UNIWERSYTET WROCŁAWSKI

Przybliżanie Okręgu

Pracownia 3.3

Antoni Tomaszewski

27 stycznia 2019

1 OPIS PROBLEMU

Tematem zadania jest konstrukcja algorytmu będącego w stanie wyznaczyć przybliżone równanie okręgu. Dowolny punkt na okręgu można wyznaczyć w zależności od kąta t, wtedy ma on współrzędne $(cos(t), sin(t))$.
Zadanie możemy więc potraktować jako problem obliczenia funkcji parametrycznej $f(t) \simeq (cos(t), sin(t))$. Zauważmy również, że $cos(t) = sin(t + \frac{\pi}{2})$, zatem wystarczy wyznaczyć przybliżenie $sinusa$ a $cosinusa$ można łatwo z niego obliczyć. W pracy zostaną przedstawione różne podejścia do wykonania tego zadania: za pomocą interpolacji Hermite'a, krzywych Beziera oraz naturalnej funkcji sklejanej.

Implementacja algorytmu została wykonana w języku Julia w pliku "program.jl". Funkcje pomocnicze zawarte są w pliku "funkcje.jl". Wykresy zostały wygenerowane korzystając z biblioteki "Plots". Do wygodnego używania wielomianów w splajnach użyta została biblioteka "Polynomials".

2 INTERPOLACJA HERMITE'A

2.1 WPROWADZENIE

Interpolacja Hermite'a jest to często używana metoda numeryczna służąca do znalezienia wielomianu przybliżającego mając dane wartości funkcji w n punktach oraz ich k pierwszych pochodnych (dla każdego punktu k może być różne oraz zakładamy że one istnieją). Metoda ta jest bardzo podobna do interpolacji Newtona z tym że u Newtona dla każdego punktu znana jest jedynie jego wartość a u Hermite'a znamy również pewną ilość pochodnych w tym punkcie. Interpolacja Hermite'a jest więc rozszerzeniem interpolacji Newtona.

Łatwo zauważyć że gdy nie znamy wartości w wielu punktach, ale za to znamy wiele wartości kolejncyh pochodnych w tych punktach Hermite będzie bardzo wygodną i znacznie bardziej dokładną metodą niż Newton. Z tej obserwacji korzystać będzie pierwsza z metod służących konstrukcji okręgu.

2.2 Zastosowanie w konstrukcji okręgu

Konstrukcja fragmentu łuku okręgu na przedziale $[-\alpha, \alpha]$ została opisana w artykule [1]. Punktami w których będziemy interpolować funkcję $cosinus$ (oraz $sinus$) będą krańce przedziału, czyli α i $-\alpha$, chcąc osiągnąć stopień wielomianu $2n+1$ będziemy potrzebować wartości n pierwszych pochodnych w tych punktach.
Główna idea stojąca za algorytmem w nim przestawionym polega na tym, aby dokonać rozwinięcia w szereg Taylora w dwóch punktach : α i $-\alpha$.

$$f(x) = f(\alpha) + f'(\alpha)\frac{x-\alpha}{1!} + f''(\alpha)\frac{(x-\alpha)^2}{2!} + ... \tag{2.1}$$

$$w_n(x) = f(\alpha) + f'(\alpha)\frac{x-\alpha}{1!} + f''(\alpha)\frac{(x-\alpha)^2}{2!} + ... + f^{(n)}(\alpha)\frac{(x-\alpha)^n}{n!} \tag{2.2}$$

$$f(x) = f(-\alpha) + f'(-\alpha)\frac{x+\alpha}{1!} + f''(-\alpha)\frac{(x+\alpha)^2}{2!} + ... \tag{2.3}$$

$$v_n(x) = f(-\alpha) + f'(-\alpha)\frac{x+\alpha}{1!} + f''(-\alpha)\frac{(x+\alpha)^2}{2!} + ... + f^{(n)}(-\alpha)\frac{(x+\alpha)^n}{n!} \tag{2.4}$$

