8 stycznia 2020

Michał Gozdera
grupa: G1
nr indeksu: 298869

64. Aproksymacja średniokwadratowa ciągła w przestrzeni funkcji sklejanych $S_1(\Delta_n, 0)$. Całkowanie 12-punktową kwadraturą Gaussa-Legendre'a. Tablicowanie funkcji, przybliżenia i błędu w m punktach oraz obliczenie błędu średniokwadratowego w tych punktach

Projekt nr 3

1 Wstęp

Celem projektu jest zbadanie właściwości aproksymacji średniokwadratowej ciągłej w przestrzeni liniowych funkcji sklejanych. Do całkowania, niezbędnego podczas obliczania iloczynów skalarnych funkcji, używamy kwadratury Gaussa-Legendre'a określonej na 12 punktach. Otrzymane wyniki, tj. wartości funkcji aproksymowanej, funkcji aproksymującej oraz błędu względnego przedstawiamy w tablicy. Dla ułatwienia oceny rezultatów rysujemy wykresy zadanej funkcji i elementu optymalnego. Obliczamy błąd średniokwadratowy.

Obliczenie współczynników funkcji aproksymującej wymaga rozwiązania układu z macierzą trójdiagonalną. W tym celu korzystamy z algorytmu eliminacji Gaussa dla macierzy trójdiagonalnych.

Zastosowanie powyższej metody do aproksymacji większości funkcji daje dobre rezultaty (co widać na załączonych w dalszej części raportu wykresach). Należy jednak brać pod uwagę błąd, który zależny jest od siatki punktów, na jakiej określona jest przestrzeń liniowych funkcji sklejanych. Wybranie odpowiedniej liczby punktów oraz odległości między nimi zapewni optymalną wartość błędu.

2 Opis metody

Pierwszą rzeczą, na jaką należy zwrócić uwagę jest konieczność obliczania całek (jest to potrzebne przy obliczaniu współczynników funkcji aproksymującej). Posłużymy się 12-punktową kwadraturą Gaussa-Legendre'a, czyli wyznaczymy współczynniki oraz węzły tej kwadratury. Niech przedział całkowania: $[-1, 1]$. Wtedy węzły kwadratury to pierwiastki wielomianu Legendre'a stopnia 12. Wielomian ten określony jest rekurencyjnie:

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_{k+1}(x) = \frac{2k+1}{k+1} x P_k(x) - \frac{k}{k+1} P_{k-1}(x) \qquad k = 2, 3...$$

Współczynniki kwadratury wyliczyć można korzystając z jej rzędu, tzn.: dla 12-punkowej kwadratury Gaussa rząd wynosi $11 * 2 + 2 = 24$, a to oznacza, że jest ona dokładna (całka jest równa kwadraturze) dla wszystkich wielomianów stopnia mniejszego niż 24. Wystarczy zatem wziąć 12 takich wielomianów i korzystając z równości całki i kwadratury, wyliczyć współczynniki. Ponieważ współczynniki i węzły będą określone jednoznacznie nie ma potrzeby wyliczania ich za każdym razem. W programie zapisujemy zatem na stałe wyliczone wartości:

Tabela 1: Wartości węzłów i współczynników kwadratury Gaussa Legendre'a na przedziale $[-1, 1]$.

węzły	współczynniki
-0.125233408511469	0.249147045813403
-0.367831498998180	0.233492536538355
-0.587317954286617	0.203167426723066
-0.769902674194305	0.160078328543346
-0.904117256370475	0.106939325995318
-0.981560634246719	0.047175336386512
0.125233408511469	0.249147045813403
0.367831498998180	0.233492536538355
0.587317954286617	0.203167426723066
0.769902674194305	0.160078328543346
0.904117256370475	0.106939325995318
0.981560634246719	0.047175336386512

Aby umożliwić obliczanie całek na dowolnym przedziale $[a, b]$ skalujemy przedział $[-1, 1]$ na $[-\frac{b-a}{2}, \frac{b-a}{2}]$, a następnie przesuwamy o $\frac{a+b}{2}$ otrzymując dla funkcji f wzór na kwadraturę:

$$S(f) = \frac{b-a}{2} f(\frac{b-a}{2} x_0 + \frac{a+b}{2}) + ... + \frac{b-a}{2} f(\frac{b-a}{2} x_{11} + \frac{a+b}{2})$$

