

Vorlesung

## **Statistische Methoden der Datenanalyse**

Prof. Dr. Dr. Wolfgang Rhode

---

## Monte-Carlo Methoden

---

## RISE-Programm

- Auslandspraktikum
- Nur für Bachelorstudenten
- <https://www.daad.de/rise/de/rise-weltweit/praktikum-finden/>

## Überblick

- Erzeugung beliebig verteilter Zufallszahlen
  - Importance Sampling
- Markov-Chain-Monte-Carlo
- Monte-Carlo-Simulationen

## Bisher gelernte Methoden zur Erzeugung von Zufallszahlen

### ■ *Inversion Sampling*

- + Kein Verwerfen von Zufallszahlen
- + Effizient
- Integration der PDF
- Invertierung der CDF
- Nicht auf jede Verteilung anwendbar

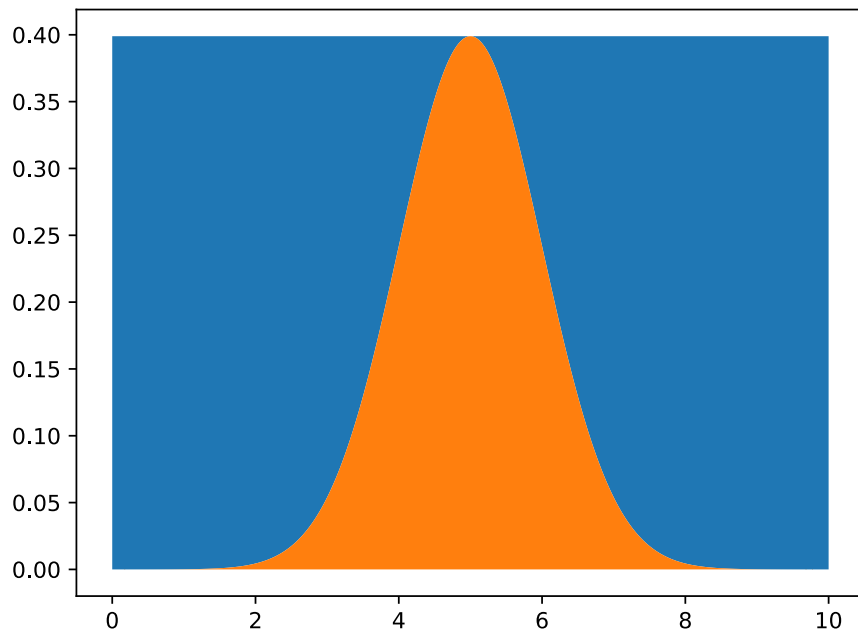
### ■ *Rejection Sampling*

- + Keine Integration oder Invertierung
- + Auf jede Funktion anwendbar
- Verwerfen von Zufallszahlen
- Ineffizienter mit steigenden Dimensionen
- Problem bei Verteilungen, die bis ins Unendliche reichen

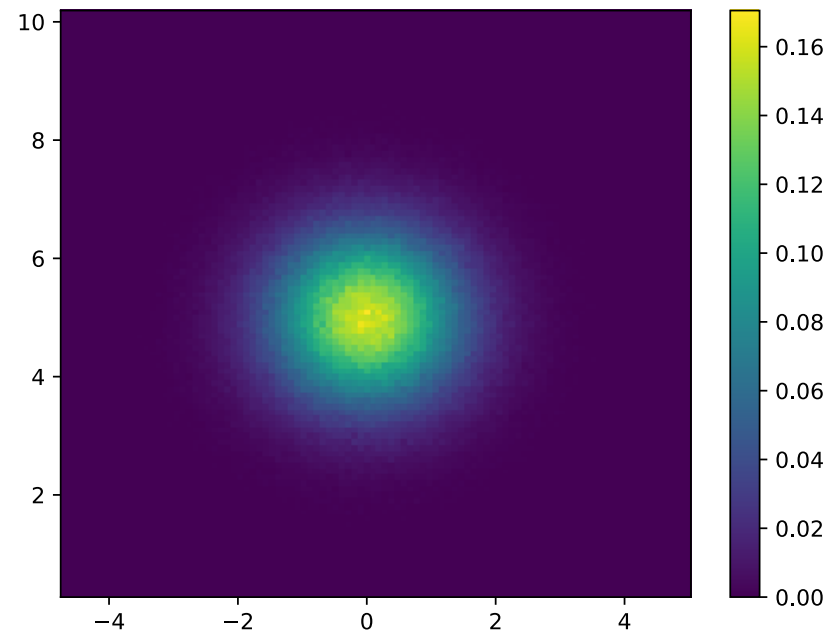
## „curse of dimensionality“

Anzahl verworfener Zufallszahlen nimmt mit mehr Dimensionen zu

1D Gauß PDF



2D Gauß PDF



## Importance Sampling


Gesuchte Verteilung:  $f(x)$

1. Finden einer geeigneten Funktion  $g(x)$  mit den Eigenschaften

- $g(x) \geq f(x) \forall x$
- $G(x) = \int_{-\infty}^x g(x') dx'$  ist invertierbar

2. Modifiziertes *rejection sampling*:

- Ziehung von zwei gleichverteilter Zufallszahlen  $\xi_1, \xi_2 \in [0, 1)$
- Transformation von  $\xi_1 \rightarrow \xi'_1$  entsprechend der Verteilung  $g(x)$  mittels *inversion sampling*
- Annahme von  $\xi'_1$ , falls  $g(\xi'_1) \cdot \xi_2 \leq f(\xi'_1)$

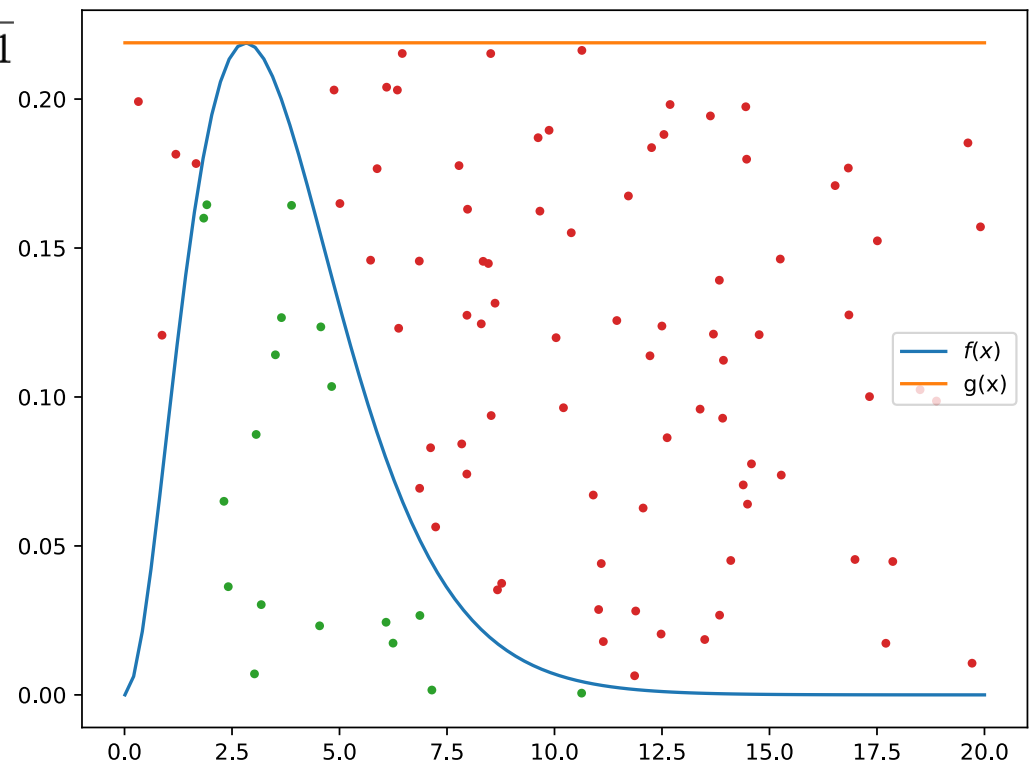
 Der Bruchteil  $\frac{g(x) - f(x)}{f(x)}$  der Zufallszahlen wird verworfen.

## Importance Sampling (Beispiel)

Planck-Verteilung:  $f(x) = N \frac{x^3}{e^x - 1}$

- Probleme:
  - Exponentiell abfallend
  - Verteilung geht bis Unendlich
  - Beispiel: 82 von 100 Zufallszahlen werden verworfen

