

# Monte-Carlo-Integration

Mathematische Herleitung

Daniel Zahnd

13. Dezember 2021

## 1 Einleitung

Die meisten reelle physikalische Probleme beschreibenden Integrale sind nicht analytisch lösbar. Deswegen benötigt man numerische Verfahren, um solche Integrale dennoch lösen zu können. Dabei gibt es streng «geometrische» Verfahren, aber auch statistische Möglichkeiten. Die Monte-Carlo-Integrationsmethode gehört zu letzteren Verfahren und hat den Vorteil gegenüber rein geometrischen Lösungsalgorithmen, dass sie bei hochdimensionalen Integrationen viel schneller ist. Demgegenüber konvergieren statistische Verfahren wie die Monte-Carlo-Integrationsmethode generell langsamer, weswegen für niedrigdimensionale Problemstellungen dennoch die geometrischen Verfahren günstiger sind.

Das nachfolgend eingeführte wahrscheinlichkeitstheoretische Material stützt sich auf [Molchanov, 2017].

## 2 Definitionen

Um die Monte-Carlo-Integrationsmethode mathematisch formal herleiten zu können, müssen zunächst einige Definitionen eingeführt werden.

**Definition 2.1** (Indikatorfunktion). *Sei  $\Omega$  eine beliebige Menge. Die Indikatorfunktion für eine Menge  $A \subseteq \Omega$  auf  $\Omega$  ist definiert durch*

$$\mathbb{1}_A = \Omega \rightarrow \{0, 1\}, \omega \mapsto \begin{cases} 0 & , \quad \text{wenn } \omega \notin A \\ 1 & , \quad \text{wenn } \omega \in A. \end{cases} \quad (1)$$

Um Wahrscheinlichkeiten mathematisch zu formalisieren, muss ferner ein sogenannter Wahrscheinlichkeitsraum definiert werden. Dies geschieht nachfolgend in allgemeiner Weise, wobei bemerkt sei, dass in der Praxis häufig diskrete Wahrscheinlichkeitsräume auftreten, welche aber in der Definition der allgemeinen Wahrscheinlichkeitsräume enthalten sind.

**Definition 2.2** (Allgemeiner Wahrscheinlichkeitsraum). *Es sein  $\Omega \neq \emptyset$  eine Menge und  $\mathcal{F}$  eine  $\sigma$ -Algebra für  $\Omega$ , das heisst*

$$(1) \quad \Omega \in \mathcal{F},$$

- (2)  $A \in \mathcal{F} \Rightarrow A^c \in \mathcal{F}$ ,  
 (3)  $A_1, A_2, \dots \in \mathcal{F} \Rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ ,

ferner sei eine Abbildung auf  $\mathcal{F}$  definiert durch  $P : \mathcal{F} \rightarrow [0, 1]$  mit den Eigenschaften

- (a)  $P(\Omega) = 1$ ,  
 (b)  $A, B \in \mathcal{F}$  mit  $A \cap B \neq \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)$  und  
 (c)  $(A_n)_{n \in \mathbb{N}} \subseteq \mathcal{F}$  mit  $A_m \cap A_n \neq \emptyset \forall m \neq n \Rightarrow P(\bigcup_{n \in \mathbb{N}} A_n) = \sum_{n=1}^{\infty} P(A_n)$ ,

wobei  $P$  Wahrscheinlichkeitsfunktion oder Wahrscheinlichkeitsmass heisst.

Eine Zufallsvariable nun, welcher eine bestimmte Wahrscheinlichkeit  $P(A)$  des Eintreffens eines darin enthaltenen Ereignisses  $A \in \mathcal{F}$  zugeordnet werden kann, wird mathematisch durch folgende Definition erfasst:

**Definition 2.3** (Allgemeine Zufallsvariable). Sei  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum. Eine Funktion

$$X : \Omega \rightarrow \mathbb{R}, \quad \{\omega \in \Omega \mid X(\omega) \leq t\} \in \mathcal{F} \quad \forall t \in \mathbb{R} \quad (2)$$

heisst allgemeine Zufallsvariable.

Wiederum sei bemerkt, dass in der Praxis häufig sogenannte diskrete Zufallsvariablen auftreten, die etwa einen Würfelwurf mit der Augenzahl  $i \in \{1, \dots, 6\}$  beschreiben. Wird das Konzept der allgemeinen Zufallsvariable auf mehr als eine Dimension ausgedehnt, so ergibt sich die Definition einer allgemeinen mehrdimensionalen Zufallsvariable, beziehungsweise eines allgemeinen Zufallsvektors. Dessen Definition sei nachfolgend gegeben:

