II. NUMERYCZNA REALIZACJA OBLICZEŃ NA WIELOMIANACH I FUNKCJACH WYMIERNYCH

2.1. Reprezentacje numeryczne wielomianów i funkcji wymiernych

W wielu zagadnieniach numerycznych (m. in. w interpolacji, aproksymacji, całkowaniu numerycznym, rozwiązywaniu zagadnienia początkowego równań różniczkowych zwyczajnych) wykorzystuje się w różny sposób wielomiany i funkcje wymierne. Sposób reprezentacji takich funkcji jest istotny dla optymalizacji kosztu obliczeń w tych zagadnieniach.

Wielomiany i funkcje wymierne mogą być reprezentowane w komputerze w trojaki sposób:

- w postaci naturalnej,
- w postaci Newtona,
- jako kombinacja liniowa wielomianów ortogonalnych.

W postaci naturalnej wykorzystuje się następujące wzory dla przedstawienia wielomianu $w(x)$ i funkcji wymiernej $v(x)$:

$$w(x) = \sum_{i=1}^{n} a_i x^i, \quad v(x) = \frac{\sum\limits_{i=0}^{n} a_i x^i}{\sum\limits_{i=0}^{m} b_i x^i}.$$

Dla reprezentacji wielomianu $w(x)$ w tej postaci potrzeba $n + 1$ miejsc pamięci (w celu zapamiętania współczynników a_i), a dla funkcji wymiernej – $n + m + 1$ (można licznik i mianownik podzielić przez $a_i \neq 0$ i współczynnik przy najwyższej potędze x w liczniku będzie równy 1). Postać naturalna wielomianu $w(x)$ jest zarazem jego rozwinięciem Maclaurina, w którym

$$a_i = \frac{w^{(i)}(0)}{i!}.$$

Postać ta jest wygodna przy obliczaniu pochodnych i całek wielomianów. Na przykład, w celu obliczenia pochodnej rzędu k mamy

$$w^{(k)}(x) = \sum_{i=0}^{n-k} b_i x^i, \quad \text{gdzie } b_i = \frac{(i+k)!}{i!} a_i.$$

Postać Newtona wielomianu jest następująca:

$$w(x) = \sum_{i=0}^{n} b_i p_i(x),$$

gdzie

$$p_0(x) = 1, \quad p_i(x) = (x - x_0)(x - x_1)\ldots(x - x_{i-1}) = \prod_{k=0}^{i-1}(x - x_k), \; i = 1, 2, \ldots, n,$$

przy czym $x_0, x_1, \ldots, x_n$ oznaczają dane liczby. Wielomiany p_i $(i = 0, 1, \ldots, n)$ tworzą bazę przestrzeni wielomianów. Do reprezentacji numerycznej wielomianu w postaci Newtoa potrzeba $2n + 2$ miejsc pamięci.

Wielomiany mogą być także przestawiane w postaci kombinacji liniowej wielomianów ortogonalnych, które także tworzą bazę przestrzeni wielomianów. Wielomianami takimi są, na przykład, wielomiany Czebyszewa T_i pierwszego rodzaju zdefiniowane zależnościami

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_i(x) = 2xT_{i-1}(x) - T_{i-2}, \; i = 2, 3, \ldots .$$

Dowolny wielomian stopnia n można wówczas przedstawić w postaci

$$w(x) = \sum_{i=0}^{n} c_i T_i(x).$$

Do reprezentacji numerycznej wielomianu w postaci kombinacji liniowej wielomianów ortogonalnych potrzeba $n + 1$ miejsc pamięci (w celu zapamiętania współczynników c_i).

2.2. Algorytm Hornera

Wielomian

$$w(x) = \sum_{i=0}^{n} a_i x^i$$

można przedstawić następująco:

$$w(x) = (\ldots(a_n x + a_{n-1})x + \ldots + a_1)x + a_0. \tag{2.1}$$

Z zapisu (2.1) wynika sposób obliczania wartości wielomianu $w(x)$, zwany *algorytmem Hornera*:

$$\begin{aligned} &w_n := a_n, \\ &w_k := w_{k+1}x + a_k \quad dla \quad k = n-1, n-2, \ldots, 0. \end{aligned} \tag{2.2}$$

Mamy przy tym $w(x) \equiv w_0$.