➡ Siehe Übung

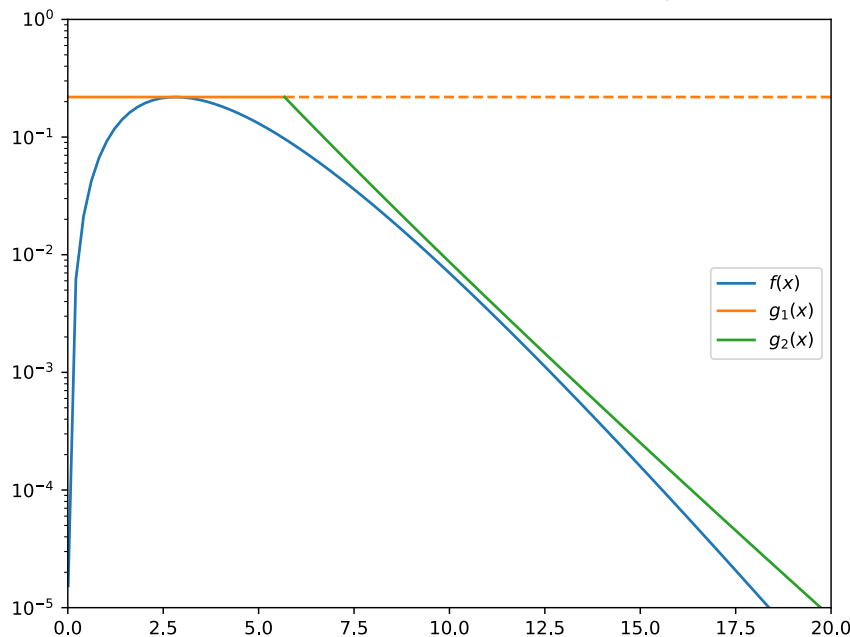


Generation von Zufallszahlen

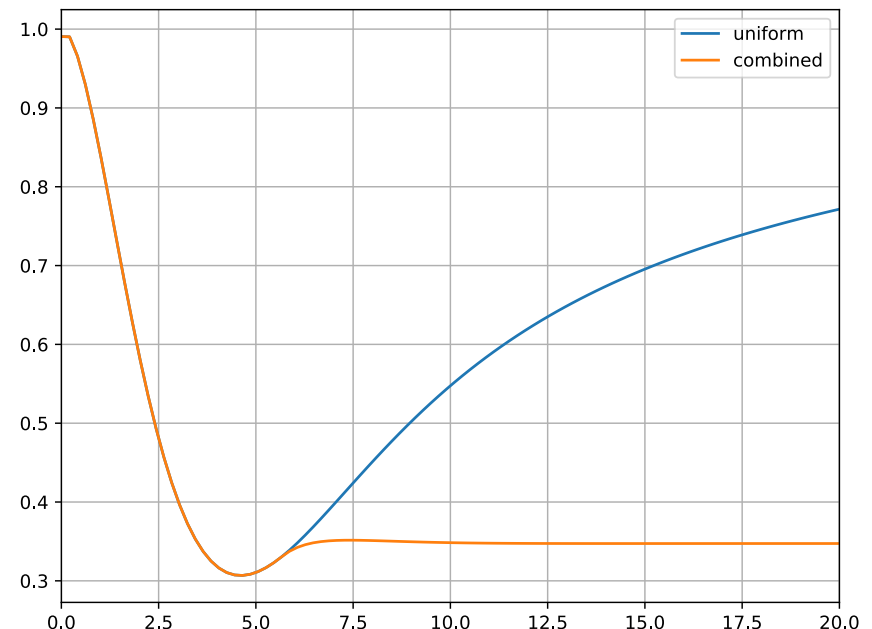
## Importance Sampling (Beispiel)

$$\text{Majorantenfunktion: } g(x) = \begin{cases} f(x_{\max}) & x \leq x_1 \\ 200N x^{-0.1} \exp(-x^{0.9}) & x > x_1 \end{cases}$$

einhüllende Verteilung



Effizienz





## Importance Sampling (Beispiel)

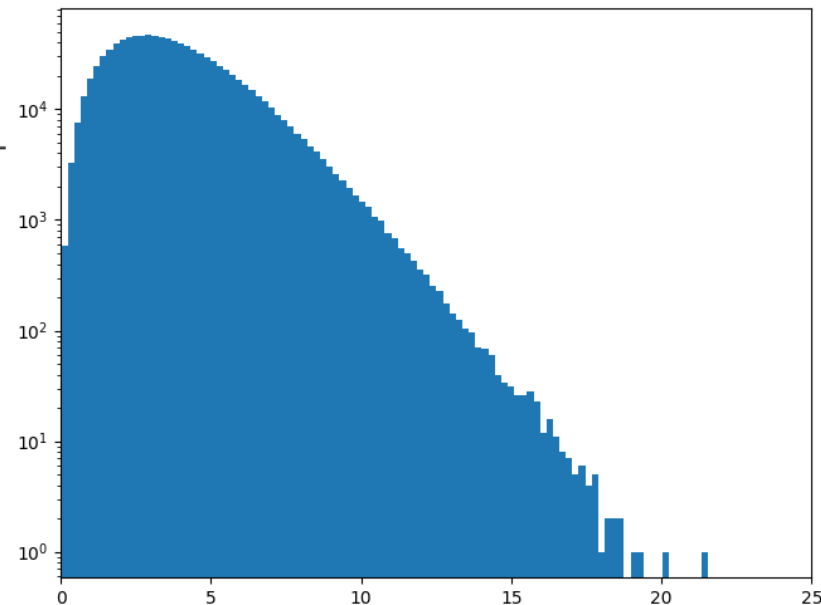
Vergleich: Erzeugen von 1 Mio. Zufallszahlen bei einem cutoff von 25

	zusätzliche Zufallszahlen	Laufzeit
Uniform	4.469.541	29.1 s
Importance	531.959	10.3 s

9 mal mehr zusätzliche Zufallszahlen,  
aber nur 3 mal mehr Laufzeit:

- Gleichverteilung ist schneller zu erzeugen
- Abhängig vom cutoff:

Je mehr Zufallszahlen erzeugt werden sollen,  
desto größer muss der cutoff gewählt werden  
und desto ineffizienter wird das einfache rejection sampling



## Importance Sampling: Zusammenfassung

### Vorteile:

- Sampeln bis Unendlich möglich
- Effizienter als einfaches rejection sampling
- Kein Dimensionalitätsproblem

### Nachteile:

- Finden einer geeigneten Majorantenfunktion oftmals schwierig bis unmöglich
- Einfaches rejection sampling kann schneller sein, als kompliziertes importance sampling

# Markov Chain Monte Carlo



In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

hand in developing the algorithm, and in other cases, the author is a leading authority.

## In this issue

Monte Carlo methods are powerful tools for evaluating the properties of complex, many-body systems, as well as nondeterministic processes. Isabel Beichl and Francis Sullivan describe the Metropolis Algorithm. We are often confronted with problems that have an enormous number of dimensions or a process that involves a path with many possible branch points, each of which is governed by some fundamental probability of occurrence. The solutions are not exact in a rigorous way, because we randomly sample the problem. However, it is possible to achieve nearly exact results using a relatively small number of samples compared to the problem's dimensions. Indeed, Monte Carlo methods are the only practical choice for evaluating problems of high dimensions.

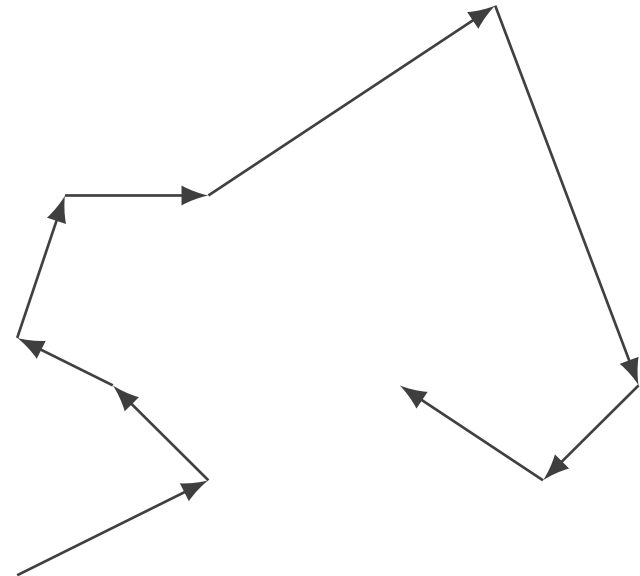
John Nash describes the Simplex method for solving linear programming problems. (The use of the word *programming* here really refers to scheduling or

**Statistische Methoden  
der Datenanalyse**

**Generation von Zufallszahlen**

# Markov-Kette

- Stochastische Übergänge
- Übergangswahrscheinlichkeit hängt nur vom Anfangs und Endzustand ab
- Dynamisches Sampeln



## Detailed Balance (Reversible Prozesse)

- Stationäre Verteilung  $f(x)$
- Übergangswahrscheinlichkeit  $M_{i \rightarrow j}$

$$M_{i \rightarrow j} f(x_i) = M_{j \rightarrow i} f(x_j)$$
$$\frac{M_{i \rightarrow j}}{M_{j \rightarrow i}} = \frac{f(x_j)}{f(x_i)}$$

Die Übergangswahrscheinlichkeit ist proportional zum Verhältnis der Zustände.

## Metropolis Algorithmus

Die Übergangswahrscheinlichkeit

$$M_{i \rightarrow j} = \min \left( 1, \frac{f(x_j)}{f(x_i)} \right)$$

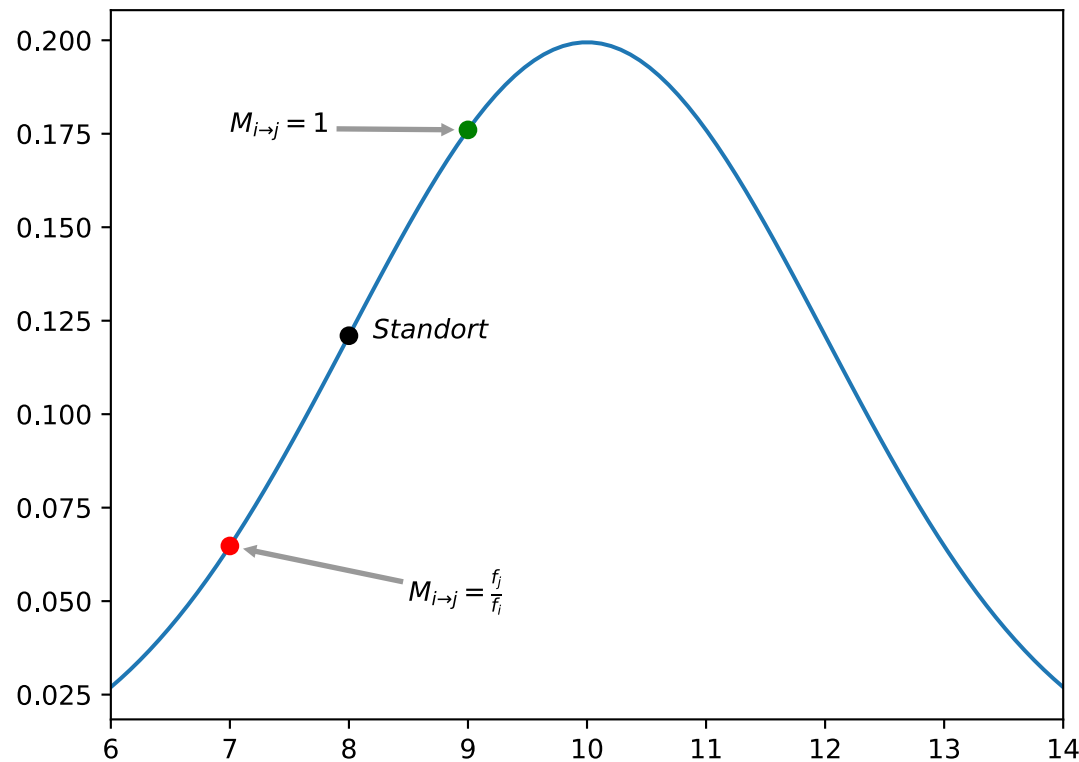
erfüllt die *detailed balance*  
Bedingung.

Generiere Zufallszahl:  $\xi \in (0, 1]$

$$x_{i+1} = \begin{cases} x_j, & \text{wenn } \xi \leq M_{i \rightarrow j} \\ x_i, & \text{wenn } \xi > M_{i \rightarrow j} \end{cases}$$

Wie wird  $x_j$  bestimmt?

