Karlsruhe Institute of Technology

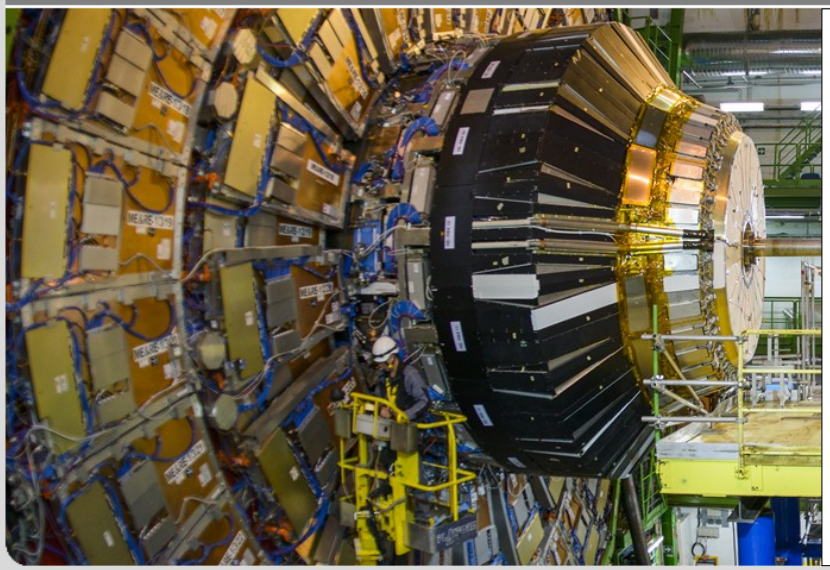
# Rechnernutzung in der Physik

## Teil 3 – Statistische Methoden in der Datenanalyse

**Roger Wolf**

8. Dezember 2015

INSTITUTE OF EXPERIMENTAL PARTICLE PHYSICS (IEKP) – PHYSICS FACULTY



- Grundlagen der Wahrscheinlichkeitstheorie, Werkzeuge zur statistischen Datenanalyse
- Gängige Wahrscheinlichkeitsverteilungen
- Monte-Carlo Methoden
- Parameterschätzung
- Hypothesentests

Kapiel 3.3:

# Charakterisierung von Wahrscheinlichkeitsverteilungen

- Zufallsvariablen und Wahrscheinlichkeitsdichten.
- Charakterisierung durch Quantilen, Lagemaß.
- Erwartungswert, algebraische Momente, Varianz.
- Mehrdimensionale Wahrscheinlickeitsdichten, Kovarianz, Korrelationen.
- Funktionen von Zufallsvariablen, Gaußsche Fehlerfortpflanzung.

Kapiel 3.4:

# Beispiele gängiger Wahrscheinlichdichteverteilungen

- Uniforme Verteilung, Exponentialverteilung (→ jeder Experimentausgang gleichwertig).
- Binomialverteilung, Poissonverteilung (→ unterscheide günstige/mögliche Experimentausgänge).
- Normalverteilung, Log-Normalverteilung, $\chi^2$-Verteilung (→ viele unabhängige Messungen mit einem bestimmten Ausgang, $\mu$ in einem bestimmten Intervall $\sigma^2$ ).

Kapiel 3.5:

# Monte Carlo Methoden

# MonteCarlo Methode (MC)

- Numerisches Verfahren aus der Stochastik, um analytisch nicht oder nur aufwändig lößbare Probleme mit Hilfe der Wahrscheinlichkeitstheorie zu lößen.
- Anwendungsgebiete:
  - Numerische Mathematik (Integration, Optimierung, Faltung, …).
  - Angewandte Statistik (Bestimmung von Korrelationen, Fehlerfortpflanzung, Hypothesentests, …).(1)
  - Nachbildung komplexer Prozesse mit statistischem Verhalten (Vielteilchensysteme, Teilchenphysik, …).
- Historie:
  - Erste Idee: Enrico Fermi 1930er Jahre.
  - Erste Ausführung: Stanislaw Ulam, John von Neumann 1947 (Los Alamos Projekt).
  - Namensgebung durch John von Neumann (als code Name innerhalb des Projektes).

(1)
Sie werden vorraussichtlich ein Bsp für die Nutzung von MC Methoden zur Fehlerfortpflanzung in der letzten Vorlesung sehen.

# (Pseudo-)Zufallszahlen

- Die Monte Carlo Methode basiert auf dem Gesetz der starken Zahlen und beginnt mit einer Reihe gleichvereteilter Zufallszahlen.
- Diese Reihe kann entweder physikalisch bestimmt werden (z.B. Werfen eines Würfels, Zeitspanne $\Delta t$ zwischen zwei Zerfällen eines radioaktiven Präparats, …), oder mit Hilfe eines Zufallszahlengenerators als Pseudo-Zufallszahlen.

Pseudo-Zufallszahlen:

Ergebnis einer de-terministischen Sequenz (insb. re-produzierbar)

Innerhalb eines vorgegebenen Intervalls nach bester Möglichkeit gleichverteilt.

- In einem zweiten Schritt werden die gleichverteilten (Pseudo-)Zufallszahlen in eine beliebige Wahrscheinlichkeitsdichte transformiert.

# Zufallszahlengeneratoren

- **Beispiel:** multiplikative Kongruenzgeneratoren

$$y_i = \left(\left(\sum_{k=1}^{n} a_k y_{i-k}\right) + b\right) \mod m \qquad n \in \mathbb{N}^+ \text{ (Zustandwerte)}$$

$$\text{für } i > n \qquad m \in \{2, 3, 4, 5, \ldots\} \text{ (Modul)}$$

$$a_1, a_2, \ldots a_n \in \{0, 1, 2, \ldots m-1\}, a_n > 0 \text{ (Faktoren)}$$

$$b \in \{0, 1, 2, \ldots m-1\} \text{ (Inkrement)}$$

$$y_1, y_2, \ldots y_n \in \{0, 1, 2, \ldots m-1\} \text{ (Startwerte, "seeds")}^{(2)}$$

- Zustand des Generators vor Erzeugung des Wertes $y_i$ bei Vorgabe von $(a_k, b, m, n)$ durch Werte $y_{i-n}, \ldots y_{n-1}$ vorgegeben.

- Wichtigste Eigenschaften:
  - Lange Periode, bevor sich die Sequenz der Pseudo-Zufallszahlen wiederholt.
  - Möglichst keine Korrelation zwischen (nächsten) Nachbarn.



(2)
Nicht alle 0 wenn b=0.

# Linearer Kongruenzgenerator (LCG)

- Einfacher aber sehr effizienter Algorithmus aus dieser Gruppe:

(3)

$$y_{i+1} = a \cdot y_i \mod m$$

- Problem:
  - Probabalistische Eigenschaften hängen von Wahl von $(a, m)$ ab und können sehr ungünstig ausfallen.
- Mögliche Abhilfe:
  - Wähle $m$ als möglichst große Primzahl und $a$ als Primitivwurzel aus $m$ .

