Pracownia z Analizy Numercznej (M)

Sprawozdanie do zadania **P2.20**
Mateusz Basiak
nr indeksu: 300487
Wrocław, 13.12.2018r.

1.Wstęp

Z definicji[1] wielomian to wyrażenie algebraiczne będące skończoną sumą jednomianów. Może więc zawsze być przedstawiony w postaci sumy jednomianów kolejnych stopni z pewnymi współczynnikami. Jest to jednak tylko jeden ze sposobów przedstawiania wielomianu jako skończonej sumy wielomianów składowych ze współczynnikami, to znaczy w postaci $\sum_{n=0}^{N} a_n p_n(x)$, gdzie $p_n(x)$ to ciąg wielomianów, a a_n to współczynniki. Jeśli wielomiany $\{p_n\}$ są liniowo niezależne, to współczynniki są wyznaczone jednoznacznie. Czasem jednak, z różnych powodów (złożoność obliczania wartości wielomianu w punkcie, unikalne własności konkretnego przedstawienia itd.) chcialibyśmy umieć przechodzić pomiędzy różnymi przedstawieniami tej postaci, tzn. na podstawie pewnej wiedzy o dwóch ciągach wielomianów $\{p_n\}$ i $\{q_n\}$ oraz znajomości współczynników A_n takich, że $f_N = \sum_{n=0}^{N} A_n p_n(x)$ umieć obliczyć odpowiednie współczynniki A_n^* dla wielomianów $\{q_n\}$. W swojej pracy przedstawię i udowodnię algorytm obliczający te współczynniki oraz pokażę przykłady jego zastosowania.

2. Opis teoretyczny rozwiązania

2.1. Opis problemu

Pewną funkcję f_N da się przedstawić jednoznacznie w bazach $\{p_n\}$ i $\{q_n\}$, tzn.

$$f_N = \sum_{n=0}^{N} A_n p_n(x) = \sum_{n=0}^{N} A_n^* q_n(x) \tag{1}$$

Wiemy również, że ciągi $\{p_n\}$ i $\{q_n\}$ spełniają następujące zależności rekurencyjne ($0 \le n \le N-1$):

$$p_{n+1} + (a_n + b_n x)p_n + c_n p_{n-1} = 0, \quad p_{-1} = 0, \quad p_0 = 1 \tag{2}$$

$$q_{n+1} + (a_n^* + b_n^* x)q_n + c_n^* q_{n-1} = 0, \quad q_{-1} = 0, \quad q_0 = 1 \tag{3}$$

Mając teraz dane ciągi $\{a_n\}_{n=0}^{N-1}, \{b_n\}_{n=0}^{N-1}, \{c_n\}_{n=0}^{N-1}, \{a_n^*\}_{n=0}^{N-1}, \{b_n^*\}_{n=0}^{N-1}, \{c_n^*\}_{n=0}^{N-1}, \{A_n\}_{n=0}^{N}$ należy wyznaczyć ciąg $\{A_n^*\}_{n=0}^{N}$.

2.2. Rozwiązanie

Podany pomysł został opisany w książce J. Wimpa. [2]

Zauważmy na początek, że zgodnie z treścią zadania jedyne informacje, jakie mamy o wielomianach to podane ciągi współczynników. Okazuje się, że to wystarczy - nie potrzebujemy żadnych wiadomości o strukturze wielomianów $\{p_n\}$ i $\{q_n\}$. Zdefiniujmy teraz ciąg wielomianów pomocniczych B_n, określony wzorem rekurencyjnym:

$$\begin{cases} B_N = A_N \\ B_{N-1} = -(a_{N-1} + b_{N-1}x)A_N + A_{N-1} \\ B_n = -(a_n + b_n x)B_{n+1} - c_{n+1}B_{n+2} + A_n \quad 0 \le n \le N-2 \end{cases} \tag{4}$$

2.2.1 Lemat 1

Dla tak zdefiniowanego ciągu B_n zachodzi $B_0 = f_N$.