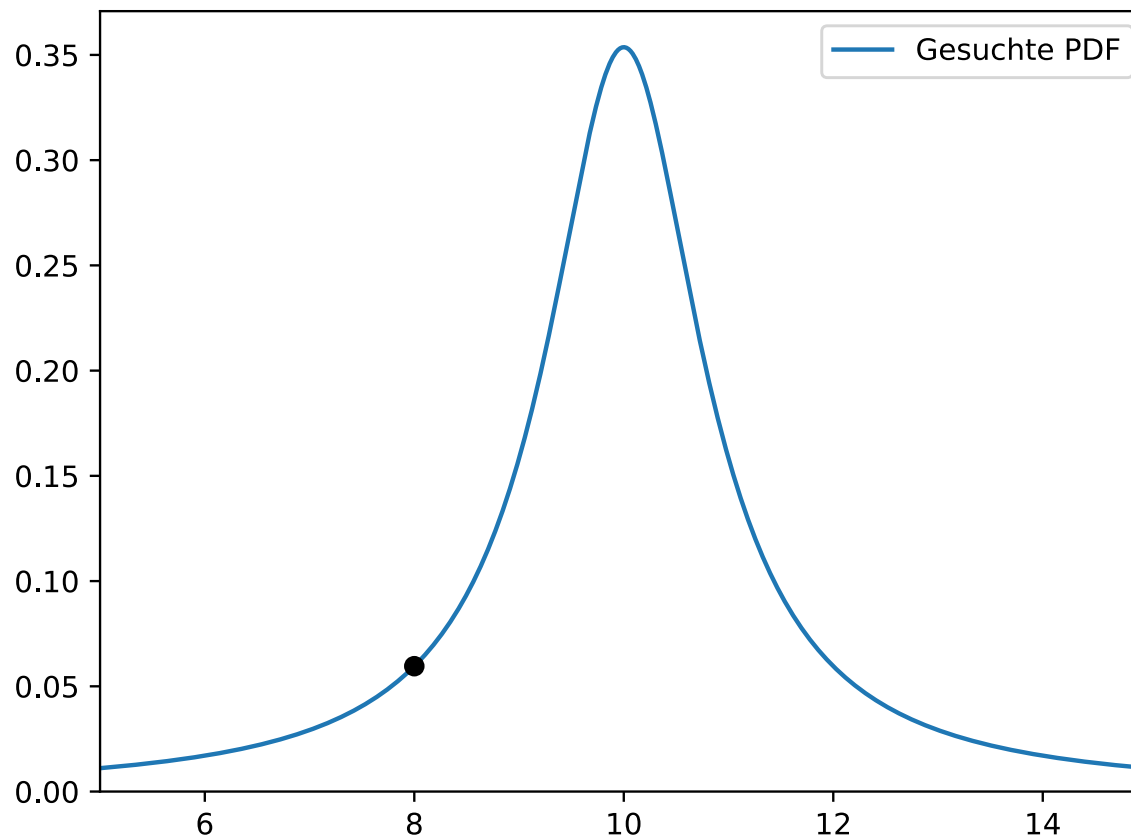
Zufälliges Ziehen aus einer festgelegten Schrittweitenverteilung



Generation von Zufallszahlen

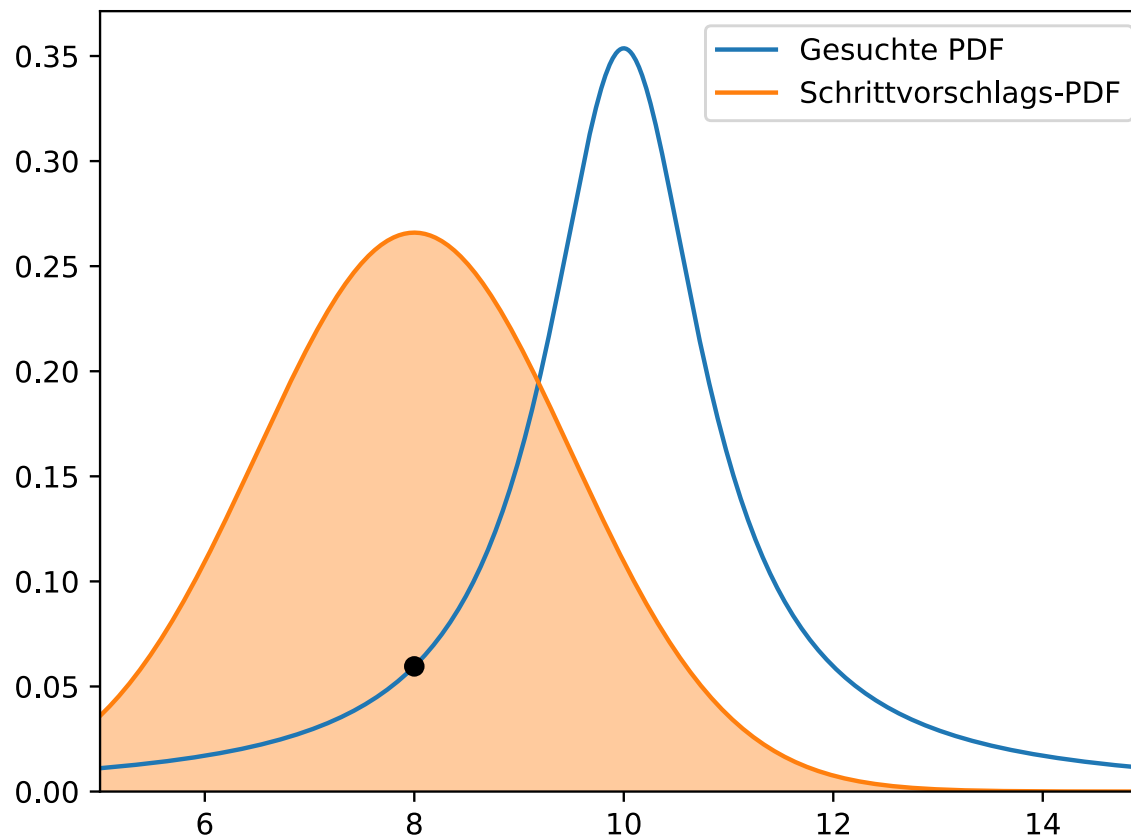
## Bestimmung der Schrittweite

Starte am Punkt i



## Bestimmung der Schrittweite

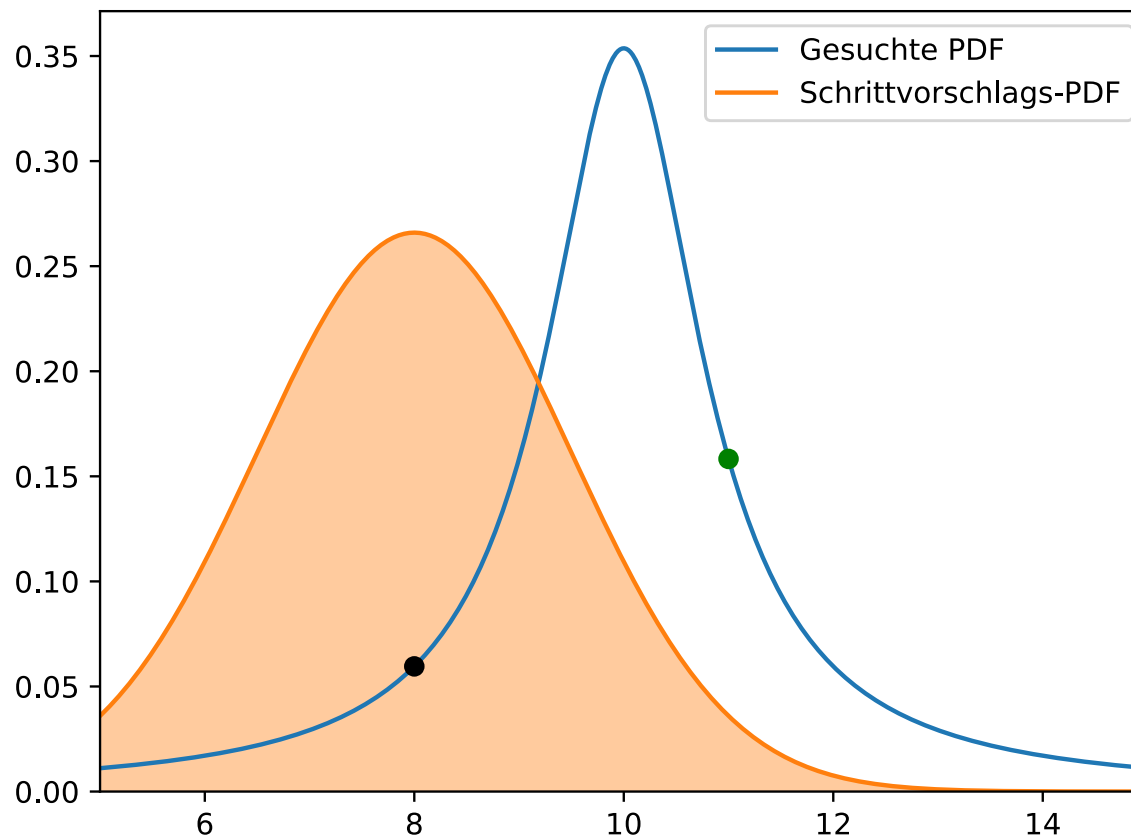
Ziehe nächsten Schritt aus Schrittvorschlags-PDF  $g(x)$





