Laboratorium 4 - Aproksymacja

Piotr Karamon

09.04.2024r.

Treści zadań

Zadania

1. Aproksymować funkcję $f(x) = 1 + x^3$ w przedziale $[0, 1]$ wielomianem pierwszego stopnia metodą średnio-kwadratową ciągłą dla $w(x) = 1$.

2. Aproksymować funkcję $f(x) = 1 + x^3$ w przedziale $[0, 1]$ wielomianem stopnia drugiego przy użyciu wielomianów Legendre'a.

Zadania domowe

1. Napisz procedurę realizującą metodę aproksymacji punktowej za pomocą wielomianów drugiego stopnia

2. Oblicz wartości funkcji $f(x) = 1 - x^2$ w dyskretnych punktach $x_i = -1 + 0.5 * i$, $i = 0..4$, a następnie aproksymuj funkcję wielomianami Grama stopnia trzeciego

3. Wykonać aproksymację funkcję $|sin(x)|$ funkcjami trygonometrycznymi w zakresie $[-\pi, \pi]$.

Zadanie 1

$f(x) = 1 + x^3$ przedział $[0, 1]$ i $w(x) = 1$

Baza ma postać: $\varphi_0(x) = 1$, $\varphi_1(x) = x$

Mamy zminimalizować wartość całki.

$$H = \int_0^1 w(x)[f(x) - q(x)]^2 dx = \int_0^1 [1 + x^3 - (c_0 + c_1 x)]^2 dx$$

Tworzymy układ równań:

$$\sum_{i=0}^{n} c_i \int_a^b w(x)\varphi_i(x) \cdot \varphi_j(x)dx = \int_0^1 w(x)f(x)\varphi_j(x)dx,\ j = 0, 1 \ldots, n$$

$$\sum_{i=0}^{1} c_i \int_0^1 \varphi_i(x) \cdot \varphi_j(x)dx = \int_0^1 (1 + x^3)\varphi_j(x)dx,\ j = 0, 1$$

$$c_0 \int_0^1 1dx + c_1 \int_0^1 xdx = \int_0^1 (1+x^3)dx$$
$$c_0 \int_0^1 xdx + c_1 \int_0^1 x^2dx = \int_0^1 (x+x^4)dx$$

Wartości całek:

$$\int_0^1 dx = 1$$
$$\int_0^1 x\,dx = \frac{1}{2}$$
$$\int_0^1 x^2\,dx = \frac{1}{3}$$
$$\int_0^1 (1+x^3)\,dx = \frac{5}{4}$$
$$\int_0^1 (x+x^4)\,dx = \frac{7}{10}$$

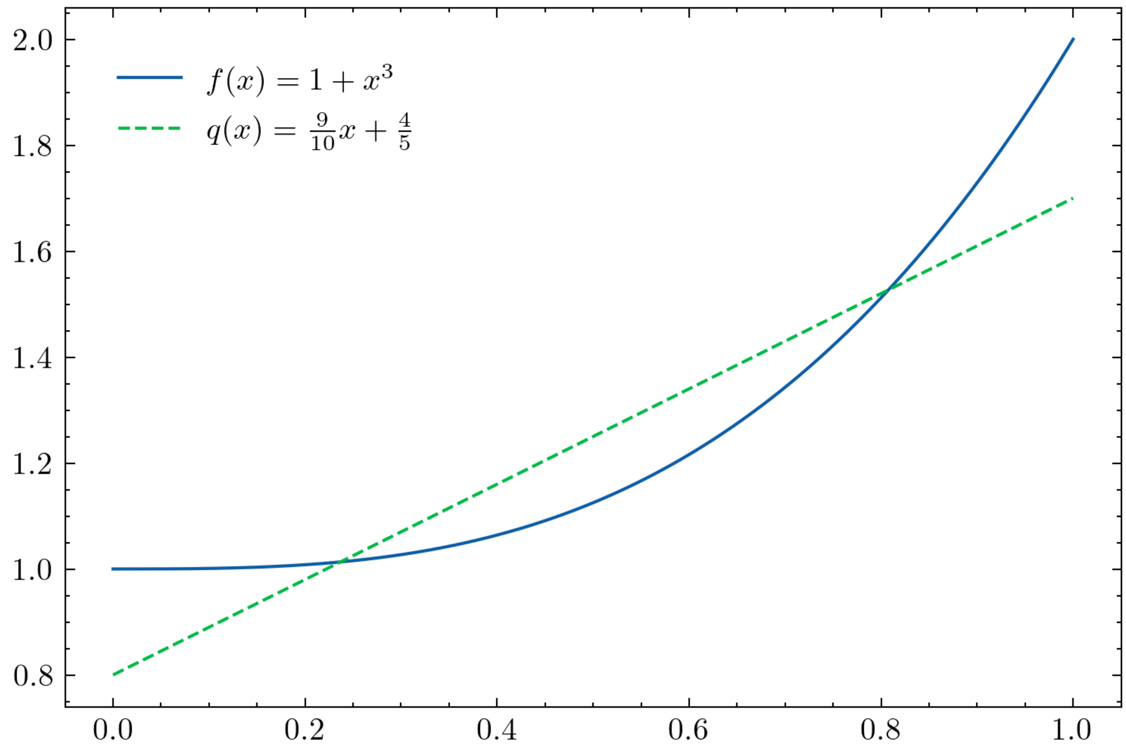
$$1c_0 + \frac{1}{2}c_1 = \frac{5}{4}$$
$$\frac{1}{2}c_0 + \frac{1}{3}c_1 = \frac{7}{10}$$

