

Die Monte Carlo Methode



Abschlussbericht der Kooperationsphase 2016/17

Durchgeführt an der Fakultät für Mathematik und Informatik
Universität Mannheim

Betreuer: Dr. Peter Parczewski

Christopher Schön
Peter-Schnellbach-Str. 10/1
69151 Neckargemünd

Xaver Kohlbrenner
Dammstraße 18
68549 Ilvesheim

Inhaltsverzeichnis

Abstract	1
1 Einleitung	2
2 Theoretische Grundlagen	3
2.1 Das starke Gesetz der großen Zahlen	3
2.2 Das Buffonsche Nadelexperiment	7
3 Ergebnisse	10
3.1 Mathematische Probleme und Programme	10
3.1.1 A Courious Coin Flipping Game	10
3.1.2 How Many Runners in a Marathon?	13
3.1.3 A Parallel Parking Question	15
4 Diskussion	17
4.1 Auswertung der Ergebnisse	17
5 Danksagung	18
6 Quellen	18
7 Anhang	19
7.1 Buffonsches Nadelexperiment	19
7.2 A Courious Coin Flipping Game	20
7.3 How Many Runners in a Marathon	21
7.4 A Parallel Parking Question	22
8 Selbstständigkeitserklärung	23

Abstract

There are mathematical problems that cannot practically be solved via calculation or analysis. If those problems are connected with random processes stochastics can provide a possible solution approach with the Monte Carlo method.

The Monte Carlo method uses computational algorithms that make use of large sequences of randomized input to calculate a numeric result. The numeric result is an approximation of the expected value of a random variable. After the definition of a domain for possible input values those values are randomly generated based on a probability distribution. A subject based on computation on this inputs is executed and the results are aggregated.

Nowadays the Monte Carlo method is used widely in different areas such as engineering, physical sciences, climate change models, artificial intelligence, financial risk evaluations etc.

As a prerequisite for the theoretical analysis and understanding of Monte Carlo type problems stochastic theory was explored with emphasis on the strong law of large numbers.

Modern mathematics makes use of tools for simple scripting and computation for Monte Carlo method like problems such as Matlab and GNU Octave which were used in this project.

1 Einleitung

Ist er vorstellbar mit Zufallsversuchen zu konkreten Ergebnissen zu gelangen?
Dieser Fragestellung widmeten sich im Laufe der vergangenen Jahrhunderte zahlreiche Wissenschaftler. Es scheint einleuchtend, dass je häufiger man eine Münze wirft, man umso genauer auf die Wahrscheinlichkeit des Auftretens von Kopf oder Zahl schließen kann.

Die Idee Zufallsmethoden in der Mathematik in einem deterministischem Sinn zu verwenden erwies sich als revolutionär und hat dazu geführt, dass Monte Carlo Methoden heutzutage in so verschiedenen Bereichen wie der Erstellung komplexer Wettermodelle bis zur Beschreibung von Prozessen in Kernreaktoren Anwendung finden.

Der erste dokumentierte Monte Carlo Versuch geht auf den französischen Naturforscher Comte de Buffon zurück. Im 18. Jahrhundert gelang ihm die Approximation der Kreiszahl Pi mithilfe seines berühmten Nadelexperimentes.

Anfänglich verwendeten Wissenschaftler die Monte Carlo Methode bei Problemen, die bereits analytisch verstanden und gelöst waren. Die moderne Simulation kehrt den Prozess um und findet Anwendung bei analytisch noch nicht gelösten Problemen.

Diese Form der Simulation wurde zum ersten Mal systematisch im Rahmen des Manhattan Projekts während des zweiten Weltkriegs verwendet. Die Mathematiker Stanislaw Ulam, John von Neumann und Nicholas Metropolis erweiterten die theoretischen Grundlagen des Verfahrens so, dass sie mithilfe der Monte Carlo Simulation die Dicke von Wänden, die benötigt wurden um Kernzerfallsexperimente abzuschirmen, ermitteln konnten. Als militärisches Projekt unterlag ihr Unternehmen der Geheimhaltung, sodass sie ihr Verfahren in dem Wissen um die Rolle des Zufalls nach der bekannten Spielbank Monte Carlo benannten.

Die Monte Carlo Methode wird häufig als wissenschaftliches Herangehen angewandt, wenn Experimente zu teuer, zu zeitaufwändig oder nicht praktikabel sind, beziehungsweise wenn ein analytisches Vorgehen nicht möglich ist. Der Nutzen ist besonders groß bei komplexen Systemen mit vielen Inputvariablen.

Einschränkungen der Methode sind der erforderliche hohe Rechenaufwand, die Tatsache, dass nur ein Näherungswert ermittelt werden kann und die Abhängigkeit der Qualität der Lösung vom verwendeten Modell.

2 Theoretische Grundlagen

2.1 Das starke Gesetz der großen Zahlen

Das starke Gesetz der großen Zahlen gehört zusammen mit dem schwachen Gesetz der großen Zahlen zu den klassischen Grenzwertsätzen der Stochastik. Das Gesetz ist elementar für das Verständnis der Monte Carlo Methode. Es trifft eine Aussage darüber, wann eine Folge von normierten Zufallsvariablen gegen einen konstanten Wert, meist den Erwartungswert, konvergiert.

Schwaches und starkes Gesetz der großen Zahlen unterscheiden sich durch die betrachteten Konvergenzarten. Bei dem starken Gesetz handelt es sich um die fast sichere Konvergenz – die strengste Konvergenz in der Stochastik.

Das starke Gesetz der großen Zahl besagt, dass mit zunehmender Anzahl an Iteration die Abweichung vom Erwartungswert gegen Null konvergiert.

Satz:

Seien $(X_i)_{i \geq 1}$ paarweise unkorrelierte Zufallsvariablen in L^2 mit $v := \sup_{i \geq 1} V(X_i) < \infty$.

Dann gilt:

$$\frac{1}{n} \sum_{i=1}^n (X_i - \mathbb{E}[X_i]) \rightarrow 0 \text{ fast sicher.}$$

Dieser Satz besagt, dass je mehr X_i realisiert werden, desto mehr konvergiert das Ergebnis gegen 0. „Fast sicher“ bedeutet in diesem Zusammenhang, dass die Wahrscheinlichkeit des Eintreffens 1 ist.

Oberer Satz könnte auch wie folgt ausgedrückt werden:

$$\begin{aligned} P(\omega : \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n (X_i(\omega) - \mathbb{E}[X_i]) = 0) &= 1 \\ &= P[\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n (X_i - \mathbb{E}[X_i]) = 0] = 1 \end{aligned}$$

Beweis des Satzes:

Es wird $\mathbb{E}[X_i] = 0$ ohne Einschränkung angenommen, also wird $X_i^0 := X_i - \mathbb{E}[X_i]$ betrachtet.

Weiter kann angenommen werden $\mathbb{E}[X_i^0] = \mathbb{E}[X_i - \mathbb{E}[X_i]]$ aufgrund der Linearität des Erwartungswerts¹ $\mathbb{E}[X_i] - \mathbb{E}[\mathbb{E}[X_i]] = \mathbb{E}[X_i] - \mathbb{E}[X_i] = 0$.

Außerdem wird gesetzt: $Y_n = \frac{1}{n} \sum_{i=1}^n X_i$

¹ Götz, Kerstin: Elementare Stochastik, S.50, o.O. 2010.

Die weitere Beweisführung erfolgt in zwei Teilen. Im ersten Teil wird gezeigt, dass Y_{n^2} fast-sicher gegen Null konvergiert. Im zweiten Teil wird die Aussage des Satzes auf die Teilfolge im ersten Teil zurückgeführt.

