

Kapitel 5

Monte-Carlo-Methoden

5.1 Einführung

Als Monte-Carlo-Methoden (MC-Methoden) werden Verfahren bezeichnet, mit denen numerische Probleme mit Hilfe von wiederholtem Ziehen von Zufallsstichproben aus bekannten Verteilungen gelöst werden. Diese Methoden werden häufig zur Simulation von mathematischen, physikalischen, biologischen, technischen oder ökonomischen Systemen benutzt, insbesondere wenn deterministische Algorithmen zu aufwendig oder vielleicht garnicht möglich sind.

Komplexe Simulationsprogramme, wie zum Beispiel die Simulation von Luftschauern hochenergetischer kosmischer Strahlung, die Simulation von Klimamodellen oder eines Öko-Systems, benötigen leistungsfähige Computer. Trotz des enormen Anstiegs von Schnelligkeit und Kernspeicherplatz der Rechner in den letzten Jahren sind viele Probleme nur mit vereinfachenden Annahmen zu simulieren. Zum Beispiel können globale Klimamodelle erst seit ein paar Jahren mit einigermaßen aussagekräftigen Ergebnissen simuliert werden.

Typische Anwendungen findet die MC-Methode zur Lösung folgender Probleme:

- Numerische Lösung von Integralen: viele Anwendungen lassen sich letztlich auf die Lösung von Integralen zurückführen. Zum Beispiel ist die Nachweiswahrscheinlichkeit eines Detektors für eine bestimmte Teilchenreaktion definiert als ein Integral über den Phasenraum der Reaktion in den Grenzen der Akzeptanz des Detektors gewichtet mit Verlustwahrscheinlichkeiten für einzelne Teilchen (in der Realität stellt sich das Problem im Allgemeinen noch komplexer dar, zum Beispiel durch kinematische Migrationen durch Streuung und Energieverlust).
- Simulation von dynamischen Prozessen: zum Beispiel Bewegungsabläufe von mechanischen Systemen in der Technik, Produktionsabläufe in der Wirtschaft oder die Entwicklung des Wetters.
- Simulation von Gleichgewichtszuständen, zum Beispiel in der statistischen Physik oder bei dem Einsatz bestimmter Typen neuronaler Netze. Diese Anwendung ist hier getrennt aufgeführt, weil dafür spezielle Methoden entwickelt wurden (zum Beispiel der Metropolis-Algorithmus).

- Statistische Untersuchung von Zufallsverteilungen, die analytisch nicht oder nur schwer zu behandeln sind. Dazu gehört zum Beispiel auch die Bestimmung von Fehlern einer Messung indem man das Experiment vielfach simuliert und den Fehler durch die Schwankung der simulierten Ergebnisse abschätzt ('bootstrap' Methode).

Auch vor der Entwicklung leistungsfähiger Computer wurden Simulationen zur Lösung komplexer mathematischer Probleme als 'analoge Simulationen' eingesetzt, wie zum Beispiel die Optimierung von Fahrzeugformen in Windkanälen oder die Lösung gekoppelter Differentialgleichungen mit Pendelsystemen. Ein schönes Beispiel, dass auf zufälligen Stichproben beruhende Simulationen auch ohne Computer gemacht werden können, ist das Buffonsche Nadelexperiment zur Bestimmung der Zahl π :

Beispiel: Auf ein Blatt Papier mit parallelen Linien im Abstand g werden Nadeln der Länge l so geworfen, dass ihre Lage und Richtung zufällig ist. Die Wahrscheinlichkeit, dass eine Nadel eine Linie kreuzt, hängt wegen der Rotationssymmetrie der Nadelorientierung mit der Zahl π zusammen:

$$p = \frac{2l}{g\pi} \quad \implies \quad \pi = \frac{2l}{gp} \quad (5.1)$$

Die Wahrscheinlichkeit p wird nun experimentell durch das Werfen von Nadeln bestimmt.

Häufig entspricht die Aufgabenstellung der Lösung eines Integrals in einem multi-dimensionalen Raum mit komplizierten Integrationsgrenzen. Mit der MC-Methode wird das Integral gelöst, indem man diskrete Punkte in dem Raum nach dem Zufallsprinzip würfelt.

Das Integral kann nun auf verschiedene Weise ausgewertet werden. Nach der einfachsten Methode werden die Punkte gleichverteilt in dem Raum erzeugt und die Integrandenfunktion wird an den diskreten Punkten aufaddiert. Das entspricht der numerischen Lösung des Integrals durch eine endliche Summe über Intervalle. Hier könnte man fragen, ob es nicht grundsätzlich am günstigsten ist, eine feste Intervallaufteilung zu machen, wodurch der Fehler des Integrals mit der Anzahl N der Intervalle abfallen würde. Dagegen fällt bei einer zufälligen Wahl der Punkte der Fehler nur wie $1/\sqrt{N}$ ab. Bei einer einzelnen Dimension ist eine gleiche Verteilung der Punkte auf jeden Fall optimaler. Allerdings ist es in höheren Dimensionen für auf einem regulären Gitter angeordnete Punkte nicht mehr richtig, dass der Fehler mit $1/N$ abnimmt, was an den Korrelationen der untereinander liegt. Da bei der MC-Methode der Fehler immer mit $1/\sqrt{N}$ abnimmt, wird die MC-Methode dimensionsabhängig (und problemabhängig) optimaler ('Monte-Carlo-Paradoxon'). Darüber hinaus bietet die MC-Methode bei komplexen Problemen viele bedenkenswerte weitere Vorteile. Ein ganz wichtiger Vorteil der Benutzung von Zufallsvariablen ist die Möglichkeit, die Simulation beliebig fortzusetzen und damit die Genauigkeit zu erhöhen. Bei diskreter Intervalschachtelung würde ein nächster Schritt mindestens eine Halbierung der Intervallabmessungen bedeuten, was die Rechenzeit bei einer Dimension n um einen Faktor 2^n verlängern würde (also schon ein Faktor von etwa 1000 bei 10 Dimensionen).