Rozwiązaniem układu jest para

$$c_0 = \frac{4}{5}$$
$$c_1 = \frac{9}{10}$$

Czyli

$$q(x) = \frac{9}{10}x + \frac{4}{5}$$



Wnioski: Funkcja $f(x)$ nie jest zbyt dobrze przybliżana za pomocą aproksymacji wielomianem pierwszego stopnia na przedziale $[0, 1]$. Co jest spodziewane, ponieważ chcemy jak najlepiej dopasować prostą do wielomianu trzeciego stopnia. Jednakże widzimy, że różnica pomiędzy $f(x)$ i $q(x)$ często nie przekracza 0.2, więc widać, że tą metodą otrzymaliśmy sensowny wynik.

Zadanie 2

$f(x) = 1 + x^3$ przedział $[0, 1]$ i $w(x) = 1$

Baza ma postać:

$$\varphi_0(x) = 1$$
$$\varphi_1(x) = x$$
$$\varphi_2(x) = \frac{1}{2}(3x^2 - 1)$$

Tworzymy układ równań:

$$\sum_{i=0}^{n} c_i \int_a^b w(x)\varphi_i(x) \cdot \varphi_j(x)dx = \int_0^1 w(x)f(x)\varphi_j(x)dx,\ j = 0, 1 \ldots, n$$

$$\sum_{i=0}^{2} c_i \int_0^1 \varphi_i(x) \cdot \varphi_j(x)dx = \int_0^1 (1 + x^3)\varphi_j(x)dx,\ j = 0, 1, 2$$

Liczymy wartości całek:

$$
\begin{aligned}
\int_0^1 \varphi_0(x)\cdot\varphi_0(x)dx &= \int_0^1 1dx && = 1\\
\int_0^1 \varphi_0(x)\cdot\varphi_1(x)dx &= \int_0^1 xdx && = \frac{1}{2}\\
\int_0^1 \varphi_0(x)\cdot\varphi_2(x)dx &= \int_0^1 \frac{3x^2-1}{2}dx && = 0\\
\int_0^1 \varphi_1(x)\cdot\varphi_1(x)dx &= \int_0^1 x^2dx && = \frac{1}{3}\\
\int_0^1 \varphi_1(x)\cdot\varphi_2(x)dx &= \int_0^1 \frac{x\left(3x^2-1\right)}{2}dx && = \frac{1}{8}\\
\int_0^1 \varphi_2(x)\cdot\varphi_2(x)dx &= \int_0^1 \frac{\left(3x^2-1\right)^2}{4}dx && = \frac{1}{5}\\
\int_0^1 (1+x^3)\varphi_0(x)dx &= \int_0^1 x^3+1dx && = \frac{5}{4}\\
\int_0^1 (1+x^3)\varphi_1(x)dx &= \int_0^1 x\left(x^3+1\right)dx && = \frac{7}{10}\\
\int_0^1 (1+x^3)\varphi_2(x)dx &= \int_0^1 \left(\frac{3x^2}{2}-\frac{1}{2}\right)\left(x^3+1\right)dx && = \frac{1}{8}
\end{aligned}
$$