Kolejnym pomocniczym algorytmem jest rozwiązywanie układów równań liniowych metodą eliminacji Gaussa w wersji dla macierzy trójdiagonalnych. Metoda polega na zmianie wejściowej macierzy na macierz złożoną z trzech diagonali (potraktowanych jako wektory pionowe). W wyniku tej operacji dostajemy macierz M rozmiaru $N \times 3$ o kolumnach: pierwszej - dolnej diagonali macierzy wejściowej, drugiej - środkowej diagonali i trzeciej - górnej diagonali. Niech b - wektor wyrazów wolnych. W każdej iteracji algorytmu (dla i od 1 do $N-1$, i oznacza numer wiersza) wykonujemy:

$$b(i+1) = b(i+1) - \frac{M(i+1,1)}{M(i,2)} b(i)$$

$$M(i+1, 1:2) = M(i+1, 1:2) - \frac{M(i+1,1)}{M(i,2)} M(i, 2:3),$$

gdzie $b(j)$ oznacza j-tą współrzędną wektora b, a $M(k,l)$ - element w k-tym wierszu i l-tej kolumnie macierzy M. Ponadto zapis $p:q$ oznacza operację na kolumnach od p-tej od q-tej macierzy M.

Wektor rozwiązań uzyskujemy iterując dla i od $N-1$ do 1:

$$x(i) = \frac{b(i) - x(i+1)M(i,3)}{M(i,2)}$$

Ostatnia współrzędna:

$$x(N) = \frac{b(N)}{M(N,2)}.$$

Przestrzenią funkcji, w jakiej przeprowadzamy aproksymację jest przestrzeń sklejanych funkcji liniowych $S_1(\Delta_n, 0)$. Niech a, b - odpowiednio pierwszy i ostatni punkt siatki Δ_n. Baza tej przestrzeni ma wymiar $n+1$ i przedstawia się jak następuje:

$$S_0(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_0} & \text{dla} \quad x \in [x_0, x_1] \\ 0 & \text{dla} \quad x \in [a,b] \setminus [x_0, x_1] \end{cases}$$

$$S_n(x) = \begin{cases} \frac{x - x_{n-1}}{x_n - x_{n-1}} & \text{dla} \quad x \in [x_{n-1}, x_n] \\ 0 & \text{dla} \quad x \in [a,b] \setminus [x_{n-1}, x_n] \end{cases}$$

$$S_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{dla} \quad x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{dla} \quad x \in [x_i, x_{i+1}] \\ 0 & \text{dla} \quad x \in [a,b] \setminus [x_{i-1}, x_{i+1}] \end{cases}$$

dla $i = 1, 2, \ldots, n-1$.

Możemy teraz przejść do właściwej części algorytmu, tj. aproksymacji średniokwadratowej ciągłej.

Niech f będzie funkcją, dla której chcemy znaleźć element optymalny $\widetilde{f}$. Niech $g_0, g_1, \ldots, g_n$ będą funkcjami bazowymi z przestrzeni $S_1(\Delta_n, 0)$. Zatem $\widetilde{f}$ można zapisać jako:

$$\widetilde{f} = \sum_{i=0}^{n} \alpha_i g_i$$

Dla każdego $g \in S_1(\Delta_n, 0)$ warunek $< \widetilde{f} - f, g >= 0$ jest równoważny warunkowi $< \widetilde{f} - f, g_j >= 0$ dla $j = 0, 1, \ldots, n$. Stąd i z liniowości iloczynu skalarnego mamy:

$$< \sum_{i=0}^{n} \alpha_i g_i - f, g_j >= 0$$

$$< \sum_{i=0}^{n} \alpha_i g_i, g_j > - < f, g_j >= 0$$

$$\sum_{i=0}^{n} \alpha_i < g_i, g_j >=< f, g_j >$$

dla $j = 0, 1, \ldots, n$.

