

## § 77: PSEUDOZUFALLSZAHLEN UND MONTE-CARLO-SIMULATION

### 77.1. Motivation

Es gibt Anwendungsgebiete (z.B. Ray Tracing in der Computergrafik, Strömungssimulation im Bereich Computational Fluid Dynamics, Berechnung hochdimensionaler Integrale), bei denen stochastische Simulationen einfache oder effiziente Alternativen zu deterministischen Algorithmen sind. Solche Simulationen nennt man auch Monte-Carlo-Verfahren. Sie benötigen die Erzeugung von Zufallszahlen auf dem Rechner. Da funktionierende Computer jedoch deterministisch arbeiten, verwendet man statt dessen Algorithmen, die Zahlen liefern, die echten Zufallszahlen ähneln. Solche Zahlen nennt man Pseudozufallszahlen.

### 77.2. Erzeugung von gleichverteilten Pseudozufallszahlen

Ziel: Erzeugung einer Sequenz  $Z_n$  von gleichverteilten Pseudozufallszahlen aus  $[0,1]$ .

Beliebte Vorgehensweise: Lineare Kongruenzmethoden  
(in vielen Compilern verwendet)

Geg.:  $m \in \mathbb{N}$  Modulus  
 $a \in \{1, \dots, m-1\}$  Multiplikator  
 $b \in \{0, \dots, m-1\}$  Inkrement  
 $x_0 \in \{1, \dots, m-1\}$  Startwert

Verfahren:  $x_n := a \cdot x_{n-1} + b \pmod{m}$

$$z_n := x_n / m$$

Dann approximiert die Sequenz  $(z_n)$  eine Folge von stochastisch unabh. Zufallsvariablen, die auf  $[0, 1]$  gleichverteilt sind.

Die Approximationsgüte hängt von der Parameterwahl ab.

Klas: Nach spätestens  $m$  Schritten wiederholt sich die Folge.

Häufig verwendet, aber schlecht:

$$m = 2^{16} = 65536$$

$$a = 25173$$

$$b = 13849$$

Besser (Minimalstandard):

$$m = 2^{31} - 1 = 2.147.483.647$$

$$a = 7^5 = 16807$$

$$b = 0.$$

### 77.3. Erzeugung von $N(0, 1)$ -verteilten Pseudozufallszahlen

Die Standardnormalverteilung im  $\mathbb{R}^2$  hat die Dichte

$$\varphi(x, y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}}$$

Für eine Kreisscheibe  $K_r(0)$  um 0 mit Radius  $t$  und zwei stoch. unabh.  $N(0, 1)$ -verteilte Variablen  $X, Y$  gilt:

$$P(X^2 + Y^2 \leq t^2) = \int_{K_r(0)} \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} dx dy$$

Gehet man zu Polarkoordinaten  $(r, \varphi)$  über, ergibt sich mit

$$x = r \cos \varphi =: f_1(r, \varphi)$$

$$y = r \sin \varphi =: f_2(r, \varphi)$$

$$Jf(r, \varphi) = \begin{pmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial \varphi} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial \varphi} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -r \sin \varphi \\ \sin \varphi & r \cos \varphi \end{pmatrix}$$

$$\det(Jf(r, \varphi)) = r \cos^2 \varphi + r \sin^2 \varphi = r$$

und der Transformationsregel (§ 60):

$$\begin{aligned}
P(X^2 + Y^2 \leq t^2) &= \int_{\varphi=0}^{2\pi} \int_{r=0}^t \frac{1}{2\pi} e^{-\frac{r^2}{2}} r \, dr \, d\varphi \\
&= \cancel{2\pi} \int_{r=0}^t \frac{1}{\cancel{2\pi}} e^{-\frac{r^2}{2}} r \, dr \\
&= \left[ e^{-\frac{r^2}{2}} \right]_0^t = 1 - e^{-t^2/2}
\end{aligned}$$

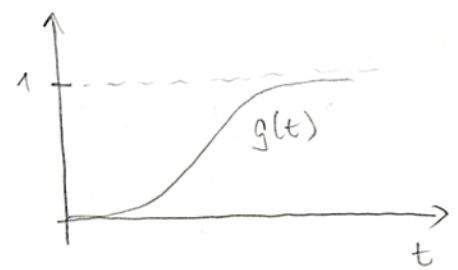
Der Radius  $R$  des Zufallszahlenvektors  $(X, Y)$  erfüllt also

$$P(R \leq t) = 1 - e^{-t^2/2} =: g(t)$$

Wertebereich von  $g$ :  $[0, 1)$ .