Dowód. Z definicji rekurencyjnej $\{B_n\}$ mamy:

$$\begin{cases} A_N = B_N \\ A_{N-1} = B_{N-1} + (a_{N-1} + b_{N-1}x)B_N \\ A_n = B_n + (a_n + b_n x)B_{n+1} + c_{n+1}B_{n+2} \quad 0 \le n \le N-2 \end{cases}$$

Stąd:

$$f_N = \sum_{n=0}^{N} A_n p_n(x) = B_N p_N + (B_{N-1} + (a_{N-1} + b_{N-1}x)B_N)p_{N-1} + \sum_{n=0}^{N-2}(B_n + (a_n + b_n x)B_{n+1} + c_{n+1}B_{n+2})p_n =$$

$$= B_N p_N + B_{N-1}p_{N-1} + (a_{N-1} + b_{N-1}x)B_N p_{N-1} + \sum_{n=0}^{N-2} B_n p_n + \sum_{n=0}^{N-2}(a_n + b_n x)B_{n+1}p_n + \sum_{n=0}^{N-2} c_{n+1}B_{n+2}p_n =$$

$$= \sum_{n=0}^{N} B_n p_n + \sum_{n=0}^{N-1}(a_n + b_n x)B_{n+1}p_n + \sum_{n=0}^{N-2} c_{n+1}B_{n+2}p_n =$$

$$= \sum_{n=2}^{N} B_n p_n + \sum_{n=1}^{N-1}(a_n + b_n x)B_{n+1}p_n + \sum_{n=0}^{N-2} c_{n+1}B_{n+2}p_n + B_0 p_0 + B_1 p_1 + (a_0 + b_0 x)B_1 p_0 =$$

$$= \sum_{n=0}^{N-2} B_{n+2}p_{n+2} + \sum_{n=0}^{N-2}(a_{n+1} + b_{n+1}x)B_{n+2}p_{n+1} + \sum_{n=0}^{N-2} c_{n+1}B_{n+2}p_n + B_0 p_0 + B_1(p_1 + (a_0 + b_0 x)p_0) =$$

$$= \sum_{n=0}^{N-2} B_{n+2}(p_{n+2} + (a_{n+1} + b_{n+1}x)p_{n+1} + c_{n+1}p_n) + B_0 p_0 + B_1(p_1 + (a_0 + b_0 x)p_0)$$

Jednakże z własności rekurencyjnej (2) wiemy, że każdy z wyrazów ostatniej sumy się zeruje. Co więcej $p_1 + (a_0 + b_0 x)p_0$ również jest równe zero, jak wynika z tej rekurencji. Dostajemy więc $f_N = B_0 p_0$, ale $p_0 = 1$, stąd teza lematu. □

Zdefiniujmy teraz zbiór współczynników $\{A_n^{(k)}\}$ w ten sposób, że dla każdego $k = 0, 1, 2, \cdots, N$ zachodzi:

$$B_{N-k} = \sum_{n=0}^{k} A_n^{(k)} q_n \tag{5}$$

Zauważmy, że takie przedstawienie jest możliwe, ponieważ B_N jest wielomianem stopnia zerowego, a każdy kolejny krok rekurencji tworzenia B_n zwiększa stopień wielomianu co najwyżej o 1 (jeśli b_n jest niezerowe), więc wielomian B_{N-k} jest co najwyżej stopnia k, czyli można skonstruować go za pomocą $k+1$ pierwszych czynników z bazy $\{q_n\}$. Co więcej dla $k = N$ dostajemy z Lematu 1:

$$\sum_{n=0}^{N} A_n^{(N)} q_n = B_0 = f_N = \sum_{n=0}^{N} A_n^* q_n$$

Wynika z tego, że współczynniki $A_n^{(N)}$ to szukane przez nas współczynniki A_n^*, trzeba tylko znaleźć sposób na ich wyznaczanie. Pokażę teraz rekurencyjny sposób wyliczania tych współczynników.

