Montecarlo Methods to Efficently Compute Volume of Solids

Kamil Czapla[1], Mariusz Pleszczyński[2]

[1]Academic High School of Silesian University of Technologies, Rudzka 13, Rybnik, 44–200, Poland
[2]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, Gliwice, 44–100, Poland

Abstract
In this work we present and compare chosen Monte Carlo methods, used to calculate solid's volume. Such solids are defined with a certain surface $\varrho(\varphi, \theta)$ wich results in classical methods being hard or impossible to apply. In designating those volume, due to the desire to shorten calculation time, we will apply parallel calculations.

Keywords
Double integral, Monte Carlo methods, parallel computations

1. Introduction

In classic task of calculating solid's volume we use double integral (or triple integral), often utilizing variable's conversion. Most often they are polar, cylindrical or spherical coordinates. The use of this approach most often occurs, when surfaces limiting given solid are given in open form $z = f(x, y)$en however, whether for curves in a plane or for surfaces in space, such limitation are given different (non-classical) way. For curve this could be for example: parametric or polar form. In case of a surface from R^3 surface, the situation may look similar, but then there is an additional variable. Wanting to calculate area of a given flat surface or volume of a surface in space, appropriate formulas should be designated. Authors discussed their use in [1] for flat areas and in [2] for spatial areas. Especially in the R^3 space, where surface limiting given volume is dependent from $\varrho(\varphi, \theta)$ suitable integral describing this volume may be hard or even impossible to determine analytically. In this work we propose the use of certain approximate methods on calculating numerically given volume, independent from the form of the funcion $\varrho(\varphi, \theta)$

2. Domains in R^3 space

In which the volume V is described by the integral:

$$V = \iint_D (f_1(x, y) - f_2(x, y))\, dxdy, \tag{1}$$

where continuous in D domain functions f_1 and f_2 are limiting the space above and below D domain, from above and below respectively. If the domain D s normal (e.g) with respect to the x-axis, i.e. can be described by dependencies:

$$D: \begin{cases} a \le x \le b, \\ g_2(x) \le y \le g_1(x), \end{cases} \tag{2}$$

hen the volume (1) can be described using (2), in the form:

$$\begin{aligned} V &= \int_a^b \left(\int_{g_2(x)}^{g_1(x)} \left(\int_{f_2(x,y)}^{f_1(x,y)} dz \right) dy \right) dx = \\ &= \int_a^b \left(\int_{g_2(x)}^{g_1(x)} (f_1(x, y) - f_2(x, y)) dy \right) dx. \end{aligned} \tag{3}$$

To describe such domain also can be used polar coordinates, cylindrical or spherical. Remembering the Jacobian determinant of given variables swap and describing right Ddomain (for flat surfaces) or Space (in case of describing solid, e.g. using spherical coordinates) we get appropriate, known to us equations for calculating volume of the solid with given coordinates. We want to focus on the case, in which area limiting given solid is defined by a function $\varrho(\varphi, \theta)$ this case any point of this surface is defined by three variables. The first of them is the distance ϱ from this point to the beginning of the coordinate system. The second is an angle φ between positive Ox is and a view, on a Oxy face of a segment joining the beginning of a coordinate system and given point. The third one is an angle θ between positive Oz is and a segment, joining the beginning of a coordinate system and given point (view Fig. 2). If for certain angles φ and for certain angles θ e define non-negative function $\varrho(\varphi, \theta)$ then all points received this way will define a certain surface in R^3 space.

It turns out that then the volume of a solid defined this way can e described by integral [2]:

ICYRIME 2021 @ International Conference of Yearly Reports on Informatics Mathematics and Engineering, online, July 9, 2021
kamil.czapla@alorybnik.polsl.pl (K. Czapla);
mariusz.pleszczynski@polsl.pl (M. Pleszczyński)

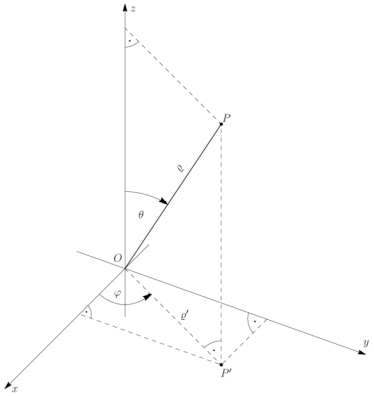

Figure 1: Spherical coordinates

$$V = \frac{1}{3}\int_{\varphi_1}^{\varphi_2}\left(\int_{\theta_1}^{\theta_2}\varrho^3(\varphi,\theta)\sin\theta\, d\theta\right)d\varphi. \tag{4}$$

In this connection that the form of the function ϱ can lead to integrals that do not express themselves with elementary functions, thereby the values of them cannot be determined precisely, therefore we will use some numeric methods to calculate approximate volume.

