Wydział Inżynierii Metali i Informatyki Przemysłowej

Imię i nazwisko Meg Paskowski	Temat laboratorium Całkowanie numeryczne – metoda prostokątów, trapezów, Simpsona i MC	Data oddania 11.04.2024r.	Data ćwiczeń 05.04.2024r.
Prowadzący dr hab. inż. Marcin Hojny		Grupa laboratoryjna 5	

1. Cel ćwiczenia

Celem laboratorium nr. 5 było zapoznanie się z pojęciami całkowania numerycznego, a dokładniej metodami prostokątów, trapezów, Simpsna i MC oraz implementacja tych zagadnień metodą w wybranym przez siebie języku programowania.

2. Wstęp teoretyczny

Całka oznaczona to wartość, która reprezentuje wartość pola powierzchni ograniczonej przez krzywą funkcji oraz osie układu współrzędnych w określonym przedziale całkowania.

Metoda prostokątów → jest to sposób polegający na przybliżeniu całki przez podział obszaru pod krzywą na równoległe *n* prostokąty o równych szerokościach, gdzie wysokością każdego prostokąta jest wartość funkcji w punkcie środkowym danego przedziału.

Do obliczenia pól prostokąta stosujemy wzór.

$$P = dx \cdot y \tag{1}$$

Podane *dx* to szerokość jednego z boków wyznaczany ze wzoru

$$dx = \frac{x_k - x_p}{n} \tag{2}$$

Gdzie:

- n – liczba prostokątów na jakie dzielimy szukane pole
- x_p, x_k – proste ograniczające zadany obszar przy założeniu, że $x_p < x_k$

Natomiast wartość y obliczona jest poprzez wyliczenie sumy wartości funkcji w punktach końcowych każdego z tych prostokątów.

$$y = |f(x_0) + f(x_1) + \ldots + f(x_i)| \tag{3}$$

Każdy następny będziemy wyznaczać zwiększając o szerokość dx.

Metoda trapezów → technika przybliżania całki przez podział obszaru pod krzywą na trapezy, których podstawy leżą na końcach przedziałów całkowania metodzie trapezów wysokością każdego trapezu jest średnia wartość funkcji na końcach przedziału.

Do obliczenia całki metodą trapezów wykorzystuje się wzór:

$$P = \frac{dx}{2} \cdot \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) \tag{4}$$

Gdzie:

- dx to szerokość jednego z boków trapezu obliczana ze wzoru (2)
- $f(x_i), f(x_{i+1})$ oznaczają wartości funkcji na końcach kolejnych trapezów.

Poprzez zastosowanie tego wzoru, otrzymujemy przybliżoną wartość całki, która jest sumą pól trapezów, a każdy z tych trapezów jest przybliżeniem obszaru pod krzywą na danym przedziale.

Metoda Simpsona → jest to najdokładniejsza metoda przybliżania całek w której funkcja podcałkowa jest przybliżana parabolą rozpiętą na dwóch krańcach przedziału całkowania oraz jego środku.

Do obliczenia wykorzystywane są obszary I obliczane za pomocą

$$I = \frac{dx}{3}\sum_{i=0}^{n-1}\left(f(x_i) + 4f\left(\frac{x_i+x_{i+1}}{2}\right) + f(x_{i+1})\right) \tag{5}$$

Gdzie:

- dx jest szerokością przedziału,
- n to liczba podprzedziałów,
- x_i i x_{i+1} to odpowiednio początek i koniec i-tego podprzedziału.

Każdy podprzedział jest podzielony na dwa równoliczne przedziały, a dla każdego takiego podprzedziału obliczamy wartość funkcji w jego punktach końcowych oraz środkowym. Następnie stosujemy odpowiednią wagę i sumujemy wartości, aby uzyskać przybliżoną wartość całki na danym przedziale.

Metoda Monte Carlo → polega na wylosowaniu n punktów z przedziału całkowania, a następnie obliczenie średniej wartości funkcji w tym przedziale.

Średnia wartość funkcji na danym przedziale jest obliczana przy pomocy

$$f_{\text{średnie}} = \frac{f(x_1)}{n} + \frac{f(x_2)}{n} + \ldots + \frac{f(x_n)}{n} \tag{7}$$

Gdzie:

- n to liczba punktur jakie będą losowane
- $f(x_n)$ to wartość funkcji w punkcie x_n

Dzięki średniej wartości funkcji można obliczyć całkę ze wzoru

$$\int_a^b f(x)dx = f_{\text{średnie}} \cdot dx \tag{8}$$

Gdzie:

- $f_{\text{średnie}}$
- dx to odległość przedziału (koniec – początek)

3. Implementacja

W ramach ćwiczeń zaimplementowano metody obliczania całek oznaczonych w języku C++. Do przyjęcia konkretnej funkcji dla której będzie liczona całka na danym przedziale utworzona została specjalna metoda „*function()*" przyjmująca wartość x dla której będzie zwracać obliczoną wartość y tej funkcji. Do ustalenia funkcji wykorzystana została w biblioteka „*cmath"* dzięki, której możemy np. podnosić dane do danej potęgi czy określać wartości bezwzględne.

```
//Funkcja dla której będzie liczona całka
double function(double x) {
	return pow(x, 3) + 2;
}
```

Następnie utworzone funkcje odpowiednio dla każdej metody liczenia całki oznaczonej.

