Laboratorium 3

Interpolacja

Dominik Marek

19 marca 2024

AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

1 Zadania laboratoryjne

1. Dane są trzy węzły interpolacji(-1,2.4), (1,1.8), (2,4.5) , proszę obliczyć wielomian interpolacyjny 2-go stopnia, wykorzystując:

a) Jednomiany
b) wielomiany Lagrange'a
c) wielomiany wg wzoru Newton'a

Pokazać, że trzy reprezentacje dają ten sam wielomian

2. Wyrazić następujący wielomian metodą Hornera: $\mathrm{p}(t) = 3t^3 + 7t^2 + 5t - 4$

3. Ile mnożeń trzeba wykonać do ewaluacji wielomianu p(t) stopnia $\mathrm{n} - 1$ w danym punkcie t jeżeli wybieramy jako reprezentacje:

(a) jednomiany
(b) wielomiany Lagrange'a
(c) wielomiany Newtona

2 Zadania domowe

1. Znaleźć kompromis między granicą błędu a zachowaniem wielomianu interpolacyjnego dla funkcji Rungego $f(t) = \frac{1}{(1+25t^2)}$ dla równoodległych węzłów na przedziale [-1,1]

2.

(a) sprawdzić czy pierwsze siedem wielomianów Legendre'a są wzajemnie ortogonalne

(b) sprawdzić czy one spełniają wzór na rekurencję

(c) wyrazić każdy z sześciu pierwszych jednomianów 1, t, …, t^6 jako liniową kombinację pierwszych siedmiu wielomianów Legendre'a, p_0, …, p_6

3. Dana jest funkcja określona w trzech punktach x_0, x_1, x_2 rozmieszczonych w jednakowych odstępach (x_1=x_0+h, x_2=x_1+h): f(x_0) = y_0, f(x_1) = y_1, f(x_2) = y_2

Proszę wykonać interpolację danej funkcji sklejanymi funkcjami sześciennymi.

3.Rozwiązania -zadania laboratoryjne

1.

(a) Jednomiany:

Mając zadane węzły interpolacji możemy skonstruować wielomian $W(x) = a_0 + a_1x + a_2x^2$. Szukane a_0, a_1, a_2 znajdziemy rozwiązując układ równań:

$$\begin{cases} a_0 - 1a_1 + (-1)^2 a_2 = 2.4 \\ a_0 + 1a_1 + 1^2 a_2 = 1.8 \\ a_0 + 2a_1 + 2^2 a_2 = 4.5 \end{cases}$$

Otrzymujemy rozwiązania: $a_0 = 1.1, a_1 = -0.3, a_2 = 1$. Stąd wielomian ma postać:

$$\mathrm{W(x)} = x^2 - 0.3x + 1.1$$

(b) Wielomiany Lagrange'a:

Otrzymujemy wielomian:

$$W(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2 =$$

$$= \frac{(x-1)(x-2)}{(-1-1)(-1-2)} 2.4 + \frac{(x+1)(x-2)}{(1+1)(1-2)} 1.8 + \frac{(x+1)(x-1)}{(2+1)(2-1)} 4.5 =$$

$$= x^2 - 0.3x + 1.1$$

(c) Wielomiany wg wzoru Newton'a

$$\mathrm{W(x)} = \mathrm{f}[\mathrm{x}_0] + \mathrm{f}[\mathrm{x}_0, \mathrm{x}_1](\mathrm{x} - \mathrm{x}_0) + \mathrm{f}[\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2](\mathrm{x} - \mathrm{x}_0)(\mathrm{x} - \mathrm{x}_1)$$

$$f[x_0] = 2.4, f[x_1] = 1.8, f[x_2] = 4.5$$

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1.8 - 2.4}{1 + 1} = -0.3$$

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{4.5 - 1.8}{2 - 1} = 2.7$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = 1$$

Finalnie otrzymujemy wielomian postaci:

$$W(x) = 2.4 - 0.3(x + 1) + 1(x + 1)(x - 1) =$$

$$= x^2 - 0.3x + 1.1$$

Wnioski:

Niezależnie od wykorzystanej metody otrzymano identyczny wielomian.

2.

Dokonujemy przekształceń i otrzymujemy:

$$P(t) = 3t^3 + 7t^2 + 5t - 4 = t(3t^2 + 7t + 5) - 4 = t(t(3t + 7) + 5) - 4$$

3.

