Computational Mathematics I - Project 3

Foivos Lambadaridis Kokkinakis

January 2024

Contents

Abstract/Objective

This assignment involves calculating the double integral of the function $f(x, y) = x^2 + y^3$ over a specified domain using three distinct methods: analytical (exact calculation), numerical (using standard numerical integration techniques), and Monte Carlo integration. In this project we will present an introduction of the theory needed to comprehend each method, the methodology of applying said method, the results each method gave us a discussion of said results, and finally the conclusions we reached during our study.

The integral is:

$$\boxed{I = \int_0^1 \int_x^{2x} \left(x^2 + y^3\right)\, dx\, dy} \tag{1}$$

A property of double integrals we will use to assess the validity of our results is that when $z = f(x, y) \geq 0$ is the integrated area D, then:

$$\int \int_D f(x, y)\, dx\, dy \geq 0 \tag{2}$$

Chapter 1

Introduction

1.1 Double integrals

Before calculating the double integral, we should understand what we are attempting to do. We see that the integrand we are is a function of x and y, which makes it useful if we set $z = f(x, y)$. Because of this, we can imagine $z = f(x, y)$ as a 3D curve [1]. Knowing that a single integral geometrically represents the area under a 2D curve, we can generalize and conclude that a double integral represents the volume under a curve $z = f(x, y)$, where $dA = dx \times dy$ is the elementary area under said curve [1]. The limits of integration will tell us the limits of x and y and will define an area on the x-y plane that we will call D. This area will be the side limits, while $z = f(x, y)$ will be the top or bottom limit of the volume we are trying to calculate.[1]

1.2 Numerical integration

This section will first mention numerical methods of calculating single integrals. From these, we will explain our choice of the numerical method and in the next section, we will generalize it and apply it to a double integral.

1.2.1 Single integrals

Similar to finding the root of an equation the calculation of an integral can't always be performed analytically, so one has to resort to numerical methods. These methods approximate the integrad $f(x)$ by interpolation. Firstly they break $f(x)$ in sections of n equally long intervals of length h depending on the limits of integration. Then each of these sections of $f(x)$ is approximated by a polynomial $P_m(x)$.[2] The reason behind using this tactic is that integrating polynomials is easy so we can get exact approximate formulas for calculating the integral. Here the approximation of $f(x)$ will be given by the Lagrange interpolation. Here we will briefly mention one method the trapezoidal method and the Simpsons rule and we will explain which one of the two we will end up applying and why.

Let's discuss our case, integral I of eq 1. We see that the limits of integration are numbers on one integral (0 and 1) and functions of x on the other ($y_1(x) = x$ and $y_2(x) = 2x$). These limits describe area D in the $x - y$ plain where y is in between the curves $y_1 = x$ and $y_2 = 2x$ and x is in between $x = 0$ and $x = 1$.

During the next part of this section, we will assume that we are trying to calculate the integral below using a step $h < 1$, that will remain constant throughout the application

of the methods.

$$I = \int_a^b f(x)\,dx \tag{1.1}$$

We will also use the denotation: $x_i = x_0 + i \times h$ and $f_i = f(x_i)$, where $a = x_0 \leq x_i \leq b = x_{n-1}$

Trapezoidal method

The trapezoidal method is characterized by approximating $f(x)$ between two neighboring points x_i and $x_{i+1} = x_n + h$ with a polynomial of 1st degree or a straight line [2],[3]. Graphically this method splits the area under $f(x)$ into n trapezoids. By adding the area of all these trapezoids, one approximates I. The formula that the trapezoidal method gives us for approximating I is [2]:

$$I = \int_{x_0}^{x_n} f(x)\,dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} P_1(x) = \frac{h}{2}\left(f_0 + 2f_1 + \ldots + 2f_{n-1} + f_n\right) \tag{1.2}$$

The error of this approximation is given by [2]:

$$E_{Trap} = -\frac{h^2}{12}(b-a)f''(\xi) \tag{1.3}$$

where $\xi \in (a, b)$. We notice from 1.3 that the error of our result will be proportional to h^2. We chose not to use the trapezoid method since the two other methods give more precise results when applied with the same step h.

