Analiza Numeryczna (M) - Pracownia 1 - Zadanie P1.6

Oskar Bujacz

Instytut Informatyki UWr

10 listopada 2019

1. Krótki wstęp

Pojęcie aproksymacji funkcji trygonometrycznych znane jest już od starożytności. Pierwsza tablice wartości trygonometrycznych powstały w II wieku p.n.e i są przypisywane Hipparchusowi. W póżniejszym czasie w Indiach Bhaskara I w VII wieku odkrył wzór

$$sin(x) \approx \frac{4x(180-x)}{40500-x(180-x)}$$

gdzie x to miara kąta w stopniach. Osiągnął on maksymalny bład bezwględny 0.0016. Około 1400 r. również w Indiach, Madhava jako pierwszy użył szeregów nieskończonych i mógł obliczać sinusa z dokładnością do 12 miejsc po przecinku, a cosinusa do 9 miejsc.
Dzisiaj zajmę się problemem aproksymacji funkcji sinus i cosinus trzema różnymi metodami, z użyciem tylko czterech podstawowych działań arytmetycznych tj. $(+,-,*,/)$ z dokładnością bliskiej maszynowej.

2. Uwarunkowanie zadania

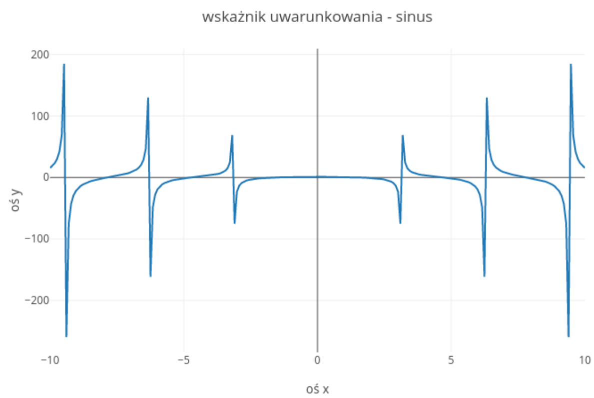
Pojęcie uwarunkowania wiąże się z wrażliwością rozwiązania na małe zaburzenie danych wejściowych. Zadanie będzie w danym punkcie źle uwarunkowane, jeśli wskaźnik uwarunkowania będzię duży.
Korzystając z twierdzenia o wartości średniej możemy zapisać:

$$\frac{f(x+h)-f(x)}{f(x)} \approx \frac{hf'(x)}{f(x)} = \left[\frac{xf'(x)}{f(x)}\right]\left(\frac{h}{h}\right)$$

Wartość bezwględną ilorazu $\frac{xf'(x)}{f(x)}$ jest szukanym wskaźnikiem uwarunkowania i skorzystamy z niego w zadaniu.
Dla funkcji sinus i cosinus wskaźniki, które oznaczymy odpowiednio wsk_{sin} i wsk_{cos} wyglądają nastepująco

$$wsk_{sin} = \left|\frac{xcos(x)}{sin(x)}\right|$$

$$wsk_{cos} = \left|\frac{xsin(x)}{cos(x)}\right|$$



Rysunek 1: wskaźnik uwarunkowania sinus

Rysunek 2: wskaźnik uwarunkowania cosinus

Analizując wskaźniki dla liczb rzeczywistych można łatwo zauważyć, że będą one duże dla wartości w otoczeniu asymptot tangensa i cotangensa. Dla cosinusa to $k\pi$, $k \in Z$ natomiast sinusa są to wartości $\frac{\pi}{2} + k\pi$, $k \in Z \setminus \{0\}$, poniewaz w granicy w zerze $\left|\frac{xcos(x)}{sin(x)}\right| = 1$

Zatem w punktach, które podałem powyżej nie otrzymamy spodziewanej dokładności obliczeń z uwagi na złe uwarunkowanie zadania w ich otoczeniu. Mniejsza precyzja w tych punktach nie będzie zatem świadczyła o niepoprawnym algorytmie.