Dla algorytmu (2.2) udowodniono następujące

Twierdzenie 2.1. *Algorytm Hornera jest jedynym algorytmem, który minimalizuje liczbę dodawań i mnożeń przy obliczaniu wartości wielomianu danego w postaci naturalnej.*

Przykład 2.1

Stosując algorytm Hornera znaleźć wartość wielomianu $w(x)$ dla $x = 2$, jeśli

$$w(x) = x^3 - 2x^2 - 5x + 5.$$

Zgodnie z algorytmem (2.2) wyznaczamy kolejno

$$\begin{aligned}
w_3 &= a_3 = 1,\\
w_2 &= w_3 x + a_2 = 1\cdot 2 + (-2) = 0,\\
w_1 &= w_2 x + a_1 = 0\cdot 2 + (-5) = -5,\\
w_0 &= w_1 x + a_0 = -5\cdot 2 + 5 = -5.
\end{aligned}$$

Zatem $w(2) = w_0 = -5$. ■

Algorytm Hornera ma także inną interpretację. Służy on bowiem do dzielenia wielomianu $w(y)$ przez dwumian $y - x$. Jeśli porównamy współczynniki przy tych samych potęgach zmiennej y w równości

$$\sum_{k=0}^{n} a_k y^k = \left(\sum_{k=0}^{n-1} b_{k+1} y^k \right)(y - x) + b_0,$$

to otrzymamy

$$a_k = b_k - b_{k+1} x,$$

czyli

$$b_k = b_{k+1} x + a_k .$$

Zatem z (2.2) wynika, że $b_i = w_i$. Algorytm Hornera wyznacza więc oprócz wartości $w(x) \equiv w_0$, także współczynniki $b_1, b_2, \ldots, b_n$ ilorazu z dzielenia $w(y)$ przez $y - x$.

Przykład 2.2

Jaki wielomian jest ilorazem z dzielenia wielomianu w z przykładu 2.1 przez dwumian $x - 2$? Aby odpowiedzieć na to pytanie, napiszmy wielomian ze wspomnianego przykładu jako wielomian zmiennej y:

$$w(y) = y^3 - 2y^2 - 5y + 5.$$

Szukamy ilorazu z dzielenia tego wielomianu przez $y - 2$, a więc w podanym wyżej algorytmie należy przyjąć $x = 2$. Ponieważ współczynniki b_i ilorazu są równe współczynnikom w_i ($i = 1$, 2, 3), a te ostatnie wyznaczyliśmy w przykładzie 2.1, więc iloraz ma postać

$$v(y) = y^2 - 5.$$

Współczynnik $b_0 = w_0 = -5$ jest resztą z tego dzielenia. ■

Algorytm Hornera można także stosować do obliczania znormalizowanych pochodnych $\frac{w^{(j)}(x)}{j!}$ wielomianu $w(x)$. Jeśli wielomian zapiszemy w postaci

$$w(y) = (y-x)^j v(y) + r(y), \tag{2.3}$$

gdzie $r(y)$ oznacza wielomian stopnia mniejszego od j, to po j-krotnym zróżniczkowaniu z zależności (2.3) dla $y = x$ mamy

$$w^{(j)}(x) = j!\,v(x).$$

Współczynniki wielomianu v i wartość $v(x)$ obliczamy dzieląc wielomian w i kolejno otrzymywane ilorazy $\frac{w(y)}{(y-x)^k}$ $(k = 1, 2, \ldots, j-1)$ przez $y-x$ algorytmem Hornera.

Poprzez indukcję matematyczną względem m można udowodnić

Twierdzenie 2.2. *Stosowanie algorytmu Hornera do obliczenia $m \le n$ początkowych znormalizowanych pochodnych $\frac{w^{(j)}(x)}{j!}$ wielomianu stopnia n wymaga*

$$(m+1)\left(n - \frac{m}{2}\right)$$

dodawań i mnożeń.