(a) Ewaluując za pomocą schematu Hornera dostajemy n − 1 mnożeń.

(b) Aby obliczyć wyraz L_k musi wykonać n − 1 mnożeń. Wyrazów jest n stąd, aby wyliczyć wszystkie wyrazy L musimy wykonać n · (n −1) mnożeń, a jeszcze dodatkowo każdy z wyrazów mnożymy przez rzędną, więc w sumie dostajemy: n · (n − 1) + n = n^2 mnożeń.

(c) Obliczenie p_k zajmuje k mnożeń, a k ∈ ⟨0, n − 1⟩ , stąd ilość mnożeń jaką musimy wykonać to:

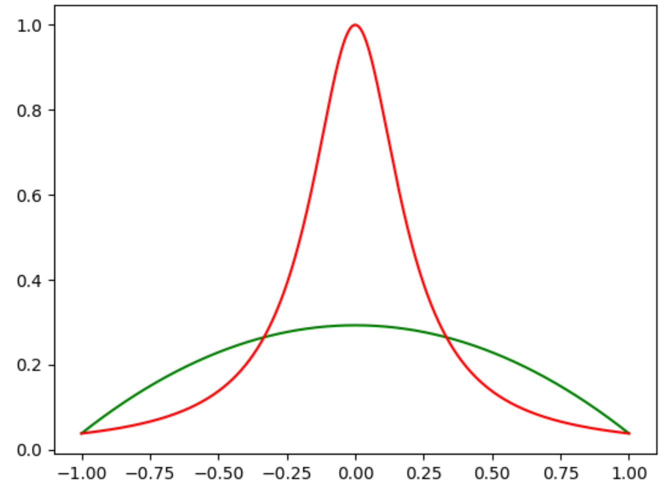
$$\sum_{k=1}^{n-1} k = n^2$$

4. Rozwiązania – zadania domowe

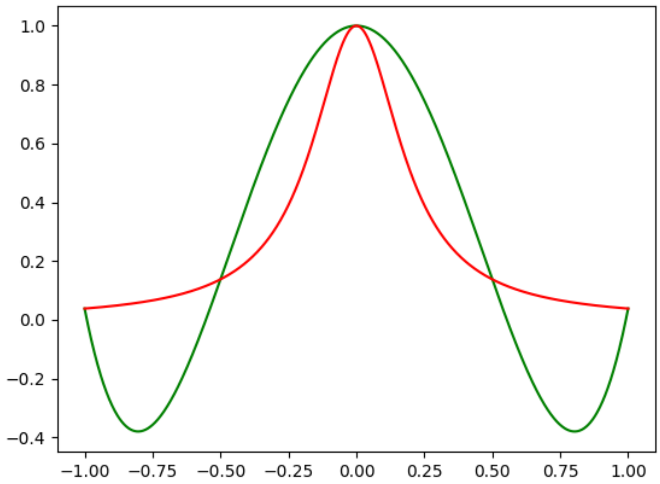
Za pomocą programu napisanego w języku Python generuje wykresy interpolujących wielomianów dla n równoodległych węzłów:

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.interpolate as inter
from typing import Final




def Runge_function(x) -> float:
    return 1/(1 + 25*x**2)


def draw_plot(min_range: int, max_range: int) -> None:
    d = np.linspace(-1, 1, num=1000, dtype=float)
    for n in range(min_range, max_range):
        knots = np.linspace(-1, 1, num=n, dtype=float)
        f_int = inter.KroghInterpolator(knots, Runge_function(knots))
        plt.plot(d, f_int(d), color='green')
        font = {
            'family': 'cambria',
            'color': 'black',
            'weight': 'bold',
            'style': 'italic',
            'size': 10
        }
        plt.title(
            f"Rysunek {n-min_range+1}: "
            f"Porównanie funkcji interpolującej oraz interpolowanej dla n = {n-2}",
            fontdict=font,
            fontsize=font['size']
        )
        plt.plot(d, Runge_function(d), color='red')
        plt.show()


def main() -> None:
    MIN_RANGE: Final = 4
    MAX_RANGE: Final = 14
    draw_plot(MIN_RANGE, MAX_RANGE)