3. Redukcja kąta do mniejszego przedziału

Pokażę teraz, że możemy rozważać tylko przedział $[\frac{-\pi}{4}, \frac{\pi}{4}]$.
Z okresowości funkcji sinus i cosinus, możemy sprowadzić argument x do przedziału $[-\pi, \pi]$, poprzez modulo 2π. Następnie z uwagi na symetryczność względem osi OX można dalej ograniczyć przedział do $[\frac{-\pi}{2}, \frac{\pi}{2}]$ poprzez operację $\pm\pi$ uwzględniając zmianę znaku na przeciwny. Wreszcie ograniczmy przedział do $[\frac{-\pi}{4}, \frac{\pi}{4}]$.
Dla odpowiednich przedziałów stosujemy odpowiednie wzory.

$$x \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right] \Rightarrow sin(x) = cos(\frac{\pi}{2} - x) \wedge cos(x) = sin(\frac{\pi}{2} - x)$$
$$x \in \left[\frac{-\pi}{2}, \frac{-\pi}{4}\right] \Rightarrow sin(x) = -cos(\frac{\pi}{2} + x) \wedge cos(x) = sin(\frac{\pi}{2} - x)$$

4. Aproksymacja wzorem Taylora

Jak wiadomo, za pomocą wzoru Taylora:

$$f(x) = \sum_{k=0}^{\infty} f^{(k)}(a)\frac{(x-a)^k}{k!}$$

możemy przybliżać funkcje (n)-razy różniczkowalne w punkcie a za pomocą wielomianu k-stopnia podanego powyżej. W szczególnośći, dla funkcji klasy C^∞, możemy przybliżać podaną funkcję z dowolną dokładnością poprzez rozwinięcie w szereg nieskończony.
W zadaniu chcemy otrzymać dokładność bliską maszynowej. Zgodnie ze standardem IEEE 745

podwójnej precyzji liczby zmiennoprzecinkowej, mamy do dyspozycji 64 bity, 1 zajmuje znak, 11 wykładnik i 53 mantysa (1 niejawny). Interesuje nas precyzja, zatem skupimy się na mantysie.
53-bitowa mantysa daje nam od 15 do 17 znaczących cyfr dziesiętnych ($2^{-53} \approx 1.11 \times 10^{-16}$) Zatem, aby otrzymać precyzję maszynową dla każdej liczby, chcielibyśmy, aby różnica między wynikiem, a wartością rzeczywistą był mniejszy od 10^{-18} czyli

$$|f(x) - f_0(x)| \leqslant 10^{-18}$$

gdzie $f_0(x)$ to nasze przybliżenie.

4.1. Analiza szeregu

W tym celu zajmiemy się analizą rozwinięcia w szereg Taylora funkcji sinus i cosinus w punkcie $x = 0$. Rozwijńmy funkcję sinus

$$sin(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)3 + ...$$

obliczając kolejne pochodne sinusa w punkcie $a = 0$ otrzymujemy

$$sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} + ... = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

analogicznie możemy zapisać cosinus

$$cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + ... = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

Sprawdźmy teraz zbieżność szeregu dla sinusa, dla cosinusa będzie analogicznie. Sprawdzimy warunki zbieżności z twierdzenie Leibnitza o szeregu naprzemiennym. Niech (a_n) będzie ciągiem. Sprawdzimy warunki:

- $(a_n) > 0$
- $(a_n) \leqslant (a_n)$
- $\lim_{n\to\infty}(a_n) = 0$

Zakładamy, że $x \in [\frac{-\pi}{4}, \frac{\pi}{4}]$, co mamy z rodziału 3.
Warunek pierwszy jest spełniony $\forall x \in R_+$, w przypadku ujemnego x wystarczy wyciagnąć minus przed sumę.
$|\frac{\pi}{4}| < 1$, zatem drugi warunek jest spełniony, każdy kolejny wyraz jest mniejszy od poprzedniego. Warunek z granicą jest spełniony, gdyż $\lim_{n\to\infty} \frac{x^n}{n!} = 0$. Nasze warunki są spełnione dla każdego n (nr wyrazu szeregu). Na podstawie twierdzenia Leibnitza o szeregu naprzemiennym szeregi przedstawiające sinusa i cosinusa są zbieżne. Wówczas możemy stosować

$$|S - S_n| \leqslant a_{n+1}$$

gdzie $S = sin(x)$, S_n to nasze przybliżenie, a a_{n+1} to kolejny wyraz sumy. Daje nam to maksymalny bład bezwględny.

