Research Letter

Performance comparison of approximation area by Monte-Carlo simulation and trapezoidal rule

Shinichi Funase (Faculty of Production Systems Engineering and Sciences, Komatsu University, shinichi.funase@komatsu-u.ac.jp, Japan)
Tomohiro Kado (Faculty of Production Systems Engineering and Sciences, Komatsu University, Japan)
Toshitsugu Moriuchi (Faculty of Production Systems Engineering and Sciences, Komatsu University, Japan)
Haruhiko Kimura (Faculty of Production Systems Engineering and Sciences, Komatsu University, haruhiko.kimura@komatsu-u.ac.jp, Japan)
Takashi Oyabu (Nihonkai International Exchange Center, oyabu24@gmail.com, Japan)

Abstract

An area means a kind of largeness of a shape on planes and curved surfaces. It is used when the largeness of a land or an office space are derived as a familiar case. There is a wide range of applications. For example, various kinds of physical quantities, population and number of votes are predicted using area density and there are many methods to derive the area largeness. Simple formulas are used to derive the area of a circle or square, and these are learned in elementary and junior high schools. Also, when finding the area of land, Heron's formula is used by laying out some triangles. Furthermore, there is a method to obtain it by the definite integral. However, the definite integral cannot be used when a formula of the indefinite integral for the shape is not understood. Namely, it is not suitable in the case of complex function. On the other hand, the Monte-Carlo method and trapezoidal rule are well known as methods of computer processing at deriving approximate area. The performance comparison of area approximation by the Monte-Carlo method and trapezoidal rule is performed in this study, in which both simple and complex shapes are processed. Optimal number of iteration and divide number are implemented to introduce a target accuracy, when using the Monte-Carlo method and trapezoidal rule. Moreover, the iteration number n_1 in the Monte-Carlo method and the divided number n_2 in trapezoidal rule are derived and the CPU times for both of the methods are introduced in the same precision.

Key words

area approximation, Monte-Carlo method, trapezoidal rule, definite integral, uniform random number

1. Introduction

There are many cases to derive the area largeness in our daily life, for example deriving a paving area. In that case, the land is divided into several triangles and the largeness of each triangle is derived using Heron's formula (Cayley-Menger Determinant, 2018). The derived total area is the desired largeness.

When the object is a simple shape, the area is derived easily using a simple formula which is learned in elementary and junior high schools. In addition, the definite integral method is used to derive the area largeness (Bourbaki, 2004). An interval of a function (i.e. exponentiation and trigonometric function) is divided into several parts in the method and the area is derived. The following process is carried out in the definite integral method, namely the start and end points of integral interval are substituted into the indefinite integral formula, and the difference is derived. The difference is the definite integral value. Although there is a myriad of indefinite integral functions for $f(x)$, the function can be identified in the definite integral formula. It can be seen by the definite integral when adding a constant C of integration in the indefinite integral. The integral constants are canceled by the subtraction and the result of the integral becomes irrelevant to the constant C. Therefore, the definite integral value is determined if the function $f(x)$ to be integrated and the integral interval $i \leq x \leq j$ is determined. However, this process cannot be utilized when the formula of the indefinite integral is not understood. Although the definite integral method can be calculated quickly, it cannot be utilized for a complex function because the formula is not understood. The Monte-Carlo method (Motwani and Raghavan, 1995; Sugihara and Murota, 2003; Suzuki and Goda, 2020) and trapezoidal rule (Richard and Douglas, 2000) are generally utilized when using a computer. The definite integral is performed using uniform random numbers in the method and the approximate value of the definite integral is derived by calculating the area sum of trapezoids. This is one of the approximate calculations and it is not obvious how many divisions and iterations to make according to the required accuracy. Generally, the numbers of divisions and iterations have to be increased to improve accuracy when the shape of the calculating area becomes complex. This study is conducted to obtain a measure of calculation accuracy using the Monte-Carlo method and trapezoidal rule for target areas which are simple and complex shapes. Moreover, the Monte-Carlo method at the iteration number n_1 and trapezoidal rule at the divided number n_2 are carried out and the areas are derived and the CPU times for both methods are introduced for the same precision.

