

Interpolowanie krzywych funkcją sklejaną

Sprawozdanie do zadania P.2.12

Michał Martusewicz 282023

Wrocław, 11 grudnia 2016

1 Wstęp

Wybrane przeze mnie zadanie polega na interpolacji pewnych krzywych. W §2 wyjaśnię, czemu wybieram metodę krzywej parametrycznej. W §3 przedstawię interpolację wielomianem. W §4 pokażę algorytm obliczania okresowej funkcji sklejaney. W §5 pokażę jego zastosowanie na przykładzie. W §6 przedstawię funkcję odczytującą dane z pliku i powiem krótko o naturalnej funkcji sklejaney. W §7 przeanalizuję odchylenie funkcji sklejaney od pewnych funkcji bibliotecznych. W §8 podsumuję moje badania.

2 Wybór metody

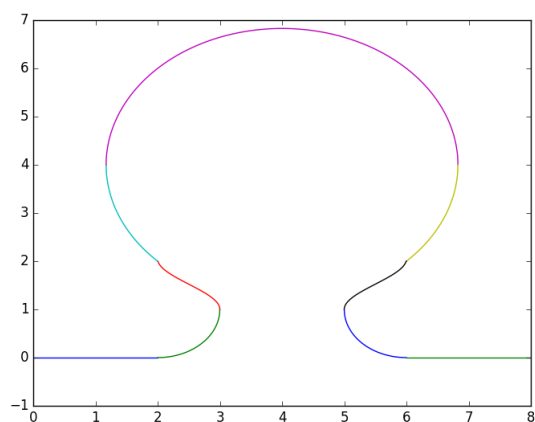
W zadaniu podane były dwie możliwe metody tworzenia odpowiednich krzywych:

- a) podzielić ciąg punktów na takie podciągi, żeby każdy z nich zawierał punkty leżące na wykresie pewnej funkcji; uzyskać przybliżoną postać linii wzorcowej łącząc wykresy przybliżeń tych funkcji;
- b) potraktować linię jako krzywą parametryczną $[x(t), y(t)]$, gdzie t jest parametrem przebiegającym przedział $[1, n]$, tak więc $x_i = x(i)$, $y_i = y(i)$ dla $i = 1, 2, \dots, n$; zrekonstruować funkcje $x(t), y(t)$ stosując interpolację.

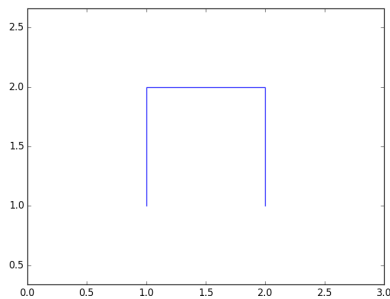
Na początek wybrałem metodę a). W celu przetestowania tej metody narysowałem prosty rysunek Ω . Oczywiście większości krzywych (w tym i mojej Ω) nie da się przedstawić za pomocą jednej funkcji. Rysunek przybliżyłem więc za pomocą różnych funkcji: prostej, arccos i okręgu, odpowiednio przesuniętych i z dobranymi odpowiednimi współczynnikami, które następnie posklejałem. Otrzymałem kształt odpowiadający oryginałowi, przedstawiony jest on na rysunku nr 1. Ze względu na czytelność sprawozdania nie będę tu szczegółowo przedstawiał programu rysującego tę Omegę, znajduje się on w pliku "omega.jl".

Szybko jednak okazało się, że wybrana metoda nie zadziała dla dowolnej krzywej. Łatwo to można zauważyć próbując przybliżyć funkcjami uproszczony rysunek stołu (na rysunku nr 2). Pionowych kresek nie przybliży żadna funkcja, ponieważ z definicji funkcji jednemu argumentowi można przyporządkować co najwyżej jedną wartość. Rozwiązaniem mogło by być stworzenie bardzo wielu funkcji których w dziedzinie znajduje się tylko jeden, ten sam argument, ale nie stworzylibyśmy w ten sposób linii ciągłej.