**Definition 2.4** (Allgemeine mehrdimensionale Zufallsvariable). Seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum sowie  $X_1, \dots, X_n$  allgemeine Zufallsvariablen  $X_i : \Omega \rightarrow \mathbb{R}$  für  $i \in \{1, \dots, n\}$ ,  $n \in \mathbb{N}$  nach Definition 2.3. Eine allgemeine mehrdimensionale Zufallsvariable, auch allgemeiner Zufallsvektor genannt, ist sodann definiert durch

$$X : \Omega \rightarrow \mathbb{R}^n, \quad X = (X_1, \dots, X_n). \quad (3)$$

Zufallsvariablen, beziehungsweise Zufallsvektoren werden üblicherweise in Grossbuchstaben notiert, während deren Realisierungen, also partikuläre Zufallswerte des entsprechenden Zufallsvektors oder der entsprechenden Zufallsvariable, in Kleinbuchstaben geschrieben werden. Nachfolgend werden die Begriffe 'Zufallsvariable' und 'Zufallsvektor' synonym verwendet, wobei, wo relevant und für das Verständnis hilfreich, explizit der exakte Begriff verwendet wird. Das Satzwerk jedoch gilt sowohl für Zufallsvariablen, als auch für Zufallsvektoren.

Das Konzept der Zufallsvariablen an einem Beispiel verdeutlicht. Es werde ein Münzwurfexperiment mit einem passenden Wahrscheinlichkeitsraum  $(\Omega, \mathcal{F}, P)$  ausgeführt. Die Grundmenge  $\Omega$ , also die Menge aller möglichen Ergebnisse des Experimentes, ist mit  $\Omega = \{\text{Kopf}, \text{Zahl}\}$  zu bezeichnen. Das Zufallsexperiment bestehe darin, dass beim Erscheinen

von 'Kopf' ein Geldstück ausgezahlt wird, beim Erscheinen von 'Zahl' jedoch kein Geld fließt. Die Zufallsvariable  $X$ , definiert durch

$$X(\omega) \doteq \begin{cases} 0 & , \quad \text{falls } \omega = \text{Zahl} \\ 1 & , \quad \text{falls } \omega = \text{Kopf}, \end{cases} \quad (4)$$

modelliert dieses Experiment. Es kann aber auch eine ganz andere Zufallsvariable  $Z$  definiert werden, die ein anderes Zufallsexperiment beschreibt, welches sich der Elemente von  $\Omega$  mit dem entsprechenden Wahrscheinlichkeitsraum  $(\Omega, \mathcal{F}, P)$  bedient.

Überdies haben Zufallsvariablen eine bestimmte Struktur, die Verteilung, beziehungsweise Verteilungsfunktion genannt wird und wie folgt gegeben ist:

**Definition 2.5** (Allgemeine Verteilungsfunktion). *Seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum und  $X$  eine allgemeine Zufallsvariable. Die Verteilungsfunktion  $F_X$  von  $X$  ist gegeben durch*

$$F_X : X(\Omega) \subseteq \mathbb{R} \rightarrow [0, 1], \quad (5)$$

$$F_X(y) \doteq P(\{\omega \in \Omega \mid X(\omega) \leq y\}) \doteq P(X \leq y), \quad y \in \mathbb{R}. \quad (6)$$

Einen Spezialfall bilden die absolut stetigen Verteilungsfunktionen, welche insbesondere für die Monte-Carlo-Integration relevant sind, wobei häufig angenommen wird, dass die Dichtefunktion  $f(x)$  einer betrachteten Zufallsvariable  $X$  glatt ist, das heisst  $f(x) \in C^\infty$ . Die Voraussetzungen dafür, dass eine Verteilungsfunktion  $F$  einer Zufallsvariable  $X$  mit einer Dichtefunktion  $f$  absolut stetig ist, sind in nachfolgender Definition enthalten.

**Definition 2.6** (Absolut stetige Verteilungsfunktion). *Seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum und  $X$  eine allgemeine Zufallsvariable mit einer Verteilungsfunktion  $F$ . Wenn  $F$  die Stammfunktion von  $f : \mathbb{R} \rightarrow [0, \infty)$  ist, d.h.*

$$F(x) = \int_{-\infty}^x f(u) \, du, \quad (7)$$

*dann heisst  $F$  absolut stetige Verteilungsfunktion. Die Funktion  $f(x)$  heisst Wahrscheinlichkeitsdichtefunktion von  $F$  und es müssen*

$$(a) \quad f(x) \geq 0 \quad \forall x \in \mathbb{R} \text{ sowie}$$

$$(b) \quad \int_{-\infty}^{\infty} f(u) \, du = 1$$

*gelten. Falls  $f(x)$  stetig in  $x \in \mathbb{R}$  ist, so gilt  $F'(x) = \frac{dF(x)}{dx} = f(x)$ . Für  $a, b \in \mathbb{R}$  mit  $a < b$  gilt*

$$P(a \leq X \leq b) = P(a < X < b) = F(b) - F(a) = \int_a^b f(u) \, du. \quad (8)$$

Oft interessiert es bei einem Zufallsexperiment mit einer oder mehreren involvierten Zufallsvariablen  $X$ , welches Resultat bei vielen Wiederholungen des Experimentes zu er-

warten ist. Aufschluss hierüber gibt der Erwartungswert, der nachfolgend für diskrete Zufallsvariable definiert wird.