if __name__ == '__main__':
    main()
```

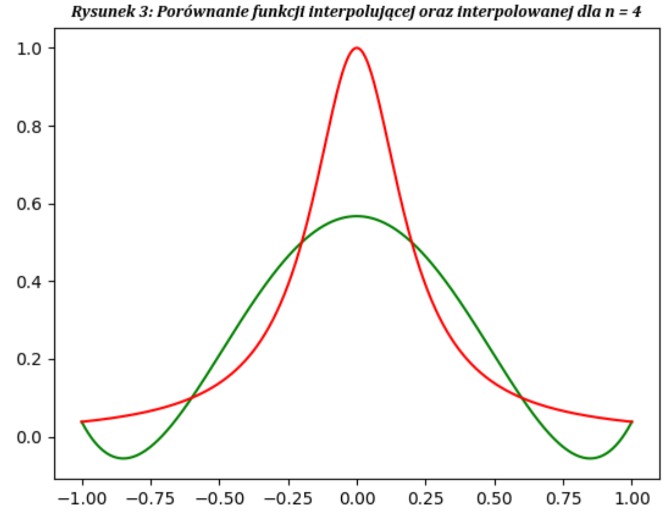
Rysunek 1: Porównanie funkcji interpolującej oraz interpolowanej dla n = 2



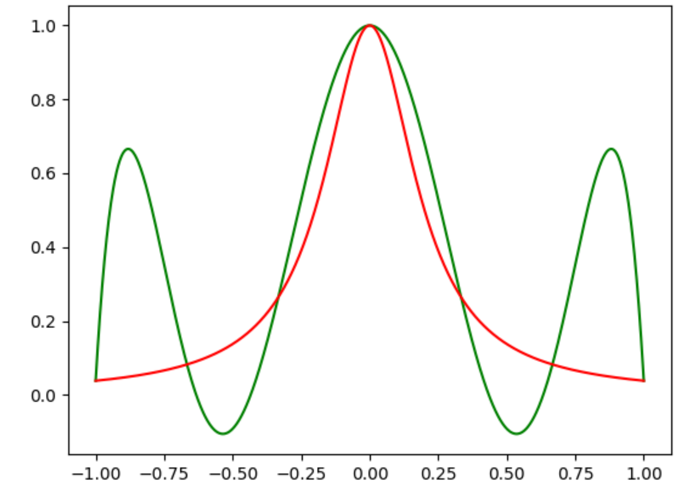
Rysunek 2: Porównanie funkcji interpolującej oraz interpolowanej dla n = 3



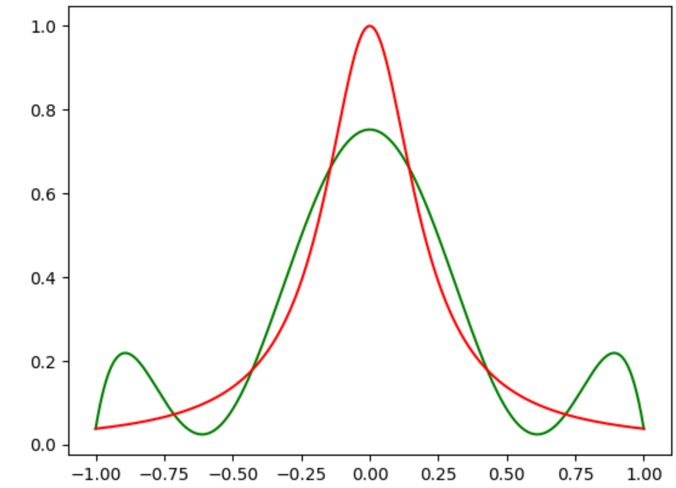
Rysunek 3: Porównanie funkcji interpolującej oraz interpolowanej dla n = 4



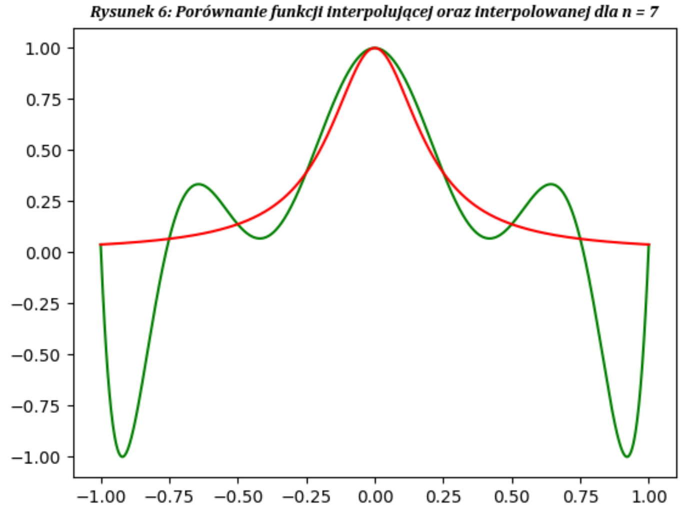
Rysunek 4: Porównanie funkcji interpolującej oraz interpolowanej dla n = 5



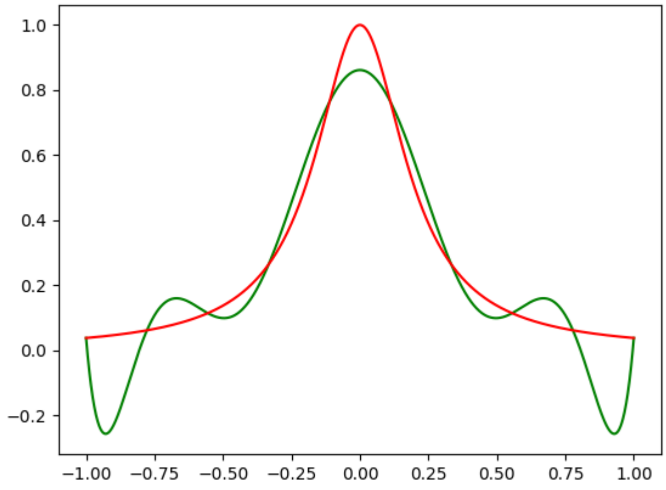
Rysunek 5: Porównanie funkcji interpolującej oraz interpolowanej dla n = 6



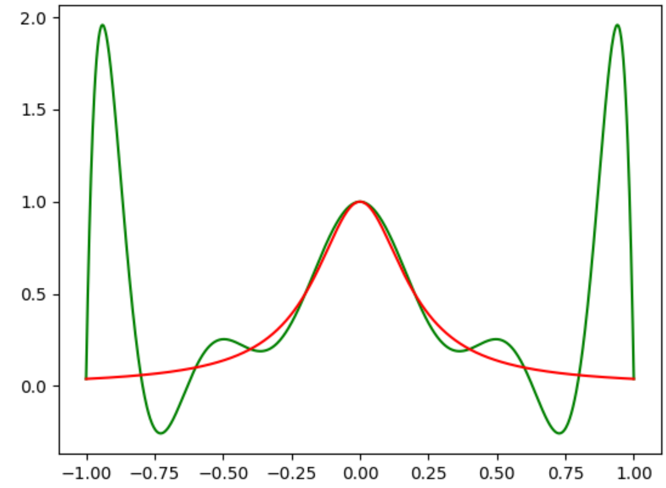
Rysunek 6: Porównanie funkcji interpolującej oraz interpolowanej dla n = 7



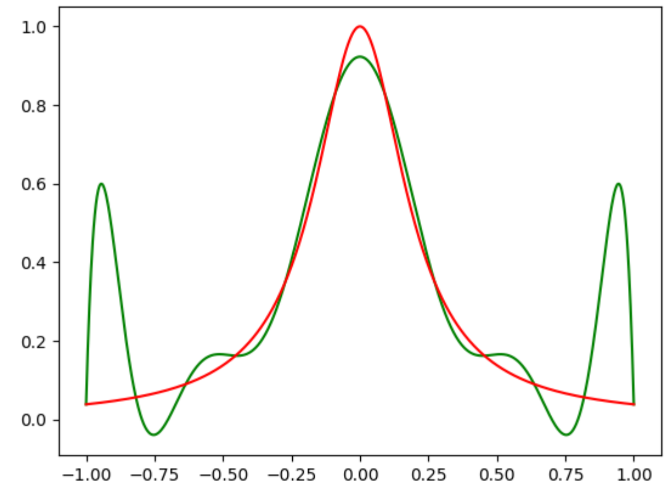
Rysunek 7: Porównanie funkcji interpolującej oraz interpolowanej dla n = 8



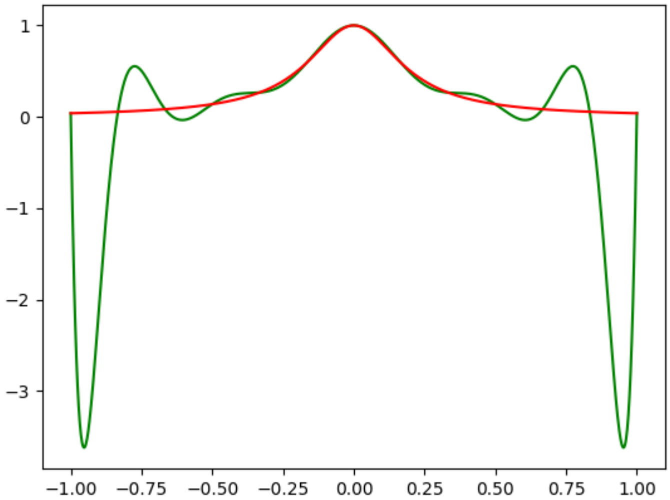
Rysunek 8: Porównanie funkcji interpolującej oraz interpolowanej dla n = 9



Rysunek 9: Porównanie funkcji interpolującej oraz interpolowanej dla n = 10