**Bsp:** $a = 12345678$
$m = 98765432$

| seed index | value |
|---|---|
| 1 | 1 |
| 2 | 12345678 |
| 3 | 61728396 |
| 4 | 86419752 |
| 5 | 12345680 |
| 6 | 86419752 |
| 7 | 12345680 |
| 8 | 86419752 |
| 9 | ... |

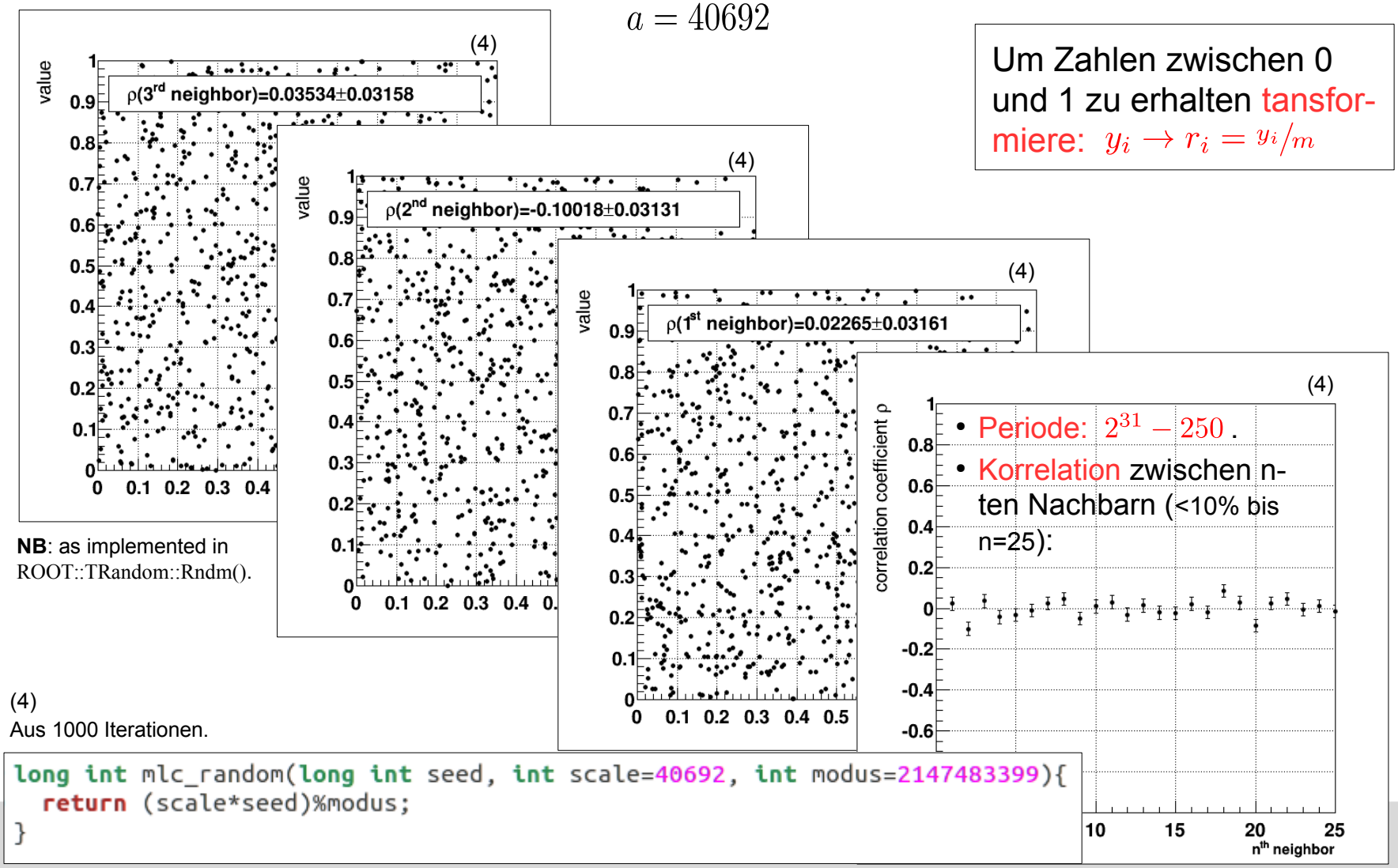


(3) Zuerst eingeführt von Derrick Henry Lemmer, 1949.

# Linearer Kongruenzgenerator (LCG)

- **Beispiel**: $y_{i+1} = a \cdot y_i \mod m$ $\quad m = 2147483399 = 2^{31} - 249$ (Primzahl)

$a = 40692$




Um Zahlen zwischen 0 und 1 zu erhalten tansformiere: $y_i \to r_i = y_i/m$

**NB**: as implemented in ROOT::TRandom::Rndm().

(4)
Aus 1000 Iterationen.

```
long int mlc_random(long int seed, int scale=40692, int modus=2147483399){
  return (scale*seed)%modus;
}
```

# Moderne Zufallszahlengeneratoren

- Der Lineare Kongruenzgenerator ist gut genug für die Zwecke dieser Vorlesung. State of the Art: Mersenne-Twister.[5]

**NB**: as implemented in ROOT::TRandom3::Rndm().

- Basierend auf Mersenne Primzahlen: $\mathcal{M}_n = 2^n - 1$ (nach Marin Mersenne (1588 – 1648)).
- 264 seeds (z.B. durch LCG).
- Periode: $2^{19937} - 1$!
- Je nach *seeding* garantiert unkorreliert bis zur Dimension 263.

wikipedia

```
#include <stdint.h>

uint32_t mersenne_twister() {
#define N     624
#define M     397
#define HI    0x80000000
#define LO    0x7fffffff
  static const uint32_t A[2] = { 0, 0x9908b0df };
  static uint32_t y[N];
  static int index = N+1;
  static const uint32_t seed = 5489UL;
  uint32_t  e;

  if (index > N) {
    int i;
    /* Initialisiere y mit Pseudozufallszahlen */
    y[0] = seed;

    for (i=1; i<N; ++i) {
      y[i] = (1812433253UL * (y[i-1] ^ (y[i-1] >> 30)) + i);
      /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier. */
      /* In the previous versions, MSBs of the seed affect   */
      /* only MSBs of the array mt[].                        */
      /* 2002/01/09 modified by Makoto Matsumoto             */
    }
  }

  if (index >= N) {
    int i;
    /* Berechne neuen Zustandsvektor */
    uint32_t h;
  ...
```



(5) Makoto Matsumoto, Takuji Nishimura, 1997.

# Moderne Zufallszahlengeneratoren

- Der Lineare Kongruenzgenerator ist gut genug für die Zwecke dieser Vorlesung. State of the Art: Mersenne-Twister.[5]

**NB**: as implemented in ROOT::TRandom3::Rndm().

- Basierend auf Mersenne Primzahlen: $\mathcal{M}_n = 2^n - 1$ (nach Marin Mersenne (1588 – 1648)).
- 264 seeds (z.B. durch LCG).
- Periode: $2^{19937} - 1$!
- Je nach *seeding* garantiert unkorreliert bis zur Dimension 263.

wikipedia

```
#include <stdint.h>

uint32_t mersenne_twister() {
#define N     624
#define M     397
#define HI    0x80000000
#define LO    0x7fffffff
  static const uint32_t A[2] = { 0, 0x9908b0df };
  static uint32_t y[N];
  static int index = N+1;
  static const uint32_t seed = 5489UL;
  uint32_t  e;

  if (index > N) {
    int i;
    /* Initialisiere y mit Pseudozufallszahlen */
    y[0] = seed;

    for (i=1; i<N; ++i) {
      y[i] = (1812433253UL * (y[i-1] ^ (y[i-1] >> 30)) + i);
      /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier. */
      In the previous versions, MSBs of the seed affect   */
      only MSBs of the array mt[].                        */
      2002/01/09 modified by Makoto Matsumoto             */



      dex >= N) {
      ;
    /* Berechne neuen Zustandsvektor */
    uint32_t h;
  ...
```

Zweiter Schritt in der Monte Carlo Methode: gleichverteilte (Pseudo-)Zufallszahlen $y_i$ auf eine beliebige Wahrscheinlichkeitsdichte $x_i$ zu transformieren.

(5) Makoto Matsumoto, Takuji Nishimura, 1997.

# Analytische Transformation

- **Möglichkeit-1**: analystische Transformation
  Finde eine Transformation $x(y)$ nach der die $x_i$ geeignet verteilt sind.

- Erinnerung and „*Funktionen von Zufallsvariablen*“ (cf. VL-09 slide 24): in diesem Fall waren $p(x)$ und $a(x)$ vorgegeben, $q(a)$ war gesucht.

- Hier: sowohl $p(x) = U_{[0,1]}(x)$ als auch $q(y)$ vorgegeben, $y(x)$ ist gefragt.