**Definition 2.7** (Erwartungswert diskreter Zufallsvariablen). *Sei  $(\Omega, P)$  ein diskreter Wahrscheinlichkeitsraum und  $X$  eine diskrete Zufallsvariable. Der Erwartungswert  $E(X)$  der Zufallsvariable  $X$  ist definiert durch*

$$E(X) = \sum_{\omega \in \Omega} X(\omega)P(\{\omega\}) = \sum_{x \in X(\Omega)} xP(X = x), \quad (9)$$

*falls  $E(|X|) < \infty$ .*

Auch der Erwartungswert von Zufallsvariablen, welche eine absolut stetige Verteilungsfunktion besitzen, kann definiert und berechnet werden.

**Definition 2.8** (Erwartungswert allgemeiner Zufallsvariablen mit absolut stetiger Verteilung). *Seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum,  $X$  eine allgemeine Zufallsvariable mit einer absolut stetigen Verteilungsfunktion  $F_X$  und  $f_X(x)$  die Dichtefunktion von  $F_X$ . Falls  $E(|X|) < \infty$  ist der Erwartungswert von  $X$  gegeben durch das Integral*

$$E(X) = \int_{-\infty}^{\infty} x f_X(x) dx. \quad (10)$$

*Ist  $Y = g(X)$  eine weitere Zufallsvariable mit absolut stetiger Verteilungsfunktion  $F_Y$  und Dichtefunktion  $f_Y(x)$ , so ist deren Erwartungswert gegeben durch*

$$E(Y) = E(g(X)) = \int_{-\infty}^{\infty} g(x) f_Y(x) dx. \quad (11)$$

### 3 Theoreme

Um das zu einer vollständigen Herleitung der Monte-Carlo-Integrationsmethodik nötige schwache Gesetz der grossen Zahlen zu beweisen, müssen zunächst zwei Ungleichungen statuiert und bewiesen werden, namentlich die Markov-Ungleichung und die Tschebycheff-Ungleichung. Mittels dieser beiden Ungleichungen, sowie elementarer Identitäten für die Rechnung mit Erwartungswerten und Varianzen, kann schliesslich das schwache Gesetz der grossen Zahlen gezeigt werden.

**Satz 3.1** (Markov-Ungleichung). *Es seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum,  $X$  eine Zufallsvariable und  $\epsilon > 0$ . Dann gilt*

$$P(|X| \geq \epsilon) \leq \frac{E(|X|)}{\epsilon}. \quad (12)$$

*Beweis.* Es gilt  $|X| \geq \mathbb{1}_{\{|X| \geq \epsilon\}}|X| \geq \mathbb{1}_{\{|X| \geq \epsilon\}}\epsilon$ . Dann ist ferner

$$\begin{aligned} E(|X|) &\geq E(\mathbb{1}_{\{|X| \geq \epsilon\}}\epsilon) = \epsilon E(\mathbb{1}_{\{|X| \geq \epsilon\}}) = \epsilon \int_{\omega \in \Omega} \mathbb{1}_{\{|X| \geq \epsilon\}} dP(\{\omega\}) \\ &= \epsilon \int_{\omega \in \Omega: |X(\omega)| \geq \epsilon} dP(\{\omega\}) = \epsilon P(|X| \geq \epsilon) \Leftrightarrow P(|X| \geq \epsilon) \leq \frac{E(|X|)}{\epsilon}. \end{aligned} \quad (13)$$

□

Die Markov-Ungleichung wird daraufhin im Beweis der Tschebycheff-Ungleichung eingesetzt.

**Satz 3.2** (Tschebycheff-Ungleichung). *Es seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum,  $X$  eine Zufallsvariable mit  $E(X^2) < \infty$  und  $\epsilon > 0$ . Es gilt dann*

$$P(|X - E(X)| \geq \epsilon) \leq \frac{V(X)}{\epsilon^2}. \quad (14)$$

*Beweis.* Es gilt unter Verwendung der Markov-Ungleichung, dass

$$\begin{aligned} P(|X - E(X)| \geq \epsilon) &= P((X - E(X))^2 \geq \epsilon^2) \\ &\leq \frac{E((X - E(X))^2)}{\epsilon^2} = \frac{V(X)}{\epsilon^2}, \end{aligned} \quad (15)$$

wobei die Definition der Varianz  $V(X) = E((X - E(X))^2)$  im letzten Schritt angewandt wurde. □

Mithilfe der beiden obigen Ungleichungen kann man nun das Gesetz der grossen Zahlen beweisen, welches zur Plausibilisierung der Monte-Carlo-Methode für die Berechnung  $d$ -Dimensionalen Integralen verwendet werden kann.

