

WYKŁAD 1

INTERPOLACJA WIELOMIANOWA

Sformułowanie problemu interpolacji. Metoda Lagrange'a

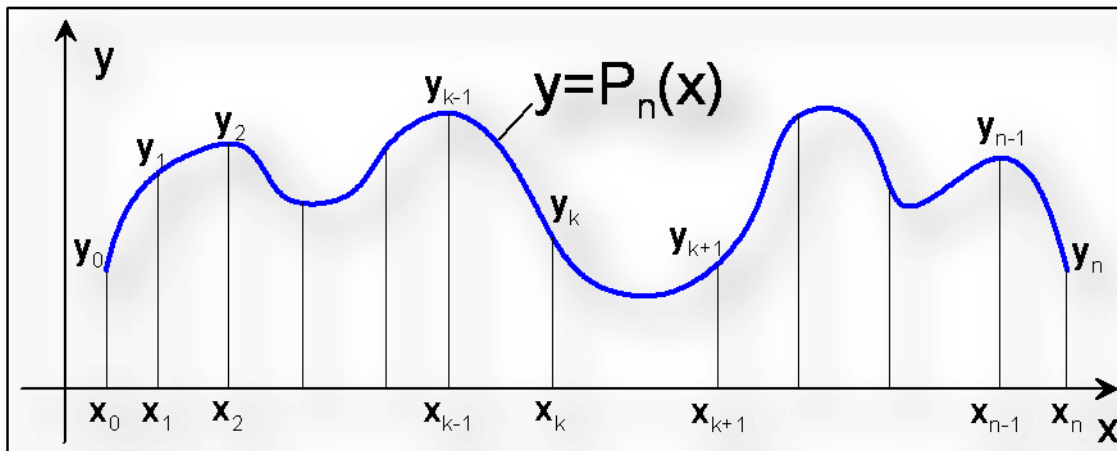
Rozważmy zadany układ punktów $\{(x_j, y_j), j = 0, 1, \dots, n\}$, zwanych dalej **węzłami interpolacyjnymi**.

Poszukujemy **wielomianu interpolacyjnego** zadanego wzorem

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \equiv \sum_{k=0}^n a_k x^k$$

i takiego, że

$$P_n(x_j) = y_j \quad , \quad j = 0, 1, \dots, n$$



Innymi słowy – wykres wielomianu powinien być linią przechodzącą przez zadany układ węzłów interpolacyjnych (vide obrazek

Trzy podstawowe pytania:

1. Czy taki wielomian istnieje?
2. Jeśli tak, to czy jest jedyny?
3. Jeśli tak, to jak go wyznaczyć?

Zacznijmy od podejścia typu „brutalna siła” (brute force approach). Podejście to polega na wypisaniu wynikającego z warunków interpolacji układu równań dla współczynników wielomianu $P_n(x_j) = y_j$, $j = 0, 1, \dots, n$. Układ ten ma postać

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_j & x_j^2 & \cdots & x_j^{n-1} & x_j^n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_j \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_j \\ \vdots \\ y_n \end{bmatrix}$$

W notacji macierzowo-wektorowej mamy

$$\mathbf{W}\mathbf{a} = \mathbf{y}$$

gdzie element macierzy \mathbf{W} (zwanej macierzą Van der Monda) dane są wzorem

$$w_{j,k} = x_j^k, \quad j, k = 0, 1, \dots, n$$

Otrzymany układ równań można (przynajmniej w teorii) rozwiązać za pomocą jednej ze standardowych metod (np. metodą eliminacji Gaussa). Można pokazać, że jeśli wszystkie węzły są różne, to macierz \mathbf{W} jest nieosobliwa i układ ma jednoznaczne rozwiązanie – wielomian interpolacyjny istnieje i jest jedyny.