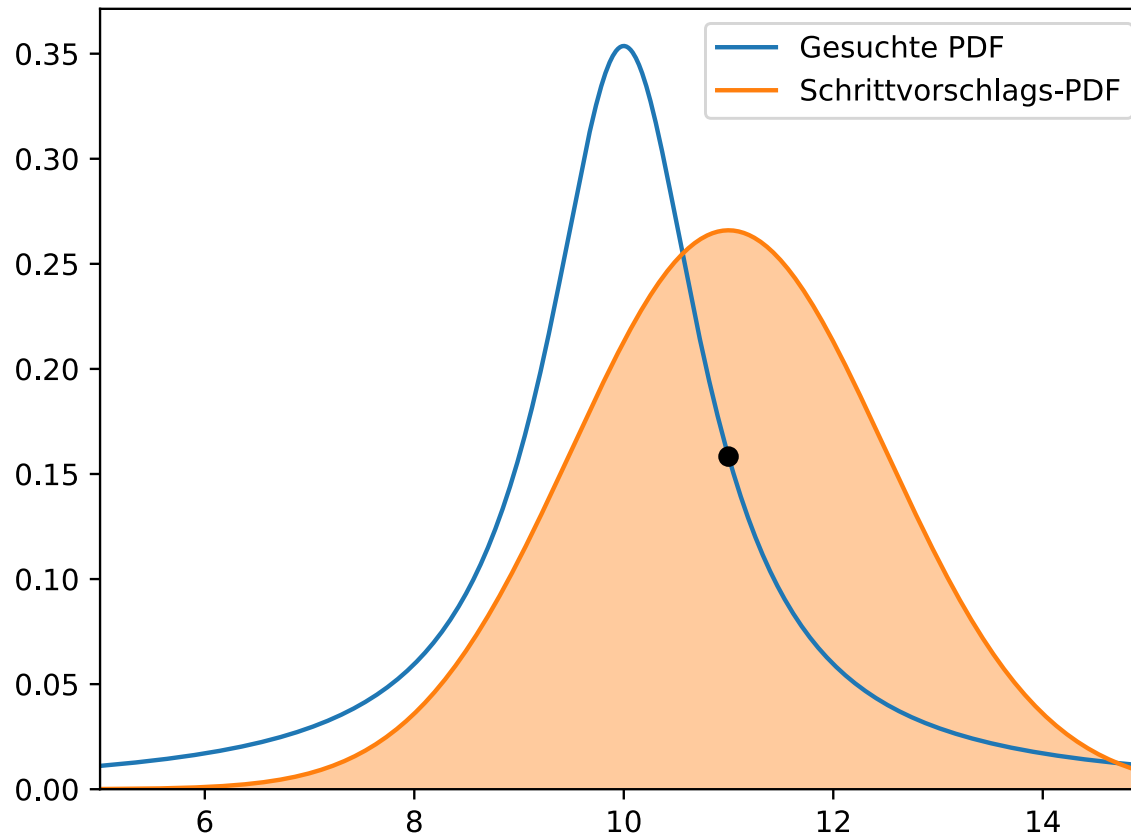
## Bestimmung der Schrittweite

Entscheide, ob Schrittorschlag akzeptiert wird



## Bestimmung der Schrittweite

Ziehe nächsten Schritt aus der Verteilung  $g$  (anderer Mittelwert)



## Metropolis-Hastings Algorithmus

Schrittorschlag bei  
nicht-symmetrischer Verteilung

$$g_{i \rightarrow j} > g_{j \rightarrow i}$$

erfüllt nicht die *detailed balance*

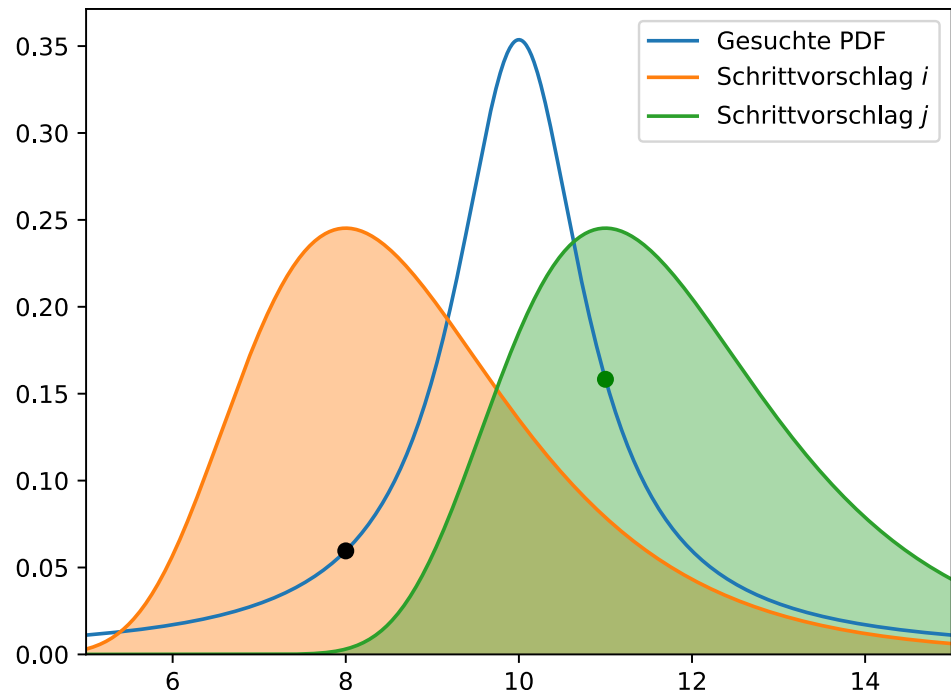
Bedingung, aber

$$M_{i \rightarrow j} f(x_i) g_{i \rightarrow j} = M_{j \rightarrow i} f(x_j) g_{j \rightarrow i}$$

Erfüllt die Bedingung.

Die Übergangswahrscheinlichkeit ist



$$M_{i \rightarrow j} = \min \left( 1, \frac{f(x_j) g(x_j | x_i)}{f(x_i) g(x_i | x_j)} \right)$$



## MCMC Algorithmus

- Initialisiere:  $t = 0$ , Startpunkt  $x_0$
- Wiederhole:
  1. Schlage Schritt vor: Ziehe  $x'$  aus  $g(x|x_t)$
  2. Berechne Akzeptanzwahrscheinlichkeit:
$$p = \min \left( 1, \frac{p(\bar{x})}{p(x_t)} \frac{g(x_t|\bar{x})}{g(\bar{x}|x_t)} \right)$$
  3. Akzeptanzschritt:
    1. Ziehe gleichverteilte Zufallszahl  $\xi \in [0, 1)$
    2. Wenn  $\xi \leq p$ , akzeptiere  $x' \Rightarrow x_{t+1} = x'$
    3. Sonst: Verwerfe  $x' \Rightarrow x_{t+1} = x_t$
  4. Erhöhe:  $t = t + 1$

## Autokorrelation sampling efficiency

- Zu kleine Akzeptanzrate ( $< 30\%$ )
  - Zu kleine Landschaft der Verteilung wird erkundet
  - Wenige Punkte repräsentieren die Verteilung
  -  Zu große Schrittweite
- Zu große Akzeptanzrate ( $> 70\%$ )
  - Viele Punkte nahe beinander
  - Flacher bereich der Verteilung
  - Verhältnis nahe 1
  -  Zu kleine Schrittweite

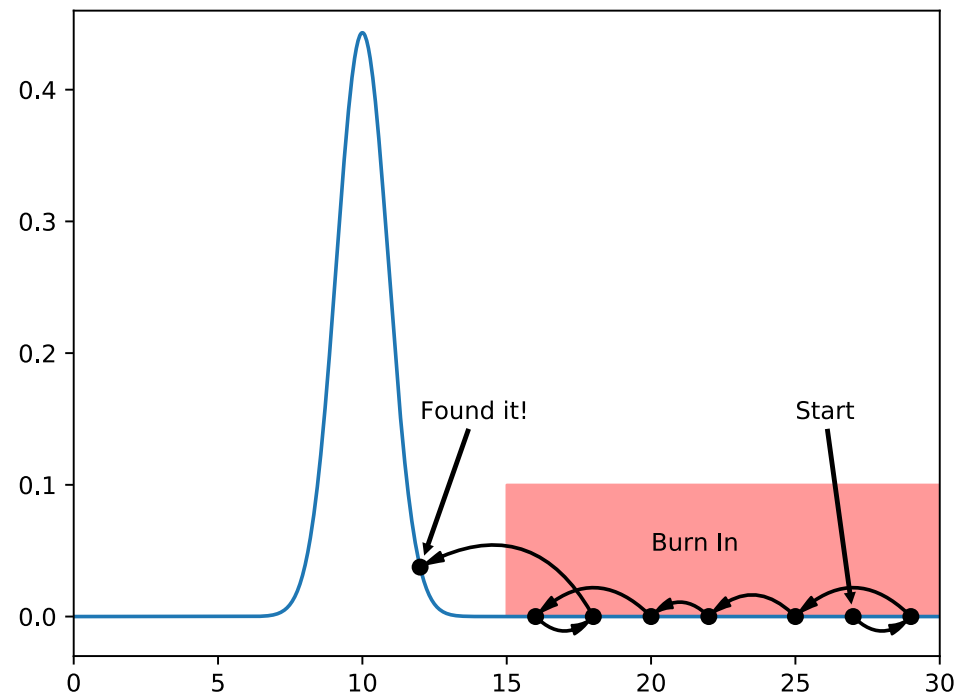
## Burn-In Phase

Szenario, wenn Startpunkt weit vom  
signifikanten Bereich entfernt liegt

- Flache Verteilung
- Großteil der Punkte nicht im  
signifikantem Bereich
- Findet signifikanten Bereich nicht