Następnie obliczyć wartości w punktach krańcowych tych funkcji oraz wartości pierwszych n pochodnych tych funkcji.
Chcemy więc, by zachodziły nastęujące równości:

$$\begin{gathered} w_n(\alpha) = f(\alpha) \\ w_n'(\alpha) = f'(\alpha) \\ ... \\ w_n^{(n)}(\alpha) = f^{(n)}(\alpha) \end{gathered}$$

oraz

$$\begin{gathered} v_n(\alpha) = f(\alpha) \\ v_n'(\alpha) = f'(\alpha) \\ ... \\ v_n^{(n)}(\alpha) = f^{(n)}(\alpha) \end{gathered}$$

Po przejściu przez stertę obliczeń otrzymujemy następujące wzory

$$C_{2n+1}(\alpha, t) = (cos\alpha + 2\sum_{k=1}^{n} x_k(t^2-\alpha^2)^k, \frac{sin\alpha}{\alpha}t + 2t\sum_{k=1}^{n} y_k(t^2-\alpha^2)^k)$$

gdzie

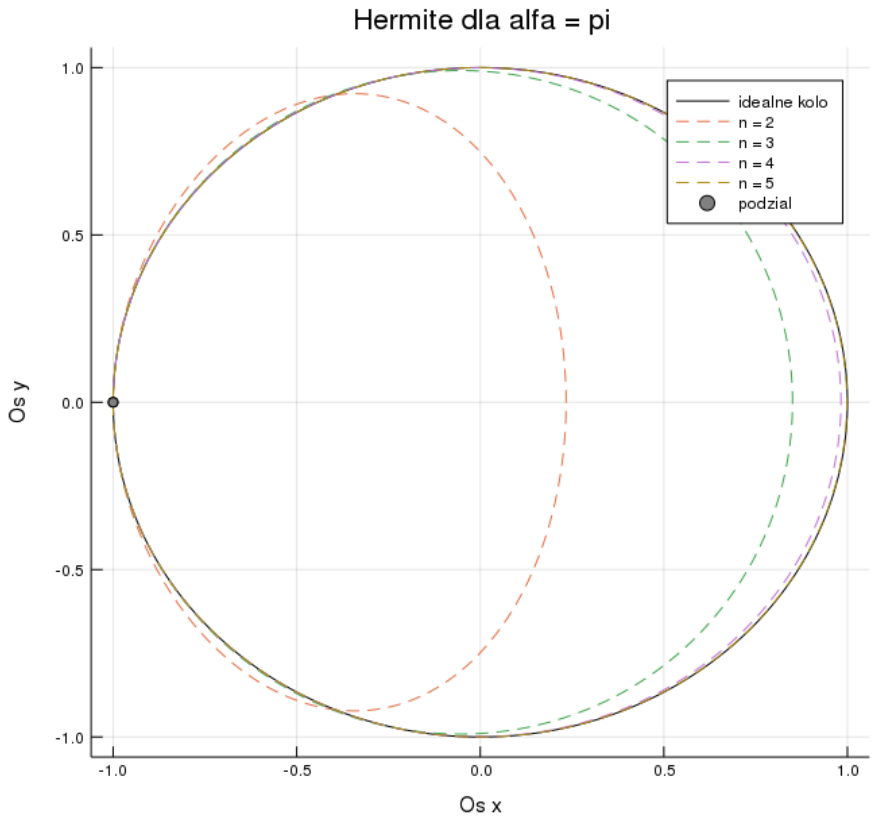
$$x_k = \frac{1}{k!}\sum_{i=0}^{k-1}\frac{(k+i-1)!}{i!(k-i-1)!}\frac{cos(\frac{k+i}{2}\pi+\alpha)}{(2\alpha)^{k+i+1}}$$