Zauważmy, że stopień wielomianu interpolacyjnego P_n jest o jeden niższy niż liczba węzłów. W przeciwnym wypadku zagadnienie wyznaczenia wielomianu interpolacyjnego albo jest **nieokreślone** (gdy stopień wielomianu jest \geq liczby węzłów; wtedy istnieje nieskończenie wiele wielomianów interpolacyjnych) lub **nadokreślony** (gdy stopień wielomianu jest $<$ liczba węzłów minus jeden; wtedy układ jest na ogół sprzeczny i wielomian interpolacyjny nie istnieje)

Dobra wiadomość: istnieją metody sprytniejsze niż metoda "brutalna"! **Ich zastosowanie pozwala uniknąć rozwiązywania jakiegokolwiek układu równań.**

Zacznijmy od **metody Lagrange'a**. Jej główna idea polega na wykorzystaniu specjalnie skonstruowanych wielomianów interpolacyjnych, danych wzorem

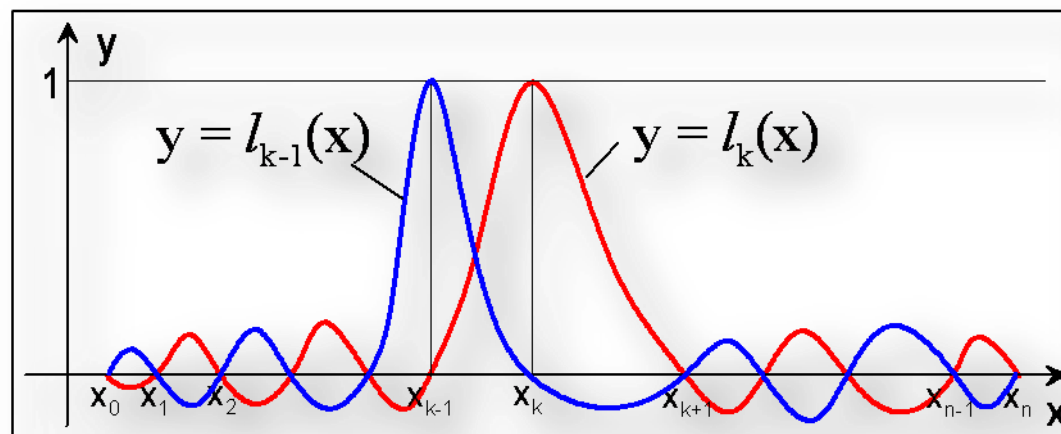
$$l_k(x) = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x-x_i}{x_k-x_i}, \quad k=0,1,\dots,n$$

Kluczowa własność tych wielomianów to

$$l_k(x_j) = \delta_{jk} = \begin{cases} 0 & \text{gdy } j \neq k \\ 1 & \text{gdy } j = k \end{cases}$$

*symbol
Kroneckera*

Wykresy takich wielomianów przedstawiają się następująco



Mając powyżej zdefiniowane wielomiany (interpolacyjne Lagrange'a; nie należy ich mylić w tzw. ortogonalnymi wielomianami Lagrange'a) rozwiązanie problemu interpolacji jest natychmiastowe. Wystarczy napisać

$$P_n(x) = \sum_{k=0}^n y_k l_k(x)$$

W istocie, weryfikacja warunków interpolacji daje następujący efekt

$$P_n(x_j) = \sum_{k=0}^n y_k l_k(x_j) = \sum_{k=0}^n y_k \delta_{jk} = y_j \quad , \quad j = 0, 1, \dots, n$$

Dla dociekliwych:

Alternatywna (ale równoważna) formuła dla wielomianu interpolacyjnego P_n ma postać

$$P_n(x) = \sum_{k=0}^n \frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)} y_k \quad (\text{wykazać !})$$

gdzie $\omega_{n+1}(x) = \prod_{k=0}^n (x-x_k) = (x-x_0)(x-x_1)\dots(x-x_n)$

Przybliżanie (aproksymacja) funkcji wielomianem interpolacyjnym

Wielomian interpolacyjny może być użyty do przybliżania innych funkcji. Załóżmy, że mamy dane węzły interpolacyjne

$$\{(x_j, y_j), j = 0, 1, \dots, n\} \quad \text{gdzie} \quad y_j = f(x_j) \quad , \quad j = 0, 1, \dots, n$$

Kluczowe pytanie: jaka jest dokładność przybliżenia (aproksymacji) zadanej funkcji f przez wielomian interpolacyjny P_n obliczony dla tych węzłów?

