Monte Carlo Simulation

M. Alexander Thomas

1. Juni 2006

Zusammenfassung

Die Monte Carlo Methode ist ein in vielen Bereichen nicht mehr wegzudenkendes Hilfsmittel zur Berechung und Simulation wissenschaftlicher Fragestellungen. Und gerade im Bereich der statistischen Physik wären viele Erfolge, ohne diese Methode kaum oder garnicht erreicht worden. Allerdings sind wie bei jedem Hilfsmittel, sei es nun ein Schraubenzieher oder wie hier die Monte Carlo Methode, ein paar wichtige Vorüberlegungen notwendig.

Abbildung 1: [Fotografie di Monte Carlo, 2006]

Inhaltsverzeichnis

1 Motivation

Die Monte Carlo Methode führt man mit einem Kanonischen Ensemble ein, da jede Problematik, die mit dieser Methode berechnet werden kann, auf ein Kanonisches Ensemble zurückgeführt werden kann.

Betrachtet man nun die Wahrscheinlichkeitsdichtefunktion $\mathfrak{p}(p^N, q^N)$ mit den beiden N-Vektoren für die Impulse und Orte der N Zustände bzw. Objekte, z.B. $p^N = (p_{1x}, p_{1y}, p_{1z}, p_{2x}, \ldots, p_{Nx}, p_{Ny}, p_{Nz},)$ für N Teilchen im 3-dimensionalen Raum.

$$\mathfrak{p}\left(p^N, q^N\right) = \frac{\exp\left(-\beta\mathcal{H}\left(p^N, q^N\right)\right)}{\int \exp\left(-\beta\mathcal{H}\left(p^N, q^N\right)\right)\, \mathrm{d}p^N\, \mathrm{d}q^N} \tag{1}$$

Wobei $\mathcal{H}(p^N, q^N)$ der Hamiltonoperator ist, der sich nach den kinetischen und den potentiellen Anteilen aufteilen lässt.

$$\mathcal{H} = \mathcal{T}(p^N) + \mathcal{U}(p^N, q^N) \tag{2}$$

Da jetzt $\mathcal{T} \sim (p^N)^2$ und mit Einschränkung auf Potentiale, die nicht von der Geschwindigkeit abhängen $\mathcal{U}(p^N, q^N) = \mathcal{U}(q^N)$ vereinfacht sich der Boltzmannfaktor auf einen Term, der nur von q^N abhängt und man kann die Impulse analytisch ausintegrieren, wenn man den Mittelwert einer Observablen $A(q^N)$ berechnen will:

$$\langle A\rangle = \frac{\int \exp\left(-\beta\mathcal{T}\left(p^N\right)\right)\mathrm{d}p^N \int \exp\left(-\beta\mathcal{U}\left(q^N\right)\right)A\left(q^N\right)\, \mathrm{d}q^N}{\int \exp\left(-\beta\mathcal{T}\left(p^N\right)\right)\mathrm{d}p^N \int \exp\left(-\beta\mathcal{U}\left(q^N\right)\right)\mathrm{d}q^N} \tag{3}$$

vereinfacht sich zu

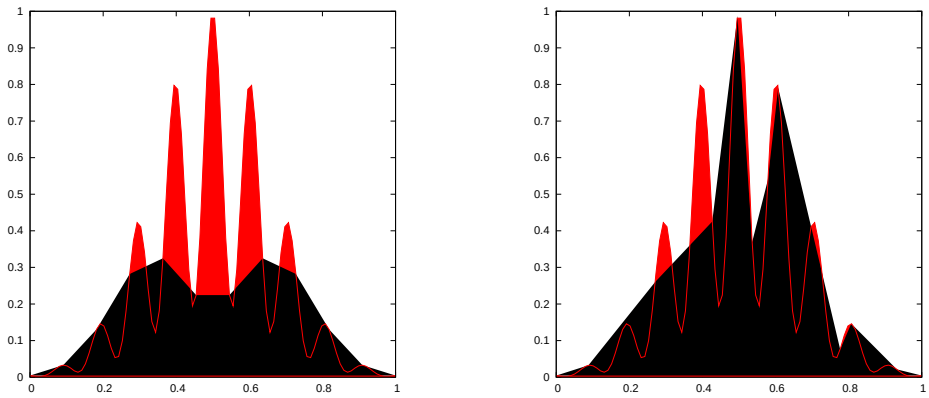
$$\langle A\rangle = \frac{\int \exp\left(-\beta\mathcal{U}\left(q^N\right)\right)A\left(q^N\right)\, \mathrm{d}q^N}{\underbrace{\int \exp\left(-\beta\mathcal{U}\left(q^N\right)\right)\mathrm{d}q^N}_{=Z}} \,. \tag{4}$$