- Lösungsansatz wie zuvor:

$$\mathcal{P}(x' \leq x) = \mathcal{Q}(y(x' \leq y(x))$$

$$\int_{-\infty}^{x} p(x')\mathrm{d}x' = \int_{0}^{x} 1\ \mathrm{d}x' = \int_{-\infty}^{y(x)} q(y')\mathrm{d}y'$$

$$x(y) = \int_{-\infty}^{y} q(y')\mathrm{y}' = \mathcal{Q}(y)$$

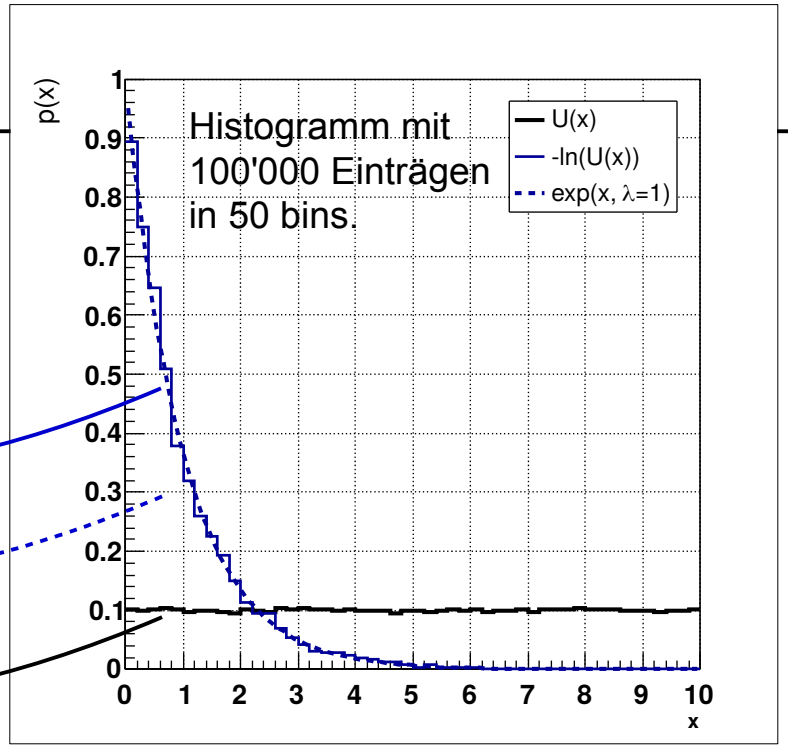
$$y(x) = \mathcal{Q}^{-1}(x(y))$$

# Beispiel: $\exp(y, \lambda) = \frac{1}{\lambda} e^{-y/\lambda}$ (7)

$-\ln(U_{[0,1]}(x))$ (gemäß (*) befüllt.)

$\exp(x, \lambda = 1)$

$U_{[0,1]}(x)$




**Beispiel:** (6)

$$q(y) = \exp(y, \lambda) = \frac{1}{\lambda} e^{-y/\lambda}$$

$$x(y) = \left[e^{-y'/\lambda}\right]_{-\infty}^{y} = \left(1 - e^{-y/\lambda}\right)$$

$$y(x) = -\lambda \ln(1 - y) \quad (*)$$

$$\int_{-\infty}^{x} p(x')\mathrm{d}x' = \int_{0}^{x} 1 \, \mathrm{d}x' = \int_{-\infty}^{y(x)} q(y')\mathrm{d}y'$$

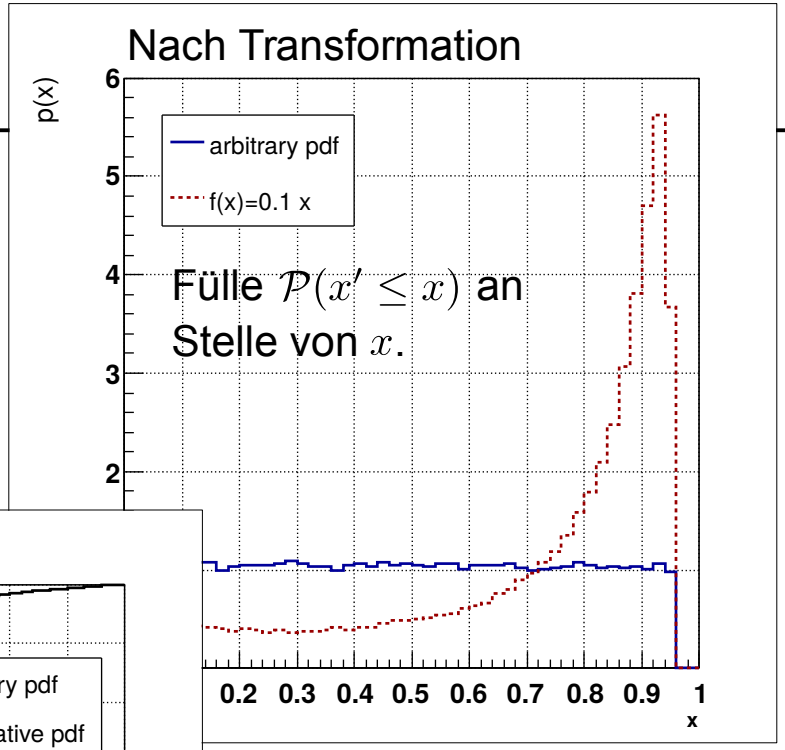
$$x(y) = \int_{-\infty}^{y} q(y')\mathrm{y}' = \mathcal{Q}(y)$$

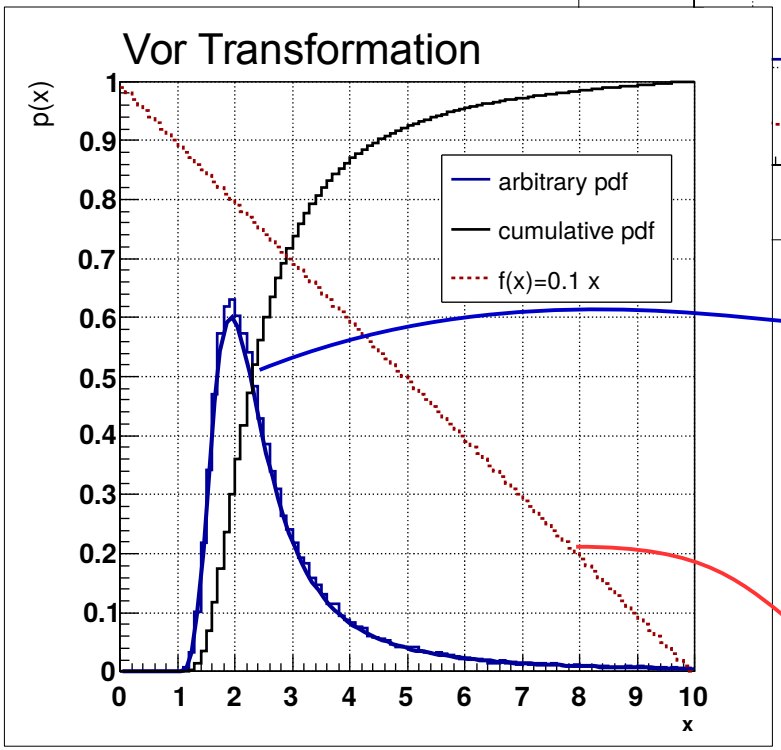
$$y(x) = \mathcal{Q}^{-1}(x(y))$$

(6) Umkehrung hier nicht eindeutig: y(x) ist ebenso exponentiell verteilt wie y(1-x).

(7) cf. VL-09 slide 33.

# Beispiel: $p(x)$ beliebig[8]




„beliebig“ verteilte Zufalls-variable.