Ansatz

- Verwerfen der Burn-In Phase
- Ensemble von *Walkern*



## Probleme und Ausblick

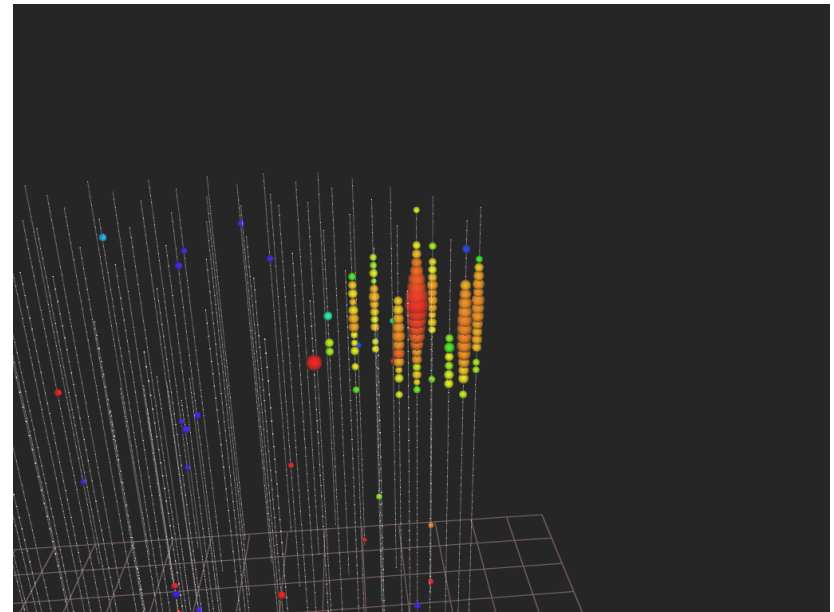
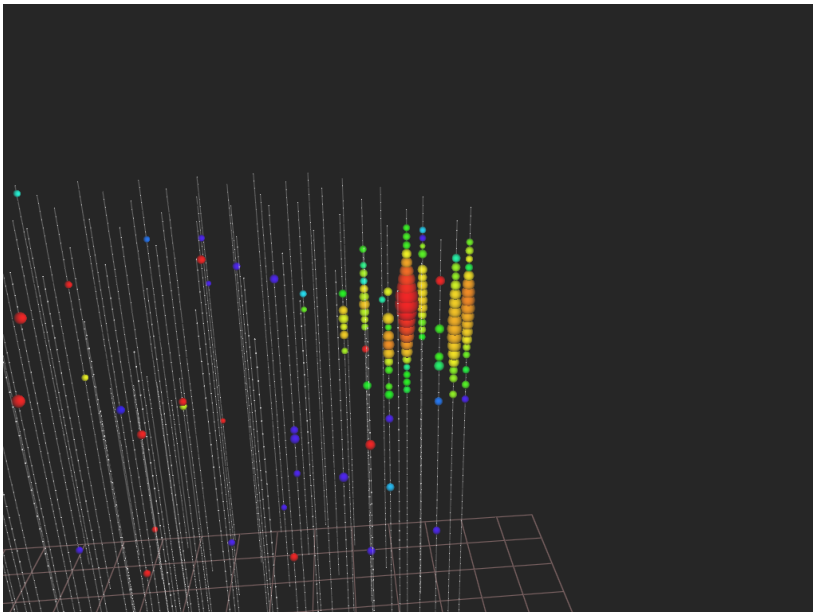
- Probleme
  - Kein unabhängiges Sampling
  - Startpunktabhängigkeit
  - Schrittweitenabhängigkeit

### Weitere Ansätze/Modifizierungen

- Hamilton Monte-Carlo:
  - Berücksichtigt Zustand und Gradient
  - Analogon: potentielle Energie und kinetische Energie
  - Schrittweitenverteilung abhängig vom Gradienten

## Monte-Carlo-Simulationen

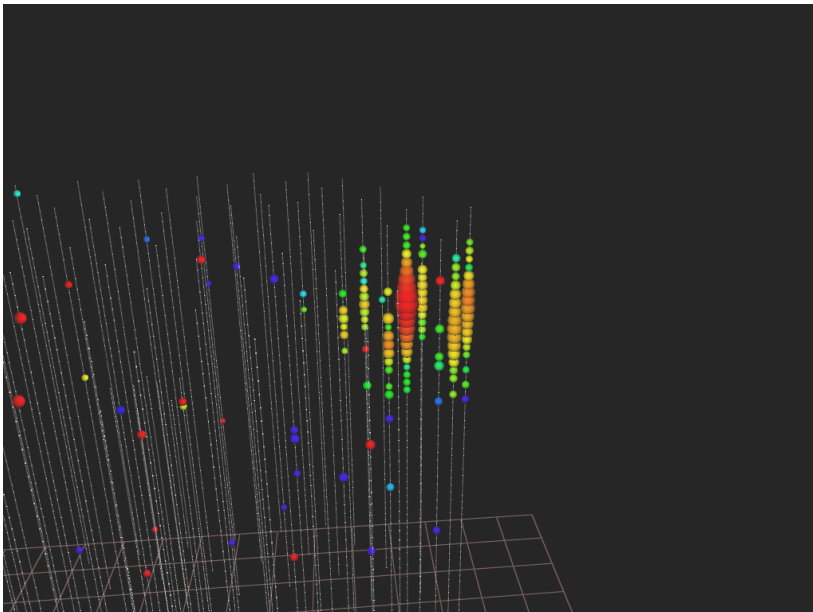
- Welches von beiden Events ist simuliert, welches gemessen?



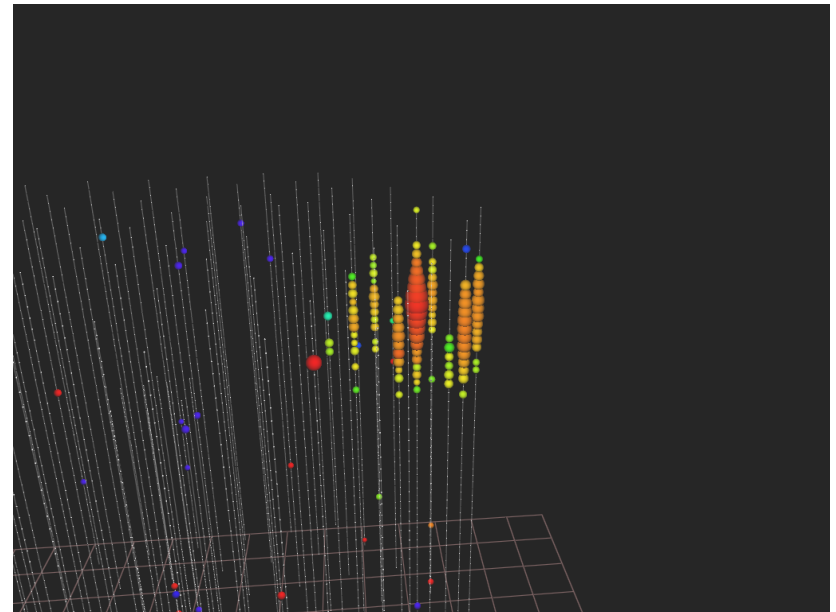


## Monte-Carlo-Simulationen

Gemessen



Simuliert

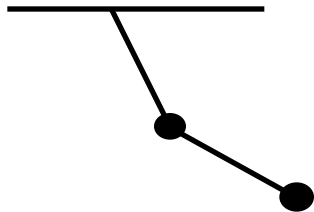


## Analytische Simulationen (ohne Monte-Carlo)

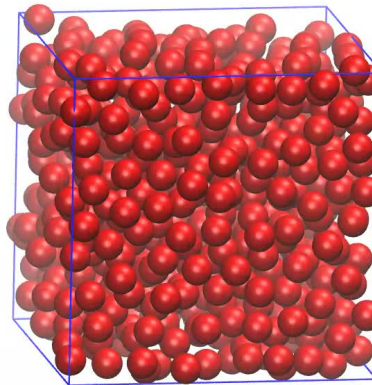
- Analytisches Lösen von Differentialgleichungen
- Aus dem Anfangszustand ist der Endzustand berechenbar

➡ Deterministisch

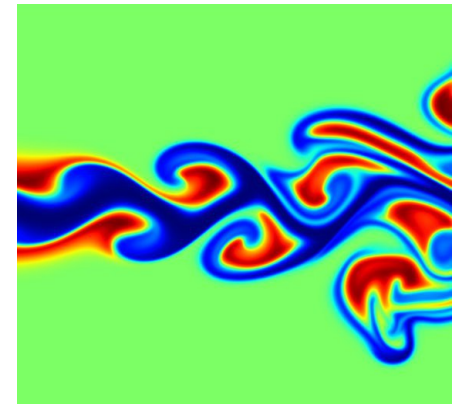
Doppelpendel  
(Euler-Lagrange)



Moleküldynamik  
(Newton)



Strömungen  
(Navier-Stokes)



## Monte-Carlo-Simulation

### **Stochastische Prozesse**

Kein gleiches Verhalten bei gleichen Startbedingungen

#### Festkörperphysik:

- Ensemblesimulation
- Berechnung von Integralen, Mittelwerten durch Mittelung über zufällig gezogenen Stichproben
- Comp. Phys: Mittlere Energie eines 2D Ising-Modells

#### Teilchenphysik (hier im Fokus):

- Einzelsimulationen
- Zufälliges Generieren, Propagieren, Detektieren und Rekonstruieren von einzelnen Ereignissen
- Mittelung über statistische Prozesse hier nicht sinnvoll

## Monte-Carlo-Simulationskette

- Ein Ereignis ist eine Überlagerung vieler komplexer Prozesse  
Analytische Lösung des Gesamtprozesses nicht möglich
- Behandlung der einzelnen Prozesse getrennt voneinander
- Für jeden Schritt: Ziehe Zufallszahl aus Verteilung und bestimme Ergebnis
- Abdeckung des Parameterraums durch häufiges Wiederholen



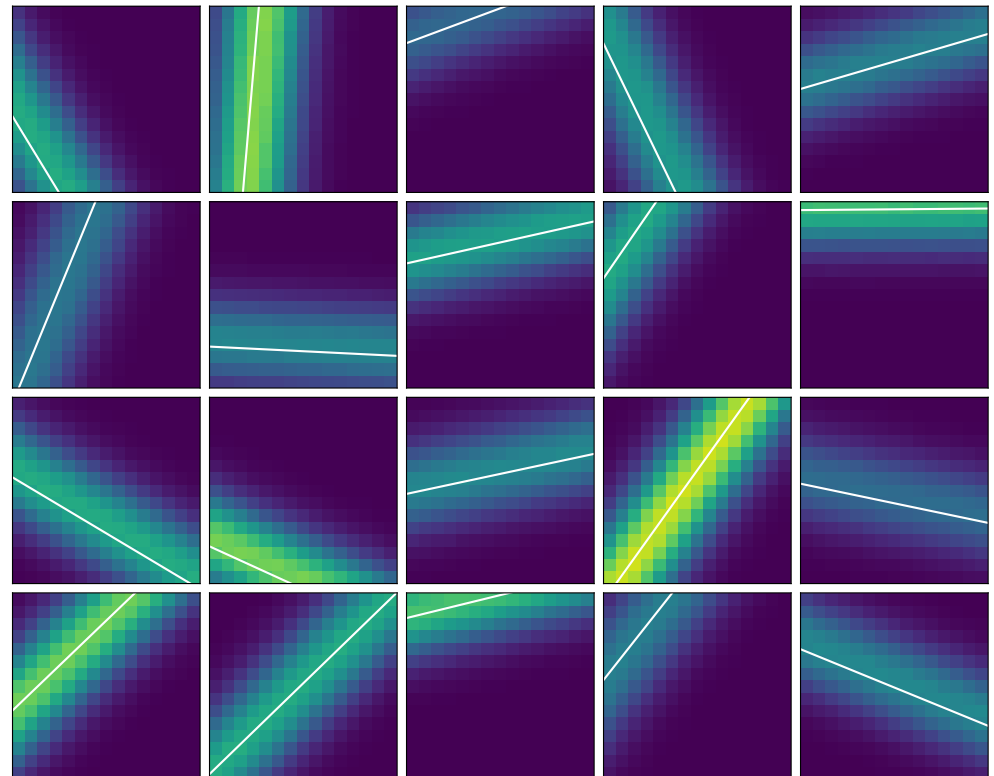
## Warum wir Monte-Carlo-Simulationen brauchen