oraz

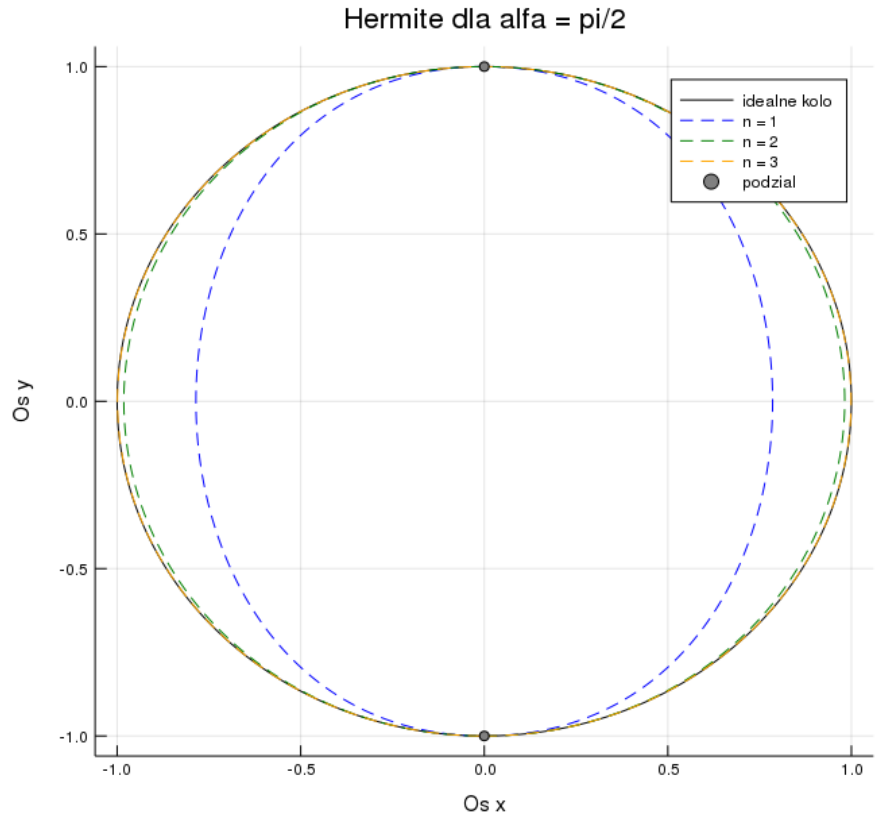
$$y_k = -2\alpha(k+1)x_{k+1}$$

Zauważmy, że mając fragment łuku postaci $[(x_1,y_1),(x_2,y_2),...,(x_k,y_k)]$ możemy odbić go względem prostej przechodzącej przez punkty (x_1,y_1) oraz $(0,0)$ i otrzymać tym sposobem dwukrotnie dłuższy łuk. Z tej obserwacji można skorzystać i uzyskać zadowalające wyniki dla niewielkich stopni wielomianów.