Furthermore, the target area is the definite integral $S = \int_a^b f(x)dx$ in this paper. It is more common using triangles in the case of finding the area of various shapes. However, in the case of the definite integral, trapezoidal rule is used because the shape of the area to be found is easily approximated by a trapezoid. When using triangles, the area to be found must be covered by the set of triangles and the area of each triangle must be found. This is possible if each trapezoid generated by equally dividing the time axis is divided into two triangles. This is exactly trapezoidal rule.

2. How to calculate an approximation area by Monte-Carlo method

It is called a random number when there is not a causal relationship before and after each number. There are probability distribution random numbers such as uniform distribution random numbers, normal distribution random numbers, exponential distribution random numbers, and Poisson distribution random numbers in random numbers. It is determined which random number is used by the appearance probability of each number. Among them, when using the uniformly distributed random numbers, which are particularly important, each number appears independently with equal probability. Namely, there is not a causal relationship before and after each number, but if a frequency for each number is taken, it is possible to understand that each number appears at the same times. The Monte-Carlo method is a general term for simulations using these probability distribution random numbers and all problems including a probabilistic factor is subject to the Monte-Carlo method. In this paper, we deal with a method to approximate $S = \int_a^b f(x)dx$ using the Monte-Carlo method. The value of S is the shaded area in Figure 1. As shown in Figure 2, the area V of the smallest square that includes the shaded area completely is derived. Next, n dots are randomly sprinkled in this square using the uniform random number. In short, x coordinate x_t and y coordinate y_t for dots v_t is set each established as below.

$$x_t = (b - a) \cdot \text{extended rand}() + a$$
$$y_t = c \cdot \text{extended rand}()$$

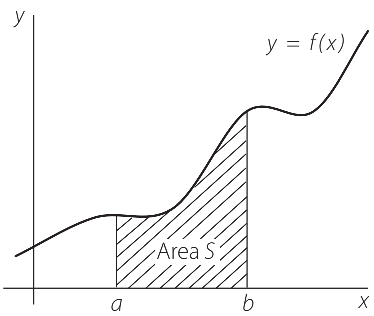

Figure 1: Mean of $S = \int_a^b f(x)dx$

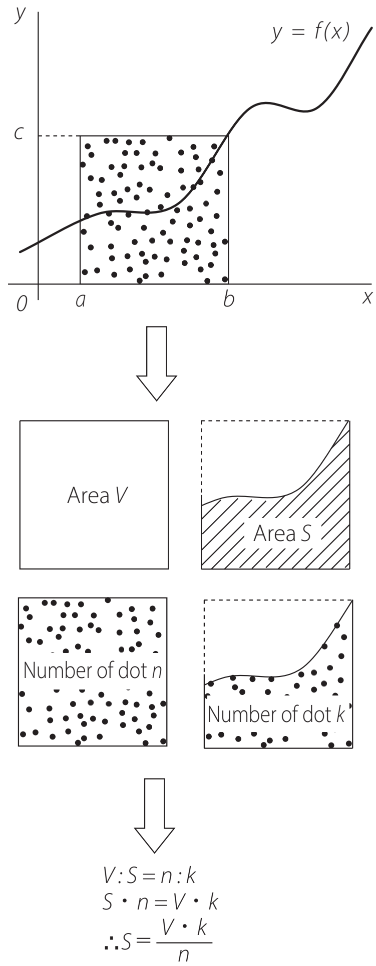

Figure 2: Approximate solution of $S = \int_a^b f(x)dx$ by Monte-Carlo method

The uniform random number usually uses the *rand(1)* but the accuracy is not so good. Therefore, we use the extended *rand()* in this paper. The other, Mersenne twister and Xorshift is used for other uniform random numbers. The extended *rand()* is a uniform random number as follows. The extended *rand()* uses a built-in function of *C* language. By dividing *rand()+0.5* by *RAND_MAX+1*, it will be evenly distributed without being biased toward both ends. However, since the value of *RAND_MAX* is relatively small depending on the processing system, a uniform random number is calculated using multiple values of the *rand()* function as follows.

```
double rand(){
double m, a;
m=RAND_MAX+1;
a=(rand()+0.5)/m;
a=(rand()+a)/m;
return (rand()+a)/m;
}
```