Mając taką własność możemy znaleźć potrzebną ilość wyrazów szeregu do ustalonej dokładności. Maksymalny bład będzie dla wartości najbardziej oddalonej od zera, czyli $\pm\frac{\pi}{4}$, bez stratu ogólności rozważmy $\frac{\pi}{4}$. Wtedy możemy policzyć, że wyrazem o najmniejszym indeksie spełniającym naszą precyzję tj. 10^{-18} jest wyraz $\frac{x^{19}}{19!}$ gdzie $x = \frac{\pi}{4}$ (dla sinusa) oraz wyraz $\frac{x^{20}}{20!}$ (dla cosinusa). Zatem przy obliczaniu będziemy rozwijać do wyrazu o potędze 17, natomiast przy cosinusie do potęgi 18.

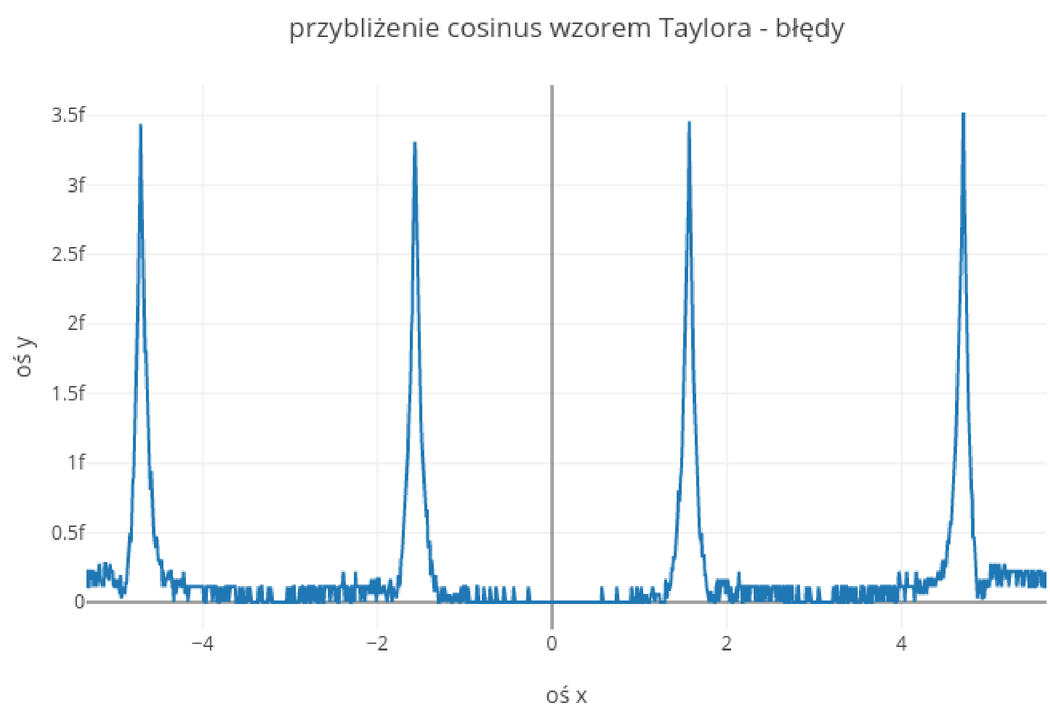
4.2. Przybliżanie wartości

Wiemy już, jak sprowadzić wartość x do przedziału $[\frac{-\pi}{4}, \frac{\pi}{4}]$ oraz ile wyrazów szeregu Taylora potrzebujemy. Skupmy się teraz na minimalizacji błędów obliczeń. Przy klasycznym