

# Modellierung und Programmierung

Dr. Martin Riplinger

7.11.2012



## Monte-Carlo-Methoden

- ▶ Oberbegriff für mathematische Verfahren, deren Funktionsprinzip der Zufall ist
- ▶ Name geht zurück auf John v. Neumann und reflektiert die Tatsache, dass immer wieder „gewürfelt“ wird.
- ▶ Grundlegendes Prinzip ist einfach zu verstehen → bei Anwendern beliebt
- ▶ Trotzdem vielseitig und effizient verwendbar
- ▶ Kommen vor allem dann zum Einsatz, wenn ein zufälliger Prozess simuliert werden soll (z. B. Finanzmathematik, Dynamik von Gasen / Partikeln / Elektronen in Materie, Ausbreitung von Krankheiten, ...)

### Mathematische Basis:

#### Gesetz der großen Zahlen (einfache Fassung)

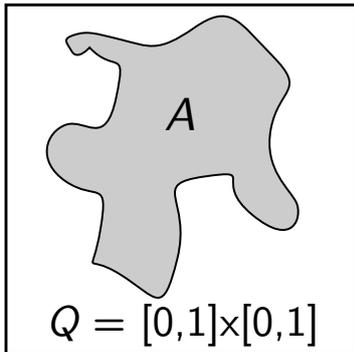
Je öfter man ein Zufallsexperiment durchführt, desto mehr nähert sich die **relative Häufigkeit** eines Ereignisses der **Wahrscheinlichkeit** desselben Ereignisses an.

#### Beispiel: Idealer Würfel

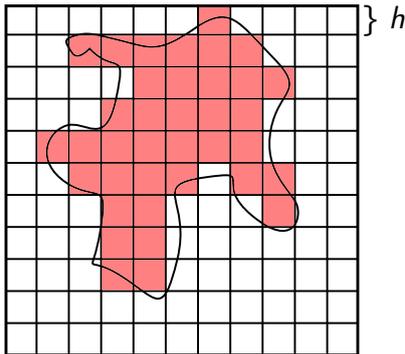
$P(X = 1) = \frac{1}{6}$  – Wahrscheinlichkeit, dass eine Eins gewürfelt wird  
 $N_1$  – Anzahl der gewürfelten Einsen nach  $N$  Würfeln

Gesetz der großen Zahlen: 
$$\lim_{N \rightarrow \infty} \frac{N_1}{N} = \frac{1}{6}$$

# Monte-Carlo-Integration



$|A|$  = Flächeninhalt von  $A = ?$



## Klassische Vorgehensweise (Quadratur):

- ▶ Unterteile  $Q$  in  $N \cdot N$  Quadrate der Kantenlänge  $h = 1/N$
- ▶ Bestimme die Anzahl  $N_A$  der Quadrate, deren Mittelpunkte in  $A$  liegen
- ▶ Berechne  $|A| \approx N_A \cdot h^2$
- ▶ Nachteil: sehr teuer in höheren Dimensionen ( $\geq 3$ )

# Monte-Carlo-Integration

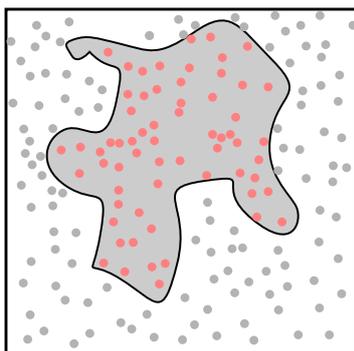
## Stochastischer Zugang:

- ▶ Wahrscheinlichkeit, dass ein zufällig erzeugter Punkt  $z \in Q$  in  $A$  landet, ist gegeben durch

$$p_A = P(z \in A) = \frac{|A|}{|Q|} = |A|$$

- ▶ Gesetz der großen Zahlen: Für eine hinreichend große Anzahl  $N$  von zufällig erzeugten Punkten ist

$$\frac{\#(\text{Punkte in } A)}{\#(\text{Punkte insgesamt})} = \frac{N_A}{N} \approx p_A = |A|$$