Z tego powodu w dalszej części sprawozdania rozważałem będę tylko metodę b).



Rysunek 1: Przybliżenie rysunku Ω kolejno funkcjami: prostej, okręgu, arccos, okręgu, arccos, okręgu, prostej.



Rysunek 2: Rysunku stołu nie da się przybliżyć żadną funkcją, problemem są tu nogi.

Tablica 1: Punkty z pewnej narysowanej krzywej

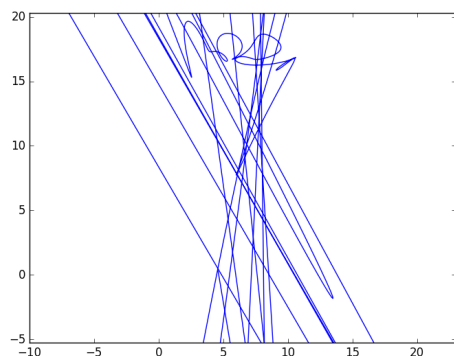
(4.6, 0.3)	(5.0, 0.7)	(7.0, 7.0)	(7.9, 9.9)
(8.0, 10.3)	(7.7, 10.7)	(7.0, 11.5)	(7.1, 12.3)
(7.8, 13.1)	(8.7, 13.2)	(9.8, 16.3)	(9.9, 16.5)
(9.6, 16.5)	(6.1, 16.5)	(6.7, 16.9)	(7.7, 16.7)
(9.2, 17.2)	(9, 18.3)	(7.7, 18.5)	(6.8, 17.3)
(5.7, 16.7)	(6.4, 17.4)	(5.5, 18.7)	(4.6, 17.4)
(5.3, 16.7)	(4.2, 17.3)	(3.3, 18.5)	(2, 18.3)
(1.8, 17.2)	(3.3, 16.7)	(4.3, 16.9)	(4.9, 16.5)
(1.5, 16.5)	(1.1, 16.5)	(1.1, 16.3)	(2.0, 11.0)
(4.2, 0.7)	(4.6, 0.3)		

3 Interpolacja wielomianowa

W celu wyboru najlepszej metody narysowałem pewną krzywą i zebrałem z niej punkty znajdujące się w tabeli nr 1. W zadaniu nie zdefiniowano konkretnej metody znajdowania funkcji interpolującej daną krzywą parametryczną, dlatego będę testował kolejne metody na tym właśnie rysunku.

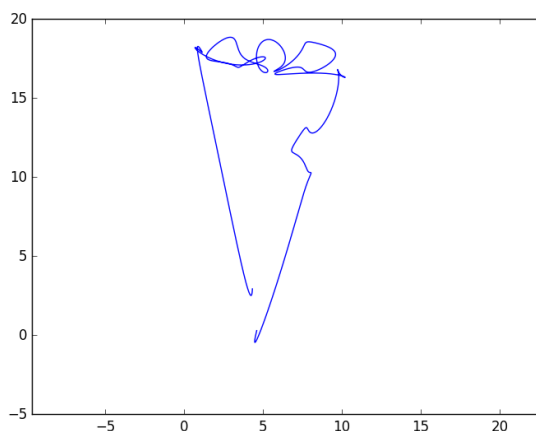
Pierwszym, najprostszym pomysłem jest interpolacja wielomianowa. Metodę tę zaprogramowałem następująco: najpierw funkcja wylicza za pomocą ilorazów różnicowych współczynniki i je tablicuje, a potem uogólnionym schematem Hornera wylicza wartości kolejnych $x(t)$ i $y(t)$ i zaznaczam je na wykresie.

Pewną trudnością było odpowiednie dobranie współczynnika t . Pierwszym wyborem były punkty równoodległe, jednak efekt był dalece nie zadowalający, co widać na rysunku 3. Niektóre wartości krzywej były nawet rzędu 10^{14} , czego jednak ze względu na czytelność rysunku na niem nie ująłem. Łatwo wytłumaczyć niepowodzenie tej metody. Wiadomo, że przy równoodległych punktach wielomiany „wybuchają” przy końcach przedziału, stąd tylko środek rysunku jest dość dobrze odwzorowany.