dodawaniu i obliczaniu kolejnych silni potrzebnych do wyrazów szeregów mamy sporo operacji, co powoduje powstawanie błedów przybliżeń. Można zminimalizować te błędy jak i efektywność poprzez stablicowanie wartości silni, jednak znacznie efektywniejsze będzie zastosowanie schematu Hornera. Będziemy wyłączać przed nawias największe możliwe czynniki. Wtedy otrzymamy

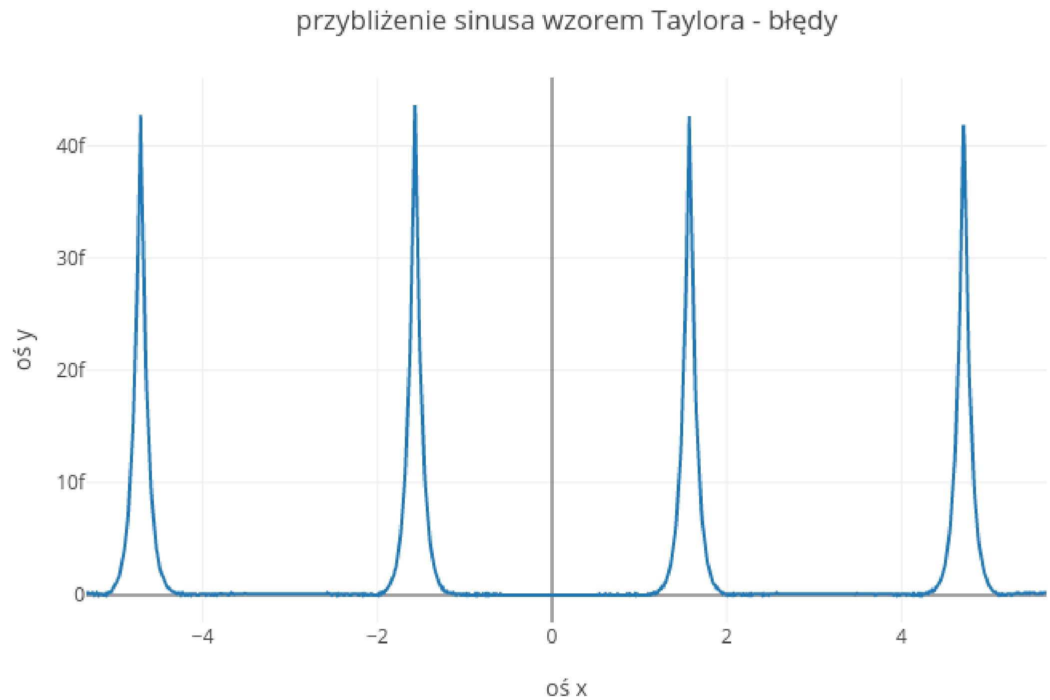
$$sin(x) \approx x(1 - \frac{x^2}{2 \cdot 3}(1 - \frac{x^2}{4 \cdot 5} \ldots (1 - \frac{x^2}{14 \cdot 15} * (1 - \frac{x^2}{16 \cdot 17})) \ldots))$$

Znacząco znimimalizujemy wtedy ilość operacji. Podobnie robimy dla cosinusa.

W kwestii precyzji, będę rozważał błąd bezwględny z uwagi na małe i ograniczone wartości rozważanych funkcji i bardziej wiarygodny przebieg w miejscach zerowych sinusa i cosinusa.