**Satz 3.3** (Schwaches Gesetz der grossen Zahlen). *Es seien  $(\Omega, \mathcal{F}, P)$  ein allgemeiner Wahrscheinlichkeitsraum,  $(X_n)_{n \in \mathbb{N}}$  eine Folge von unabhängigen und identisch verteilten Zufallsvariablen- oder Vektoren mit  $E(X_1^2) < \infty$ . Es sei zudem*

$$\bar{X}_n \doteq \frac{1}{n} \sum_{i=1}^n X_i \quad (16)$$

*für  $n \in \mathbb{N}$ . Dann gilt*

$$\lim_{n \rightarrow \infty} P(|\bar{X}_n - E(X_1)| \geq \epsilon) = 0 \quad \forall \epsilon > 0. \quad (17)$$

*Beweis.* Es gilt zunächst

$$E(\bar{X}_n) = \frac{1}{n} \sum_{i=1}^n E(X_i) = \frac{1}{n} n E(X_1) = E(X_1). \quad (18)$$

Hierbei wurde die Definition der Zufallsvariable  $\bar{X}_n$  in Kombination mit der Additivität und Linearität von Erwartungswerten verwendet. Darüber hinaus wurde die identischen Verteilungen der  $X_i$  benutzt, was auch zur Berechnung der folgenden Varianz benutzt wird. Für die Varianz  $V(\bar{X}_n)$  gilt ebenfalls nach Definition der Zufallsvariable  $\bar{X}_n$ , sowie wegen der Additivität unabhängiger Zufallsvariablen und der Skaliereneigenschaft von Varianzen mit einer Konstante

$$V(\bar{X}_n) = V\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{1}{n^2} \sum_{i=1}^n V(X_i) = \frac{1}{n^2} n V(X_1) = \frac{V(X)}{n}. \quad (19)$$

Gemäss der Tschebycheff-Ungleichung 3.2 und obigen Resultaten gilt für  $\epsilon > 0$ :

$$\begin{aligned} P(|\bar{X}_n - E(\bar{X}_n)| \geq \epsilon) &= P(|\bar{X}_n - E(X_1)| \geq \epsilon) \leq \frac{V(\bar{X}_n)}{\epsilon^2} \\ &= \frac{V(X)}{\epsilon^2 n} \xrightarrow{n \rightarrow \infty} 0. \end{aligned} \quad (20)$$

□

## 4 Herleitung

Mit den gegebenen Definitionen, Theoremen und Beweisen kann nun die eigentliche Monte-Carlo-Integration hergeleitet werden. Zunächst sei die Menge  $D \doteq [a_1, b_1] \times \dots \times [a_d, b_d] \subseteq \mathbb{R}^d$  für  $d \in \mathbb{N}$  definiert. Es seien ferner  $X_1, \dots, X_n \subseteq D$  unabhängige, identisch und gleichverteilte Zufallsvektoren nach Definition 2.4 mit  $n \in \mathbb{N}$  und glatten Dichtefunktionen

$$f_{X_i}(x_i) = \begin{cases} \frac{1}{b_{i_j} - a_{i_j}} & , \quad x_{i_j} \in [a_{i_j}, b_{i_j}] \\ 0 & , \quad \text{sonst} \end{cases}, \quad x_i = (x_{i_1}, \dots, x_{i_d}), \quad i \in \{1, \dots, n\} \quad (21)$$

gegeben. Ausserdem seien die Zufallsvektoren  $X_i$  nur auf den Mengen  $D_i \doteq [a_{i_1}, b_{i_1}] \times \dots \times [a_{i_d}, b_{i_d}]$  für  $i \in \{1, \dots, n\}$  vom Nullvektor verschieden. Es gelte darüber hinaus, dass  $a_j \doteq a_{i_j}$  und  $b_j \doteq b_{i_j} \quad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, d\}$ . Folglich gilt  $D_i = D \quad \forall i \in \{1, \dots, n\}$ . Ein Zufallsvektor  $X_i$  für ein  $i \in \{1, \dots, n\}$  stellt also hierbei einen zufällig, beispielsweise durch ein Computerprogramm ausgewählten Punkt  $p = (p_1, \dots, p_d) \in D$  dar. Ein Zufallsvektor  $X_i$  ist entsprechend Definition 2.4 definiert durch  $X_i : \Omega \rightarrow \mathbb{R}^d$  mit der Grundmenge  $\Omega = D_i \subseteq \mathbb{R}^d$ . Im vorliegenden Fall kann die Zuordnungsvorschrift des Zufallsvektors  $X_i : \Omega \rightarrow \mathbb{R}^d$  ferner durch  $X_i = \text{Id}_{X_i}$  mit der Identitätsabbildung  $\text{Id}$  angegeben werden, da  $X_i$  gemäss Voraussetzung eine Gleichverteilung aufweist, was zur Identifikation der Zielmenge von  $X_i$  mit  $D_i = D$  führt, sodass sich  $X_i : D_i \rightarrow D_i$  ergibt. Des Weiteren sei eine über den Integrationsbereich  $D$  zu integrierende integrierbare Funktion

$$\varphi : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x_i = (x_{i_1}, \dots, x_{i_d}) \mapsto \varphi(x_{i_1}, \dots, x_{i_d}) = \varphi(x_i) \quad (22)$$

eingeführt. Das mit den Monte-Carlo-Methoden zu berechnende Integral ist lautet dabei

$$I \doteq \int_D \varphi(u) \, du = \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} \varphi(u_1, \dots, u_d) \, du_1 \dots du_d. \quad (23)$$

