Computerphysik II

Teil 3 - Stochastische Methoden

S. Gerlach

WiSe 2020/21

Universität
Konstanz

INHALT

Zufallszahlen

Echte Zufallszahlen (Würfeln, Radioaktiver Zerfall, etc.) sind aufwendig zu bekommen. Verwende also sog. Pseudozufallszahlen, die mittels det. Algorithmen berechnet werden. Diese Algorithmen nennt man Pseudozufallszahlengeneratoren (PRNG).
Wichtige Eigenschaften: (stat.) Qualität, Geschwindigkeit, Periode
Die Qualität lässt sich mit stat. Tests prüfen. Die Periode ist durch den Alg. festgelegt und die Geschw. hängt von der Implementierung ab.
Wichtige PRNGs: Kongruenzgeneratoren, Mersenne-Twister

Zufallszahlengenerator: Linearer Kongruenzgenerator

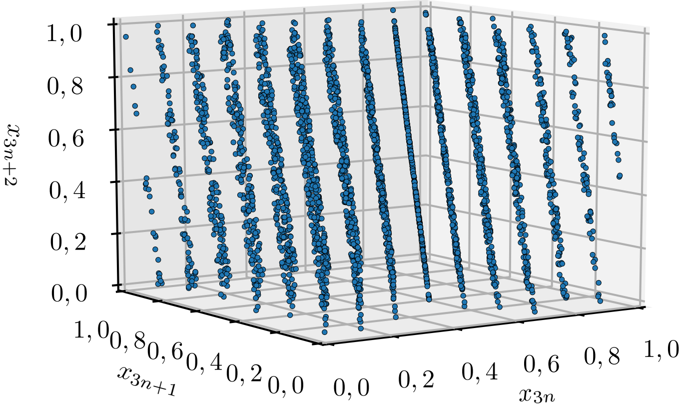
Iterative Berechung einr Folge von Zufallszahlen:

$$x_{n+1} = (ax_n + b) \mod m.$$

Parameter: a, b, m
x_0 ist der Startwert ("seed")
m bestimmt die Periode und je nach Wahl a und b schwankt die Qualität. Allgemein aber schlecht, da z. B. Hyperebenenverhalten.

Hyperebenenverhalten

3D Spektraltest des lin. Kongruenzgenerators für $a = 65539$, $b = 0$ und $m = 2^{31}$:



Mersenne-Twister

MT19937: Periode $2^{19937} - 1$, gute stat. Qualität

Wird deshalb oft in Monte-Carlo-Simulationen (MC) und Molekulardynamik (MD) verwendet.

Wird auch von Python verwendet:

`random.random()`: Zufallszahlen im Bereich $[0, 1)$

`random.uniform(a, b)`: Zufallszahlen im Bereich $[a, b)$

`random.randint(a, b)`: ganzzahlige Zufallszahlen im Bereich $[a, b)$

`random.seed(S)`: Startwert S setzen

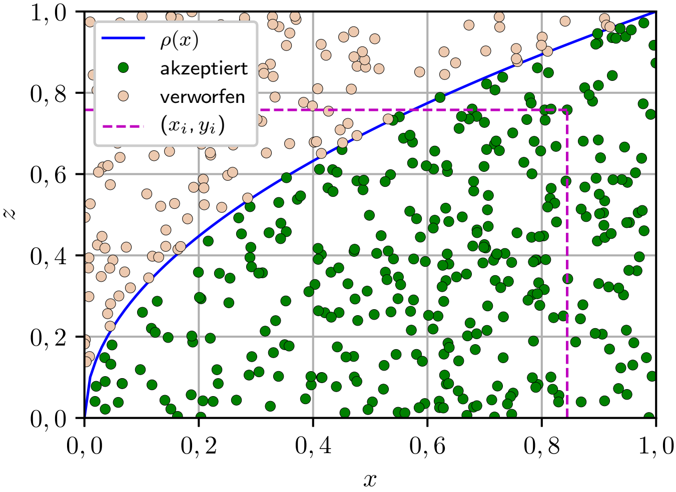
Zufallszahlenverteilungen

Bisher gleichverteilte Zufallszahlen. Jetzt: Zufallszahlen mit einer stat. Verteilung: wichtig für viele Anwendungen.
Methoden (s. Buch):

- Inversionmethode (Bsp. Exponentialvert., Lorentz-vert.)
- Verwerfungsmethode
- Box-Muller, Polarmethode, Ziggurat