Rysunek 10: Porównanie funkcji interpolującej oraz interpolowanej dla n = 11



Wnioski:

Można zauważyć, iż początkowo rozbieżności są mniejsze dla n parzystych, jednak od n = 10 również one mają bardzo duże odstępstwa od funkcji interpolowanej. Analizując powyższe wykresy należy przyjąć jako kompromis n =8.

2.

a)

Jeśli wielomiany Legendre'a są ortogonalne na przedziale [-1. 1] względem iloczynu skalarnego w przestrzeni L^2 to:

$$\forall n_1 m \in \{0,1,..,6\}_{,}, n \neq m \int_{-1}^{1} P_n P_m \, dx = 0$$

Kolejne wielomiany Legendre'a są dane wzorem:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^2} (x^2 - 1)^n$$

$$2^n n! \int_1^1 f(x) P_n(x) \, dx = \frac{2^n n!}{2^n n!} \int_{-1}^{1} f(x) \frac{d^n}{dx^n} (x^2 - 1)^n \, dx = \left| \begin{matrix} u = f(x) & u' = f'(x) \\ v' = \frac{d^n}{dx^n}(x^2-1)^n & v = \frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n \end{matrix} \right| =$$

$$f(x) \frac{d^{n-1}}{dx^{n-1}} (x^2 - 1)^n |_{-1}^{1} = \int_{-i}^{1} f(x) \frac{d^{n-1}}{dx^{n-1}} (x^2 - 1)^n \, dx = (*)$$

