

Wydział WIMiP	Imię i nazwisko: Zuzanna Będkowska	Rok: 2	Grupa: 1	Data: 19.04.2022
Metody Numeryczne	Temat: Całkowanie numeryczne			

Zadanie 1: Wyznaczenie wyniku całkowania danej funkcji dla podziału na 4 przedziały

Zgodnie z instrukcją, w trakcie zajęć stworzono następujący kod:

```

1  #include <iostream>
2  #include <fstream>
3  #include <cmath>
4  #include <iomanip>
5  #include <vector>
6
7  using namespace std;
8
9  double sinus(double x)
10 {
11     return sin(x);
12 }
13
14 double parabola(double x)
15 {
16     return (pow(x, 2) + 2 * x + 5);
17 }
18
19 double exp(double x)
20 {
21     double e = 2.718281828459;
22     return (pow(e, x));
23 }
24
25 double prostakaty(double a, double b, double &n, vector<double>& elementy, double (f)(double))
26 {
27     double s = (a - b) / n;
28     double wyn = 0.0;
29     for (int i = 0; i < n; ++i)
30     {
31         wyn += f(elementy[i] + s / 2);
32     }
33     wyn *= s;
34     return wyn;
35 }
36
37 double trapezy(double a, double b, double& n, vector<double>& elementy, double (f)(double))
38 {
39     double wyn = 0.0;
40     for (int i = 0; i < n; ++i)
41     {
42         wyn += (elementy[i] + 1 - elementy[i]) / 2 * (f(elementy[i]) + f(elementy[i] + 1)));
43     }
44     return wyn;
45 }
46
47 double parabole(double a, double b, double& n, vector<double>& elementy, double (f)(double))
48 {
49     double s = (a - b) / n;
50     double wyn = 0.0;
51     for (int i = 0; i < n; ++i)
52     {
53         wyn += (elementy[i] + 1 - elementy[i]) / 6 * (f(elementy[i]) + 4 * f((elementy[i] + 1 + elementy[i]) / 2) + f(elementy[i] + 1)));
54     }
55     return wyn;
56 }
57
58 void zadanie1()
59 {
60     cout << "Zadanie1:\n";
61     double n = 4;
62     vector<double> elementy(n + 1, 0);
63     cout << "Wybierz funkcje podcalkowa:\n\t1. sin(x) w przedziale 0,5 - 2,5\n\t2. x^2 + 2x + 5 w przedziale 0,5 - 5,0\n\t3. exp(x) w przedziale 0
64     int wybor;
65     cin >> wybor;
66     switch (wybor)
67     {
68     case 1:
69     {
70         for (int i = 0; i < n + 1; ++i)
71         {
72             elementy[i] = 0.5 + i * (2.5 - 0.5) / n;
73         }
74         double wynik = prostakaty(2.5, 0.5, n, elementy, sinus);

```

```

75     cout << "Wartosc calki metoda prostokatow: " << wynik << "\n";
76     wynik = trapezy(2.5, 0.5, n, elementy, sinus);
77     cout << "Wartosc calki metoda trapezow: " << wynik << "\n";
78     wynik = parabole(2.5, 0.5, n, elementy, sinus);
79     cout << "Wartosc calki metoda parabol: " << wynik << "\n";
80     break;
81 }
82 case 2:
83 {
84     for (int i = 0; i < n + 1; ++i)
85     {
86         elementy[i] = 0.5 + i * (5.0 - 0.5) / n;
87     }
88     double wynik = prostakaty(5.0, 0.5, n, elementy, parabola);
89     cout << "Wartosc calki metoda prostokatow: " << wynik << "\n";
90     wynik = trapezy(5.0, 0.5, n, elementy, parabola);
91     cout << "Wartosc calki metoda trapezow: " << wynik << "\n";
92     wynik = parabole(5.0, 0.5, n, elementy, parabola);
93     cout << "Wartosc calki metoda parabol: " << wynik << "\n";
94     break;
95 }
96 case 3:
97 {
98     for (int i = 0; i < n + 1; ++i)
99     {
100         elementy[i] = 0.5 + i * (5.0 - 0.5) / n;
101     }
102     double wynik = prostakaty(5.0, 0.5, n, elementy, exp);
103     cout << "Wartosc calki metoda prostokatow: " << wynik << "\n";
104     wynik = trapezy(5.0, 0.5, n, elementy, exp);
105     cout << "Wartosc calki metoda trapezow: " << wynik << "\n";
106     wynik = parabole(5.0, 0.5, n, elementy, exp);
107     cout << "Wartosc calki metoda parabol: " << wynik << "\n";
108     break;
109 }
110 default:
111     break;
112 }
113 }

```

Każdą z danych funkcji podcałkowych zaimplementowano jako funkcję, która zostaje podana jako argument każdej z metod całkowania.