Otrzymaliśmy układ równań liniowych z niewiadomymi $\alpha_0, \alpha_1, \ldots, \alpha_n$. Jest to tzw. układ równań normalnych. Macierzowa postać tego układu jest następująca:

$$\begin{pmatrix} < g_1, g_1 > & < g_1, g_2 > & \ldots & < g_1, g_n > \\ < g_2, g_1 > & < g_2, g_2 > & \ldots & < g_2, g_n > \\ \vdots & \vdots & \ddots & \vdots \\ < g_n, g_1 > & < g_n, g_2 > & \ldots & < g_n, g_n > \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} < f, g_1 > \\ < f, g_1 > \\ \vdots \\ < f, g_n > \end{pmatrix}$$

Oznaczmy przez G macierz układu. Ponieważ rozważając przypadek rzeczywisty mamy:

$$< g_i, g_j >=< g_j, g_i >$$

to G jest symetryczna. Z tego powodu, w celu rozwiązania powyższego układu, możemy użyć metody eliminacji Gaussa (eliminacja Gaussa dla macierzy symetrycznych jest zawsze wykonalna).

W aproksymacji średniokwadratowej ciągłej iloczyn skalarny funkcji ma postać:

$$< h_1, h_2 >= \int_a^b w(x) h_1(x) h_2(x) dx,$$

gdzie $w(x) \equiv 1$ (kwadratura Gaussa-Legendre'a), a, b - krańce przedziału aproksymacji określone przez przez siatkę punktów Δ_n.

Warto zauważyć, że $(\forall i, j)(|j - i| \geqslant 2) \implies (< g_i, g_j >= 0)$ (wynika to w oczywisty sposób z definicji funkcji bazowych). G jest zatem macierzą trójdiagonalną. Wprowadzamy więc optymalizację polegającą na zaimplementowaniu metody eliminacji Gaussa dla macierzy trójdiagonalnych. Przyspiesza to działanie programu oraz umożliwia wykonywanie obliczeń na bardzo dużych zestawach danych (tj. dla dużej liczby punktów siatki Δ_n).

W celu prezentacji wyników działania algorytmu stosujemy tablicowanie wartości funkcji, przybliżenia oraz błędu względnego w m punktach. Rysujemy wykres f oraz $\widetilde{f}$ na przedziale określonym przez siatkę punktów Δ_n. Obliczamy również błąd średniokwadratowy (MSE) według wzoru:

$$MSE = \frac{\sum_{i=1}^{m} (w(i) - g(i))^2}{m}$$

gdzie: w - wektor wartości dokładnych, g - wektor wartości przybliżonych.

3 Opis programu obliczeniowego

Program składa się z siedmiu funkcji (argumenty wejściowe zgodnie z wcześniej zdefiniowanymi oznaczeniami):

- `[val] = integral_value(fun, a, b)` - zwraca wartość `val` całki $\int_a^b fun(x)dx$, obliczonej za pomocą 12-punktowej kwadratury Gaussa-Legendre'a.
- `[x] = tridiag_GE(d1, d2, d3, b)` - zwraca wektor rozwiązań układu równań liniowych (metoda eliminacji Gaussa dla macierzy trójdiagonalnych o diagonalach `d1`, `d2`, `d3` i wektorze wyrazów wolnych `b`).
- `[y] = fun(x)` - dowolna funkcja używana podczas testowania metody.
- `[yy] = spline(i, x_points)` - zwraca i-tą funkcję z bazy liniowych funkcji sklejanych, określonych na siatce punktów `x_points`.
- `[approx_fun] = approximation(x_points, fun)` - zwraca element optymalny (funkcję) `approx_fun` dla funkcji `fun` i siatki punktów `x_points`.
- `[mse] = mean_squared_error(w, g)` - oblicza błąd średniokwadratowy.
- `[TAB, MSE] = test(x_points, fun, m)` - zwraca macierz używaną w tablicowaniu (`TAB` - jej kolumny to odpowiednio: argumenty, wartości f, wartości $\widetilde{f}$, błąd względny) oraz wartość błędu średniokwadratowego (`MSE`). Rysuje wykres funkcji aproksymowanej i aproksymującej na przedziale $[a, b]$.