Teil 1:

Es wird gezeigt, dass $Y_{n^2} \rightarrow 0$ fast sicher ist. Durch das Schwache Gesetz der großen Zahlen ist $P(|Y_{n^2}| > \varepsilon) \leq \frac{v}{n^2 \varepsilon^2}$ gegeben für alle $\varepsilon > 0$ ²:

Diese Wahrscheinlichkeiten bilden die konvergente Reihe $\sum_{n=1}^{\infty} P(|Y_{n^2}| > \varepsilon) < \infty$

Beweis nach Leonhard Euler:

$$\sum_{n=1}^{\infty} P(|Y_{n^2}| > \varepsilon) \leq \sum_{n=1}^{\infty} \frac{v}{n^2 \varepsilon^2} = \frac{v}{\varepsilon^2} \sum_{n=1}^{\infty} \frac{1}{n^2} < \infty$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{v}{\varepsilon^2} \cdot \frac{\pi^2}{6}$$

Da der vorige Term geringer als die Konstante ist, muss er konvergieren.

Nun kann das Lemma von Borel-Cantelli auf unsern Term $P(|Y_{n^2}| > \varepsilon)$ angewendet werden:

$$P(\lim_{n \rightarrow \infty} |Y_{n^2}| > \varepsilon) \leq P(|Y_{n^2}| > \varepsilon \text{ für unendlich viele } n) = 0$$

Borel-Cantelli besagt, wenn $\sum_{n \geq 1} P(A_n) < \infty$, so ist $P(A) = 0$. Es ist $A_n = |Y_{n^2}| > \varepsilon$.

$P(A)$ ist hierbei $\lim_{n \rightarrow \infty} A_n = [\omega : |Y_{n^2}| > \varepsilon \text{ der für unendlich viele } n \geq 1 \text{ ist}]$.

Es wird jetzt eine zweite Menge B in den Limes Supremum für Y_{n^2} eingesetzt.

Daraus folgt $P(\omega : \lim_{n \rightarrow \infty} |Y_{n^2}(\omega)| > \varepsilon)$, da für alle $\omega \in B \Rightarrow \omega \in A$ gilt und der Limes Supremum von Y_{n^2} nur kleiner-gleich Y_{n^2} für unendlich viele n sein kann.

Daher gilt:

$$P(\omega : \lim_{n \rightarrow \infty} |Y_{n^2}(\omega)| > \varepsilon) \leq (\lim_{n \rightarrow \infty} A_n = [\omega : |Y_{n^2}| > \varepsilon \text{ für unendlich viele } n]) = 0$$

Es ist nun bekannt, dass die Wahrscheinlichkeit, dass der Limes Supremum von Y_{n^2} größer als ε ist, gleich Null ist.

Bezugnehmend auf Satz(1.11(e))³, der hier in $P(A) = \lim_{n \rightarrow \infty} P(A_n)$ vorliegt und unter Berücksichtigung, dass Limes $\varepsilon \rightarrow 0$ gilt wird gefolgert:

² ε : eine beliebig kleine Zahl größer Null

³ Georgii, Hans-Otto: Stochastik, Einführung in die Wahrscheinlichkeitstheorie und Statistik, 2015, S.16

$$P(Y_n^2 \not\rightarrow 0) = P(\lim_{n \rightarrow \infty} |Y_n^2| > 0) = \lim_{\varepsilon \rightarrow 0} P(\lim_{n \rightarrow \infty} |Y_n^2| > \varepsilon) = 0$$

Die Wahrscheinlichkeit, dass Y_n nicht gegen 0 konvergiert ist also

$$P(\lim_{n \rightarrow \infty} |Y_n^2| > 0) = \lim_{n \rightarrow \infty} P(A_n) = P(A) = P(|Y_n^2| > 0)$$

0) = für unendlich viele n).

Nach Einfügen des Limes von ε erhält man wie behauptet Null.

Teil 2:

$$m \in \mathbb{N} \text{ sei } n = n(m)$$

Im zweiten Schritt wird gesetzt.

$$P(|Y| \geq (\varepsilon n^2)^2) \leq \frac{(\quad)}{\varepsilon^2}$$

Es werden die Differenzen zwischen den n Potenzen und

betrachtet. Weiterhin wird betrachtet, ob die Differenzen gegen Null konvergieren.

Es werden n und m so gewählt, dass $n^2 \leq m < (n+1)^2$ gilt.

Y_m und Y_{n^2} werden verglichen und eingesetzt. Aus der Cebysev-Ungleichung⁴ folgt nun

$$S_k = kY_k = \sum_{i=1}^k X_i \quad \text{und}$$

$$P(|S_m - S_n| > \varepsilon n^2) \leq (\varepsilon^{-1} n^{-2})^2 \sum_{i=n^2+1}^m X_i \leq \frac{v(m - n^2)}{\varepsilon^2 n^4}$$

Aufgrund der Ungleichung im ersten Teil des Beweises ist $\varepsilon^{-2} n^{-4}$ gleich $(\varepsilon n^2)^2$.

Zudem ist $|S_m - S_n| > \varepsilon n^2$.

$$Y = S_m - S_n = \sum_{i=1}^m X_i - \sum_{i=1}^{n^2} X_i = \sum_{i=n^2+1}^m X_i \quad \text{wird für } V(Y_i) \text{ eingesetzt.}$$

Der Erwartungswert von Y_i ist weiterhin Null.

$$\mathbb{E}(S_m - S_n) = \mathbb{E}(\sum_{i=n^2+1}^m X_i) = \sum_{i=n^2+1}^m \mathbb{E}(X_i) = 0$$

Der Index m läuft gegen Unendlich:

$$\sum_{m \geq 1} P(|S_m - S_{n(m)}| > \varepsilon n(m)^2) \leq \frac{v}{\varepsilon^2} \sum_{n \geq 1} \sum_{m=n^2}^{(n+1)^2-1} \frac{m-n^2}{n^4}$$

$$= \frac{v}{\varepsilon^2} \sum_{n \geq 1} \sum_{k=1}^{2n} \frac{k}{n^4} = \frac{v}{\varepsilon^2} \sum_{n \geq 1} \frac{(2n)(2n+1)}{2n^4} < \infty$$

Es ist $k = m - n^2$. Es handelt sich um eine Indexverschiebung, folglich ist $n^2 \leq k \leq m^2$.

Zudem ist nach der Summenformel

$$\sum_{k=1}^{2n} \frac{k}{n^4} = \frac{1}{n^4} (\sum_{k=1}^{2n} k) = \frac{1}{n^4} * \frac{1}{2} 2n(2n+1)$$

Der Term ist kleiner Unendlich. Beispiel $2n + 1 \leq 3n$:

⁴ ebd. S.122

$$\sum_{n \geq 1} \frac{2n(2n+1)}{2n^4} \leq 3 \sum_{n \geq 1} \frac{n^2}{n^4} < \infty$$

Es wird umgeformt und anschließend das Lemma von Borel-Cantelli angewendet, da $\sum_{m \geq 1} P(|S_m - S_{n(m)}| > \varepsilon n(m)^2) < \infty$.