Simpson's rule

Simpson's rule is similar to the trapezoidal method only this time $f(x)$ is approximated in 3-point-long sections by a 2nd-degree polynomial [3],[2]. This means that for the method to be applicable, the space $[a, b]$ must be divided in an even number of intervals n. Then the area under each one of these parabolas is added and one gets an approximation of I. The formula that Simpson's method gives us for approximating I is [3],[2]:

$$I = \int_{x_0}^{x_n} f(x)\,dx = \sum_{i=0}^{n-2} \int_{x_i}^{x_{i+2}} P_2(x) \Rightarrow$$

$$\boxed{I = \frac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 \ldots + 2f_{n-2} + 4f_{n-1} + f_n\right) = \frac{h}{3}\left(L_{end} + 2L_{even} + 4L_{odd}\right)} \tag{1.4}$$

where $L_{end}, L_{even}, L_{odd}$ are the sums of the terms on either end, of the even and odd terms respectfully. The error of this method is given by [3],[2]:

$$E_{Sim} = -\frac{b-a}{180}h^4 f^{(4)}(\xi) \tag{1.5}$$

where $\xi \in (x_0 = a, x_n = b)$. From 1.4 we notice that the error of Simpson's method will be proportional to h^4. Comparing Simpson's method's error with that of the trapezoids, we can expect the first to be a lot smaller than the latter for the same step h.

One important note is that due to this increase of two orders of h from the trapezoidal rule, Simpson's rule can calculate the integral of any 3rd-order polynomial or lower perfectly [3].

1.2.2 Simpson's double integration

In this section, we will attempt to apply Simpson's rule to the double integral:

$$I = \int_a^b \int_{y_1(x)}^{y_2(x)} f(x,y)\,dy\,dx = \int_a^b \left(\int_{y_1(x)}^{y_2(x)} f(x,y)\,dy \right) dx \tag{1.6}$$

To do so we will divide the x-plain of D: $x \in [a,b]$, in an even number n intervals each one of equal length: [2],[3]

$$h = \frac{b-a}{n} \tag{1.7}$$

as we did in the simple integral. However when we look at the y plain: $y \in [y_1(x), y_2(x)]$) we see that it's bounded by the two curves y_1 and y_2. This means that if we attempt to divide the y-plain into m intervals, those won't be equally spaced since each interval will be of length [3]:

$$k(x) = \frac{y_2(x) - y_1(x)}{m} \tag{1.8}$$

Our first step is to apply Simpson's rule to the most inward integral. Since the integral is concerning y, we will keep x constant and apply Simpson's rule from eq. 1.4 with step $k(x)$:

$$I = \int_a^b \frac{k(x)}{3} \left(f(x, y_1(x)) + 4f(x, y_1(x) + k(x)) + \ldots + 4f(x, y_1(x) + (m-1)k(x)) + f(x, y_2(x)) \right) dx \Rightarrow$$

$$\boxed{I = \int_a^b \left(L_{end}(x) + 2L_{even}(x) + 4L_{odd}(x) \right) dx = \int_a^b g(x)\,dx} \tag{1.9}$$

where L_{end}, L_{even}, L_{odd} are the sums of the functions $f(x, y_1(x) + ik(x))$ on either end, of the even and odd functions respectfully, each multiplied by $\frac{k(x)}{3}$ [3].

Now we can see that eq. 1.9, describes an integral like eq. 1.1, meaning, that we can apply the standard Simpson's rule with step h to it. In other words, the value of the double integral I is given by applying Simpson's rule in $g(x)$ of eq. 1.9:

$$\boxed{I_{Sim} = \int_a^b g(x)\,dx = \frac{h}{2} \left(J_{end} + 2J_{even} + 4J_{odd} \right)} \tag{1.10}$$

where J_{end}, J_{even} and J_{odd} is the sum of the end, even, and odd numerical terms that Simpson's rule produces due to the integrals: $\int_a^b L_{end}(x)\,dx$, $2\int_a^b L_{even}(x)\,dx$ and $4\int_a^b L_{odd}(x)\,dx$ [3].

A really important note is that for the calculation of the double integral I, in every 2D section of D: $(x,y) = ((a+(i-1)h, a+ih), (y_1(x)+(j-1)k(x), y_1(x)+jk(x)))$ where $0 \le i \le n$ and $0 \le j \le m$, one has to firstly calculate $g(x)$ (or $L_{end}(x), L_{even}(x), L_{odd}(x)$ from 1.9 and then reapply the method on them, to calculate the numerical value of I or the value of J_{end}, J_{even} and J_{odd} [3].