Aber auch nachdem man die möglichen Potentiale eingeschränkt hat, können diese in einer analytisch nicht lösbaren Form vorliegen. Der nächste Schritt wäre demnach die Zustandssumme Z und die Observable numerisch zu integrieren. Will man aber dieses Integral mit einfachen numerischen Methoden wie zum Beispiel mit der Simpsonregel berechnen, bekommt man Probleme, da man selbst bei Einschrńkung auf wenige Stützstellen und Teilchenzahlen sehr viele Iterationen zu berechnen hat. Will man z.B. ein Ensemble mit 100 Teilchen berechnen und beschränkt sich zudem auf nur 5 Stützstellen, kommt man trotzdem auf 10^{210} Iterationen, die zu berechnen illusorisch sind![1]

[1] Auch muss man sich überlegen, inwieweit man das System vereinfachen kann, denn selbst mit den modernsten Computer dürfte es äusserst schwer sein, wenn nicht gar

2 Einführung der Monte Carlo Methode

Und für solche Probleme ist Monte Carlo genau die richtige Methode, um mit einer relativ kleinen Anzahl von Iterationen[2] dennoch ein genügend genaues Ergebnis zu erhalten. Dabei sind allerdings einige wichtige Dinge zu beachten. Dazu betrachtet man zuerst, ausgehend von einer stark oszillierenden Funktion, die Methode der numerischen Quadration über feste bzw. zufällige Stützstellen, um das Integral und somit die Fläche zu berechnen. Wie man in Abbildung 2 sieht, ist das Problem das gleiche geblieben. Man



(a) Feste Intervallbreite

(b) Zufällige Intervallbreite

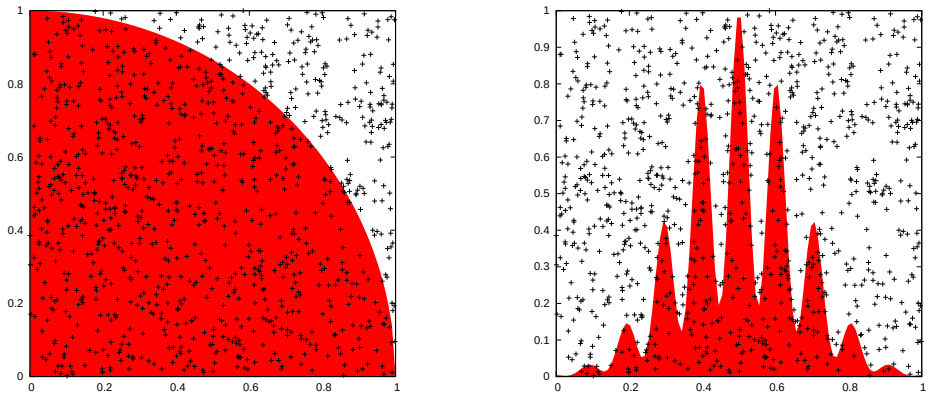
Abbildung 2: Einfluss der Stützstellen

hat zwar zufällige Stützstellen, aber dennoch können die auch sehr schlecht sein. Zudem muss man auch diese alle berechnen, was wieder zum Problem der vielen Iterationen zurückführt. Deswegen mögen die gerade besprochenen Vorgehensweisen bei zweidimensionalen Problemen vielleicht noch sehr hilfreich sein, stossen aber bei höherdimensionalen Berechnungen schnell an ihre Grenzen.