3. Monte Carlo methods

Monte Carlo methods appeared (most frequently this is assumed, although the theory describing these methods appeared the most frequently) in 1949, after the publication of the work [3], and allowed their creators to solve a computationally complex problem, that emerged during the work on the construction of the atomic bomb. The Monte Carlo methods family provides an approximate solution to a wide class of computationally complex problems, including numerical integration. We will focus on this group of problems. The computational complexity and error estimation obtained by this method show [4] that as the dimension increases, Monte Carlo methods become more and more useful in relation to classical methods (e.g. to quadrature based methods). We will deal with the double integral, but the ideas of the Monte Carlo methods, for the sake of clarity, will be shown on the example of the integral.

3.1. Monte Carlo methods no. 1

In classical terms, the domain between the graphs of the functions f(x) and g(x)we assume that $f(x) \geqslant g(x)$ for $b \geqslant x \geqslant a$ can be approximated as follows: enter the first graphs of the functions f and g into the rectangle R:

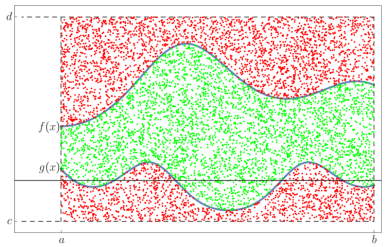

Figure 2: Monte Carlo method illustration

$$R = \left\{(x,y) \in \mathbb{R}^2 \ : \ a \leq x \leq b,\ c \leq y \leq d\right\}, \tag{5}$$

where c $\geqslant$ min g(x), d $\leq$ max f(x) The next step is to draw n points belonging to the rectangle R. If this choice is random (from uniform distribution) and independent, then these points should cover the rectangle R evenly. Therefore, the ratio of the number of m points lying between the graphs of the functions f and g to the number of all points n is the same as the ratio of the fields: S– between the graphs of the functions f and g |R| – area of the rectangle R We have:

$$\begin{aligned}\frac{S}{|R|} \approx \frac{m}{n} \Longrightarrow \int_a^b (f(x)-g(x))dx \\ = S \approx \frac{m}{n}|R| = \frac{m}{n}(b-a)(d-c).\end{aligned} \tag{6}$$

In Fig. 2 we present these dependencies graphically, with green color marking the points that went to the area of interest, and red color those that did not go to this area.

3.1.1. Example 1

We will show how it is possible to use this method to find the approximate value of the number e. For this we consider the integral:

$$\int_0^1 (e^x+1)dx = (e^x+x)\Big|_0^1 = (e+1)-(1+0) = e. \tag{7}$$

Knowing its exact value, we can estimate the value of the number e rising the formula (4). Let's take a rectangle:

$$R = \{(x,y) \in \mathbb{R}^2 \ : \ 0 \leq x \leq 1,\ x \leq y \leq 4\} \tag{8}$$

and let's conduct a series of experiments involving the drawing and verification of n points on the rectangle R For each of the n values presented in Table 1, we

Table 1
Monte Carlo method no. I

n	50	100	500	10^3	10^4
min	2.08	2.40	2.56	2.524	2.68
max	3.36	3.04	3.024	2.868	2.787
$\bar{S}$	2.659	2.68	2.728	2.741	2.722
σ	0.288	0.179	0.1004	0.068	0.021
Δ	0.0591	0.037	0.0097	0.022	0.003

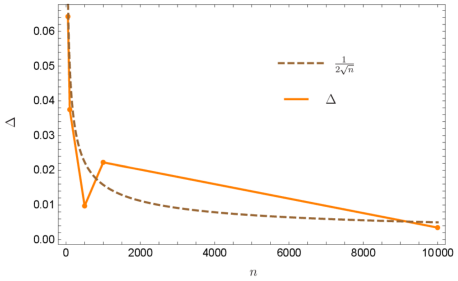

Figure 3: Monte Carlo method no.I

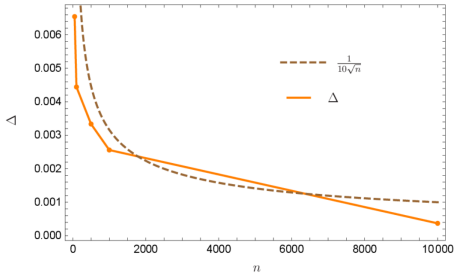

Figure 4: Monte Carlo method no.I – dependence of the Δ error on the number of ndraws

Table 2
Monte Carlo method no.II

n	50	100	500	10^3	10^4
min	2.577	2.604	2.655	2.666	2.709
max	2.894	2.859	2.775	2.753	2.729
$\bar{S}$	2.725	2.714	2.715	2.716	2.719
σ	0.0689	0.0496	0.0247	0.0179	0.0047
Δ	0.0065	0.0044	0.0033	0.0026	0.0004

conducted 50 draws (we want to check the stability of the method) and determined the following values: min – smallest area value, max – largest area value, S- average area value (as the arithmetic mean of the 50 results obtained), σ – standard deviation of the obtained results, Δ – calculation error in the formula $\Delta = |S - \bar{S}|$ the area S is the exact value of the area, and thus the number |e| and $\bar{S}$ s the obtained value.