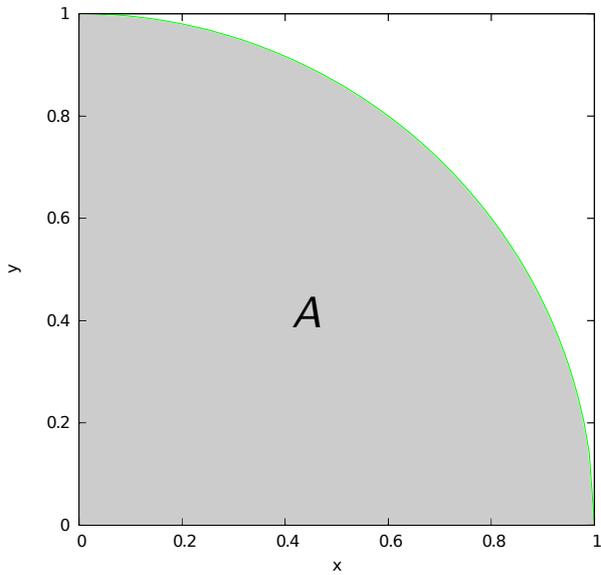


## Stochastische Vorgehensweise:

- ▶ Erzeuge  $N$  zufällige Punkte in  $Q$
- ▶ Zähle die Anzahl  $N_A$  der Punkte, die in  $A$  liegen
- ▶ Berechne  $|A| = \frac{N_A}{N} [ \cdot |Q| ]$
- ▶ Vorteile:
  - ▶ simpel, einfach zu implementieren
  - ▶ man muss nur zwischen  $z \in A$  und  $z \notin A$  unterscheiden können
  - ▶ in höheren Dimensionen sehr effizient

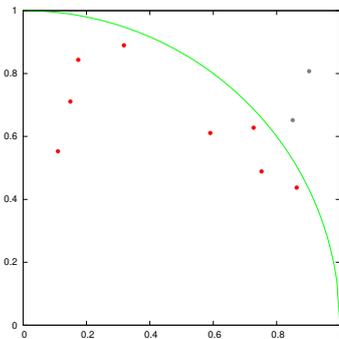
# Monte-Carlo-Integration

## Beispiel: Einheitskreisscheibe im 1. Quadranten

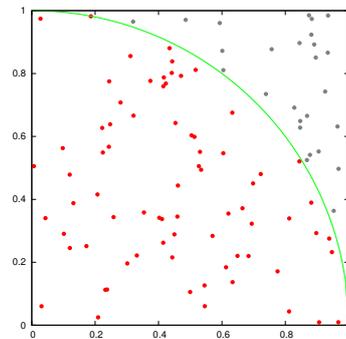


- ▶ Umgebendes Quadrat  
 $Q = [0, 1]^2 \Rightarrow |Q| = 1$
- ▶ Einheitskreis-Viertel  
 $A = \{(x, y) \in Q \mid x^2 + y^2 < 1\}$
- ▶ Pseudocode:
  1. Lies eine Zahl  $N$  ein
  2. Für  $i = 1, \dots, N$ :
    - ▶ Erzeuge zufällige Zahlen  $x$  und  $y$
    - ▶ Falls  $x^2 + y^2 < 1$ :  
 $N_A \leftarrow N_A + 1$
  3. Gib Flächeninhalt  $= N_A/N$  aus

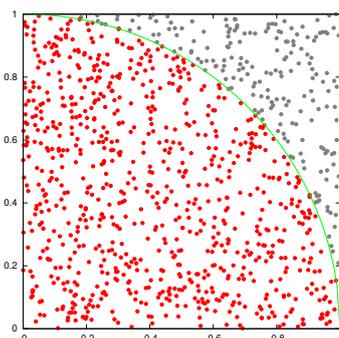
## Monte-Carlo-Integration: Ergebnisse



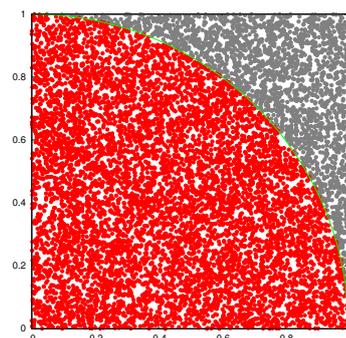
$N = 10$ , Fehler:  $2.85 \cdot 10^{-1}$  (36.3 %)