Ponieważ $\frac{d}{dx}(x^2-1)^n = n(x^2-1) = 0\ dla\ x = 1\ i\ x = -1, to$:

$$(*) = \int_{-1}^{1} f'(x) \frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n\, dx$$

Otrzymaliśmy rekurencyjny wzór na całkę. Po jego zastosowaniu otrzymujemy:

$$\int_{1}^{1} f(x)P_n(x)\, dx = \frac{1}{2^n n!} \int_{-1}^{1} f^{(n)}(x)(x^2-1)^n\, dx$$

B.S.O przyjmijmy, że m < n, tak więc stopień wielomianu P_m to co najwyżej n − 1. Skoro tak to jego n-ta pochodna wynosi 0 (czyli $P_m^{(n)} = 0$). Po podstawieniu f (x) = $P_m(x)$ otrzymujemy:

$$\int_{-1}^{1} P_n P_m\, dx = \frac{1}{2^n n!} \int_{-1}^{1} P_m^{(n)}(x^2-1)^n\, dx = 0$$

Wynika z tego że wszystkie kolejne wielomiany Legendre'a są ortogonalne w sensie metryki z przestrzeni L^2, czyli w szczególności jest siedem pierwszych.

b)

Znając wzór rekurencyjny :

$$(n+1)P_{n_{+1}}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

$$P_0(x) = 1$$

$$P_1(x) = x$$

Korzystając z poniższego programu napisanego w Pythonie wyznaczam pierwsze siedem wielomianów Legendre'a.