4 Opis eksperymentów

Metodę testujemy dla różnych funkcji, porównując błąd średniokwadratowy oraz położenie względem siebie f i $\widetilde{f}$ na wykresach. To właśnie analiza wykresów pozawala w prosty i efektywny sposób ocenić błąd aproksymacji. Zmieniamy również siatkę punktów, na której określone są liniowe funkcje sklejane. Testujemy dla różnej liczby punktów oraz różnych odległości miedzy nimi. Dla każdej funkcji poszukujemy takiej siatki, aby błąd był satysfakcjonująco mały.

5 Przykłady obliczeniowe

Poniżej przedstawiam kilka przykładowych wywołań funkcji `test` wraz z fragmentami macierzy `TAB` oraz wykresy porównujące f i $\widetilde{f}$:

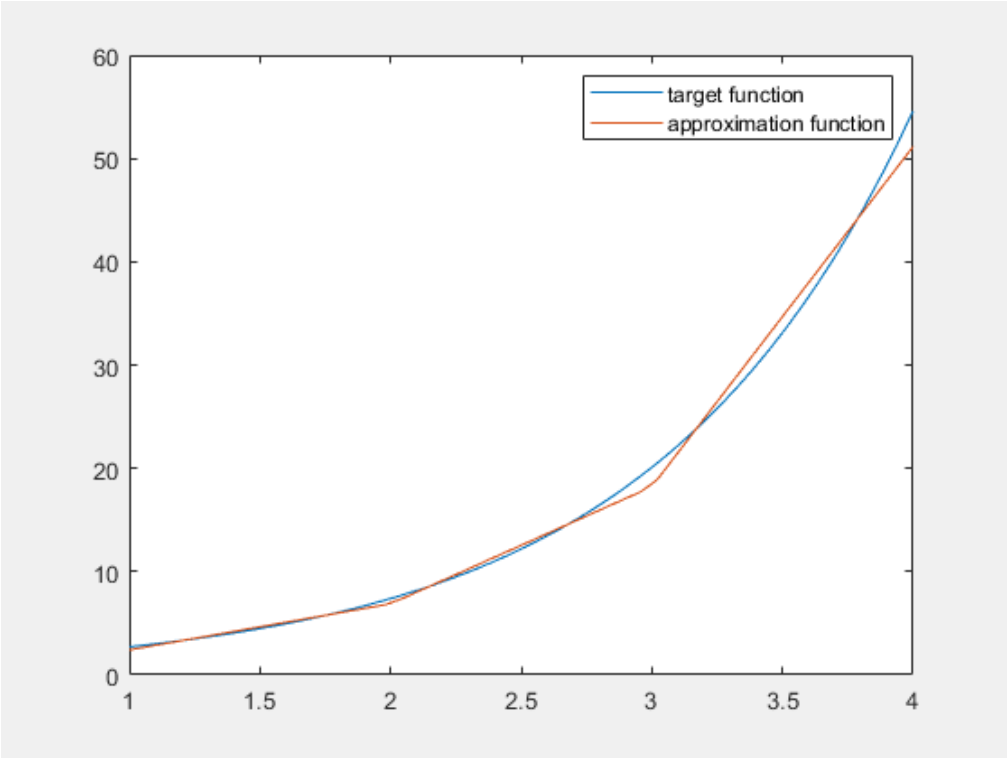
1. `test([1 2 3 4], @fun, 50)`

   $fun(x) = exp(x)$

Tabela 2: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
3.2041	24.6329	24.9378	0.0124
3.2653	26.1881	26.9540	0.0292
3.3265	27.8416	28.9701	0.0405

Wartość MSE: 0.8130



Rysunek 1: Wykres f (target function) i $\widetilde{f}$ (approximation function).