2.2.2 Podstawa rekurencji

Współczynniki będę wyznaczał rekurencyjnie względem indeksu górnego k. W wyliczeniu $\{A_n^{(k+1)}\}_{n=0}^{k+1}$ będę używał $\{A_n^{(k)}\}_{n=0}^{k}$ i $\{A_n^{(k-1)}\}_{n=0}^{k-1}$, stąd na początek muszę wyznaczyć wartości współczynników dla $k = 0$ i $k = 1$. Najpierw $k = 0$:

$$B_N = \sum_{n=0}^{0} A_n^{(0)} q_n = A_0^{(0)} q_0 = A_0^{(0)}$$

Czyli $A_0^{(0)} = A_N$. Dla $k = 1$ jest nieco trudniej. Zauważmy na początek, że z rekurencji (3) wynika $q_1 + (a_0^* + b_0^* x)q_0 + c_0^* q_{-1} = 0$, czyli $q_1 = -(a_0^* + b_0^* x)q_0 = -(a_0^* + b_0^* x)$. Mamy więc:

$$B_{N-1} = \sum_{n=0}^{1} A_n^{(1)} q_n = A_0^{(1)} q_0 + A_1^{(1)} q_1 = A_0^{(1)} - A_1^{(1)}(a_0^* + b_0^* x) = A_0^{(1)} - A_1^{(1)} a_0^* - A_1^{(1)} b_0^* x$$

Z definicji (4) mamy zaś $B_{N-1} = -(a_{N-1}+b_{N-1}x)A_N + A_{N-1} = A_{N-1} - a_{N-1}A_N - b_{N-1}A_N x$. Grupując czynniki przy zmiennej x oraz wyrazy wolne dostajemy układ równań:

$$\begin{cases} A_{N-1} - a_{N-1}A_N = A_0^{(1)} - A_1^{(1)}a_0^* \\ b_{N-1}A_N = A_1^{(1)}b_0^* \end{cases}$$

Jego rozwiązaniem są współczynniki: $A_0^{(1)} = -a_{N-1}A_N + A_{N-1} + \dfrac{a_0^* b_{N-1}A_N}{b_0^*}$ oraz $A_1^{(1)} = \dfrac{b_{N-1}A_N}{b_0^*}$

2.2.3 Krok rekursji

Zgodnie z definicją (5) wiemy, że dla każdego $k = 2, 3, \cdots, N$:

$$B_{N-(k-2)} = \sum_{n=0}^{k-2} A_n^{(k-2)} q_n \qquad B_{N-(k-1)} = \sum_{n=0}^{k-1} A_n^{(k-1)} q_n \qquad B_{N-k} = \sum_{n=0}^{k} A_n^k q_n$$

Gdzie współczynniki w dwóch pierwszych równościach znamy, a te w trzeciej chcielibyśmy na ich podstawie obliczyć. Z tożsamości rekurencyjnej (3) dostajemy $-xb_n^* q_n = q_{n+1} + a_n^* q_n + c_n^* q_{n-1}$ czyli $xq_n = -\dfrac{1}{b_n^*}q_{n+1} - \dfrac{a_n^*}{b_n^*}q_n - \dfrac{c_n^*}{b_n^*}q_{n-1}$ Z definicji (4) otrzymujemy teraz:

$$B_{N-k} = -(a_{N-k} + b_{N-k}x)B_{N-(k-1)} - c_{N-(k-1)}B_{N-(k-2)} + A_{N-k} =$$

$$= -(a_{N-k} + b_{N-k}x)\sum_{n=0}^{k-1} A_n^{(k-1)} q_n - c_{N-(k-1)}\sum_{n=0}^{k-2} A_n^{(k-2)} q_n + A_{N-k} =$$

$$= -\sum_{n=0}^{k-1} a_{N-k}A_n^{(k-1)}q_n - \sum_{n=0}^{k-1} xq_n b_{N-k}A_n^{(k-1)} - \sum_{n=0}^{k-2} c_{N-(k-1)}A_n^{(k-2)}q_n + A_{N-k} =$$

$$= -\sum_{n=0}^{k-1} a_{N-k}A_n^{(k-1)}q_n - \sum_{n=0}^{k-1}\left(-\frac{1}{b_n^*}q_{n+1} - \frac{a_n^*}{b_n^*}q_n - \frac{c_n^*}{b_n^*}q_{n-1}\right) b_{N-k}A_n^{(k-1)} - \sum_{n=0}^{k-2} c_{N-(k-1)}A_n^{(k-2)}q_n + A_{N-k} =$$

$$= -\sum_{n=0}^{k-1} a_{N-k}A_n^{(k-1)}q_n + \sum_{n=0}^{k-1}\frac{1}{b_n^*}b_{N-k}A_n^{(k-1)}q_{n+1} + \sum_{n=0}^{k-1}\frac{a_n^*}{b_n^*}b_{N-k}A_n^{(k-1)}q_n + \sum_{n=0}^{k-1}\frac{c_n^*}{b_n^*}b_{N-k}A_n^{(k-1)}q_{n-1} -$$