Ein anderer Ansatz wäre keine Stützstellen zu wählen, sondern die Funktion mit zufällig gesetzten Punkten abzutasten. In der folgenden Abbildung 3 sieht man, wie man so versucht die Fläche unter der Funktion zu bestimmen und damit das Integral. Wenn man sich jetzt beide Funktionen graphisch betrachtet, erkennt man sehr schnell, warum die MC Methode einen rela-

unmöglich Systeme zu beschreiben, die mehr als 10^9 Teilchen beinhalten. Da diese Teilchen einen $6N$-dimensionalen Phasenraum aufspannen und jede Koordinate je 4 Byte Speicherplatz belegt, braucht man dafür schon einen Rechner mit mehr als 22 Gigabyte Hauptspeicher. Dies nur als Randbemerkung, um sich die Grössenverhältnisse vor Augen zu halten. Realistischere Ensembles mit einem Mol (10^{23}) oder mehr rücken damit in weite Ferne.

[2] ca. 10^6 - 10^7 Iterationen



(a) Kreisausschnitt

(b) Kompliziertere Fkt

Abbildung 3: Zwei verschiedene Funktionen im Vergleich

tiv grossen Fehler produziert. Der Flächeninhalt wird über die Anzahl der Punkte unterhalb der Funktion gegenüber den Gesamtpunkten berechnet. Und anhand der Abbildung 3 ist ersichtlich, dass sich zwar in der linken Graphik[3] viele Punkte unterhalb der Funktion befinden und die Wahrscheinlichkeit sehr hoch ist auch die Funktion zu treffen. Bei der linken dagegen ist die Wahrscheinlichkeit viel grösser die Funktion zu verfehlen und damit wird diese nicht mehr ausreichend mit Punkten abgedeckt, was wiederum zu einem grossen statistischen Fehler führt. Betrachtet man nun weiter die linke Abbildung[4], ergibt sich nach analytischer Integration für das Interval von 0 bis 1 der Wert $F(x)|_0^1 = 0.237343$, aber man erhält mit der einfachen MC Methode $F(x)|_0^1 = 0.255 \pm 2 \cdot 10^4$ bei 1000 Punkten.

3 Importance Sampling

Nun kann man sich überlegen, ob man nicht vielleicht die Punkte geschickter verteilen könnte, da die äusseren Bereiche deutlich weniger zum Flächeninhalt beitragen als die inneren. Und mit dieser Überlegung kommt nun die sogenannte Gewichtungsfunktion ins Spiel, da nun die Punkte, die man weniger oft gestreut hat, wiederum schwerer ins Gewicht fallen als die Punkte, die man bevorzugt hat.

Mathematisch betrachtet handelt es sich dabei um eine Substitution der Form

$$F|_0^1 = \int f(w)\,\mathrm{d}w = \int \frac{f(x)}{w(x)}\,w(x)\,\mathrm{d}x. \tag{5}$$

[3] Berechnung der Kreiszahl π

[4] $f(x) = \mathrm{e}^{-20\left(x-\frac{1}{2}\right)^2} \frac{1+4\cos^2\left(30\left(x-\frac{1}{2}\right)\right)}{5}$