Przykład 2.3

Obliczmy wartości wszystkich znormalizowanych pochodnych w punkcie $x = 2$ wielomianu z przykładu 2.1. Wykonując obliczenia bez stosowania algorytmu Hornera mamy

$$w'(x) = 3x^2 - 4x - 5 \quad \Rightarrow \quad \frac{w'(2)}{1!} = -1,$$
$$w''(x) = 6x - 4 \quad \Rightarrow \quad \frac{w''(2)}{2!} = 4,$$
$$w'''(x) = 6 \quad \Rightarrow \quad \frac{w'''(2)}{3!} = 1.$$

Zastosujmy teraz algorytm Hornera. W celu wyznaczenia wartości znormalizowanej pochodnej rzędu pierwszego danego wielomianu, dzielimy wielomian $w(y)$ (dany wielomian rozpatrywany jako wielomian zmiennej y) przez $y - 2$. Mamy (zob. przykład 2.2)

$$v_1(y) = y^2 - 5.$$

Wartość $v_1(2)$ obliczamy algorytmem Hornera:

$$\begin{aligned} w_2 &= 1, \\ w_1 &= w_2 \cdot 2 + 0 = 2, \\ w_0 &= w_1 \cdot 2 + (-5) = -1 \end{aligned}$$

i widzimy, że $w_0 = \dfrac{w'(2)}{1!}$. Wielomian $v_1(y)$ jest ilorazem $\dfrac{w(y)}{y-2}$. Dzielimy teraz $v_1(y)$ znowu przez $y - 2$ algorytmem Hornera. Współczynniki nowego ilorazu są równe obliczonym już współczynnikom w_i $(i = 1, 2)$. Zatem

$$v_2(y) = y + 2.$$

Stosując ponownie algorytm Hornera do obliczenia $v_2(2)$ otrzymujemy

$$\begin{aligned} w_1 &= 1, \\ w_0 &= w_1 \cdot 2 + 2 = 4. \end{aligned}$$

Tym razem $w_0 = \dfrac{w''(2)}{2!}$, a wielomian $v_2(y)$ jest ilorazem $\dfrac{w(y)}{(y-2)^2}$. Jedyny współczynnik wielomianu $v_3(y)$ są równy w_1, czyli

$$v_3(y) = 1$$

i oczywiście $v_3(2) = \dfrac{w'''(2)}{3!}$. Obliczając wszystkie znormalizowane pochodne wielomianu $w(x)$ w punkcie $x = 2$ wykonaliśmy 6 dodawań i tyle samo mnożeń. ■

2.3. Algorytm Shaw-Trauba

Algorytm ten jest stosowany do obliczenia wszystkich lub wielu pochodnych wielomianu. Jest on wówczas istotnie tańszy od algorytmu Hornera.

Niech wielomian $w(x)$ stopnia n dany będzie w postaci naturalnej. Załóżmy, że $x \neq 0$. Niech $n + 1 = pq$, gdzie p i q oznaczają pewne liczby naturalne. Określmy funkcje pomocnicze:

$$s(j) = (n - j) \bmod q, \quad r(j) = \begin{cases} 0, & \text{dla } j \bmod q \neq 0, \\ q, & \text{dla } j \bmod q = 0, \end{cases} \quad j = 0, 1, \ldots, n,$$

gdzie $a \bmod b$ oznacza resztę z dzielenia a przez b.

