03-05-H
-709.53

Informatik für Nichtinformatiker (5)

Prof. Dr. Udo Frese
Tobias Hammer

Anwendung: Monte-Carlo-Simulation

Was bisher geschah

- **Unified Modeling Language (UML) Klassendiagramme**
  - eine Stufe detaillierter, als CRC Karten
  - Klassen: Kasten mit Name, Attribute, Methoden
  - Pfeil für Assoziationen mit Kardinalitäten
  - Pfeil für „ist ein“ (Unterklasse)
  - Pfeil für „hat ein“ (Teile eines Ganzen: Aggregation und Komposition)
- **Container Array: `[e1, e2,…]`**
  - Zugriff: `[idx]`, `[start,count]`, `[start..end]`, `[start…excl]`
- **Container Hash: `{k1=>o1, k2=>o2}`**
  - Zugriff `[key]`, `nil`, falls noch nicht vorhanden
- **durchlaufen mit `.each`**
- **elementweise abbilden mit `.collect`**
- **durchsuchen mit `.find`**

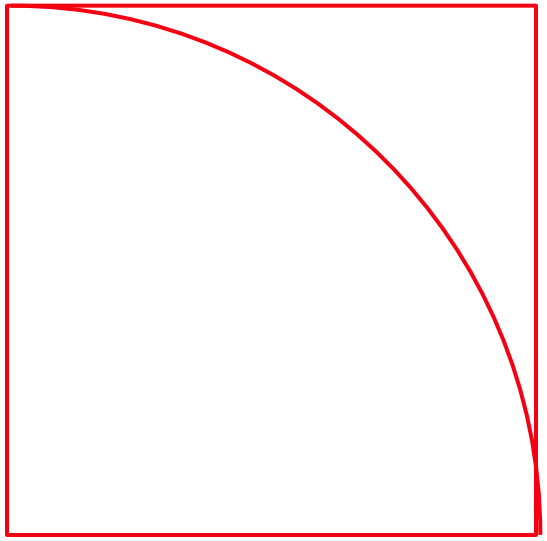
Anwendung: Monte-Carlo-Simulation

- **Berechnung statistischer Vorgänge durch die wiederholte Simulation einzelner Ereignisse durch Zufallszahlen.**
  - modelliere betrachteten Vorgang als Programm,…
  - …, bei dem der wirkliche Zufall durch Zufallszahlen simuliert wird.
  - simuliere Vorgang mit sehr vielen Versuchen (z.B. Millionen mal)
  - berechne aus den einzelnen Simulationsergebnissen, gesuchte Größe, z.B. Wahrscheinlichkeiten, Erwartungswert, Varianz, etc.
  - ACHTUNG: Ergebnis ist Näherungslösung!
- **Anwendungen: Statistik, Physik, (physikalische) Chemie, Reaktionssimulation, Proteinsimulation, Finanzwirtschaft, Zuverlässigkeitsstudien, Kernphysik, Partikelphysik, Robotik, Spieltheorie (Quelle: wikipedia/Monte_Carlo_method)**
- **„Offered the choice between mastery of a five-foot shelf of analytical statistics book and middling ability at performing statistical Monte Carlo simulations, we would surely choose the latter skill.“ (Press et al., Numerical Recipes, Cambridge University Press, 2002)**

Anwendung: Monte-Carlo-Simulation

Beispiel: Berechnung von π

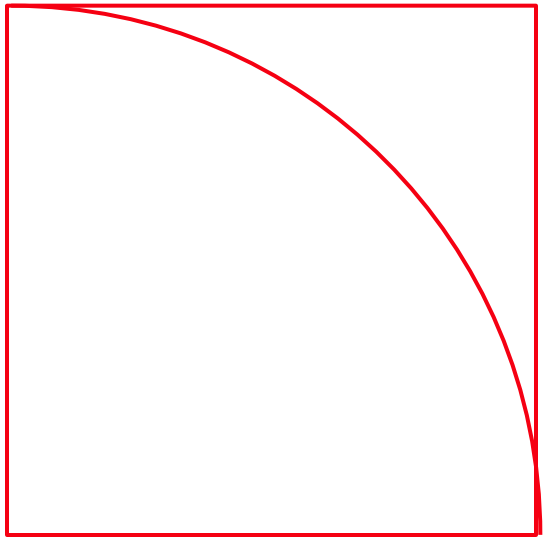
- **Frage an das Auditorium: Wie hoch ist die Wahrscheinlichkeit, dass ein zufälliger Punkt des Quadrates im Kreis liegt?**



Anwendung: Monte-Carlo-Simulation

Beispiel: Berechnung von π

- **Frage an das Auditorium: Wie hoch ist die Wahrscheinlichkeit, dass ein zufälliger Punkt des Quadrates im Kreis liegt?**
- **Verhältnis der Flächen: $\pi/4$**