Weil die Funktion  $\varphi$  integrierbar ist, berechnet sich der Erwartungswert einer Zufallsvariable  $Y_i \doteq \varphi(X_i)$  für ein  $i \in \{1, \dots, n\}$  nach Definitionen 2.4 und 2.8 zu

$$\begin{aligned}
E(Y_i) &= E(\varphi(X_i)) = \int_{\mathbb{R}^d} \varphi(x_i) f_{X_i}(x_i) dx_i \\
&= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \varphi(x_{i_1}, \dots, x_{i_d}) f_{X_i}(x_{i_1}, \dots, x_{i_d}) dx_{i_1} \dots dx_{i_d} \\
&= \frac{1}{(b_{i_1} - a_{i_1}) \cdot \dots \cdot (b_{i_d} - a_{i_d})} \int_{a_{i_1}}^{b_{i_1}} \dots \int_{a_{i_d}}^{b_{i_d}} \varphi(x_{i_1}, \dots, x_{i_d}) dx_{i_1} \dots dx_{i_d} \\
&= \frac{1}{\mathcal{V}_i} \int_{a_{i_1}}^{b_{i_1}} \dots \int_{a_{i_d}}^{b_{i_d}} \varphi(x_{i_1}, \dots, x_{i_d}) dx_{i_1} \dots dx_{i_d} \stackrel{(23)}{=} \frac{I}{\mathcal{V}_i},
\end{aligned} \tag{24}$$

wobei das Volumen  $\mathcal{V}_i$  durch

$$\mathcal{V}_i \doteq (b_{i_1} - a_{i_1}) \cdot \dots \cdot (b_{i_d} - a_{i_d}) \tag{25}$$

definiert wurde und per Definition der Intervallgrenzen  $\mathcal{V}_i = \mathcal{V}_j \doteq \mathcal{V} \quad \forall i, j \in \{1, \dots, n\}$  gilt. Gleichungen (23) und (4) ergeben zusammen die Gleichung

$$E(\varphi(X_i)) \mathcal{V}_i = \int_D \varphi(u) du = \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} \varphi(u_1, \dots, u_d) du_1 \dots du_d = I. \tag{26}$$

Aus den Voraussetzungen, woraus obige Gleichung hergeleitet wurde, ergibt sich insbesondere die Relation  $E(\varphi(X_i)) \mathcal{V}_i = E(\varphi(X_j)) \mathcal{V}_j \quad \forall i, j \in \{1, \dots, n\}$ . Man definiere nun die Zufallsvariable

$$\bar{Y}_n \doteq \frac{1}{n} \sum_{i=1}^n Y_i = \frac{1}{n} \sum_{i=1}^n \varphi(X_i). \tag{27}$$

Es sei alsdann  $\epsilon > 0$ . Setzt man nun die in obiger Gleichung (27) definierte Variable mit der weiter oben definierten Variable  $Y_i = \varphi(X_i)$  im Gesetz der Grossen Zahlen nach Satz 3.3 ein, so erhält man für ein beliebiges  $j \in \{1, \dots, n\}$

$$\begin{aligned}
\lim_{n \rightarrow \infty} P(|\bar{Y}_n - E(Y_j)| \geq \epsilon) &= P\left(\left|\frac{1}{n} \sum_{i=1}^n \varphi(X_i) - \frac{I}{\mathcal{V}_j}\right| \geq \epsilon\right) \\
&= P\left(\left|\frac{1}{n} \sum_{i=1}^n \varphi(X_i) - \frac{I}{\mathcal{V}}\right| \geq \epsilon\right) \xrightarrow[\text{Satz 3.3}]{n \rightarrow \infty} 0.
\end{aligned} \tag{28}$$

Betrachtet man diese Gleichung danach heuristisch im Limit  $\epsilon \rightarrow 0$ , dann verschwindet die Wahrscheinlichkeit  $P$  dafür, dass die Summe  $\frac{1}{n} \sum_{i=1}^n \varphi(X_i)$  vom Wert des Integrals  $I$  dividiert durch das Volumen  $\mathcal{V}$  abweicht, daher gilt für ein 'genügend grosses'  $n \in \mathbb{N}$

$$I = \int_D \varphi(u) du = \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} \varphi(u_1, \dots, u_d) du_1 \dots du_d \approx \frac{\mathcal{V}}{n} \sum_{i=1}^n \varphi(X_i). \quad (29)$$

