Zadanie numeryczne 6 – Kacper Drużdżel
Algorytm: Interpolacja Lagrange'a

Z wykładu wiadomo jak wygląda wzór:

Współczynniki wielomianu interpolacyjnego

Przypuśćmy, że mamy funkcję w postaci stabelaryzowanej, jak na stronie 11. Dodatkowo załóżmy, że żaden z węzłów interpolacji nie jest zerem. Wiemy, że wielomian interpolacyjny ma postać

$$y(x) = \sum_{j=1}^{n} l_j(x) f_j \,. \tag{12a}$$

Jednocześnie *wiemy*, że

$$y(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0 \,. \tag{12b}$$

Natychmiast widzimy, że współczynnik a_0 znajdziemy obliczając wartość wielomianu (12a) w zerze:

$$a_0 = y(0) = \sum_{j=1}^{n} l_j(0) f_j \,. \tag{13}$$

Więc tworzymy funkcję która liczy lj(x)

aby obliczyć a0

$$a_k = \sum_{j=1}^{n} l_j(0) f_j^{(k)} \,, \; k = 0, \ldots, n-1 \,, \tag{19a}$$

$$f_j^{(k)} = \begin{cases} \dfrac{f_j^{(k-1)} - a_{k-1}}{x_j} & k = 1, \ldots, n-1 \,, \\ f_j & k = 0 \,. \end{cases} \tag{19b}$$

Mając a0 możemy korzystać z następujących wzorów aby obliczyć kolejne fj^k,
którego następnie użyjemy aby obliczyć kolejne współczynniki ak

Złożoność algorytmu wynosi O(n^2), co jest znacznie szybsze niż pozostałe algorytmy O(n^3),
które rozwiązywałyby układ:

$$\begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

UWAGA!
ABY WYŚWIETLIĆ WYKRESY NALEŻY ZAINSTALOWAĆ GNUPLOT

$ sudo apt-get install gnuplot

następnie kompilacja programów:

g++ -o zad6.x zad6.cpp

oraz

g++ -o wykresy.x wykresy.cxx

Uruchomienie:
aby wyświetlić wyniki:
./zad6.x

aby wyświetlić wykres: (plik tekstowy wyniki.txt tworzony jest aby umieścić punkty na wykresie)
./wykresy.x