

応用解析レポート

氏名: 重近 大智 (SHIGECHIKA, Daichi)

学生番号: 09501527

提出日: 2020 年 08 月 04 日

締切日: 2020 年 08 月 06 日

目次

1	概要	2
2	プログラムの説明と計算結果	2
3	考察	2
4	作成したプログラム	2

1 概要

本レポートでは台形積分及びシンプソン積分を行う C 言語のプログラムを作成し、その積分の結果、絶対誤差と考察を報告する。なお各積分において、分割数が $n = 2, 4, 8, 16$ の 4 通りの場合を計算する。

2 プログラムの説明と計算結果

計算結果と絶対誤差は、表 1 に示す。また、計算に使用したプログラムは 4 章に示す。

簡単にプログラムの各関数の役割について説明する。プログラムの 9–17 行目で宣言されている `f()` 関数で、今回の被積分関数である $1/x^3$ の計算を行う。19–41 行目で宣言されている `trape()` 関数は、台形積分の計算を行う。43–67 行目で宣言されている `simp()` 関数は、シンプソン積分の計算を行う。69–84 行目で宣言されている `main()` 関数では、積分方法と分割数を `printf()` 関数により表示し、使用者に入力を求める。積分方法で 1 を選択すると台形積分、2 を選択するとシンプソン積分となる。

3 考察

今回の積分の真値は 0.375 である。台形積分では分割数が 2 の場合に、大きな絶対誤差がある。それに対し、シンプソン積分では分割数が 2 の場合でも、誤差は約 0.01 と少ない。分割数を増やしていくといずれの場合も、台形積分に比べシンプソン積分の方が絶対誤差が小さい。16 分割したシンプソン積分では小数第 5 位までの絶対誤差がなくなり、かなり精度の高い計算が行われている。以上の結果からシンプソン積分の方が台形積分に比べ、速く真値に収束することが分かる。

4 作成したプログラム

作成したプログラムを以下に添付する。プログラムは 84 行からなる。

```
1      #include <stdio.h>
2      #include <math.h>
3      #include <stdlib.h>
4
5      double a = 1;
6      double b = 2;
7      double t_value = 0.375;
8
9      double f(double x)
10     {
11         double y;
12
```

表 1 計算結果と絶対誤差

積分方法	分割数 n	計算結果	絶対誤差
台形積分	2	0.429398	0.054398
	4	0.389346	0.014346
	8	0.378642	0.003642
	16	0.375914	0.000914
シンプソン積分	2	0.385031	0.010031
	4	0.375996	0.000996
	8	0.375074	0.000074
	16	0.375005	0.000005

```

13     x = x * x * x;
14     y = 1 / x;
15
16     return y;
17 }
18
19 void trape(int part)
20 {
21     int i;
22     double h;
23     double x;
24     double y_l;
25     double y_r;
26     double S = 0;
27
28     h = fabs((b - a) / part);
29     x = a;
30
31     for(i = 0; i < part; i++)
32     {
33         y_l = f(x);
34         y_r = f(x + h);
35         S = S + ((y_l + y_r) * h) / 2;
36         x = x + h;
37     }
38     fprintf(stderr, "\n");
39     printf("Trapezoidal Answer:%f\n", S);
40     printf("The difference      :%f\n", fabs(S - t_value));
41 }
42
43 void simp(int part)
44 {
45     int i;
46     double h;
47     double x;
48     double y_l;
49     double y_c;
50     double y_r;
51     double S = 0;
52
53     h = fabs((b - a) / part);
54     x = a;
55
56     for(i = 0; x < b; i++)
57     {
58         y_l = f(x);
59         y_c = f(x + h);
60         y_r = f(x + 2 * h);
61         S = S + h * (y_l + 4 * y_c + y_r) / 3;
62         x = x + 2 * h;
63     }
64     fprintf(stderr, "\n");
65     printf("Simpson's Answer:  %f\n", S);
66     printf("The difference   :  %f\n", fabs(S - t_value));
67 }
68
69 int main(void)
70 {
71     int mode = 0;
72     int part = 2;
73
74     fprintf(stderr, "Please select integral rule. (1:Trapezoidal rule 2:Simpson's ru
le)\n");
75     scanf("%d", &mode);
76
77     fprintf(stderr, "Please input number of partitions.\n");
78     scanf("%d", &part);
79

```

```
80         if(mode == 1) trape(part);
81         if(mode == 2) simp(part);
82
83     return 0;
84 }
```