Count the number of dots that are in shaded area within n pieces dots sprinkled in square, then calculate the S using $V : S = n : k$ which is the nature of a random number. Here, the necessary and sufficient condition for the dot vt entering in the shaded area is,

$f(x_t) \geqq y_t$

This is shown in Figure 3.

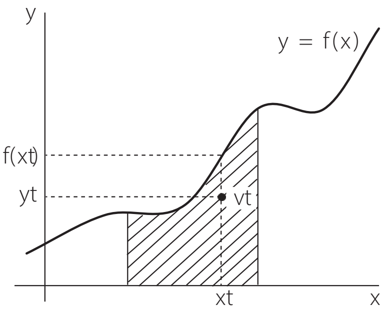

Figure 3: Necessary and sufficient condition for the dot vt entering in shaded area ($f(x_t) \geqq y_t$)

And, the flowchart to solve the approximate solution of $S = \int_a^b f(x)dx$ by the Monte-Carlo method is shown in Figure 4.

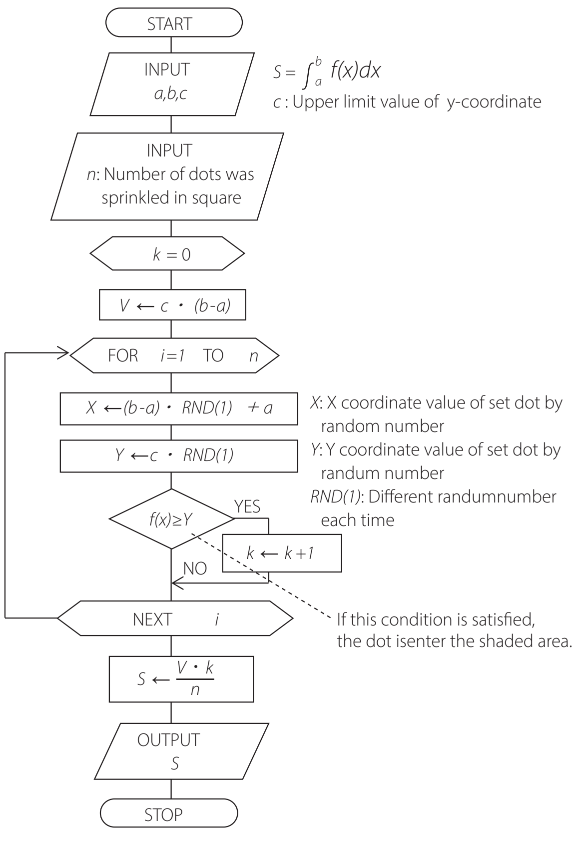

Figure 4: Flowchart to solve the approximate solution of $S = \int_a^b f(x)dx$ by Monte-Carlo method

3. How to calculate an approximation area with trapezoidal rule

Trapezoidal rule divides the shape of the target definite integral (the find area) into multiple trapezoids, and it is a method of approximately calculating the definite integral by summing the area of each trapezoid. For example, the area S of the shape which is sandwiched between the curve $y = f(x)$ shown in Figure 1 and the specified range $[a, b]$ on the x-axis, equally n-decompose between a and b, find the coordinates of intersection of $x = x_i$ and $y = f(x)$ from the xi of x-axis at that time, if the adjacent intersection is connected with a straight line, and it is understood that the area S is approximately found as the summation of n-pieces trapezoids. Also, if the value of n is larger, the accuracy of approximation is improved. Figure 5 shows the shape in a case with the range $[a, b]$ divided into 3. In this case, the approximation of S is $S_1 + S_2 + S_3$. The flowchart using trapezoidal rule which derives the area S of the figure sandwiched between the curve $y = f(x)$ and the x-axis is shown in Figure 6.