Bei der Standard-MC-Methode zur Lösung eines Integrals werden die Punkte in dem Raum mit der durch die normierte Integrandenfunktion gegebenen Wahrscheinlichkeitsdichte erzeugt. Man erhält dann “Ereignisse” mit der entsprechenden Wahrscheinlichkeitsdichte, die dann auch weiteren Analysen unterworfen werden können, was eine hohe Flexibilität bei dem Vergleich der Simulation mit gemessenen Daten ergibt.

In diesem Kapitel werden verschiedene Methoden zur Erzeugung von Stichproben mit bestimmten Wahrscheinlichkeitsdichten und optimale Methoden zur Bestimmung von Integralen besprochen. Für die Anwendung der MC-Methode benötigt man Generatoren von (Pseudo)-Zufallszahlen, deren Eigenschaften wir zunächst kurz besprechen wollen. Wir orientieren uns in diesem Kapitel besonders an [4]; einen guten Überblick gibt auch der Artikel [16].

5.2 Zufallszahlengeneratoren

In der Regel geht man von einem Zufallszahlengenerator aus, der bei jedem Aufruf eine neue Zahl z , die im Intervall $[0, 1]$ gleichverteilt ist, zurückgibt. Aus diesen Zufallszahlen werden die Zufallsvariablen des betrachteten Problems erzeugt.

Die Zufallszahlen werden fast ausschließlich durch geeignete Algorithmen als ‘Pseudozufallszahlen’ im Rechner erzeugt. Ein Problem ist, dass wegen der digitalen Darstellung der reellen Zahlen mit einer endlichen Bit-Anzahl, die Zahlengeneratoren im allgemeinen eine **Periodizität** haben können. Man versucht die Periode möglichst lang zu machen, um große Ereignismengen unabhängig erzeugen zu können. Gute Generatoren sollten auch keine Korrelationen in der Abfolge der Zufallszahlen aufweisen, um Muster in einem multi-dimensionalen Raum zu vermeiden.

Da Zufallszahlengeneratoren im Prinzip ‘deterministisch’ sind, ist eine Wiederholbarkeit von Rechnungen, die statistisch unabhängige Fortsetzung und die parallele Ausführung auf verschiedenen Rechnern möglich. Die Zufallszahlengeneratoren liefern dafür so genannte ‘seeds’, Zahlen mit denen man einen Generator an wohldefinierten Stellen einer Zufallszahlenfolge initiieren kann.

5.2.1 Multiplikativ kongruentielle Generatoren

Es gibt eine Vielzahl von Algorithmen zur Erzeugung von Pseudozufallszahlen. Viele der in der Vergangenheit sehr populären Zufallsgeneratoren gehören zur Klasse der multiplikativ oder gemischt kongruentiellen Generatoren (engl. linear congruential generator, LCG). Das Prinzip soll hier kurz erläutert werden. Eine Zufallszahl erzeugt ein LCG über die Rekursionsrelation:

$$x_{i+1} = (ax_i + b) \pmod{m} \quad (5.2)$$

wobei Modul m , Faktor a , Inkrement b und Startwert x_1 die Zufallssequenz vollständig bestimmen. In der Praxis hängen die Eigenschaften eines LCG sensitiv von der Wahl dieser Parameter ab. Für $m = 2^k$ mit $k \geq 4$ ist die maximale Periode eines LCG bei optimaler Wahl der Parameter $m/4$.

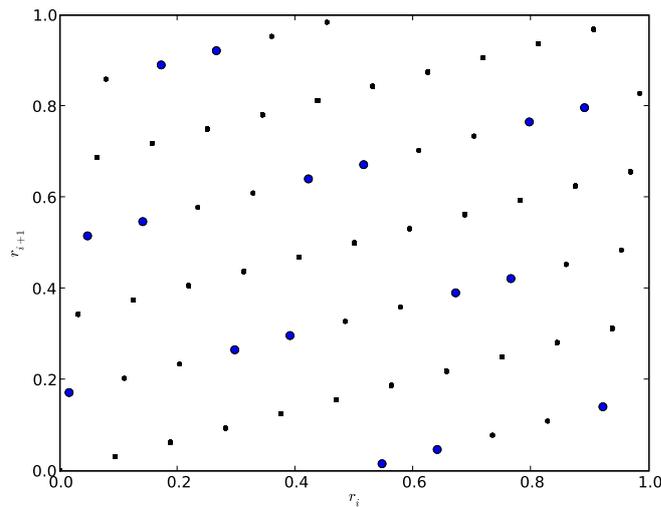


Abbildung 5.1: Iterationsfunktion eines LCG mit $m = 64$, $a = 11$, $b = 0$. Für diesen LCG gibt zwei Sequenzen mit Periode 16 (d.h. der maximalen Periode für $m = 64$), die zusammen alle ungeraden Zahlen $< m$ enthalten. Die fetten Punkte entsprechen einer dieser Sequenzen. Gerade Zahlen als Startwerte liefern (teilweise deutlich) kürzere Perioden.

