

# Programowanie i metody numeryczne

Zadania – seria 6.

## Interpolacja.

*Interpolacja* to metoda generowania nowych punktów danych na podstawie danego skończonego zbioru takich punktów.

Rozważmy skończony,  $(n + 1)$ -elementowy zbiór punktów płaszczyzny

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n), \quad \text{gdzie } x_i \neq x_j \text{ dla } i \neq j.$$

Wiemy, że punkty te należą do wykresu pewnej nieznannej funkcji liczbowej  $f$ , czyli że  $y_k = f(x_k)$  dla  $0 \leq k \leq n$ . Naszym zadaniem jest znalezienie wartości tej funkcji w „nowym” punkcie  $x \neq x_k$  dla  $0 \leq k \leq n$ . Wykorzystamy w tym celu *funkcję interpolującą*  $\phi$ , która również spełnia  $y_k = \phi(x_k)$  dla  $0 \leq k \leq n$ . Funkcję tę wyznaczymy zakładając jej konkretną postać.

### Zadanie 1. linear – Interpolacja liniowa.

*Interpolacja liniowa* polega na wykorzystaniu wielomianów pierwszego stopnia. W celu znalezienia wartości funkcji interpolującej  $\phi$  w pewnym punkcie  $x$  szukamy dwóch węzłów  $x_i$  i  $x_j$ ,  $0 \leq i, j \leq n$ , które leżą najbliżej punktu  $x$ , a następnie konstruujemy wielomian liniowy  $\phi$  spełniający  $\phi(x_i) = y_i$  oraz  $\phi(x_j) = y_j$  (funkcja interpolująca  $\phi$  nie jest więc *de facto* wielomianem pierwszego stopnia, a złożeniem takich wielomianów na przedziałach wyznaczonych przez węzły interpolacji).

Napisz funkcję

```
double LinearInterpolation(const std::vector<std::pair<double, double>> &p,  
    double x)
```

obliczającą metodą interpolacji liniowej i zwracającą wartość funkcji interpolacyjnej dla argumentu  $x$  w oparciu o współrzędne punktów danych zapisane w wektorze  $p$ . Węzły interpolacji nie muszą być posortowane (tzn. nie można założyć, że  $x_i \leq x_j$  dla  $i < j$ ).

Korzystając z tej funkcji napisz program `linear`, który wczytuje ze standardowego wejścia liczbę rzeczywistą  $x$  z przedziału  $[0, 2\pi]$  oraz liczbę całkowitą  $n$ . Program powinien stworzyć tablicę wartości funkcji sinus w  $n$  różnych, równomiernie rozłożonych punktach z przedziału  $[0, 2\pi]$ , a następnie wypisać na standardowe wyjście przybliżoną wartość funkcji sinus dla argumentu  $x$  obliczoną metodą interpolacji liniowej na podstawie tych punktów, dokładną wartość funkcji sinus dla argumentu  $x$  oraz moduł różnicy tych wartości.

### Zadanie 2. lagrange – Interpolacja wielomianowa.

*Interpolacja wielomianowa* polega na wybraniu funkcji interpolującej w postaci wielomianu najmniejszego możliwego stopnia, którego wykres „przechodzi” przez wszystkie dane punkty. Mamy  $n + 1$  punktów, zatem będzie to wielomian stopnia  $n$ . Wielomian taki jest wyznaczony jednoznacznie i może zostać przedstawiony w postaci *wielomianu interpolacyjnego Lagrange’a*

$$w(x) = \sum_{i=0}^n y_i \prod_{j=0 \wedge j \neq i}^n \frac{x - x_j}{x_i - x_j}.$$

Napisz funkcję

```
double LagrangeInterpolation(const std::vector<std::pair<double, double>> &p,  
double x)
```

obliczającą metodą interpolacji wielomianowej i zwracającą wartość funkcji interpolacyjnej dla argumentu  $x$  w oparciu o współrzędne punktów danych zapisane w wektorze  $p$ . Przyjmij funkcję interpolującą w postaci wielomianu interpolacyjnego Lagrange'a.

Korzystając z tej funkcji napisz program `lagrange`, który wczytuje ze standardowego wejścia liczbę rzeczywistą  $x$  z przedziału  $[0, 2\pi]$  oraz liczbę całkowitą  $n$ . Program powinien stworzyć tablicę wartości funkcji sinus w  $n$  różnych, równomiernie rozłożonych punktach z przedziału  $[0, 2\pi]$ , a następnie wypisać na standardowe wyjście przybliżoną wartość funkcji sinus dla argumentu  $x$  obliczoną metodą interpolacji liniowej na podstawie tych punktów, dokładną wartość funkcji sinus dla argumentu  $x$  oraz moduł różnicy tych wartości.

### Zadanie 3. `neville` – Algorytm Neville'a.

*Algorytm Neville'a* to zaproponowana przez angielskiego matematyka Erica Harolda Neville'a rekurencyjna metoda wyznaczania wartości wielomianu interpolacyjnego. Jej opis można znaleźć w literaturze.

Napisz funkcję

```
double NevilleInterpolation(const std::vector<std::pair<double, double>> &p,  
double x)
```

obliczającą metodą interpolacji wielomianowej i zwracającą wartość funkcji interpolacyjnej dla argumentu  $x$  w oparciu o współrzędne punktów danych zapisane w wektorze  $p$ . W celu wyznaczenia wartości wielomianu interpolacyjnego funkcja powinna posłużyć się algorytmem Neville'a.

Korzystając z tej funkcji napisz program `neville`, który wczytuje ze standardowego wejścia liczbę rzeczywistą  $x$  z przedziału  $[0, 2\pi]$  oraz liczbę całkowitą  $n$ . Program powinien stworzyć tablicę wartości funkcji sinus w  $n$  różnych, równomiernie rozłożonych punktach z przedziału  $[0, 2\pi]$ , a następnie wypisać na standardowe wyjście przybliżoną wartość funkcji sinus dla argumentu  $x$  obliczoną metodą interpolacji liniowej na podstawie tych punktów, dokładną wartość funkcji sinus dla argumentu  $x$  oraz moduł różnicy tych wartości.

### Zadanie 4. `coeffs` – Współczynniki wielomianu interpolacyjnego.

Napisz funkcję

```
void NevilleCoeffs(const std::vector<std::pair<double, double>> &p,  
std::vector<double> &c)
```

wyznaczającą w oparciu o współrzędne punktów danych zapisane w wektorze  $p$  współczynniki wielomianu interpolacyjnego otrzymanego za pomocą algorytmu Neville'a.

Korzystając z tej funkcji napisz program `coeffs`, który wczytuje ze standardowego wejścia liczbę całkowitą  $n$ . Program powinien stworzyć tablicę wartości funkcji sinus w  $n$  różnych, równomiernie rozłożonych punktach z przedziału  $[0, 2\pi]$ , a następnie wypisać na standardowe wyjście wielomian interpolacyjny otrzymany za pomocą algorytmu Neville'a. Wielomian powinien zostać przedstawiony w czytelnej formie.

*Opracowanie: Bartłomiej Zglinicki.*