Rysunek 3



Rysunek 4

Okazuje się, że możemy uzyskać dużą dokładność obliczeń, poza punktami gdzie nasza zadanie jest źle uwarunkowane, osiągamy dla obydwóch funkcji dokładność rzędu 10^{-16}, przy czym dla cosinusa dokładność spada do około 10^{-15}, natomiast dla sinusa jest to wartość 10^{-14}

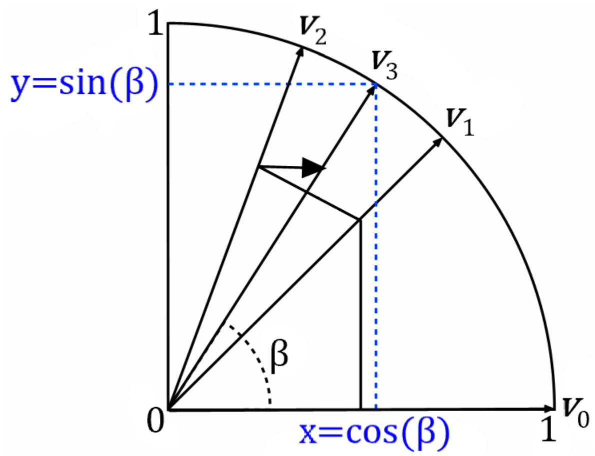
5. Jak to robia języki programowania?

Jedną z metoda obliczania wartości funkcji trygonometrycznych i nie tylko jest metoda CORDIC (COordinate Rotation DIgital Computer). W celu obliczenia wartości sinusa i cosinusa, metoda ta zaczyna od wektora $v_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ i obraca go o $\frac{\pi}{4}$. Następne iteracje obracają wektor w odpowiednią stronę o coraz mniejsze kąty. Krok $i - ty$ ma długość $atan(\frac{1}{2^{i-1}})$. Kolejne wektory v_i są obliczne przez przemnożenie przez macierz obrotu R

$$v_i = R_i v_{i-1}$$

$$R = \begin{pmatrix} cos(\gamma_i) & -sin(\gamma_i) \\ sin(\gamma_i) & cos(\gamma_i) \end{pmatrix}$$

$cos(\alpha) = \frac{1}{\sqrt{1+tan^2(\alpha)}}$ $sin(\alpha) = \frac{tan(\alpha)}{\sqrt{1+tan^2(\alpha)}}$



Rysunek 5: mechanizm algorytmu CORDIC

wprowadzimy odpowiednie wzory trygonometryczne z użyciem arcusa tangensa otrzymamy następująca macierz

$$R = \frac{1}{\sqrt{1+tan^2(\alpha)}} \begin{pmatrix} 1 & -tan(\gamma_i) \\ tan(\gamma_i) & 1 \end{pmatrix}$$

gdzie x_i i y_i to współrzędne wektora v. Otrzymujemy nowy wzór na kolejny obrócony wektor v

$$v_i = \frac{1}{\sqrt{1+tan^2(\alpha)}} \begin{pmatrix} x_{i-1} - y_{i-1}tan(\gamma_i) \\ x_{i-1}tan(\gamma_i) + y_{i-1} \end{pmatrix}$$

Ograniczając wartości kąta γ_i tak, że $tan(\gamma_i)$ przyjmuje tylko argumenty $\pm 2^{-i}$

$$v_i = K_i \begin{pmatrix} x_{i-1} - y_{i-1}2^{-i}\sigma_i \\ x_{i-1}2^{-i}\sigma_i + y_{i-1} \end{pmatrix}$$