1.3 Monte Carlo integration

The Monte Carlo algorithm uses randomly generated numbers and it's used vastly, in a large number of numerical applications. Here we will attempt to calculate the double integral of eq. 1.6 with it. More specifically we will apply the dartboard Monte Carlo

integration, which simulates the calculation of the integral by throwing darts (generating points) and counting the number of points that are under the surface $z = f(x, y)$ [2]. While [2] talks about single integrals we will expand on its' ideas to perform a double integration.

Since we are calculating the double integral of $z = f(x, y)$, in reality, we are calculating the volume under $f(x, y)$ bounded by D, V_2. Firstly one has to define the volume we are interested in $V_1 = (x_{min}, x_{max}), (y_{min}, y_{max}), (z_{min}, z_{max})$, where v_{min} and v_{max} are calculated from the limits of D and $z = f(x, y)$. One has to note that in this model V_1 would be the volume of a cube with sides $x_{max} - x_{min}$, $y_{max} - y_{min}$ and $z_{max} - z_{min}$.

Then we generate N random values from a uniform distribution that are inside V_1 for x,y and z respectfully, basically creating N pairs of point inside V_1. These points, since they come from a uniform distribution should cover all of V_1 uniformly[3]. All of the above leads to the ability to find the number of points that are inside V_2, q, which we can divide by the number of points we generated N to get the statistical probability of finding any point we generated inside V_2. In other words:

$$\boxed{p = \lim_{N\to\infty} \frac{q}{N} \approx= \frac{q}{N} = \frac{V_2}{V_1}} \tag{1.11}$$

The later equality of eq. 1.11, becomes apparent if we consider that uniformity means that any point we generated has an equal probability to appear anywhere inside V_1 [2]. In other words, using eq. 1.11 we can say that the value of the double integral will be given by:

$$\boxed{I = V_2 = pV_1} \tag{1.12}$$

We expect that the bigger our sample is, the more precise our final result will be since the points will be more dense inside V_1 and thus we can make the analogy of our resolution increasing [2].

Chapter 2

Methodology

In this chapter we will discuss how we we applied everything mentioned in the first chapter to calculate the integral I analytically and numerically (using standard numerical integration techniques), and Monte Carlo integration.

2.1 Analytical solution

The first step of calculating said integral is to integrate $f(x, y) = x^2 + y^3$ with respect to y or in other words:

$$I = \int_{x=0}^{x=1} \left(\int_{y_1(x)=x}^{y_2(x)=2x} f(x,y)\, dy \right) dx = \int_{x=0}^{x=1} \left(\int_{y_1(x)=x}^{y_2(x)=2x} x^2 + y^3\, dy \right) dx$$
$$= \int_{x=0}^{x=1} \left(x^2 y + \frac{y^4}{4} \right) dx|_{y_1(x)=x}^{y_2(x)=2x} = \int_{x=0}^{x=1} \left(2x^3 + \frac{16x^4}{4} - x^3 - \frac{x^4}{4} \right) dx \Rightarrow$$
$$I = \int_{x=0}^{x=1} \left(x^3 + \frac{15x^4}{4} \right) dx = \int_{x=0}^{x=1} g(x)\, dx \tag{2.1}$$

Eq. 2.1, describes a single integral, so by applying the usual integration techniques we get:

$$\boxed{I_{an} = \frac{x^4}{4} + \frac{15x^5}{20}|_0^1 = 1} \tag{2.2}$$

where we have named the analytical solution of I as I_{an}. I_{an} is the exact value of I and it will be used to measure the precision of the results we will get from numerical methods.

2.2 Simpson's Rule

As described in the previous chapter, the first step we are going to take is to split $x \in [a, b]$ in n intervals, where n has to be an even number. This would mean we would have to iterate over x with a step h, that's given from eq. 1.7. The next step is to set $J_{end} = J_{even} = Jodd = 0$. These will be calculated later and we will get our final answer by applying 1.10

For each numerical value of $x_i = a + ih,\ 0 \leq i \leq n$ we have to first calculate $g(x)$ that's given in eq. 1.9. To do so we break the y plain up in an even number of intervals m. The step $k(x_i)$ is given by eq.1.8. Similar to x, for each numerical value of y given by $y_j = y_1(x_i) + k(x_i)m, 0 \leq j \leq m$, ($y_1(x)$ and $k(x_i)$ are numerical since x has taken the numerical value x_i) we have to calculate the quantity: $Q = f(x_i, y_j)$.