Algorytm Shaw-Trauba polega na wyznaczeniu elementów tablicy T_i^j zgodnie z następującymi wzorami:

$$\begin{aligned} T_i^{-1} &= a_{n-i-1} x^{s(i+1)}, \quad i = 0, 1, \ldots, n-1, \\ T_j^j &= a_n x^{s(0)}, \quad j = 0, 1, \ldots, m, \\ T_i^j &= T_{i-1}^{j-1} + T_{i-1}^j x^{r(i-j)}, \quad j = 0, 1, \ldots, m, \quad i = j+1, j+2, \ldots, n. \end{aligned} \tag{2.4}$$

Okazuje się, że

$$T_n^j = \frac{w^{(j)}(x)}{j!} x^{j \bmod q}.$$

Liczba dodawań w algorytmie (2.4) wynosi $(m+1)\left(n-\frac{m}{2}\right)$. Po obliczeniu T_n^j w celu wyznaczenia $\frac{w^{(j)}(x)}{j!}$ należy wykonać $m-\left[\frac{m}{q}\right]$ dzieleń. Ogólna liczba mnożeń i dzieleń potrzebnych do obliczenia m początkowych znormalizowanych pochodnych wynosi

$$n-1+q+\frac{m(n+1)}{q}-(m+2)p+\frac{q}{2}(p+1)$$

(dowód indukcyjny względem m).

Dla $q = 1$ otrzymujemy algorytm Hornera. W przypadku, gdy $q = n + 1$ mamy algorytm, który wymaga $(m+1)\left(n-\frac{m}{2}\right)$ dodawań, $2n - 1$ mnożeń i m dzieleń. Mamy wówczas algorytm o najmniejszej znanej liczbie działań, który jest optymalny ze względu na liczbę mnożeń. Shaw i Traub udowodnili bowiem, że w celu obliczenia $\frac{w^{(j)}(x)}{j!}$ dla $j = 0, 1, \ldots, m$. Potrzeba co najmniej $2n - 1$ mnożeń. Algorytm Shaw-Trauba jest numerycznie poprawny.

Przykład 2.4

W przykładzie 2.3 wyznaczyliśmy algorytmem Hornera wartości wszystkich znormalizowanych pochodnych w punkcie $x = 2$ wielomianu z przykładu 2.1. Znajdźmy teraz te wartości algorytmem Shaw-Trauba.

Najpierw wybieramy liczby naturalne p i q tak, by $n + 1 = pq$. W naszym przypadku $n = 3$, czyli $pq = 4$. Możemy zatem za p i q przyjąć następujące wartości: $p = 4$ i $q = 1$ lub $p = q = 2$ lub $p = 1$ i $q = 4$. W przypadku, gdy $q = 4$ otrzymujemy algorytm o najmniejszej znanej liczbie mnożeń. Rozważmy zatem ten przypadek.

Określamy funkcje $s(j)$ i $r(j)$:

$$s(j) = (3-j) \bmod 4, \quad j = 0, 1, 2, 3,$$

czyli

$$s(j) = \begin{cases} 3, & \text{dla } j = 0, \\ 2, & \text{dla } j = 1, \\ 1, & \text{dla } j = 2, \\ 0, & \text{dla } j = 3, \end{cases}$$

$$r(j) = \begin{cases} 0, & \text{dla } j \bmod 4 \neq 0, \\ q, & \text{dla } j \bmod 4 = 0, \end{cases}$$

czyli

$$r(j) = \begin{cases} 0, & \text{dla } j = 1, 2, 3, \\ 4, & \text{dla } j = 0. \end{cases}$$

Mamy

$$T_i^{-1} = a_{2-i} x^{s(i+1)}, \quad i = 0, 1, 2,$$

czyli

$$T_0^{-1} = a_2 x^{s(1)} = -2 \cdot 2^2 = -2 \cdot 2 \cdot 2 = -8,$$
$$T_1^{-1} = a_1 x^{s(2)} = -5 \cdot 2^1 = -10,$$
$$T_2^{-1} = a_0 x^{s(3)} = 5$$

i do wyznaczenia T_0^{-1} potrzebne były 2 mnożenia, T_1^{-1} – 1 mnożenie, a przy wyznaczaniu T_2^{-1} nie było potrzebne żadne działanie. Dalej mamy

$$T_j^j = a_3 x^{s(0)} = 1 \cdot 2^3 = 1 \cdot 2 \cdot 2 \cdot 2 = 8, \quad j = 0, 1, 2, 3$$