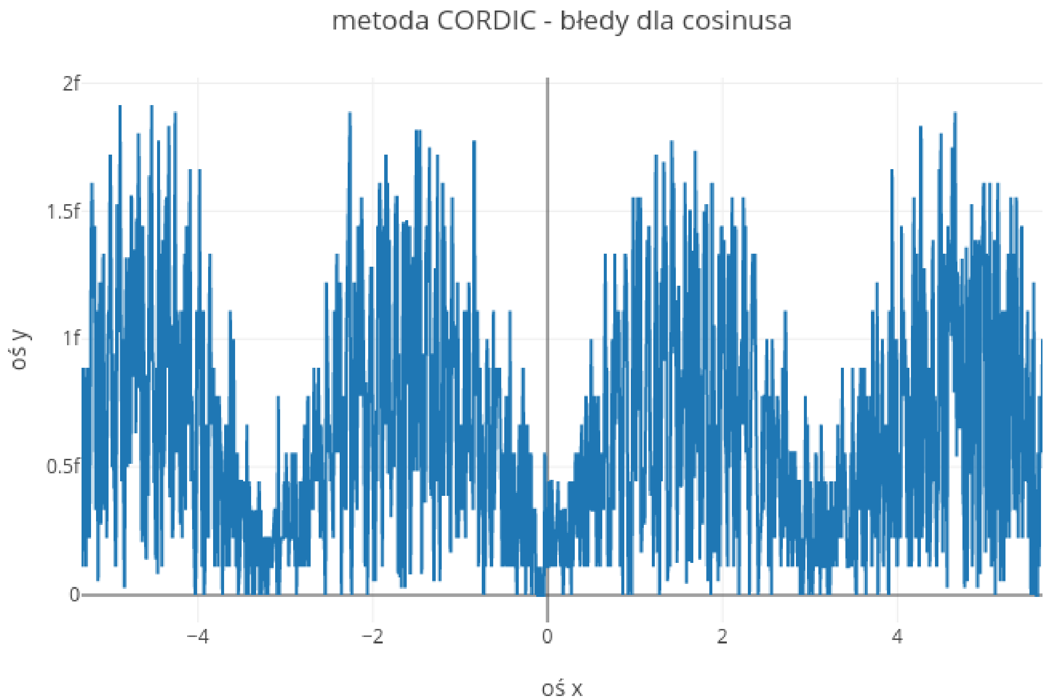
gdzie

$$K_i = \frac{1}{\sqrt{1+2^{-2i}}}$$

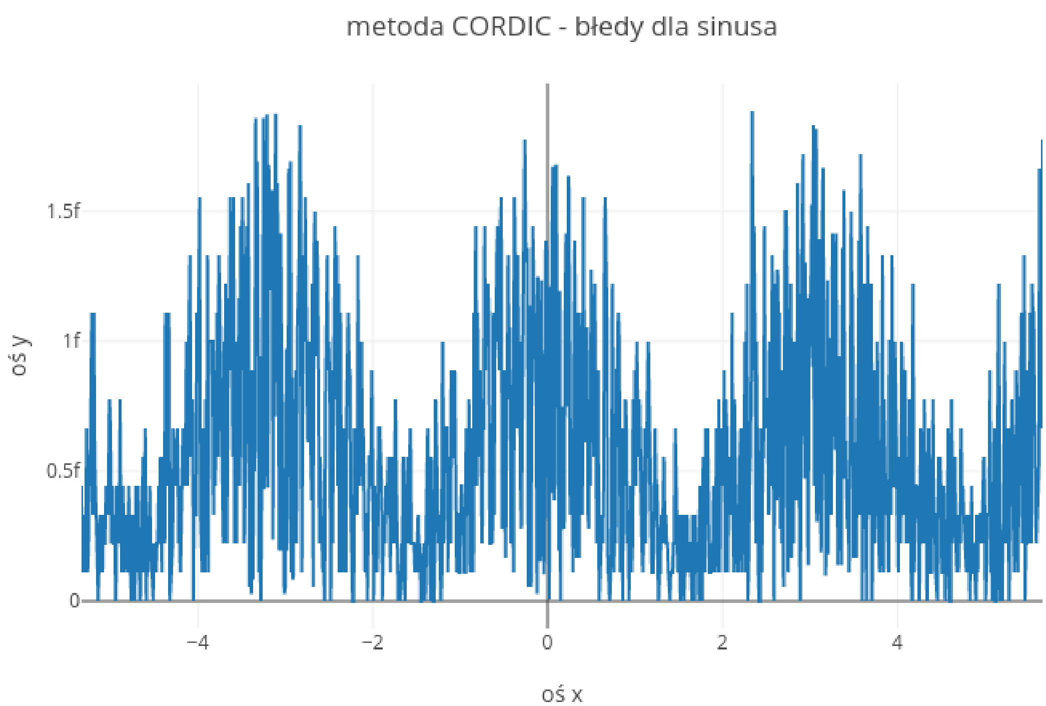
wartość σ_i to 1 lub -1, odpowiada ona na kierunek obrotu Wartość K_i możemy pominąć w kolejnych iteracjacj, i przemnożyć wynik przez odpowiednią sumę skumulowaną, ktorą będziemy mieć stablicowaną celme zwiększenia efektywności

