56983634*
5. Parabola (matematyka) [online]. Wikipedia : wolna encyklopedia, 2020-04-19 19:44Z  
[dostęp: 2020-11-09 14:35Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Parabola\_(matematyka)&oldid=59501645*
6. Cykloida [online]. Wikipedia : wolna encyklopedia, 2019-05-04 20:48Z  
[dostęp: 2020-11-09 14:35Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Cykloida&oldid=56599970*
7. Brachistochrona [online]. Wikipedia : wolna encyklopedia, 2018-12-06 21:13Z  
[dostęp: 2020-11-09 14:36Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Brachistochrona&oldid=55218554*
8. Epitrochoida [online]. Wikipedia : wolna encyklopedia, 2019-07-17 09:58Z  
[dostęp: 2020-11-09 14:36Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Epitrochoida&oldid=57087417*
9. Krzywa łańcuchowa [online]. Wikipedia : wolna encyklopedia, 2020-01-28 19:45Z [dostęp: 2020-11-09 14:37Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Krzywa\_%C5%82a%C5%84cuchowa&oldid=58637935*
10. Klotoida [online]. Wikipedia : wolna encyklopedia, 2019-02-07 14:37Z  
[dostęp: 2020-11-09 14:39Z]. Dostępny w Internecie:  
*pl.wikipedia.org/w/index.php?title=Klotoida&oldid=55834475*

11. Ewolwenta [online]. Wikipedia : wolna encyklopedia, 2020-08-15 22:04Z [dostęp: 2020-11-09 14:40Z]. Dostępny w Internecie:  
*[pl.wikipedia.org/w/index.php?title=Ewolwenta&oldid=60626285](http://pl.wikipedia.org/w/index.php?title=Ewolwenta&oldid=60626285)*
12. Linia śrubowa [online]. Wikipedia : wolna encyklopedia, 2019-04-18 06:47Z [dostęp: 2020-11-09 14:41Z]. Dostępny w Internecie:  
*[pl.wikipedia.org/w/index.php?title=Linia\\_%C5%9Brubowa&oldid=56471122](http://pl.wikipedia.org/w/index.php?title=Linia_%C5%9Brubowa&oldid=56471122)*