In der Abbildung 4 sieht man nun, wie die Punkte nach der Funktion

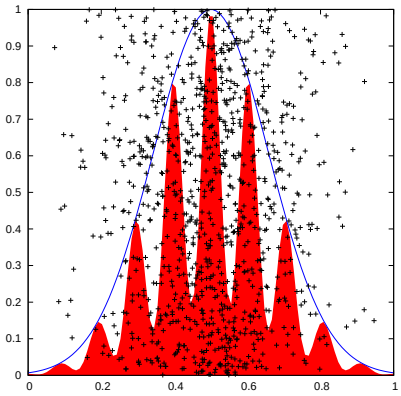
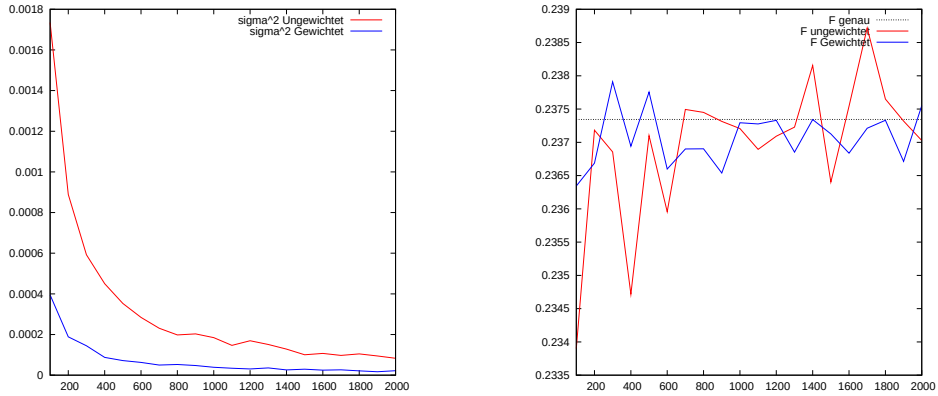


Abbildung 4: Gewichtete Verteilung

$$w(x) = \mathrm{e}^{-20\left(x-\frac{1}{2}\right)^2} \tag{6}$$

gewichtet wurden. Man erhält daraufhin für das Integral: $F(x)|_0^1 = 0.236635$ $\pm 3 \cdot 10^5$ bei 1000 Punkten.



(a) Varianzen

(b) Abweichungen

Abbildung 5: Beide Methoden im Vergleich

Abbildung 5 zeigt die Varianz der Methoden aufgetragen auf die Anzahl der Samplepunkte. Dazu wurden je Punktzahl 300 Durchläufe durchgeführt und darüber die Varianz und das Integral gemittelt.

Aber da man sich in der statistischen Physik weniger für die Werte bestimmter Observablen an bestimmten Orten bzw. Zuständen als für deren Mittelwerte interessiert, kann man das Metropolis Schema benutzen, da es

bei diesem nicht nötig ist, für die Berechnung der Mittelwerte die Integrationen durchzuführen.

4 Detailed Balance

Jetzt betrachten wir noch die Bedingung, dass sich das zu simulierende System im Gleichgewichtszustand befinden muss genauer. Geht nämlich das System von einem alten Zustand o in einen neuen Zustand n über, geschieht dies mit einer gewissen Wahrscheinlichkeit π. Nun muss im Gleichgewichtszustand jeder Zustand im Mittel in gleich vielen Schritten erreicht werden können. Wäre dies nicht der Fall, wäre ein Zusand bevorzugt zu den anderen und damit wäre das System nicht im Gleichgewichtszustand. Man kann also schreiben, dass die Wahrscheinlichkeitsdichte des alten Zustands mit der Wahrscheinklichkeit vom alten in den neuen Zustand zu gelangen gleich sein muss wie die Wahrscheinlichkeitsdichte des neuen Zustands mit der Wahrscheinlichkeit vom neuen in den alten Zustand zu gelangen.

$$\mathfrak{p}(o)\pi(o \longrightarrow n) = \mathfrak{p}(n)\pi(n \longrightarrow o) \tag{7}$$

Wenn man nun versucht die Wahrscheinlichkeit für den Übergang $o \to n$ zu bestimmen, schreibt man π um:

$$\pi(o \longrightarrow n) = \alpha(o \longrightarrow n) \times \mathrm{acc}(o \longrightarrow n) \tag{8}$$