Now that we know the value of Q, we can move on to the calculation of $g(x_i)$. To do so we use eq. 1.9. We set $L_{end} = L_{even} = Lodd = 0$ and then add Q to each term according to the value of j: If $j = 0, m$ we add to L_{end}, if j is even we add to L_{even} and if j is odd we add to L_{odd}. This process is repeated for every $j \in [0, m]$.

Having the final values of L_{end}, L_{even} and L_{odd}, using 1.9 we calculate the value of $g(x_i)$. We add $g(x_i)$ to our starting variables J_{end}, J_{even} and J_{odd} according to the value of i: If $i = 0, n$ we add to J_{end}, if i is even we add to J_{even} and if i is odd we add to J_{odd}. This process is repeated for every $i \in [0, n]$ In the end we will have the final values of J_{end}, J_{even}, and J_{odd} so by applying eq. 1.10 we will get the numerical approximation of I that Simpson's rule gives us.

To implement Simpson's rule we have created a Python function that takes as input:

1. The integral $f(x, y)$
2. The lower a and upper b limits of integration with respect to x
3. A list containing the intervals that x will be broken up into at each iteration
4. The lower $y1(x)$ and upper $y2(x)$ limits of integration with respect to y
5. A list containing the intervals that y will be broken up into at each iteration

and that will follow all of the above steps for every combination of n and m that are on n_list and m_list respectfully. The final result we will get is a matrix I_{sim} whose elements will be the numerical values we will get by applying Simpson's rule to I for all the combinations of n and m.

2.3 Monte Carlo

As mentioned before to apply the Monte Carlo method over has to define the volume V_1 we are interested in meaning the limits in all three dimensions x, y, and z. In our problem ,we can see that $x \in [0, 1]$, $y \in [0, 2]$ and $z = \in [0, 9]$ since the max value of y is given by $y_2(x_{max})$ and the max value of z is given by $z = f(x_{max}, y_{max})$ since f is an increasing function. From the above we can see that:

$$V_1 = x_{max} y_{max} z_{max} = 18 \tag{2.3}$$

Our next task is to generate uniform pairs of 3D points $p = (x, y, z)$ inside V_1. In our application the number of points N_i we will generate isn't constant, but:

$$N_i = 10^i \, i \in [1, 8] \tag{2.4}$$

For every i we will get another numerical approximation of I. We must also calculate the value of $z_p = f(x, y)$ at every point.

Now we are ready to approximate the volume V_2 bounded by $f(x, y)$ and D. If call q the number of points p that are inside V_2 then:

$$q_i = \sum p = (x, x \leq y \leq 2x, z \leq f(x, y)) \tag{2.5}$$

Since we have calculated q_i by applying eq.2.5 we calculate p_i and then using 1.12 we get the numerical approximation of V_2 or I for every N_i.

Chapter 3

Results and Discussion

In this chapter, we will present the results we got from the implementation of Simpson's Method and the Monte Carlo integration, and comment on them.

3.1 Simpson's Rule

As stated before, when applying Simpson's rule, we calculated integral I a total of 25 times, each one for taking a different number of intervals in the x and y direction.

	4	8	50	100	200
4	1.001953125	1.001953125	1.001953125	1.001953125	1.001953125
8	1.0001220703125	1.0001220703125	1.0001220703125	1.0001220703125	1.0001220703125
50	1.00000008	1.00000008	1.00000008	1.00000008	1.00000008
100	1.000000005	1.000000005	1.000000005	1.000000005	1.000000005
200	1.0000000003125	1.0000000003125	1.0000000003125	1.0000000003125	1.0000000003125

Table 3.1: The numerical results we got from applying Simpson's rule to $I = \int_0^1 \int_x^{2x} \left(x^2 + y^3\right)\, dx\, dy$, with a different number of in the x and y directions, m and n respectably. The first row represents the number of intervals in the x direction, while the first column represents the number of intervals in the y direction.

As we can see from table 3.1, the numerical result we get is not affected when the number of intervals in the y direction changes, meaning that no matter how big or small the step $k(x)$ is, our results will be unaffected! As noted previously Simpson's method can derive exact results for polynomials of the 3rd degree and under. We can see that concerning y $f(x, y) = x^2 + y^3$ is indeed a 3rd-degree polynomial, thus when we integrate concerning y, using step $k(x_i)$, the result we get is precisely $g(x_i)$, where $g(x)$ is defined in eq. 2.1.