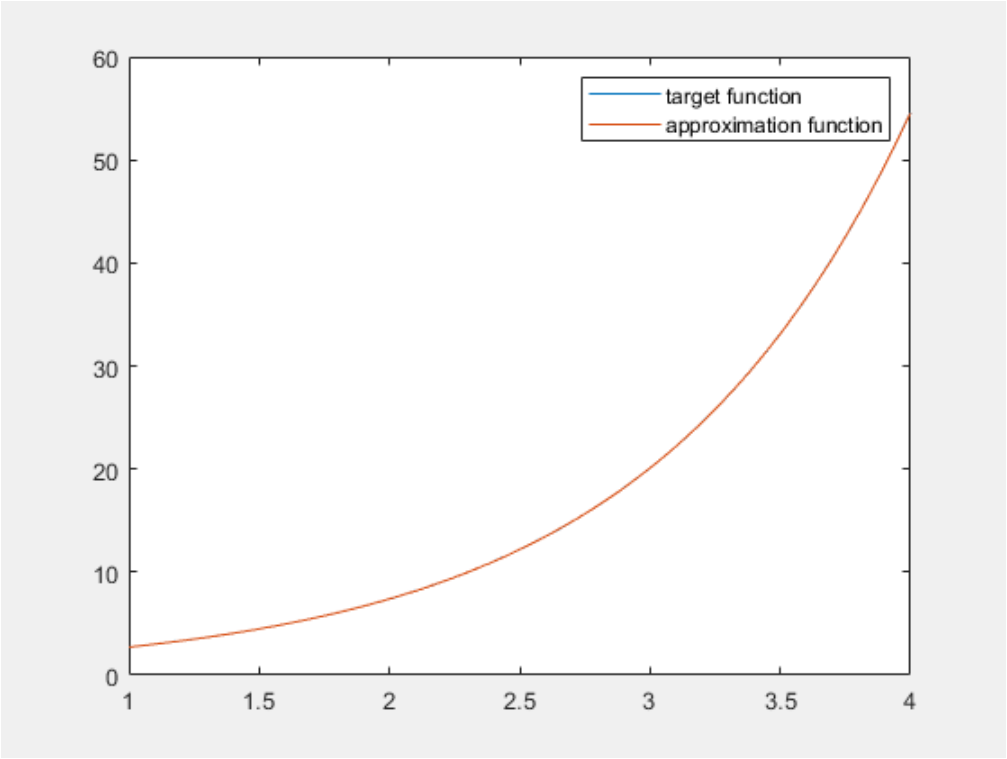
2. `test([1:0.1:4], @fun, 50)`

   $fun(x) = exp(x)$

Tabela 3: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
3.2041	24.6329	24.6173	0.0006
3.2653	26.1881	26.1957	0.0003
3.3265	27.8416	27.8459	0.0002

Wartość MSE: $9.2506e-5$



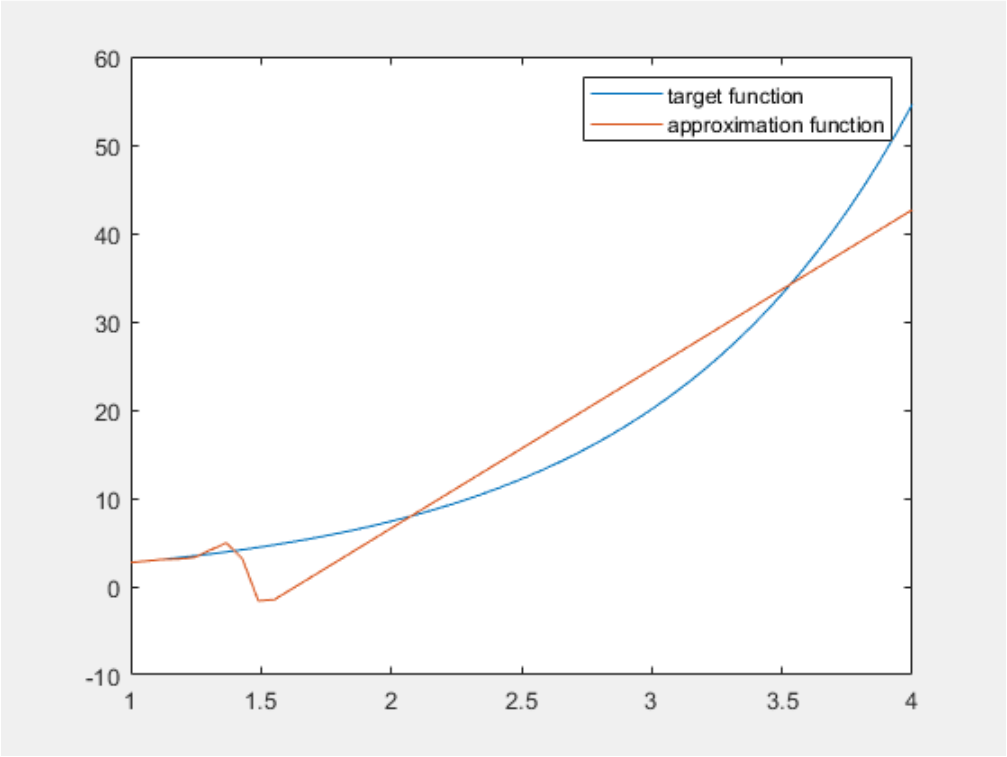
Rysunek 2: Wykres f (target function) i $\widetilde{f}$ (approximation function).