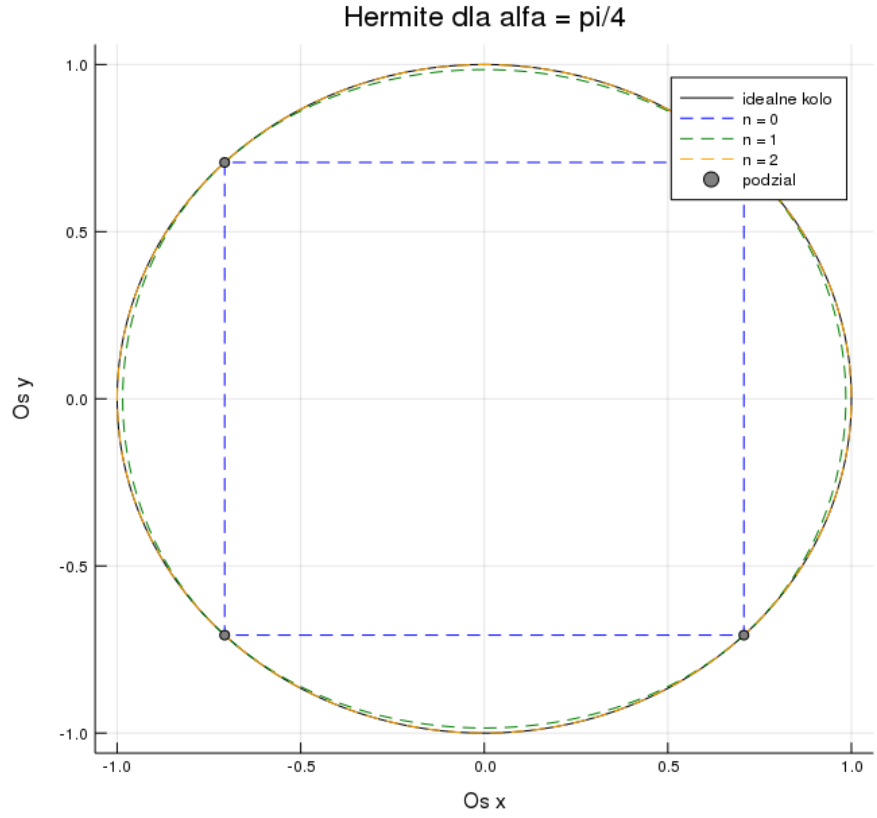
2.3 Wyniki



Rysunek 2.1: Okrąg za pomocą hermite'a przy przedziale interpolacji $[\alpha,-\alpha]$, 0 odbić



Rysunek 2.2: Okrąg za pomocą hermite'a przy przedziale interpolacji $[\frac{\alpha}{2}, \frac{-\alpha}{2}]$, 1 odbicie



Rysunek 2.3: Okrąg za pomocą hermite'a przy przedziale interpolacji $[\frac{\alpha}{4}, \frac{-\alpha}{4}]$, 2 odbicia

Zauważmy, że na ”oko” porównywalne wydają się wyniki kolejnych rysunków dla $n = 4$, $n = 2$, $n = 1$.

3 Krzywe Beziera

3.1 Wprowadzenie

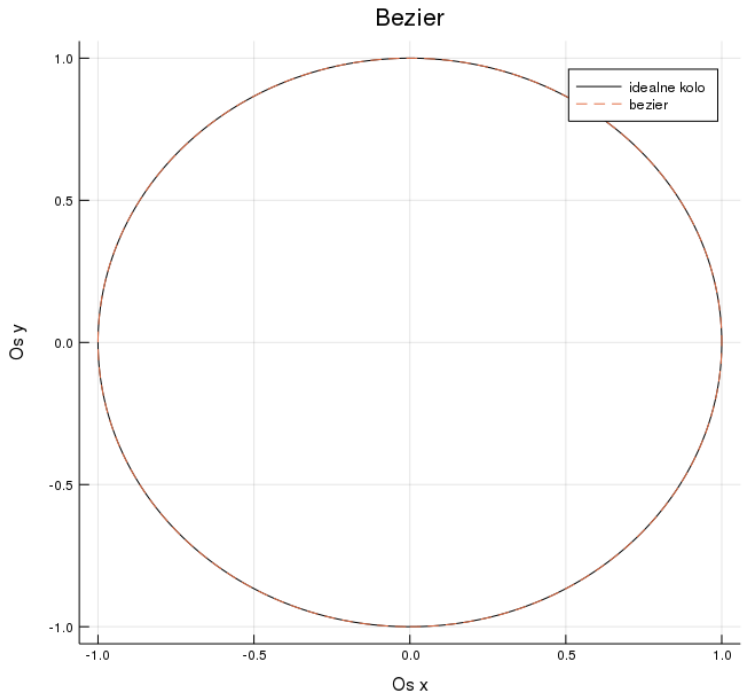
Krzywe Beziera to częsta i dosyć prosta technika wykorzystywana w grafice komputerowej. Polega na wybraniu $n+1$ punktów $(P_0, ..., P_n)$ zwanymi punktami kontrolnymi. Funkcja wynikowa będzie przechodzić przez pierwszy i ostatni punkt oraz nie przechodzić przez żaden pozostały. Funkcja takej krzywej dla dowolnej liczby punktów kontrolnych wyraża się wzorem

$$B(t) = (1-t)^n P_0 + \binom{n}{1}(1-t)^{n-1} t P_1 + ... + \binom{n}{n-1}(1-t)\, t^{n-1} P_{n-1} + t^n P_n \tag{3.1}$$