Anwendung: Monte-Carlo-Simulation

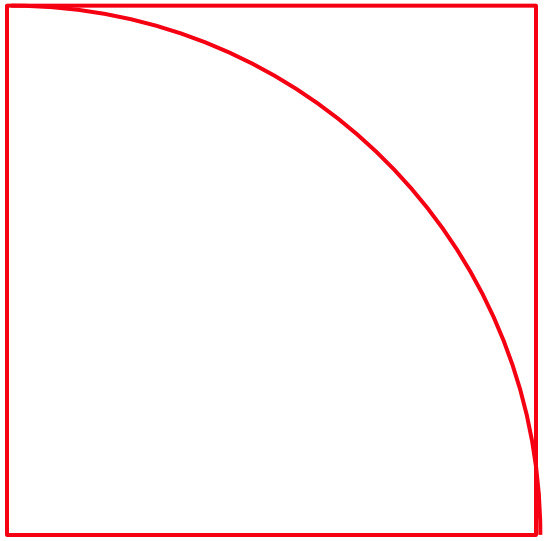
Zufallszahlen in Ruby

- **rand(n) ganze Zahl zwischen 0..n-1**
- **rand reelle Zahl zwischen 0…1**
- **Kein echter Zufall, sondern eine Formel, die wild durcheinander Zahlen produziert.**
  - akzeptiert für Statistik, Simulation, etc.
  - nicht anwenden für Kryptographie

Anwendung: Monte-Carlo-Simulation

Beispiel: Berechnung von π

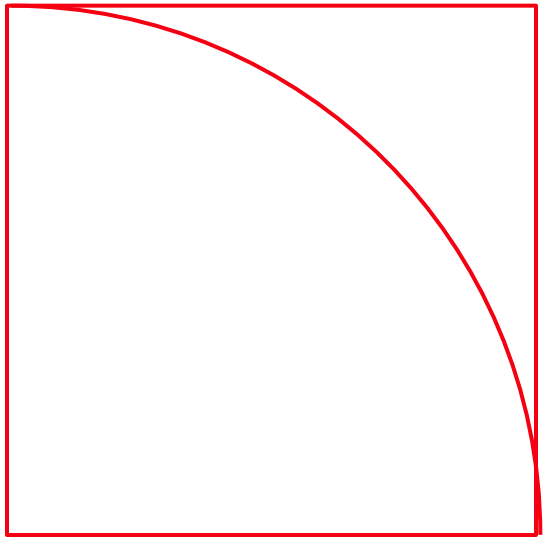
- **Frage an das Auditorium: Wie hoch ist die Wahrscheinlichkeit, dass ein zufälliger Punkt des Quadrates im Kreis liegt?**
- **Verhältnis der Flächen: $\pi/4$**
- **Frage an das Auditorium: Wie simuliert man das Experiment „liegt ein zufälliger Punkt im Kreis?“**



Anwendung: Monte-Carlo-Simulation

Beispiel: Berechnung von π

- **Frage an das Auditorium: Wie hoch ist die Wahrscheinlichkeit, dass ein zufälliger Punkt des Quadrates im Kreis liegt?**
- **Verhältnis der Flächen: π/4**
- **Frage an das Auditorium: Wie simuliert man das Experiment „liegt ein zufälliger Punkt im Kreis?“**
- **ziehe Zufallszahl x zwischen 0..1**
- **ziehe Zufallszahl y zwischen 0..1**
- **berechne d=Math.sqrt(x*x+y*y)**
- **Ergebnis: d<=1**
- **Dann zählen, wie oft dies passiert.**
- **Ruby: (inifrese0905.montecarlo.rb)**



Anwendung: Monte-Carlo Simulation

Anwendung: statistische Signifikanztests

- **Signifikanztest einer Hypothese**
  - definiere zu widerlegende 0-Hypothese, wie Experimentaldaten X zufällig zustande kommen
  - definiere aus den Experimentaldaten X berechneten Wert I(X), wobei großes I(X) gegen die 0-Hypothese spricht
  - berechne Wahrscheinlichkeit $p(I(X)<I(X_e))$ aus echten Experimentaldaten X_e, wie wahrscheinlich es ist, dass zufällige Daten nach 0-Hypothese einen niedrigeren I(X) Wert liefern
  - mit dieser Signifikanz ist 0-Hypothese verworfen
- **Monte-Carlo Simulation um $p(I(X)<I(X_e))$ zu bestimmen**
  - simuliere Experiment unter 0-Hypothese
  - berechnet I(X)
  - zähle, wie oft $I(X)<I(X_e)$

