Moderne Methoden in der Auswertung von Experimenten

Johannes Elmsheuser, Günter Duckeck

Ludwig-Maximilians-Universität München

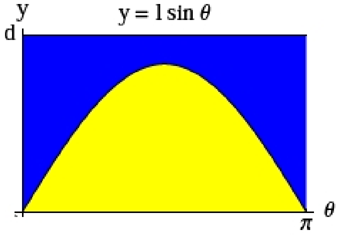
29 Apr 2008

Inhalt

Outline

Ein klassisches Beispiel

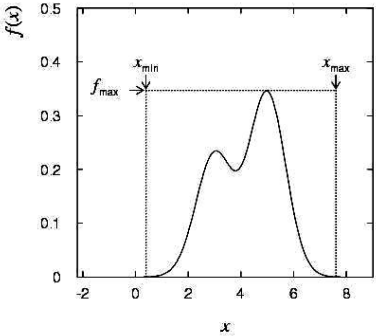
- Historisches Beispiel zur Berechnung der Zahl π: Buffon's Nadel (Graf G.L.L. von Buffon, 1707-1788)
- Nadel der Länge l wird auf Fläche mit gleich weit voneinander entfernter paralleler Gerade geworfen (Abstand d).
- Häufigkeit für ,,Geradentreffer" $p = k/n = 2/\pi$ (für $d = l$)



Häufigkeit für Geradentreffer:
Anzahl mögliche Fälle: $\pi \cdot d$
Anzahl günstige Fälle: $\int_0^\pi l \sin\theta d\theta = 2l$

Monte Carlo Integration

Integration einer
Wahrscheinlichkeitsdichtefunktion
pdf



Wie im vorhergehenden Beispiel: simuliere die Anzahl der ,,Treffer" und ,,Nicht-Treffer"

Monte Carlo Methode

Monte Carlo Methode:

Eine numerische Methode zur Berechnung von Wahrscheinlichkeiten und abgeleiteten Größen unter Benutzung von Zufallszahlen

- Erzeuge eine Sequenz von gleichförmig verteilten Zufallszahlen $r_1, .., r_m$ im Intervall $[0, 1]$
- Benutze diese Sequenz, um eine andere Sequenz $x_1, ..., x_n$ zu erzeugen, die einer für uns interessanten Wahrscheinlichkeitsdichtefunktion (pdf) $f(x)$ folgt
- Benutze die Werte x, um Eigenschaften von $f(x)$ zu bestimmen, z.B. Anzahl von x in $a < x < b$: $\int_a^b f(x)dx$
- $\Rightarrow$ Monte Carlo Berechung $\widehat{=}$ Integration
- generierte Monte Carlo Werte $\widehat{=}$ „Simulierte Daten"

Outline

1. Monte Carlo Methode
2. Zufallszahlen
3. Beliebig verteilte Zufallszahlen
4. Zusammenfassung

Erzeugung von Zufallszahlen (I)

Erzeuge gleichverteilte Zahlen im Intervall $[0, 1]$

- Würfle eine Folge von Zahlen → ,,Zufallszahlengenerator"
- Computeralgorithmen zur Erzeugung der Folge $I_1, ..., I_n$
- Computeralgorithmen: nur deterministische Zahlenfolgen, sog. ,,Pseudo-Zufallszahlen"

Anforderungen:

- ,,zufällig" verteilt mit langer Periode
- reproduzierbar ist manchmal erwünscht, d.h. gleiche Zahlenfolge von Zufallszahlen bei gleichen Startbedingungen
- Schneller Algorithmus

Erzeugung von Zufallszahlen (II)

Beispiele:

- linear kongruenter Generator:
  - $I_j = (a \cdot I_{j-1} + c) \mod m$
  - 3 ganzzahlige Konstanten: Multiplikator a, Summand c, Modul m
  - I_0: Saatzahl (seed)
  - Zahlenfolge $I_1, I_2, ...$ zwischen 0 und $m - 1$
  - periode Folge mit maximaler Periode m
  - gleichförmiges $u_j = I_j/m$ in $[0, 1]$
- Multiplikativer linear kongruenter Generator:
  - Setze: $c = 0$

Erzeugung von Zufallszahlen (III)

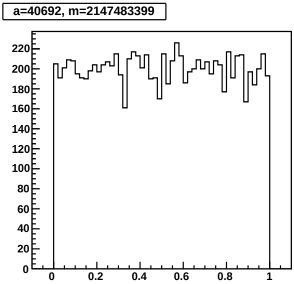
- Beispiel einer periodischen Folge: $a = 3$,$m = 7$, $I_0 = 1$

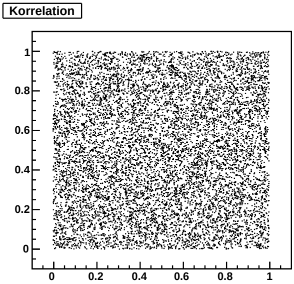
$$
\begin{aligned}
I_0 &= 1 \\
I_1 &= (3 \cdot 1) \mod 7 = 3 \\
I_2 &= (3 \cdot 3) \mod 7 = 2 \\
I_3 &= (3 \cdot 2) \mod 7 = 6 \\
I_4 &= (3 \cdot 6) \mod 7 = 4 \\
I_5 &= (3 \cdot 4) \mod 7 = 5 \\
I_6 &= (3 \cdot 5) \mod 7 = 1
\end{aligned}
$$