Układ przyjmuje postać:

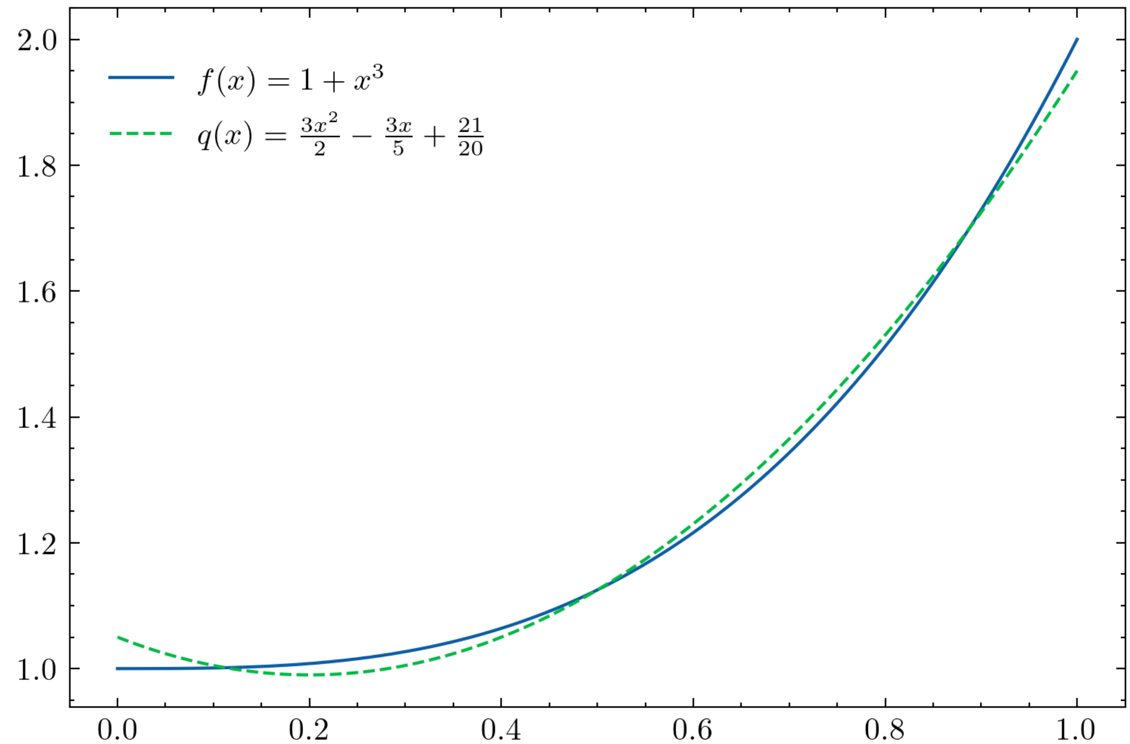
$$
\begin{aligned}
1c_0 + \frac{1}{2}c_1 + 0c_2 &= \frac{5}{4}\\
\frac{1}{2}c_0 + \frac{1}{3}c_1 + \frac{1}{8}c_2 &= \frac{7}{10}\\
0c_0 + \frac{1}{8}c_1 + \frac{1}{5}c_2 &= \frac{1}{8}
\end{aligned}
$$

Rozwiązaniem jest trójka liczb:

$$
\begin{aligned}
c_0 &= \frac{31}{20}\\
c_1 &= -\frac{3}{5}\\
c_2 &= 1
\end{aligned}
$$

Zatem:

$$q(x) = \frac{3x^2}{2} - \frac{3x}{5} + \frac{21}{20}$$



Wnioski: Funkcja $f(x)$, jak widać z wykresu, jest dobrze przybliżana wielomianem stopnia dwa przy użyciu wielomianów Legendre'a na przedziale $[0, 1]$. Baza której użyliśmy w tym zadaniu była bogatsza od tej użytej w zadaniu 1, dzięki temu mogliśmy lepiej dopasować funkcję aproksymującą. Aproksymując funkcję powinniśmy używać odpowiednich funkcji bazowych dla niej.