Wie aus Gleichung (3) ersichtlich ist, wird der Fehler  $\xi(\epsilon, n)$  der Approximation (29) bei konstantem  $\epsilon$  mit zunehmendem  $n$  kleiner. Damit der Bruch in (3) für variables  $\epsilon$  tatsächlich gegen null geht, wie es für die exakte Gleichheit in der Monte-Carlo-Integration (29) gefordert ist, muss  $n$  für  $n, \epsilon^{-1} \rightarrow \infty$  schneller wachsen als  $\epsilon^{-2}$ . Man kann mit dem in (29) begründeten Verfahren also ein Integral beliebig genau approximieren, indem man ein beliebig kleines  $\epsilon$  und im Grenzfall  $\epsilon \rightarrow 0$  wählt. Allerdings ist zu beachten, dass die Approximation

$$I \approx \frac{\mathcal{V}}{n} \sum_{i=1}^n \varphi(X_i) \quad (30)$$

mit kleiner werdendem  $\epsilon$  ein umso höheres  $n$  fordert, damit die Approximation in guter Näherung mit dem exakten Integral  $I$  übereinstimmt. Man erinnere sich nun, dass die  $X_1, \dots, X_n$   $d$ -dimensionale Zufallsvektoren sind. Das heisst, dass mit einem Computerprogramm  $n$  Punkte im  $d$ -Dimensionalen Raum  $D$  erzeugt werden können, welche dann als eine Realisierung der Zufallsvektoren  $X_1, \dots, X_n$  als  $x_1, \dots, x_n$  zu betrachten sind, wobei  $x_i = (x_{i_1}, \dots, x_{i_d})$  für  $i \in \{1, \dots, n\}$  gilt.

## 5 Implementierung

Wie im obigen Abschnitt bereits angedeutet, werden zur Monte-Carlo-Integration zunächst  $n$  Realisierungen  $r_i$  eines  $d$ -dimensionalen Zufallsvektors  $X_i$  mit  $r_i \in D_i = D$  erzeugt. Mithilfe dieser erzeugten Zufallspunkte kann nun die Funktion  $\varphi(r_i)$  für jede Realisierung  $r_i$  evaluiert werden. Das zu berechnende Integral (23) kann sodann durch die Vorschrift

$$I \approx \frac{\mathcal{V}}{n} \sum_{i=1}^n \varphi(r_i) \quad (31)$$

mit einer `for`-Schleife berechnet beliebig genau berechnet werden.

Im nachfolgenden Abschnitt ist eine Beispielimplementierung in der Programmiersprache Python zu finden, der eine vierdimensionale Funktion  $h(x_0, x_1, x_2, x_3)$  integrieren kann. Somit bietet der Integrator Möglichkeiten, viele Probleme der Physik lösen zu können, da dort oftmals vierdimensionale Integrale vorkommen - über drei Raumdimensionen und eine Zeitdimension.

### 5.1 Beispielimplementierung

#### 5.1.1 Code

Nachfolgend sei zunächst der folgende Programmcode in der Programmiersprache Python betrachtet:

```
#Import packages
import math as m
import numpy as np
```



```

def monteCarloInt(a_0, b_0, a_1, b_1, a_2, b_2, a_3, b_3, n, dim):
    """
    Parameters
    -----
    a_0 : TYPE: Floating point number.
           DESCRIPTION: Lower boundary of integration in zeroth
           ↪ component.
    b_0 : TYPE: Floating point number.
           DESCRIPTION: Upper boundary of integration in zeroth
           ↪ component.
    a_1 : TYPE: Floating point number.
           DESCRIPTION: Lower boundary of integration in first
           ↪ component.
    b_1 : TYPE: Floating point number.
           DESCRIPTION: Upper boundary of integration in first
           ↪ component.
    a_2 : TYPE: Floating point number.
           DESCRIPTION: Lower boundary of integration in second
           ↪ component.
    b_2 : TYPE: Floating point number.
           DESCRIPTION: Upper boundary of integration in second
           ↪ component.
    a_3 : TYPE: Floating point number.
           DESCRIPTION: Lower boundary of integration in third
           ↪ component.
    b_3 : TYPE: Floating point number.
           DESCRIPTION: Upper boundary of integration in third
           ↪ component.
    n : TYPE: Integer, natural number.
           DESCRIPTION: Number of random vectors to be generated for
           ↪ sampling the
           function.
    dim : TYPE: Integer, natural number.
           DESCRIPTION: Number of dimensions of implemented
           ↪ function.

    Important
    -----
    The function to be integrated using this Monte-Carlo integration
    ↪ method
    needs to be implemented directly in the code. That means, it must
    ↪ be hard
    coded in line xxx.\n\n
    If there is not a 4-dimensional integration to be carried out, it
    ↪ shall be

```

begun with a\_0, b\_0 and so on to fill moneCarloInt(). The  
 ↪ corresponding  
 number of dimensions for the integration is to be assigned to  
 ↪ dim.