Ogólnej odpowiedzi na tak zadane pytanie udziela następujące twierdzenie.

Twierdzenie 1: Załóżmy, że $f \in C^{(n+1)}(I_x)$ dla pewnego przedziału I_x z jej dziedziny. Niech x_0, x_1, \dots, x_{n+1} będą zadanymi i różnymi węzłami interpolacyjnymi zawartymi w I_x i niech x oznacza dowolną liczbę w tym przedziale.

Wówczas istnieje taka liczba $\xi \in I_x$, że błąd aproksymacji funkcji f przez wielomian interpolacyjny P_n może być zapisany w postaci

$$E_n(x) \equiv f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x)$$

gdzie $\omega_{n+1}(x) = \prod_{k=0}^n (x - x_k) = (x - x_0)(x - x_1) \dots (x - x_n)$

Dowód:

Ustalmy x and rozważmy funkcję postaci

$$g(t) = E_n(t) - \frac{E_n(x)}{\omega_{n+1}(x)} \omega_{n+1}(t) \quad , \quad x \neq x_k \quad , \quad k = 0, 1, \dots, n$$

Twierdzimy, że funkcja g ma dokładnie $n+2$ miejsca zerowe (pierwiastki).

W istocie, mamy ...

$$g(x_k) = \underbrace{E_n(x_k)}_0 - \frac{E_n(x)}{\omega_{n+1}(x)} \underbrace{\omega_{n+1}(x_k)}_0 = 0, \quad k = 0, 1, \dots, n$$

$$g(x) = E_n(x) - \frac{E_n(x)}{\cancel{\omega_{n+1}(x)}} \cancel{\omega_{n+1}(x)} = 0$$

Skoro tak, to pochodna g' ma w przedziale I_x $n+1$ miejsc zerowych, pochodna g'' ma n miejsc zerowych, itd. Wreszcie, pochodna $g^{(n+1)}$ ma w przedziale I_x dokładnie jedno miejsce zerowe – oznaczmy je symbolem ξ

$$g^{(n+1)}(\xi) = 0, \quad \xi \in I_x$$

Co to oznacza? Otóż mamy:

$$0 = g^{(n+1)}(\xi) = E_n^{(n+1)}(\xi) - \frac{E_n(x)}{\omega_{n+1}(x)} \omega_{n+1}^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \underbrace{P_n^{(n+1)}(\xi)}_{=0 \text{ bo } P_n \text{ jest stopnia } n!} - \frac{E_n(x)}{\omega_{n+1}(x)} \underbrace{(n+1)!}_{\substack{\text{współczynnik} \\ \text{przy } x^{n+1} \text{ w } \omega_{n+1}(x) \\ \text{jest równy } 1}}$$

co po przekształceniu daje natychmiast formułę z tezy twierdzenia. **Koniec dowodu!**

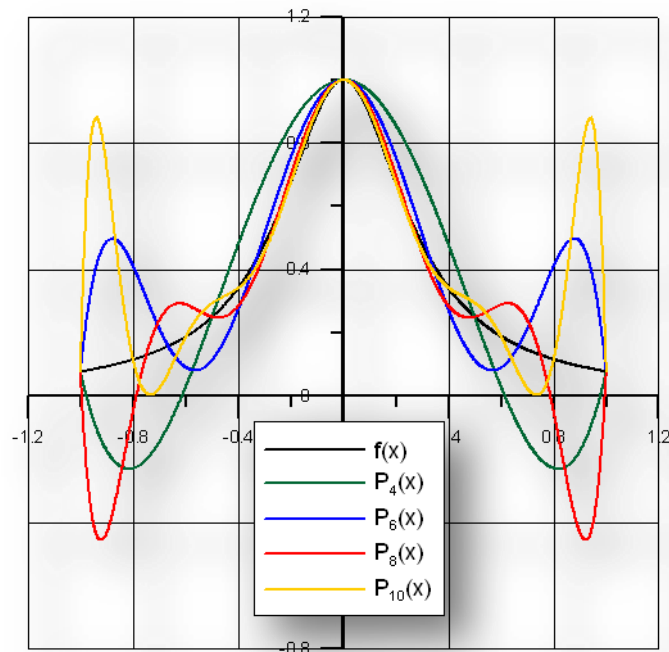
W przypadku szczególnym równoodległych węzłów interpolacyjnych mamy