- $\Rightarrow$ Wähle a, m entsprechend, um lange Periode zu erhalten
- m nahe der größten Integer Zahler des Computers
- Benutze nur Untersequenz
- $\Rightarrow$ DEMO

Erzeugung von Zufallszahlen (IV)

- u_i sind in $[0, 1]$ aber auch wirklich zufällig ?
- Wähle a, m, so daß I_i Zufallszahlentests bestehen (s.Blobel)
  - Gleichförmige Verteilung
  - χ^2 Test für Unterintervalle von $[0, 1]$
  - Korrelationstest für n-dim Gitter ($\Rightarrow$ DEMO)
  - Weitere Tests: Gap-Test, Random-Walk





Erzeugung von Zufallszahlen (IV)

Zufallszahlengeneratoren in ROOT: (Auszug aus Doku)

- *TRandom3*, is based on the ,,Mersenne Twister generator", and is the recommended one, since it has good random proprieties (period of about 10^{6000}) and it is fast.
- *TRandom1*, based on the RANLUX algorithm, has mathematically proven random proprieties and a period of about 10^{171}. It is however slower than the others.
- *TRandom2*, is based on the Tausworthe generator of L'Ecuyer, and it has the advantage of being fast and using only 3 words (of 32 bits) for the state. The period is 10^{26}.

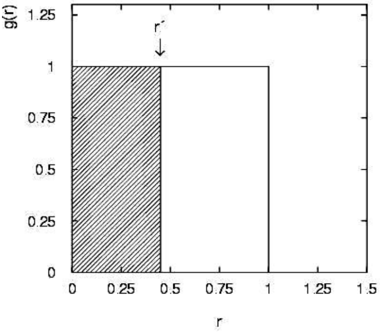
Outline

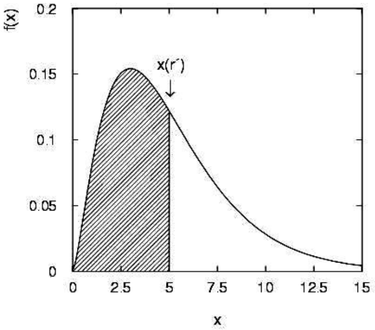
Erzeugung von beliebige verteilter Zufallszahlen

- Transformationsmethode
  u_i in $[0, 1]$ gleichförmig verteilt
  Suche $x(u)$, so daß $x_I = x(u_i)$ Verteilungsfunktion $f(x)$ folgen
- v. Neumann'sches Rückweisungsverfahren
  („acceptance-rejection method")
- Kombination der beiden Methoden

Die Transformationsmethode (I)

- Gegeben $u_1, ..., u_n$ gleichförmig verteilt in $[0, 1]$
- Finde $x_1, ..., x_n$ die $f(x)$ folgen, durch geeignete Transformation $x(u)$



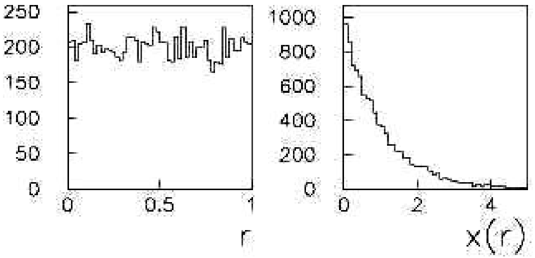


- Sei: $P(u \leq u') = P(x \leq x(u'))$
  d.h. $\int_{-\infty}^{u'} g(u)du = u' = \int_{-\infty}^{x(u')} f(u')du' = F(x(u'))$
- d.h. Setze: $F(x) = u$ and löse diese Gl. für $x(u)$

Die Transformationsmethode (II)

Beispiel:

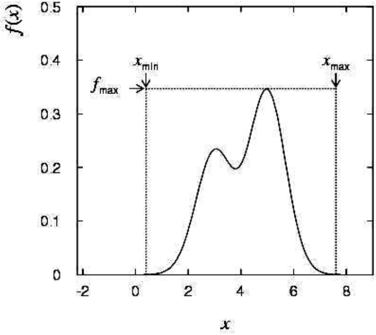
- Exponetielle pdf: $f(x, \lambda) = \lambda \cdot \exp(-\lambda \cdot x)$ für $x \geq 0$
  $\Rightarrow u = \int_0^x \lambda \cdot \exp(-\lambda \cdot t)dt = 1 - exp(-\lambda \cdot x)$
  $\Rightarrow x_j = -\ln(1 - u_j)/\lambda$ bzw. $x_J = -\ln(u_j)/\lambda$



- Integral $F(x)$ der Verteilung $f(x)$ muß bekannt und invertierbar sein. $F^{-1}(u)$ folgt dann Verteilung $f(x)$

Die Akzeptanzmethode

- Integration einer Wahrscheinlichkeitsdichtefkt. *pdf*
- In 2D: umschließe Funktion mit einer definierten Fläche