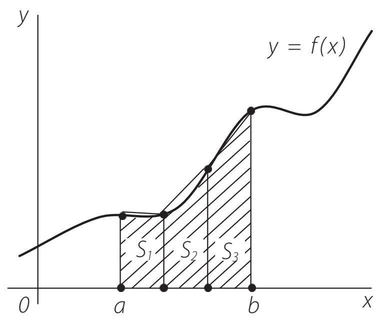

Figure 5: Trapezoidal rule when $S = \int_a^b f(x)dx$ is divided into three parts

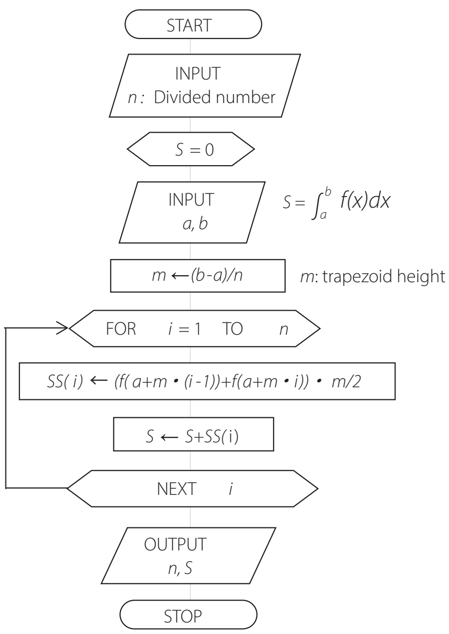

Figure 6: Flowchart which derives the area $S = \int_a^b f(x)dx$ using trapezoidal rule

4. Performance comparison of both methods

In this section, the performance comparison of the Monte-Carlo method and trapezoidal rule for approximate area is performed by experiments. The shape of the target area is $\int_1^3 x^2dx$ as a simple one, and $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$ as a complicated one. Both diagrams are shown in Figure 7 and Figure 8, respectively. When using the Monte-Carlo method, the area of the square where the dots are sprinkled is the same for both a simple one and a complicated one. In this case both are 18. The accuracy of experimental values in this paper is defined by the following equation.

Accuracy (%) = (Theory – |Experiment – Theory|) * 100 / Theory

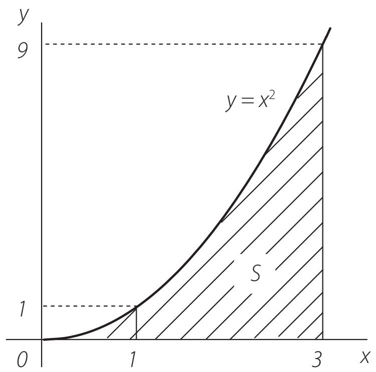

Figure 7: Shape of $\int_1^3 x^2dx$

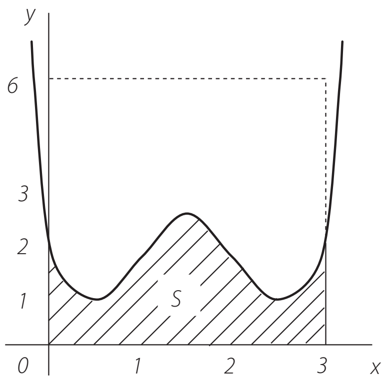

Figure 8: Shape of $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$

4.1 Experiment 1

For $\int_1^3 x^2dx$, the iteration number n1 is closest to 70 %, 75 %, 80 %, 85 %, 90 %, 95 % accuracy using the Monte-Carlo method, respectively. However, the upper limit (c) of the y-coordinate will be *9*.

4.1.1 Result

Table1 shows the result. However, "average accuracy of 100 times" is the average accuracy when 100 simulations with the stated iteration number are performed.