$$x_k = x_0 + kh \quad , \quad k = 0, 1, \dots, n \quad , \quad h = \frac{x_n - x_0}{n}$$

Można pokazać, że dla węzłów równoodległych mają miejsce oszacowania

$$|\omega_{n+1}(x)| = \prod_{k=0}^n |x - x_k| \leq \frac{1}{4} h^{(n+1)} n! \quad \Rightarrow \quad |f(x) - P_n(x)| \leq \frac{h^{n+1}}{4(n+1)} \max_{\xi \in [x_0, x_n]} |f^{(n+1)}(\xi)|$$

Czy zwiększenie liczby użytych węzłów interpolacyjnych prowadzi do ulepszenia aproksymacji, tj. zmniejszenia różnicy pomiędzy oryginalną funkcją a przybliżającym ją wielomianem interpolacyjnym? **Niekoniecznie!**



Rozważmy aproksymację następującej funkcji wymiernej

$$f(x) = \frac{1}{1+10x^2}$$

na odcinku $[-1, 1]$ wielomianami interpolacyjnym rozpiętymi na węzłach równoodległych.

Efekt pokazuje obrazek.

Amplituda “oscylacji” wielomianu interpolacyjnego w pobliżu końców przedziału powiększa się ze wzrostem jego stopnia n . Błąd aproksymacji nie maleje, lecz wzrasta. Ściślej, ciąg liczbowy postaci

$$\varepsilon_n = \max_{x \in [-1,1]} |f(x) - P_n(x)|$$

jest rozbieżny. Jest to przykład tzw. **efektu Rungego**.

Lekarstwem na efekt Rungego (do pewnego stopnia) jest użycie węzłów rozmieszczonych nierównomiernie. Intuicja podpowiada, że w pobliżu końców przedziału interpolacji węzły powinny być rozmieszczone gęściej. Okazuje się, że istnieje **optymalny wybór węzłów!** Dla przedziału $[-1,1]$ są to liczby miejsca zerowe **wielomianu Czebyszewa** (2-ego rodzaju) stopnia $n+1$, tj.

$$x_k^T = \cos\left(\frac{2k+1}{n+1} \frac{\pi}{2}\right), \quad k = 0, 1, \dots, n$$

Wielomiany Czebyszewa definiuje następująca reguła rekurencyjna

$$T_0(x) \equiv 1, \quad T_1(x) = x$$

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad k = 1, 2, \dots$$

Na przykład: $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, $T_4(x) = 8x^4 - 8x^2 + 1$, itd. Zachodzi również następujący związek z funkcjami trygonometrycznymi

$$\cos kx = T_k(\cos x)$$

Z punktu widzenia aktualnego problemu, kluczowa własność wielomianów Czebyszewa to

$$\forall x \in [-1,1] |T_k(x)| \leq 1 \quad , \quad k = 0,1,\dots$$

Możemy zatem napisać $T_{n+1}(x) = 2^n \prod_{k=0}^n (x - x_k^T) \Rightarrow \left| \prod_{k=0}^n (x - x_k^T) \right| \leq \frac{1}{2^n}$

Głębokie twierdzenie mówi, że dla każdego innego wyboru $n+1$ punktów w przedziale $[-1,1]$ mamy zawsze