3. `test([1 1.12 1.13 1.14 1.15 1.16 1.17 1.2 1.4 1.5 4], @fun, 50)`
   $fun(x) = exp(x)$

Tabela 4: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
3.2041	24.6329	28.3158	0.1495
3.2653	26.1881	29.4196	0.1234
3.3265	27.8416	30.5233	0.0963

Wartość MSE: 16.1056



Rysunek 3: Wykres f (target function) i $\widetilde{f}$ (approximation function).

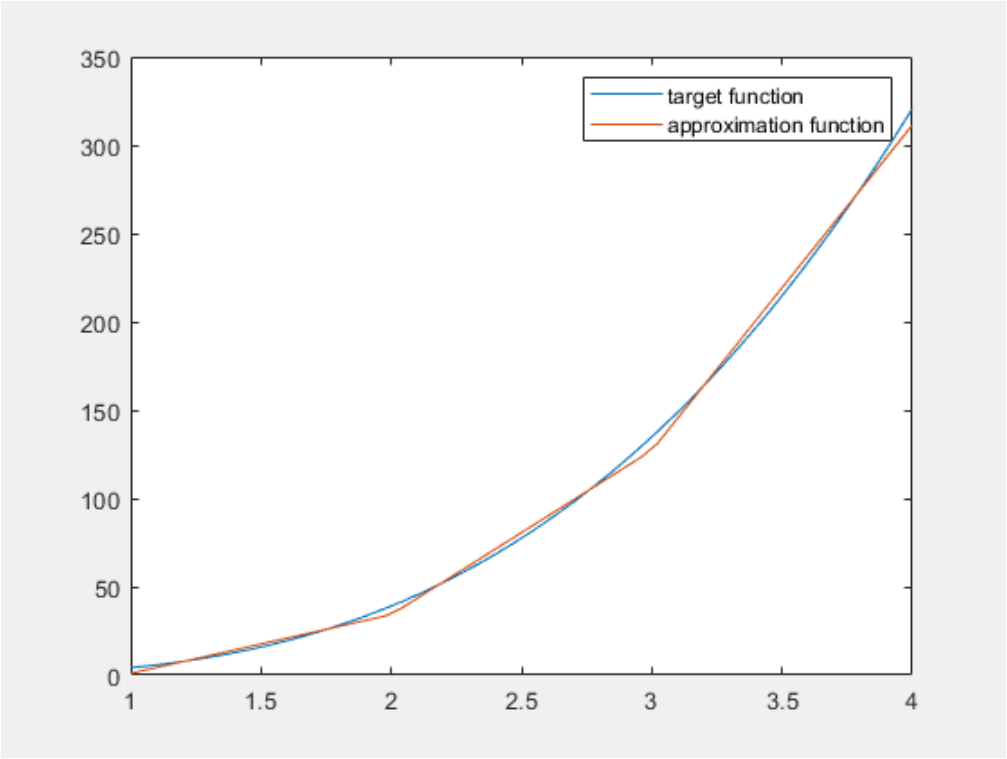
4. `test([1 2 3 4], @fun, 50)`

   $f(x) = 5x^3 - sin(x)$

Tabela 5: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
3.2041	164.5302	164.7534	0.0014
3.2653	174.2005	176.0345	0.0105
3.3265	184.2376	187.3157	0.0167

Wartość MSE: 9.8062



Rysunek 4: Wykres f (target function) i $\widetilde{f}$ (approximation function).

