



Numerical Integration

Numerical Introductory Course
Marvin Gauer (580553)

Contents

1. Motivation	1
2. Literature Review	1
3. Theory	1
3.1 Review: The Riemann Integral	1
3.2 One-Dimensional Procedures	2
3.2.1 Midpoint Quadrature	2
3.2.2 Simpson-Rule	3
3.2.3 Adaptive Algorithm	3
3.3 Multi-Dimensional Procedures	4
3.3.1 Crude	4
3.3.2 Hit or Miss	4
3.4 Integrals over infinite Intervals	5
4. Application: Approximation of the Normal Distribution	5
5. Conclusion	6
6. Bibliography	6

1. Motivation

Integration is an important operation in mathematics. Unfortunately, in real life applications one might find it extremely difficult or even impossible to solve certain integrals in a closed form. Due to the continuous improvement in computational power one might address this issue by numerically approximating the integral of interest. In order to do so, several procedures have been developed, each with its own advantages and disadvantages. In this document we want to present 5 methods for numerical integration and furthermore apply one of those methods to numerically integrate the normal distribution's density in order to approximate the normal's cumulative distribution function.

2. Literature Review

lalalala

3. Theory

In the following section I want to explain some of the most popular methods in numerical integration. These can be distinguished into one and multi-dimensional methods. Furthermore one might distinguish numerical integration methods further into deterministic and probabilistic methods. But before I start introducing the methods of interest I will do a little recap of the basics:

3.1 Review: The Riemann Integral

The Riemann Integral is one of the two classic concepts of integrals in analysis. It is named after the German mathematician Bernhard Riemann and its aim is to calculate the area between the x -axis and a certain limited function $f : [a; b] \rightarrow \mathbb{R}$. Loosely speaking, the basic idea behind the concept is to approximate the desired integral by summing up different areas of easier to compute rectangles.

The kind of definition I want to present here is the definition using upper and lower sums introduced by Jean Gaston Darboux:

Let $f : [a; b] \rightarrow \mathbb{R}$ be a limited function and $[a; b]$ be an interval. Furthermore, let P be a partition of $[a; b]$ where $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$. Then we can define the upper and lower sums accordingly:

$$U(P) = \sum_{k=1}^n ((x_k - x_{k-1}) \cdot \sup_{x_{k-1} < x < x_k} f(x))$$

$$L(P) = \sum_{k=1}^n ((x_k - x_{k-1}) \cdot \inf_{x_{k-1} < x < x_k} f(x))$$

Now we can compute the infimum and supremum of the upper and lower sum over all partitions P . Therefore it follows:

$$\sup_P L(P) \leq \inf_P U(P)$$

In case of equality, one says that f is Riemann integrable.

3.2 One-Dimensional Procedures

The one dimensional procedures elaborated on in this chapter are classified as deterministic methods. Throughout this document the function $f : [-4; 4] \rightarrow \mathbb{R}$ with $f(x) = x^2 + 3 \cdot x + 4$ is used for visualizing the procedures introduced.

3.2.1 Midpoint Quadrature

The idea of the Midpoint Quadrature directly derives from the definition of the Riemann Integral. We therefore want to calculate the area between the x -axis and a limited function $f : [a; b] \rightarrow \mathbb{R}$. The algorithm works in the way, that we start by partitioning our interval of interest $[a; b]$ into equidistant subintervals $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ with stepwidth $h = \frac{b-a}{n}$. Afterwards we calculate the midpoint $x^{(i)}$ within each subinterval $[x_i; x_{i+1}]$ for $i \in \{0, 1, \dots, n-1\}$ and evaluate f for each $x^{(i)}$. For our approximation it then holds that $\int_a^b f(x) dx \approx \sum_{k=0}^{n-1} f(x^{(i)}) \cdot (x_{i+1} - x_i)$.

An illustration of the procedure can be found in Figure 1.