Table 1: Iteration number for accuracy when finding $\int_1^3 x^2dx$ using Monte-Carlo method

Accuracy (%)	Iteration number (n_1)	Average accuracy of 100 times (%)
70	8	70.230
75	11	75.979
80	18	80.115
85	30	84.408
90	80	90.552
95	250	94.884

4.2 Experiment 2

For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$, the iteration number n_1 is closest to 70 %, 75 %, 80 %, 85 %, 90 %, 95 % accuracy using the Monte-Carlo method, respectively. However, the upper limit (c) of the y-coordinate will be *6*.

4.2.1 Result

The result is shown in Table 2.

Table 2: Iteration number for accuracy when finding $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$ using Monte-Carlo method

Accuracy (%)	Iteration number (n_1)	Average accuracy of 100 times (%)
70	21	69.025
75	25	75.247
80	38	79.783
85	85	85.460
90	120	89.353
95	550	94.308

4.3 Experiment 3

For $\int_1^3 x^2dx$, using trapezoidal rule is respectively the divided number n2 closest to accuracy 70 %, 75 %, 80 %, 85 %, 90 % and 95 %. However, if there is no corresponding accuracy, it will be omitted.

4.3.1 Result

The result is shown in Table 3. The accuracy of the approximate area is already high from the stage where the divided number is 1, reaching about 85 %. Also, when the divided number becomes 2, the accuracy of approximate area is over 95 %. The accuracy of the approximate area for the divided number is shown in Table 4. And, Figure 9 shows the graph of Table 4.

4.4 Experiment 4

For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$, using trapezoidal rule respectively the divided number n_2 is closest to accuracy 70 %, 75 %, 80 %, 85 %, 90 % and 95 %. However, if there is no cor-

Table 3: Divided number for accuracy when finding $\int_1^3 x^2 dx$, using trapezoidal rule

Accuracy (%)	Divided number (n_2)	Accuracy of approximate area (%)
70		
75		
80		
85	1	84.615
90		
95	2	96.154

Table 4: Accuracy of approximate area for the divided number of Experiment 3

Divided number (n_2)	Accuracy of approximate (%)
1	84.615
2	96.154
3	98.291
4	99.038
5	99.385
6	99.573
7	99.686
⋮	⋮
30	99.983

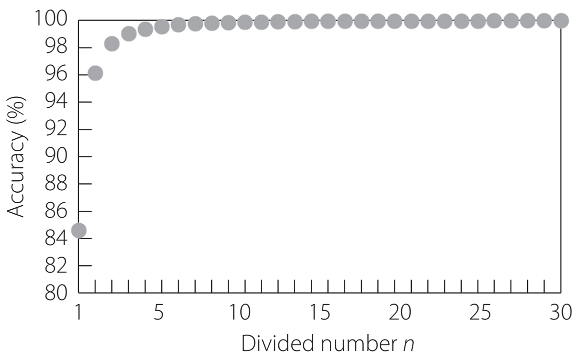

Figure 9: Graph of divided number and accuracy from Table 4

responding accuracy, it will be omitted.

4.4.1 Result

The result is shown in Table 5. Also, the accuracy of the approximate area for the divided number is shown in Table 6. And, Figure 10 shows the graph of Table 6.

4.5 Experiment 5

For $\int_1^3 x^2 dx$, the accuracy of trapezoidal rule divided number n_2 closest to accuracy of the Monte-Carlo method of the iteration number n1 is derived. However, derive n_1 as 100, 1000 and 10000. Also, when using the Monte-Carlo method, the

Table 5: Divided number for accuracy when finding $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\, dx$ using trapezoidal rule

Accuracy (%)	Divided number (n_2)	Accuracy of approximate area (%)
70	2	65.809
75		
80	1, 3	82.353
85		
90	4	89.591
95	6	95.221

Table 6: Accuracy of approximate area for the divided number of Experiment 4

Divided number (n_2)	Accuracy of approximate (%)
1	82.353
2	65.809
3	82.353
4	89.591
5	93.195
6	95.221
7	96.465
⋮	⋮
24	99.694
⋮	⋮
30	99.804