The difference shows itself when we apply Simpson's rule for the second time (concerning x). If we focus on eq. 2.1, we can see that in reality we are integrating over a fourth-order polynomial, the solution is not exact and thus it is affected by the value of step h or the number of intervals n.

This makes the reason for choosing Simpson's rule more clear: The error of the numerical result the Simpson's rule will give us will only be a function of h, in other words, it reminds us of the single integral problem. Using eq. 1.5, we expect the error of a numerical

result calculated with step $h = \frac{1-0}{n}$ in our problem to be:

$$E_h = -\frac{1-0}{180}h^4 g^{(4)}(\xi) = -\frac{1}{180}h^4 90 = -\frac{h^4}{2} \tag{3.1}$$

We can calculate the true error E_t of every numerical result we got by finding the absolute value of the difference between the values of table 3.1 and the real value of I, I_{an} that's given by eq. 2.2. If we do that we get:

	4	8	50	100	200
4	1.953125×10^{-3}	1.953125×10^{-3}	1.953125×10^{-3}	1.953125×10^{-3}	1.953125×10^{-3}
8	1.220703×10^{-4}	1.220703×10^{-4}	1.220703×10^{-4}	1.220703×10^{-4} 5	1.220703×10^{-4}
50	8×10^{-8}	8×10^{-8}	8×10^{-8}	8×10^{-8}	8×10^{-8}
100	5×10^{-09}	4.999999×10^{-9}	4.999999×10^{-9}	4.999999×10^{-9}	5×10^{-09}
200	3.125002×10^{-10}	3.125002×10^{-10}	3.125002×10^{-10}	3.125002×10^{-10}	3.125002×10^{-10}

Table 3.2: The absolute value of the total error we got from applying Simpson's rule to $I = \int_0^1 \int_x^{2x} \left(x^2 + y^3\right) dx\, dy$, with a different number of in the x and y directions, m and n respectably. The first row represents the number of intervals in the x direction, while the first column represents the number of intervals in the y direction.

From table 3.2, we can see that the results we got from the same values of m don't always have exactly equal values. This is due to the limited way the computer stores our results or the round-off error due to our calculations.

Finally, we will check if E_h of eq. 3.1 describes the errors of table 3.2. To do so we have created the plot below:

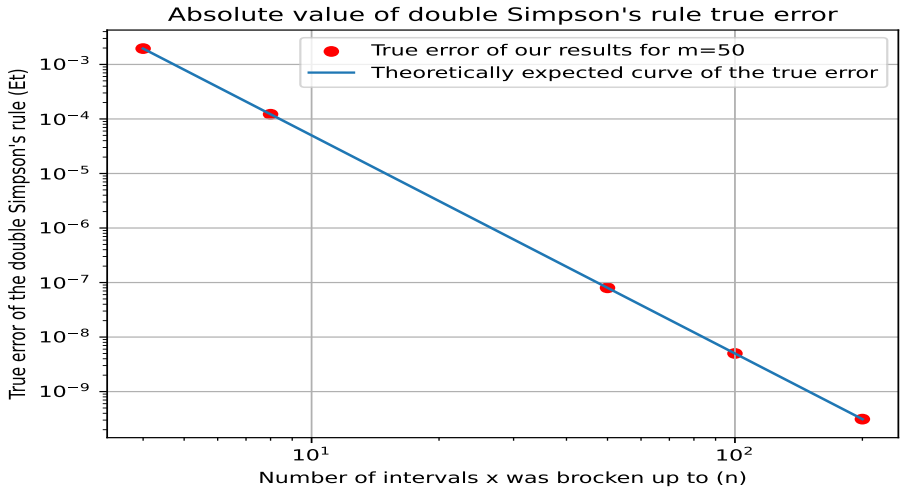

Figure 3.1: The blue curve represents the absolute value of E_h and it describes the expected total error of the numerical result one would get if they tried to calculate the integral $I = \int_0^1 \int_x^{2x} \left(x^2 + y^3\right) dx\, dy$ by splitting the x axis into n intervals by applying Simpson's rule. The red points represent the numerical values of the total error we got from calculating I with different values of n.

From figure 3.1, we can see that our numerical results agree with the theoretical framework we described earlier, making our choice of using Simpson's rule to numerically calculate I, the correct one.