- Erzeuge eine Zufallszahl x, gleichverteilt in $[x_{min}, x_{max}]$, d.h. $x = x_{min} + r_1(x_{max} - x_{min})$ wobei r_1 gleichverteilt in $[0, 1]$.
- Generiere eine 2. unabhängige Zufallszahl gleichverteilt zwischen 0 und f_{max}, d.h. $u = r_2 f_{max}$
- Wenn $u < f(x)$, akzeptiere x; falls nicht, verwerfe x und wiederhole die Prozedur

Die Akzeptanzmethode - Beispiel (I)

Berechnung der Zahl π

$$f(x) = \sqrt{1 - x^2}$$

im Intervall $[0, 1]$

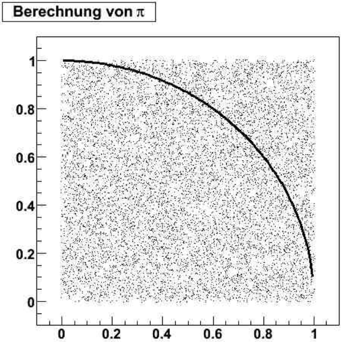
Akzeptiere Punktepaar, wenn

$$x^2 + y^2 <= 1$$

Bilde Verhältnis von

,,Innerhalb/Gesamt"

$$\pi = 4\cdot \text{ Innerhalb/Gesamt}$$



$\Rightarrow$ DEMO

Erzeugung Gaussförmig verteilter Zufallszahlen

Standardisierte Normalverteilung: $\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$

- Einfacher Algorithmus basierend auf Zentralem Grenzwertsatz: $x_i = \sum_{j=1}^{12} u_j - 6$
- Box-Muller-Verfahren:
  - Erzeuge gleichförmig verteilte u_1, u_2 in $[0, 1]$
  - Berechne: $v_1 = 2u_1 - 1$, $v_2 = 2u_2 - 1$ (v_1, v_2 gleichförmig verteilt in $[-1, +1]$)
  - Berechne: $r^2 = v_1^2 + v_2^2$.
  - Falls $r^2 > 1$, beginne von vorne. sonst: $x_1 = v_1\sqrt{\frac{-2\ln r^2}{r^2}}$ und $x_2 = v_2\sqrt{\frac{-2\ln r^2}{r^2}}$
  - x_1 und x_2 sind unabhängig normal verteilt.

Genauigkeit von Monte Carlo Berechnungen

Vergleiche Genauigkeit von Monte Carlo mit anderen Methoden

- Monte Carlo Berechung $\widehat{=}$ Integration
- Vergleich mit numerischer Integration:
  - Trapezregel
  - Simpsonsche Regel

Für 1-dim Integral:

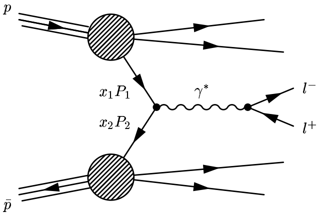
- MC:
  - $n \sim$ Anzahl der generierten Zufallszahlen
  - Genauigigkeit: $\sim 1/\sqrt{n}$
- Trapez-Regel (Simpson):
  - $n \sim$ Anzahl der Intervalle
  - Genauigigkeit: $\sim 1/n^2$ $(\sim 1/n^4)$
- In 1-dim sind numerische Methoden genauer mit weniger Rechenaufwand

N-dim:

- MC: Genauigkeit $\sim 1/\sqrt{n}$, unabhängig von Dimension
- Trapez: Genauigkeit $\sim 1/(\sqrt[d]{n^2})$
- Für $d > 4$ MC-Methode besser

Monte Carlo Ereignis Generatoren

Einfache Reaktion:
$p\bar{p} \rightarrow Z/\gamma^* \rightarrow e^+e^-$



- Ereignis-Generatoren werden benutzt um Teilchenreaktionen zu simulieren
- z.B. PYTHIA, HERWIG, ALPGEN, ...
- Ausgabe: ,,Ereignisse", d.h. für jedes Ereignis, wird eine Liste von Teilchen generiert zusammen mit den 4er-Vektoren, etc.

Monte Carlo Detektor Simulation

- Detektor Simulation erhält Teilchenliste aus Ereignis-Generator als Eingabe
- Detektor Simulation:
  - Simuliert Durchgang der verschiedenen Teilchen durch Detektorkomponenten
  - Coulombstreuung (simuliert Streuwinkel)
  - Teilchenzerfälle (simuliert Lebensdauer)
  - Ionisierungsenergie (simuliert ΔE)
  - Elektromagnetische/Hadronische Schauer
  - Elektronische Signale in Detektoren
- Simulierte Ausgabe hat gleiches Format wie echte Daten
- Einfacher Vergleich zwischen Daten/MC (vorausgesetzt die Effizienzen sind gleich)
- Programmpaket: GEANT

Outline

Zusammenfassung

- Monte Carlo Methode
- Erzeugung von Zufallszahlen
- Beliebig verteilte Zufallszahlen
- Beispiele für Monte Carlo Methoden