

Zadanie 1

Damian Soliński

Piotr Kulas

Marek Skiba

15.10.2012

1 Treść zadania

Ustalić liczbę naturalną n_{max} . Wczytać $n \in \{0, 1, \dots, n_{max}\}$ oraz wartości A_0, A_1, \dots, A_n . Następnie, dopóki "użytkownik się nie znudzi", wczytywać $x_0 \in \mathbf{R}$ i wyznaczać w postaci ogólnej wielomian $W = W(x_0)$ stopnia co najwyżej n taki, że $W^{(i)}(x_0) = A_i$, $i=0, 1, \dots, n$.

2 Schemat rozwiązywania zadania

Ustalamy n_{max} . Wczytujemy n , a następnie wartości A_0, A_1, \dots, A_n , i na końcu x_0 , aż użytkownik się znudzi. Układ n równań liniowych o n niewiadomych ma postać :

$$W^{(0)}(x_0) = B_n x_0^n + B_{n-1} x_0^{n-1} + \dots + B_1 x_0^1 + B_0 x_0^0 = A_0$$

$$W^{(1)}(x_0) = (B_n x_0^n)^{(1)} + (B_{n-1} x_0^{n-1})^{(1)} + \dots + (B_1 x_0^1)^{(1)} = A_1$$

$$W^{(2)}(x_0) = (B_n x_0^n)^{(2)} + (B_{n-1} x_0^{n-1})^{(2)} + \dots + (B_2 x_0^2)^{(2)} = A_2$$

.

.

.

$$W^{(n-1)}(x_0) = (B_n x_0^n)^{(n-1)} + (B_{n-1} x_0^{n-1})^{(n-1)} = A_{n-1}$$

$$W^{(n)}(x_0) = (B_n x_0^n)^{(n)} = A_n$$

Dla każdego działania $(B_k x_0^k)^{(i)}$ gdzie $k < i$ wynik jest równy 0 i jest pominięte w układzie równań.

Liczac od ostatniego równania możemy łatwo uzyskać wyniki naszych niewiadomych A_0, A_1, \dots, A_n .

3 Przykładowe rozwiązanie

Niech $n_{max} = 100$. Wczytujemy następujące dane :

$$n=3, A_0=1, A_1=2, A_2=3, A_3=6, x_0=2$$

Powstaje układ równań :

$$B_3 x_0^3 + B_2 x_0^2 + B_1 x_0^1 + B_0 x_0^0 = 1$$

$$3B_3 x_0^2 + 2B_2 x_0^1 + B_1 x_0^0 + 0 = 2$$

$$6B_3 x_0^1 + 2B_2 x_0^0 + 0 + 0 = 4$$

$$6B_3 x_0^0 + 0 + 0 + 0 = 6$$

Wstawiamy wartość w miejsce x_0 :

$$B_3 \cdot 8 + B_2 \cdot 4 + B_1 \cdot 2 + B_0 = 1$$

$$3B_3 \cdot 4 + 2B_2 \cdot 2 + B_1 = 2$$

$$6B_3 \cdot 2 + 2B_2 = 4$$

$$6B_3 = 6$$

Liczymy od końca :

$$B_3 = \frac{6}{6} = 1$$

$$12 \cdot 1 + 2B_2 = 4$$

$$2B_2 = -8$$

$$B_2 = -4$$

$$3 \cdot 1 \cdot 4 + 2 \cdot (-4) \cdot 2 + B_1 = 2$$

$$12 - 16 + B_1 = 2$$

$$B_1 = 6$$

$$1 \cdot 8 + (-4) \cdot 4 + 6 \cdot 2 + B_0 = 1$$

$$8 - 16 + 12 + B_0 = 1$$

$$4 + B_0 = 1$$

$$B_0 = -3$$

Rozwiązaniem zadania jest :

$$W(x_0) = x_0^3 - 4x_0^2 + 6x_0 - 3$$

4 Źródło programu:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NMAX 100

double MyPow(double x0, int exp) {
    return exp > 0 ? MyPow(x0, exp - 1) * x0 : 1;
}

void fillMatrix(int n, double **t, double x0) {
    int i, j;
    for (j = 0; j < n; j++) {
        t[0][j] = 1;
        for (i = 1; i < n; i++) {
            if (i + j < n) t[i][j] = t[i - 1][j] * (n - i - j);
        }
    }
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (i + j < n) t[i][j] *= MyPow(x0, n - j - i - 1);
        }
    }
}

void resolveMatrix(int n, double **t) {
    int i, j;
    for (i = n - 1; i >= 0; i--) {
        t[i][n] /= t[i][n - i - 1];
        for (j = 0; j < i; j++) {
            t[j][n] -= t[j][n - i - 1] * t[i][n];
        }
    }
}

void printMatrix(int n, double **t) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j <= n; j++) {
            printf("%lf ", t[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
```

```

void clearMatrix(int n, double **t, double *A) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            t[i][j] = 0;
        }
        t[i][n] = A[i];
    }
}

void printPolynomial(int n, double **t) {
    int i = n;
    printf("Wynik:  $W(x)$  = ");
    while (--i) {
        printf("%lf", t[i][n]);
        printf("x^%d + ", i);
    }
    printf("%lf x^0\n", t[0][n]);
}

int main() {
    double *A, **t, x0;
    int i, n;
    char chce_podac_x[2];

    do {
        printf("Podaj liczbę naturalną z przedziału od 0 do %d: ", NMAX);
        scanf("%d", &n);
    } while (n < 0 || n > NMAX);

    n += 1;
    A = malloc(n * sizeof (double));
    for (i = 0; i < n; i++) {
        printf("Podaj A[%d]: ", i);
        scanf("%lf", &A[i]);
    }
    t = malloc(n * sizeof (double*));
    for (i = 0; i < n; i++) {
        t[i] = malloc((n + 1) * sizeof (double));
    }

    do {
        printf("Podaj X0: ");
        scanf("%lf", &x0);
        clearMatrix(n, t, A);
        fillMatrix(n, t, x0);
        resolveMatrix(n, t);
        printPolynomial(n, t);
    }
}

```

```

    do {
        printf("Chcesz_podać_kolejne_X0?(tak/nie):_");
        scanf("%s", chce_podac_x);
        if(strcmp(chce_podac_x, "tak") == 0 || strcmp(chce_podac_x, "nie") == 0)
    } while(1);

} while (strcmp(chce_podac_x, "tak") == 0);

return 0;
}

```

5 Wynik programu:

```

[mskiba@sigma] ~/MetodyObliczeniowe $ gcc zad1.c -o zad1
[mskiba@sigma] ~/MetodyObliczeniowe $ ./zad1
Podaj liczbe naturalna z przedziału od 0 do 100: 101
Podaj liczbe naturalna z przedziału od 0 do 100: -1
Podaj liczbe naturalna z przedziału od 0 do 100: 3
Podaj A[0]: 2
Podaj A[1]: 3
Podaj A[2]: 4
Podaj A[3]: 6
Podaj X0: 2
Wynik:  $W(x) = 1.000000x^3 + -4.000000x^2 + 7.000000x^1 + -4.000000x^0$ 
Chcesz podać kolejne X0? (tak/nie): s
Chcesz podać kolejne X0? (tak/nie): tak
Podaj X0: 4
Wynik:  $W(x) = 1.000000x^3 + -10.000000x^2 + 35.000000x^1 + -42.000000x^0$ 
Chcesz podać kolejne X0? (tak/nie): nie
[mskiba@sigma] ~/MetodyObliczeniowe $

```