The conducted experiments show that this method is stable and gives a good approximation. The Monte Carlo error theory says that the error obtained is proportional to the number $\frac{c}{\sqrt{n}}$, where $c < 0$ is a constant and n s the number of points drawn. Figure 3 shows that this relationship is preserved (in this figure, 3 has the value 0.5)

3.2. Monte Carlo methods no. 2

In this method, we have a slightly different approach to determining the value of the integral previously approximated by the formula (4). This time we refer to the definition of the Riemann definite integral:

$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(\xi_i)\Delta x_i, \tag{9}$$

where $\xi_i, i = 1, 2, ...n$ are any points up to the subinterval $[x_i, x_{i+1}]$,$i = 0, 1, \ldots n$ of the interval $[a, b]$ These sub-intervals are determined by a certain natural division, Δx_i is the length of such a sub-interval. If we run ndraws again (the assumptions for the drawing are the same as before), we can assume that the interval $[a, b]$ is divided evenly, that is $\Delta x_i = \Delta x = \frac{b-a}{n}$. Thanks to this, formula (5) will take the form:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^{n} f(\xi_i), \tag{10}$$

where ξ_i, i 1, 2, … n are the drawn points of the interval [a, b. On this basis, the field described by formula (4) will take the form:

$$\int_a^b (f(x) - g(x))dx = S \approx \frac{b-a}{n} \sum_{i=1}^{n} (f(\xi_i) - g(\xi_i)). \tag{11}$$

3.2.1. Example

Now let's do the same experiment as in Example 3.2.1. Table 2 shows the results of these experiments, and Fig. 4 we present the equivalent of Fig. 3, where the constant c as the value 0.1.

Intuitively, it is difficult to estimate which of the formulas (4) or (6) is represented by the greater error. It turns out that the formula (6) is more useful, and the results summarized in Tables 1 and 2 confirm this theory

3.3. Other Monte Carlo methods

In addition to the many advantages of Monte Carlo methods, such as simplicity of implementation, regardless of the dimension of the task, or the non-growing of errors

as the dimension increases, there are also disadvantages of this method. The most frequently mentioned is the value of the obtained errors, which is proportional to $\frac{1}{\sqrt{n}}$, where nis the number of points drawn. To increase the error by one row, e.g. from 1/10 to 1/100 , the number of points drawn should be increased from 100 to 104 . The methods discussed by us in this part of the work want to counteract this problem. There are many different approaches to such a task, we will cover the three most commonly used.

3.3.1. Separating of the main part

This method consists in finding a certain function which is close to the integral function, but for which we can find the exact value of the integral. To refer to our example of finding the approximate value of e where we integrate the function $f(x = e^x + 1)$, we can use the well-known Maclaurin series expansion of the function e^x

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + ... \tag{12}$$

Also note that:

$$\int_0^1 g(x)dx = 2x + \frac{1}{2}x^2 + \frac{1}{6}x^3\Big|_0^1 = 2 + \frac{1}{2} + \frac{1}{6} = \frac{8}{3} \tag{13}$$

Since the difference between the functions f(x and g(x) is determined, so:

$$\int_a^b f(x)dx = \int_a^b (f(x) - g(x))dx + \int_a^b g(x)dx \tag{14}$$

and in the discussed example, this formula takes the form:

$$\int_0^1 (e^x + 1)dx = \int_0^1 (e^x - 1 - x - \frac{1}{2}x^2)dx + \int_0^1 g(x)dx \tag{15}$$

which leads to the formula for the approximate value of this integral:

$$\begin{aligned}\int_a^b f(x)dx &\approx \frac{b-a}{n}\sum_{i=1}^{n}(f(\xi_i) - g(\xi_i)) + \int_a^b g(x)dx \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(e^{\xi_i} - 1 - \xi_i - \frac{1}{2}\xi_i^2\right) + \frac{8}{3},\end{aligned} \tag{16}$$

where $\xi = 1, 2, \dots n$ are random interval points $[a, b]$ from uniform distribution.