Verwerfungsmethode

Erzeugung von Wurzelförmigverteilt. Zufallszahlen



Monte-Carlo-Integration

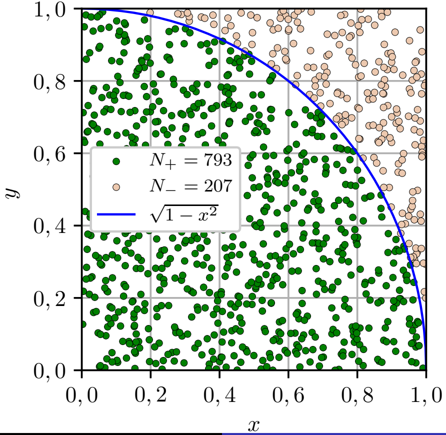
Bisher: Numerische Integration durch Verwendung von äqudistanten Stützstellen.
Jetzt: verwende zufällige Stützstellen x_i und summiere einfach die Funktionswerte, d. h.

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i).$$

MC-Integration

MC-Integration anhand der Steinwurfmethode für den Viertelkreis ($\sqrt{1-x^2}$)

$$\pi = 4A \approx 4\frac{N_+}{N} = 4\frac{793}{1000} = 3,172.$$



MC-Integration

Aufwand der numerischen Integration hängt von Dimension ab, nicht aber bei MC-Integration. Bei hochdimensionalen Integralen (z.B. Phasenraum) macht nur MC-Integration Sinn.

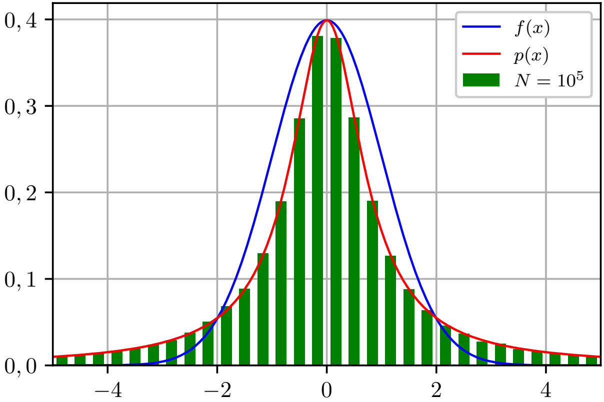
Importance Sampling:

Verwende Zufallszahlen, deren Verteilung etwa der zu integrierenden Funktion entsprechen. Sonst werden viele Zufallszahlen an Stellen erzeugt, wo die Funktion sowieso praktisch null ist.

Importance Sampling

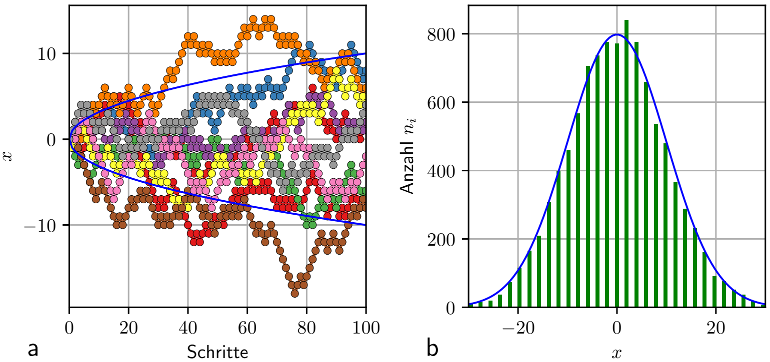
$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}.$$

Gauss-Integral mithilfe von Lorentz-vert. Zufallszahlen:



Random-Walk

Zufallsbewegung auf Gitter simuliert viele phys. Prozesse (Brownsche Bewegung, Langevin-Dynamik, Spindynamik, ...)
Eindimensionaler *Random Walk* für zehn verschiedene Beispiele (**b**)
Verteilung der Endposition nach 100 Schritten für 10.000 Durchläufe:



Zelluläre Automaten

Simuliere diskrete dynamische Systeme "auf einem Gitter". Einfache Regeln beschreiben iterative Entwicklung von Zellen. Verhalten kann trotzdem komplex werden (vgl. nicht-lineare Dynamik).
Anwendungen: (Selbst-)Organisation, Evolution (Bio), Strukturbildung (Chemie), ...
Beispiele:

- Pascalsches Dreieck
- Wolfram elementary celluar automata (`https://de.wikipedia.org/wiki/Zellul%C3%A4rer_Automat`)
- Langton-Schleife (`https://de.wikipedia.org/wiki/Langton-Schleife`)
- Game of Life (Convay) (`https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens`)
- Nagel-Schreckenberg-Modell (`https://de.wikipedia.org/wiki/Nagel-Schreckenberg-Modell`)
- HPP/FHP-Modell (Lattice Gas)

Zelluläre Automaten