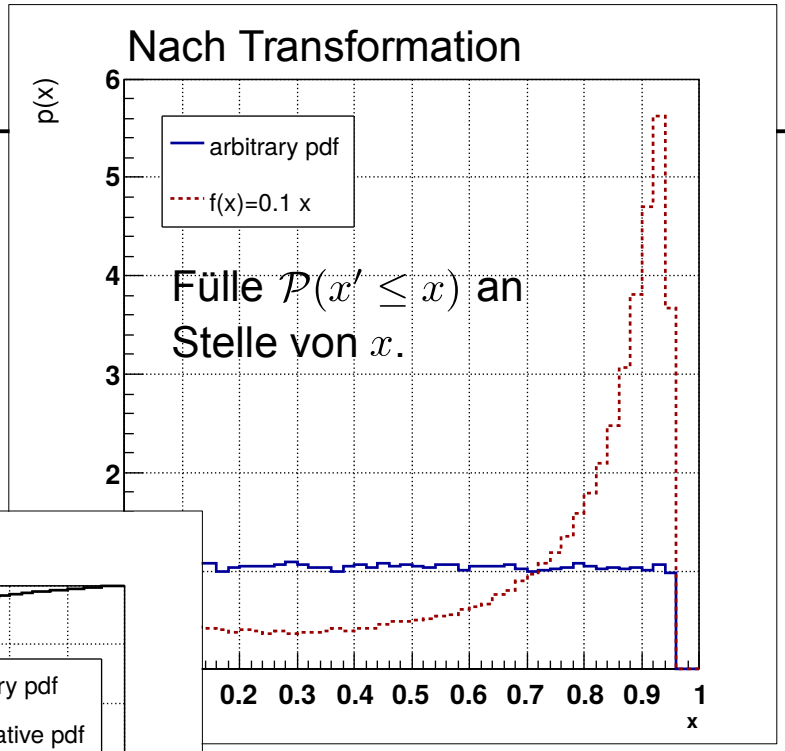
„Referenzverteilung“ (zeigt Wirken der Transformation).

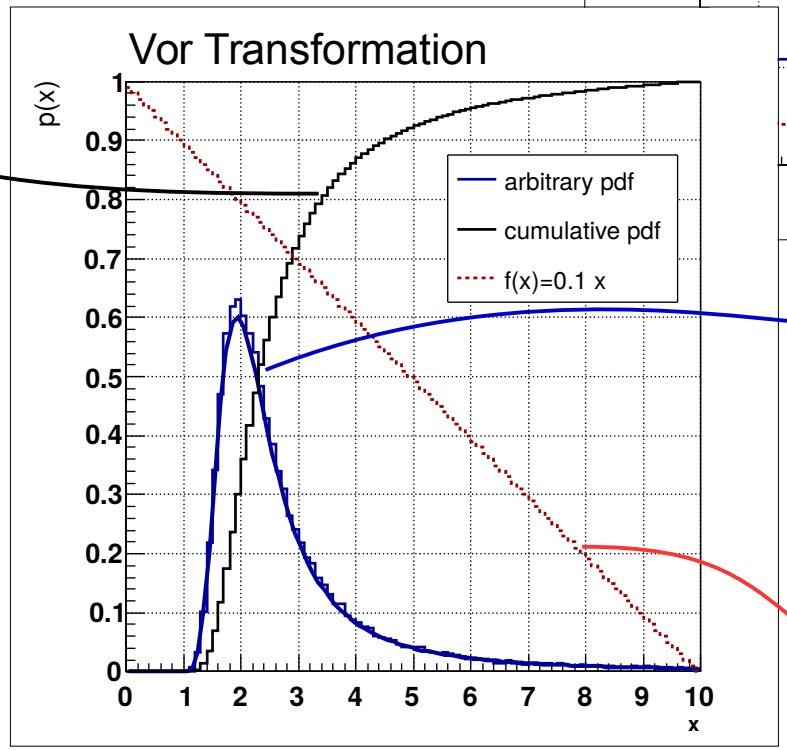
(8) cf. VL-09 slide 31.

# Beispiel: $p(x)$ beliebig[8]

Kummulative Verteilungsfunktion, $\mathcal{P}(x)$:

- Ein-eindeutige Abbildung.
- Jedem möglichen Wert $x$ wird genau ein Wert $\mathcal{P}(x) \in [0, 1]$ zugewiesen.



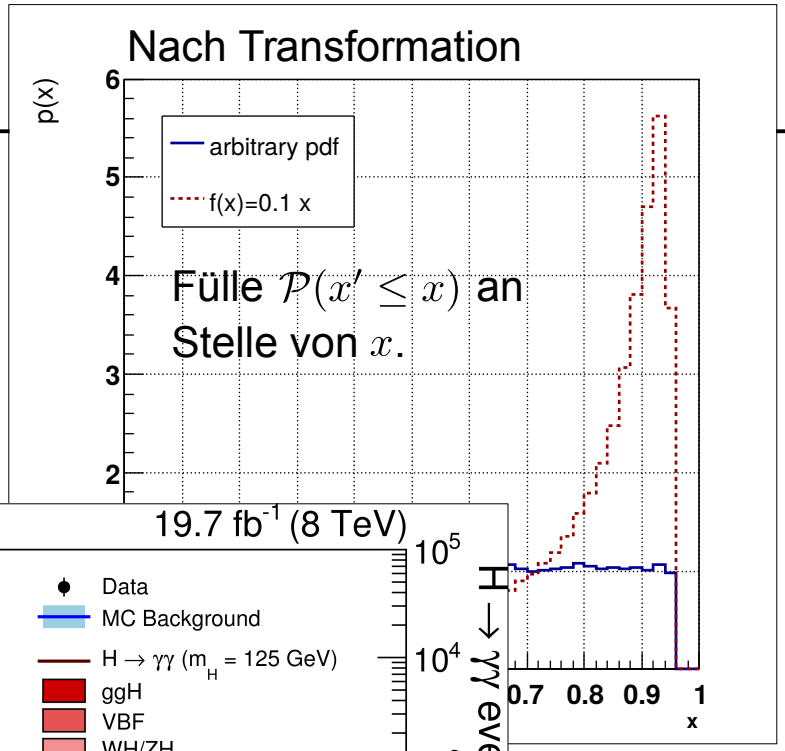




„beliebig“ verteilte Zufalls-variable.

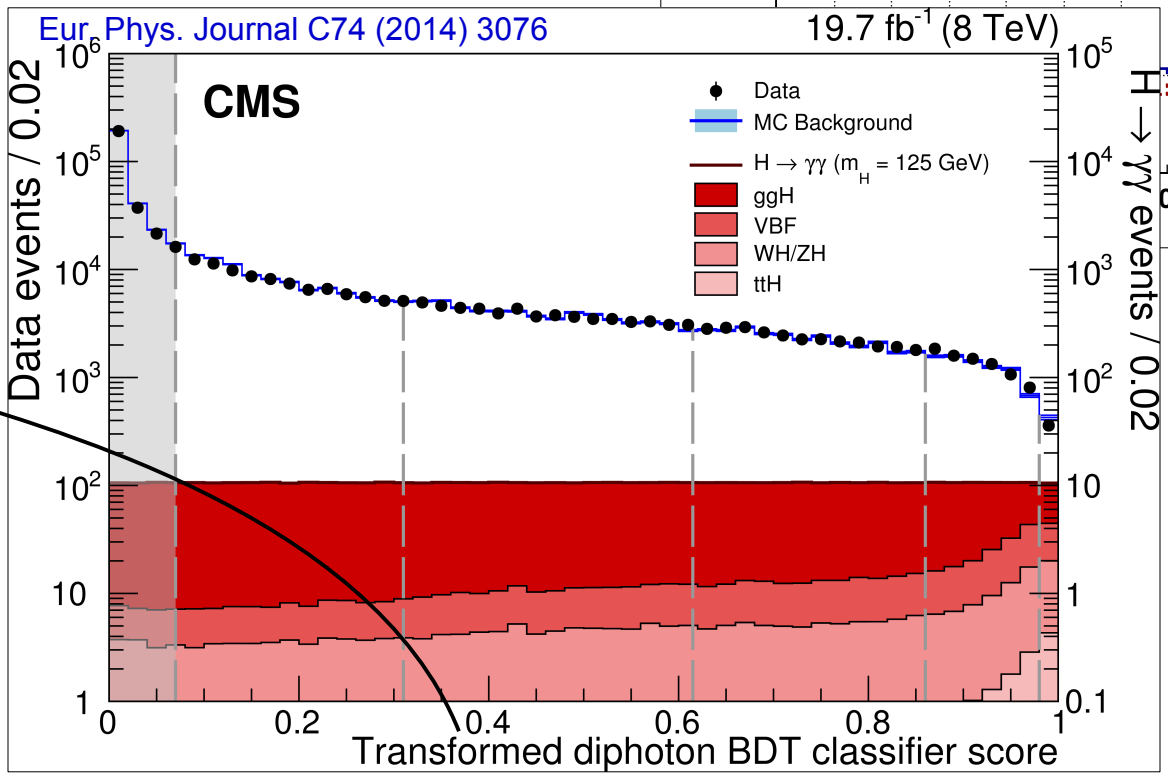
„Referenzverteilung“ (zeigt Wirken der Transformation).