$$- \sum_{n=0}^{k-2} c_{N-(k-1)}A_n^{(k-2)}q_n + A_{N-k} =$$

$$= -\sum_{n=0}^{k-1} a_{N-k}A_n^{(k-1)}q_n + \sum_{n=1}^{k}\frac{1}{b_{n-1}^*}b_{N-k}A_{n-1}^{(k-1)}q_n + \sum_{n=0}^{k-1}\frac{a_n^*}{b_n^*}b_{N-k}A_n^{(k-1)}q_n + \sum_{n=-1}^{k-2}\frac{c_{n+1}^*}{b_{n+1}^*}b_{N-k}A_{n+1}^{(k-1)}q_n -$$

$$- \sum_{n=0}^{k-2} c_{N-(k-1)}A_n^{(k-2)}q_n + A_{N-k} =$$

$$= -\sum_{n=1}^{k-2} a_{N-k}A_n^{(k-1)}q_n + \sum_{n=1}^{k-2}\frac{1}{b_{n-1}^*}b_{N-k}A_{n-1}^{(k-1)}q_n + \sum_{n=1}^{k-2}\frac{a_n^*}{b_n^*}b_{N-k}A_n^{(k-1)}q_n + \sum_{n=1}^{k-2}\frac{c_{n+1}^*}{b_{n+1}^*}b_{N-k}A_{n+1}^{(k-1)}q_n -$$

$$- \sum_{n=1}^{k-2} c_{N-(k-1)}A_n^{(k-2)}q_n + A_{N-k} - a_{N-k}A_0^{(k-1)}q_0 - a_{N-k}A_{k-1}^{(k-1)}q_{k-1} + \frac{1}{b_{k-1}^*}b_{N-k}A_{k-1}^{(k-1)}q_k + \frac{1}{b_{k-2}^*}b_{N-k}A_{k-2}^{(k-1)}q_{k-1} +$$

$$+ \frac{a_0^*}{b_0^*}b_{N-k}A_0^{(k-1)}q_0 + \frac{a_{k-1}^*}{b_{k-1}^*}b_{N-k}A_{k-1}^{(k-1)}q_{k-1} + \frac{c_0^*}{b_0^*}b_{N-k}A_0^{(k-1)}q_{-1} + \frac{c_1^*}{b_1^*}b_{N-k}A_1^{(k-1)}q_0 - c_{N-(k-1)}A_0^{(k-2)}q_0 =$$

$$= \sum_{n=1}^{k-2}\left(-a_{N-k}A_n^{(k-1)} + \frac{1}{b_{n-1}^*}b_{N-k}A_{n-1}^{(k-1)} + \frac{a_n^*}{b_n^*}b_{N-k}A_n^{(k-1)} + \frac{c_{n+1}^*}{b_{n+1}^*}b_{N-k}A_{n+1}^{(k-1)} - c_{N-(k-1)}A_n^{(k-2)}\right)q_n +$$

$$+ \left(A_{N-k} - a_{N-k}A_0^{(k-1)} + \frac{a_0^*}{b_0^*}b_{N-k}A_0^{(k-1)} + \frac{c_1^*}{b_1^*}b_{N-k}A_1^{(k-1)} - c_{N-(k-1)}A_0^{(k-2)}\right)q_0 +$$

$$+\left(\frac{1}{b^*_{k-2}}b_{N-k}A^{(k-1)}_{k-2}+\frac{a^*_{k-1}}{b^*_{k-1}}b_{N-k}A^{(k-1)}_{k-1}-a_{N-k}A^{(k-1)}_{k-1}\right)q_{k-1}+\frac{1}{b^*_{k-1}}b_{N-k}A^{(k-1)}_{k-1}q_k$$

Wyrażenie stojące przy wielomianie q_i jest w tej równości zatem równe współczynnikowi $A^{(k)}_i$. Otrzymaliśmy w ten sposób wzory na wszystkie współczynniki $\{A^{(k)}_n\}^k_{n=0}$ zależne tylko od zmiennych, które już wcześniej mieliśmy wyznaczone.