Umformung:

$$P(|S_m - S_{n(m)}| > \varepsilon n(m)^2) = P\left(\left|\frac{S_m}{n(m)^2} - \frac{1}{n(m)^2} S_{n(m)}\right| > \varepsilon\right) = P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| > \varepsilon\right)$$

Es gilt wie im ersten Teil des Beweises:

$$P\left(\lim_{m \rightarrow \infty} P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| > \varepsilon n(m)^2\right)\right) \leq P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| > \varepsilon n(m)^2 \text{ für } \infty \text{ viele } m\right)$$

Anwendung des Lemma von Borel-Cantelli und Limes $\varepsilon \rightarrow \infty$:

$$P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| \not\rightarrow 0\right) = P\left(\lim_{m \rightarrow \infty} P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| > 0\right)\right) = \lim_{\varepsilon \rightarrow 0} P\left(\lim_{n \rightarrow \infty} P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| > \varepsilon\right)\right) = 0$$

Es folgt

$$P\left(\left|\frac{S_m}{n(m)^2} - Y_{n(m)}\right| \xrightarrow{m \rightarrow \infty} 0\right) = 1$$

Zudem folgt aus $|Y_m| = \frac{|S_m|}{m} \leq \frac{|S_m|}{n(m)^2}$ die Aussage $P(Y_m \rightarrow 0) = 1$, da $n^2 \leq m$ ist.

q.e.d. ■

2.2 Das Buffonsche Nadelexperiment

Georges Louis Leclerc, Comte de Buffon, stellte sein später berühmt gewordenes Nadelproblem erstmals 1733 an der Pariser Akademie der Wissenschaften vor. Aus der allgemeinen Wahrscheinlichkeitsbetrachtung wie häufig willkürlich geworfene Nadeln gleicher Länge parallele Linien mit identischen Abständen schneiden, entwickelt er ein Verfahren zur Approximation der Kreiszahl Pi.

Der Überlieferung nach soll er seine Theorie überprüft haben, indem er ein Baguette mehrfach über seine Schulter auf den gefliesten Fußboden warf.

Man stelle sich eine Fläche vor, die von parallelen Linien des gleichen Abstandes d durchzogen ist und werfe eine Nadel der Länge l ($l \leq d$) auf diese Fläche. Es soll die Wahrscheinlichkeit ermittelt werden, mit der die Nadel eine Linie schneidet.

Dabei kann zwischen zwei grundlegenden Herangehensweisen unterschieden werden. Zum einen kann die Wahrscheinlichkeit analytisch durch Ermittlung eines Integrals bestimmt werden, zum anderen kann sie durch die Monte Carlo Methode approximiert werden.

Für die analytische Herangehensweise ist zu definieren, was zufälliges Werfen bedeutet. Die Position einer Nadel ist definiert durch ihren Mittelpunktsabstand y zur nächsten parallelen Linie sowie den Winkel α zwischen Nadel und der Parallelen p durch den Nadelmittelpunkt. Es leuchtet ein, dass die Domain von $y = [0; d/2]$ ist und die Domain von $\alpha = [0; \pi/2]$. Der zufällige Wurf einer Nadel entspricht folglich der gleichverteilten Auswahl eines Tupels $(y; \alpha)$ aus dem Wahrscheinlichkeitsraum $\Omega = (0; d/2) \times (0; \pi/2)$.

Nun wird betrachtet, welche Bedingung erfüllt sein muss, damit eine Nadel eine Linie schneidet. Die Nadel schneidet eine Linie, wenn der Abstand y kleiner-gleich des Abstandes A zwischen der Parallelen p und dem Endpunkt der Nadel ist. Der Abstand A ist $\sin \alpha \times l/2$, beweisbar durch die Anwendung des Sinussatzes.

Die Menge der Tupel, auf die genannte Bedingung zutrifft ist folglich gleich

$$\int_0^{\pi/2} \sin \alpha \, d\alpha = l/2.$$

Die Wahrscheinlichkeit für das Eintreten des Ereignisses „Nadel schneidet Linie“ ergibt sich aus der Division „Menge der Tupel für die Bedingung gilt“

durch „Menge aller Tupel“, also $P(E) = \frac{l/\pi}{d/2} \times \frac{d}{2} = 2l/\pi d$.

Durch Umformung nach π erhält man $\pi = 2l/d \times P(E)$.

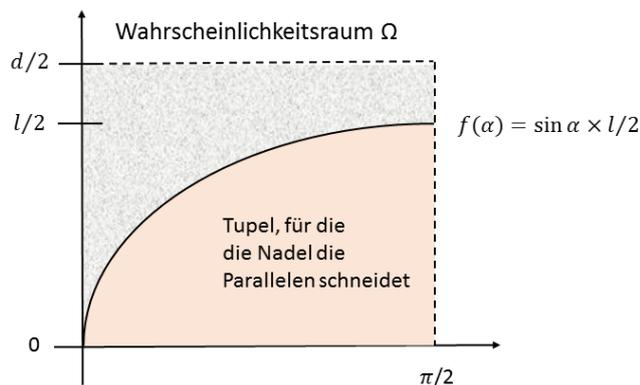
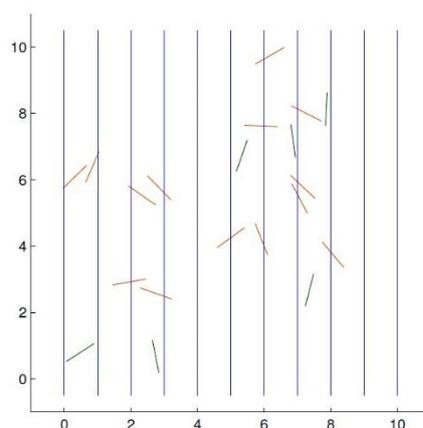


Abbildung 1: Analytische Herangehensweise

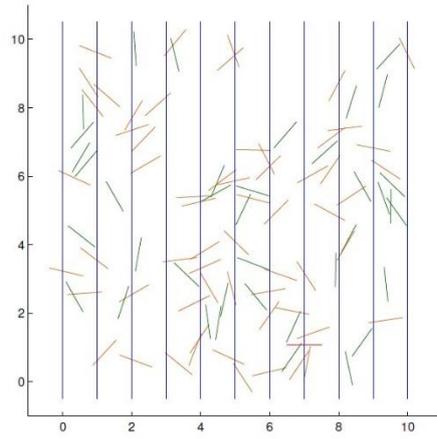
Die grundlegende Idee bei Verwendung der Monte Carlo Simulation ist die Approximation von $P(E)$. Hierzu wurde ein Programm in Matlab geschrieben, welches den Nadelwurf simuliert. Zunächst wird eine großen Anzahl n von *Tupeln* $(y; \alpha)$ generiert, jedes Tupel ist das Analog einer geworfenen Nadel. Die Tupel werden anschließend auf die oben erläuterte Bedingung $y \leq \sin \alpha \times l/2$ geprüft. Die Tupel auf die die Bedingung zutrifft werden gezählt, ihre Menge sei m . Je größer die Anzahl n der generierten Tupel ist, desto genauer approximiert $m/n \approx P(E)$.

Im Programm wird m/n für $P(E)$ in $\pi = 2l/d \times P(E)$ eingesetzt und man erhält die gesuchte Approximation von π .