Returns

-----

Value of the desired integral.

```
'''
if (dim == 1):
    # print('1-dimensional integration chosen.')
    # Generate n random vectors of dimension 1
    x = np.random.rand(1, n)
    x[0][:] = x[0][:]*abs(b_0-a_0)-(0-a_0)

    # Define integral value
    I = float(0)

    # Calculate integral
    for i in range(len(x[0])):
        I = I + m.exp(-2*(x[0][i]**2)) # Function
        ↪ implementation
    I = I*abs(b_0-a_0)*n**(-1)

    # Return integral value
    return I

if (dim == 2):
    # print('2-dimensional integration chosen.')
    # Generate n random vectors of dimension 2
    x = np.random.rand(2, n)
    x[0][:] = x[0][:]*abs(b_0-a_0)-(0-a_0)
    x[1][:] = x[1][:]*abs(b_1-a_1)-(0-a_1)

    # Define integral value
    I = float(0)

    # Calculate integral
    for i in range(len(x[0])):
        I = I + m.exp(-2*(x[0][i]**2+x[1][i]**2)) #
        ↪ Function implementation
    I = I*abs(b_0-a_0)*abs(b_1-a_1)*n**(-1)

    # Return integral value
    return I

if (dim == 3):
```

```

# print('3-dimensional integration chosen.')
# Generate n random vectors of dimension 3
x = np.random.rand(3, n)
x[0][:] = x[0][:]*abs(b_0-a_0)-(0-a_0)
x[1][:] = x[1][:]*abs(b_1-a_1)-(0-a_1)
x[2][:] = x[2][:]*abs(b_2-a_2)-(0-a_2)

# Define integral value
I = float(0)

# Calculate integral
for i in range(len(x[0])):
    I = I +
    ↪ m.exp(-2*(x[0][i]**2+x[1][i]**2+x[2][i]**2))
    ↪ # Function implementation
I = I*abs(b_0-a_0)*abs(b_1-a_1)*abs(b_2-a_2)*n**(-1)

# Return integral value
return I

if (dim == 4):
    # print('4-dimensional integration chosen.')
    # Generate n random vectors of dimension 3
    x = np.random.rand(4, n)
    x[0][:] = x[0][:]*abs(b_0-a_0)-(0-a_0)
    x[1][:] = x[1][:]*abs(b_1-a_1)-(0-a_1)
    x[2][:] = x[2][:]*abs(b_2-a_2)-(0-a_2)
    x[3][:] = x[3][:]*abs(b_3-a_3)-(0-a_3)

    # Define integral value
    I = float(0)

    # Calculate integral
    for i in range(len(x[0])):
        I = I + m.exp(-2*(x[0][i]**2+x[1][i]**2+x[2][i]**
        ↪ *2+x[3][i]**2)) # Function
        ↪ implementation
    I = I*abs(b_0-a_0)*abs(b_1-a_1)*abs(b_2-a_2)*abs(b_3-a_3)
    ↪ )*n**(-1)

    # Return integral value
    return I

else:
    I = 0
    print('Something with the input was wrong, please check
    ↪ input and compare with function description of
    ↪ monteCarloInt!')
```

```

return I

# Execute program
Integral_1 = monteCarloInt(-1, 3, 0, 0, 0, 0, 0, 0, 1000000, 1)
Integral_2 = monteCarloInt(-1, 3, 1, 5, 0, 0, 0, 0, 1000000, 2)
Integral_3 = monteCarloInt(-1, 3, 1, 5, -1, 1, 0, 0, 1000000, 3)
Integral_4 = monteCarloInt(-1, 3, 1, 5, -1, 1, -18, 7, 1000000, 4)

# Print results
print('Integral_1 is I = ',Integral_1)
print('Integral_2 is I = ',Integral_2)
print('Integral_3 is I = ',Integral_3)
print('Integral_4 is I = ',Integral_4)

```

Das hier gezeigte Beispiel realisiert eine Monte-Carlo-Integration der vierdimensionalen Funktion

$$h(x_0, x_1, x_2, x_3) = e^{-2(x_0^2 + x_1^2 + x_2^2 + x_3^2)} \quad (32)$$

über die Menge

$$D_{bsp} \doteq [a_{i_1}, b_{i_1}] \times [a_{i_2}, b_{i_3}] \times [a_{i_4}, b_{i_5}] \times [a_{i_1}, b_{i_1}] \doteq [-1, 3] \times [1, 5] \times [-1, 1] \times [-18, 7], \quad (33)$$

wobei für die Anzahl der erzeugten Zufallsvektoren in  $D_{bsp}$  die Zahl  $n \doteq 10^6$  gewählt wurde und  $i \in \{1, \dots, n\}$  ist, hierbei ferner  $a_{i_k} = a_{j_k}$  und  $b_{i_k} = b_{j_k} \quad \forall i, j \in \{1, \dots, n\} \quad \forall k \in \{1, 2, 3, 4\}$  aufgrund der identischen Dichtefunktionen der erzeugten Zufallsvektoren gilt. Die nachfolgenden Integrale wurden in einem ersten Schritt mittels des obenstehenden Programmcodes, welcher die Monte-Carlo-Methode implementiert, berechnet. In einem zweiten Schritt wurden die vier Integrale mithilfe des Programms Mathematica gelöst, um Vergleichswerte für die ungefähre Genauigkeit der in Python programmierten Monte-Carlo-Methode zu erhalten. Die zu berechnenden Integrale sind