$$K(n) = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}}$$

Również musimy mieć stablicowane wartości $atan(\gamma i)$



Rysunek 6



Rysunek 7

Błedy w przypadku metody CORDIC zarówno dla cosinusa jak i sinusa są bardziej jednolite niz w przypadku przybliżania wzorem Taylora. Sam wykres błędu ma sinusoidalny kształt. Wartości nie przekraczają $2 * 10^{-15}$, chociaz większość wartość rzędu 10^{-16}

6. Przybliżanie kątami połówkowymi

Kolejną metodą obliczania przybliżeń funkcji sinus i cosinus jest skorzystanie ze wzorów na kąty połówkowe. Znamy wartości $cos(\frac{\pi}{2}) = 0$ oraz $sin(\frac{\pi}{2}) = 1$. Możemy oczywiście ograniczyć rozpatrywany przedział do takiego o długości $\frac{\pi}{2}$, w naszym przyapdku $[0, \frac{\pi}{2}]$. Wtedy skorzystamy ze wzorów.

$$cos(\frac{\alpha}{2}) = \sqrt{\frac{1}{2}(1 + cos(\alpha)}$$

$$sin(\frac{\alpha}{2}) = \sqrt{\frac{1}{2}(1 - cos(\alpha)}$$

oraz ze wzorów na kąt sumy.

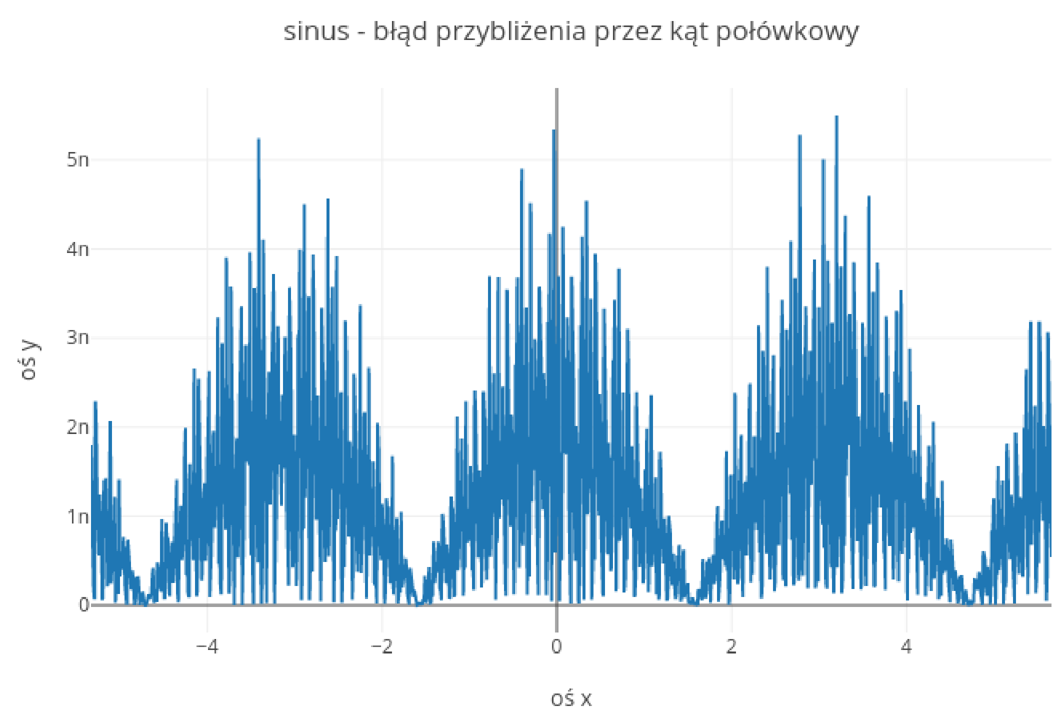
$$cos(\alpha + \beta) = cos(\alpha)cos(\beta) - sin(\alpha)sin(\beta)$$