Zadanie domowe 1

$x_i, y_i = f(x_i)\, i = 0, 1, \ldots, n$ czyli mamy $n + 1$ węzłów interpolacji

Funkcje bazowe:

$$\varphi_0(x) = 1$$
$$\varphi_1(x) = x$$
$$\varphi_2(x) = x^2$$

Chcemy zminimalizować funkcję H

$$H = \sum_{i=0}^{n} w(x_i) \left[f(x_i) - \sum_{j=0}^{2} c_j \varphi_j(x_i) \right]^2$$

$$\frac{\partial H}{\partial c_k} = -2 \cdot \sum_{i=0}^{n} w(x_i) \left[f(x_i) - \sum_{j=0}^{2} c_j \varphi_j(x_i) \right] \cdot \varphi_k(x) = 0$$

$$\sum_{i=0}^{n} w(x_i) \left[f(x_i) - \sum_{j=0}^{2} c_j \varphi_j(x_i) \right] \cdot \varphi_k(x) = 0$$

$$\sum_{j=0}^{2} \sum_{i=0}^{n} w(x_i) \cdot c_j \cdot \varphi_j(x_i) \cdot \varphi_k(x) = \sum_{i=0}^{n} w(x_i) f(x_i) \varphi_k(x)$$

$$\sum_{j=0}^{2} \left(\sum_{i=0}^{n} w(x_i) \cdot x_i^{j+k} \right) c_j = \sum_{i=0}^{n} w(x_i) y_i x_i^k,\ k = 0, 1, 2$$

$$g_{k,j} = \sum_{i=0}^{n} w(x_i) \cdot x_i^{j+k}$$

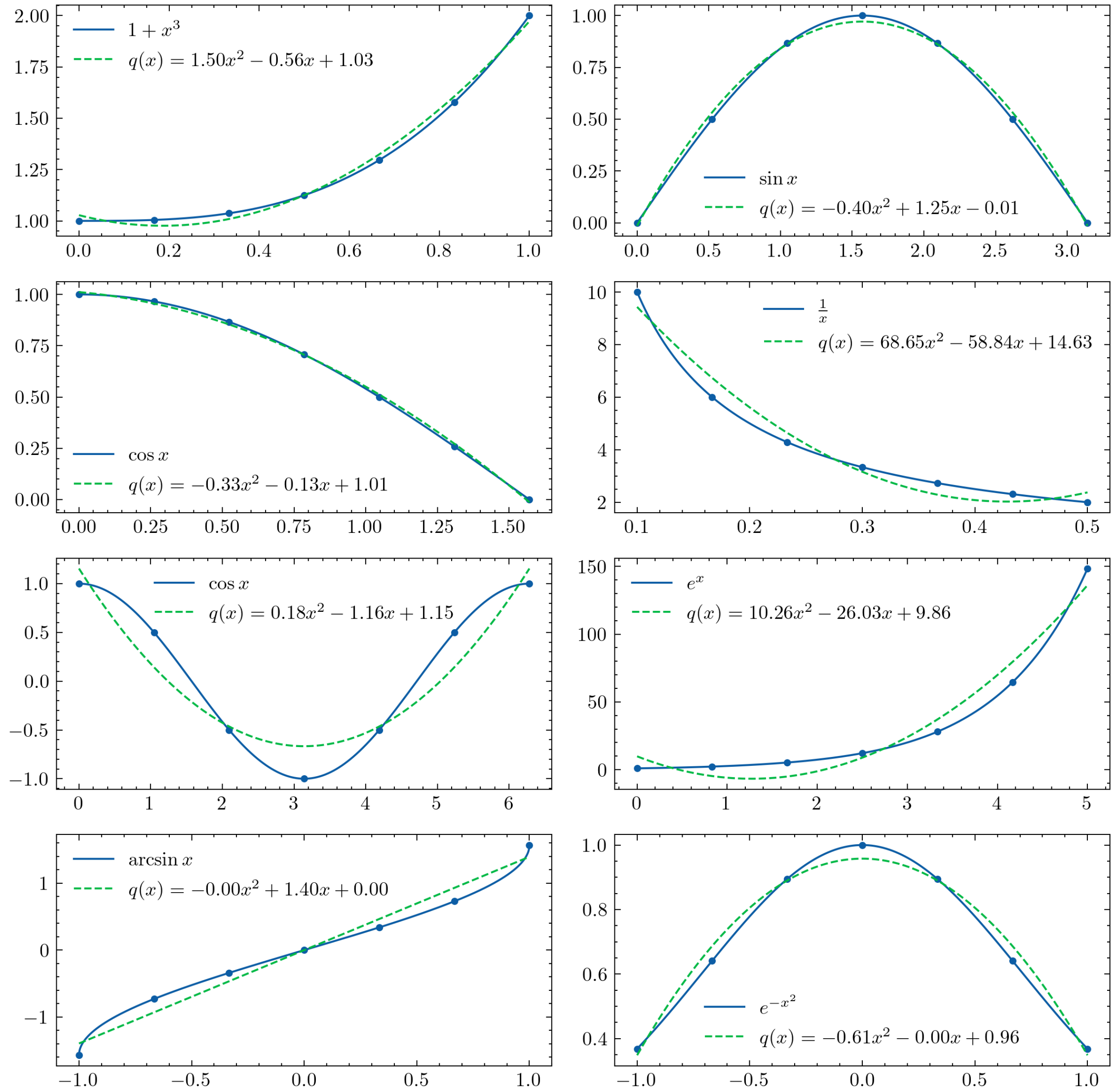
$$b_k = \sum_{i=0}^{n} w(x_i) y_i x_i^k$$

