| Grupa lab.3 | Data wykonania<br>28.06.2024 | Inżynieria Obliczeniowa<br>2023/2024 |
|---|---|---|
| Temat ćwiczenia<br>**Aproksymacja wielomianami Grama dla węzłów równoodległych** | | |
| Imię i nazwisko<br>Karolina Kurowska | | Ocena i uwagi |

## Cel ćwiczenia:

Celem ćwiczenia było przeprowadzenie aproksymacji danych za pomocą wielomianów ortogonalnych Grama dla równoodległych węzłów aproksymacyjnych. Implementacja miała na celu obliczenie współczynników $c_k$ i $s_k$ dla wielomianu aproksymacyjnego dowolnego stopnia oraz wyznaczenie wartości funkcji aproksymującej $y_m(x)$ w podanych punktach.

## Wstęp:

Aproksymacja za pomocą wielomianów Grama polega na wykorzystaniu specjalnej grupy wielomianów ortogonalnych, które umożliwiają stworzenie macierzy diagonalnej, co upraszcza obliczenia i eliminuje problem złego uwarunkowania macierzy układu normalnego. Wykorzystując wzory na dwumian Newtona oraz wielomiany czynnikowe, obliczamy wartości wielomianu aproksymacyjnego dla podanych danych wejściowych.

## Implementacja ćwiczenia:

Program został napisany w języku C++ i miał za zadanie:

- Wczytanie danych aproksymacji (węzłów i wartości w węzłach) z pliku tekstowego.
- Obliczenie współczynników $c_k$ i $s_k$
- Wyświetlenie liczby węzłów, współczynników $c_k$ i $s_k$ ,węzłów aproksymacji oraz wartości funkcji aproksymującej.
- Obliczenie wartości funkcji aproksymującej w podanym z klawiatury węźle aproksymacji.

Kod: ⬇

```cpp
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <cmath>
using namespace std;

int silnia(int n) {
    if (n == 0) {
        return 1;
    } else {
        int wynik = 1;
        for (int i = n; i > 1; i--) {
            wynik *= i;
        }
        return wynik;
    }
}

double dwumian_newtona(double n, double k) {
    if (k > n) return 0;
    double wynik = silnia(static_cast<int>(n)) / (silnia(static_cast<int>(k)) * silnia(static_cast<int>(n - k)));
    return wynik;
}

double wiel_czyn(double x, double n) {
    double wynik = 1;
    for (int j = 0; j < n; j++) {
        wynik *= (x - j);
    }
    return wynik;
}

double wiel_grama(double n, double q, double k) {
    double suma = 0;
    for (int s = 0; s <= k; s++) {
        suma += (pow(-1, s) * dwumian_newtona(k, s) * dwumian_newtona(k + s, s) * (wiel_czyn(q, s) / wiel_czyn(n, s)));
    }
    return suma;
}

void wyswietl_tablice(double* tab, int n) {
    for (int i = 0; i < n; i++) {
        cout << tab[i] << " ";
    }
    cout << endl;
}
```

```cpp
void oblicz_y_m (int n, double* tab_x, double* y_m, double h, double* c_k, double* s_k, int stop_wielo) {
    for (int m = 0; m < n; m++) {
        double q = ((tab_x[m] - tab_x[0]) / h);

        for (int k = 0; k <= stop_wielo; k++) {
            y_m[m] += ((c_k[k] / s_k[k]) * wiel_grama (n - 1, q, k));
        }
        cout << "y_m[" << m << "] = " << y_m[m] << endl;
    }
}

void oblicz_y_dla_x (double x, int n, double* tab_x, double h, double* c_k, double* s_k, int stop_wielo) {
    double q = ((x - tab_x[0]) / h);
    double moj_y = 0;
    for (int k = 0; k <= stop_wielo; k++) {
        moj_y += ((c_k[k] / s_k[k]) * wiel_grama (n - 1, q, k));
    }

    cout << "\ny_m[" << x << "] = " << moj_y << endl;
}

int main() {
    int n;
    ifstream plik;
    plik.open("dane_gram.txt", ios::in);

    if (!plik.good()) {
        cout << "Plik nie istnieje\n";
        exit(0);
    }

    plik >> n;
    cout << "Ilosc wezlow: " << n << endl;

    // Stworzenie macierzy
    double* tab_x = new double[n];
    double* tab_y = new double[n];
    double* y_m = new double[n];
```

```
87      for (int i = 0; i < n; i++) {
88          plik >> t
89          y_m[i] =
90      }
91      for (int i = 0; i < n; i++) {
92          plik >> tab_y[i];
93      }
94      plik.close();
95
96      cout << "Punkty x: ";
97      wyswietl_tablice(tab_x, n);
98      cout << "Punkty y: ";
99      wyswietl_tablice(tab_y, n);
100
101     double h = tab_x[1] - tab_x[0];
102     const int stop_wielo = 1;
103     cout << "Stopien wielomianu: " << stop_wielo << endl;
104
105     double x = 0;
106     cout << "\nwprowadz x = ";
107     cin >> x;
108
109     double c_k[stop_wielo + 1];
110     double s_k[stop_wielo + 1];
111
112     for (int k = 0; k <= stop_wielo; k++) {
113         c_k[k] = 0;
114         s_k[k] = 0;
115         double q = 0;
116         for (int i = 0; i < n; ++i) {
117             q = (tab_x[i] - tab_x[0]) / h;
118             double F_k = wiel_grama (n - 1, q , k);
119             s_k[k] += F_k * F_k;
120             c_k[k] += tab_y[i] * F_k;
121         }
122     }
123
124     for (int i = 0; i <= stop_wielo; i++) {
125         cout << "c_k = " << c_k[i] << "   s_k = " << s_k[i] << endl;
126     }
127     cout << endl;
```