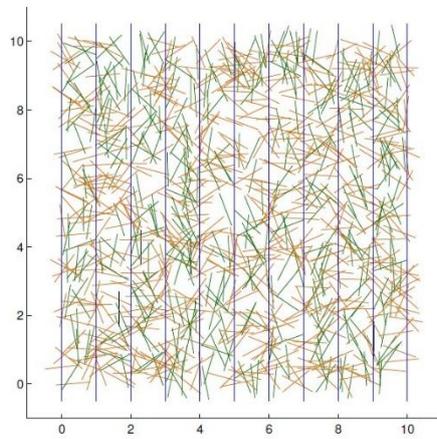
Graphische Abbildung eines praktischen Versuchs unter Verwendung von n Nadeln:



$n = 20, \pi \approx 2.857142857142857$



$$n = 100, \pi \approx 3.225806451612903$$



$$n = 1000, \pi \approx 3.241491085899514$$

Abbildung 2-4: Buffonsches Nadelexperiment

3 Ergebnisse

3.1 Mathematische Probleme und Programme

Die folgenden Probleme wurden ausgewählt, um an ihnen die Anwendung der Monte Carlo Methode zu demonstrieren. Dabei wurden die bei der Bearbeitung des Buffonschen Nadelexperiments gewonnenen Erfahrungen und Erkenntnisse auf bis dato im Projekt unbekannte Problemkreise angewendet. In einem ersten Schritt ging es um die Analyse des Problems und mögliche theoretische Lösungsansätze. Das theoretische Verständnis des Problems wurde verifiziert, indem der antizipierte Erwartungswert mithilfe von mit GNU Octave erstellten Skripten ermittelt wurde.

GNU Octave ist eine kostenlose Alternative zum dem zu Grunde liegenden kommerziellen Programm Matlab und basiert auf denselben quelloffenen Bibliotheken LAPACK und BLAS. GNU Octave wird vorrangig zur numerischen Lösung von Problemen, insbesondere numerischer Simulation, Datenerfassung und Datenanalyse verwendet.

3.1.1 A Curious Coin Flipping Game

In diesem Problem wird ein einfaches Münzspiel beschrieben:

Drei Personen bekommen eine bestimmte Anzahl an Münzen, diese können gleich oder unterschiedlich verteilt sein. Die Münzen haben zwei verschiedene Seiten, Kopf und Zahl. Die Spieler werfen jeweils eine Münze. Die Person, deren Münze eine andere Seite als die der restlichen Spieler zeigt, gewinnt alle drei Münzen. Bei einem Unentschieden wird die Runde wiederholt. Es wird so lange gespielt bis eine Person keine Münzen mehr besitzt. Wie viele Runden werden im Durchschnitt benötigt bis eine Person keine Münzen mehr besitzt?

Dieser Sachverhalt soll durch ein entsprechende Monte Carlo Simulation abgebildet werden. Das Programm simuliert zunächst den Spielverlauf einer Partie. Die für die Partie notwendige Anzahl an Runden wird gemerkt. Der Verlauf wird für k Spielpartien simuliert und die Anzahl der jeweils erforderlichen Runden summiert und anschließend durch k geteilt zur Ermittlung des Mittelwerts.

Im Einzelnen stellt sich der Programmablauf wie folgt dar.

Definiert werden l , m und n , die die einzelnen Spieler und die ihnen zugeordnete Anzahl von Münzen repräsentieren. Eine Wahrscheinlichkeit p für das Auftreten von Kopf oder Zahl wird beliebig im Intervall $[0; 1]$ gesetzt.

Bei jeder Iteration wird zunächst für jeden Spieler die Anzahl der Münzen um 1 reduziert. Es wird pro Spieler eine Zufallszahl in dem Intervall $[0; 1]$ erzeugt und mit p verglichen. Ist die Zufallszahl kleiner p wird dem jeweiligen Spieler 0 zugeordnet, ist sie größer wird 1 zugeordnet. Die den Spielern zugeordneten Werte werden addiert, wodurch eine Zahl zwischen 0 und 3 entsteht. Bei Summe 0 bzw. 3 haben alle Spieler Kopf bzw. Zahl geworfen, jeder bekommt seine Münze zurück und die nächste Runde beginnt. Ist die Summe 1 bzw. 2 haben zwei Personen die gleiche Seite geworfen und die Person, die die andere Seite geworfen hat bekommt alle 3 Münzen.

(l, m, n)	Rundenzahl mit $p = 0,5$ (faire Münze)	Rundenzahl mit $p = 0,4$ (unfaire Münze)
2,2,2	2,6638	2,7814
1,2,3	1,9995	2,0820
3,3,3	5,1510	5,3512
2,7,9	10,583	10,783

Ist die Gesamtzahl von Münzen durch 3 teilbar, dann führt eine Gleichverteilung der Münzen zu einem maximalen Erwartungswert für die Anzahl der Runden. Werden unfaire Münzen verwendet, erhöht sich der Erwartungswert für die benötigten Runden, da es häufiger zu einem Unentschieden kommt.

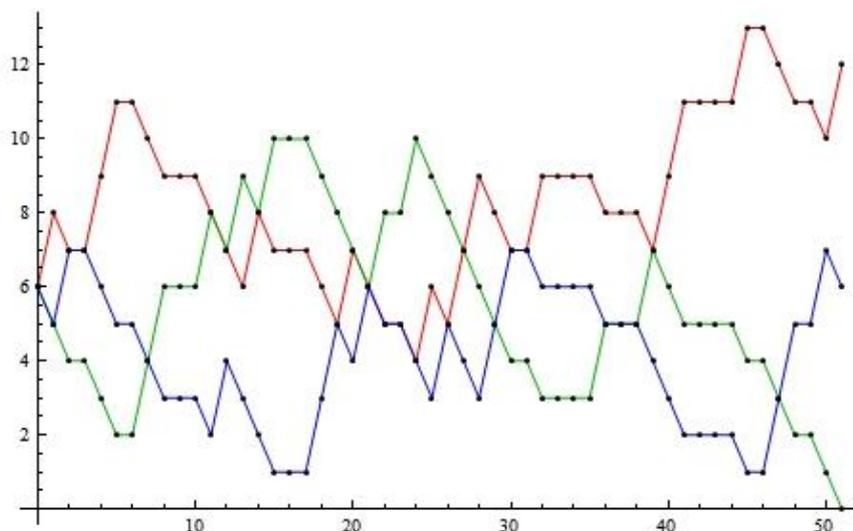


Abbildung 5: Exemplarische Durchführung mit jeweils 6 Münzen

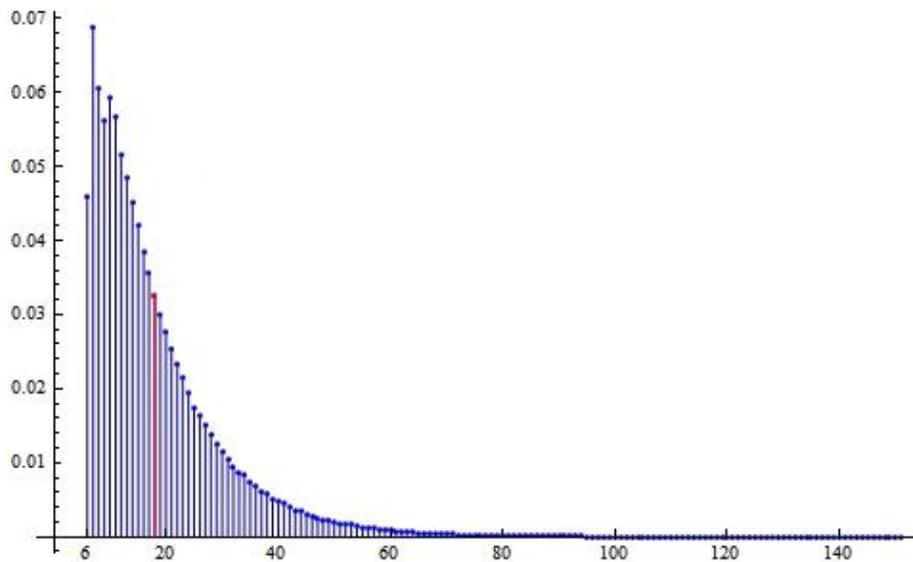


Abbildung 6: Verteilung der relativen Häufigkeiten der Spieldauer nach Runden;
 10^6 Durchführung mit jeweils 6 Münzen; Erwartungswert 18

Die theoretische Analyse für den Fall $l = m = n = 1$ und beliebige Wahrscheinlichkeit p für das Eintreffen von Kopf ist folgende.