Gdzie wyrażenie przy $k-tym$ punkcie (P_k) to $k-ty$ wielomian Bernsteina stopnia n.

3.2 Zastosowanie w konstrukcji okręgu

Do wyznaczenia okręgu użyjemy wielomianu stopnia 3. W następujących punktach kontrolnych: $[(0,1),(c,1),(1,c),(0,1)]$, gdzie $c = \frac{4}{3}(\sqrt{2}-1)$. Wartość c można wyliczyć z wielomianu $B(t)$ dla punktu $(0.5, 0.5)$, który leży na tej krzywej.



Rysunek 3.1: Wykres okręgu uzyskanego za pomocą krzywych Beziera

4 NATURALNA FUNKCJA SKLEJANA

4.1 WPROWADZENIE

Splajn jest często wykorzystywaną i uniwersalną techniką interpolacji. Jest "odporny" na efekt Rungego oraz działa dla dowolnego wyboru punktów (zarówno równoodległych jak i nie). W miarę zwiększania ilości danych metoda ta staje się bardziej dokładna. Technika polega na stworzeniu, dla każdej pary sąsiednich punktów, wielomianu stopnia co najwyżej trzeciego,w taki sposób aby wykres był gładki oraz ciągły.

Mając $n+1$ punktów $(x_1, y_1), (x_2, y_2)...(x_{n+1}, y_{n+1})$ chcemy na każdym z przedziałów $[x_k, x_{k+1}]$ skonstruować wielomian 3 st.
Niech

$$s_k(x) = A_k + B_k(x - x_k) + C_k(x - x_k)^2 + D_k(x - x_k)^3 \tag{4.1}$$
$$s'_k(x) = B_k + 2C_k(x - x_k) + 3D_k(x - x_k)^2 \tag{4.2}$$
$$s''_k(x) = 2C_k + 6D_k(x - x_k) \tag{4.3}$$
$$\tag{4.4}$$

Chcemy więc mieć n funkcji i dla każdej z nich 4 warunki, co daje w sumie $4n$ warunków.
$n+1$ punktów daje nam $2n$ warunków ($n-1$ podwójnych oraz dwa pojedyncze (brzegowe))
Żądamy również, aby pierwsze oraz drugie pochodne na krańcach przedziałów były sobie równe (kolejne $2(n-1)$ warunków).

