Grupa lab.3	Data wykonania 11.05.2024	Inżynieria Obliczeniowa 2023/2024
Temat ćwiczenia **Całkowanie Numeryczne - Metoda Gaussa-Legendre'a**		
Imię i nazwisko Karolina Kurowska		Ocena i uwagi

Cel ćwiczenia:

Celem ćwiczenia było napisanie funkcji obliczającej wartość całki oznaczonej zadaną metodą Gaussa-Legendre'a z wykorzystaniem kwadratury dwu-, trzy- i cztero-węzłowej oraz porównanie wyników z metodami prostokątów, trapezów i parabol. Wyniki miały zostać zweryfikowane poprzez porównanie ich z wartościami dokładnymi.

Wstęp

Kwadratury Gaussa-Legendre'a to metoda numerycznego całkowania, która polega na aproksymacji funkcji podcałkowej wielomianem interpolacyjnym. Ogólny wzór na obliczenie wartości całki przy użyciu kwadratury Gaussa-Legendre'a jest następujący:

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\sum_{i=1}^{n} A_i f(t_i),$$

gdzie $t_i = \frac{a+b}{2} + \frac{b-a}{2} * x_i$ są przekształconymi węzłami, a A_i to odpowiednie wagi.

Implementacja ćwiczenia

Poniżej zamieszczono główną część kodu, który oblicza wartości całek dla zdefiniowanych funkcji przy użyciu kwadratury Gaussa-Legendre'a oraz metod prostokątów, trapezów i parabol.

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

double rectagle_integral (double a, double b, int n, double (*f)(double)) {
    double s = (b - a) / n;
    //cout << s;
    double sum = 0;
    for (int i = 0; i < n; i++) {
        sum += f ((a + (i * s)) + (0.5 * s));
    }
    sum *= s;
    return sum;
}
double trapezium_integral (double a, double b, double n, double (*f)(double)) {
    double s = (b - a) / n;
    double sum = 0;
    for (int i = 0; i < n; i++) {
        double x_i = a + (i * s);
        double x_i1 = a + ((i + 1) * s);
        sum += ((x_i1 - x_i) / 2) * (f (x_i) + f (x_i1));
    }
    return sum;
}
double simpson_integral (double a, double b, double n, double (*f)(double)) {
    double s = (b - a) / n;
    double sum = 0;
    for (int i = 0; i < n; i++) {
        double x_i = a + (i * s);
        double x_i1 = a + ((i + 1) * s);
        sum += ((x_i1 - x_i) / 6) * (f (x_i) + 4 * f ((x_i + x_i1) / 2) + f (x_i1));
    }
    return sum;
}
```

```cpp
double lagrange_integral (double a, double b, double n, double (*f)(double), double* waga, double* wezel) {
    double sum = 0;
    double iloraz = ((b - a) / 2);
    for (int i = 0; i < n; i++) {
        double t_i;
        t_i = ((a + b) / 2) + iloraz * wezel[i];
        sum += (waga[i] * f (t_i));
    }
    return iloraz * sum;
}

double function (double x) {
    double sum = (x * x) + (2 * x) + 5;
    return sum;
}
int main ()
{
    cout << "Wynik calki Lagrange'a: " << endl;
    double a, b;
    a = a = 0.5;
    b = 2.5;

    //Dla dwóch węzłów
    double* waga2 = new double[2];
    double* wezel2 = new double[2];
    double n = 2;
    wezel2[0] = -1.0 * (sqrt (3) / 3);
    wezel2[1] = sqrt (3) / 3;
    waga2[0] = 1;
    waga2[1] = 1;
    cout << "\nsinus" << endl;
    double integ2 = lagrange_integral (a, b, n, sin, waga2, wezel2);
    cout << "Wartosc dla " << n << " wezlow: " << integ2 << endl;
```

```cpp
    //Dla trzech węzłów
    double* waga3 = new double[3];
    double* wezel3 = new double[3];
    n = 3;
    wezel3[0] = -1 * sqrt (3.0 / 5.0);
    wezel3[1] = 0;
    wezel3[2] = sqrt (3.0 / 5.0);

    waga3[0] = 5.0 / 9.0;
    waga3[1] = 8.0 / 9.0;
    waga3[2] = 5.0 / 9.0;
    double integ3 = lagrange_integral (a, b, n, sin, waga3, wezel3);
    cout << "Wartosc dla " << n << " wezlow: " << integ3 << endl;