$E(l, m, n)$ sei die erwartete Anzahl an Runden des Spiels.

Die Wahrscheinlichkeit, dass alle Spieler das gleiche Ergebnis beim Münzwurf haben ist $p^3 + (1 - p)^3$. In diesem Fall ist $E(l, m, n)$ unverändert, da sich die Münzverteilung der Spieler nicht geändert hat.

Die Wahrscheinlichkeit, dass ein Spieler die Runde gewinnt ist $p(1 - p)^2 + (1 - p)p^2 = p(1 - p)$. In diesem Fall ist $E(l + 2, m - 1, n - 1)$ falls der Gewinner l ist. Entsprechend gestaltet sich $E(\text{neu})$ wenn m oder n gewinnt. Folglich:

$$E(l, m, n) = [1 + E(l, m, n)][p^3 + (1 - p)^3] + [1 + E(l + 2, m - 1, n - 1)]p(1 - p) + [1 + E(l - 2, m + 2, n - 1)]p(1 - p) + [1 + E(l - 2, m - 1, n + 2)]p(1 - p)$$

Es leuchtet ein, dass $E(l, m, n) = 0$ wenn l, m oder n gleich 0 ist, da das Spiel in diesem Fall beendet ist. Folglich vereinfacht sich oberhalb aufgeführter $E(l, m, n)$ im Falle $l = m = n = 1$ zu: $E(1,1,1) = [1 + E(1,1,1)][p^3 + (1 - p)^3] + 3p(1 - p)$

Durch Umformung erhält man: $E(1,1,1)(3p - 3p^2) = 1 - 3p + 3p^2 + 3p(1 - p) = 1$.
 Damit ergibt sich: $E(1,1,1) = 1/[3p(1 - p)]$.

Die gewonnene Formel für den Erwartungswert sei für den Fall der Verwendung einer

fairen Münze ($p = 0,5$) und der Ausgangssituation, dass alle Spieler genau 1 Münze besitzen betrachtet: $E(1,1,1/p = 0,5) = 1/[3 \times 0,5(1 - 0,5)] = 4/3$.

3.1.2 How Many Runners in a Marathon?

Das folgende Problem hat seinen Ursprung im zweiten Weltkrieg. Damals wollte die Aufklärung der Alliierten erfahren wie viele Panzer, Flugzeuge und anderer Waffen der Gegner produziert. Hierzu machten sie sich die Seriennummern der Waffensysteme zu Nutze, um deren Gesamtmenge zu schätzen.

Mit dieser Herangehensweise kann man z.B. auch das „Läufer bei einem Marathon“ Problem lösen, bei dem man als Beobachter anhand der Trikotnummern der vorbeilaufenden Läufer herausfinden soll, wie viele Läufer an einem Marathon teilnehmen.

Gegeben sei eine unbekannte Anzahl N von Personen, die von 1 bis N durchnummeriert sind, dabei ist N die Gesamtzahl der Teilnehmer am Marathon. Der Beobachter selektiert in einem ersten Schritt eine Teilmenge von n Läufern aus der Gesamtzahl N .

In der folgenden Formel, die gegeben ist, wird nun die Anzahl der selektierten Personen n und die höchste gesehene Läufernummer $E(\max X_i)$ eingesetzt.

$$\mathbb{E}(\max X_i) = \frac{n(N+1)}{n+1}$$

Die Formel wird nach N , der Gesamtzahl der Läufer, umgestellt.

$$N = \frac{n+1}{n} E(\max X_i) - 1$$

Erläuterung der Formel anhand eines Beispiels:

Der Beobachter sieht 50 Personen an sich vorbeilaufen, die höchste Trikotnummer ist 423.

$$N \approx \frac{51}{50} \cdot 423 - 1 = 430,46$$

Basierend auf der Formel lassen sich zwei weitere Aussagen treffen:

Das kleinste mögliche Ergebnis bei einer Menge n ist n , das Ergebnis für N kann folglich nie kleiner n sein:

$$\frac{n+1}{n} - 1 = n$$

Wenn n gleich N ist, alle teilnehmenden Läufer den Beobachter passiert haben, gilt:

$$N = (N + 1/N) \times N - 1.$$

Trotzdem kann das Ergebnis der Formel stark von der tatsächlichen Gesamtzahl abweichen. Anhand eines konkreten Beispiels soll dies transparent gemacht werden. Angenommen wird, dass insgesamt 800 Läufer am Rennen teilnehmen, von denen der Beobachter 5 Läufer einbezieht. Gesetzt den Fall, dies sind die 5 Läufer mit den ersten 5 Startnummern. Daraus ergäbe sich als Schätzwert $N \approx 5$ und somit ein stark abweichendes Ergebnis:

$$\frac{5+1}{5} \cdot 5 - 1 = 5$$

Nun wird angenommen, dass der Beobachter statt der Läufer mit den ersten 5 Startnummern die Läufer mit den letzten 5 Startnummern sieht.

Daraus ergäbe sich als Schätzwert $N \approx 959$ und damit ein zu hohes Ergebnis:

$$\frac{5+1}{5} \cdot 800 - 1 = 959$$

Um die Abweichung genauer analysieren zu können wird die Monte Carlo Methode benutzt.

So lässt sich die durchschnittliche Abweichung des Schätzwertes von der Gesamtzahl N ermitteln, abhängig vom prozentualen Anteil der beobachteten Läufer n an der Gesamtmenge N .

Das Programm dazu läuft in folgenden Hauptschritten ab.

Zunächst wird die Gesamtzahl der Läufer N auf einen beliebigen Wert gesetzt. Ebenso wird die Anzahl der beobachteten Läufer auf einen Wert n gesetzt, der ein bestimmter Prozentsatz von N ist, exemplarisch 2%, 5%, 10% und 20%. Anschließend werden n ganze Zufallszahlen im Intervall $[1; N]$ erzeugt. Die größte der erzeugten Zufallszahlen wird in die Formel $N(\text{Schätzung}) = n/n + 1 \times \text{Max } X_n$ eingesetzt. Die Differenz $N - N(\text{Schätzung})$ wird berechnet und der absolute Wert ABS gebildet. Der Prozess wird k mal durchgeführt und die dabei ermittelten ABS summiert. Die Summe wird durch die Anzahl der Iterationen k dividiert. Der so ermittelte Wert wird durch N geteilt und stellt die prozentuale Abweichung der Schätzung bei gegebenem n von der Gesamtzahl N dar. Zusätzlich erzeugt das Programm ein Histogramm, welches die Streuung der Abweichung darstellt.

Bei Ausführung des Programms mit den exemplarisch definierten Prozentsätzen $n = \{2; 5; 10; 20\} \% \times N$ konnte festgestellt werden, dass die relative Abweichung des Schätzwertes von der tatsächlichen Anzahl N umso geringer war, je höher der Prozentsatz für die Festlegung der zu beobachtenden n gewählt wurde. Folgende Grafik zeigt die Streuung der Abweichung in Abhängigkeit von der Größe der untersuchten Teilmenge.