$N = 100$ , Fehler:  $4.54 \cdot 10^{-2}$  (5.8 %)



$N = 1000$ , Fehler:  $2.26 \cdot 10^{-2}$  (2.9 %)



$N = 10000$ , Fehler:  $4.80 \cdot 10^{-3}$  (0.6 %)

## Monte-Carlo-Simulation: Radioaktiver Zerfall

- ▶ Radioaktive Nuklide (Atomsorten) sind instabil und werden unter Aussendung von Strahlung in andere, stabile Atomsorten umgewandelt.
- ▶ Der Zerfall eines einzelnen Atoms geschieht spontan und kann als zufälliges Ereignis angesehen werden.
- ▶ Das Isotop  $^{131}\text{I}$  (Jod-131) ist ein Betastrahler. Es wird zum stabilen Isotop  $^{131}\text{Xe}$  (Xenon-131) umgewandelt. Dabei wird ein Elektron (Beta-Teilchen) emittiert.
- ▶ Jod-131 besitzt eine Halbwertszeit von  $T_{1/2} = 8.02070$  Tagen, d. h. nach  $T_{1/2}$  ist (etwa) die Hälfte einer betrachteten Menge Jod-131 zerfallen.
- ▶ Experimente zeigen den Zusammenhang

$$\Delta N = -\lambda \cdot N \cdot \Delta t$$

mit

$\Delta t$  : kleines Zeitintervall

$N$  : Anzahl der radioaktiven Kerne

$\Delta N$  : Änderung von  $N$  im Zeitintervall  $\Delta t$

$\lambda > 0$  : Zerfallskonstante (Proportionalitätsfaktor, materialabhängig)

## Monte-Carlo-Simulation: Radioaktiver Zerfall

**Klassische Herangehensweise (für große Anzahl  $N$ ):**

$$\Delta N(t) = -\lambda N(t) \Delta t$$

mit  $\Delta N(t) = N(t + \Delta t) - N(t)$ . Division durch  $\Delta t$  und der anschließende Grenzübergang  $\Delta t \rightarrow 0$  liefern die Differentialgleichung

$$\dot{N}(t) = -\lambda N(t).$$

Die (eindeutige) Lösung ist gegeben durch

$$N(t) = N(0) e^{-\lambda t}.$$

Die *Halbwertszeit*  $T_{1/2}$  ist der Zeitraum, nach dem die Hälfte der Kerne zerfallen ist, d. h.  $N(T_{1/2}) = N(0)/2$ . Durch eine einfache Umformung ergibt sich

$$T_{1/2} = \frac{\ln 2}{\lambda} \quad \text{bzw.} \quad \lambda = \frac{\ln 2}{T_{1/2}}.$$

Halbwertszeiten radioaktiver Isotope sind üblicherweise tabelliert.

**Stochastische Herangehensweise:**

$\lambda \Delta t =$  Wahrscheinlichkeit, dass ein Kern im Zeitraum  $\Delta t$  zerfällt

In jedem Zeitschritt wird für jeden Kern zufällig entschieden, ob er zerfällt.

# Monte-Carlo-Simulation: Radioaktiver Zerfall

## Beispiel: Jod-131

- ▶ Als Zeitschritt wählen wir  $\Delta t = 1$  min.
- ▶  $T_{1/2} = 8.0207$  d = 11549.808 min  $\Rightarrow$