    //Dla czterech węzłów
    n = 4;
    double* waga4 = new double[4];
    double* wezel4 = new double[4];
    wezel4[0] = -1.0 * (1.0 / 35.0) * sqrt (525.0 + 70.0 * sqrt (30.0));
    wezel4[1] = -1.0 * (1.0 / 35.0) * sqrt (525.0 - 70 * sqrt (30.0));
    wezel4[2] = (1.0 / 35.0) * sqrt (525.0 - 70.0 * sqrt (30.0));
    wezel4[3] = (1.0 / 35.0) * sqrt (525.0 + 70.0 * sqrt (30.0));

    waga4[0] = (1.0 / 36.0) * (18.0 - sqrt (30.0));
    waga4[1] = (1.0 / 36.0) * (18.0 + sqrt (30.0));
    waga4[2] = (1.0 / 36.0) * (18.0 + sqrt (30.0));
    waga4[3] = (1.0 / 36.0) * (18.0 - sqrt (30.0));

    double integ4 = lagrange_integral (a, b, n, sin, waga4, wezel4);
    cout << "Wartosc dla " << n << " wezlow: " << integ4 << endl;
```

```cpp
cout << "\nkwadratowa" << endl;;
n = 2;
b = 5;
integ2 = lagrange_integral (a, b, n, function, waga2, wezel2);
cout << "Wartosc dla " << n << " wezlow: " << integ2 << endl;
n = 3;
integ3 = lagrange_integral (a, b, n, function, waga3, wezel3);
cout << "Wartosc dla " << n << " wezlow: " << integ3 << endl;
n = 4;
integ4 = lagrange_integral (a, b, n, function, waga4, wezel4);
cout << "Wartosc dla " << n << " wezlow: " << integ4 << endl;


cout << "\nexp" << endl;
n = 2;
integ2 = lagrange_integral (a, b, n, exp, waga2, wezel2);
cout << "Wartosc dla " << n << " wezlow: " << integ2 << endl;
n = 3;
integ3 = lagrange_integral (a, b, n, exp, waga3, wezel3);
cout << "Wartosc dla " << n << " wezlow: " << integ3 << endl;
n = 4;
integ4 = lagrange_integral (a, b, n, exp, waga4, wezel4);
cout << "Wartosc dla " << n << " wezlow: " << integ4 << endl;


double integral;
a = 0.5;
b = 2.5;
n = 4;
cout << "\n\nZadanie zajecia lab07" << endl;
```

Wyniki:

- Dla funkcji $sin(x)$ w przedziale [0.5, 2.5]:
  - Metoda Gaussa-Legendre'a

```
sinus
Wartosc dla 2 wezlow: 1.67163
Wartosc dla 3 wezlow: 1.67879
Wartosc dla 4 wezlow: 1.67873
```

  - Metody Prostokątów, Trapezów i Simpsona

```
Przedzial a = 0.5, b = 2.5, n = 20
Calkowanie funckji sinus
Wynik calki prostokaty: 1.67943
Wynik calki trapezy: 1.67733
Wynik calki simpson: 1.67873
```

  - **Dokładny wynik = 1.67877**
- Dla funkcji $x^2 + 2x + 5$ w przedziale [0.5, 5]:
  - Metoda Gaussa-Legendre'a

```
kwadratowa
Wartosc dla 2 wezlow: 88.875
Wartosc dla 3 wezlow: 88.875
Wartosc dla 4 wezlow: 88.875
```

  - Metody Prostokątów, Trapezów i Simpsona

```
Przedzial a = 0.5, b = 5, n = 20

Calkowanie funckji kwadratowej
Wynik calki prostokaty: 88.856
Wynik calki trapezy: 88.913
Wynik calki simpson: 88.875
```

- **Dokładny wynik = 88.875**

- Dla funkcji $exp(x)$ w przedziale [0.5, 5]:
    - Metoda Gaussa-Legendre'a

```
exp
Wartosc dla 2 wezlow: 138.621
Wartosc dla 3 wezlow: 146.426
Wartosc dla 4 wezlow: 146.757
```

    - Metody Prostokątów, Trapezów i Simpsona

```
Calkowanie funckji exp
Wynik calki prostokaty: 146.455
Wynik calki trapezy: 147.383
Wynik calki simpson: 146.765
```

    - **Dokładny wynik = 146.765**

Wnioski:

Wyniki obliczeń za pomocą metody Gaussa-Legendre'a z kwadraturą dwu-, trzy- i cztero-węzłową są bardzo zbliżone do siebie oraz do wyników uzyskanych metodą parabol. Metoda prostokątów daje nieco mniej dokładne wyniki, co jest szczególnie widoczne dla funkcji wykładniczej. Metoda trapezów daje wyniki zbliżone do metody Gaussa-Legendre'a, jednakże przy większej liczbie węzłów metoda Gaussa-Legendre'a może dawać bardziej dokładne rezultaty.

Podsumowując, metoda Gaussa-Legendre'a jest bardzo efektywną metodą numerycznego całkowania, szczególnie gdy zależy nam na dużej dokładności wyniku przy relatywnie małej liczbie węzłów.