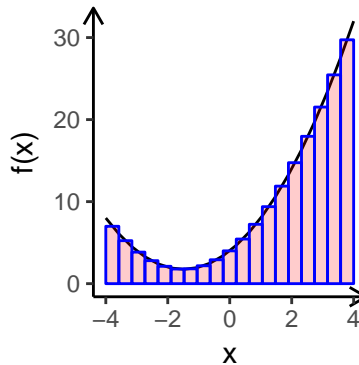


Figure 1: Illustration of the Midpoint or Rectangular Quadrature

When numerically solving an integral one is naturally interested in the error of the approximation which will in the following be denoted by $E(f)$. In case of f having a continuous

second derivative on $[a; b]$ meaning $f \in C_{[a;b]}^{(2)}$, an upper bound for the error of the Midpoint quadrature can be specified as follows:

$$E(f) = \frac{(b-a)^3}{24 \cdot n^2} \cdot \max_{a \leq x \leq b} |f''(x)|$$

3.2.2 Simpson-Rule

The Simpson-Rule is similar to the Rectangular Quadrature, but instead of rectangles quadratic functions are used in order to calculate the area between the x -axis and our limited function $f : [a; b] \rightarrow \mathbb{R}$ more accurately. We again start by partitioning our interval of interest $[a; b]$ into n subintervals $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ with equidistant distances which we will in the following denote by $\Delta x = \frac{b-a}{n}$. Afterwards we calculate the midpoint $x^{(i)}$ within each subinterval $[x_i; x_{i+1}]$ for $i \in \{0, 1, \dots, n-1\}$. Now we use the 3 points $(x_i; f(x_i))$, $(x^{(i)}; f(x^{(i)}))$ and $(x_{i+1}; f(x_{i+1}))$ within each subinterval to interpolate our quadratic functions $g_i(x) : [x_i; x_{i+1}] \rightarrow \mathbb{R}$. For our approximation it then holds that $\int_a^b f(x)dx \approx \frac{\Delta x}{6} \cdot (f(x_0) + 2 \cdot \sum_{k=1}^{n-1} f(x_k) + f(x_n) + 4 \cdot \sum_{k=0}^{n-1} f(x^{(k)}))$.

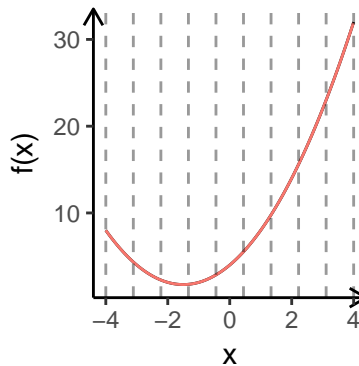


Figure 2: Illustration of the Simpson Rule

The Simpson's rule is an approximation. As with any approximation, before you can safely use it, you must know how good (or bad) the approximation might be. The error of this approximation will in the following be denoted by $E(f)$. In case of f having a continuous fourth derivative on $[a; b]$ meaning $f \in C_{[a;b]}^{(4)}$, an upper bound for the error of the Simpson's quadrature can be specified as follows:

$$E(f) \leq \frac{(b-a)^5}{2880 \cdot n^4} \cdot \max_{a \leq x \leq b} |f^{(4)}(x)|$$

3.2.3 Adaptive Algorithm

When calculating integrals numerically one might in some cases not have unlimited computational power. In this case it might not be smart to use an equidistant stepwidth for in your

numerical integration method. In this case it might be clever to use a wider stepwidth in an area where we can approximate our function well and a narrower one where our function cannot be approximated that well. The adaptive algorithm solves exactly that issue by minimizing the local error (the error within each subinterval) until it reaches a certain error threshold. The algorithm therefore starts with an initial number of subintervals m . It then approximates

3.3 Multi-Dimensional Procedures

In lots of applications multidimensional integrals need to be solved numerically. This means that we now want to calculate the integral $\int_{\Omega} f(\mathbf{x}) d\mathbf{x} = \int_{a_1}^{b_1} \dots \int_{a_m}^{b_m} f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m$. One might have the idea, to use the quadrature rules explained above in a multidimensional setting to calculate the integral at hand, but we can easily see that this will end in the **Curse of dimensionality**. Therefore I want to concentrate on Monte Carlo integration methods going on.