Procedura będzie zrealizowana przy użyciu języka Python i biblioteki NumPy.

```
import numpy as np

def approx(x, y, w=None):
    m = 3
    if len(x) < m:
        raise ValueError(f"not enough data points, must be at least {m}")

    if w is None:
        w = np.ones(x.shape)

    G = np.zeros((m, m))
    for k in range(m):
        for j in range(m):
            G[k, j] = np.sum(w * x**(k+j))

    b = np.zeros(m)
    for k in range(m):
        b[k] = np.sum(w * y * x**k)

    coeffs = np.linalg.solve(G, b)
    return coeffs
```

Jest to proste zastosowanie wzór wyprowadzonych powyżej oraz użycie funkcji `solve` która rozwiązuje układ równań. W przypadku nie podania wag w ustawiamy je na 1.

Teraz pora na testy:



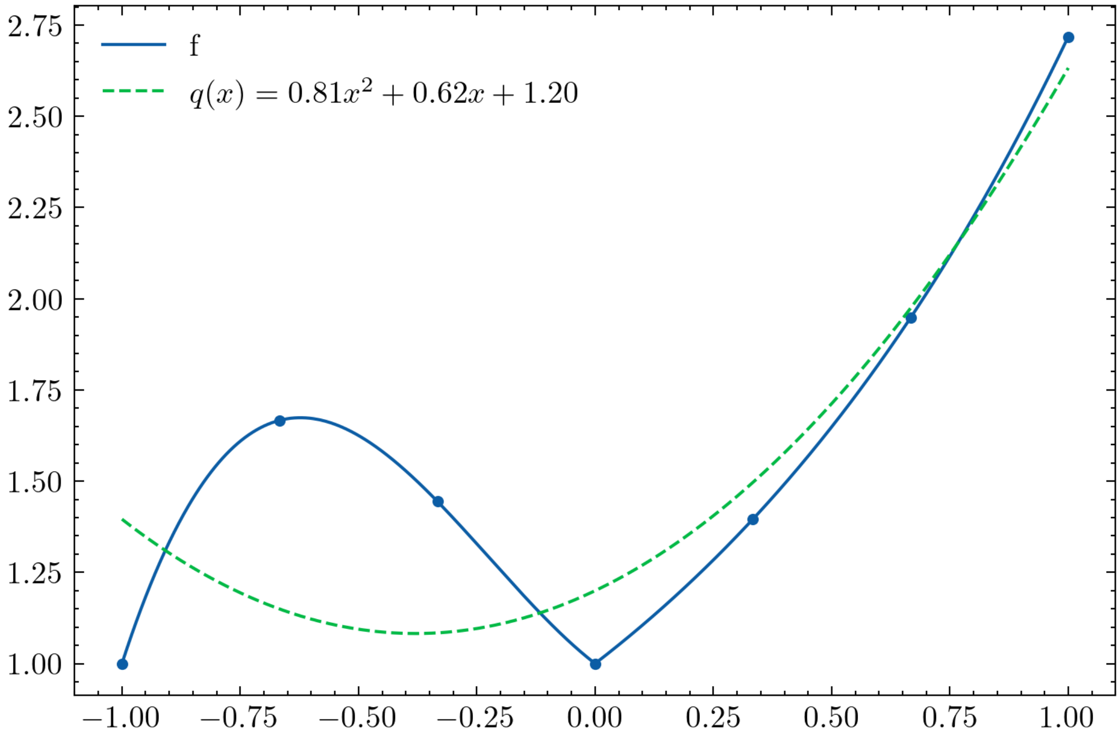
Rysunek 1: Aproksymacja punktowa różnych funkcji za pomocą wielomianów drugiego stopnia. W każdym przypadku użyto 7 równoodległych węzłów. Funkcja wagowa $w(x) = 1$.