LCG haben deutliche Schwächen, z.B. kleine Periode der Sequenz (und noch geringere Perioden für nicht signifikante Stellen), sowie deutliche Korrelation von aufeinander folgenden Zufallszahlen. Letztere Eigenschaft führt bei Erzeugung von mehrdimensionalen Tupeln zu ungewünschten Strukturen (“Hyperebenen”). Abbildung 5.1 zeigt die Verteilung von aufeinanderfolgenden Zufallszahlen für $m = 64$, $a = 11$, $b = 0$. Einige, aber nicht alle, Probleme der LCG werden durch Verwendung von verallgemeinerten kongruentiellen Generatoren umgangen.

5.2.2 Mersenne-Twister

Als der “Zufallszahlengenerator der Wahl” hat sich in den letzten Jahren der Mersenne-Twister etabliert. Der Algorithmus gehört zu der Klasse der *twisted generalised feedback shift register*. Er zeichnet sich unter anderem durch eine extrem lange Periode aus ($2^{19937} - 1 \approx 4.3 \cdot 10^{6001}$), erzeugt sehr gute Gleichverteilungen (nachgewiesen bis zur Dimension 623), und ist dennoch schneller als andere (hinreichend gute) Zufallsgeneratoren. Er wird in allen modern Programmbibliotheken wie z.B. der *GNU Scientific Library* eingesetzt. Der Zustand des Generators wird eindeutig durch 624+1 Integer-Zahlen (32bit) beschrieben, für deren Initialisierung in der Regel ein einfacher LCG Algorithmus verwendet wird.

5.2.3 Quasi-Zufallszahlen

Um das Konvergenzverhalten von Monte Carlo Integrationsalgorithmen zu verbessern, werden gelegentlich Quasi-Zufallszahlen eingesetzt. Quasi-Zufallszahlen Genera-

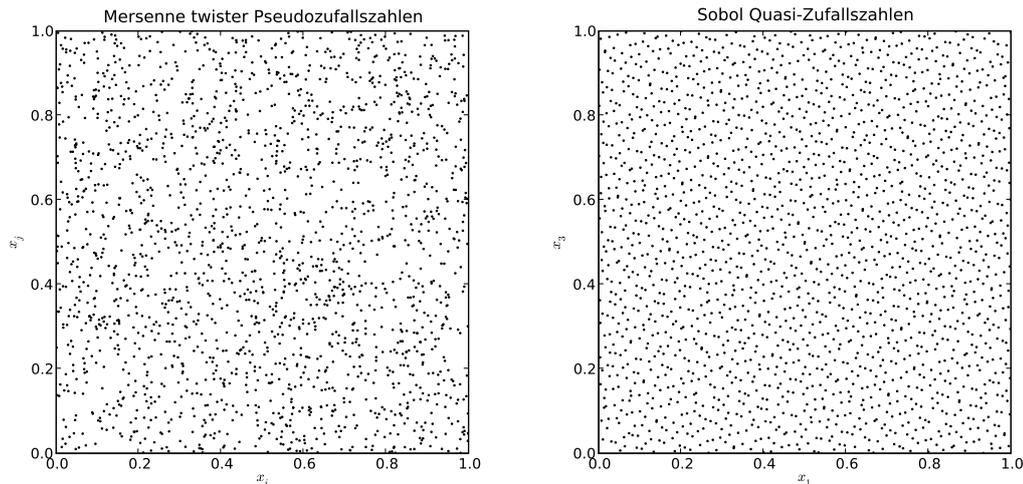


Abbildung 5.2: Pseudozufallszahlen erzeugt durch einen Mersenne-Twister, und Quasi-Zufallszahlen nach der Sobol-Sequenz in 3 Dimensionen.

toren (QZG) sind deterministische Algorithmen die eine gleichmäßigere Füllung des Integrationsvolumen mit Punkten garantiert, ohne dass die Punkte wie auf einem regulären Gitter korreliert sind. In Abbildung 5.2 werden Zufallsverteilung erzeugt mit dem Mersenne-Twister mit Quasi-Zufallszahlen verglichen. Quasi-Zufallszahlen basieren häufig auf der van-der-Corput-Sequenz (Halton-Sequenz) oder auf der Sobol-Sequenz.

Quasi-Zufallszahlen eignen sich **nur** zum Integrieren. Auch muss die Dimension des Problems bereits bei der Erzeugung der Zufallszahlen feststehen. Zur Verbesserung der Genauigkeit der Integration muss an der Stelle der Sequenz, an der unterbrochen wurde, fortgesetzt werden. QZG stellen also einen besonderen Anspruch an die Disziplin des Programmierers.

5.3 Monte-Carlo-Erzeugung von Ereignissen

Einfache Beispiele für die Erzeugung von Ereignissen entsprechend vorgegebenen Zufallsverteilungen sind bereits in den Kapiteln 1 und 3 gegeben worden. Der Vollständigkeit halber werden wir die dort eingeführten Inversions- und Hit-and-Miss-Methoden wiederholen und auf multi-dimensionale Verteilungen erweitern.

Das prinzipielle Problem ist die Zuordnung von Sätzen $\vec{z} = (z_1, z_2, \dots, z_n)$ von jeweils im Einheitsintervall gleichverteilten Zufallsvariablen zu den Variablen $\vec{x} = (x_1, x_2, \dots, x_n)$ einer Wahrscheinlichkeitsdichte $f(\vec{x})$, so dass die entsprechenden Ereignisse \vec{x}_j ($j = 1, \dots, N$) Stichproben der Verteilung $f(\vec{x})$ sind. Das ist im Allgemeinen keine leichte Aufgabe, insbesondere wenn die Variablen x_i untereinander korreliert sind.