3.3.2. Importance sampling

This method [5] is based on the assumption that more points should be drawn in more significant parts of the interval $[a, b]$. To do this, one needs to find a function $p(x)$ which in the interval$[a, b]$ is the probability density, i.e. $p(x) > 0$

Of course, there are infinitely many such functions, and its choice has an impact on the resulting error. Most often, choosing the best such function is impossible, because it is related to the value of the integral (which is unknown). In practice, the approach is to select the function $p(x)$in such a way that, similar to the previous example, it is close to the function $f(x)$ i.e. that $\frac{p(x)}{f(x)} \approx c$ where c is a constant. After finding such a function, we can use the formula:

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)}p(x)dx \tag{17}$$

which leads to the formula for the approximate value of this integral:

$$\int_a^b f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n}\frac{f(\xi_i)}{p(\xi_i)} \tag{18}$$

where $\xi = 1, 2, \dots n$ are random interval points $[a, b]$ In the discussed example, using the previously found function $g(x) = 2 + x + 0.5x^2$

we can take: $p(x) = \frac{3}{8}g(x)$

$$p(x) > 0 \wedge \int_0^1 p(x)dx = \frac{3}{8}\int_0^1 g(x)dx = 1 \tag{19}$$

So we will get the following formula which allows us to find the approximate value of the number e

$$e = \int_0^1 f(x)dx \approx \frac{8}{3n}\sum_{i=1}^{n}\frac{f(\xi_i)}{2 + \xi_i + \frac{1}{2}\xi_i^2} \tag{20}$$

where $\xi = 1, 2, \dots n$ are random distribution $p(x)$

(You can read about the drawing of points from a given distribution in e.g. [6, 7, 8]).

3.3.3. Stratified sampling

This approach [9] is similar to the previous one. This time we divide the integration interval [a,b into k subintervals and randomize n_k points in each of them. The numbersn_k e chosen so that their sum is n and that their size is proportional to the rate of change of the integral value of the function $f(x)$ Based on this approach, the sought integral can be approximated by the formula:

Table 3
Separation of the main part

n	50	100	500	10^3	10^4
min	2.706	2.707	2.712	2.714	2.717
max	2.734	2.737	2.724	2.722	2.719
$\bar{S}$	2.72	2.719	2.718	2.718	2.718
σ	0.0073	0.0066	0.0027	0.0019	0.0005
Δ	0.0021	0.001	$5E-5$	$2E-5$	$2E-5$

Table 4
Importance sampling

n	50	100	500	10^3	10^4
min	2.705	2.704	2.712	2.717	2.717
max	2.739	2.73	2.723	2.722	2.72
$\bar{S}$	2.72	2.72	2.718	2.719	2.718
σ	0.0069	0.0054	0.0023	0.0013	0.0006
Δ	0.0017	0.0021	0.0002	0.0005	0.00006

$$\int_a^b f(x)dx \approx \sum_{j=1}^{k} \frac{d_j}{n_j} \sum_{i=1}^{n_j} f\left(\xi_i^{(j)}\right) \tag{21}$$

where $\xi_i^{(j)}$ is the i-th number drawn from the j-th subinterval, and d_j is the length of the jth sub-interval.

If the interval [0, 1] we will divide it into 5 equal parts and choose the next sub-ranges accordingly 1/25 , 3/25 , 5/25 , 7/25 and 9/25 the number n of all drawn points, then the formula (9) adjusted to approximate the value of the number e will take the form:

$$\begin{aligned} e = \int_0^1 (e^x + 1)dx \approx \frac{5}{n} \sum_{i=1}^{n/25} \left(e^{\xi_i^{(1)}} + 1\right) \\ + \frac{5}{3n} \sum_{i=1}^{3n/25} \left(e^{\xi_i^{(2)}} + 1\right) + \frac{1}{n} \sum_{i=1}^{n/5} \left(e^{\xi_i^{(3)}} + 1\right) \\ + \frac{5}{7n} \sum_{i=1}^{7n/25} \left(e^{\xi_i^{(4)}} + 1\right) + \frac{5}{9n} \sum_{i=1}^{9n/25} \left(e^{\xi_i^{(5)}} + 1\right) \end{aligned} \tag{22}$$

3.3.4. Result for other Monte Carlo methods

As for the previous methods, for the same values of n we carry out the process of approximating the value of the number e using the formulas (7), (8) and (9). In tables 3, 4 and 5 we will collect the appropriate results and in drawing 5 we illustrate the absolute error Δ similarly as we did in the drawings 3 and 4.