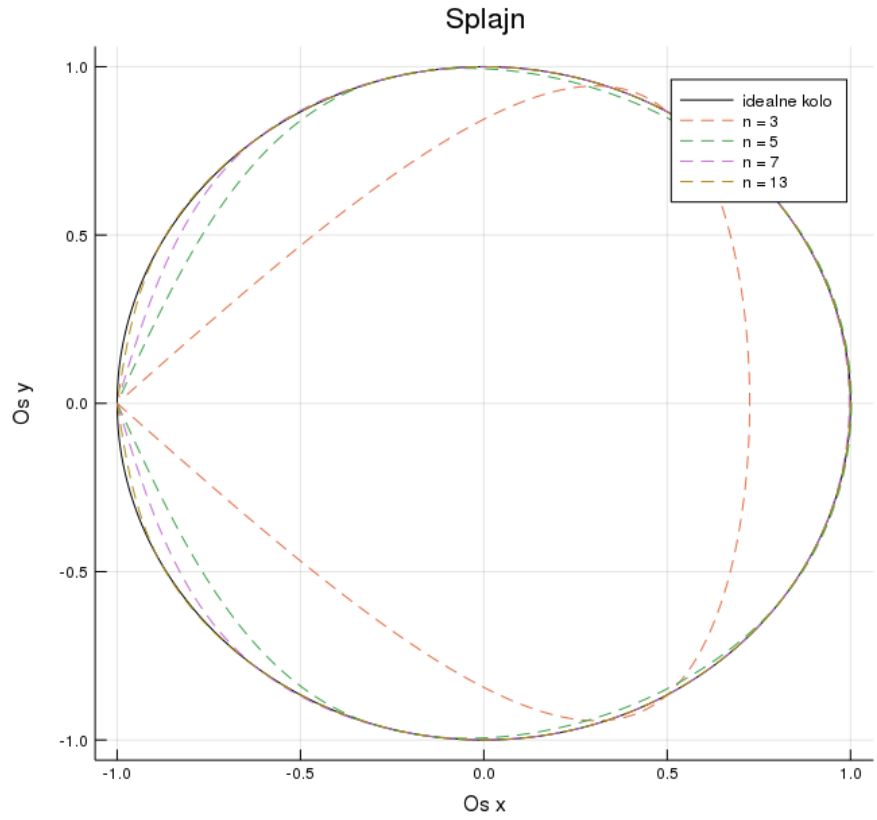
$$s'_{k+1}(x_{k+1}) = s'_k(x_{k+1}) \tag{4.5}$$
$$s''_{k+1}(x_{k+1}) = s''_k(x_{k+1}) \tag{4.6}$$

Brakuje nam jeszcze dwóch warunków, które zapewniamy sobie ustawiając wartości drugich pochodnych w punktach brzegowych na 0.

$$s''_1(x_1) = s''_n(x_{n+1}) = 0 \tag{4.7}$$

4.2 ZASTOSOWANIE W KONSTRUKCJI OKRĘGU

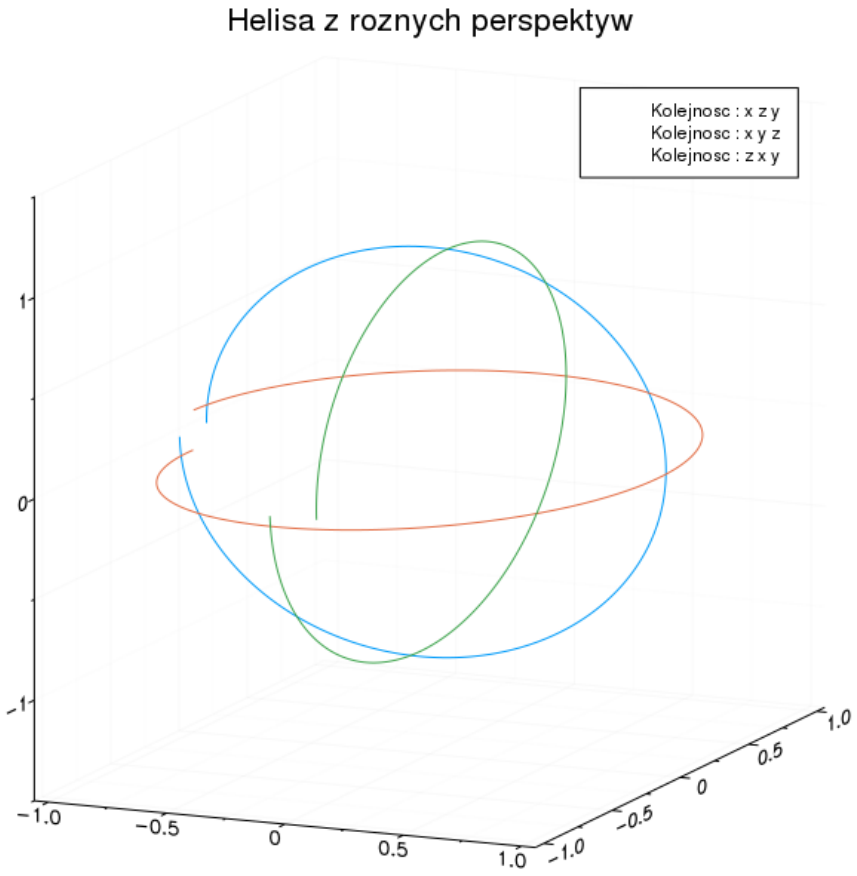
Użycie splajnów do tego zadania wydaje się sensownym pomysłem, ponieważ wraz z zwiększaniem ilości punktów wykres staje się coraz bardziej dokładny, czego nie gwarantują nam interpolacja Netwona ani interpolacja Lagrange'a.