3.2 Monte Carlo

As stated above we have created three-dimensional points (x, y, z) inside of V_1. The number of points we have created is given by eq. 2.4, meaning we will calculate I eight different times. After these points are generated we check if they are inside V_2 by checking the conditions given by eq. 2.5 for each one. Finally by applying eqs. 1.11 and 1.12 we get our numerical results:

N_i	I	E_t
10^1	1.8	8×10^{-1}
10^2	1.08	8×10^{-2}
10^3	0.864	1.36×10^{-1}
10^4	1.0782	7.82×10^{-2}
10^5	1.00314	3.14×10^{-3}
10^6	1.005768	5.768×10^{-3}
10^7	0.9980172	1.9828×10^{-3}
10^8	1.00004778	4.778×10^{-5}

Table 3.3: A table showing our results after eight Monte Carlo simulations with different sample sizes. The sample size N_i is shown in the first column, the numerical result of the integral I in the second column and the true error of each result E_t in the third column.

Looking at the values of table 3.3, we can see that the true error doesn't necessarily decrease when we increase the number of points N. This is to be expected since our results are based on random numbers and thus our results are subject to that random nature (variance of the uniform distribution). We can conclude that overall as we increase the number of points generated the results' precision increases.

If we compare the error calculated from Simpson's rule from table 3.2 with that of the Monte Carlo method from table 3.3, we can see that Simpson's rule is the more efficient method in this instance. Another advantage of using Simpson's rule is that the results aren't influenced by chance like with Monte Carlo integration. Finally, we present a graph showing how the absolute value of the true error changes when we increase the sample size of our Monte Carlo simulation.

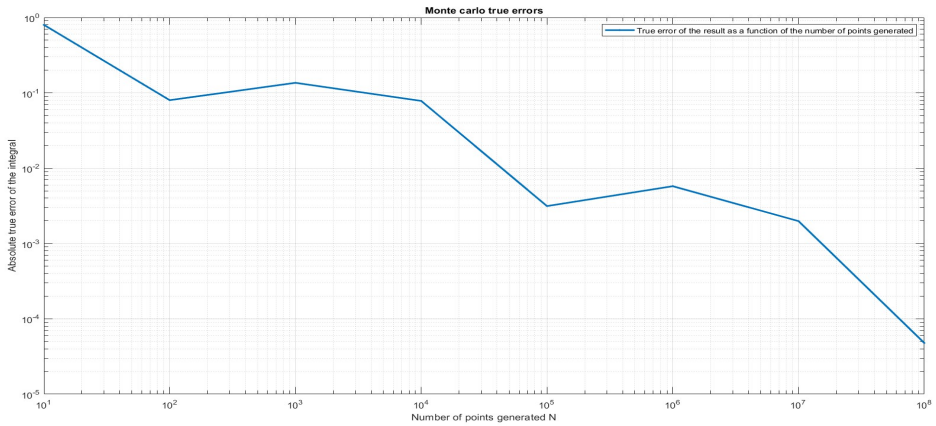

Figure 3.2: The absolute value of the true results we got in a Monte Carlo simulation as a function of the number of points that we generated. These results are subject to change since the points are generated from a uniform distribution, and thus every time we run our program we are going to get different results

Chapter 4

Conclusion

In conclusion, we calculated the double integral I of eq/ 1 in three different ways. The first way was analytically, something that won't always be doable. Using our analytical result we assessed the precision of the two numerical methods we used next. The first numerical method was Simpson's rule which gave us really precise results (up to 10 significant figures) without a big computational cost. On the other hand, by using Monte Carlo integration we didn't get as precise results for the same computational power.

The biggest drawback of applying Simpson's rule in a double integral format is that it requires a big theoretical understanding of the method itself. On the other hand, Monte Carlo integration is a brute-force way of getting a good result, that doesn't require any theoretical background other than drawing pseudorandom numbers coming from a uniform distribution.

A final note is that all three of our methods of calculating I have produced results that fall in line with property 2, as expected.

Bibliography

[1]C. Moustakidis, *Vector analysis second edition* (Sofia publications, 2020).

[2]S. C. Chapra, *Numerical methods for engineers* (Mcgraw-hill, 2010).

[3]R. L. Burden, *Numerical analysis* (Brooks/Cole Cengage Learning, 2011).