The conducted research confirms that, compared to the best known Monte Carlo method I, all other methods are more effective. For the same number n of drawn points, they give a better (statistically) approximation of the value of e

Table 5
Stratified sampling

n	50	100	500	10^3	10^4
min	2.695	2.695	2.708	2.711	2.716
max	2.753	2.744	2.729	2.728	2.72
$\bar{S}$	2.72	2.718	2.718	2.719	2.718
σ	0.0128	0.0114	0.0042	0.0037	0.0009
Δ	0.0022	$4E-6$	0.0002	0.0008	0.0002

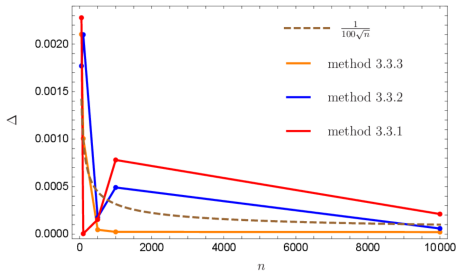

Figure 5: Comparison of other Monte Carlo methods

4. Monte Carlo methods for double integrals

The advantages of the Monte Carlo method, highlighted by us previously, are perfect for double integrals, both the simplicity of implementation (for both classical methods and their improvements) and the estimated error are transferred to double integrals (in general, to multiple integrals). As we mentioned in the introduction, we want to use Monte Carlo methods for integrals (3). This integral is formed when the surface bounding a given spatial domain is given by the function$\varrho(\varphi, \theta)$ [2] authors discussed the formula (3) and applied it to several surfaces, incl. peanut, bluebell, shell and exotic fruit. In this paper, we will use three approaches to the problem of determining these integrals using the Monte Carlo method I, the Monte Carlo method II and parallel calculations used in the Monte Carlo method II, the aim of which is to shorten the time needed to determine a given integral. Moreover, as some integrals (3) are reduced to a single integral in the above-mentioned surfaces, we will consider domains both on the plane and in space. Parallel computations will be performed using a multithreaded process, and the aaabbbccc process itself is very simple. As the points drawn are independent and do not affect each other, as well as the calculations of the function values in these points, it is enough to divide the

number of draws n into k equal parts and each thread will perform its calculations in parallel. After making the calculations by all threads, the results obtained are summed up to the final result. Note that the formula (6) can be implemented in parallel – the number n of all points is divided into k equal parts, where k is the number of threads. Each thread will do its own sum:

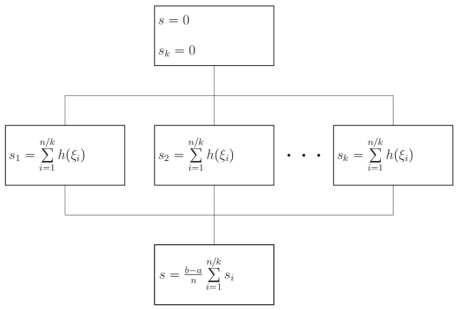

Figure 6: Comparison of other Monte Carlo methods

$$s_k = \sum_{i=1}^{n/k} (f(\xi_i) - g(\xi_i)) = \sum_{i=1}^{n/k} h(\xi_i). \tag{23}$$

where $h(x) = f(x) - g(x)$ are random points from the interval $[a, b]$ And then we will create the final sum:

$$S = \frac{b-a}{n} \sum_{i=1}^{k} s_i \tag{24}$$

This process is shown in Fig. 6.

The calculations were carried out on a computer with a 12-thread Intel Core i7-3930k CPU 3.20 GHz processor, 16 Gb RAM and a 64 bit Windows 7 system. Programs that implement all Monte Carlo methods were written in Mathematica 12.2 (including those for parallel computing) [10]. We will now present the application of the proposed Monte Carlo methods on the example of two surfaces introduced and discussed in [2] – bell and exotic flower

4.1. Bell Area

This area is represented by the relationship: $\varrho(\varphi, \theta) = a + sin(2\theta) + cos(3\theta)$ and its graph, for a= 2 is shown in Fig. 7.

As we showed in [2] the volume of the solid limited by this area is:

$$V = \frac{2a\pi}{3}\left(\frac{214}{35} + 2a^2 - \frac{3\pi}{2}\right) \tag{25}$$

which for a= 2 gives the value 39.382572945...