Sprawdzamy teraz działanie wag. Będziemy aproksymować funkcję

$$f(x) = \begin{cases} 3x^3 + 2x^2 - x + 1 & \text{dla } x \in [-1, 0), \\ e^x & \text{dla } x \in [0, 1). \end{cases}$$

Funkcja wagowa ma następującą postać:

$$w(x) = \begin{cases} 1 & \text{dla } x \in [-1, 0), \\ 2 & \text{dla } x \in [0, 1). \end{cases}$$



Jak widać funkcja aproksymująca jest o wiele lepiej dopasowana dla $x \in [0, 1]$, niż dla $x \in [-1, 0)$.

Wnioski: Ta prosta metoda aproksymacji daje całkiem imponujące wyniki. Metoda działa najlepiej na małych przedziałach gdzie funkcja jest monotoniczna. Możemy dopasować jedynie wielomian stopnia drugiego, więc próba aproksymacji funkcji $\cos x$ na przedziale $[0, 2\pi]$ nie jest udana, tak samo jak z funkcją e^x na przedziale $[0, 5]$.

Zadanie domowe 2

$f(x) = 1 - x^2$

Węzły:

i	x_i	y_i
0	-1	0
1	-0.5	0.75
2	0	1.
3	0.5	0.75
4	1.0	0

Funkcja aproksymująca zbudowana na wielomianach Grama ma postać

$$y_m(x) = \sum_{k=0}^{m} \frac{c_k}{s_k} F_k^{(n)}(q).$$

gdzie:

$$s_k = \sum_{q=0}^{n} \left(F_k^{(n)}(q) \right)^2,$$

$$c_k = \sum_{i=0}^{n} y_i F_k^{(n)}(x_i),$$

$$q = \frac{x - x_0}{h} \quad \text{punkty są równoodległe} \quad x_i = x_0 + ih \quad i = 0, 1, \ldots, n$$

$$F_k^{(n)}(q) = \sum_{s=0}^{k} (-1)^s \binom{k}{s} \binom{k+s}{s} \frac{q^{[s]}}{n^{[s]}} \quad \text{wielomiany Grama}$$

$$r^{[s]} = r(r-1)\ldots(r-s+1)$$

Wielomiany te do stopnia trzeciego mają postać:

$$\begin{aligned}
F_0^{(n)}(q) &= 1 \\
F_1^{(n)}(q) &= 1 - 2\frac{q}{n} \\
F_2^{(n)}(q) &= 1 - 6\frac{q}{n} + 6\frac{q(q-1)}{n(n-1)} \\
F_3^{(n)}(q) &= 1 - 12\frac{q}{n} + 30\frac{q(q-1)}{n(n-1)} - 20\frac{q(q-1)(q-2)}{n(n-1)(n-2)}
\end{aligned}$$