```python
import numpy as np
from typing import Final


def generate_Legendre_polynomial(min_range: int, max_range: int):
    p = [np.poly1d([1]),np.poly1d([1, 0])]
    x = np.poly1d([1, 0])
    for i in range(min_range, max_range):
        p.append((2*i+1)/(i+1) * p[i]*x - (i/(i+1))*p[i-1])

    return p


def main() -> None:
    MIN_RANGE: Final = 1
    MAX_RANGE: Final = 6
    polynomials = generate_Legendre_polynomial(MIN_RANGE, MAX_RANGE)
    for poly in polynomials:
        print(poly)


if __name__ == '__main__':
    main()
```

Po wykonaniu programu w Pythonie otrzymuje poniższe wielomiany:

$$P_2(x) = 1.5x^2 - 0.5$$

$$P_3(x) = 2.5x^3 - 1.5x$$

$$P_4(x) = 4.375x^4 - 3.75x^2 + 0.375$$

$$P_5(x) = 7.875x^5 - 8.75x^3 - 1.875x$$

$$P_6(x) = 14.44x^6 - 19.69x^4 + 6.562x^2 - 0.3125$$

Są to dokładnie wielomiany Legendre'a podawane w tablicach.

c)

$$\mathrm{P}_0 = 1$$

$$\mathrm{P}_1 = \mathrm{x}$$

$$x^2 = \frac{2}{3}\left(\frac{1}{2}\cdot 1 + \frac{3}{2}x^2 - \frac{1}{2}\right) = \frac{2}{3}\left(\frac{1}{2}P_o + P_2\right)$$

$$x^3 = \frac{2}{5}\left(\frac{5}{2}x^3 - \frac{3}{2}x + \frac{3}{2}x\right) = \frac{2}{5}\left(P_3 - \frac{3}{2}P_1\right)$$

$$x^4 = \left[\frac{8}{35}\left(\frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8} + \frac{20}{8}\left(\frac{3}{2}x^2 - \frac{1}{2}\right) + \frac{7}{8}\cdot 1\right)\right] =$$

$$= \frac{8}{35}P_4 + \frac{8}{35}\cdot\frac{20}{8}P_2 + \frac{8}{35}\cdot\frac{7}{8}P_0 = \frac{8}{35}P_4 + \frac{20}{35}P_2 + \frac{7}{35}\cdot P_0$$

$$x^5 = \left[\frac{8}{63}\left(\frac{63}{8}x^5 - \frac{70}{8}x^3 + \frac{15}{8}x + \frac{7}{2}\left(\frac{5}{2}x^3 - \frac{3}{2}x\right) + \frac{27}{8}\cdot x\right]\right. = \frac{8}{63}P_5 + \frac{28}{63}P_3 + \frac{27}{8}P_1$$

$$x^6 = \left[\frac{16}{231}\left(\frac{231}{16}x^6 - \frac{315}{16}x^4 + \frac{105}{16}x^2 - \frac{5}{16} + \frac{9}{2}\left(\frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8}\right) + \frac{55}{8}\left(\frac{3}{2}x^2 - \frac{1}{2}\right) + \frac{33}{16}\cdot 1\right)\right] =$$

$$= \frac{16}{231}\left(P_6 + \frac{9}{2}P_4 + \frac{55}{8}P_2 + \frac{33}{16}P_0\right) = \frac{16}{231}P_6 + \frac{72}{231}P_4 + \frac{110}{231}P_2 + \frac{33}{231}P_0$$

3.

x	x_0	$x_1 = x_0 + h$	$x_2 = x_0 + 2h$
y	y_0	y_1	y_2

$$\begin{cases} S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3 \,, x_0 \le x \le x_0 + h \\ S_1(x) = a_1 + b_1(x - x_0 - h) + c_1(x - x_0 - h)^2 + d_1(x - x_0 - h)^3 \,, x_0 + h \le x \le x_0 + 2h \end{cases}$$

$$S_0(x_0) = a_0 = y_0$$