4.1.1. Monte Carlo method no. I

According to the idea of this method, the surface of the bell will be entered in a cuboid C

$$C = \{(x, y, z) \in \mathbb{R}^3 : |x| \le 2.2,\ |y| \le 2.2,\ -1.5 \le z \le 3.5\} \tag{26}$$

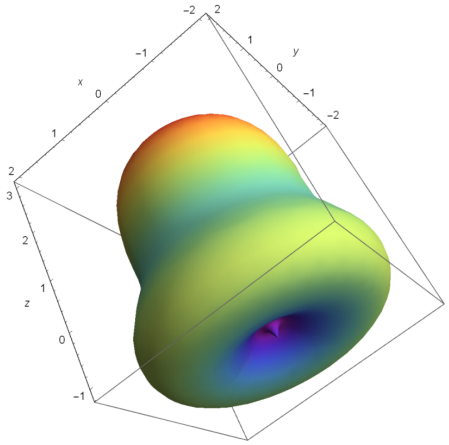

Figure 7: Bell area

This time, the equivalent of formula (4) should be a formula adapted to the size, so the volume V of the sought solid would be approximate by the formula:

$$V \approx \frac{m}{n} \cdot |C| = \frac{m}{n}(2.2 - (-2.2))^2(3.5 - (-1.5)) = \frac{5m}{n}(4.4)^2, \tag{27}$$

where, like before, m is the number of points that hit the inside of the bell, and n is the number of all drawn points of the cuboid C However, using the spaces performed in the work [2] the volume of the bell is described by the single integral:

$$V = \frac{2\pi}{3}\int_0^{\pi} (2 + \sin 2\theta + \cos 3\theta)^3 \sin\theta d\theta \tag{28}$$

So we have a situation where (without taking into account the unit) the volume of the bell equals (numerically) the area under the curve:

$$f(x) = \frac{2\pi}{3}(2 + \sin 2x + \cos 3x)^3 \sin x \tag{29}$$

for x in $[0, \pi]$, which means that it is enough to draw n points of the rectangle R.

$$R = \{(x, y) \in \mathbb{R}^2 :\ 0 \le x \le \pi,\ 0 \le y \le 12\} \tag{30}$$

(the number 12 can be determined on the basis in Fig. 8 which shows the graph of the function $f(x)$ along with the drawn $n = 10^3$ points) and use the formula (4).

After conducting the experiment consisting in drawing $n = 9!$ points of the rectangle R e obtained the following approximate result of the bell volume value: $V = 39.3611$ what the error corresponds to $\Delta = 0.0214$ This result, which will be important when comparing the

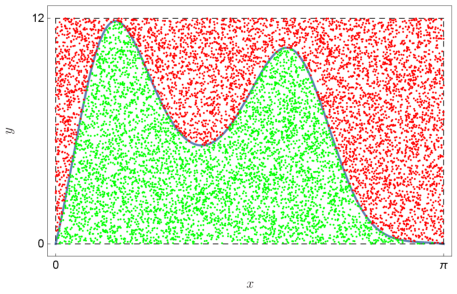

Figure 8: Area under the curve f(x) corresponding to the volume of the bell

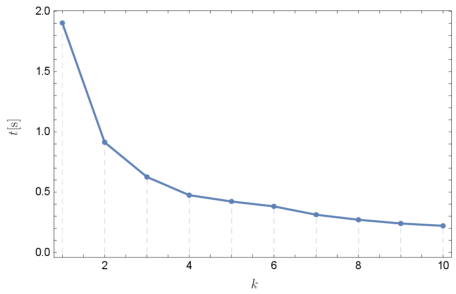

Figure 9: Dependence of t time on the number of k threads

Table 6
Monte Carlo method no.I for the bell, $n = 9!$

min	max	$\overline{S}$
39.2254	39.5313	39.3867
σ	Δ	$\overline{t}$
0.0728889	0.004098	3.18523

Table 7
Monte Carlo method no.II for the bell, $n = 9!$

min	max	$\overline{S}$
39.2835	39.504	39.3789
σ	Δ	$\overline{t}$
0.042718	0.003628	1.8848

methods, was achieved in $t = 3.19802$ seconds. Proceeding similarly to chapter 3 we then conducted a series of 50 such experiments, and their results (designations analogous to the previous ones with an additional value of the calculation time $\overline{t}$ are presented in Table 6.

4.1.2. Monte Carlo method no.II

As before, we will go straight to the calculation of the area under the curve (10) and use the formula (6). Therefore, we will select $n = 9!$ points from the interval $[0, \pi]$and a searched area (and thus the volume of the bell) we will determine approximately. This time we got an area (volume) equal to 39.3762, which corresponds to error $\Delta = 0.0064$. We obtained this result in $t = 1.903$ seconds. After conducting 50 such experiments, we obtained the results summarized in Tab. 7.