$$\text{Integral 1: } I_1 = \int_{a_{i_1}}^{b_{i_1}} e^{-2x_{i_0}^2} dx_{i_0} = \int_{-1}^3 e^{-2x^2} dx, \quad (34)$$

$$\text{Integral 2: } I_2 = \int_{a_{i_1}}^{b_{i_1}} \int_{a_{i_2}}^{b_{i_2}} e^{-2(x_{i_0}^2 + x_{i_1}^2)} dx_{i_0} dx_{i_1} = \int_{-1}^3 \int_1^5 e^{-2(x^2 + y^2)} dx dy, \quad (35)$$

$$\begin{aligned} \text{Integral 3: } I_3 &= \int_{a_{i_1}}^{b_{i_1}} \int_{a_{i_2}}^{b_{i_2}} \int_{a_{i_3}}^{b_{i_3}} e^{-2(x_{i_0}^2 + x_{i_1}^2 + x_{i_2}^2)} dx_{i_0} dx_{i_1} dx_{i_2} \\ &= \int_{-1}^3 \int_1^5 \int_{-1}^1 e^{-2(x^2 + y^2 + z^2)} dx dy dz \quad \text{und} \end{aligned} \quad (36)$$

$$\begin{aligned} \text{Integral 4: } I_4 &= \int_{a_{i_1}}^{b_{i_1}} \int_{a_{i_2}}^{b_{i_2}} \int_{a_{i_3}}^{b_{i_3}} \int_{a_{i_4}}^{b_{i_4}} e^{-2(x_{i_0}^2 + x_{i_1}^2 + x_{i_2}^2 + x_{i_3}^2)} dx_{i_0} dx_{i_1} dx_{i_2} dx_{i_3} \\ &= \int_{-1}^3 \int_1^5 \int_{-1}^1 \int_{-18}^7 e^{-2(x^2 + y^2 + z^2 + u^2)} dx dy dz du. \end{aligned} \quad (37)$$

Der in Mathematica implementierte Code zum Vergleichen der mittels obenstehendem Python-Code erhaltenen Resultate lautet wie folgt:

```
SetAttributes[nameAndValue, HoldFirst];
nameAndValue[x_] := Print[Unevaluated[x], "=", x];

Subscript[Integral, 1] = N[Integrate[E^(-2*(x^2)), {x, -1, 3}]];
nameAndValue[Subscript[Integral, 1]]

Subscript[Integral, 2] = N[Integrate[E^(-2*(x^2 + y^2)), {x, -1, 3}, {y,
  ↪ 1, 5}]];
nameAndValue[Subscript[Integral, 2]]

Subscript[Integral, 3] = N[Integrate[E^(-2*(x^2 + y^2 + z^2)), {x, -1,
  ↪ 3}, {y, 1, 5}, {z, -1, 1}]];
nameAndValue[Subscript[Integral, 3]]

Subscript[Integral, 4] = N[Integrate[E^(-2*(x^2 + y^2 + z^2 + u^2)), {x,
  ↪ -1, 3}, {y, 1, 5}, {z, -1, 1}, {u, -18, 7}]];
nameAndValue[Subscript[Integral, 4]]
```

### 5.1.2 Resultate

Der Python Code liefert folgende Ergebnisse:

```
Integral_1 is I = 1.2213193885584246
Integral_2 is I = 0.03502552728359172
```

Integral\_3 is I = 0.041539963653965  
Integral\_4 is I = 0.05096407733798501

Die mit Mathematica berechneten Vergleichswerte ergeben folgendes:

Subscript[Integral, 1] = 1.2248  
Subscript[Integral, 2] = 0.0349228  
Subscript[Integral, 3] = 0.0417778  
Subscript[Integral, 4] = 0.0523607

Durch einen Vergleich der beiden Ergebnisse ist ersichtlich, dass die implementierte Monte-Carlo-Methode zur Berechnung von hochdimensionalen Integralen funktionieren sollte. Wählt man höhere Zahlen für  $n$ , als es im vorliegenden Beispiel getan wurde, nimmt die für die Prozessierung des Programms benötigte Zeit massiv zu. Mit genügenden Rechenressourcen kann man aber, wie es im Theorieteil dargelegt wurde, ein beliebiges Integral im Prinzip beliebig genau berechnen, indem man  $n$  so wählt, dass damit die gewünschte Genauigkeit erreicht wird.

## 6 Literatur

[Molchanov, 2017] Molchanov, I. (2017). *Kombinatorik und Wahrscheinlichkeitsrechnung: Skript zur Vorlesung*. Universität Bern, Bern.