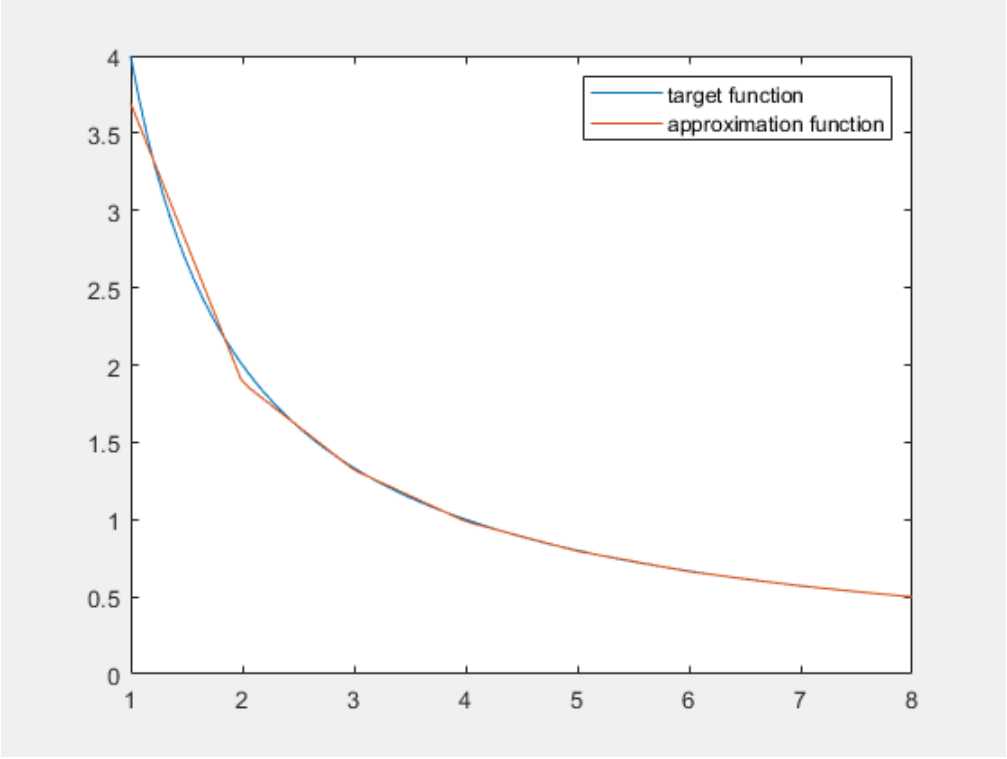
5. `test([1 2 3 4 5 6 7 8], @fun, 100)`

   $f(x) = \frac{4}{x}$

Tabela 6: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
6.0202	0.6644	0.6615	0.0044
6.0909	0.6567	0.6549	0.0028
6.1616	0.6492	0.6482	0.0014

Wartość MSE: 0.0023



Rysunek 5: Wykres f (target function) i $\widetilde{f}$ (approximation function).

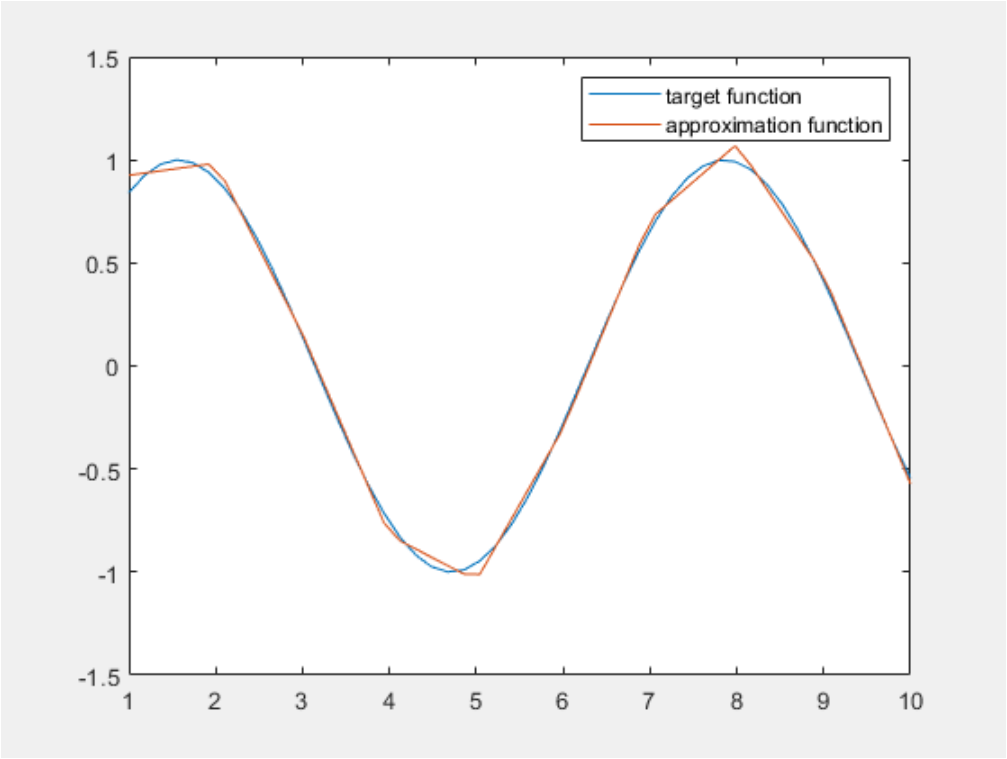
6. `test([1 2 3 4 5 6 7 8 9 10], @fun, 50)`