Podstawiamy $n = 4$ i $q = \frac{x-x_0}{h} = 2x + 2$

$$\begin{aligned}
F_0^{(4)}(x) &= 1 \\
F_1^{(4)}(x) &= -x \\
F_2^{(4)}(x) &= -1 + 2x^2 \\
F_3^{(4)}(x) &= \frac{17}{3}x - \frac{20}{3}x^3
\end{aligned}$$

```python
from sympy import symbols, expand, Rational

xs = []
x,q = symbols('x q')

F_0 = expand(1)
F_1 = -x
F_2 = -1 + 2*x**2
F_3 = Rational(17,3)*x - Rational(20,3)*x**3

Fs = [F_0, F_1, F_2, F_3]
Fqs = [expand(f.subs(x, (q-2)/2))  for f in Fs]

xs = [-1, -0.5, 0, 0.5, 1.0,]
ys = [0, 0.75, 1., 0.75, 0]
```

```
for k in range(4):
    s_k = sum((Fqs[k].subs(q, value))**2 for value in range(5))
    c_k = sum( y_val*Fs[k].subs(x, x_val) for y_val, x_val in zip(ys,xs))
    print(f'c_{k}/s_{k} = {c_k/s_k}')
```

```
c_0/s_0 = 0.500000000000000
c_1/s_1 = 0
c_2/s_2 = -0.500000000000000
c_3/s_3 = 0
```

Podstawiamy

$$y(x) = 0.5 \cdot (1) - 0.5 \cdot (-1 + 2x^2) = -x^2 + 1$$

Wnioski: Funkcja którą uzyskaliśmy jest dokładnie taka sama jak zadana funkcja do aproksymacji. Jest to spodziewane, ponieważ aproksymujemy wielomian drugiego stopnia oraz nasza baza generuje wielomiany stopnia drugiego a nawet trzeciego. Jest to przykład pokazujący słuszność pokazanej metody, uzyskana aproksymacja bowiem ma zerowy błąd.

Zadanie domowe 3

Trygonometrycznym szeregiem Fouriera funkcji f nazywamy szereg funkcyjny następującej postaci:

$$S(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2n\pi}{T}x\right) + b_n \sin\left(\frac{2n\pi}{T}x\right)\right)$$

o współczynnikach określonych następującymi wzorami:

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos\left(\frac{2n\pi}{T}x\right)\, dx, \quad n = 0, 1, 2, \ldots,$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin\left(\frac{2n\pi}{T}x\right)\, dx, \quad n = 1, 2, 3, \ldots.$$