Rysunek 3: Przybliżenie narysowanej krzywej wielomianem, dla równoodległych wartości t .

Kolejnym sposobem było wybranie kolejnych t w miejscach, w których znajdują się kolejne pierwiastki wielomianu Czebyszewa n -tego stopnia. Kolejne wartości zostały wyrażone zatem według wzoru: $t_i = \cos\left(\frac{2*i-1}{2*n} * \pi\right)$, dla $i = 1, 2, \dots, n$. Wykres dla takich t został przedstawiony na rysunku nr 4. Krzywa jest już znacząco lepiej przybliżona, przede wszystkim wszystkie wartości krzywej są rzędu mniejszego niż 10^2 i nie „wybuchają” przy końcach. Niestety odwzorowanie nie było w pełni satysfakcjonujące z dwóch powodów. Po pierwsze rysunek nie jest aż tak dobrze oddany, pewne nieprzecinające się w rzeczywistości linie przecinają się na wykresie. Po drugie krzywa podana była jako krzywa zamknięta, a wykres jest krzywą otwartą. Na niedokładność tej metody prawdopodobnie wpływa również pewna niedokładność obliczeń komputerowych, choć wszystkie były wykonywane na BigFloatach przy wartości `setprecision()` ustawionej na co najmniej 300. To jednak pokazuje, że interpolacja wielomianem nie jest najlepszym pomysłem.



Rysunek 4

4 Algorytm okresowej funkcji sklejaney

Twierdzenie

Dla danej liczby naturalnej n , danych węzłów x_0, x_1, \dots, x_n ($a = x_0 < x_1 < \dots < x_n = b$) i danej funkcji f istnieje dokładnie jedna funkcja $\tilde{s} \in C^2[a, b]$, zwana okresową funkcją sklejaną interpolacyjną III stopnia, spełniająca następujące warunki:

1. w każdym z podprzedziałów $[x_{k-1}, x_k]$ ($k = 1, 2, \dots, n$) funkcja \tilde{s} jest identyczna z pewnym wielomianem stopnia co najwyżej trzeciego;
2. $\tilde{s}(x_k) = f(x_k)$ ($k = 0, 1, \dots, n$);
3. $\tilde{s}^{(i)}(a+0) = \tilde{s}^{(i)}(b-0)$ ($i = 0, 1, 2$).

(Łatwo zauważyć, że warunek 3. pociąga za sobą równość $f(x_n) = f(x_0)$). W każdym z przedziałów $[x_{k-1}, x_k]$ ($k = 1, 2, \dots, n$) jest

$$\begin{aligned} \tilde{s}(x) = & h_k^{-1} \left\{ \frac{1}{6} M_{k-1} (x_k - x)^3 + \frac{1}{6} M_k (x - x_{k-1})^3 \right. \\ & \left. + \left[(x_{k-1}) - \frac{1}{6} M_{k-1} h_k^2 \right] (x_k - x) + \left[f(x_k) - \frac{1}{6} M_k h_k^2 \right] (x - x_{k-1}) \right\} \end{aligned}$$

Wartości $M_k := \tilde{s}''(x_k)$ ($k = 0, 1, \dots, n$) spełniają układ

$$\lambda_k M_{k-1} + 2M_k + (1 - \lambda_k) M_{k+1} = d_k \quad (k = 1, 2, \dots, n)$$

gdzie $M_{n+1} := M_1, M_0 = M_n$ oraz

$$d_k := \frac{6}{h_k + h_{k+1}} \left[\frac{f(x_{k+1}) - f(x_k)}{h_{k+1}} - \frac{f(x_k) - f(x_{k-1}))}{h_k} \right] \quad \lambda_k := h_k / (h_k + h_{k+1}) \quad (k = 1, 2, \dots, n),$$