Anwendung: Monte-Carlo Simulation

Anwendung: statistische Signifikanztests

- **Beispiel: Würfel würfelt bei 10 Würfen 5 mal 6er**
- **Vermutung: Würfel ist gezinkt und würfelt zu viele 6er**
- **0-Hypothese: Würfel intakt, Wahrscheinlichkeit 1/6**
- **I(X) sei Anzahl 6er unter n=10 Würfen**
- **$I(X_e)=5$, $p(I(X)<I(X_e)) \cong 0.984$**
- **0-Hypothese mit Signifikanz 98% verworfen, Würfel gezinkt!**
- **Berechnung $p(I(X)<I(X_e))$ in Ruby: (inifrese0905.montecarlo.rb)**

Anwendung: Monte-Carlo-Simulation

Beispiel: Wie wahrscheinlich ist ein 6er Kniffel?

- **Analytisch auszurechnen, aber nicht ganz einfach**
- **Frage an das Auditorium: Wie simuliert man einen Versuch (umgangssprachlich)?**

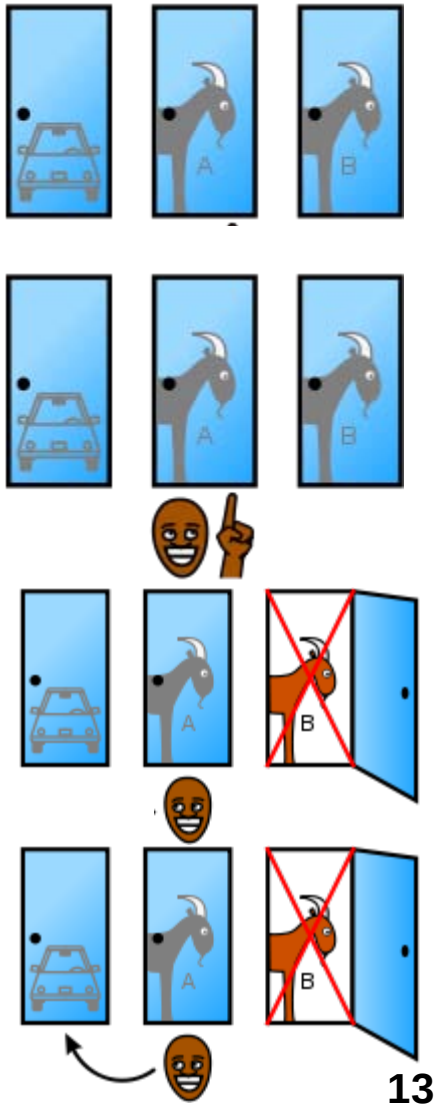
Anwendung: Monte-Carlo-Simulation

Beispiel: Wie wahrscheinlich ist ein 6er Kniffel?

- Analytisch auszurechnen, aber nicht ganz einfach
- Frage an das Auditorium: Wie simuliert man einen Versuch (umgangssprachlich)?
- Gesamtzahl 6er sei 0
- wiederhole 3 mal:
- wiederhole 5-Gesamtzahl mal
- würfele, wenn Würfel =6, erhöhe Gesamtzahl 6er um 1
- Beispiel in Ruby: (inifrese0905.montecarlo.rb)

Anwendung: Monte-Carlo-Simulation

- **Beispiel: Das Ziegenproblem (Monty Hall problem)**
- **Quelle: wikipedia/Ziegenproblem**
- **Spielshow mit 3 Türen**
  - hinter einer Tür: Auto, hinter zwei Türen: Ziege
  - Kandidat wählt Tür
  - Moderator öffnet eine andere Tür mit Ziege (weiß, wo das Auto steht)
  - Kandidat kann sich umentscheiden
- **Ist es besser zu wechseln, oder nicht?**
- **Beispiel in Ruby: (inifrese0905.montecarlo.rb)**



Anwendung: Monte Carlo Simulation

Zusammenfassung

- **Berechnung statistischer Vorgänge durch die wiederholte Simulation einzelner Ereignisse durch Zufallszahlen.**
  - modelliere betrachteten Vorgang als Programm,…
  - …, bei dem der wirkliche Zufall durch Zufallszahlen simuliert wird.
  - simuliere Vorgang mit sehr vielen Versuchen (z.B. Millionen mal)
  - berechne aus den einzelnen Simulationsergebnissen, gesuchte Größe, z.B. Wahrscheinlichkeiten, Erwartungswert, Varianz, etc.
  - ACHTUNG: Ergebnis ist Näherungslösung!
- **Sehr mächtiges Werkzeug**
- **Kann Probleme lösen, die analytisch sehr schwer oder unlösbar sind**
- **Vorsicht: Manchmal in der jeweiligen Fachkultur unüblich**