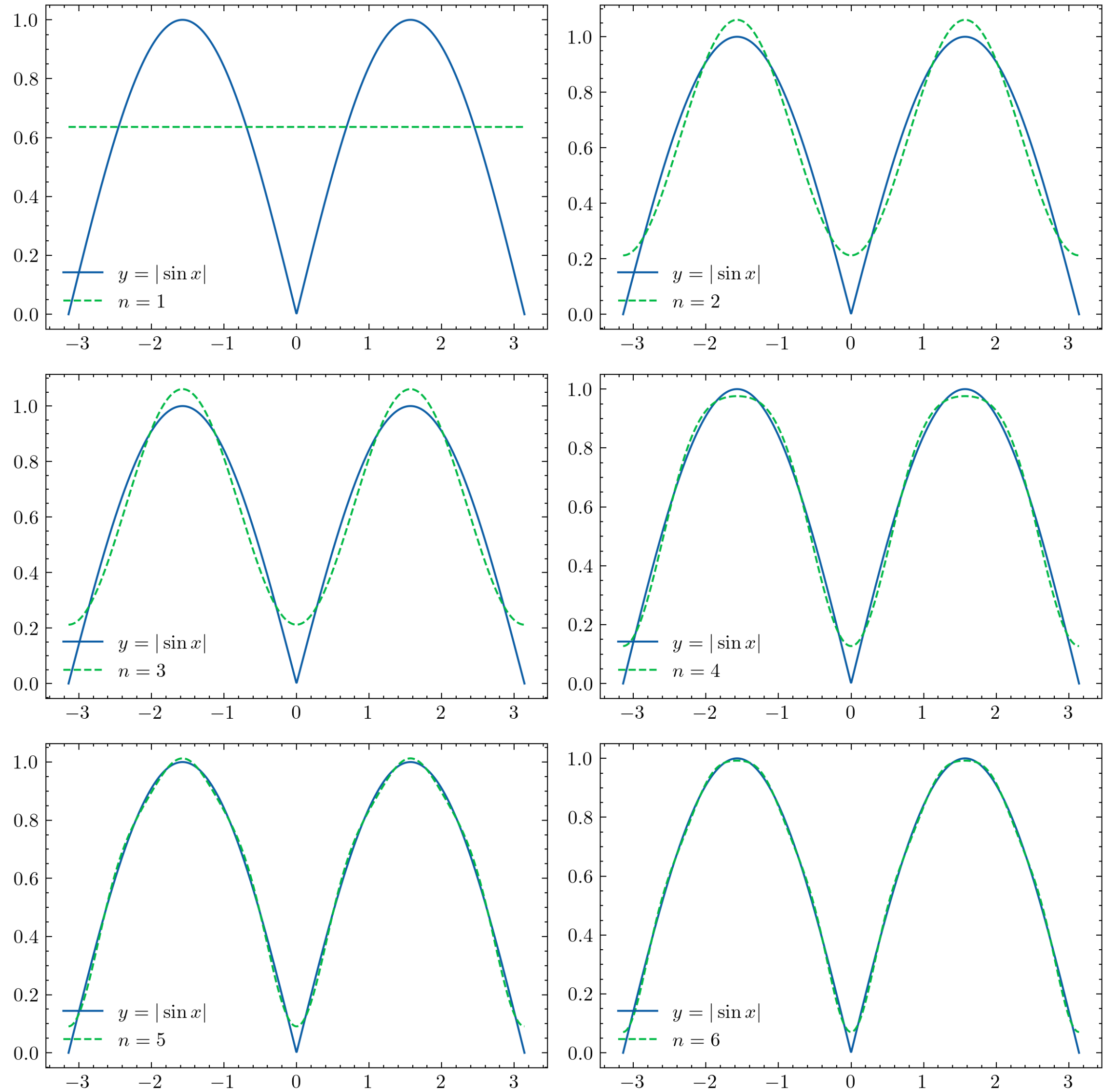
Funkcja $f(x) = \sin x$ spełnia warunki Dirichleta na przedziale $[-\pi, \pi]$. Jest ona również funkcją parzystą zatem $b_n = 0,\ n = 0, 1, 2, \ldots$. Zatem:

$$a_n = \frac{2}{\pi} \int_0^{\pi} \sin x \cos(nx)\, dx, \quad n = 0, 1, 2, \ldots,$$

Po obliczeniu całki dla a_n otrzymujemy:

$$a_n = -\frac{2}{\pi} \cdot \frac{cos(\pi n) + 1}{n^2 - 1} = -\frac{2}{\pi} \cdot \frac{(-1)^n + 1}{n^2 - 1}, \quad n = 0, 2, \ldots,$$
$$a_1 = 0$$

$$S(x) = \frac{2}{\pi} + \sum_{n=2}^{\infty} \frac{-2}{\pi} \cdot \frac{1 + (-1)^n}{n^2 - 1} \cdot \cos(nx) = \frac{2}{\pi} + \sum_{n=1}^{\infty} \frac{-4\cos(2nx)}{\pi((2n)^2 - 1)}$$



Wnioski: W oparciu o przeprowadzone przybliżenie funkcji $|\sin x|$ za pomocą trygonometrycznego szeregu Fouriera na przedziale $[\pi, \pi]$, możemy zaobserwować, że z każdym kolejnym wyrazem szeregu, aproksymacja staje się dokładniejsza.

Bibliografia

- Marian Bubak, Katarzyna Rycerz: *Metody Obliczeniowe w Nauce i Technice. Aproksymacja*
- Włodzimierz Funika: Aproksymacja za pomocą wielomianów Grama
- Włodzimierz Funika: Aproksymacja