2.3. Algorytm

Powyższe rozumowanie doprowadziło nas do następującego algorytmu obliczania współczynników A^*_n:

$$\begin{cases} A^{(0)}_0 = A_N \\ A^{(1)}_0 = -a_{N-1}A_N + A_{N-1} + \dfrac{a^*_0 b_{N-1}A_N}{b^*_0} \\ A^{(1)}_1 = \dfrac{b_{N-1}A_N}{b^*_0} \\ A^{(k)}_0 = A_{N-k} - a_{N-k}A^{(k-1)}_0 + \dfrac{a^*_0}{b^*_0}b_{N-k}A^{(k-1)}_0 + \dfrac{c^*_1}{b^*_1}b_{N-k}A^{(k-1)}_1 - c_{N-(k-1)}A^{(k-2)}_0 \\ A^{(k)}_{k-1} = \dfrac{1}{b^*_{k-2}}b_{N-k}A^{(k-1)}_{k-2} + \dfrac{a^*_{k-1}}{b^*_{k-1}}b_{N-k}A^{(k-1)}_{k-1} - a_{N-k}A^{(k-1)}_{k-1} \\ A^{(k)}_k = \dfrac{1}{b^*_{k-1}}b_{N-k}A^{(k-1)}_{k-1} \\ A^{(k)}_n = -a_{N-k}A^{(k-1)}_n + \dfrac{1}{b^*_{n-1}}b_{N-k}A^{(k-1)}_{n-1} + \dfrac{a^*_n}{b^*_n}b_{N-k}A^{(k-1)}_n + \dfrac{c^*_{n+1}}{b^*_{n+1}}b_{N-k}A^{(k-1)}_{n+1} - c_{N-(k-1)}A^{(k-2)}_n \end{cases} \tag{6}$$

dla $k = 2, 3, \cdots, N$ i $n = 1, 2, \cdots, k-2$. Otrzymane współczynniki A^N_n to szukane A^*_n. Algorytm dla każdego k od 0 do N musi policzyć $k+1$ współczynników, z których każdy oblicza w czasie stałym, więc czas jego działania to $O(N^2)$.

3. Opis programu

Wszystkie doświadczenia wykonywane były za pomocą programu zaimplementowanego w języku Julia v.1.0.1. Program ten uruchamiano na komputerze z procesorem Intel Core i5, 1.60GHz, 4 GB RAM, długość słowa maszynowego 64 bit. Precyzja arytmetyki to $u = 1,11 \cdot 10^{-16}$.Program, umieszczony w pliku "*program.jl*", składa się z następujących funkcji:

$algorytm(N, a, b, c, a\_, b\_, c\_, A)$

Główna funkcja algorytmu wyliczająca wynik. Wywołuje ona pozostałe funkcje z odpowiednimi argumentami. Przyjmuje ona argumenty: N - stopień wielomianu; a,b,c - tablice rozmiaru N zawierające współczynniki równania rekurencyjnego (2); $a\_$,$b\_$,$c\_$ - tablice rozmiaru N zawierające współczynniki równania rekurencyjnego (3); A - tablica rozmiaru $N+1$ zawierająca współczynniki A_n z równania (1). Zwraca tablicę rozmiaru $N+1$ z obliczonymi współczynnikami.

$podstawa\_rekursji(N, a, b, c, a\_, b\_, c\_, A)$

Funkcja obliczająca $A^{(1)}_0$ i $A^{(1)}_1$. Wywoływana przez funkcję *algorytm* i przyjmująca takie same argumenty. Zwraca ona tablicę rozmiaru $N+1$, której dwa pierwsze elementy to $A^{(1)}_0$ i $A^{(1)}_1$.

$rekursja(N, a, b, c, a\_, b\_, c\_, A, k, A\_2, A\_1)$

Funkcja obliczająca współczynniki $A^{(k)}_i$ dla danego k i zwracająca tablicę rozmaru $N+1$ zawierającą te współczynniki na miejscach od 1 do $k+1$. Jest ona wywoływana wielokrotnie przez funkcję *algorytm*. Pierwsze osiem jej argumentów jest identycznych z argumentami funkcji *algorytm*. Dziewiąty to liczba k. Dwa ostatnie - A_2, A_1 to tablice rozmiaru $N+1$, zawierające na swoich początkowych miejscach odpowiednio współczynniki $\{A^{(k-2)}_n\}^{k-2}_{n=0}$ oraz $\{A^{(k-1)}_n\}^{k-1}_{n=0}$.