$$\lambda = \frac{\ln 2}{T_{1/2}} = 6.001 \cdot 10^{-5} \text{ min}^{-1}$$

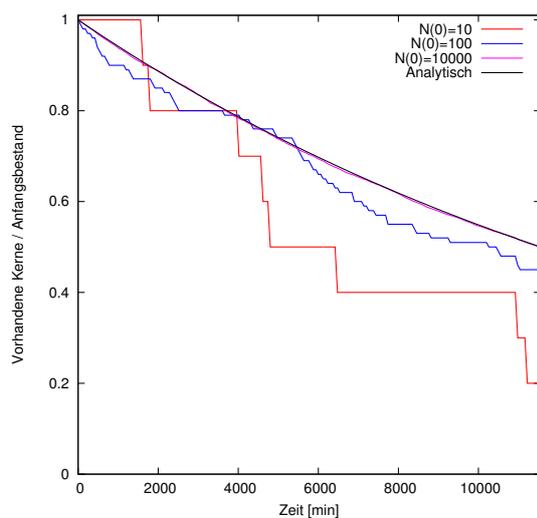
- ▶ Die Wahrscheinlichkeit, dass ein Kern *innerhalb einer Minute* zerfällt, ist gegeben durch

$$p = \lambda \Delta t = 6.001 \cdot 10^{-5}.$$

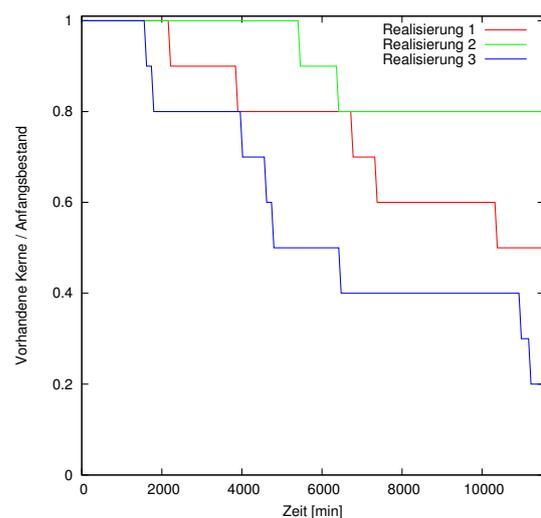
## Pseudocode:

1. Lies eine Zahl  $N$  (Anzahl zu Beginn) und eine Zahl  $T$  (Endzeitpunkt) ein
2. Für  $i = 1, \dots, T$ :
  - ▶ Für  $j = 1, \dots, N$ :
    - Erzeuge eine zufällige Zahl  $x \in [0, 1[$
    - Falls  $x < p$ , setze  $N \leftarrow N - 1$
3. Gib die Anzahl  $N$  der übriggebliebenen Kerne nach  $T$  Minuten aus

## Monte-Carlo-Simulation: Radioaktiver Zerfall – Ergebnisse



Zerfallskurven für verschiedene Anfangsbestände  $N(0)$



3 Realisierungen für  $N(0) = 10$

## Monte-Carlo-Simulation: Räuber-Beute-Modell

- ▶ Grundlegendes Modell in der Theorie der **dynamischen Systeme**
- ▶ Beschreibt die Zusammenhänge, welche die Entwicklung mehrerer interagierender Populationen (konkurrierende Spezies) bestimmen
- ▶ Anwendungsgebiete: Systembiologie, Epidemiologie, Ökonomie, ...

**Beispiel:**

Hecht

vs.

Karpfen



- ▶ Räuber
- ▶ sehr gefräßig
- ▶ mag Karpfen

- ▶ Beute
- ▶ harmlos
- ▶ lecker

## Monte-Carlo-Simulation: Räuber-Beute-Modell

**Grundannahmen:**

- ▶ In einem großen Weiher leben ausschließlich Karpfen ( $K$ ) und Hechte ( $H$ ).
- ▶ Der Nahrungsvorrat für Karpfen ist unbegrenzt. Ihr Wachstum in einem Zeitintervall  $\Delta t$  ist proportional zur Anzahl der vorhandenen Individuen:

$$\Delta K \sim K \Delta t$$

- ▶ Hechte, die keine Nahrung finden, fressen andere Hechte, d. h. der Rückgang ihres Bestandes in einem Zeitintervall  $\Delta t$  ist proportional zur Zahl der vorhandenen Exemplare:

$$\Delta H \sim -H \Delta t$$

↷ Ohne Interaktion gilt für beide Spezies das gleiche Modell wie beim radioaktiven Zerfall!