Beispiel: Muonspur im Detektor

Annahmen

- Konstante Muonenergie
- Gaußartige Abschwächung des Signals

Simulation ohne Monte-Carlo



Generation von Zufallszahlen

Statistische Methoden  
der Datenanalyse

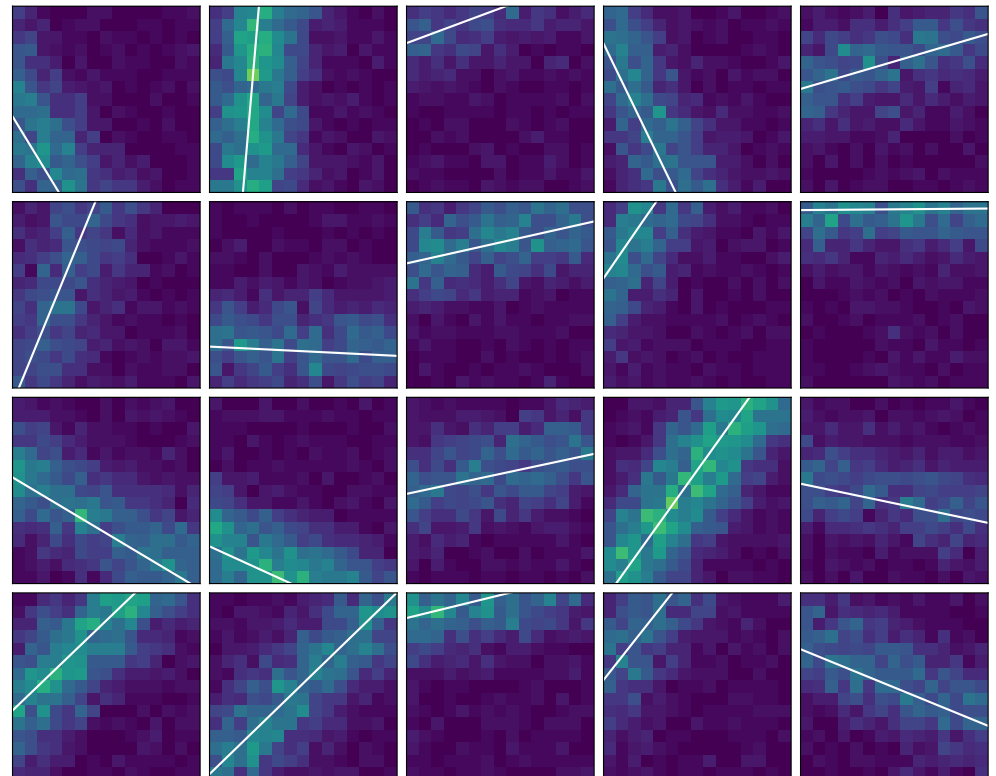
## Warum wir Monte-Carlo-Simulationen brauchen

Beispiel: Muonspur im Detektor

Annahmen

- Konstante Muonenergie
- Gaußartige Abschwächung des Signals
- Poissonverteilte Photonhits

Simulation mit Monte-Carlo



Generation von Zufallszahlen

Statistische Methoden  
der Datenanalyse

## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

Initialisiere:

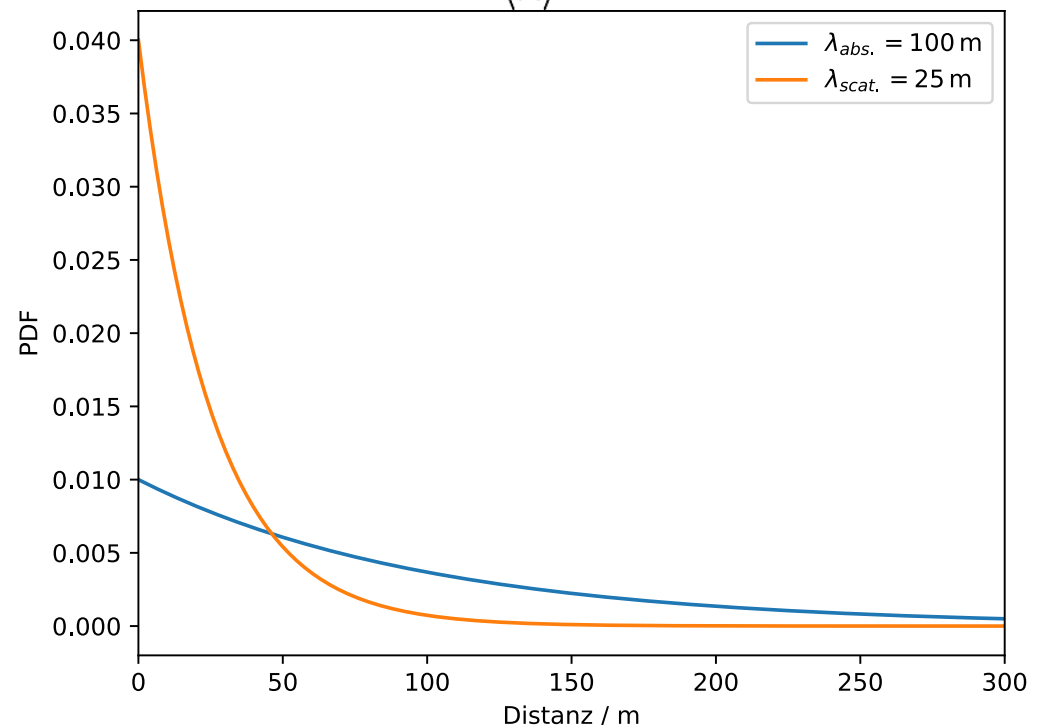
- Startposition
- Richtung
- Mittlere Absorptionslänge  $\langle \lambda_{\text{abs}} \rangle$
- Mittlere Streulänge  $\langle \lambda_{\text{scat}} \rangle$
- Mittlerer Streuwinkel  $\langle \cos \theta \rangle$

## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

Ziehe exponentialverteilte Absorptions- und Streulänge mit Inversionsmethode

$$p(x) = \frac{1}{\langle \lambda \rangle} e^{-x/\langle \lambda \rangle}$$

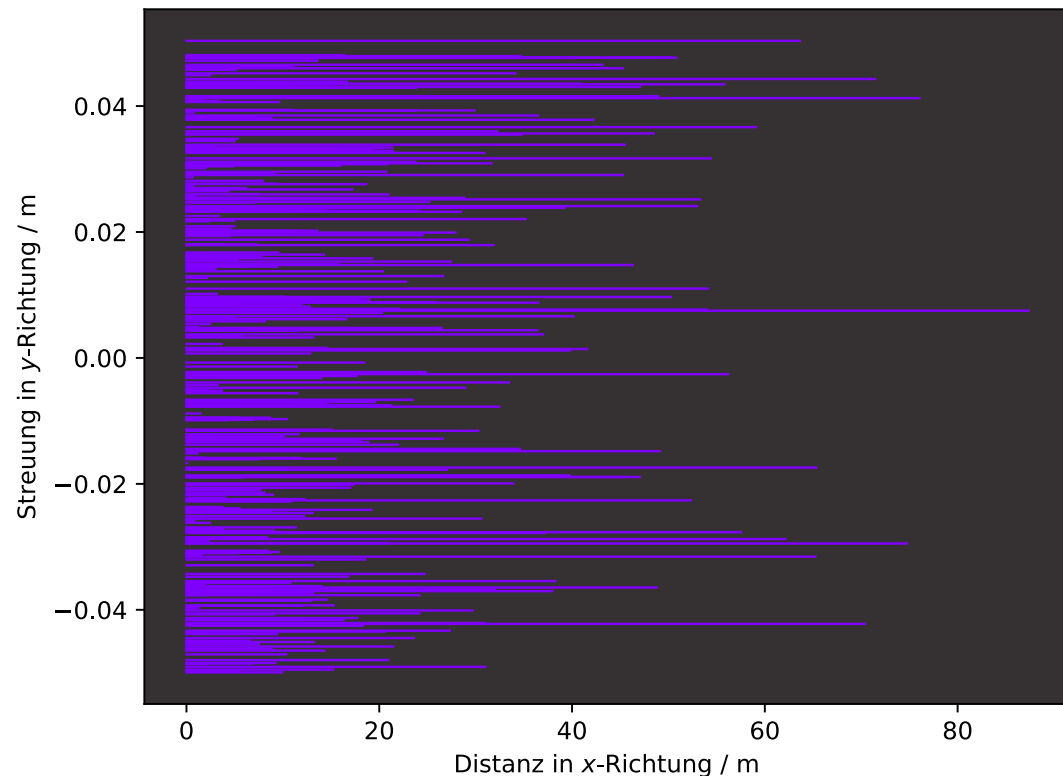




## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

Vor der 1. Streuung

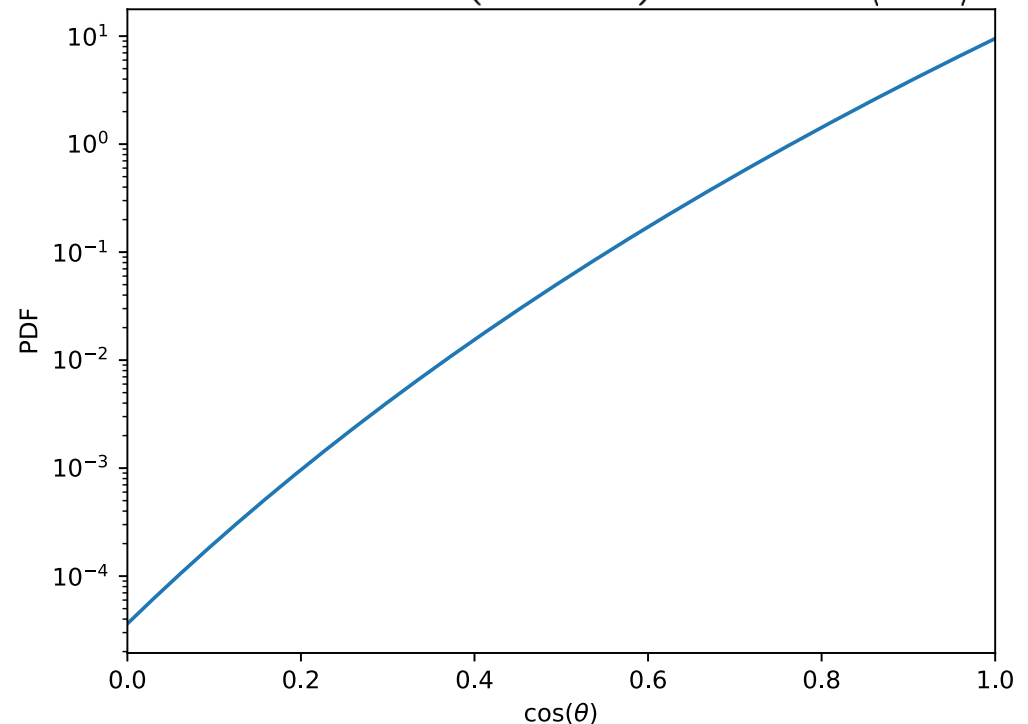


## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

Die Absorptions- und Streulänge sind exponentialverteilt

$$f(\cos \theta) = \frac{1 + \alpha}{2} \left( \frac{1 + \cos \theta}{2} \right)^\alpha, \alpha = \frac{2\langle \cos \theta \rangle}{1 - \langle \cos \theta \rangle}$$