5.3.1 Inversionsmethode

Eine Methode haben wir bereits in Abschnitt 1.3.1 für eine einzelne Variable eingeführt. Zusammengefasst sind die Ergebnisse: Aus der Forderung, dass bei einer Transformation einer Zufallsvariablen x mit einer Wahrscheinlichkeitsdichte $f(x)$ auf eine im Einheitsintervall gleichverteilte Zufallsvariable z die differentielle Wahrscheinlichkeit gleich bleiben muss,

$$f(x)dx = dz, \quad (5.3)$$

ergibt sich

$$z = F(x) \quad \text{und die Umkehrung:} \quad x = F^{-1}(z), \quad (5.4)$$

wobei $F(x)$ die Verteilungsfunktion von $f(x)$ ist (die ja zwischen 0 und 1 gleichverteilt ist). Falls F analytisch invertierbar ist, ist damit das Problem gelöst. Es ist im Prinzip auch möglich, numerisch zu invertieren, was aber häufig zeitaufwendiger ist, als die Anwendung anderer Methoden, die zum Teil im Folgenden besprochen werden.

Im mehr-dimensionalen Fall muß man im Allgemeinen schrittweise vorgehen, beginnend zum Beispiel mit der Erzeugung der Variablen x_1 . Aus der Randverteilung $h_1(x_1)$ (siehe (3.6)) zu dieser Variablen kann man mit der Inversionsmethode die Zufallsvariable $x_{1,j}$ (j ist das jeweilige Ereignis) erzeugen. Da im Allgemeinen die übrigen Variablen von x_1 abhängig sein können, muss man im weiteren die bedingte Wahrscheinlichkeit für $x_1 = x_{1,j}$ wie in (3.8) definiert betrachten:

$$f^*(x_2, x_3, \dots, x_n | x_1 = x_{1,j}) = \frac{f(x_1 = x_{1,j}, x_2, \dots, x_n)}{h_1(x_1 = x_{1,j})}$$

Diese Verteilung wird nun benutzt, um mit der Inversionsmethode die Variable x_2 zu erzeugen. Die Schritte wiederholen sich bis zur letzten Variablen x_n . Am einfachsten ist natürlich, wenn in jedem Schritt die Invertierung analytisch gemacht werden kann; grundsätzlich ist aber auch eine numerische Invertierung möglich. Das Verfahren lässt sich, wie in Abschnitt 1.3.1 besprochen, entsprechend auch auf diskrete Verteilungen anwenden.

Beispiel: Ein Beispiel für die eindimensionale Lebensdauerverteilung ist in Abschnitt 1.2.1 gegeben worden (Gleichungen (1.35 - 1.37)).

Für eine zwei-dimensionalen Verteilung wurde ein Beispiel in Abschnitt 3.6 gegeben (allerdings für den einfacheren Fall unabhängiger Variablen).

Für eine zwei-dimensionale, unkorrelierten Gauss-Verteilung lässt sich durch Transformation der kartesischen Koordinaten (x_1, x_2) auf Polarkoordinaten (r, ϕ) (Box-Müller-Transformation) analytisch invertierbare Verteilungsfunktionen erhalten (was im Eindimensionalen nicht der Fall ist). Man erhält dann mit der oben angegebenen Methode und nach Rücktransformation die unabhängig gauss-verteilten Zufallszahlen (x_1, x_2) als Funktion der im Einheitsintervall gleichverteilten Zufallszahlen (z_1, z_2) :

$$\begin{aligned} x_1 &= \sigma \sqrt{-\ln z_1^2} \cdot \sin(2\pi z_2) \\ x_2 &= \sigma \sqrt{-\ln z_1^2} \cdot \cos(2\pi z_2) \end{aligned} \quad (5.5)$$

Tabelle 5.1: Erzeugungsalgorithmen von Zufallszahlen einiger wichtiger Wahrscheinlichkeitsverteilungen aus gleichverteilten Zufallszahlen z in $[0, 1[$.

Wahrscheinlichkeitsdichte	Wertebereich	Algorithmus
$f(x) = \frac{1}{b-a}$	$[a, b[$	$x = (b-a) \cdot z + a$
$f(x) = 2x$	$[0, 1[$	$x = \max(z_1, z_2)$ or $x = \sqrt{z}$
$f(x) \sim x^{r-1}$	$[a, b[$	$x = [(b^r - a^r) \cdot z + a^r]^{1/r}$
$f(x) \sim \frac{1}{x}$	$[a, b[$	$a \cdot (b/a)^z$
$f(x) = \frac{1}{x^2}$	$]1, \infty]$	$x = 1/z$
$f(x) = \frac{1}{k} e^{-x/k}$	$]0, \infty]$	$x = -k \ln z$
$f(x) = x e^{-x}$	$]0, \infty]$	$x = -\ln(z_1 \cdot z_2)$
$f(x) = -\ln x$	$[0, 1[$	$x = z_1 \cdot z_2$
Gauss: $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{x^2}{2\sigma^2}}$	$[-\infty, \infty]$	$x = \sigma \sqrt{-\ln z_1^2} \cdot \cos(2\pi z_2)$
Breit-Wigner: $f(x) = \frac{\Gamma}{2\pi} \cdot \frac{1}{(x-\mu)^2 + (\Gamma/2)^2}$	$[-\infty, \infty]$	$x = [\tan \pi(z - 0.5)] \cdot \Gamma/2 + \mu$