(8) cf. VL-09 slide 31.

# Beispiel: $p(x)$ beliebig[8]




Beispiel aus der Physik:




Hier wurde (für Daten und MC) nicht der output $x$ des *classifiers* gefüllt, sondern $\mathcal{P}(x)$ für das erwartete Signal des Higgs Bosons.

(8) cf. VL-09 slide 31.

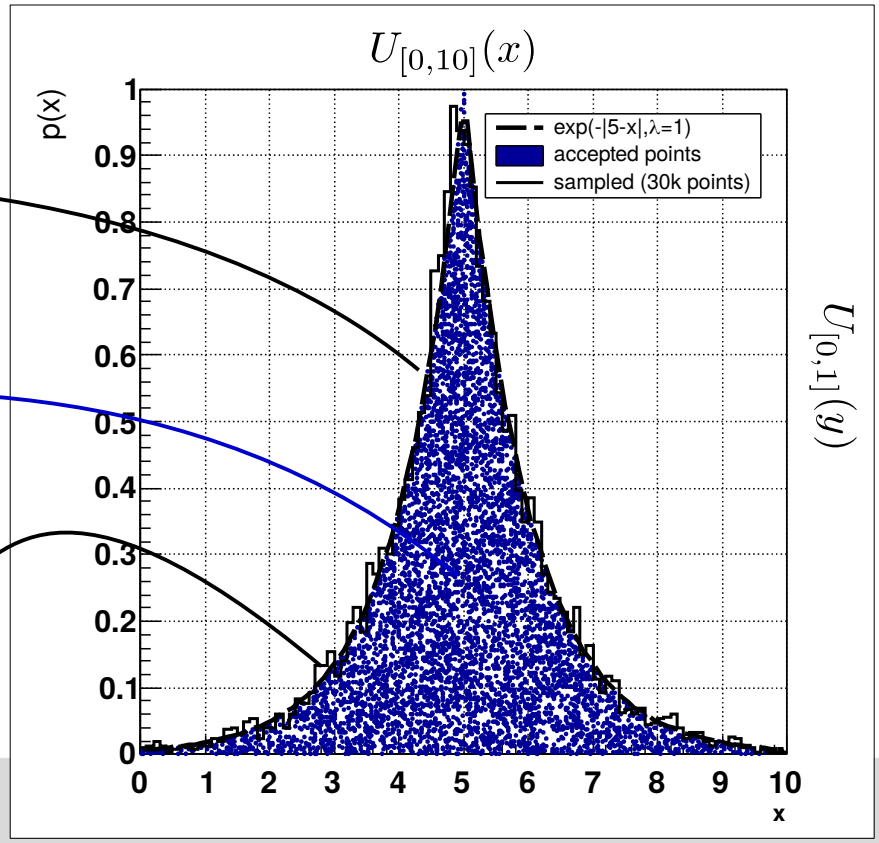
# Rejection sampling

- **Möglichkeit-2**: Verwerfungsmethode (= engl. rejection sampling)[9]
  - Zur Transformation in eine beliebige Wahrscheinlichkeitsdichte $p(x)$ erzeuge ein Paar gleichverteilter Zufallszahlen $(x, y)$.
  - Akzeptiere Ereignis wenn $y \leq p(x)$ (z.B. fülle Histogramm) und verwerfe das Ereignis sonst.
  - Beispiel Funktion: $p(x) = \exp(-|5 - x|)$

Ursprüngliche Wahrscheinlich-keitsdichte

Akzeptierte Inte-grationspunkte (30k stochastische Ereignisse).

Abgeleitetes Histogramm (30k Integrationspunkte)






(9) Zuerst vorgeschlagen von John v. Neumann (1951).

# Numerische Integration

- Stochastiches *sampling* läßt sich auch zur numerischen Integration nutzen:
  - Teile Ereignisse in *pass* und *fail*.
  - Das Integral ergibt sich aus: $\int p(x)\mathrm{d}x = 10 \cdot \frac{n(\mathrm{pass})}{n(\mathrm{pass})+n(\mathrm{fail})}$ (10)

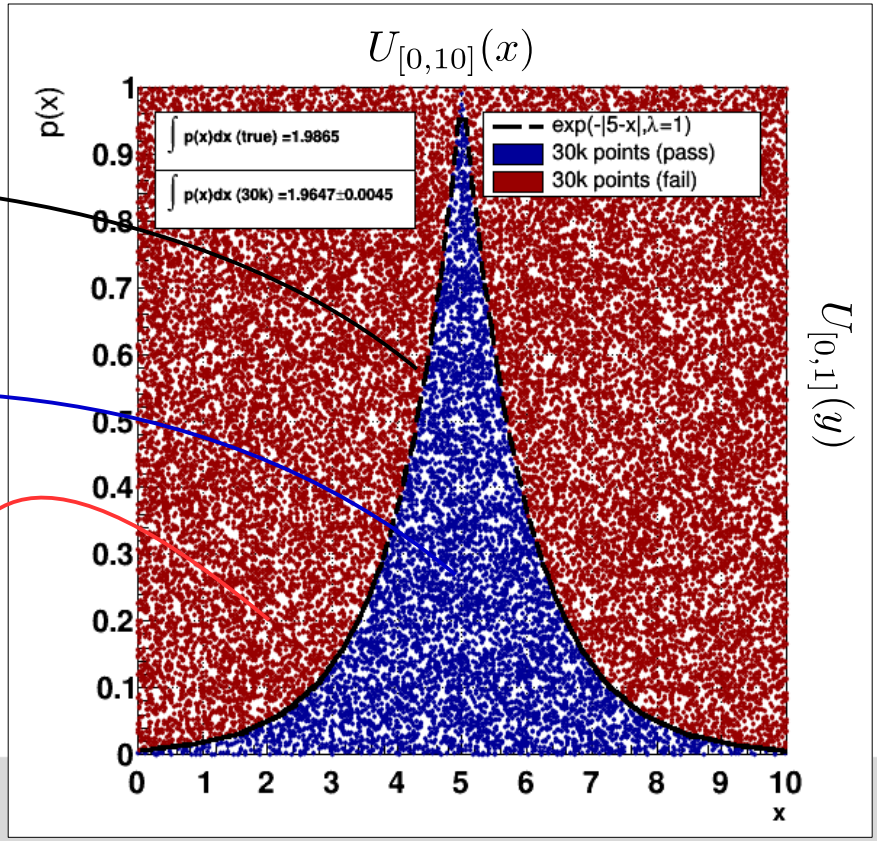
Einheitsfläche (blau+rot).

- Beispiel Funktion: $p(x) = \exp(-|5-x|)$

Ursprüngliche Wahrscheinlich-keitsdichte

Akzeptierte Inte-grationspunkte (30k stochastische Ereignisse).