$$sin(\alpha + \beta) = cos(\alpha)sin(\beta) + sin(\alpha)cos(\beta)$$

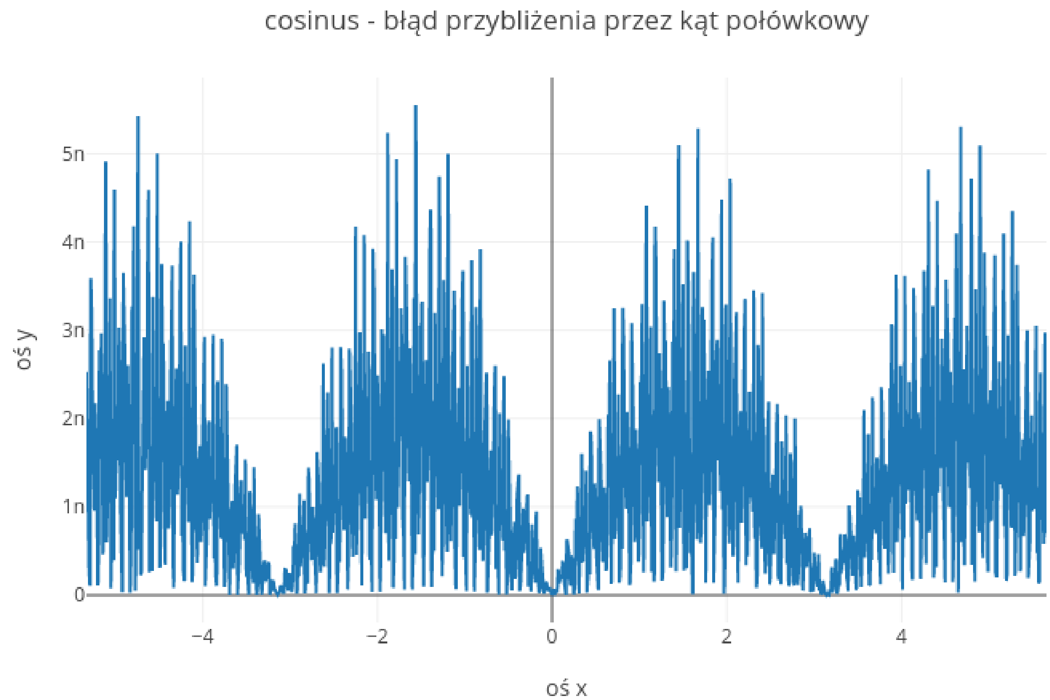
Aby znaleźć kąta tą metodą , będziemy aproksymować kąt jako

$$x = \sum_{i=0}^{n} \frac{\pi}{2^i}$$

w n krokach. Niestety moja implementacja tej metody nie jest tak efektywna, żeby porównywać ją z dokładnością maszynową. Otrzymujemy tylko dokładność rzędu 10^{-9}, co jest zdecydowanie gorszym wynikiem od poprzednich metod.



Rysunek 8



Rysunek 9

Literatura

[1] David Kincaid, Ward Cheney. *Analiza numeryczna*. Wydawnictwo Naukowo-Techniczne, Warszawa, 2002.

[2] John Burkardt *Biblioteki do metod z rodziny CORDIC*
`https://people.sc.fsu.edu/` Florida State University - The Department of Scientific Computing

[3] Biblioteki do obliczen numerycznych w języku C.
`http://www.netlib.org/fdlibm/`