i wydawać by się mogło, że do wyznaczenia tych wartości potrzebne jest wykonanie 3 mnożeń. Ponieważ jednak $x^{s(0)} = x^{s(1)} \cdot x$, a wartość $x^{s(1)}$ zostało już obliczona wcześniej, więc wystarczą tu 2 mnożenia. Następnie mamy

$$T_i^j = T_{i-1}^{j-1} + T_{i-1}^j 2^{r(i-j)}, \quad j = 0, 1, 2, 3, \quad i = j+1, \ldots, 3.$$

Dla każdej możliwej wartości i oraz j mamy $r(i-j) = 0$, a więc zawsze $2^{r(i-j)} = 1$. Otrzymujemy więc

$$T_1^0 = T_0^{-1} + T_0^0 = -8 + 8 = 0,$$
$$T_2^0 = T_1^{-1} + T_1^0 = -10 + 0 = -10,$$
$$T_3^0 = T_2^{-1} + T_2^0 = 5 - 10 = -5,$$
$$T_2^1 = T_1^0 + T_1^1 = 0 + 8 = 8,$$
$$T_3^1 = T_2^0 + T_2^1 = -10 + 8 = -2,$$
$$T_3^2 = T_2^1 + T_2^2 = 8 + 8 = 16.$$

Do obliczenia tych wartości wystarczyło wykonać 6 dodawań. Zatem do obliczenia

$$T_3^j = \frac{w^{(j)}(2)}{j!} x^{j \bmod 4}, \quad j = 1, 2, 3,$$

wykonaliśmy 5 mnożeń i 6 dodawań. Mamy

$$T_3^1 = \frac{w'(2)}{1!} 2^{1 \bmod 4} = \frac{w'(2)}{1!} \cdot 2$$

i aby obliczyć $\dfrac{w'(2)}{1!}$ trzeba jeszcze wykonać 1 dzielenie. Ponadto

$$T_3^2 = \frac{w''(2)}{2!} 2^{2 \bmod 4} = \frac{w''(2)}{2!} \cdot 2 \cdot 2,$$

a więc dodatkowo trzeba wykonać 2 dzielenia,

$$T_3^3 = \frac{w'''(2)}{3!} 2^{3 \bmod 4} = \frac{w'''(2)}{3!} \cdot 2 \cdot 2 \cdot 2$$

i potrzebne są jeszcze 3 dzielenia. Jednak zgodnie z podanym wzorem liczba dzieleń wynosi $m - \left\lfloor \frac{m}{q} \right\rfloor = 3 - \left\lfloor \frac{3}{4} \right\rfloor = 3$, a my otrzymaliśmy 6 dzieleń. Jeśli jednak uwzględnimy fakt, że

$2^{2 \bmod 4} = 2^{s(1)}$, a to już policzyliśmy oraz $2^{3 \bmod 4} = 2^{s(0)}$ i to też już policzyliśmy, to faktycznie okaże się, że wystarczy wykonać tylko 3 dzielenia.

Zadania

1. Stosując algorytm Hornera znaleźć $w(1)$ dla wielomianu

$$w(x) = x^4 + x^3 - 4x^2 - 3x + 3.$$

2. Jaki wielomian jest ilorazem z dzielenia wielomianu z zadania 1 przez dwumian $x - 1$?
3. Za pomocą algorytmu Hornera znaleźć wartości wszystkich znormalizowanych pochodnych w punkcie $x = 1$ wielomianu z zadania 1.
4. Stosując algorytm Hornera znaleźć wartości wszystkich znormalizowanych pochodnych wielomianu

$$w(x) = 4x^4 - 2x^2 + x - 3$$

w punkcie $x = -1$.

5. Za pomocą algorytmu Shaw-Trauba znaleźć wartości wszystkich znormalizowanych pochodnych wielomianu z zadania 1 w punkcie $x = 1$. Rozważyć różne możliwości doboru liczb naturalnych p i q.