gdzie z kolei $f(x_{n+1}) := f(x_1)$ i $h_{n+1} := h_1$. Zauważmy, że ten układ równań można przedstawić za pomocą następującej macierzy:

$$\begin{bmatrix} 2 & 1 - \lambda_1 & 0 & 0 & \dots & \lambda_1 \\ \lambda_2 & 2 & 1 - \lambda_2 & 0 & \dots & 0 \\ 0 & \lambda_3 & 2 & 1 - \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \lambda_{n-1} & 2 & 1 - \lambda_{n-1} \\ 1 - \lambda_n & 0 & \dots & 0 & \lambda_n & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

Algorytm ich rozwiązywania jest de facto rozwiązywaniem za pomocą eliminacji Gaussa i jest lekko zmodyfikowaną wersją dla naturalnej funkcji sklejaney. Algorytm ten przedstawia się następująco:

Obliczamy pomocnicze wielkości $p_1, p_2, \dots, p_{n-1}, q_0, q_1, \dots, q_{n-1}, u_0, u_1, \dots, u_{n-1}$ i s_0, s_1, \dots, s_{n-1} w następujący sposób rekurencyjny:

$q_0 := u_0 := 0, \quad s_0 := 1,$
dla $k = 1, 2, \dots, n-1$:

$$\begin{aligned} p_k &:= \lambda_k q_{k-1} + 2, \\ q_k &:= (\lambda_k - 1)/p_k, \\ s_k &:= -\lambda_k s_{k-1}/p_k, \\ u_k &:= (d_k - \lambda_k u_{k-1})/p_k \end{aligned}$$

Zauważmy, że $n-1$ początkowych równań naszego układu można zapisać jako $M_k = q_k M_{k+1} + s_k M_n + u_k$ dla $k = 1, 2, \dots, n-1$
Następnie definiujemy t_0, t_1, \dots, t_n i v_0, v_1, \dots, v_n za pomocą wzorów

$$t_n := 1, \quad v_n := 0,$$

dla $k = n-1, n-2, \dots, 1$:

$$\begin{aligned} t_k &:= q_k t_{k+1} + s_k, \\ v_k &:= q_k v_{k+1} + u_k \end{aligned}$$

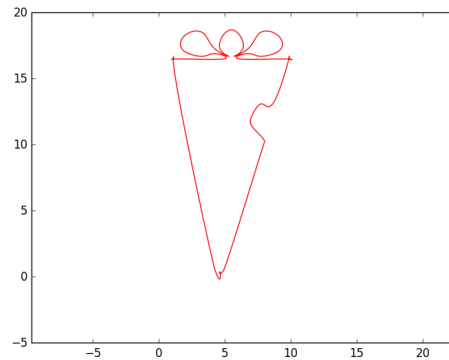
Wówczas

$$\begin{aligned} M_n &= [d_n - (1 - \lambda_n)v_1 - \lambda_n v_{n-1}]/[2 + (1 - \lambda_n)t_1 + \lambda_n t_{n-1}], \\ M_k &= v_k + t_k M_n \text{ dla } k = 1, 2, \dots, n-1. \end{aligned}$$

([3])