Umkehrfunktion zu  $g(t)$ :

$$g^{-1}(z) = \sqrt{-2 \ln(1-z)}$$



Hat man eine in  $[0, 1]$  gleichverteilte Zufallsvariable  $Z_1$ , ergibt sich als Zufallsvar.  $R$  für den Radius der 2D-Standardnormalverteilung:

$$R = \sqrt{-2 \ln(1-Z_1)}$$

Aus einer zweiten, auf  $[0,1]$  gleichverteilten Zufallsvariable  $Z_2$  erhält man als Zufallsvariable für einen auf  $[0, 2\pi]$  gleichverteilten Winkel  $T$ :

$$T = 2\pi Z_2.$$

Dies motiviert folgenden Algorithmus (Box-Muller-Verfahren) zur Erzeugung zweier  $N(0,1)$ -verteilter Zufallszahlen  $X, Y$  aus zwei auf  $[0,1]$  gleichverteilten Zufallszahlen  $Z_1, Z_2$ :

$$\begin{aligned} R &:= \sqrt{-2 \ln(1-Z_1)} \\ T &:= 2\pi Z_2 \\ X &:= R \cos T \\ Y &:= R \sin T \end{aligned}$$

Bem.: Benötigt man  $N(\mu, \sigma^2)$ -verteilte Zufallszahlen  $\tilde{X}, \tilde{Y}$ , setzt man (vgl. 65.15).

$$\tilde{X} = \mu + \sigma X$$

$$\tilde{Y} = \mu + \sigma Y$$

(Pseudo-) Zufallszahlen benötigt man z. B. bei probabilistischen Algorithmen.

### 77.4. Beispiel: Quicksort

Sortieren einer Liste mit Quicksort:

- Suche ein zufälliges Element  $z$  der Liste  
(mit gleichverteilten Pseudozufallszahl aus 77.2)
- Sortiere Teilliste mit Elementen  $\leq z_1$  mit Quicksort
- " " " " " "  $> z_1$  " "
- usw.

Quicksort hat eine mittlere Laufzeit von  $O(n \log n)$ , ist also nicht schlecht.

### 77.5. Beispiel: Buffon'sches Nadelexperiment

Ein probabilistischer Algorithmus zur Bestimmung von  $\pi$ :

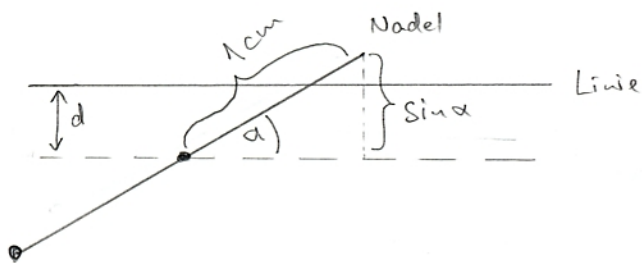
- Zeichne auf einem Blatt Papier parallele Linien im Abstand einer Stechnadellänge
- Lasse die Stechnadel auf das Papier fallen und überprüfe, ob sie eine der Linien trifft. Denke dabei fest an  $\pi$ .
- Zähle die Zahl  $N$  der Versuche und die Zahl  $T$  der Treffer. Dann gilt:

$$\frac{N}{T} \rightarrow \frac{\pi}{2} \quad \text{für } N \rightarrow \infty.$$

Warum funktioniert dies?

Ann.: Nadellänge und Linienabstand seien  $2$  [cm].

Nadel kann nur nächstgelegene Linie schneiden und nur dann ist Abstand  $d$  zwischen Nadelmittle und Linie  $d < 1$  erfüllt:

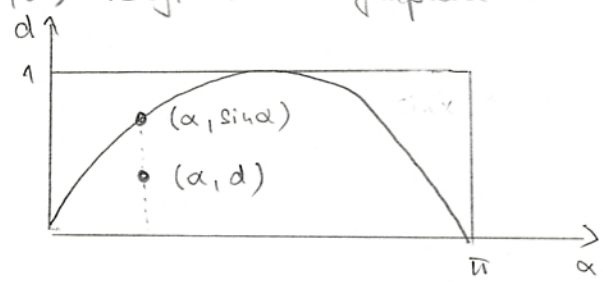


$$0 \leq \alpha \leq \pi$$

Nadel schneidet Linie  $\Leftrightarrow d < \sin \alpha$

Zu jedem Wurf gehört ein Wertepaar  $(\alpha, d) \in [0, \pi] \times [0, 1]$ .

$d < \sin \alpha \Leftrightarrow (\alpha, d)$  liegt unter Graphen von  $\sin x$



Im Zufallsexperiment sind die Punkte  $(\alpha, d)$  gleichverteilt auf  $[0, \pi] \times [0, 1]$ .

$$\Rightarrow \frac{T}{N} = \frac{\int_0^{\pi} \sin x \, dx}{\pi \cdot 1} = \frac{[-\cos x]_0^{\pi}}{\pi} = \frac{2}{\pi}$$

Unser Experiment war also ein stochastisches Integrationsverfahren für  $\int_0^{\pi} \sin x \, dx$ .

Im 1D Fall sind solche Verfahren ineffizient, aber bei hochdimensionalen Integrationsproblemen haben sie eine bessere Komplexität als deterministische numerische Ansätze.