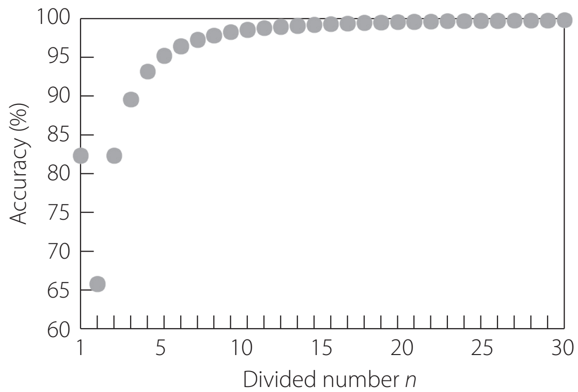

Figure 10: Graph of divided number and accuracy from Table 6

upper limit value (c) of y coordinate is set to 9.

4.5.1 Result

The result is shown in Table 7.

4.6 Experiment 6

For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\, dx$, the accuracy of trapezoidal rule divided number n_2 closest to the accuracy of the

Table 7: For $\int_1^3 x^2 dx$, the accuracy of trapezoidal rule divided number n_2 closest to the accuracy of the Monte-Carlo method of the iteration number n_1

n_1	Accuracy of Monte-Carlo method (%)	n_2	Accuracy of trapezoidal rule (%)
100	91.832	2	96.154
1000	97.707	3	98.291
10000	99.066	4	99.038

Monte-Carlo method of the iteration number n_1 is derived. However, derive n_1 as 100, 1000 and 10000. Also, when using the Monte-Carlo method, the upper limit value (*c*) of y coordinate is set to *6*.

4.6.1 Result

The result is shown in Table 8.

Table 8: For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$, the accuracy of trapezoidal rule divided number n_2 closest to the accuracy of the Monte-Carlo method of the iteration number n_1

n_1	Accuracy of Monte-Carlo method (%)	n_2	Accuracy of trapezoidal rule (%)
100	86.600	4	89.591
1000	95.973	7	96.465
10000	98.827	12	98.781

4.7 Experiment 7

For $\int_1^3 x^2 dx$, when the accuracy between the Monte-Carlo method and trapezoidal rule is the closest, derive both of the CPU time. However, the accuracy of basis is set to 70 %, 75 %, 80 %, 85 %, 90 %, 95 % but if there is no corresponding accuracy in both approximation area of accuracy, the CPU time is omitted. Also, when using the Monte-Carlo method, the upper limit value (*c*) of y coordinate is set to *9*. The experiment used a personal computer, LIFEBOOK SH75/C3, FMVS75DUV1 by Fujitsu.

4.7.1 Result

The result is shown in Table 9. However, the CPU time (s) of the Monte-Carlo method is taken by an average value from 100 times experiments because of using a random number.

4.8 Experiment 8

For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$, when the accuracy between the Monte Carlo method and trapezoidal rule is the closest, derive both of the CPU time. However, the accuracy of basis is set to 70 %, 75 %, 80 %, 85 %, 90 %, 95 % but if there is no corresponding accuracy in both approximation area of accuracy, the CPU time is omitted. Also, when using the Monte-Carlo method, the upper limit value (*c*) of y coordinate is set to *6*. The experiment used a personal computer, LIFEBOOK SH75/C3, FMVS75DUV1 by Fujitsu.

4.8.1 Result

The result is shown in Table 10. However, the CPU time (s)

Table 9: For $\int_1^3 x^2 dx$, the CPU time when the accuracy between the Monte-Carlo method and trapezoidal rule is the closest

Accuracy of Monte-Carlo method (%)	n_1	CPU time (s)	Accuracy of trapezoidal rule (%)	n_2	CPU time (s)
69.423	8				
75.077	11				
80.692	18				
85.777	30	0.594	84.614	1	under 0.001
90.939	80				
94.904	250	0.640	96.153	2	under 0.001

Table 10: For $\int_0^3 \{x(x-1)(x-2)(x-3)+2\}\,dx$, the CPU time when the accuracy between the Monte-Carlo method and trapezoidal rule is the closest

Accuracy of Monte-Carlo method (%)	n_1	CPU time (s)	Accuracy of trapezoidal rule (%)	n_2	CPU time (s)
69.714	21				
74.541	25				
80.947	38	0.609	82.353	1, 3	Under 0.001
84.415	85				
89.676	120	0.656	89.591	4	0.016
94.712	550				

of the Monte-Carlo method is taken by an average value from 100 times experiments because of using a random number.