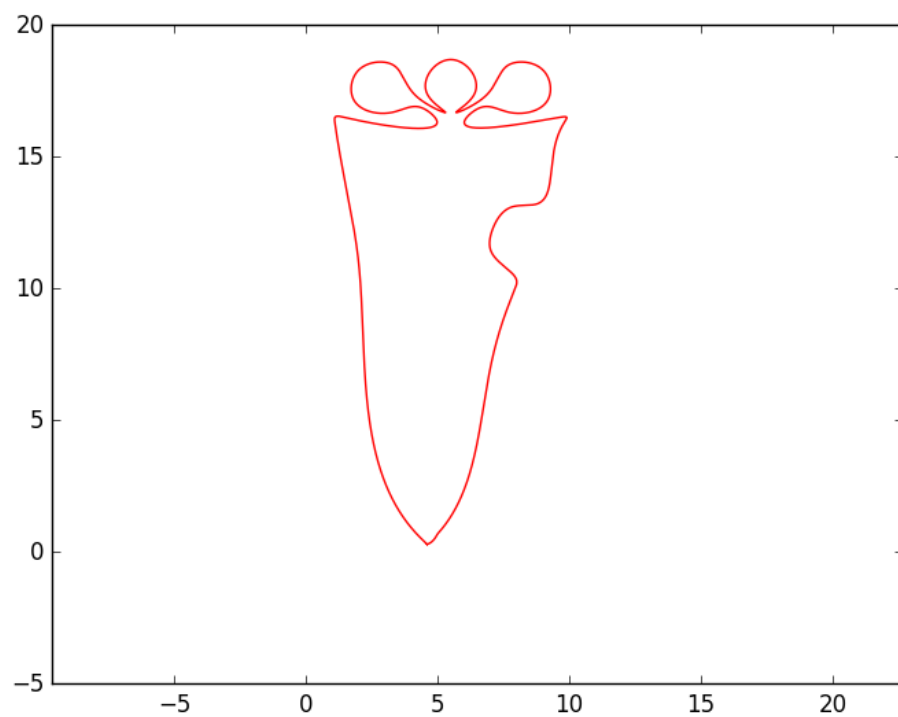
5 Użycie okresowej funkcji sklejaney

Zastosowałem powyższy algorytm do wyliczenia dwóch funkcji $\tilde{s}_x(t)$ i $\tilde{s}_y(t)$ składających się na krzywą parametryczną. Początkowo zastosowałem równoodległe t (rysunek 5). Jest to jednak niezbyt dobre rozwiązanie, gdyż odległości między punktami są znacząco różne. Ponadto powstają różne zakłamania, pętle i "dziubki".

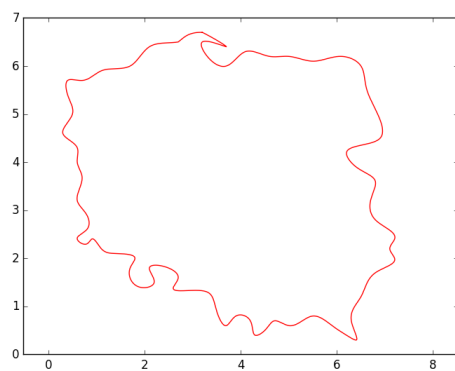


Rysunek 5: Okresowa funkcja sklejana dla równoodległych t

Postanowiłem zatem użyć tutaj twierdzenia Pitagorasa, wyliczając kolejne t według wzoru: $t_i = t_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$, dla $i = 1, 2, \dots, n$, przy czym $t_0 = 0$. Jest to w przybliżony sposób wyliczona długość łuku pomiędzy kolejnymi punktami. Dało to doskonałe efekty, co widać na rysunku numer 6. Ponadto użyłem tej funkcji do zinterpolowania tajemniczej krzywej z zadania **P2.11**. Efekt widoczny jest na rysunku numer 7



Rysunek 6: Okresowa funkcja sklejana dla t określonych za pomocą odległości pomiędzy kolejnymi punktami.



Rysunek 7: Okresowa funkcja sklejana dla t określonych za pomocą odległości pomiędzy kolejnymi punktami.

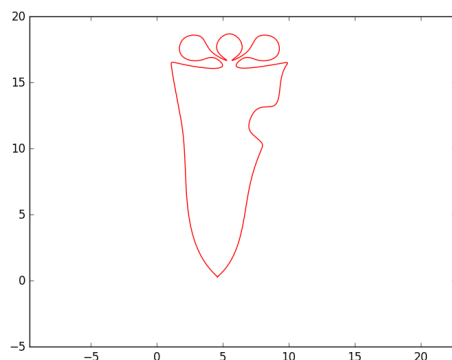
6 Odczytywanie punktów z pliku i naturalna funkcja sklejana