4.1.3. Parallel Monte Carlo method no.II

Monte Carlo methods are perfect for parallelization. We can do it because the independent draws of n points from the interval $[a, b]$ is equivalent to k draws of n/k points from this interval (we sum the k set obtained in this way, obtaining one n-elementary set). Let us now carry out experiments analogous to the above experiments, except

Table 8
Parallel Monte Carlo method no.II for the bell, $n = 9!$

k	$\overline{t}$	min	max	$\overline{S}$	σ	Δ
2	0.91385	39.223	39.5096	39.3768	0.05743	0.00581
3	0.62608	39.2124	39.5729	39.3856	0.06559	0.00298
4	0.47479	39.1406	39.6401	39.3858	0.0848	0.00325
5	0.4222	39.137	39.696	39.3933	0.09703	0.0107
6	0.38184	39.1023	39.6581	39.3914	0.10243	0.00883
7	0.31249	39.0822	39.6525	39.3837	0.11084	0.00109
8	0.2707	39.064	39.7337	39.3778	0.12086	0.00481
9	0.2399	39.0196	39.7041	39.3794	0.1233	0.00315
10	0.22059	38.9041	39.7851	39.3768	0.13131	0.00575

that we will use parallel computations with the use of $k = 2, 3, ..., 10$ threads. The results of these experiments are collected in Table 8, we will pay special attention to the average time $\overline{t}$ – its change is to prove the usefulness of parallel computations. As indicated by the data in table 8, the introduction of calculations, on the one hand does not worsen the obtained volume results, and on the other hand reduces the time needed to obtain this result. The comparison of the dependence of this time t to the number of threads k is shown in Fig. 9.

4.2. Exotic fruit surface

This area will be represented by the relationship:

$$\varrho(\varphi, \theta) = a + \cos 2\theta \sin 5\varphi \tag{31}$$

$0 \leq \varphi \leq 2\pi, 0 \leq \theta \leq \varphi$ and $a \geq 1$, and its graph for a= 1 is shown in Fig. 10.

As we showed in [2], the volume of a solid limited by this area is:

$$\begin{aligned} V &= \frac{1}{3}\int_0^{2\pi}\left(\int_0^{\pi}(a + \cos 2\theta \sin 5\varphi)^3 \sin\theta \, d\theta\right) d\varphi \\ &= \frac{2\pi a}{15}(7 + 10a^2) \end{aligned} \tag{32}$$

what for a=1 gives value of 7.120943348.

4.2.1. Monte Carlo method no. I

This time we will refer to the formula (4) which says that the surface of an exotic fruit is limited by an area whose

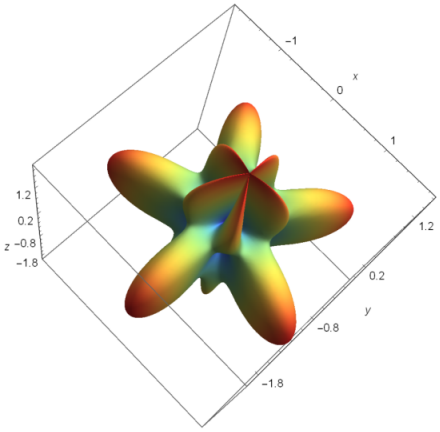

Figure 10: Exotic fruit surface

volume corresponds to the volume of the area limited from above by the area $f(x, y)$ above the rectangle

$$R = \{(x, y) \in \mathbb{R}^2 : 0 \le x \le 2\pi,\ 0 \le y \le \pi\} \tag{33}$$

where

$$f(x, y) = \frac{1}{3}(1 + \cos 2y \sin 5x)^3 \sin y \tag{34}$$

which means that it is enough to draw n points of the cuboid C

$$C = \left\{(x, y, z) \in \mathbb{R}^3_+ : x \le 2\pi,\ y \le \pi,\ z \le \frac{8}{3}\right\} \tag{35}$$

(the number 8/3 can be determined on the basis in Fig. 11, which shows the graph of the function $f(x, y)$ with the drawn $n = 10^4$ points) and use the formula (6), which this time will have to change by one form dimension:

$$V \approx \frac{m}{n}|C| = \frac{m}{n} \cdot 2\pi \cdot \pi \cdot \frac{8}{3} = \frac{16m\pi^2}{3n} \tag{36}$$

where m is the number of points that hit the inside of the fruit (and thus are in the cuboid C below the surface $f(x, y)$)

After conducting a series of 50 experiments described above, assuming n= 9!, we obtained the results presented in table Tab. 9.