Rysunek 4.1: Wykres Splajna dla punktów równoodległych, gdzie n = liczba węzłów interpolacyjnych

5 Helisa

W artykule [1] jest opisane jak z wykresu koła uzyskać helisę. Możemy ją obliczyć korzystając z każdej z trzech wyżej opisanych metod, a jej dokładność będzie zależeć jedynie od dokładności wcześniej obliczonego wykresu okręgu, dlatego można przy porównaniu tych metod ograniczyć się do zestawienia jedynie dokładności okręgów.
Helisę opisuje wzór: $H(\alpha, t) = (\cos t, \sin t, t),\ t \in [-\pi, \pi]$



Rysunek 5.1: Wykres Helisy za pomocą Hermite'a, różne widoki

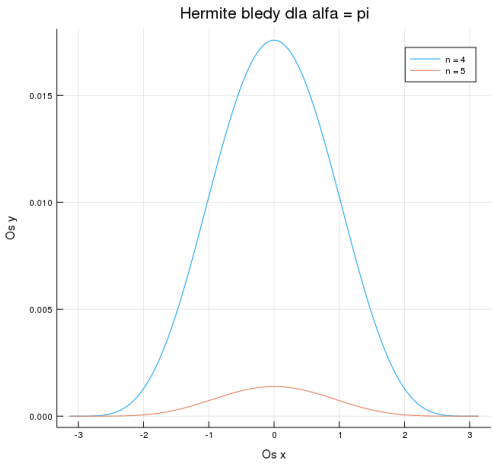
6 Porównanie

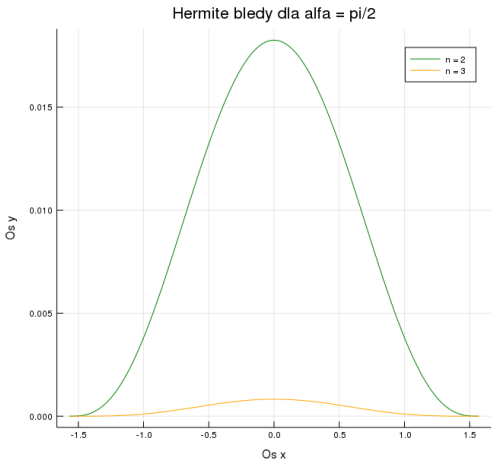
Poniżej przedstawione są błędy, z których wynikają następujące własności:

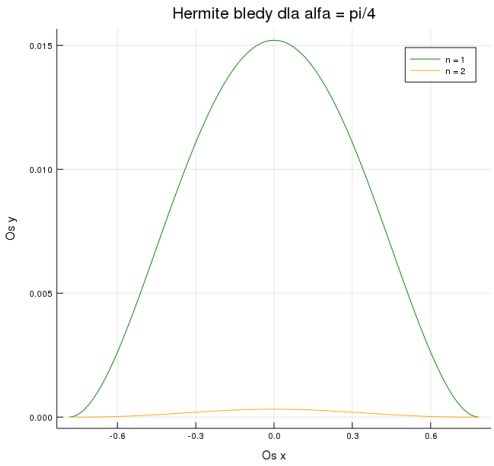
Dla wielomianów stopnia 3 najmniejsze maksimum z błędów ma Bezier, równe 0.007. Wykorzystuje on 2 wielomiany sześcienne, więc używa 8 współczynników.
Maksimum błędów Hermite'a dla $\alpha = \frac{\pi}{4}$, $n = 1$ (czyli 2 wielomianów st. 3) wynosi 0.015, również 8 współczynników
Natomiast splajn osiąga porównywalne maksimum błędu co Bezier dla 17 funkcji sklejanych ($4 * 17 = 68$ współczynników).