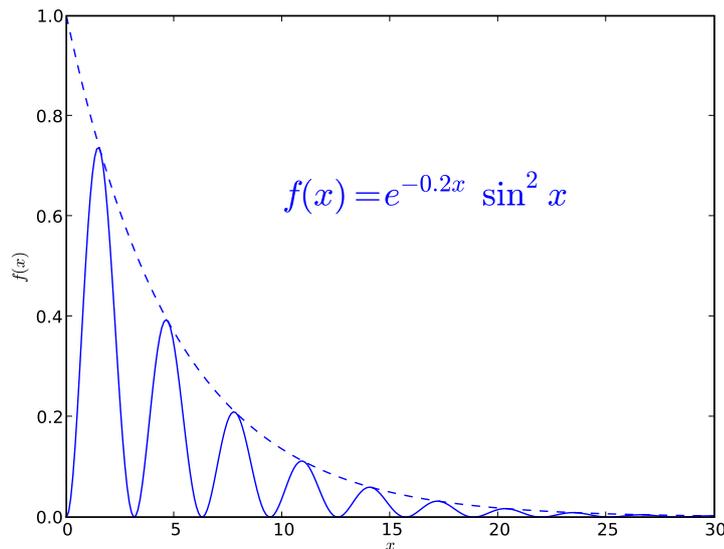


Abbildung 5.3: Verteilung mit Majorante.

5.3.2 ‘Hit-and-Miss’-Verfahren

Das in Abschnitt 1.3.2 besprochene ‘Hit-and-Miss’-Verfahren für eine einzelne Variable lässt sich einfach auf mehrere Dimensionen übertragen: Man definiert in n Dimensionen einen Quader der den gesamten Wertebereich der Zufallsvariablen x_i enthält. In diesem Quader würfelt man gleichverteilt Punkte; zu jedem Punkt wird eine zusätzliche Variable z_f gleichverteilt zwischen 0 und f_{max} , dem Maximalwert der Funktion $f(\vec{x})$ (oder größer), gewürfelt. Wenn $z_f < f_{max}$ gilt, wird der Punkt akzeptiert (‘hit’) oder verworfen (‘miss’).

Wie im ein-dimensionalen Fall gilt auch hier, dass die Methode ineffizient wird, wenn es große Unterschiede in den Funktionswerten gibt. Das lässt sich verbessern, indem das gesamte Quadervolumen in kleinere Quader mit geringeren Funktionsschwankungen unterteilt wird. Die Anzahl der Versuche in jedem Unterquader ist proportional zu dessen Volumen. Man wird also zunächst durch Würfeln in der diskreten Verteilung $p_i = V_i/V_{tot}$ (entsprechend Abb. 1.7) das entsprechende Volumen auswählen, in dem man das nächste Ereignis erzeugt. Der Übergang von diskreten Intervalschachtelungen zu einer kontinuierlichen Umhüllenden der Wahrscheinlichkeitsdichte wird mit der im Folgenden beschriebenen Majorantenmethode vollzogen.

5.3.3 Majorantenmethode

Für eine Majorante $g(\vec{x})$ einer Funktion $f(\vec{x})$ gilt:

$$g(\vec{x}) \geq f(\vec{x}) \quad \forall \vec{x} \quad (5.6)$$

Man sucht sich nun eine Majorante $g(\vec{x})$ von $f(\vec{x})$, für die man einfacher als für $f(\vec{x})$ Ereignisse erzeugen kann, zum Beispiel mit der Inversionsmethode. Wie bei der ‘Hit-

and-Miss'-Methode akzeptiert man dann ein Ereignis j mit der Wahrscheinlichkeit

$$w_j = \frac{f(\vec{x}_j)}{g(\vec{x}_j)}. \quad (5.7)$$

Dazu würfelt man gleichverteilt zwischen 0 und $g(\vec{x}_j)$ und verwirft die Ereignisse, wenn die gewürfelte Zahl größer als $f(\vec{x}_j)$ ist.

5.3.4 Wichtung der Ereignisse

Statt Ereignisse zu verwerfen, kann man einem Ereignis auch ein Gewicht geben. Bei der Majoranten-Methode ist es das in (5.7) definierte Gewicht $w_j = f(\vec{x}_j)/g(\vec{x}_j)$; beim einfachen 'Hit-and-Miss'-Verfahren ist es $w_j = f(\vec{x}_j)/f_{max}$ (die Majorante ist hier die Konstante f_{max}).

Auch wenn man zunächst ungewichtete Ereignisse erzeugt, kann es vorteilhaft sein, bei einer Datenanalyse auf jeden Fall zu jedem Ereignis einen Speicherplatz für ein Gewicht mitzuführen. Häufig möchte man nämlich unterschiedliche Modelle, zum Beispiel unterschiedliche Matrixelemente eines Wirkungsquerschnitts, austesten, was durch 'Umwichten' der Ereignisse einfach und ökonomisch möglich ist. Wurde ein Ereignis entsprechend der Wahrscheinlichkeitsdichte $f(\vec{x})$ erzeugt und soll nun der Wahrscheinlichkeitsdichte $g(\vec{x}_j)$ folgen, wird jedes Ereignisgewicht mit dem entsprechenden Verhältnis multipliziert:

$$w'_j = w_j \cdot \frac{g(\vec{x}_j)}{f(\vec{x}_j)} \quad (5.8)$$

Bei der Verarbeitung von gewichteten Ereignissen, zum Beispiel bei grafischen Darstellungen, müssen die Gewichte immer berücksichtigt werden. Das gilt insbesondere auch bei der Fehlerrechnung. Wenn man zum Beispiel N Einträge in einem Intervall eines Histogramms hat, ist der Fehler bei ungewichteten Ereignissen \sqrt{N} , entsprechend der Poisson-Statistik. Bei gewichteten Ereignissen ist der entsprechende Eintrag