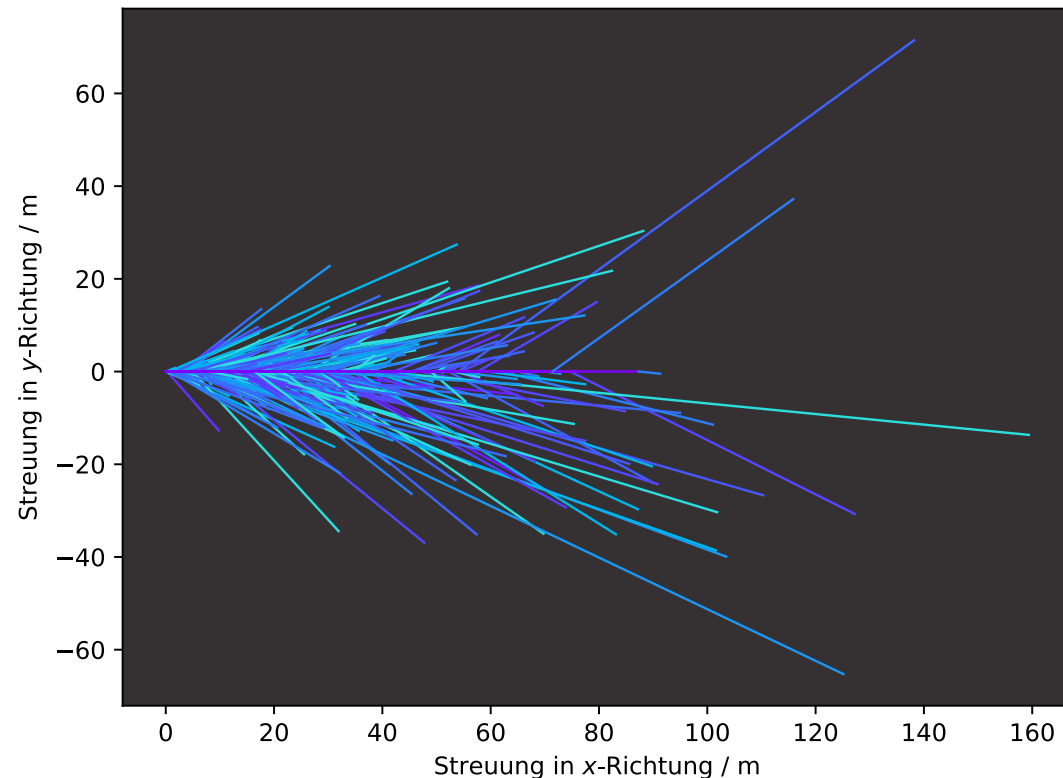
Generation von Zufallszahlen

Statistische Methoden  
der Datenanalyse

## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

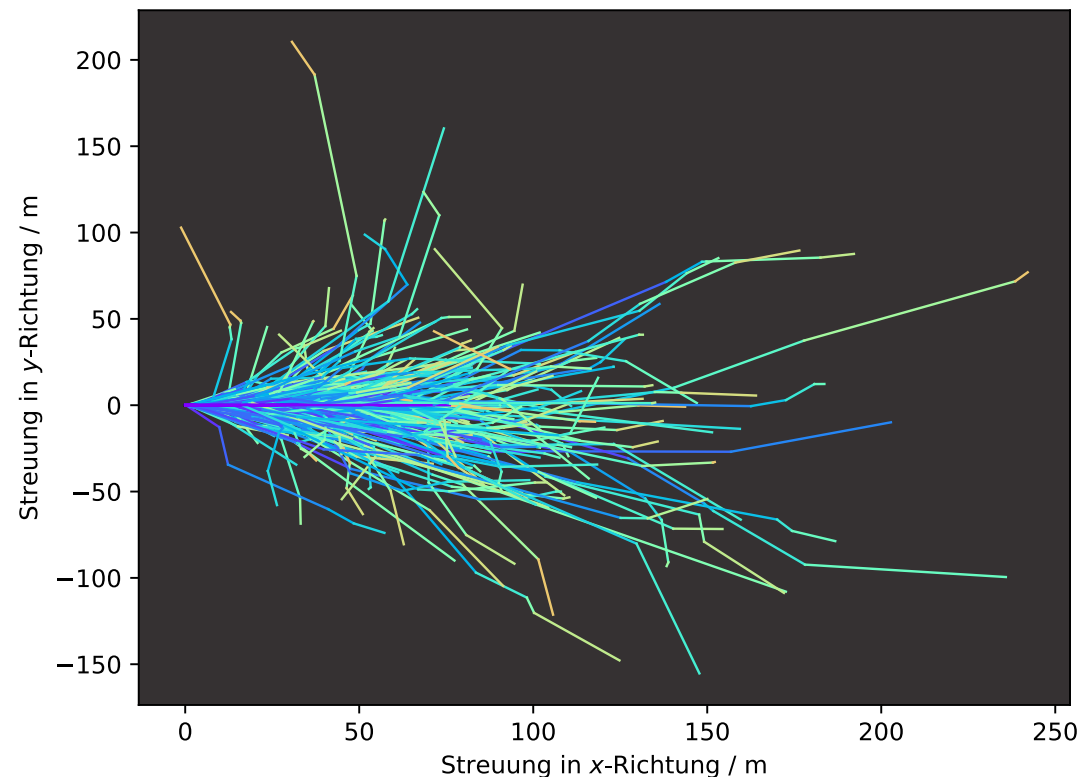
Nach der 1. Streuung



## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

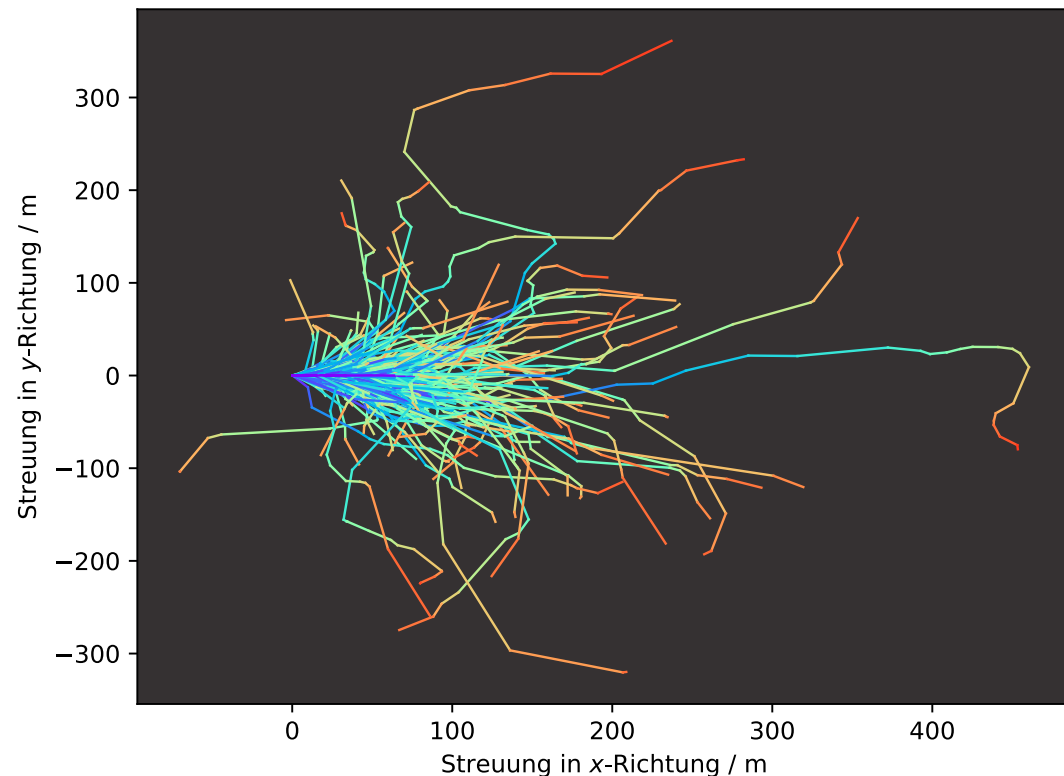
Nach 5 Streuungen



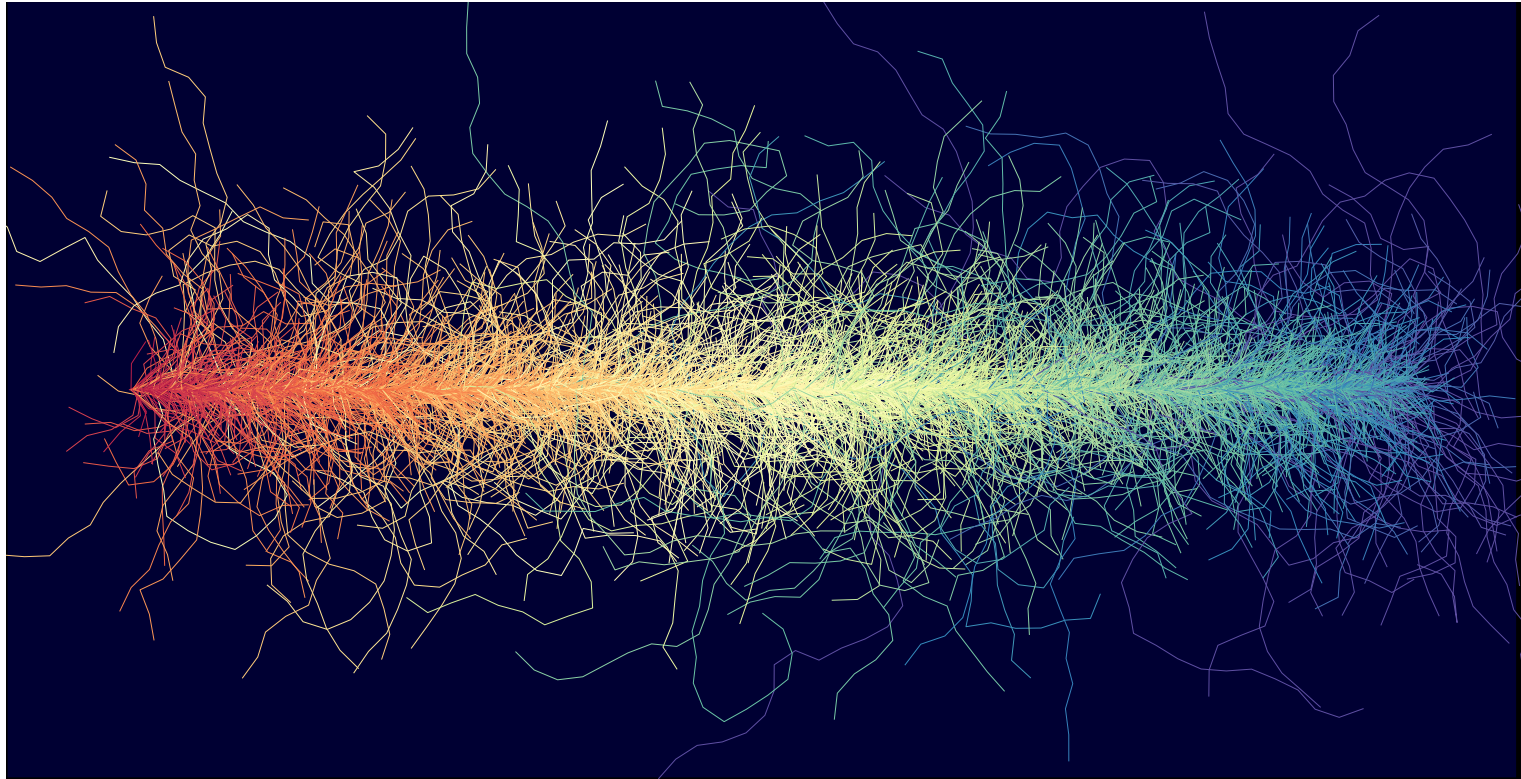
## Beispiel: Photonpropagation

1. Initialisierung
2. Ziehen der Absorptionslänge
3. Propagation und Streuung bis Absorptionslänge erreicht ist
  - I. Ziehen der Streulänge
  - II. Propagation zum nächsten Streupunkt
  - III. Ziehen des Streuwinkels
  - IV. Drehung um Streuwinkel

### Propagation bis zur Absorption



## Photon Propagation entlang eines Muonpfades



Hier noch keine stochastischen Verluste des Muons

## Teilchenpropagation

- Analog zur Photonpropagation
  - Absorption → Zerfall
  - Streuung → stochastischer Energieverlust + Streuung
  - + kontinuierliche Verluste

## Einschub: Ziehung der Zufallsereignisse aus Verteilung

- Photonpropagation
  - PDF invertierbar (Exponentialverteilung)  
→ Inversionsmethode
- Muonpropagation
  - Komplexe Verteilungen (Wirkungsquerschnitte)  
oftmals nur numerisch vorhanden  
oder es muss zunächst integriert werden
  - Frage: Wie wird hier gesampelt?



## Einschub: Ziehung der Zufallsereignisse aus Verteilung

- Photonpropagation
  - PDF invertierbar (Exponentialverteilung)