- ▶ Bei einer Begegnung wird der Karpfen vom Hecht gefressen. Hat ein Hecht eine bestimmte Anzahl Karpfen gefressen, „entsteht“ ein zusätzlicher Hecht. Je mehr Individuen es von beiden Sorten gibt, desto wahrscheinlicher ist solch eine Begegnung:

$$\Delta K \sim -KH \Delta t$$

$$\Delta H \sim KH \Delta t$$

## Monte-Carlo-Simulation: Räuber-Beute-Modell

**Gekoppeltes Gesamtmodell** ( $\Delta t = 1$ ):

$$\Delta K = \lambda_K K - p K H$$

$$\Delta H = n^{-1} p K H - \lambda_H H$$

$\Delta K/H$  : Änderung des Bestandes von  $K/H$

$\lambda_K$  : Zuwachsrates  $K$

$\lambda_H$  : Sterberates  $H$

$p$  : Wahrscheinlichkeit einer Begegnung

$n$  : Anzahl der  $K$ , die ein  $H$  zur Reproduktion fressen muss

**Parameterwahlen für die Simulation:**

$$\lambda_K = 2 \cdot 10^{-3},$$

$$\lambda_H = 5 \cdot 10^{-4},$$

$$p = 5 \cdot 10^{-6},$$

$$n = 5.$$

**Aus dem kontinuierlichen Modell bekannt:**

Ein stabiler Gleichgewichtszustand ist gegeben durch

$$K^* = \frac{n\lambda_H}{p} = 500, \quad H^* = \frac{\lambda_K}{p} = 400.$$

## Monte-Carlo-Simulation: Räuber-Beute-Modell

**Pseudocode:**

1. Lies die Anfangsbestände  $K$  und  $H$  sowie die Anzahl  $T$  der Zeitschritte ein

2. Für  $i = 1, \dots, T$ :

▶ Für  $k = 1, \dots, K$ :

- Erzeuge eine zufällige Zahl  $x \in [0, 1[$
- Falls  $x < \lambda_K$ : Setze  $K \leftarrow K + 1$

▶ Für  $k = 1, \dots, K$ :

▶ Für  $j = 1, \dots, H$ :

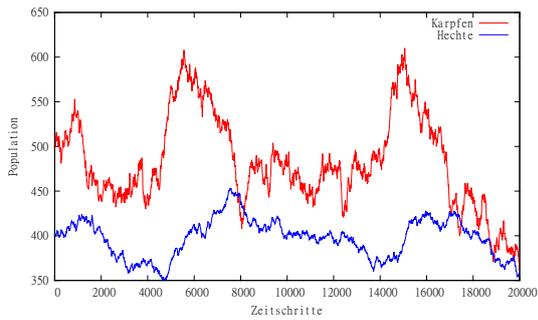
- Erzeuge eine zufällige Zahl  $x \in [0, 1[$
- Falls  $x < p$ : Setze  $K \leftarrow K - 1$   
Setze  $n_{\text{temp}} \leftarrow n_{\text{temp}} + 1$   
Falls  $n_{\text{temp}} = n$ : Setze  $H \leftarrow H + 1$  und  $n_{\text{temp}} \leftarrow 0$   
break;

▶ Für  $j = 1, \dots, H$ :

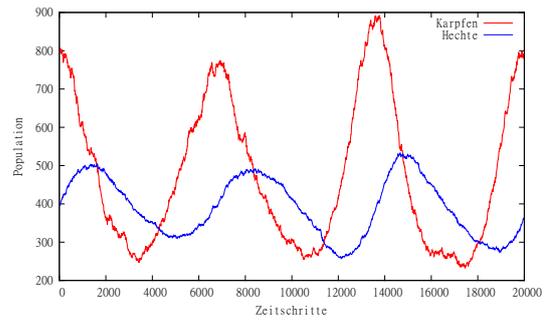
- Erzeuge eine zufällige Zahl  $x \in [0, 1[$
- Falls  $x < \lambda_H$ : Setze  $H \leftarrow H - 1$