Verworfene Integrationspunkte (30k stochastische Ereignisse)






(10) Zur Bestimmung der Unsicherheit auf diesen Wert cf. VL-09 slide 34.

# Bestimmung der Zahl $\pi$

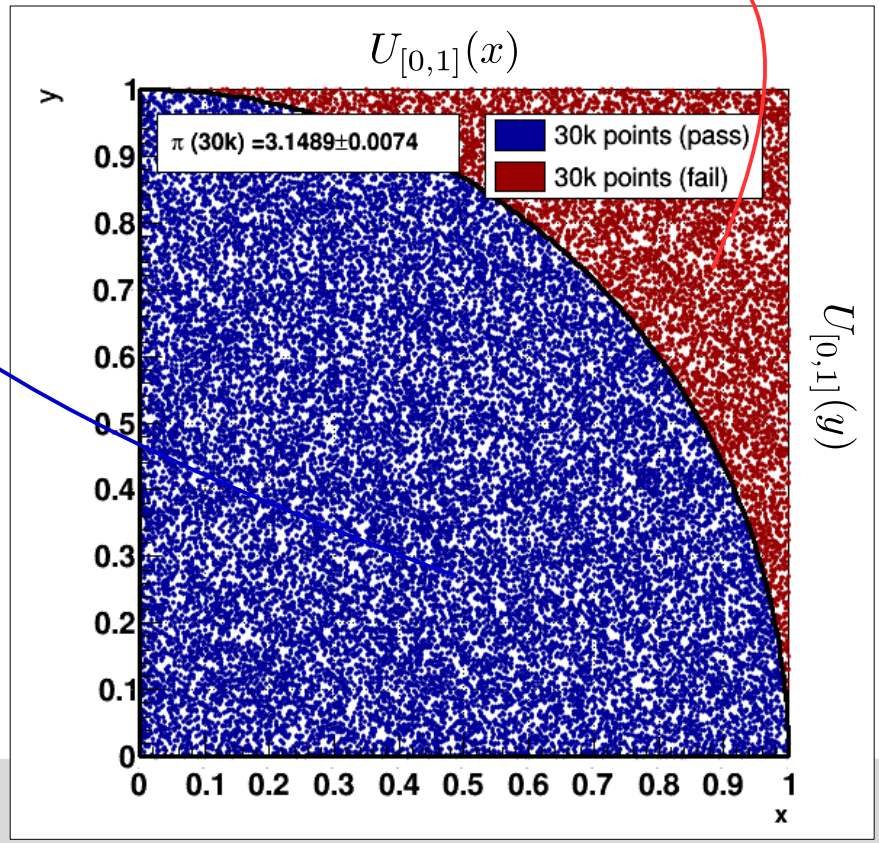
- Weiteres Beispiel eines einfachen Integrationsproblems (Buffonsches Nadelproblem):[11]

$$\hat{\pi} = 4 \cdot \frac{n_{\text{pass}}}{n_{\text{pass}} + n_{\text{fail}}}$$ (in diesem Beispiel für 30k stochastische Integrationspunkte)

[11] Georges-Louis Leclerce de Buffon (1707 – 1788)

Akzeptierte Integrationspunkte

Verworfene Integrationspunkte




Sie werden dieses Problem selbst als Übung behandeln können.

# Bestimmung der Zahl $\pi$

- Weiteres Beispiel eines einfachen Integrationsproblems (Buffonsches Nadelproblem):[11]

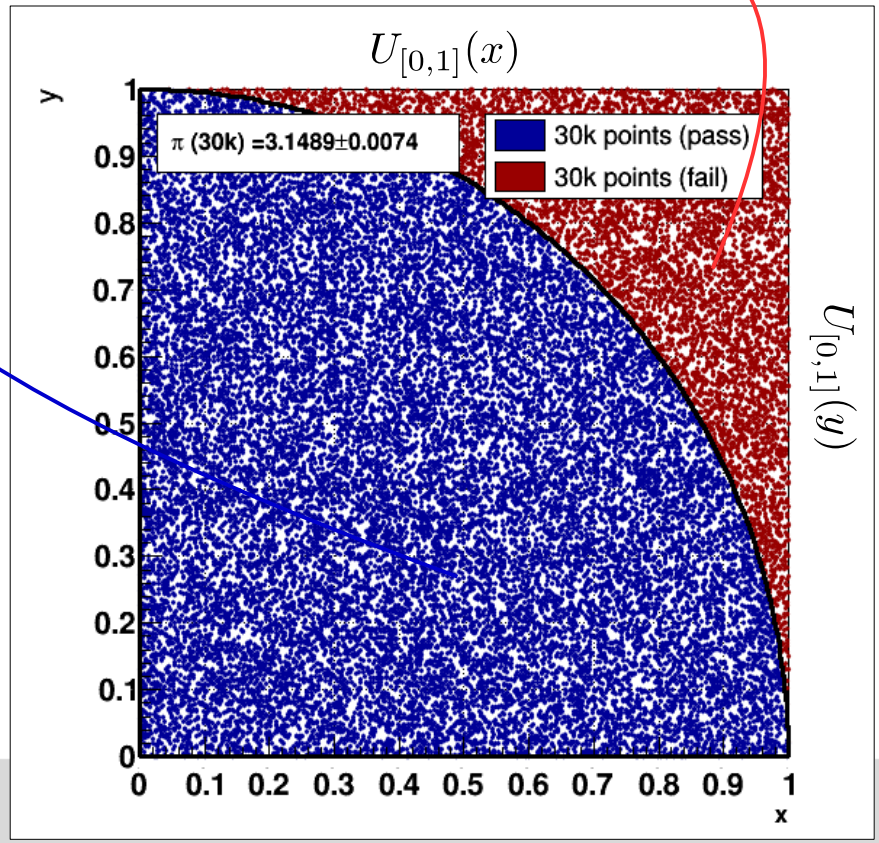
$$\hat{\pi} = 4 \cdot \frac{n_{\text{pass}}}{n_{\text{pass}} + n_{\text{fail}}}$$ (in diesem Beispiel für 30k stochastische Integrationspunkte)

[11] Gianluigi Buffon (1978)

Akzeptierte Integrationspunkte

Verworfene Integrationspunkte

Nicht zu verwechseln mit Gigi Buffon (Italienischer Nationaltorhüter)...