Wobei α die zugrundeliegende Markovkette[5] repräsentiert und $\mathrm{acc}(o \to n)$ die Wahrscheinlichkeit bezeichnet, nachdem ein Übergang akzeptiert wird. Wählt man nun α symmetrisch ist, kann man Gleichung (8) umschreiben zu

$$\mathfrak{p}(o) \times \mathrm{acc}(o \longrightarrow n) = \mathfrak{p}(n) \times \mathrm{acc}(n \longrightarrow o). \tag{9}$$

Durch Umstellen der Gleichung (9) folgt die später wieder auftauchende Gleichung(10):

$$\frac{\mathrm{acc}(o \longrightarrow n)}{\mathrm{acc}(n \longrightarrow o)} = \frac{\mathfrak{p}(n)}{\mathfrak{p}(o)} = \exp\left(-\beta \Delta U\right) \tag{10}$$

Aus Erfahrung[6] wählt man für $\mathrm{acc}(o \to n)$:

$$\frac{\mathrm{acc}(o \longrightarrow n)}{\mathrm{acc}(n \longrightarrow o)} = \begin{cases} \frac{\mathfrak{p}(n)}{\mathfrak{p}(o)} & \text{für} \quad \mathfrak{p}(n) < \mathfrak{p}(o) \\ 1 & \text{für} \quad \mathfrak{p}(n) \geq \mathfrak{p}(o) \end{cases} \tag{11}$$

Um dies nun tatsächlich zu erreichen, wählt man die Punkte, die man verschieben will per Zufall aus. Damit ist gegeben, dass es eine von Null verschiedene Wahrscheinlichkeit gibt, dass der gleiche Punkt noch einmal ausgewählt und an seinen ursprünglichen Platz zurückverschoben wird[7].

[5] Eine Markovkette ist gegeben, wenn ein neuer Zustand nur von dem Vorhergehenden abhängt und unabhängig von den vorangegangen Zuständen ist.

[6] Siehe Fussnote 6 auf Seite 6.

[7] Das macht die Benutzung eines echten Zufallszahlengenerators unumgänglich. Siehe Kapitel 7.

5 Metropolis Schema

Man betrachte dazu das vielzitierte Beispiel für die Berechung der Tiefe des Nils indem man über ganz Afrika integriert in Abbildung 6.

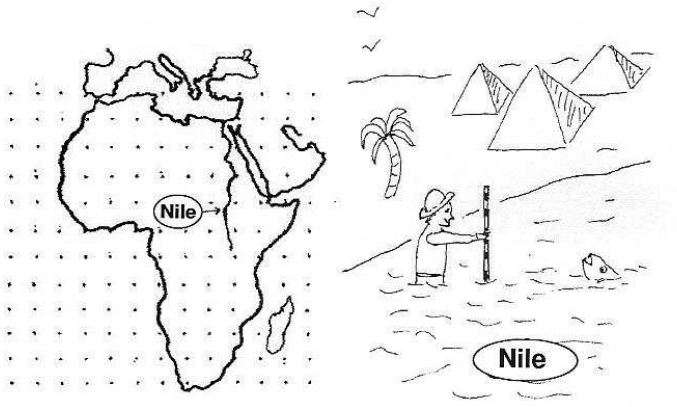


Abbildung 6: Bestimmung der Tiefe des Nils, im Vergleich rechts die numerische Quadration und links das Metropolis Verfahren[Frenkel, 2004, Seite 32f].

Denn genau dies entspricht den vorher besprochenen Verfahren. Nun kann man sich überlegen, ob man nicht die Anzahl der Samplepunkte variabel gestalten könnte, d.h. man startet an einem zufälligen Punkt des Phasenraums und lässt sich per Zufall einen Schritt vorgeben. Ist das Potential nach diesem Schritt kleiner als vorher, akzepiert man ihn, andernfalls verwirft man ihn mit $q(x_{neu}) = 1 - \exp(-\beta\Delta U)$. Man darf den Schritt nicht gleich sofort verwerfen, da man sonst, leicht ersichtlich, in einem lokalen Minimum hängen bleiben kann, deshalb die von Null verschiedene Wahrscheinlichkeit $p(x_{neu}) = \exp(-\beta\Delta U)$[8], dass dieser Schritt doch ausgeführt wird[9]. Wichtig ist dabei, dass man wirklich jeden Punkt sampelt, auch wenn man diesen Schritt verworfen hat.