4. Doświadczenia

4.1. Zamiana postaci potęgowej na postać Czebyszewa

Postać potęgowa to najbardziej naturalna postać wielomianu, w której ciąg $\{p_n\}$ ma postać $p_n = x^n$. Współczynniki A_n są wtedy współczynnikami w jego najbardziej powszechnej postaci: $\sum_{n=0}^{N} A_n x^n$. Również współczynniki rekurencji nietrudno wyliczyć, gdyż dla każdego n ma ona postać:

$$x^n + (a_n + b_n x)x^{n-1} + c_n x^{n-2} = 0$$

Widać stąd, że $a_n = c_n = 0$ oraz $b_n = -1$ dla każdego n.
Przy wielomianach Czebyszewa wyliczenie współczynników rekurencyjnych jest nieco (ale tylko nieco) trudniejsze. Przypomnijmy, że wielomiany Czebyszewa to wielomiany postaci:

$$\begin{cases} T_0 = 1 \\ T_1 = x \\ T_k = 2x \cdot T_{k-1} - T_{k-2} \quad 2 \le k \le N \end{cases} \tag{7}$$

Daje to następujące równania rekurencyjne:

$$\begin{cases} T_1 + (a_1 + b_1 x)T_0 = x + a_1 + b_1 \cdot x = 0 \\ T_k + (a_k + b_k x)T_{k-1} + c_k T_{k-2} = (2x + a_k + b_k x) \cdot T_{k-1} + (c_k - 1) \cdot T_{k-2} = 0 \quad 2 \le k \le N \end{cases}$$

Z powyższego układu równań widać już, że dla wielomianów Czebyszewa współczynniki rekurencyjne wynoszą $b_0 = -1$, $b_n = -2$ dla $n > 1$, $c_k = 1$, $a_k = 0$ dla każdego k.

W pliku "*program.ipynb*" znajdują się dwa testy zamiany pomiędzy tymi postaciami. W teście pierwszym wpisane zostały ręcznie wielomiany Czebyszewa $T_0, \cdots T_5$. Dla czterech wielomianów podanych w postaci potęgowej zostały wyliczone ich współczynniki w bazie wielomianów Czebyszewa. (Wielomiany te w bazie wielomianów potęgowych mają współczynniki postaci kolejno: $A_i = 1$, $A_i = i$, $A_i = (-1)^i$ oraz $A_i = \frac{1}{2^i}$). Sprawdzenie poprawności algorytmu dokonało się poprzez wymnożenie otrzymanych współczynników przez odpowiednie wielomiany Czebyszewa i porównanie otrzymanych współczynników z tymi uzyskanymi na wejściu.

W teście drugim algorytm otrzymuje wielomian $\sum_{n=0}^{N} x^n$ i wylicza jego współczynniki w bazie wielomianów Czebyszewa. Następnie otrzymuje on wyliczone współczynniki i wylicza współczynniki tego wielomianu ponownie w bazie standardowych wielomianów potęgowych (wykonuje przekształcenie $'odwrotne'$). Sprawdzenie dokonuje się poprzez porównanie obliczonych współczynników z tymi otrzymanymi na wejściu. Czynności te powtarzane są dla kolejnych wartości N, od 2 do 50.

Dla wszystkich zaprezentowanych przeze mnie testów algorytm działa poprawnie i zwraca dokładne wyniki. Warto odnotować, że wyniki są dokładne także dla testu z wartościami A_i niecałkowitymi.