Częścią zadania było również napisanie programu było zaimplementowanie funkcji umożliwiającej interpolowanie plików zawartych w osobnym pliku. Zrealizowałem to za pomocą funkcji `interpol(ϵ , czyZpliku,nazwa,sposob,typ)`, gdzie:

- ϵ to odległość pomiędzy kolejnymi punktami gotowej krzywej sklejaney,
- `czyZpliku` przyjmuje wartość 1 (odczyt z pliku) lub 0 (wybór zaimplementowanej funkcji matematycznej);
- `nazwa` to nazwa pliku (gdy `czyZpliku=1`) lub żądanej funkcji matematycznej (w przeciwnym przypadku);
- `sposob` to sposób tworzenia t ("pitagoras", "równoodległe", "czebyszew");
- `typ` to typ funkcji ("okresowa", "naturalna", "wielomian").

Funkcja zwraca krotkę (X,Y) , gdzie X i Y są gotowymi tablicami z argumentami i odpowiadającymi im wartościami. Format punktów w pliku musi być następujący: $(x_0, y_0) < tab > (x_1, y_1) < tab > \dots < tab > (x_n, y_n)$, gdzie $< tab >$ to znak tabulatora. W folderze `prog` załączone są przykładowe pliki: "marchewka.pkt" i "polska.pkt".

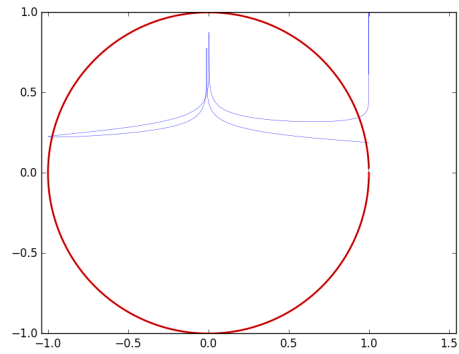
Ponieważ nie każda krzywa jest krzywą zamkniętą, zaimplementowałem również naturalną funkcję sklejaną. naturalna funkcja sklejana różni się od okresowej tylko tym, że w naturalnej na krańcach przedziałów druga pochodna jest równa zero, zaś w okresowej pierwsza i druga pochodna na krańcach przedziału równają się sobie. Algorytm obliczania funkcji okresowej jest ogólnie znany, dość prosty i podobny do algorytmu okresowej funkcji [3]. Ponieważ szczęśliwie mój rysunek zaczyna się w odpowiednim miejscu, dlatego też naturalna funkcja sklejana interpoluje ją właściwie, co widać na rysunku numer 8.



Rysunek 8: Naturalna funkcja sklejana dla t określonych za pomocą odległości pomiędzy kolejnymi punktami.