1. <u>Metoda prostokątów</u> → przyjmuje ona informacje o starcie „*x_s*", końcu „*x_e*" przedziału oraz numerze prostokątów na jakie zostanie podzielony obszar całkowania „*numberOfRectangles*". Następnie deklarowane są wartości szerokości „*dx"* (2), sumy „*sum*" i pomocniczej „*x_i*" przechowującej start naszego obszaru całkowania. W kolejnym etapie tej metody za pomocą pętli *for* obliczana zostaje wartość „*y*" określonej we wzorze (3). Na sam koniec zwracany jest obliczony wynik P określony (1).

```
double RectangularlMethod(double x_s, double x_e, double numberOfRectangles) {
    double dx = (x_e - x_s) / numberOfRectangles;
    double sum = 0., x_i = x_s;

    for (int i = 0; i < numberOfRectangles; i++) {
```

```
        x_i += dx;
        sum += function(x_i);
    }

    return dx * sum;
}
```

2. Metoda trapezów → do metody przekazywane są informacje o starcie „*x_s*", końcu „*x_e*" przedziału oraz numerze trapezów na jakie zostanie podzielony obszar całkowania „*numberOfTrapezoids*". Kolejno określane są wartości szerokość „*dx*" (2), suma „*sum*" przechowująca wartości funkcji na końcach przedziału (4).. Za pomocą pętli for liczone i zwrócone jest pole.

```
double TrapezoidalMethod(double x_s, double x_e, double numberOfTrapezoids) {
    double dx = (x_e - x_s) / numberOfTrapezoids;
    double sum = (function(x_s) + function(x_e)) / 2; //wartości funkcji na końcach
przedziału
    double x_i = x_s + dx; //Rozpoczęcie od pierwszego trapezu

    for (int i = 1; i < numberOfTrapezoids; i++) {
        sum += function(x_i);
        x_i += dx;
    }

    return dx * sum;
}
```

3. Metoda Simpsona → funkcja przyjmuje informacje o starcie „*x_s*", końcu „*x_e*" przedziału oraz szerokości przedziału pomocny do obliczenia „*dx*" (2). Następnie deklarowane są zmienne „*I*" oznaczający całkowitą sumę przedziałów, „a", „b" odpowiednio początek i koniec danych przedziałów i ich środek „*midpoint*". Oraz korzystając ze wzoru (5) obliczamy poszczególne przedziały.

```
double SimpsonMethod(double x_s, double x_e, double number) {
    double dx = (x_e - x_s) / number; // szerokość przedziału
    double I = 0.0, a, b, midpoint;

    for (int i = 0; i < number; i++) {
        a = x_s + i * dx; // początek podprzedziału
        b = x_s + (i + 1) * dx; // koniec podprzedziału
        midpoint = (a + b) / 2.; // środek podprzedziału

        // Obliczenie wartości funkcji w punktach a, b i środku podprzedziału
        double fa = function(a);
        double fb = function(b);
        double fmid = function(midpoint);

        // Obliczenie sumy dla tego podprzedziału i dodanie do całkowitej sumy
        I += dx*(fa + 4 * fmid + fb)/6.;      //6 poniewaz dx to 2*szerokość h
    }

    return I;
}
```

4. Metoda Monte Carlo → w ostatniej metodzie również przekazujemy początek „*x_s*" i koniec „*x_e*" przedziału całowania oraz numer punktur losowych „numer". Deklarowana jest suma „sum" oraz szerokość przedziału „*dx*". Przy pomocy pętli for obliczamy wartości losowe x i odpowiednio dla nich zgodnie ze wzorem (8) średnią. Zwracana jest tutaj zmienna sum przekształcona do wzoru (9).

```
double MCMethod(double x_s, double x_e, double number) {
    srand(time(nullptr));
    double sum = 0.0, dx = x_e - x_s;

    for (int i = 0; i < number; i++) {
        double u = (double)rand() / RAND_MAX; // Losowa liczba zmiennoprzecinkowa z
przedziału [0, 1)
        double x = x_s + u * dx; // Losowy punkt w przedziale [x_s, x_e]
```

```
            sum += function(x);
        }

        sum *= dx / number;

        return sum;
    }
```