$$N_w = \sum_{j=1}^N w_j. \quad (5.9)$$

Den Fehler von N_w kann man durch Fehlerfortpflanzung bestimmen. Man nimmt dazu an, dass man N unabhängige Ereignisse hat, jedes mit dem Poisson-Fehler $\sigma_j = \sqrt{1} = 1$ (zu einem einzelnen Ereignis, $N_j = 1$). Dann ergibt die Fehlerfortpflanzung:

$$\sigma^2(N_w) = \sum_{j=1}^N \left(\frac{\partial N_w}{\partial N_j} \right)^2 \sigma_j^2 = \sum_{j=1}^N w_j^2 \quad (5.10)$$

Der relative Fehler ist

$$\frac{\sigma(N_w)}{N_w} = \frac{\sqrt{\sum_{j=1}^N w_j^2}}{\sum_{j=1}^N w_j}. \quad (5.11)$$

Gewichte sollten nicht stark variieren, weil sonst die statistischen Fluktuationen sehr groß werden können.

5.4 Monte-Carlo-Integration

Die Monte-Carlo-Methode kann zur numerischen Bestimmung von Integralen benutzt werden. Dazu werden in dem Definitionsbereich der Integrandenfunktion Zufallsereignisse generiert, deren Gesamtheit das Integral bestimmt. Im einzelnen können dazu die Methoden herangezogen werden, die im vorigen Abschnitt zu Erzeugung von Ereignissen benutzt wurden. Für die Bestimmung eines Integrals ist es wichtig, mit welcher Methode am effektivsten eine gewünschte Genauigkeit erreicht werden kann. Wir werden im Folgenden die verschiedene Methoden daraufhin untersuchen.

Wir nehmen an, dass die Integrandenfunktion $f(\vec{x})$ im Integrationsvolumen Ω^n nur positive oder nur negative Werte annimmt:

$$f(\vec{x}) \geq 0 \quad \text{oder} \quad f(\vec{x}) \leq 0, \quad \vec{x} \in \Omega^n. \quad (5.12)$$

Falls das nicht der Fall ist, muss Ω^n in entsprechende Bereiche zerlegt werden, in denen f nur ein Vorzeichen hat. In solchen Bereichen kann $f(\vec{x})$ nach einer geeigneten Normierung als Wahrscheinlichkeitsdichte interpretiert werden, bezüglich der MC-Ereignisse generiert werden können.

5.4.1 Majoranten-Methode mit Hit-or-Miss

Mit dem Hit-or-Miss-Verfahren ergibt sich eine Schätzung des gesuchten Integrals I zu:

$$I = \int_{\Omega^n} f(\vec{x}) dx_1 \dots dx_n \approx I_{ref} \frac{k}{N} \quad (5.13)$$

Dabei ist I_{ref} das Integral einer Majorantenfunktion $g(x)$ über Ω^n (im einfachsten Hit-or-Miss-Verfahren ist $g(x) = f_{max}$ eine Konstante); N ist die Anzahl der in Ω^n generierten MC-Ereignisse und k die der akzeptierten.

Die Effizienz der MC-Erzeugung

$$\epsilon = \frac{k}{N} \quad (5.14)$$

ist ein Parameter der Binomialverteilung von k mit der Varianz

$$\sigma_k^2 = N\epsilon(1 - \epsilon). \quad (5.15)$$

Damit kann der relative Fehler des Integrals abgeschätzt werden:

$$\frac{\sigma_I}{I} = \frac{\sigma_k}{k} = \sqrt{\frac{1 - \epsilon}{k}}. \quad (5.16)$$

Der Fehler wird also klein, wenn die Anzahl der akzeptierten MC-Ereignisse groß wird, und hat mit $1/\sqrt{k}$ das erwartete Poisson-Verhalten. Der Fehler nimmt auch mit wachsender Effizienz ϵ ab und kann sogar im Grenzfall $\epsilon \rightarrow 1$ ganz verschwinden. Das bedeutet, dass die Majorantenfunktion $g(x)$ den Integranden $f(x)$ möglichst gut approximieren sollte. Falls eine analytisch integrierbare Majorantenfunktion nicht gefunden werden kann, kann der gesamte Integrationsbereich so zerlegt werden, dass in jedem einzelnen Untervolumen die Integrandenfunktion nicht stark schwankt und deshalb die Effizienzen hoch sein können. Wir werden dieses Verfahren unter den 'varianz-reduzierende Methoden' weiter unten näher betrachten.

5.4.2 MC-Integration mit Ereigniswichtung

Wenn das Ziel nur die Bestimmung des Integrals unter der Funktion $f(\vec{x})$ ist und die gleichzeitige Gewinnung einer Ereignisstichprobe keine Rolle spielt, gibt es eigentlich keinen Grund Ereignisse nach dem Hit-or-Miss-Verfahren zu verwerfen.