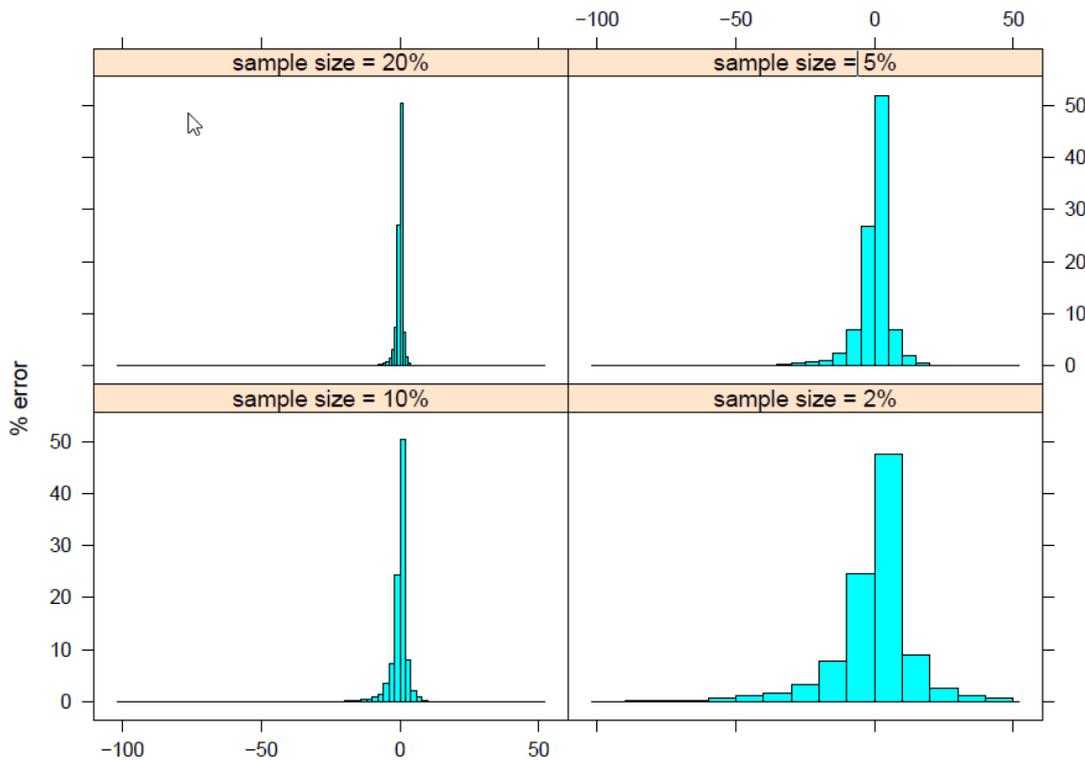


Abbildung 7: Streuung der Abweichung in Abhängigkeit vom prozentualen Anteil der betrachteten Untermenge n an der Gesamtmenge N

3.1.3 A Parallel Parking Question

Basierend auf einer Alltagssituation wird hier ein mathematisches Problem formuliert. Autos stehen dicht an dicht auf Parkplätzen, parken parallel zueinander und haben unterschiedliche Abstände voneinander.

Es werden zwei nebeneinanderstehende Wagen als „Paar“ definiert, wenn ihr Abstand zueinander geringer ist als der zu ihrem anderen Nachbarn. Zur Vereinfachung werden die Autos als eindimensionale Punkte auf einer Strecke betrachtet. Ohne Verlust der Allgemeingültigkeit kann diese Strecke auf die Länge 1 normiert werden.

Die Frage ist nun: Mit welcher Wahrscheinlichkeit ist ein zufällig ausgewähltes Auto Teil eines Paares?

Zur Verdeutlichung des Sachverhalts soll folgendes Beispiel dienen. Man betrachte den Fall für 3 Autos. In diesem Fall bildet das mittlere Auto entweder mit seinem linken oder mit seinem rechten Nachbarn ein Paar. Wird zufällig ein Auto aus den drei vorhanden gewählt, ist die Wahrscheinlichkeit, dass dieses Auto Teil eines Paares ist $2/3$.

Wenn die Anzahl an Autos größer wird, ist ein analytischer Beweis nicht mehr einfach zu führen. Die Anwendung der Monte Carlo Simulation ist sinnvoll.

Das dazu erzeugte Programm führt folgende Teilschritte aus.

Die Anzahl der parkenden Autos wird mit n definiert. Es werden n Zufallszahlen aus dem Intervall $[0; 1]$ generiert und sortiert. Die sortierte Liste stellt die Positionen der parkenden Autos dar, aus denen die Abstände zweier benachbarter Autos berechnet werden.

Die Liste der Abstände wird zunächst von oben nach unten paarweise abgearbeitet und anschließend von unten nach oben, um sicherzustellen, dass nicht nur eine „einseitige“ Partnerschaft vorliegt. Beim Abarbeiten der Abstandsliste werden nebeneinanderliegende Abstände betrachtet. Wenn bei den betrachteten Zweiergruppen der erste Eintrag der kleinere ist, wird ihm eine 1 zugeordnet, ansonsten eine 0. Entsprechend wird die Liste von unten nach oben abgearbeitet. Im Ergebnis entstehen zwei Listen aus einer Abfolge von 0 und 1. Erscheint auf derselben Position in beiden Listen eine 1, haben zwei Autos zueinander den geringsten Abstand und es handelt sich um ein Paar. Die beiden Autos am linken bzw. rechten Rand müssen dabei als Spezialfall betrachtet werden.

Die Anzahl der Paare wird gezählt und verdoppelt, da an jedem Paar zwei Autos beteiligt sind. Anschließend wird die Anzahl der Autos, die einem Paar zugehörig sind, durch n geteilt um die gesuchte Wahrscheinlichkeit zu erhalten. Der gesamte Prozess wird k mal durchgeführt und die Ergebnisse gemittelt.

Bei großen Werten von n ist eine Annäherung der Wahrscheinlichkeit an den Wert $2/3$ zu erkennen.

4 Diskussion

4.1 Auswertung der Ergebnisse

Die Ergebnisse der verschiedenen Versuche zeigen, dass durch eine möglichst hohe Zahl an Durchführungen ein relativ genaues Endergebnis erzielt werden kann. Dadurch können mit vergleichsweise einfachen Programmen komplexe Aufgaben gelöst werden, die analytisch nur schwierig oder unmöglich zu lösen wären.

Der Erfolg der Monte- Carlo- Methode spiegelt sich in ihrer weit gefächerten Nutzung in Wissenschaft und Wirtschaft wieder.

Das Ziel ist immer dasselbe: durch eine Vielzahl an Variablen können verschiedene Situationen eintreten, welche alle berücksichtigt werden müssen, um ein der Realität nahekommendes Ergebnis zu kommen. Durch das Weglassen oder Setzen bestimmter Variablen, wird die Ausgangslage zwar weniger komplex und somit leichter in einer Simulation abzubilden, aber ein „gutes“ bzw. „zufriedenstellendes“ Ergebnis immer unwahrscheinlicher.

Vor allem in der Medizin aber auch in der Wirtschaft ist es wichtig, möglichst viele Variablen berücksichtigen zu können, weil in realen Anwendungen mannigfaltige Faktoren wirken und bereits kleinste Veränderungen erheblichen Einfluss auf das Ergebnis haben können.