4.2.2. Monte Carlo method no. II

Proceeding as before, i.e. drawing n= 9! points and using the formula (11), after adjusting it to the dimension of the task:

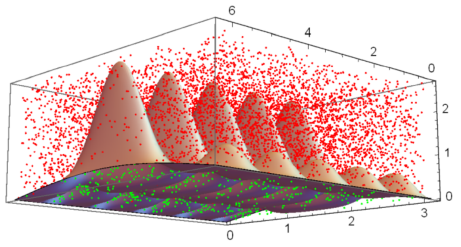

Figure 11: Exotic fruit surface. The volume limited by the area of $f(x, y)$ corresponding to the volume of an exotic fruit

Table 9
Monte Carlo method no.I for fruit, $n = 9!$

min	max	$\bar{S}$
7.04261	7.18854	7.12308
σ	Δ	$\bar{t}$
0.030614	0.00214	3.77585

Table 10
Monte Carlo method no.II for fruit, $n = 9!$

min	max	$\bar{S}$
7.09186	7.15213	7.12213
σ	Δ	$\bar{t}$
0.01635	0.00119	3.37867

$$\int_a^b \left(\int_c^d f(x, y)dy\right)dx = V \approx \frac{(d-c)(b-a)}{n}\sum_{i=1}^{n} f(p_i), \tag{37}$$

where $\rho_i, i = 1, 2, \ldots n$ are the points of the rectangle on which the integration takes place, we will get the results (we have carried out 50 such experiments again) which are summarized in Tab. 10.

4.2.3. Parallel Monte Carlo method

In this method we will act in a similar way to what we did in subsection 4.1.3. Again we will draw n parallel points, using k threads, each of which will draw n/k points. As before, we will use $k = 2, 3, \ldots, 10$ threads, and the results of these experiments are presented in Tab. 11. Again, it turned out that multi-threading does not deteriorate the quality of the solution, while reducing the computation time. A graphical interpretation of the

dependence of this time on the number of threads used is shown in Fig. 12.

5. Conclusions

The authors in [2] introduced a certain formula for determining the volume of solids. They also calculated the volume of the selected solids.Since the formula is non-standard (in this work it is formula (4)) the authors decided to verify it. for this purpose, we used selected Monte Carlo methods that are well suited for this purpose. We discussed 5 such methods, and to verify the formula (4) we used 2 of them and an additional sixth method, which is a parallel version of the second method. The conducted research has shown that Monte Carlo methods are effective in this type of tasks, that their modifications are more useful, that formula (4) is correct and that the disadvantage of Monte Carlo methods can be effectively dealt with. The introduced parallel method does not have a negative impact on the quality, and at the same time shortens the calculation time.

References

[1] K. Czapla, M. Pleszczyński, Some formulas for determining areas and volumes, part I: areas in space R^2, MINUT 3 (2021) 208–228. In Polish

[2] K. Czapla, M. Pleszczyński, Some formulas for determining areas and volumes, part II: areas in space R^3, MINUT 3 (2021) 229–244. In Polish.

[3] N. Metropolis, S. Ulam, The monte carlo method, Journal of the American Statistical Association 44 (1949) 335–341.

[4] N. P. Buslenko, D. I. Golenko, I. M. Sobol, W. G. Sragowicz, J. A. Szrejder, The Monte Carlo Method, PWN, Warsaw, 1967. In Polish.

[5] Capizzi G., Sciuto G.L., Napoli C., Shikler R., Wozniak M., Optimizing the organic solar cell manufacturing process by means of AFM measurements and neural networks (2018) Energies, 11 (5), doi: 10.3390/en11051221.

[6] Kloek, T., and Van Dijk, H. K. (1978). Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. Econometrica: Journal of the Econometric Society, 1-19.

[7] Devroye, Luc. "Sample-based non-uniform random variate generation." Proceedings of the 18th conference on Winter simulation. 1986.

[8] Wróbel M., Starczewski J.T., Napoli C., Handwriting recognition with extraction of letter fragments (2017) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10246 LNAI, pp. 183 - 192, doi: 10.1007/978-3-319-59060-8_18.

[9] The stochastic collocation monte carlo sampler: Highly efficient sampling from 'expensive' distributions,Quantitative Finance, 19(2), 339-356.

[10] Połap D., Wózniak M., Napoli C., Tramontana E., Is Swarm Intelligence Able to Create Mazes? (2015) International Journal of Electronics and Telecommunications, 61 (4), pp. 305 - 310, doi: 10.1515/eletel-2015-0039.

[11] Połap D., Wózniak M., Borowik G., Napoli C., A First Attempt to Cloud-Based User Verification in Distributed System (2015) Proceedings - 2015 Asia-Pacific Conference on Computer-Aided System Engineering, APCASE 2015, art. no. 7287024, pp. 226 - 231, doi: 10.1109/APCASE.2015.47.

[12] Steinbrecher, G., and Shaw, W. T. (2008). Quantile mechanics. European journal of applied mathematics, 19(2), 87-112.

[13] Botev, Z., and Ridder, A. (2017). Variance reduction. Wiley statsRef: Statistics reference online, 1-6.

[14] Morrison, Michael, Cliff Hastings, and Kelvin Mischo. "Hands-on start to wolfram mathematica." (2015).