5. Conclusion

As an overall evaluation, the efficiency of trapezoidal rule was higher than the efficiency of the Monte-Carlo method. However, we realized that there were faults. The following is a comparison of both methods from the evaluation of each experiment.

In Experiment 1, if a required area was a simple shape, in order to output about 70 % accuracy by the Monte-Carlo method, it was necessary for the iteration number of about 10 times. And, in order to output over 95 % accuracy, the iteration number became suddenly big and it was necessary for over 250 times.

In Experiment 2, because the required area was a complex shape, the iteration number that was necessary increased. In order to output about 70 % accuracy by the Monte-Carlo method, it was necessary for the iteration number of over 20 times. And, in order to output over 95 % accuracy, it became suddenly big and it was necessary for the iteration number of over 550 times.

In Experiment 3, from the step of the divided number 1, already the accuracy of approximation area increased to near 85 %. In the divided number 2, the accuracy of approximation area went over 95 %. Then, as the divided number increased, a log curve was drawn and converged to 100 %.

The divided number of Experiment 4 was necessary more than Experiment 3, but the accuracy of the divided number 2 decreased more than the divided number 1 in Experiment 4, as the divided number increased to 3, 4, 5, etc., the accuracy increased while drawing a log curve and converged to 100 %. The phenomenon in which the accuracy does not improve even if the number of the divided number increases is that the shape of the area is an electrocardiogram with a large number of cycles, a sound waveform, a vibration waveform, a sawtooth waveform, and a function such as a comb and it happens when the difference in height is big. Trapezoidal rule is prone to error for such an area of shape, so this can be said to be a fault.

In Experiment 5 and 6, trapezoidal rule achieves higher accuracy with a smaller divided number, but the Monte-Carlo method requires an order of magnitude with more iterations to achieve the same accuracy. Also, in order to realize the improvement obtained by increasing the divided number by 1 in the Monte-Carlo method, it was necessary to increase the iteration number tenfold. Furthermore, as the function governing the outline of the target area became more complex, the divided number was required to maintain the same accuracy increased rapidly. Trapezoidal rule has faults as mentioned above, although the Monte-Carlo method has flexibilities and it can be used for any area of shape.

From Experiment 7 and 8, it can be judged that the efficiency of trapezoidal rule is overwhelmingly higher about CPU time. However, this was previously mentioned. It is predictable that even if the divided number is increased, the result as expected may not be produced when the area of the shape is an electrocardiogram with a large number of cycles, a sound waveform, a vibration waveform, a sawtooth waveform, a function such as a comb, and the height difference is big. This is the weak case for trapezoidal rule.

References

Bourbaki, N. (2004). *Integration I*. Springer Verlag.

Cayley-Menger Determinant (2018). Mathworld (Retrieved December 3, 2022 from https://mathworld.wolfram.com/Cayley-MengerDeterminant.html).

Motwani, R. and Raghavan, P (1995). *Randomized algorithms*. Cambridge University Press.

Richard, L. B and Douglas, J. F. (2000). *Numerical analysis*. Brooks-Cole.

Sabitov, I. Kh.(1998). The volume as a metric invariant of polyhedra. *Discrete & Computational Geometry*, Vol. 20, 405-425.

Sugihara, M. and Murota, K. (2003). *Suuchikeisan no suuri* [Mathematics of numerical methods]. Iwanami Shoten. (in Japanese)

Suzuki, K. and Goda, T. (2020). The state of the art in Quasi-Monte Carlo Methods. *JSIAM*. Vol. 30, No. 4, 320-374.

(Received: April 12, 2023; Accepted: May 2, 2023)