▶ Gib die Zahlen  $K$  und  $H$  am Bildschirm aus

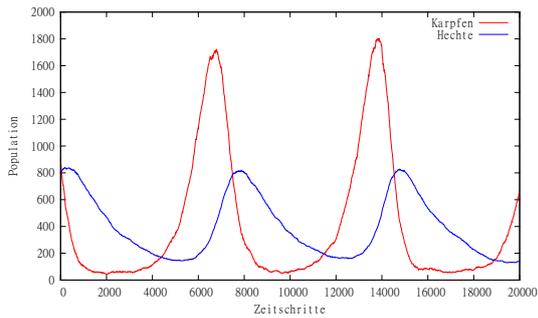
## Monte-Carlo-Simulation: Räuber-Beute-Modell – Ergebnisse



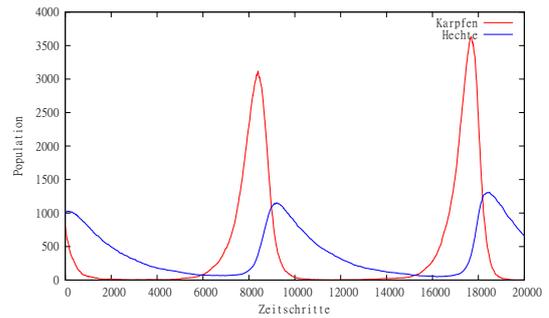
$K = 500, H = 400$



$K = 800, H = 400$

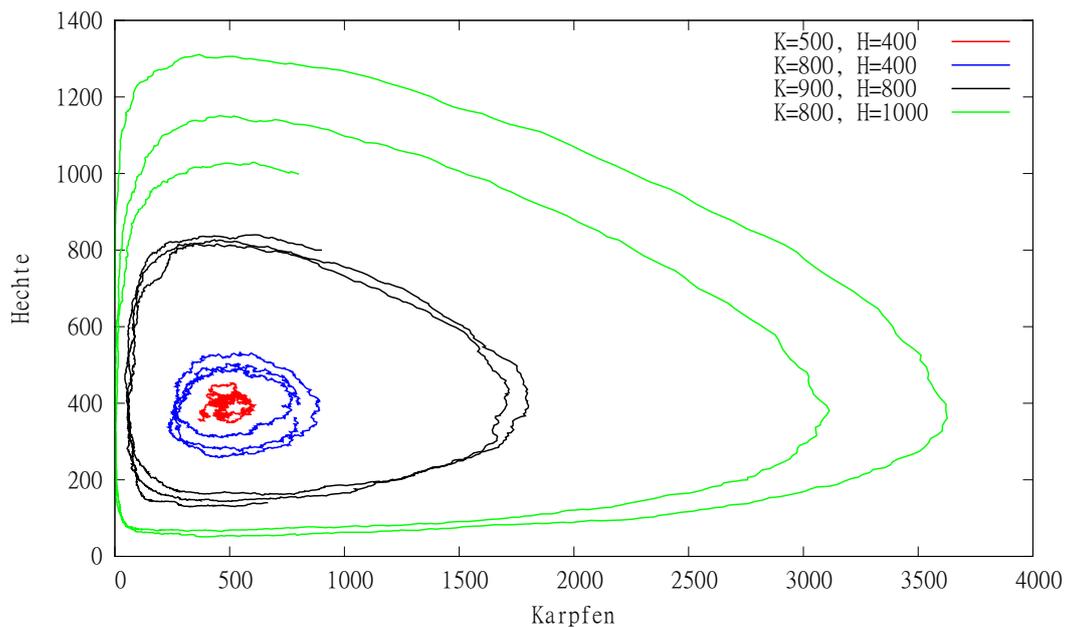


$K = 900, H = 800$



$K = 800, H = 1000$

## Monte-Carlo-Simulation: Räuber-Beute-Modell – Ergebnisse



Phasenraumdiagramm für verschiedene Startwerte