(int)0

Wyszukaj w trybie online

```
128
129     oblicz_y_m (n, tab_x, y_m, h, c_k, s_k, stop_wielo);
130     oblicz_y_dla_x (x, n, tab_x, h, c_k, s_k, stop_wielo);
131
132     delete[] tab_x;
133     delete[] tab_y;
134     delete[] y_m;
135
136     return 0;
137 }
138
```

Wyniki:

| y = 2.5 | y = 6.5 |
| --- | --- |
| Ilosc wezlow: 8<br>Punkty x: 1 2 3 4 5 6 7 8<br>Punkty y: 2 4 3 5 6 9 11 11<br>Stopien wielomianu: 1<br><br>wprowadz x = 2.5<br>c_k = 51  s_k = 8<br>c_k = -16.7143  s_k = 3.42857<br><br>y_m[0] = 1.5<br>y_m[1] = 2.89286<br>y_m[2] = 4.28571<br>y_m[3] = 5.67857<br>y_m[4] = 7.07143<br>y_m[5] = 8.46429<br>y_m[6] = 9.85714<br>y_m[7] = 11.25<br><br>y_m[2.5] = 3.58929 | Ilosc wezlow: 8<br>Punkty x: 1 2 3 4 5 6 7 8<br>Punkty y: 2 4 3 5 6 9 11 11<br>Stopien wielomianu: 1<br><br>wprowadz x = 6.5<br>c_k = 51  s_k = 8<br>c_k = -16.7143  s_k = 3.42857<br><br>y_m[0] = 1.5<br>y_m[1] = 2.89286<br>y_m[2] = 4.28571<br>y_m[3] = 5.67857<br>y_m[4] = 7.07143<br>y_m[5] = 8.46429<br>y_m[6] = 9.85714<br>y_m[7] = 11.25<br><br>y_m[6.5] = 9.16071 |

Wnioski:

Przeprowadzone ćwiczenie umożliwiło dokładne zrozumienie procesu aproksymacji za pomocą wielomianów ortogonalnych Grama. Użycie tych wielomianów pozwoliło na uproszczenie obliczeń poprzez uzyskanie macierzy diagonalnej, co znacząco ułatwiło obliczenia. Program poprawnie wczytał dane z pliku, obliczył współczynniki $c_k$ i $s_k$, oraz wyznaczył wartości funkcji aproksymującej $y_m(x)$ w zadanych punktach.

Zaobserwowano, że wartości funkcji aproksymującej w punktach x = 2.5 i x=6.5 były zgodne z przewidywaniami. Wyniki te potwierdzają skuteczność metody aproksymacji wielomianami Grama dla danych równoodległych.
Ostatecznie, ćwiczenie było wartościowe zarówno pod względem teoretycznym, jak i praktycznym, umożliwiając zrozumienie i implementację zaawansowanych technik aproksymacyjnych w praktyce inżynierskiej.