Der einfachste Fall ist die Summation von in Ω^n gleichverteilten Zufallsereignissen gewichtet mit ihren jeweiligen Funktionswerten. Das ist zwar sehr ähnlich dem Quadraturverfahren mit gleichmässigen Intervallen, die MC-Methode ist aber, wie bereits in der Einleitung zu diesem Kapitel angesprochen, bei höheren Dimensionen vorteilhafter. Mit dem Integrationsvolumen V ergibt sich die Schätzung des Integrals zu:

$$I \approx \frac{V}{N} \sum_{j=1}^N f(\vec{x}_j) = V \bar{f} \quad (5.17)$$

Mit der Varianz des Mittelwertes

$$\sigma_{\bar{f}}^2 = \frac{1}{N(N-1)} \sum_{j=1}^N (f(\vec{x}_j) - \bar{f})^2 \quad (5.18)$$

ergibt sich der relative Fehler des Integrals zu

$$\frac{\sigma_I}{I} = \frac{\sigma_{\bar{f}}}{\bar{f}} \quad (5.19)$$

5.4.3 Varianz-reduzierende Verfahren

Die Varianz des Integrals ist also proportional zu der Varianz der Funktion f im Integrationsvolumen. Deshalb sind Methoden zur Reduzierung der Varianz ein wichtiges Hilfsmittel bei der numerischen Integration mit der MC-Methode. Eine Möglichkeit der Varianzreduktion ist die Anwendung des Majoranten-Verfahrens. Die entsprechend der Majoranten $g(\vec{x})$ erzeugten Ereignisse werden ohne Verwerfen wie in (5.13), aber gewichtet, aufsummiert:

$$I = \int_{\Omega^n} f(\vec{x}) dx_1 \dots dx_n = I_{ref} \frac{1}{N} \sum_{j=1}^N w_j \quad (5.20)$$

Das entspricht (5.13), wenn man die Zahl der akzeptierten Ereignisse k durch $N_w = \sum_{j=1}^N w_j$, die Summe der Gewichte, ersetzt. Der Fehler des Integrals ist dann:

$$\frac{\sigma_I}{I} = \frac{\sigma_{N_w}}{N_w} = \frac{\sqrt{\sum_{j=1}^N w_j^2}}{\sum_{j=1}^N w_j}, \quad (5.21)$$

wobei (5.11) auf der rechten Seite eingesetzt wurde. Am geringsten wird der Fehler, wenn die Gewichte alle gleich sind (Beweis wie für die Effizienz des Mittelwertes, Gleichung (4.16) und folgende), das heißt, dass die Majorante g dem Integranden f sehr gut folgt. Es sei $w_j = 1/N$ für alle Ereignisse j . Dann ergibt sich für den relativen Integralfehler in (5.21):

$$\frac{\sigma_I}{I} = \frac{1}{\sqrt{N}}. \quad (5.22)$$

Das ist offensichtlich eine untere Grenze für den Fehler.

5.4.4 Stratified Sampling ('Geschichtete Stichproben')

Es sind verschiedene Methode der Varianzreduktion entwickelt worden, die nicht voraussetzen, dass man die zu integrierende Funktion gut kennt, insbesondere, dass man keine Majorante finden muss. Die Idee ist, Untervolumen so zu definieren, dass die Varianzen jeweils klein werden ('stratified sampling', 'geschichtete Stichproben'). Die Varianzen können beliebig klein gemacht werden, wenn man in beliebig viele Untervolumen aufteilt. Dem steht bei der praktischen Ausführung der Rechenzeit-aufwand und der Bedarf an Speicherplatz limitierend entgegen. Die Frage stellt sich dann eher so: Wenn N Ereignisse erzeugt und auf m Untervolumen aufgeteilt werden sollen, wie finde ich die Untervolumengrenzen, die die Varianz minimieren.

Betrachtet man, z.B., eine Aufteilung des Integrationsvolumen V in zwei gleich große Untervolumen V_a und V_b , und würfelt in jedes Volumen $n/2$ Punkte, dann kann die Varianz durch die Varianzen der Untervolumen σ_a^2 und σ_b^2 ausgedrückt werden

$$\begin{aligned}\frac{\sigma_{(2)}^2}{n} &= \frac{1}{4} \left(\frac{\sigma_a^2}{n/2} + \frac{\sigma_b^2}{n/2} \right) \\ &= \frac{\sigma_a^2 + \sigma_b^2}{2n}\end{aligned}\tag{5.23}$$

Zum Vergleich bestimmt sich die Varianz der Integration im Gesamtvolumen durch (die x_i für $i = [1, n/2]$ sollen im Untervolumen V_a und die x_i für $i = [n/2 + 1, n]$ im Untervolumen V_b liegen)

$$\begin{aligned}\sigma_{(1)}^2 &= \frac{1}{n} \sum_{i=1}^n (f(x_i) - \bar{f})^2 \\ &= \frac{1}{n} \left\{ \sum_{i=1}^{n/2} ((f(x_i) - \bar{f}_a) + (\bar{f}_a - \bar{f}))^2 + \dots \right\} \\ &= \frac{1}{n} \left\{ \sum_{i=1}^{n/2} \left((f(x_i) - \bar{f}_a)^2 + 2 \overbrace{(f(x_i) - \bar{f}_a)(\bar{f}_a - \bar{f})}^0 + (\bar{f}_a - \bar{f})^2 \right) + \dots \right\} \\ &= \left\{ \underbrace{\frac{1}{n} \sum_{i=1}^{n/2} (f(x_i) - \bar{f}_a)^2}_{\sigma_a^2/2} + \frac{1}{2} (\bar{f}_a - \bar{f})^2 + \dots \right\} \\ &= \frac{\sigma_a^2}{2} + \frac{\sigma_b^2}{2} + \frac{1}{2} (\bar{f}_a - \bar{f})^2 + \frac{1}{2} (\bar{f}_b - \bar{f})^2 \\ &= \frac{\sigma_a^2}{2} + \frac{\sigma_b^2}{2} + \frac{1}{2} \left[\bar{f}_a - \frac{1}{2} (\bar{f}_a + \bar{f}_b) \right]^2 + \frac{1}{2} \left[\bar{f}_b - \frac{1}{2} (\bar{f}_a + \bar{f}_b) \right]^2 \\ &= \frac{\sigma_a^2}{2} + \frac{\sigma_b^2}{2} + \frac{1}{4} [\bar{f}_a - \bar{f}_b]^2\end{aligned}\tag{5.24}$$