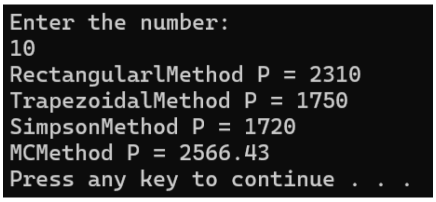
W programie głównym przyjmowana jest od użytkownika informacja o starcie i końcu przedziału całkowania i kolejno zapisywana do zmiennej „*x_s*” „*x_e*”. Następnie sprawdzany jest warunek czy określony przedział jest zgodny z założeniem, że koniec jest większy od początku. Jeżeli założenie zostało spełnione użytkownik przechodzi do podania wartości „*n*” odpowiedniej dla każdego z przypadków całkowania oraz kolejno wyświetlane zostają obliczone wartości dla konkretnych metod.

```
double x_s, x_e, number;

cout << "Give the interval of the integral.\nStart:" << endl;
cin >> x_s;
cout << "End:" << endl;
cin >> x_e;

if (x_s < x_e) {
        cout << "Enter the number: " << endl;
        cin >> number;

        cout << "RectangularlMethod P = " << RectangularlMethod(x_s, x_e, number) << endl;
        cout << "TrapezoidalMethod P = " << TrapezoidalMethod(x_s, x_e, number) << endl;
        cout << "SimpsonMethod P = " << SimpsonMethod(x_s, x_e, number) << endl;
        cout << "MCMethod P = " << MCMethod(x_s, x_e, number) << endl;
}
else {
        cout << "Invalid data :<" << endl;
}
```

4. Testy jednostkowe

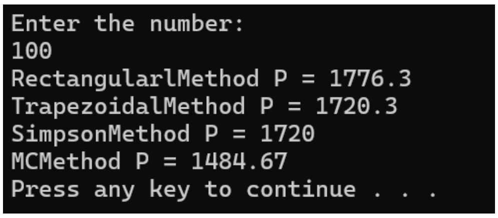
Testy zostały wykonane na przyjętej funkcji

- $f(x) = 3x^3+x-4$, na przedziale całkowania a=-3 b=7
  - Dla n=10:



Rysunek 1 Wyniki programu dla testu 1, n=10

  - Dla n = 100



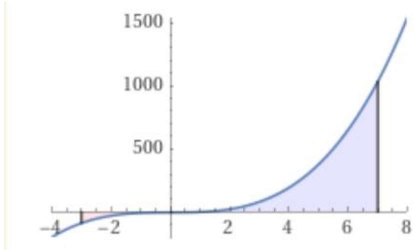
Rysunek 2 Wyniki programu dla testu 1, n=100

  - Dla n = 1000

```
Enter the number:
1000
RectangularlMethod P = 1725.6
TrapezoidalMethod P = 1720
SimpsonMethod P = 1720
MCMethod P = 1616.51
Press any key to continue . . .
```

Rysunek 3 Wyniki programu dla testu1, n=1000

Graficzna interpretacja całki pod funkcją Rysunek 4.



Rysunek 4 Graficzne przedstawienie całki dla testu 1

Wynik uzyskany na stronie „*WolframAlpha*”:

$$\int_{-3}^{7} f(x)\, dx = 1720$$

- f(x) = sinx , na przedziale całkowania a=0 b=4
  - Dla n = 10

```
Enter the number:
10
RectangularlMethod P = 1.48018
TrapezoidalMethod P = 1.63154
SimpsonMethod P = 1.65366
MCMethod P = 1.86013
Press any key to continue . . .
```

Rysunek 5 Wyniki programu dla testu2, n=10

  - Dla n = 100

```
Enter the number:
100
RectangularlMethod P = 1.63829
TrapezoidalMethod P = 1.65342
SimpsonMethod P = 1.65364
MCMethod P = 1.82552
Press any key to continue . . .
```

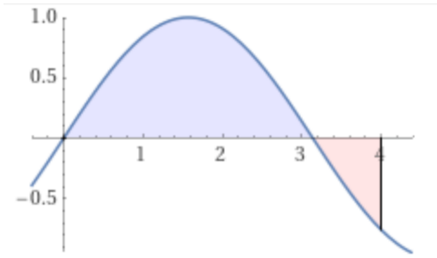
Rysunek 6 Wyniki programu dla testu2, n=100

  - Dla n = 1000

```
Enter the number:
1000
RectangularlMethod P = 1.65213
TrapezoidalMethod P = 1.65364
SimpsonMethod P = 1.65364
MCMethod P = 1.73416
Press any key to continue . . .
```

Rysunek 7 Wyniki programu dla testu2, n=1000

Graficzna interpretacja całki pod funkcją Rysunek 8.