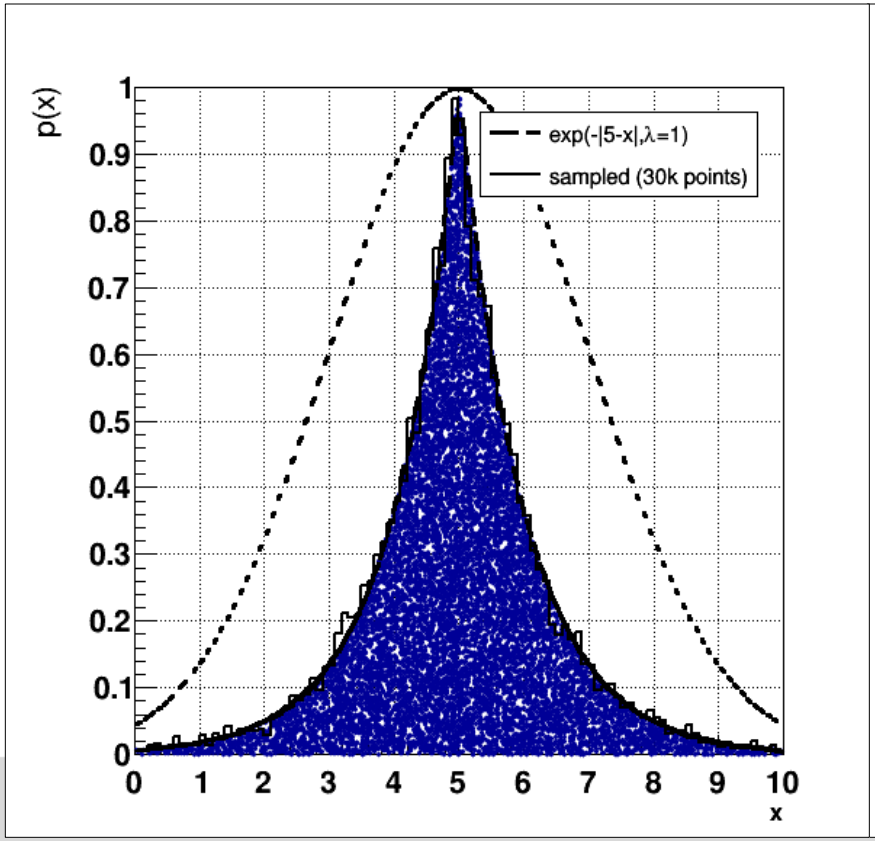


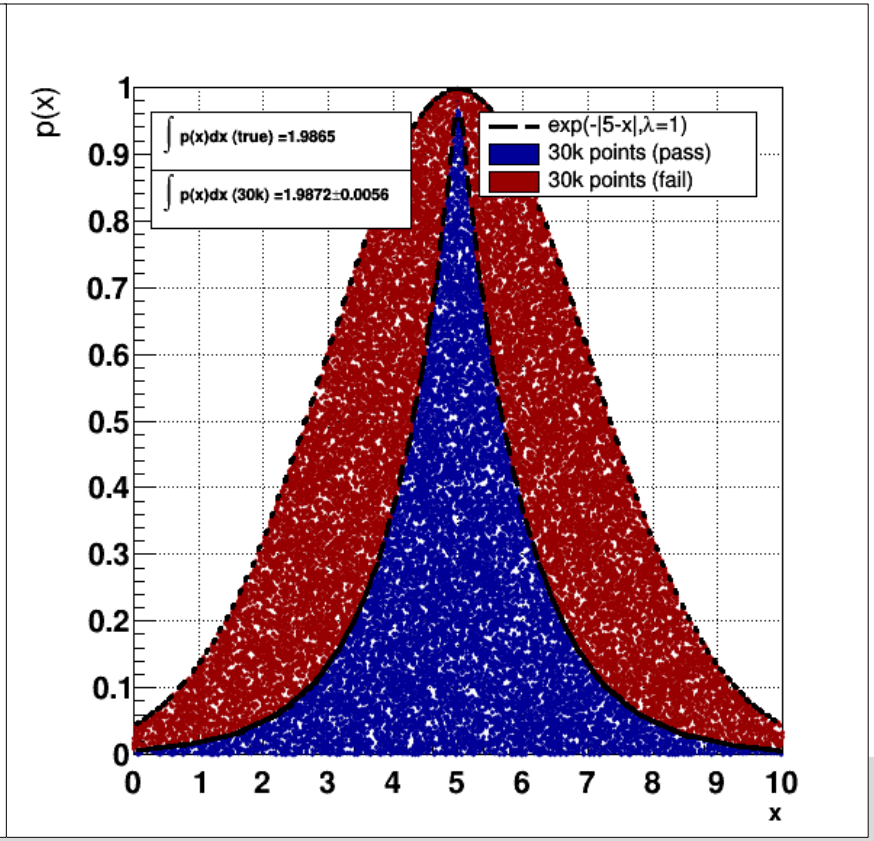
# Majoranten Methode

- Das Konvergenzverhalten der Methode zur wahren Wahrscheinlichkeitsdichte / dem wahren Integralwert läßt sich durch Reduktion der *fail* Ereignisse verbessern:

- Beispiel: $\mathcal{M}(x) = 5 \cdot \exp(-((x-5)/2)^2)$ als Einhüllende (=Majorante).

Wie zuvor basierend auf 30k Integrationspunkten



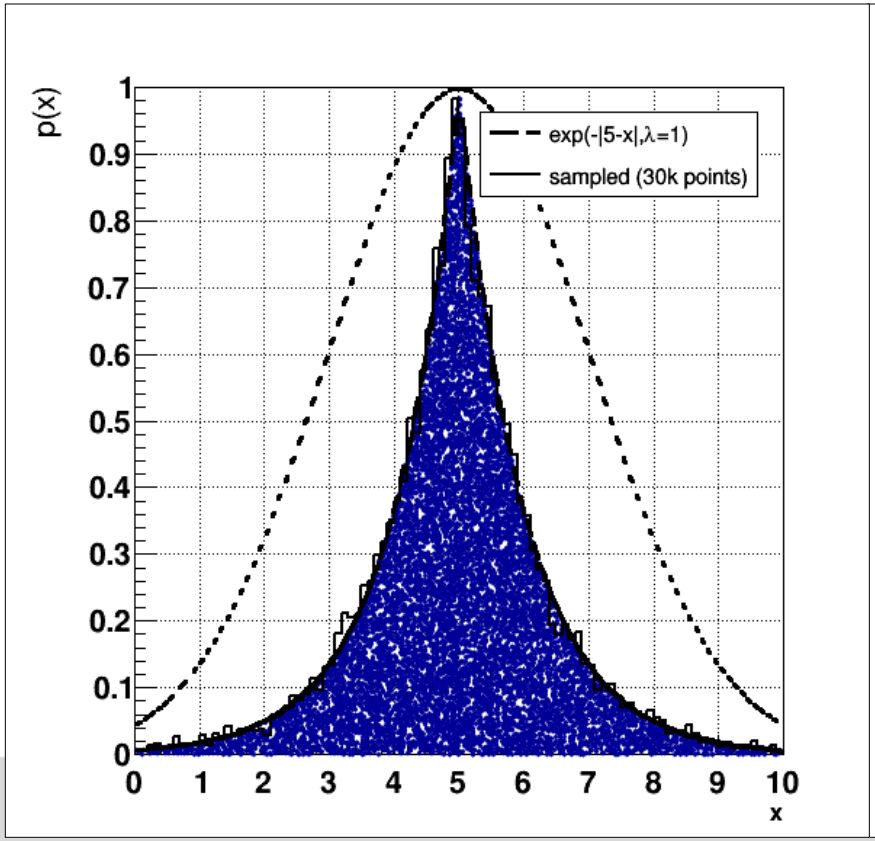



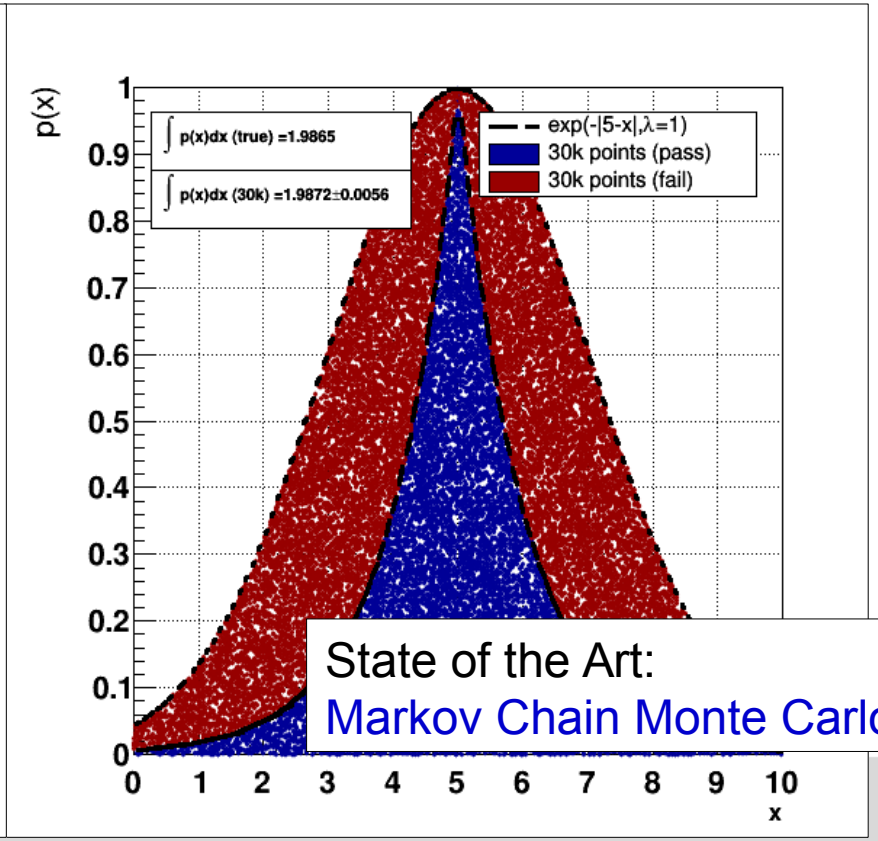
# Majoranten Methode

- Das Konvergenzverhalten der Methode zur wahren Wahrscheinlichkeitsdichte / dem wahren Integralwert läßt sich durch Reduktion der *fail* Ereignisse verbessern:

- Beispiel: $\mathcal{M}(x) = 5 \cdot \exp(-((x-5)/2)^2)$ als Einhüllende (=Majorante).