In der Praxis werden Modelle nicht nur bezüglich ihrer Komplexität optimiert, sondern auch in Bezug auf ihre Konvergenzrate und Varianz, d.h. nach wie vielen Iterationen sich der Ergebniswert hinreichend asymptotisch einem prognostizierten Erwartungswert genähert hat. Die Varianz, d.h. die Streuung der Abweichung vom Erwartungswert, sollte durch geeignete Auswahl der Variablen und Strukturierung des Modells möglichst gering gehalten werden. Berücksichtigt werden muss, das die Varianz in manchen Fällen auch Teil des Phänomens sein kann.

5 Danksagung

Unser besonderer Dank gilt Herrn Dr. Hector und seiner Frau für die Möglichkeit am Hector-Seminar und damit auch an diesem Kooperationsprojekt teilzunehmen zu können. Durch ihre großzügige Förderung haben sie es uns und vielen anderen ermöglicht, neue Erfahrungen zu sammeln und unser Wissen zu erweitern.

Unseren Kursleitern Frau Krämer, Herrn Taulien, Herrn Gözl sowie Herrn Raqué gilt unser Dank für die langjährige, engagierte und verständnisvolle Betreuung.

Besonders bedanken wir uns bei Herrn Dr. Parczewski von der Universität Mannheim für die umfassende Unterstützung und Betreuung während unseres Kooperationsprojektes.

6 Quellen

Textquellen

Nahin, Paul J.: Digital Dice. Computational Solutions to Practical Probability Problems, Princeton 2008.

Henze, Norbert: Stochastik für Einsteiger. Eine Einführung in die faszinierende Welt des Zufalls, o.O. 2013.

Hairer, Ernst /Wanner, Gerhard: Analysis in historischer Entwicklung, Springer, o.O. 2011.

Müller-Gronbach, Thomas /Novak, Erich /Ritter, Klaus: Monte-Carlo Algorithmen. o.O. 2012.

Georgii, Hans-Otto: Stochastik. Einführung in die Wahrscheinlichkeitstheorie und Statistik, o.O. 2015.

Kersting, Götz /Wakolbinger, Anton: Elementare Stochastik, o.O. 2010.

Bildquellen

Parczewski, Peter: Monte Carlo Methoden, 10.10.2016.

Ruskeepää, Heikki: A Curious Coin Flipping Game, 16.04.2012.

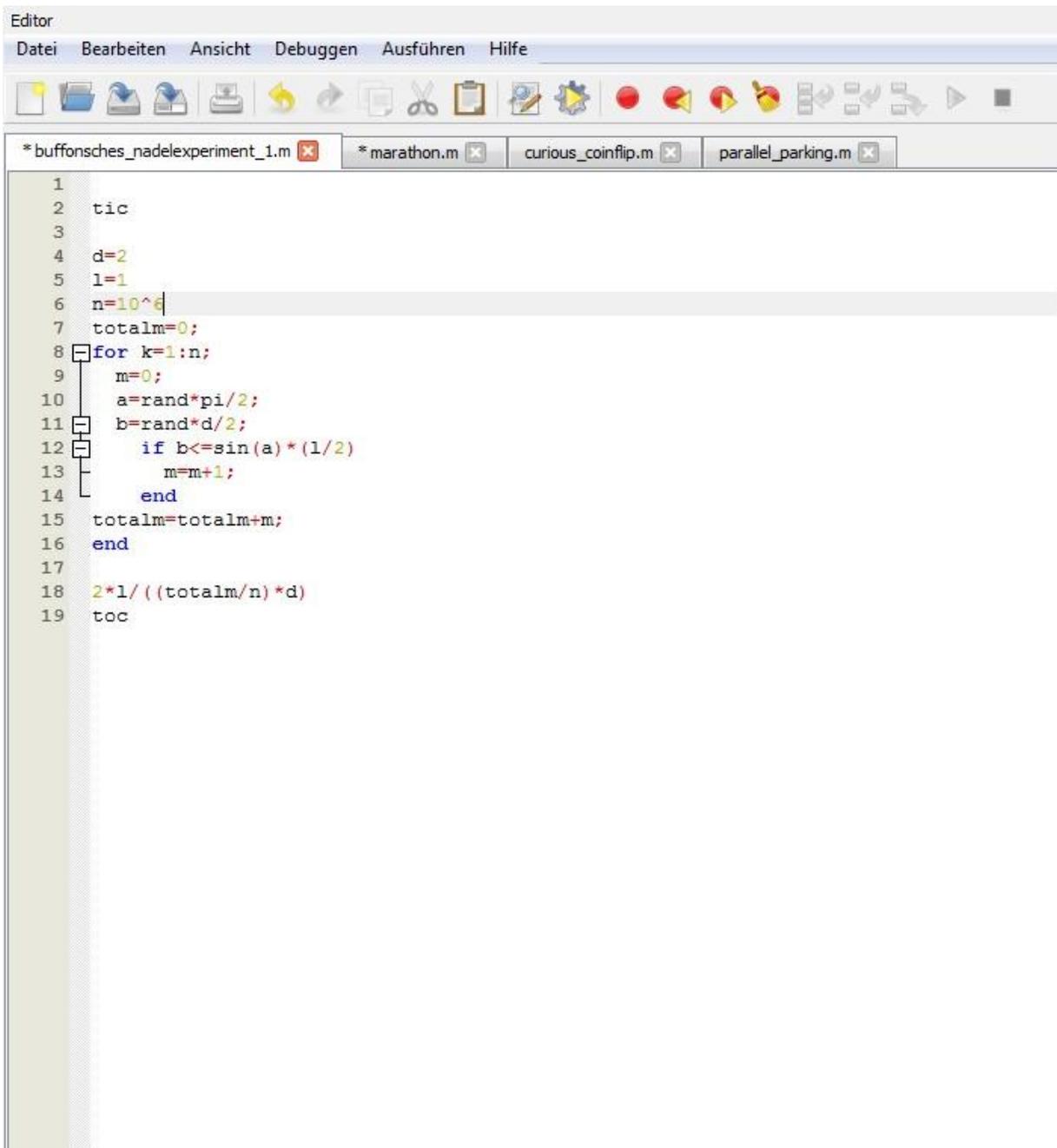
<http://demonstrations.wolfram.com/ACuriousCoinFlippingGame/> (Stand: 02.10.2017)

Radhakrishna: Digital Dice RK, 15.03.2015. <http://radhakrishna.typepad.com/digital-dice---solving-puzzles-with-r.pdf> (Stand: 02.10.2017)

7 Anhang

Im Rahmen des Projekts mithilfe von GNU Octave erstellte Skripte zur Lösung der oben beschriebenen Monte Carlo Probleme:

7.1 Buffonsches Nadelexperiment



```
1
2 tic
3
4 d=2
5 l=1
6 n=10^6
7 totalm=0;
8 for k=1:n;
9     m=0;
10    a=rand*pi/2;
11    b=rand*d/2;
12    if b<=sin(a)*(l/2)
13        m=m+1;
14    end
15    totalm=totalm+m;
16 end
17
18 2*l/((totalm/n)*d)
19 toc
```