Wyniki:

- Dla $\int_{0.5}^{2.5} \sin(x) dx$:

```

Konsola debugowania programu Microsoft Visual Studio
Zadanie1:
Wybierz funkcje podcalkowa:
1. sin(x) w przedziale 0,5 - 2,5
2. x^2 + 2x + 5 w przedziale 0,5 - 5,0
3. exp(x) w przedziale 0,5 - 5,0
Twój wybór: 1
Wartosc calki metoda prostokatow: 1.69634
Wartosc calki metoda trapezow: 1.64361
Wartosc calki metoda parabol: 1.67876
C:\Users\Zuza\source\repos\MetodyNumeryczne_lab6\x64\Debug\MetodyNumeryczne_lab6.exe (proces 3196) zakończono z kodem 0.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

- Dla $\int_{0.5}^{5.0} (x^2 + 2x + 5) dx$:

```

Konsola debugowania programu Microsoft Visual Studio
Zadanie1:
Wybierz funkcje podcalkowa:
1. sin(x) w przedziale 0,5 - 2,5
2. x^2 + 2x + 5 w przedziale 0,5 - 5,0
3. exp(x) w przedziale 0,5 - 5,0
Twój wybór: 2
Wartosc calki metoda prostokatow: 88.4004
Wartosc calki metoda trapezow: 89.8242
Wartosc calki metoda parabol: 88.875
C:\Users\Zuza\source\repos\MetodyNumeryczne_lab6\x64\Debug\MetodyNumeryczne_lab6.exe (proces 7036) zakończono z kodem 0.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

- Dla $\int_{0.5}^{5.0} e^x dx$:

```
Konsola debugowania programu Microsoft Visual Studio

Zadanie1:
Wybierz funkcje podcalkowa:
    1. sin(x) w przedziale 0,5 - 2,5
    2. x^2 + 2x + 5 w przedziale 0,5 - 5,0
    3. exp(x) w przedziale 0,5 - 5,0
Twój wybór: 3
Wartosc calki metoda prostokatow: 139.301
Wartosc calki metoda trapezow: 161.927
Wartosc calki metoda parabol: 146.843

C:\Users\Zuza\source\repos\MetodyNumeryczne_lab6\x64\Debug\MetodyNumeryczne_lab6.exe (proces 18312) zakończono z kodem 0
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Gdzie wynik rzeczywisty to: 147.413

Zadanie 2: Wyznaczenie wyniku całkowania funkcji e^x dla zwiększającej się ilości przedziałów

Kod z zadania 1 zmodyfikowano w następujący sposób:

```
115 void zadanie2()
116 {
117     cout << "Zadanie2:\n";
118     cout << "Funkcja podcalkowa: exp(x) w przedziale 0,5 - 5,0\n";
119     int przedzialy = 1000;
120     vector<double> pomoc(3, 0);
121     vector<vector<double>> wyniki(przedzialy, pomoc);
122     ofstream czytaj("wynik.txt");
123     for (double n = 1; n <= przedzialy; ++n)
124     {
125         vector<double> elementy(n + 1, 0);
126         for (int i = 0; i < n + 1; ++i)
127         {
128             elementy[i] = 0.5 + i * (5.0 - 0.5) / n;
129         }
130         double wynik = prostakaty(5.0, 0.5, n, elementy, exp);
131         wyniki[n - 1][0] = wynik;
132         wynik = trapezy(5.0, 0.5, n, elementy, exp);
133         wyniki[n - 1][1] = wynik;
134         wynik = parbole(5.0, 0.5, n, elementy, exp);
135         wyniki[n - 1][2] = wynik;
136     }
137     czytaj << "n" << " " << "p" << " " << "t" << " " << "s" << "\n";
138     int licz = 1;
139     for (auto i : wyniki)
140     {
141         czytaj << licz << " ";
142         for (auto j : i)
143         {
144             czytaj << setprecision(15) << j << " ";
145         }
146         czytaj << "\n";
147         licz++;
148     }
149 }
```

Wynik działania programu:

