# 3. Stochastik : Wahrscheinlichkeitstheorie und Statistik (siehe Kapitel 1)

Jetzt: Zufallszahlen und Anwendungen
Zufall: Ereignisse ohne kausalen Zusammenhang
Viele Prozesse in der Natur sind stoch. Prozesse,
beruhen also auf Zufallsereignissen

wichtige Anwendungen von Zufallszahlen:
1) Kryptographie
2) statistische Simulationen,
d.h. Simulation von stoch. Prozessen und Anwendung von Statistik

## 3.1 Zufallszahlengeneratoren (RNG - random number generator) Erzeuge eine Folge von Zahlen, die möglichst zufällig sind

### 3.1 .1 Echte Zufallszahlen

→ verwende Prozesse aus der Natur, die stochastisch sind
*) Radioaktive Zerfall (sehr langsam)
*) elektronisches Rauschen: Bsp. Intel Prozessoren (ab Ivy Bridge Generation)
eingebauter Zufallszahlengenerator (RdRand) (NIST (NSA?))
*) Lottozahlen, ...

### 3.1 .2 Pseudozufallszahlengeneratoren ( PRNG - pseudo random number generator)

Benutze deterministischen Algorithmus um die Folge von Zufallszahlen zu generieren.
deterministisch: reproduzierbare Zufallszahlen
Initialisierung durch Startwert (SEED)
produziert Zufallszahlen im Bereich [Min, Max]

Eigenschaften von PRNG:
*) Periode: wann wiederholt sich die Folge?
*) Geschwindigkeit: wieviele Zufallszahlen pro Sekunde?
*) Qualit t: wie "zuf llig" sind die Zahlen?
→ statistische Tests mit den Zufallszahlen

### 3.1 .3 Zufallszahlen - Tests

A) Normierung und Verteilung: typ. Gleichverteilung
→ Histogramm (numpy .histogram (), plplot.hist())

B) Spektraltest/Parking-Lot Test
Bilde Paare/Tripel von Zufallszahlen erzeugt und als 2D/3D-Plot darstellt

C) Korrelationstest: Unabh ngigkeit der Zufallszahlen

Bedingung: $P(z_i, z_{i+1}) = P(z_i)\, P(z_{i+1})$

$$< z_i\, z_{i+1} > = \sum_{i=1}^{N} \sum_{j=1}^{N} z_i\, z_j\, P(z_i, z_j) = < z_i > < z_{i+1} >$$

$$< r_i, r_j >_{corr} = < z_i\, z_{i+1} > - < z_i > < z_{i+1} >$$

D) vieleweitereTests, z.B. sieheDieharder-Paket

## 3.1 .4 Wichtige PRNG

1) Nachkommastellen von irrationalenZahlen, z.B. $\pi$, $\sqrt{2}$, e

2) LinearenKongruenzgenerator

$Z_{i+1} = (a\, z_i + b) \bmod m$

Wahl von a, b, m :
Bereich ist [0, m - 1]
a, b sehr mystisch
Problem : Hyperfl chenproblem

3) Inversen Kongruenzgenerator
→ aufwendiger, aber kein Hyperfl chenproblem

4) Mersenne-Twister (MT19937) - twistedgeneralizedfeedback shift-register
Periode: $2^{19937} - 1 = 10^{6000}$
komplexer Alg., aber schnell
→ nicht f r Kryptographie geeignet, aber sehr gut f r stat. Simulationen

5) Sonstige PRNGs (Tausworth, GFSR4)

## 3.1 .5 Implementierungen

Betriebssystem /C
/dev/random (mit Entropie)
/dev/urandom
stdlib.h
lrand48 (),
drand48 () → LineareKongruenzgeneratoren mit m = $2^{48}$ lifernZahlen [0, RAND_MAX]
rand (), srand () (Algorithmus abh ngig von Betriebssystem )
random () = rand () ("non-linear additive feedback" ?? ?)
→ alle ziemlich schlecht f r stat. Simulationen

GSL: gsl_rng_mt19937 , gsl_rng_taus2, gsl_rng_gfsr4, ...

Python: MT19937 ("from random import random ")
random () [0, 1]
uniform (Min, Max) [Min, Max]
randint (Min, Max) [Min, Max] ∈ Z
Startwert (SEED) : normalerweise Systemzeit (ns) oder "seed (int)"

## 3.1 .6 Nichtgleichverteilte Zufallszahlen

→ vieleAnwendungenbrauchen Zufallszahlenmit einerbestimmten Verteilung
→ ausgehend von gleichverteilten
Zufallszahlen generiereZufallszahlenmit einerVerteilung

A) Invertierungsmethode

Transformation einerWahrscheinlichkeitsverteilung $q(x) \rightarrow$
$p(z)$ durch Substitution $x = x(z)$

$$1 = \int_{-\infty}^{\infty} q(x)\,dx = \int_{-\infty}^{\infty} q(x(z)) \left|\frac{dx}{dz}\right| dz$$

$$\text{d.h. } p(z) = q(x(z)) \left|\frac{dx(z)}{dz}\right|$$

$$\text{F r } q(x) \text{ gleichverteilt: } p(z) = \text{const.} \left|\frac{dx(z)}{dz}\right|$$

$$\int_{z_0}^{z} p(z')\,dz' = P(z) = \text{const.}\, x$$

$$z = \text{const.}\, P^{-1}(x)$$

z.B. Exponentialverteilung $p(z) = a e^{-az}\ (z >= 0)$

$$P(z) = \int_{0}^{z} p(z')\,dz' = 1 - e^{-az} = x$$

$$\rightarrow z = -\frac{1}{a}\text{Log}[1-x]$$

z.B. Lorentzverteilung $p(z) = \frac{\gamma}{\pi\left(\gamma^2 + z^2\right)}$

$$P(z) = \int_{-\infty}^{z} p(z')\,dz' = \frac{1}{2} + \frac{1}{\pi}\text{atan}\left(\frac{z}{\gamma}\right) = x$$

$$\rightarrow z = \gamma \tan\left(\pi\left(x - \frac{1}{2}\right)\right)$$

N chstes Mal: Verwerfungsmethode