7.2 A Curious Coin Flipping Game

```
Editor
Datei Bearbeiten Ansicht Debuggen Ausführen Hilfe
*buffonsches_nadelexperiment_1.m x *marathon.m x curious_coinflip.m x parallel_parking.m x
1
2 K=10^6
3 totalflips=0;
4 for sequences=1:k
5     sequenceflips=0;
6     man(1)=1;
7     man(2)=2;
8     man(3)=3;
9     while man(1)>0&man(2)>0&man(3)>0
10        flip(1)=rand;
11        flip(2)=rand;
12        flip(3)=rand;
13        for j=1:3
14            if flip(j)<p
15                flip(j)=1;
16            else
17                flip(j)=0;
18            end
19        end
20        test=sum(flip);
21        if test==1|test==2
22            if test==1
23                for j=1:3
24                    if flip(j)==0
25                        flip(j)=-1;
26                    else
27                        flip(j)=2;
28                    end
29                end
30            else
31                for j=1:3
32                    if flip(j)==0
33                        flip(j)=2;
34                    else
35                        flip(j)=-1;
36                    end
37                end
38            end
39            for j=1:3
40                man(j)=man(j)+flip(j);
41            end
42        end
43        sequenceflips=sequenceflips+1;
44    end
45    totalflips=totalflips+sequenceflips;
46 end
47 totalflips/K
48
```

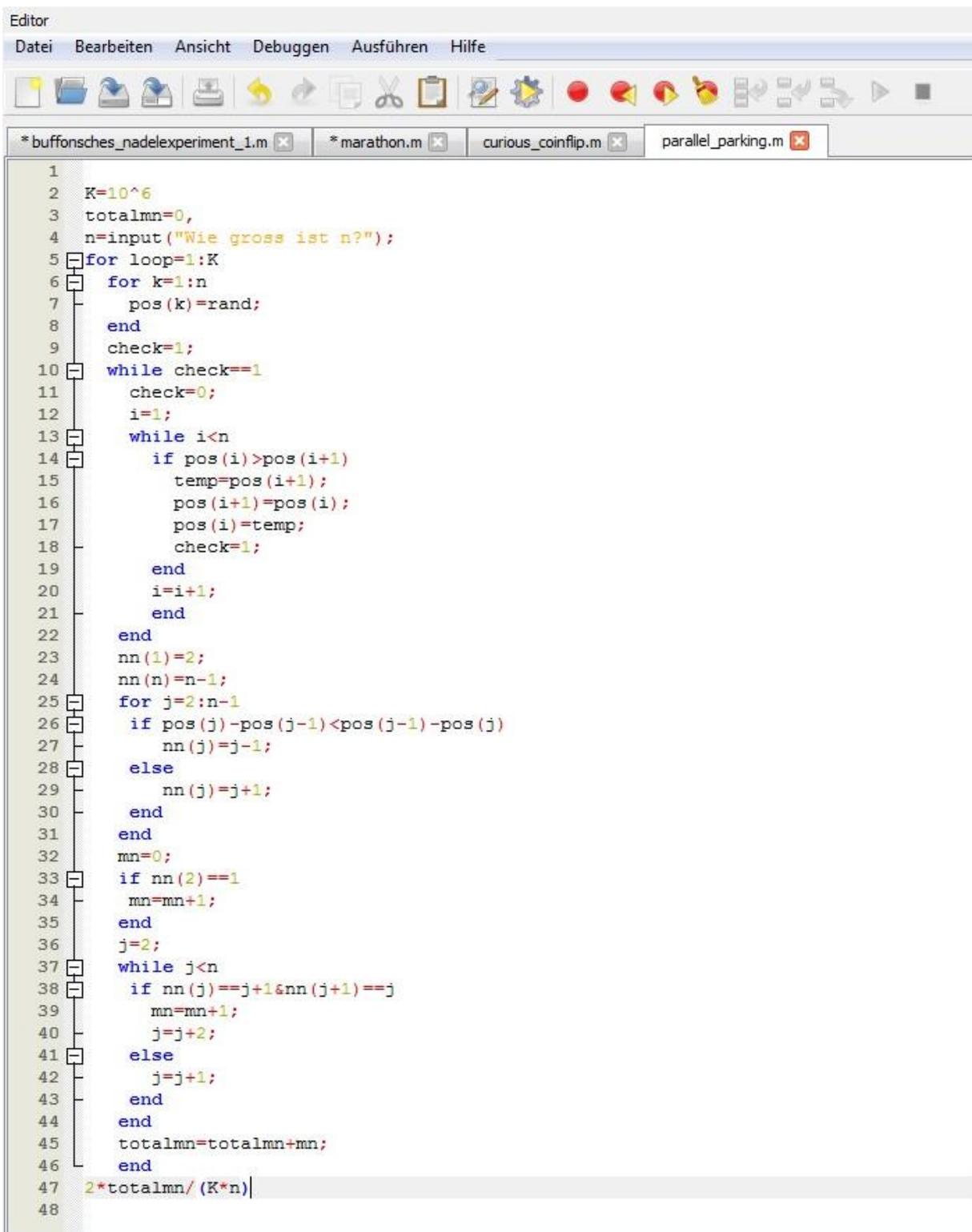
7.3 How Many Runners in a Marathon?

```
Editor
Datei Bearbeiten Ansicht Debuggen Ausführen Hilfe

*buffonsches_nadelexperiment_1.m x *marathon.m x curious_coinflip.m x parallel_parking.m x

1
2 tic
3 size(1)=2;size(2)=5;size(3)=10;size(4)=20;
4 for z=1:4
5     s=size(z);
6     for loop=1:10^5
7         N=round(900*rand)+100;
8         laeuferauswahl=randperm(N);
9         n=round(s*N/100);
10        newN=N;newn=n;
11        beobachtet=zeros(1,n);
12        j=1;
13        for k=1:N
14            p=newn/newN;
15            newN=newN-1;
16            if rand<p
17                beobachtet(j)=laeuferauswahl(k);
18                j=j+1;
19                newn=newn-1;
20            end
21        end
22        maximum=max(beobachtet);
23        abweichung(loop)=((n+1)/n)*maximum-1-N)*100/N;
24    end
25    subplot(2,2,z)
26    hist(error,100)
27    xlabel("prozent abweichung")
28    ylabel("anzahl durchfuehrungen")
29    title_str=["größe n="int2str(size(z))"5"];
30    title(title_str)
31 end
32 toc
```

7.4 A Parallel Parking Question



The image shows a MATLAB editor window with a menu bar (Datei, Bearbeiten, Ansicht, Debuggen, Ausführen, Hilfe) and a toolbar. The active file is 'parallel_parking.m'. The code is as follows:

```
1
2 K=10^6
3 totalmn=0,
4 n=input("Wie gross ist n?");
5 for loop=1:K
6     for k=1:n
7         pos(k)=rand;
8     end
9     check=1;
10    while check==1
11        check=0;
12        i=1;
13        while i<n
14            if pos(i)>pos(i+1)
15                temp=pos(i+1);
16                pos(i+1)=pos(i);
17                pos(i)=temp;
18                check=1;
19            end
20            i=i+1;
21        end
22    end
23    nn(1)=2;
24    nn(n)=n-1;
25    for j=2:n-1
26        if pos(j)-pos(j-1)<pos(j-1)-pos(j)
27            nn(j)=j-1;
28        else
29            nn(j)=j+1;
30        end
31    end
32    mn=0;
33    if nn(2)==1
34        mn=mn+1;
35    end
36    j=2;
37    while j<n
38        if nn(j)==j+1&nn(j+1)==j
39            mn=mn+1;
40            j=j+2;
41        else
42            j=j+1;
43        end
44    end
45    totalmn=totalmn+mn;
46 end
47 2*totalmn/(K*n)
48
```

8 Selbstständigkeitserklärung

Hiermit versichern wir, dass wir diese Arbeit unter der Beratung von Herrn Dr. Parzcewski selbstständig verfasst haben und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden, sowie Zitate kenntlich gemacht haben.

Neckargemünd, den 02.10.2017

Ilvesheim, den 02.10.2017

Christopher Schön

Xaver Kohlbrenner