Die grundlegende mathematische Überlegung dazu ist, dass im Mittel die Anzahl der Punkte n_i, die pro Einheitsphasenraumvolumen um den Punkt q^N erzeugt wurden gleich der Gesamtpunktzahl M multipliziert mit der Wahrscheinlichkeitsdichtefunktion $\mathfrak{p}$ ist.

$$n_i = M\mathfrak{p}\left(r^N\right) \tag{12}$$

Damit kann man die Mittelwertberechnung umschreiben und die Integration

[8] Gleichung 10

[9] Man kann auch andere Wahrscheinlichkeitsfunktionen einsetzen, allerdings hat sich diese am meisten bewährt.

durch eine Summation annähern.

$$\langle A \rangle = \frac{1}{Z} \int \exp\left(-\beta \mathcal{U}\left(q^N\right)\right) A\left(q^N\right)\, \mathrm{d}q^N = \int A\left(q^N\right) p\left(q^N\right)\, \mathrm{d}q^N \qquad (13)$$

wird zu

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^{M} n_i A\left(q_i^N\right). \qquad (14)$$

Da aber das Metropolis Schema die Punkte genau nach der Wahrscheinlichkeitsdichtefunktion zeichnet, vereinfacht sich die Formel letztendlich auf das algebraische Mittel aller gezeichneten Werte der Observablen.

Nun nachdem man diese Methoden verstanden hat, kann man die MC Simulation betrachten, in der das Kanonisches Ensemble, also ein System im Gleichgewichtszustand, betrachtet wird.

6 Monte Carlo Simulation

Wenn man nun die oben besprochene Methode anwendet, kommt man auf den einfachen Algorithmus der Monte Carlo Simulation. Schon bei Betrachtung der Gleichung (7) auf Seite 7 sieht man, dass der Algorithmus aus mindestens einer Schleife bestehen muss. Am besten sieht man man sich den Algorithmus mit einem Pseudo-C++-Code Schnipsel an:

```
10 while (Laufzeitbedingung)
11 {
12     gew_Punkt = Random();
13     Alter_Punkt = Punkte[gew_Punkt];
14     Punkte[gew_Punkt] += RandomMove(Delta_q);
15     if (Random() > exp(-beta*Delta_U(Alter_Punkt, Punkte[gew_Punkt])))
16         Punkte[i] = Alter_Punkt;
17     if (!(Iterationen % Tastung))
18         Sample(Punkte[gew_Punkt]);
19 }
```

Was passiert nun genau in diesem Programmfragment? Wenn man ihn Schritt für Schritt durchgeht, dann sieht man, dass eine grosse **while**-Schleife durchlaufen wird, solange die Laufzeitbedingung wahr ist. Die Laufzeitbedingung kann z.B. sein, dass deterministisch n Durchläufe stattfinden oder eine gewisse Genauigkeit erreicht wird. Dann wird ein beliebiger Punkt in dem Ensemble ausgewählt, die Position gespeichert und um einen zufälligen Betrag $\pm Delta\_q/2$ verschoben, anschliessend wird eine Zufallszahl zwischen 0 und 1 gezogen und der Sprung wird rückgängig gemacht, wenn die Zufallszahl grösser ist als der Boltzmannfaktor. Am Ende der Schleife wird alle $Tastung$ Punkte der Mittelwert der Observablen gesampled.