```
Konsola debugowania programu Microsoft Visual Studio

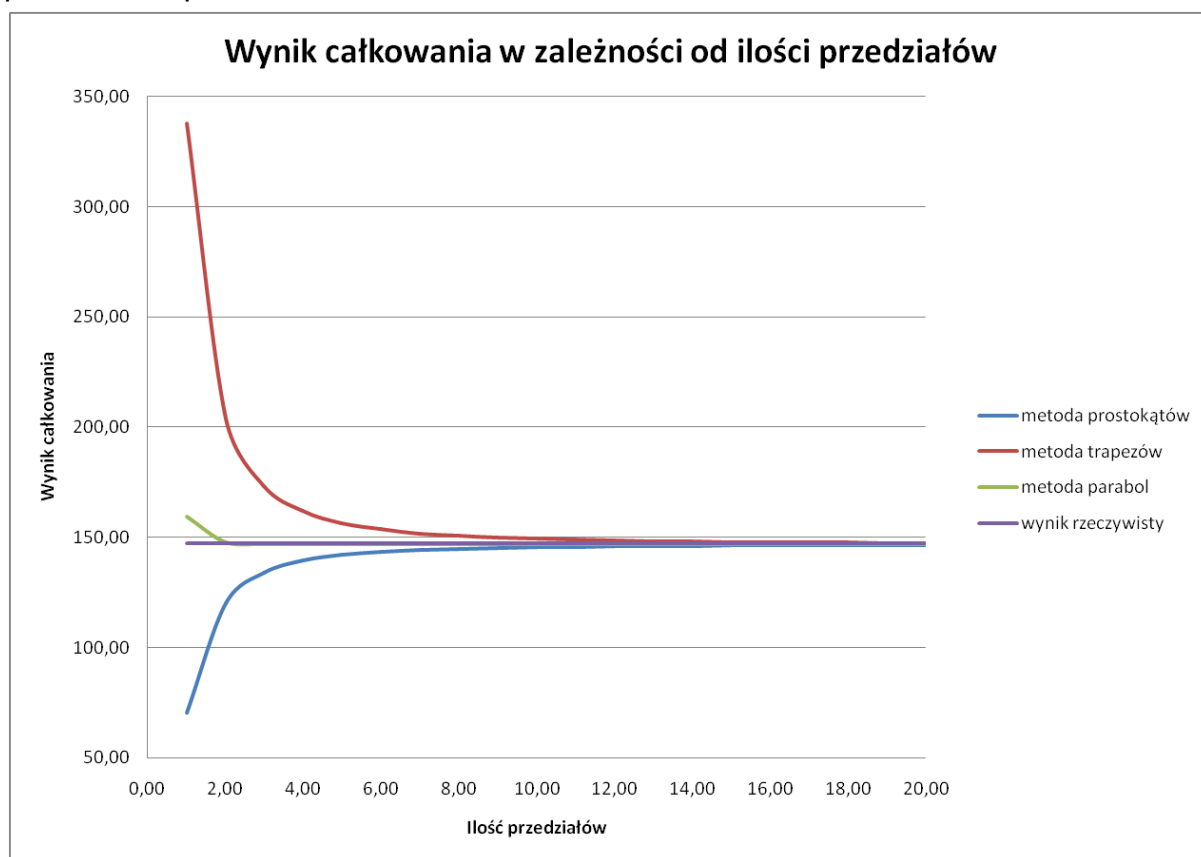
Zadanie2:
Funkcja podcalkowa: exp(x) w przedziale 0,5 - 5,0

C:\Users\Zuza\source\repos\MetodyNumeryczne_lab6\x64\Debug\MetodyNumeryczne_lab6.exe (proces 18684) zakończono z kodem 0
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Plik wynik.txt:

	n	p	t	s
1	70.3918434788435	337.639230839845	159.474305932511	
2	119.83751398862	204.015537159344	147.896855045528	
3	133.857580782498	173.303172366379	147.006111310459	
4	139.301372116719	161.926525573982	146.843089935807	
5	141.925711887046	156.539826362481	146.797083378858	
6	143.380255488402	153.580376574439	146.780295850414	
7	144.267377406933	151.784351024709	146.773035279525	
8	144.847263467677	150.613948845351	146.769491926902	
9	145.246716080869	149.809365680298	146.767599280679	
10	145.533387904313	149.232769124764	146.766514977797	
11	145.746002305765	148.805569475418	146.765858028982	
12	145.908004055925	148.48031603142	146.76544138109	
13	146.034253899304	148.226992852166	146.765166883592	
14	146.134538076429	148.025864215821	146.764980122893	
15	146.215512482806	147.863523545809	146.764849503807	
16	146.281830834155	147.730606156514	146.764755941608	
17	146.336825768193	147.620410988879	146.764687508421	
18	146.382934346012	147.528040880583	146.764636524203	
19	146.421971921447	147.449849895297	146.76459791273	
20	146.455313102626	147.383078514539	146.76456823993	
21	146.484014140617	147.325607120515	146.764545133916	
22	146.508897443102	147.275785890591	146.764526925599	
23	146.530611034823	147.232315191773	146.764512420473	
24	146.549671103898	147.194160043672	146.764500750489	
25	146.566492915744	147.160487996401	146.764491275963	

Zamiast stałej ilości przedziałów, obliczenia wykonano w pętli dla ilości przedziałów $n \in \langle 1, 1000 \rangle$. Otrzymane wyniki z każdej z metod zapisano w pliku, na podstawie którego wykonano wykres. W celu zwiększenia czytelności, wykorzystano pierwsze 20 pomiarów:



Zgodnie z wykresem, można zauważyć, że dla małej ilości przedziałów, najdokładniejsza jest metoda prostokątów. Żadna jednak z metod nie jest wystarczająco bliska wyniki rzeczywistego.