4.2. Potęgi dwumianu $(x + r)^n$ i wielomiany Czebyszewa

Oznaczmy $p_n = (x+r)^n$ dla pewnego $r \in \mathbb{R}$ oraz $n \in \mathbb{N}$. Ciąg wielomianów $\{p_n\}$ ma stale rosnący stopień i może być określony zależnością rekurencyjną podaną w zadaniu, której współczynniki wyznacza wzór:

$$p_k + (a_k + b_k x)p_{k-1} + c_k p_{k-2} = ((x + r)^2 + (a_k + b_k x)(x + r) + c_k) \cdot (x + r)^{n-2} = 0$$

Wynika stąd następujący układ równań ($1 \le k \le N$):

$$\begin{cases} 1 + b_k = 0 \\ 2r + a_k + b_k r = 0 \\ r^2 + a_k r + c_k = 0 \end{cases}$$

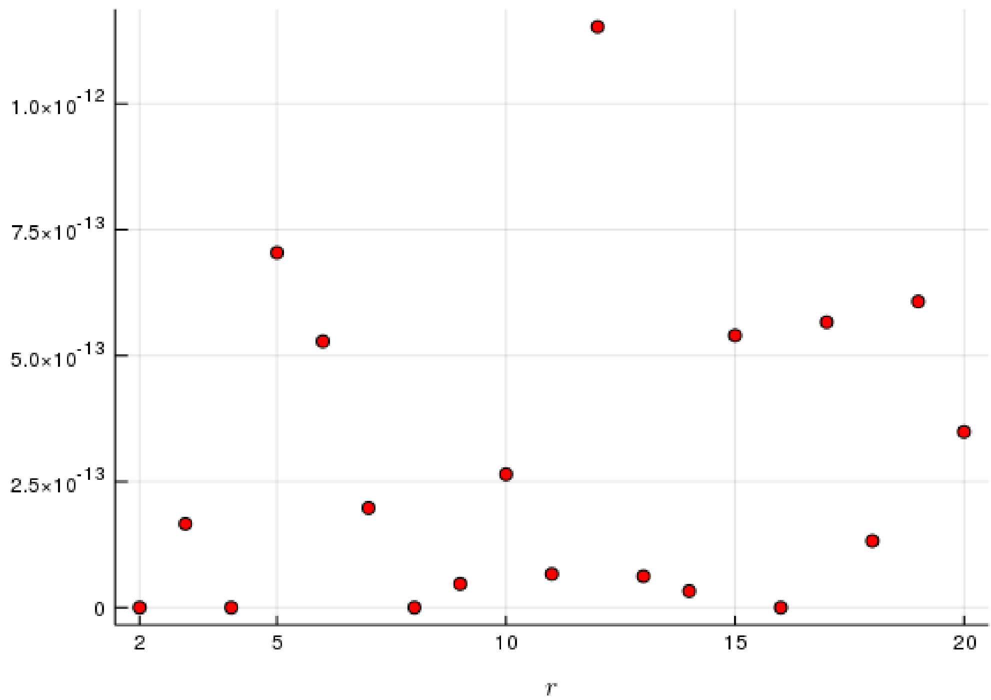
Który po kilku przekształceniach daje współczynniki rekurencyjne:

$$\begin{cases} a_k = -r \\ b_k = -1 \\ c_k = 0 \end{cases}$$

Testy trzeci, czwarty i piąty w pliku "*program.ipynb*" odnoszą się do obliczania współczynników w bazie $\{p_n\}$ potęg dwumianu $(x+r)^n$. We wszystkich używane są ciągi wielomianów stopni od zero do dziesięć ($n \leq 10$). Zmienną różniącą test trzeci od pozostałych jest r. W teście trzecim przyjmuje ona wartości całkowite $r = -10, -9, \cdots, -1, 1, \cdots, 10$, natomiast w czwartym i piątym wartości ułamkowe $r = \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{20}$. W dwóch pierwszych testach współczynniki A_i są równe $i+1$ dla każdego $0 \leq i \leq 10$, natomiast w ostatnim $A_i = 10^{-i}$. Tym razem będę wyliczał współczynniki korzystając z tych w bazie wielomianów Czebyszewa. Ma to podnieść jakość testów przez urozmaicenie przypadków testowych. Podobnie jak w teście drugim sprawdzenie odbywa się poprzez wykorzystanie algorytmu powtórnie do wyliczenia współczynników w postaci, od której zaczęliśmy. Następnie są one porównywane ze współczynnikami danymi na początku by określić, czy są poprawne i jaki jest ewentualny błąd.

W teście trzecim algorytm oblicza wszystkie współczynniki poprawnie.