Die Aufteilung in Volumina, die die Varianz minimieren, erfolgt im Allgemeinen iterativ mit einer Regel, die große Varianzbeiträge minimiert. Ein Beispiel sind folgende Regeln:

- unterteile das Integrationsvolumen in 2 gleiche Untervolumen,
- erzeuge Punkte in beiden Volumina,
- berechne aus den Funktionswerten des Integranden an diesen Punkten die Varianzen für jedes Untervolumen getrennt,
- unterteile das Untervolumen mit der größten Varianz weiter,
- wiederhole die Unterteilung des Untervolumens mit der jeweils größten Varianz,
- breche die Prozedur ab, wenn ein vorgegebenes Abbruchkriterium erfüllt ist (zum Beispiel Erreichen einer Höchstzahl an Untervolumen oder Unterschreiten einer Grenze für die maximale Varianz).

In den Untervolumina kann während des Aufteilungsprozesses weiter gewürfelt werden, um mit Verfeinerung des Rasters die Punktdichte zu erhöhen. Es kann gezeigt werden, dass die optimale Anzahl der Punkte N_k im Untervolumen gegeben ist durch die Bedingung:

$$\frac{N_k}{\sigma_k} = \text{const} \quad (5.25)$$

Ziel des Algorithmus ist es eine Unterteilung in Volumen mit gleicher Varianz zu finden. Dann kann in alle Volumen die gleiche Anzahl an Punkten gewürfelt werden.

Das Integral ergibt sich dann wie in (5.17), mit zusätzlicher Summation über m Untervolumen:

$$I \approx \sum_{k=1}^m \frac{V_k}{N_k} \sum_{j=1}^{N_k} f(\vec{x}_j) = \sum_{k=1}^m V_k \bar{f}_k \quad (5.26)$$

Der geschätzte Fehler des Integrals ist dann:

$$\sigma_I = \sqrt{\sum_{k=1}^m V_k^2 \frac{\sigma_{f_k}^2}{N_k}} \quad (5.27)$$

Die Formel zeigt, dass der absolute Fehler mit der Verkleinerung der Varianzen sinkt (der relative Fehler fällt bei gegebener Volumenaufteilung weiterhin wie $1/\sqrt{N}$).

Eine typische Zerlegung eines Integrationsbereiches durch einen auf "stratified sampling" basierenden Monte-Carlo-Integrationsalgorithmus (Divonne) ist in Abb. 5.4 dargestellt.

Beispiel: Den Effekt des 'stratified samplings' kann man gut an folgendem Beispiel klarmachen: Man betrachte eine lineare Funktion $f(x) = x$ in dem Interval $[0, 1]$ und berechne das Integral durch Würfeln von gleichverteilten x -Werten und Summation der Funktionswerte. Das Würfeln soll einmal im gesamten Interval $[0, 1]$ und dann getrennt in den Intervallen $V_a = [0, 0.5]$ und $V_b = [0.5, 1]$ ausgeführt werden.

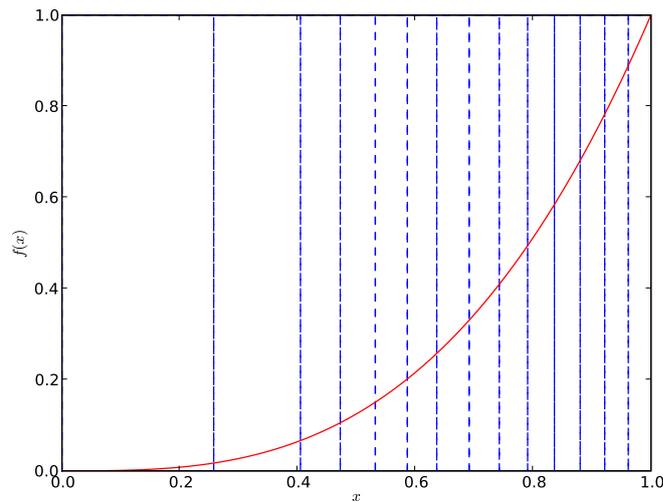


Abbildung 5.4: Zerlegung des Integrationsbereiches durch den Monte-Carlo-Integrationsalgorithmus Divonne (stratified sampling) für die Funktion $f(x) = x^3$.

Die Varianz der Funktion im Intervall $[x_1, x_2]$ ist $\sigma_f^2 = (x_2 - x_1)^2/12$. Die Varianz des Integrals ohne Unterteilung in Untervolumen ist dann

$$\sigma_{I,(1)}^2 = \frac{1}{12N} \quad (5.28)$$

Durch Unterteilung in zwei Volumen erhalten wir $\sigma_a^2 = \sigma_b^2 = 1/48$. Die Varianz des Integrals berechnet sich dann aus Gl. (5.27) zu

$$\sigma_{I,(2)}^2 = \left(\frac{1}{2}\right)^2 \frac{\sigma_a^2}{N/2} + \left(\frac{1}{2}\right)^2 \frac{\sigma_b^2}{N/2} = \frac{1}{48N} \quad (5.29)$$