   $f(x) = sin(x)$

Tabela 7: Wartości przybliżone, dokładne i błąd w 3 wybranych punktach.

argument	wartość dokładna	wartość przybliżona	błąd
8.1633	0.9526	0.9722	0.0206
8.3469	0.8809	0.8559	0.0284
8.5306	0.7797	0.7397	0.0513

Wartość MSE: $9.2704e-4$



Rysunek 6: Wykres f (target function) i $\widetilde{f}$ (approximation function).

6 Analiza wyników

Porównując przykład nr 1 i 2 możemy zaobserwować, jak wraz ze wzrostem gęstości punktów na Δ_n maleje błąd aproksymacji (przy równoodległych punktach). Widać, że odpowiednio gęste dobranie punktów skutkuje bardzo dokładnym wynikiem algorytmu - wykresy funkcji aproksymującej i aproksymowanej praktycznie się pokrywają (Rysunek 2). Przykłady nr 1 i 3 ilustrują sytuację równomiernego (1) i nierównomiernego (3) rozłożenia punktów siatki na przedziale $[a, b]$ - widać, że w przykładzie 3 błąd znacznie wzrasta, mimo że liczba punktów jest większa niż w przykładzie 1. Jest to konsekwencja koncentracji tych punktów w początkowej części przedziału. Powyższe wnioski są także potwierdzone przez wartości `MSE`: dla przykładu 1 - 0.8130, dla 2 - $9.2506e-5$ i dla 3 - 16.1056. W przykładzie 3 obserwujemy też w jaki sposób dokładność aproksymacji zależy od gęstości punktów siatki: w początkowej części przedziału (gdzie punkty są wybrane gęsto) aproksymacja jest dokładna, w dalszej części przedziału (gdzie odległości między punktami są duże) widać już bardzo duże błędy (Rysunek 3).

Przykłady nr 4, 5 i 6 ilustrują działanie dla różnych (w odniesieniu do przykładów 1, 2, i 3) parametrów wywołania funkcji `test`.

Jak widać, aproksymacja jest dość precyzyjna dla różnych rodzajów funkcji, a jej dokładność zależy głównie od odpowiednio dobranej siatki punktów Δ_n. Sposób takiego doboru zależy oczywiście od konkretnej funkcji. Gęste siatki powinniśmy dobierać dla funkcji, które na małych przedziałach często zmieniają monotoniczność oraz znacznie swoje wartości. Spowoduje to jak najlepsze dopasowanie funkcji aproksymującej. Jeżeli funkcja ma charakter zbliżony do liniowej, wówczas punkty mogą być od siebie bardziej oddalone. Również rozmieszczenie punktów siatki ma znaczenie - w tych częściach przedziału, w których funkcja często zmienia monotoniczność i znacznie wartości należy rozmieścić więcej punktów. Jeżeli nie dysponujemy informacją, które części przedziału mają taki charakter, lepiej zdecydować się na punkty równoodległe.