Rysunek 8 Graficzne przedstawienie całki dla testu 2

Wynik uzyskany na stronie „*WolframAlpha*”:

$$\int_0^4 f(x)\,dx = 1.6536$$

- f(x) = e^{3x} ,na przedziale całkowania a=-1 b=3
  - Dla n = 10

```
Enter the number:
10
RectangularlMethod P = 4638.22
TrapezoidalMethod P = 3017.61
SimpsonMethod P = 2702.88
MCMethod P = 8646.72
Press any key to continue . . .
```

Rysunek 9 Wyniki programu dla testu3, n=10

  - Dla n = 100

```
Enter the number:
100
RectangularlMethod P = 2866.31
TrapezoidalMethod P = 2704.25
SimpsonMethod P = 2701.01
MCMethod P = 2395.55
Press any key to continue . . .
```

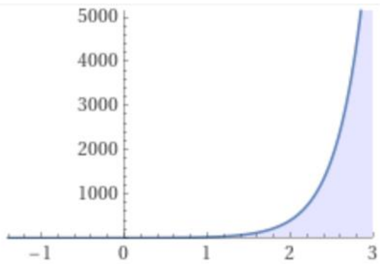
Rysunek 10 Wyniki programu dla testu3, n=100

  - Dla n = 1000

```
Enter the number:
1000
RectangularlMethod P = 2717.25
TrapezoidalMethod P = 2701.04
SimpsonMethod P = 2701.01
MCMethod P = 2710.19
Press any key to continue . . .
```

Rysunek 11 Wyniki programu dla testu3, n=1000

Graficzna interpretacja całki pod funkcją Rysunek 12.



Rysunek 12 Graficzne przedstawienie całki dla testu 3

Wynik uzyskany na stronie „*WolframAlpha*”:

$$\int_{-1}^{3} f(x)\,dx = 2701.0$$

5. Opracowanie wyników

Zestawienie zawarte w poniższej tabeli 1 przedstawia wszystkie wyniki dla konkretnych metod całkowania numerycznego porównane do wyników uzyskanych na stronie „Wolfram Alpha”.

Tabela 1. Opracowanie uzyskanych wyników testów

Numer Testu	**Wartość n**	**Metoda prostokątów**	**Metoda trapezów**	**Metoda Simpsona**	**Metoda MC**	**Wartość oczekiwana (Wolfram Alpha)**
1.1	10	2310	1750	1720	2566.43	1720
1.2	100	1776.3	1720.3	1720	1484.67	
1.3	1000	1725.6	1720	1720	1616.51	
2.1	10	1.48018	1.63154	1.65366	1.86013	1.6536
2.1	100	1.63829	1.65342	1.65364	1.82552	
2.3	1000	1.65213	1.65364	1.65364	1.73416	
3.1	10	4638.22	3017.61	2702.88	8646.72	2701.0
3.2	100	2866.31	2704.25	2701.01	2395.55	
3.3	1000	2717.25	2701.04	2701.01	2710.19	

Patrząc na uzyskane wyniki można zauważyć, że :

- Metoda prostokątów zwraca wynik który jest znacząco różny dla małej liczby prostokątów (n). Im mniejsza liczba tym wynik jest mniej dokładny.
- Metoda trapezów jest bardziej dokładniejsza niż poprzednia, ale również dla małych wartości n wynik jest znacząco różniący się od oczekiwanego.
- Metoda Simpsona jest najdokładniejszą metodą obliczania całki oznaczonej, już dla najmniejszy wartości n uzyskane wyniki są bardzo zbliżone lub prawie identyczne co z oczekiwanym.
- Metoda MC dla przeprowadzonych testów w większości miała znaczące różnice od oczekiwanego rezultatu.

6. Wnioski

Po przeprowadzonej analizie wyników można stwierdzić, że najdokładniejsze wyniki przybliżania całki oznaczonej zwraca metoda Simpsona – już od najmniejszej liczby przedziałów (n). Metoda trapezów jedynie dla bardzo małej ilość n zwracała wyniki niezgodne z oczekiwanymi. Najgorsze wyniki natomiast uzyskamy metodą Monte Carlo (MC) oraz prostokątów. Jeżeli chcielibyśmy, aby w tych przypadkach były one bardziej dokładne należałoby zwiększyć liczbę przedziałów – im większa tym większa dokładność.

7. Źródła

- Prezentacja autorstwa dr hab. inż. Marcina Hojnego „Całkowanie numeryczne – metoda prostokątów, trapezów, Simpsona i MC”.
- https://www.algorytm.edu.pl/algorytmy-maturalne/metoda-trapezow.html