7 Zufallszahlen und Generatoren

Im vorangegangen Kapitel ist stillschweigend davon ausgegangen worden, dass die benutzten Zufallszahlen auch wirklich zufällig verteilt sind. Dem Computer als streng deterministische Maschine ist es immanent unmöglich echte Zufallszahlen zu liefern. Er liefert mehr oder weniger gute Pseudozufallszahlen, die einer Reihe von modulo-Rechnungen entspringen. Setzt man den Startwert bei jedem Durchlauf gleich, erhält man jedesmal die gleiche Folge von Zufallszahlen. Nun kann man das hinnehmen und hinreichend gute

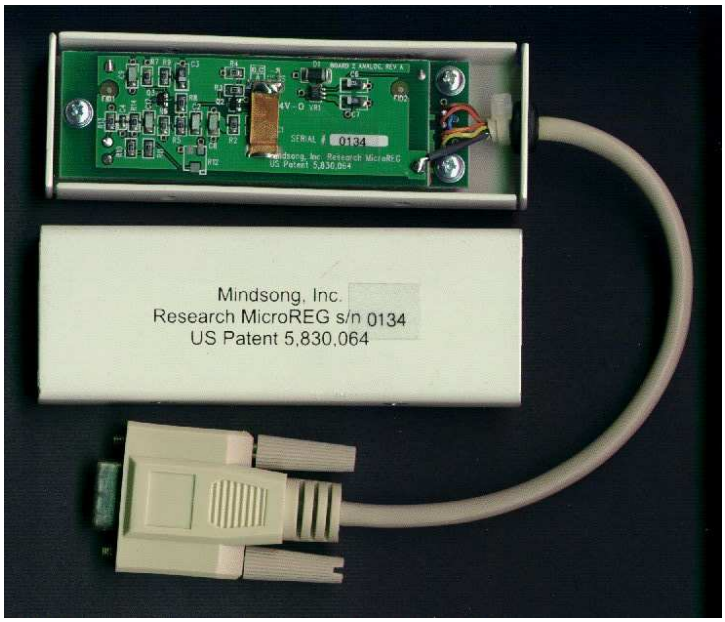


Abbildung 7: MOS-FET Zufallszahlengenerator[REG Design]

Zufallszahlengeneratoren benutzen, diese bei bei jedem Start mit einer anderen (Zufalls-)Zahl starten lassen, z.B. Uhrzeit oder gerade ankommendes Netzwerkpaket[10]. Oder man führt eine Messung an einem zufallsverteilten System aus und nimmt diese Werte als Zufallszahl. Dies könnte z.B. ein radioaktives Präparat sein. Diese liefern aber meist nicht in hinreichender Geschwindigkeit neue Zufallszahlen, so dass man diese gelieferten Zufallszahlen als „Seed“ für den Pseudozufallszahlengenerator benutzen kann, der dann zwar wieder nur Pseudozufallszahlen liefert, aber an einem zufälligen Startpunkt der Folge.

Wer sich mehr für dieses Thema interessiert, dem sei die Wikipedia-Seite[Wikipedia] oder das Buch von K.P.N. Murthy[Murthy, 2004, Seite 9ff] empfohlen.

[10] Z.B. /dev/urandom unter Linux.

8 Literatur

Literatur

[Fotografie di Monte Carlo, 2006] Fotografie di Monte Carlo, (2006). *Fotografie di Monte Carlo*. http://www.fotoeweb.it/montecarlo/index.htm.

[Frenkel, 2004] Frenkel, D., (2004). *Introduction to Monte Carlo Methods*. http://www.fz-juelich.de/nic-series/volume23/frenkel.pdf.

[Wikipedia] Wikipedia, 30.05.2006. *Hardware random number generator*. http://en.wikipedia.org/wiki/Hardware_random_number_generator.

[REG Design] Global Consciousness Project, Princton, 30.05.2006. *REG Design*. http://noosphere.princeton.edu/reg.html.

[Murthy, 2004] Universities Press, Hyderabad, 2004. *Monte Carlo Methods in Statistical Physics*. 3-5-819 Hyderguda, Hyderabad 500 029.