3.3.1 Crude

The Crude Monte Carlo Integration method is a fairly simple method to calculate integrals numerically. It works in the way, that we partition k bins $[a_{1,j}; b_{1,j}] \times [a_{2,j}; b_{2,j}] \times \dots \times [a_{m,j}; b_{m,j}] \forall j \in \{1, \dots, k\}$. Furthermore, one needs to generate n m -dimensional uniformly distributed points $\mathbf{x}_i = (x_1, \dots, x_m) \forall i \in \{1, \dots, n\}$. In the next step, we calculate $f(\mathbf{x}_i) \forall i \in \{1, \dots, n\}$ and the means m_i for each bin. Therefore it holds that $\int_a^b f(x) dx \approx \sum_{t=1}^k (\prod_{l=1}^m (b_{l,t} - a_{l,t}) \cdot m_t)$

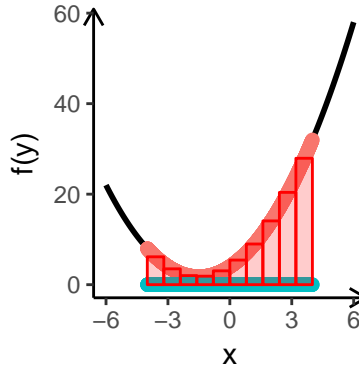


Figure 3: Illustration of the Crude Monte Carlo integration method

3.3.2 Hit or Miss

On the other hand, the Hit or Miss Monte Carlo Integration method works slightly differently. Here we create n $m + 1$ -dimensional uniformly distributed points $(x_{i,1}, \dots, x_{i,m}, y_i)$ for $i \in \{1, \dots, n\}$. The first m dimensions are for our domain $[a_1; b_1] \times [a_2; b_2] \times \dots \times [a_m; b_m]$ and are always uniformly distributed on $[a_i; b_i] \forall i \in \{1, \dots, m\}$. The y_i coordinate is for our

$f(\mathbf{x})$ -values and these are uniformly distributed on $[0; \max(f(\mathbf{x}))]$. Now the percentage of points p for which holds $y_i \leq f(x_1, \dots, x_m)$ and the total area/volume A surrounding the graph needs to be calculated. A is basically the area/volume of the domain of the generated random points. Therefore it holds that $\int_a^b f(x)dx \approx A \cdot p$

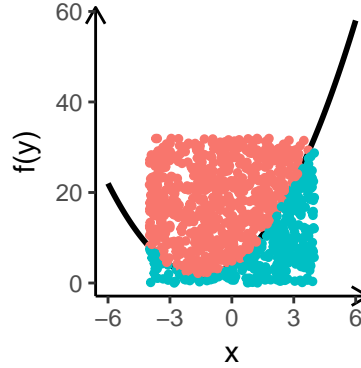


Figure 4: Illustration of the Hit or Miss Monte Carlo integration method

Obviously, the quality of the answer depends on the quality of the random number generator sequence which is used.

3.4 Integrals over infinite Intervals

Working with the procedures elaborated on above one needs to have certain limits for the integral to approximate, but in many applications one might also be interested in numerically solving improper integrals. I therefore want to elaborate on a method known as change of variables in order to account for the issue described above. The basic idea of the procedure is to adjust the integrand in such a way, that we have an equivalent definite integral at the end. Therefore we can use the following three functions to transform our integral at hand:

$$\begin{aligned}\int_{-\infty}^{\infty} f(x)dx &= \int_{-1}^1 f\left(\frac{t}{1-t}\right) \cdot \frac{1+t^2}{(1-t^2)^2} dt \\ \int_a^{\infty} f(x)dx &= \int_0^1 f\left(a + \frac{t}{1-t}\right) \cdot \frac{1}{(1-t)^2} dt \\ \int_{-\infty}^a f(x)dx &= \int_0^1 f\left(a - \frac{1-t}{t}\right) \cdot \frac{1}{t^2} dt\end{aligned}$$

4. Application: Approximation of the Normal Distribution

lalalala

5. Conclusion

lalalala

6. Bibliography

(“Monte-Carlo-Integration”)

“Monte-Carlo-Integration.” <https://www.mathematik.tu-clausthal.de/interaktiv/integration/montecarlo/>.