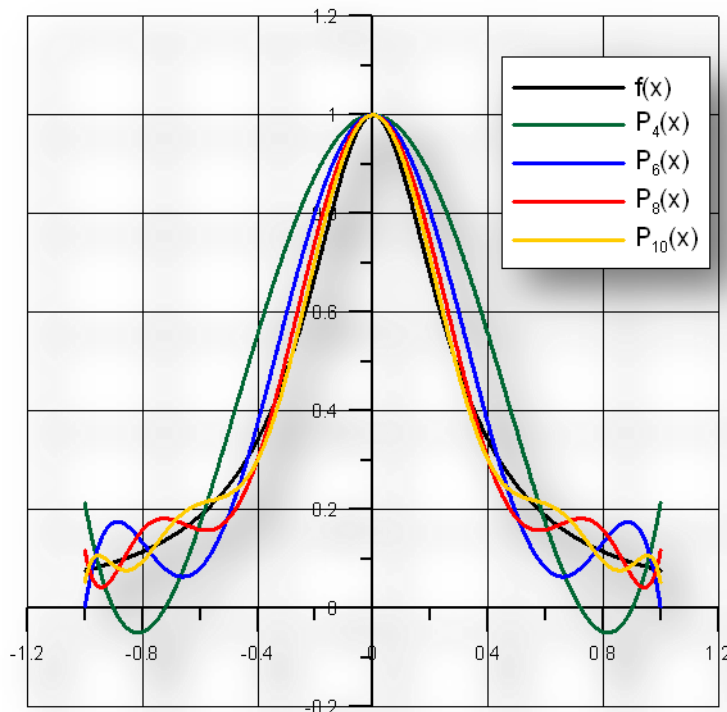
$$\max_{x \in [-1,1]} \left| \prod_{k=0}^n (x - z_k) \right| \geq \frac{1}{2^n} \quad , \quad z_k \in [-1,1] \quad , \quad k = 0,1,\dots,n$$

tj. wybór miejsc zerowych wielomianu $T_{n+1}(x)$ w roli węzłów interpolacyjnych minimalizuje wartość bezwzględną wielomianu $\omega_{n+1}(x)$ w przedziale $[-1,1]$.

Oszacowanie błędu aproksymacji przez wielomian interpolacyjny zbudowany na węzłach Czebyszewa ma postać

$$|f(x) - P_n(x)| \leq \frac{1}{2^n (n+1)!} \max_{\xi \in [-1,1]} |f^{(n+1)}(\xi)|$$

Zauważmy, że mianownik bardzo szybko maleje ze wzrostem stopnia wielomianu n . Oznacza to, że dobre przybliżenie funkcji f jest możliwe nawet wtedy, gdy maksimum modułu jej $(n+1)$ -ej pochodnej rośnie szybko z n . Jak pokazuje rysunek, wybór węzłów Czebyszewa eliminuje efekt Rungego w naszym przykładzie.



Dla przedziału $[a,b]$ węzły Czebyszewa definiujemy wzorem

$$x_k^T = \frac{b-a}{2} \cos\left(\frac{2k+1}{n+1} \frac{\pi}{2}\right) + \frac{a+b}{2}$$

a oszacowanie błędu aproksymacji ma postać

$$|f(x) - P_n(x)| \leq \frac{(b-a)^{n+1}}{2^{2n+1} (n+1)!} \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|$$

Konstrukcja wielomianu interpolacyjnego metodą Newtona

Na koniec przedstawimy alternatywną metodę wyznaczania wielomianu interpolacyjnego.