$$S_0(x_0 + h) = a_0 + b_0 h + c_0 h^2 + d_0 h^3 = y_1$$

$$S_1(x_0 + h) = a_1 = y_1$$

$$S_1(x_0 + 2h) = a_1 + b_1 h + c_1 h^2 + d_1 h^3 = y_2$$

$$S'(x) = b_0 + 2c_0(x - x_0) + 3d_0(x - x_0)^2$$

$$S_1'(x) = b_1 + 2c_1(x - x_0 - h) + 3d_1(x - x_0 - h)^2$$

$$S_0''(x) = 2c_0 + 6\,d_0(x - x_0)$$

$$S_1''(x) = x_1 + 6d_1(x - x_0 - h)$$

Wiemy, że:

$$S_0'(x_0 + h) = S_1'(x_0 + h) \Rightarrow b_0 + 2c_0 h + 3d_0 h^2 = b_1$$

$$S_0''(x_0 + h) = S_1''(x_0 + h) \Rightarrow 2c_0 + 6d_0 h = 2c_1$$

Z warunków brzegowych:

$$S''(x_0) = 0 \Rightarrow 2c_0 = 0 \Rightarrow c_0 = 0$$

$$S_1''(x_0 + 2h) = 0 \Rightarrow 2c_1 + 6d_1 h = 0 \Rightarrow c_1 = -3d_1 h$$

$$\begin{cases} 2c_0 + 6d_0 h = 2c_1 \\ c_0 = 0 \\ c_1 = -3d_1 h \end{cases}$$

$$c_1 = 3d_0 h = -3d_1 h$$

$$\begin{cases} a_0 + b_0 h + c_0 h + d_0 h^3 = y_1 \\ a_0 = y_0 \\ c_0 = 0 \end{cases}$$

$$b_0 h + d_0 h^3 = y_1 - y_0$$

$$\begin{cases} a_1 + b_1 h + c_1 h^2 + d_1 h^3 = y_2 \\ a_1 = y_1 \\ c_1 = -3d_1 h \end{cases}$$

$$b_1 h - 2d_1 h^3 = y_2 - y_1$$

$$\begin{cases} b_0 h + 2c_0 h + 3d_0 h^2 = b_1 \\ c_0 = 0 \end{cases}$$

$$b_0 + 3d_0 h^2 = b_1$$

Łącząc powyższe równania otrzymujemy:

$$\begin{cases} b_0 h + d_0 h^3 = y_1 - y_0 \\ b_1 h - 2d_1 h^3 = y_2 - y_1 \\ b_0 + 3d_0 h = b_1 \\ d_0 = -d_1 \end{cases}$$

$$\begin{cases} b_0 h + d_0 h^3 = y_1 - y_0 \\ b_0 h + 5d_0 h^3 = y_2 - y_1 \end{cases}$$

$$4d_0 h^3 = y_2 + y_0 - 2y_1$$

$$d_0 = \frac{y_0 - 2y_1 + y_2}{4h^3}$$

$$d_1 = \frac{2y_1 - y_2 - y_o}{4h^3}$$

$$c_0 = 0$$

$$c_1 = 3d_0 h = \frac{3(y_0 - 2y_1 - y_2)}{4h^3}$$

$$b_0 h = y_1 - y_0 - d_0 h^3 = y_1 - y_0 - \frac{y_0 - 2y_1 + y_2}{4}$$

$$b_0 = \frac{-5y_0 + 6y_1 - y_2}{4h}$$

$$b_1 h = y_2 - y_1 + 2d_1 h^3 = y_2 - y_1 + \frac{2y_1 - y_0 - y_2}{2} = \frac{y_2 - y_0}{2}$$

$$b_1 = \frac{y_2 - y_0}{2h}$$

$$a_0 = y_0$$
$$a_1 = y_1$$

Końcowe wartości współczynników wynoszą:

$$\begin{cases} a_o = y_0 \\ a_1 = y_1 \\ b_0 = \dfrac{-5y_0 + 6y_1 - y_2}{4h} \\ b_1 = \dfrac{y_2 - y_0}{2} \\ c_1 = \dfrac{3(y_0 - 2y_1 - y_2)}{4h^3} \\ c_0 = 0 \\ d_0 = \dfrac{y_0 - 2y_1 + y_2}{4h^3} \\ d_0 = \dfrac{-y_0 + 2y_1 - y_2}{4h^3} \end{cases}$$

Finalnie otrzymujemy:

$$\begin{cases} S_0(x) = y_0 + \dfrac{-5y_0 + 6y_1 - y_2}{h}(x - x_0) + \dfrac{y_0 - 2y_1 + y_2}{4h^3}(x - x_0)^3 \, , x_0 \le x \le x_0 + h \\ S_1(x) = y_1 + \dfrac{y_2 - y_0}{2h}(x - x_0 - h) + \dfrac{3(y_0 - 2y_1 - y_2)}{4h^3}(x - x_0 - h)^2 + \dfrac{2y_1 - y_2 - y_0}{4h^3}(x - x_0 - h)^3 \, , x_0 + h \le x \le x_0 + 2h \end{cases}$$