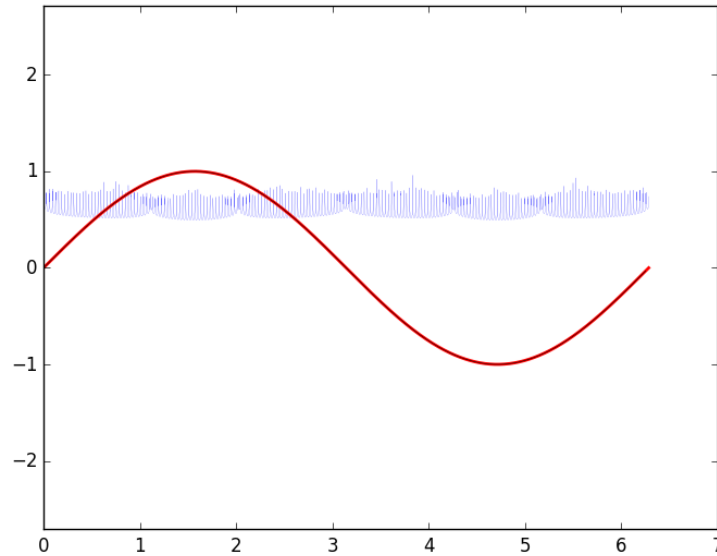
7 Analiza błędów

Postanowiłem sprawdzić dokładność interpolowanej krzywej na kilku przykładach. Pierwszym wyborem był zwykły okrąg o promieniu 1. Dla takiej krzywej $x(t) = \cos t$, $y(t) = \sin t$. Na rysunku numer 9 zostały nałożone na siebie wykresy funkcji sklejanej i funkcji bibliotecznych. Została również przedstawiona krzywa błędu, dla wygody przedstawiona w postaci $e(x) = \frac{-1}{16} * \log_{10} |f(x) - \tilde{s}(x)|$. Jak widać krzywa błędu jest dość symetryczna (niesymetryczności wynikają zapewne z zaokrągleń i zbyt małej dokładności). Można zauważyć, że funkcja najdokładniejsza jest w okolicach zera i jeden, im dalej od zera, tym większy błąd. Jednak nawet w -1, gdzie błąd jest największy, nie przekracza on 10^{-2} co jest i tak jest nie do odróżnienia na rysunku. Stosując gęstsze próbkowanie i większą precyzję otrzymujemy lepsze przybliżenie, jednak program ma dość dużą złożoność obliczeniową i każde zwiększenie precyzji znacząco wydłuża czas obliczeń.



Rysunek 9: Okrąg. Czerwona linia to wykres okresowej krzywej sklejaney, czarna to krzywa parametryczna korzystająca z funkcji bibliotecznych, a niebieska to $-1/8 \log_{10}(|\tilde{s}(x) - f(x)|)$

Jako inny przykład postanowiłem wybrać funkcję $f(x) = \sin x$ ze względu na to, że jest bardzo różna od funkcji wielomianowych. Interpolowałem ją naturalną funkcją sklejaną. Efekt jest przedstawiony na rysunku numer 10. Łatwo zauważyć, że generalnie błąd zawiera się w przedziale $[10^{-16}, 10^{-8}]$



Rysunek 10: $f(x) = \sin(x)$. Czerwona linia to wykres naturalnej krzywej sklejaney, czarna to krzywa parametryczna korzystająca z funkcji bibliotecznych, a niebieska to $-1/16 \log_{10}(|s(x) - f(x)|)$

8 Wnioski

Jest wiele różnych sposobów przybliżania krzywych. Nie każdą funkcję da się przybliżyć składanymi funkcjami, dlatego bardziej wartą uwagi wydaje się metoda krzywych parametrycznych. Do tego sposób przybliżenia tych krzywych również ma znaczenie. Najmniej dokładna jest interpolacja jednym wielomianem każdej krzywej, przy równoodległym parametrze t , wynika to z charakterystyki wielomianów, które "wybuchają" przy końcowych punktach. Gdy wybierzemy t jako kolejne zera wielomianu Czebyszewa n -tego stopnia, sytuacja znacząco poprawia się, jednak odwzorowanie dalej jest mocno niedokładne. Znacząco lepsza jest funkcja sklejana (zarówno naturalna, jak i okresowa), jednak przy równoodległym t mogą powstawać zaburzenia. Odległości pomiędzy punktami które interpolujemy zazwyczaj nie są równe, stąd lepszym rozwiązaniem jest przybliżenie tych odległości za pomocą twierdzenia Pitagorasa. Taka metoda jest już wystarczająco dokładna, co pokazują przykłady. Na dokładność możemy wpływać poprzez zwiększenie próbkowania krzywej, lub zwiększenie precyzji BigFloatów, jednak złożoność obliczeniowa metody jest znacząca, stąd czas dokładniejszych obliczeń jest dużo dłuższy.

Literatura

- [1] David Kincaid, Ward Cheney. Analiza numeryczna
- [2] A. Bjorck G. Dahlquist - Metody numeryczne (przekład S. Paszkowski, wydanie drugie, Warszawa 1987)
- [3] S.Lewanowicz Analiza numeryczna, wykład 5