Jak poprzednio, mamy dane interpolacyjne

$$\{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$$

Konstruujemy sekwencje (tablicę) tzw. różnic dzielonych wg przedstawionych formuł. Należy zwrócić uwagę na sposób numerowania kolejnych różnic dzielonych na kolejnych poziomach.

$$\left\{ \begin{array}{l} Y_k = y_k \quad , \quad k = 0, 1, \dots, n \\ Y_{k,k+1} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{Y_{k+1} - Y_k}{x_{k+1} - x_k} \quad , \quad k = 0, 1, \dots, n-1 \\ Y_{k,k+1,k+2} = \frac{Y_{k+1,k+2} - Y_{k,k+1}}{x_{k+2} - x_k} \quad , \quad k = 0, 1, \dots, n-2 \\ \vdots \\ Y_{k,k+1,\dots,k+m} = \frac{Y_{k+1,k+2,\dots,k+m} - Y_{k,k+1,\dots,k+m-1}}{x_{k+m} - x_k} \quad , \quad k = 0, 1, \dots, n-m \\ \vdots \\ Y_{0,1,\dots,n} = \frac{Y_{1,2,\dots,n} - Y_{0,1,\dots,n-1}}{x_n - x_0} \quad , \quad m = n \end{array} \right.$$

Następnie, definiujemy rodzinę wielomianów

$$\omega_k = (x - x_0)(x - x_1)\dots(x - x_k) \equiv \prod_{j=0}^k (x - x_j) , k = 0, \dots, n-1$$

Wreszcie, wielomian interpolacyjny P_n jest konstruowany następująco

$$\begin{aligned} P_n(x) &= Y_0 + \sum_{k=1}^n Y_{0,1,\dots,k} \omega_{k-1}(x) = \\ &= y_0 + Y_{0,1}(x - x_0) + Y_{0,1,2}(x - x_0)(x - x_2) + \dots + Y_{0,1,2,\dots,n}(x - x_0)(x - x_2)\dots(x - x_{n-1}) \end{aligned}$$

Powyższa formuła jest nieoczywista, ale jej dowód jest dość długi i „techniczny”. Można do znaleźć w większości podręczników do analizy numerycznej. Ograniczmy się do pokazania jak działa wzór Newtona dla $n = 2$.

Zgodnie z tym wzorem, wielomian interpolacyjny dla węzłów $\{(x_0, y_0), (x_1, y_1), (x_2, y_2)\}$ ma postać

$$P_2(x) = Y_0 + Y_{0,1}(x - x_0) + Y_{0,1,2}(x - x_0)(x - x_1)$$

Pokażemy, że wielomian $P_2(x)$ istotnie spełnia warunki interpolacji tj.

$$P_2(x_j) = y_j \quad , \quad j = 0, 1, 2$$

Rachunki przebiegają następująco:

$$\left\{ \begin{array}{l} P_2(x_0) = y_0 \\ P_2(x_1) = y_0 + Y_{0,1}(x_1 - x_0) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cancel{(x_1 - x_0)} = y_1 \\ P_2(x_2) = y_0 + Y_{0,1}(x_2 - x_0) + Y_{0,1,2}(x_2 - x_0)(x_2 - x_1) = \\ = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x_2 - x_0) + \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{\cancel{x_2 - x_0}} \cancel{(x_2 - x_0)} (x_2 - x_1) = \\ = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x_2 - x_0) + y_2 - y_1 - \frac{y_1 - y_0}{x_1 - x_0} (x_2 - x_1) = \\ = y_0 + y_2 - y_1 + \frac{y_1 - y_0}{x_1 - x_0} (\cancel{x_2} - x_0 + x_1 - \cancel{x_2}) = y_2 \end{array} \right.$$

Algorytm Hornera

Efektywną numerycznie metodą obliczania wartości wielomianu interpolacyjnego danego w formie Newtona jest **algorytm Hornera**. Jego **pseudokod** można zapisać następująco

```
% Algorytm Hornera
% Wektor w przechowuje różnice dzielone
%  $w(k) = Y_{0,1,\dots,k}$  ,  $k = 0,1,\dots,n$ 
s ← W(n)
for k = n - 1 : 0 : -1 (pętla chodzi wstecz!)
    s ← s · (x - xk) + W(k)
end
return s
```

Ćwiczenie: napisz w języku C/C++ funkcję realizującą algorytm interpolacyjny Newtona.