    - ➡ Inversionsmethode
- Muonpropagation
  - Komplexe Verteilungen (Wirkungsquerschnitte)  
oftmals nur numerisch vorhanden  
oder es muss zunächst integriert werden
  - Frage: Wie wird hier gesampelt?
    - ➡ Inversionsmethode, da effizienter:
      - MC-Simulationen sind ressourcenintensiv
      - Großer MC Datensatz für Analysen notwendig
      - Interpolation statt Integration
      - Abspeichern von Tabellen statt Neuberechnung

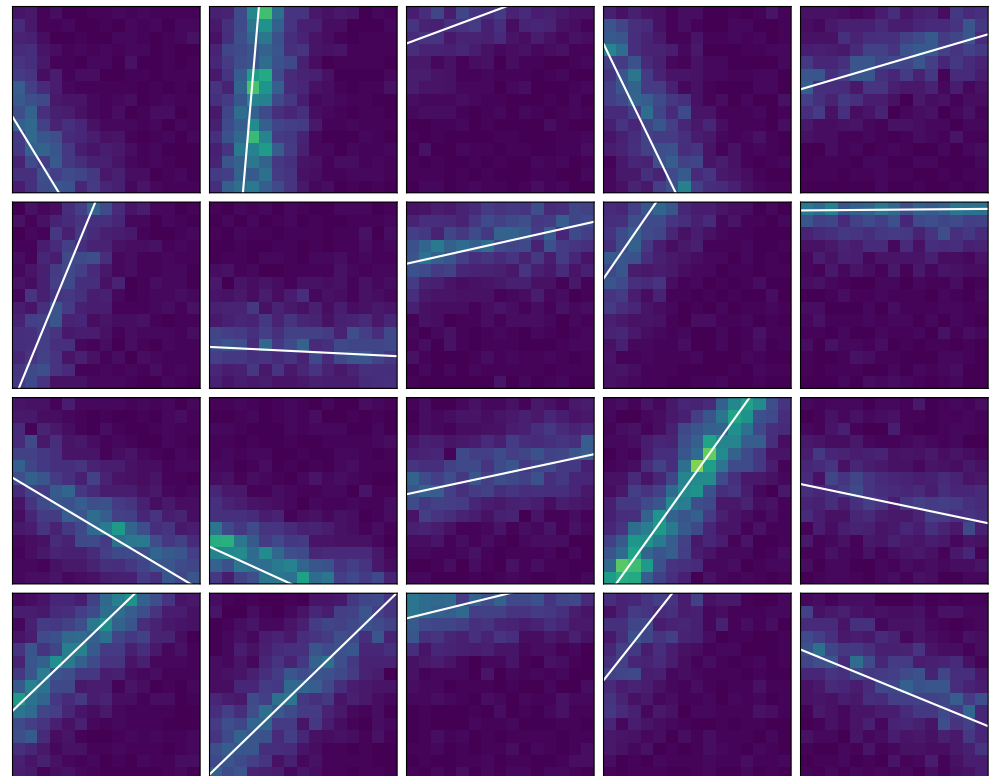
## Warum wir Monte-Carlo-Simulationen brauchen

Beispiel: Muonspur im Detektor

Annahmen

- Konstante Muonenergie
- Gaußartige Abschwächung des Signals
- Poissonverteilte Photonhits
- Stochastische Energieverluste

Muon mit Energieverlusten



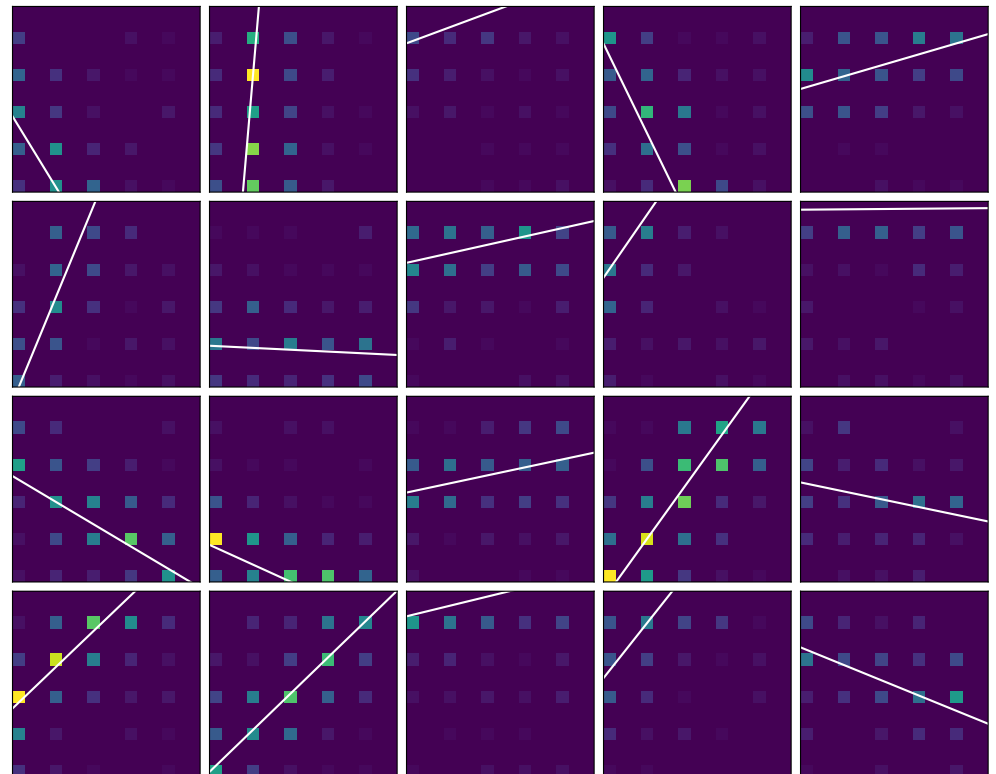
## Warum wir Monte-Carlo-Simulationen brauchen

Beispiel: Muonspur im Detektor

Annahmen

- Konstante Muonenergie
- Gaußartige Abschwächung des Signals
- Poissonverteilte Photonhits
- Stochastische Energieverluste
- Geringe Auflösung

Detektor mit geringer Auflösung



Generation von Zufallszahlen

Statistische Methoden  
der Datenanalyse

## Vollständige Monte-Carlo-Simulationskette

- Bisher nur Monte-Carlo-Propagation
  - Generator
    - Energiespektrum der Teilchen
    - Teilchenfluss
  - Detektor
    - Akzeptanzwahrscheinlichkeit
    - Trigger
    - Elektronik (z.B. weißes Rauschen)
- Vollständig simulierte Events entsprechen Messdaten
- Rekonstruktion des einzelnen Ereignisses