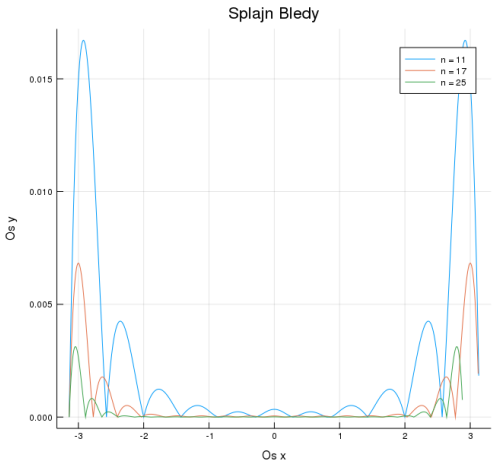
Analizując Hermite'a i splajn, zauważymy że maksimum najgorszego z Hermite'ów czyli $\alpha = \pi$, $n = 4$ jest porównywalne z splajnem dla $n = 11$.
20 współczynników u Hermite'a przeciw 44 w splajnach.

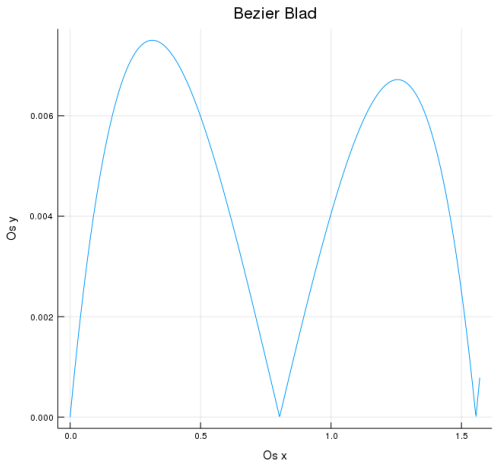
Patrząc wyłącznie na wyniki w splajnach widzimy, że chociaż na większości przedziału błąd jest niewielki, to jednak na krańcach zawsze osiąga całkiem duże wartości.


Hermite bledy dla alfa = pi
n = 4
n = 5
Os y
Os x


Hermite bledy dla alfa = pi/2
n = 2
n = 3
Os y
Os x


Hermite bledy dla alfa = pi/4
n = 1
n = 2
Os y
Os x


Splajn Bledy
n = 11
n = 17
n = 25
Os y
Os x


Bezier Blad
Os y
Os x

7 WNIOSKI

Wybranie odpowiedniej metody powinno zależeć od problemu, jaki mamy rozwiązać. Krzywe Beziera oraz naturalne funkcje sklejane są metodami bardziej ogólnymi niż algorytm opisany w [1], nie powinno stanowić więc zaskoczenia, że radzą sobie one z przybliżeniem okręgu gorzej niż sposób do tego jedynie dedykowany. Jeśli chcielibyśmy użyć jak najmniej danych należałoby skorzystać z krzywych Beziera. Jednak w sytuacji, gdy priorytetem jest precyzja istnieje lepsze rozwiązanie. Jest nim metoda zawarta w artykule [1]. Używa mniej danych, a błąd ma istotnie mniejszy oraz bardziej stabilny niż splajn oraz Bezier.

LITERATURA

[1] Lizheng Lu *On polynomial approximation of circular arcs and helices*,
Department of Mathematics,
Zhejiang Gongshang University,
2011

[2] J.L.López, N.M.Temme, *Two-point Taylor expansions of analytic functions*,
Studies in Applied Mathematics
2002
str. 297-311