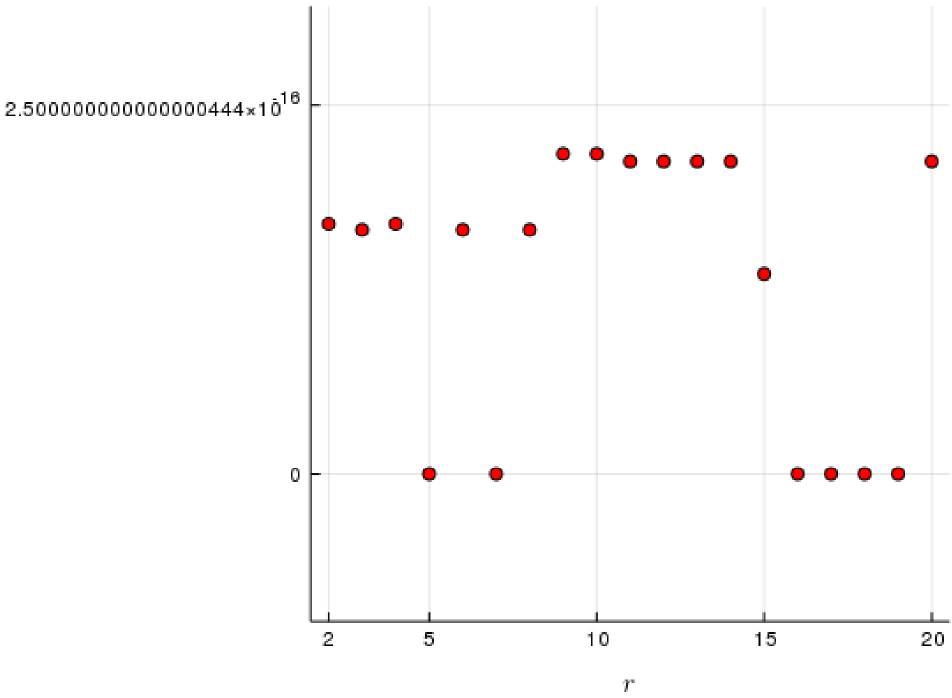
W teście czwartym pojawiają się niedokładności związane z obliczeniami na liczbach niecałkowitych. Błędy obliczeniowe przedstawione zostały na wykresie:



Maksymalny błąd względny obliczenia współczynnika wielomianu dla danego r i współczynników A_i rzędu liczb naturalnych do 11

Na podstawie powyższego wykresu można poczynić kilka obserwacji. Po pierwsze obliczenia są najdokładniejsze, gdy r jest potęgą dwójki, co może tłumaczyć brak błędów numerycznych w teście pierwszym. Po drugie błędy względne (poza jednym) mieszczą się w przedziale $\left[0, 10^{-12}\right]$.

W teście piątym również pojawiły się błędy numeryczne związane z użyciem liczb niecałkowitych. Jednak, jak widać na rycinie na następnej stronie, poprzez fakt, że współczynniki są w tym teście bardzo małe, błąd względny zmalał do wielkości porównywalnych z dokładnością arytmetyki. Widać więc, że dokładność wyniku jest na tyle duża, na ile tylko pozwala implementacja liczb zmiennoprzecinkowych.



Maksymalny błąd względny obliczenia współczynnika wielomianu dla danego r i niewielkich współczynników A_i

5.Wnioski

W mojej pracy przedstawiłem algorytm rozwiązujący zadany problem wraz z jego uzasadnieniem. Algorytm działa w czasie kwadratowym względem długości wielomianu wynikowego, ale złożoność zależności obliczanych wspólczynników od danych wejściowych pozwala postawić hipotezę, że lepsza złożoność jest bardzo trudna, jeśli nie niemożliwa do uzyskania.

Jak wynika z przeprowadzonych testów, algorytm oblicza współczynniki poprawnie i z dużą dokładnością, niezależnie od współczynników i stopnia wielomianu oraz postaci, między którymi jest on konwertowany. Błędy wynikają głównie z niedokładności arytmetyki i są niewielkiego rzędu w porównaniu do wyniku.

Literatura

[1] Za Wikipedią: https://pl.wikipedia.org/wiki/Wielomian

[2] J. Wimp, Computation with Recurrence Relations, Pitman Advanced Publishing Program, Boston, 1984