Wie zuvor basierend auf 30k Integrationspunkten




State of the Art:
Markov Chain Monte Carlo

# Bedeutung: Monte Carlo Integration

- Monte Carlo Methode bezieht Relevanz aus Einfachheit & Konvergenzverhalten:
- Man kann zeigen, daß die Varianz des Monte Carlo Schätzwertes für das Integral unabhängig von der Dimension des Integrals mit der Menge der Integrationspunkte $1/\sqrt{n}$ skaliert.[12]
- Konvergenz von Quadraturformeln skaliert mit $1/n^{2/d}$ ($n$ Anzahl der Stützstellen, $d$ Dimension des zu integrenden Phasenraums).
- Bessere Konvergenz von Quadraturformeln für $d < 4$. Ab $d = 5$ ist die Monte Carlo Integration Quadraturformeln überlegen.

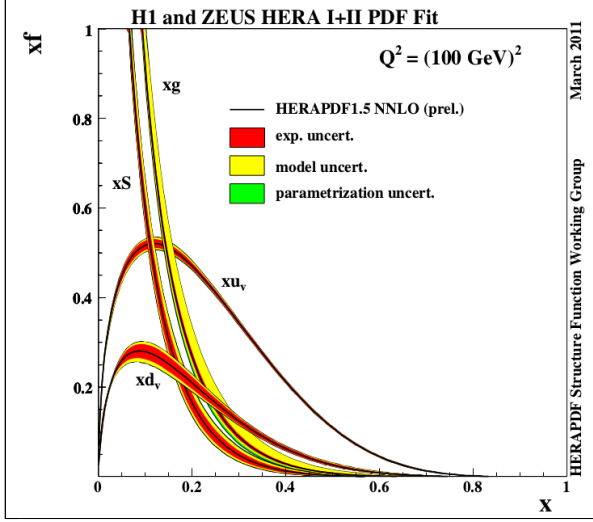
(12) cf. VL-11 slide 8 um Schätzfunktionen.

# Monte Carlo Methoden in der Teilchenphysik (13)

- Monte Carlo Methoden spielen eine zentrale Rolle bei der Simulation Teilchen-physikalischer Prozesse (z.B. Proton-Proton Kollisionen am LHC).

- Sie haben jetzt alles Rüstzeug in der Hand um eine Monte Carlo Simulation für die Beobachtung eines bestimmten Prozesses am LHC schreiben zu können:

Wahrscheinlichkeit ein bestimmtes Parton im Proton anzutreffen.

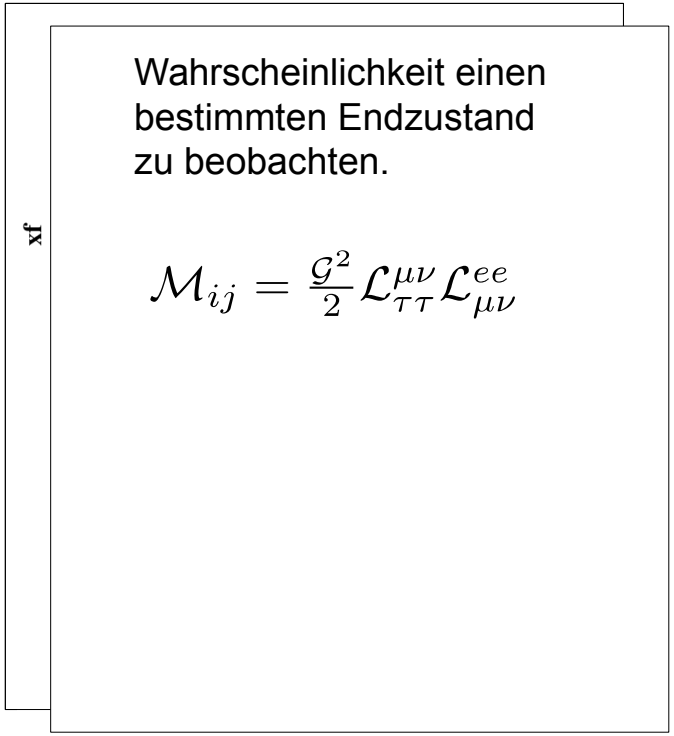



(13) cf. VL-09 slide 2.

# Monte Carlo Methoden in der Teilchenphysik (13)

- Monte Carlo Methoden spielen eine zentrale Rolle bei der Simulation Teilchen-physikalischer Prozesse (z.B. Proton-Proton Kollisionen am LHC).

- Sie haben jetzt alles Rüstzeug in der Hand um eine Monte Carlo Simulation für die Beobachtung eines bestimmten Prozesses am LHC schreiben zu können:

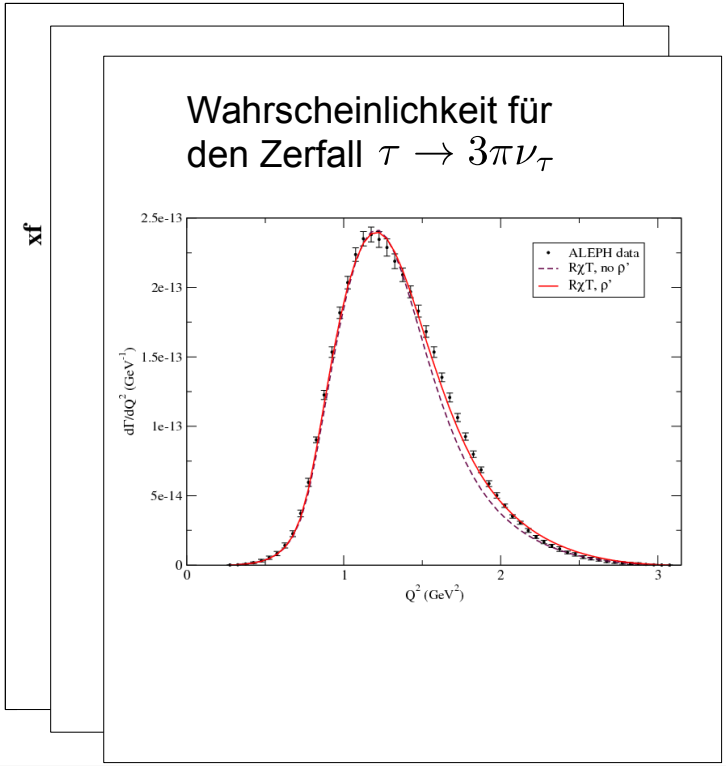



(13)
cf. VL-09 slide 2.

# Monte Carlo Methoden in der Teilchenphysik (13)

- Monte Carlo Methoden spielen eine zentrale Rolle bei der Simulation Teilchen-physikalischer Prozesse (z.B. Proton-Proton Kollisionen am LHC).

- Sie haben jetzt alles Rüstzeug in der Hand um eine Monte Carlo Simulation für die Beobachtung eines bestimmten Prozesses am LHC schreiben zu können:

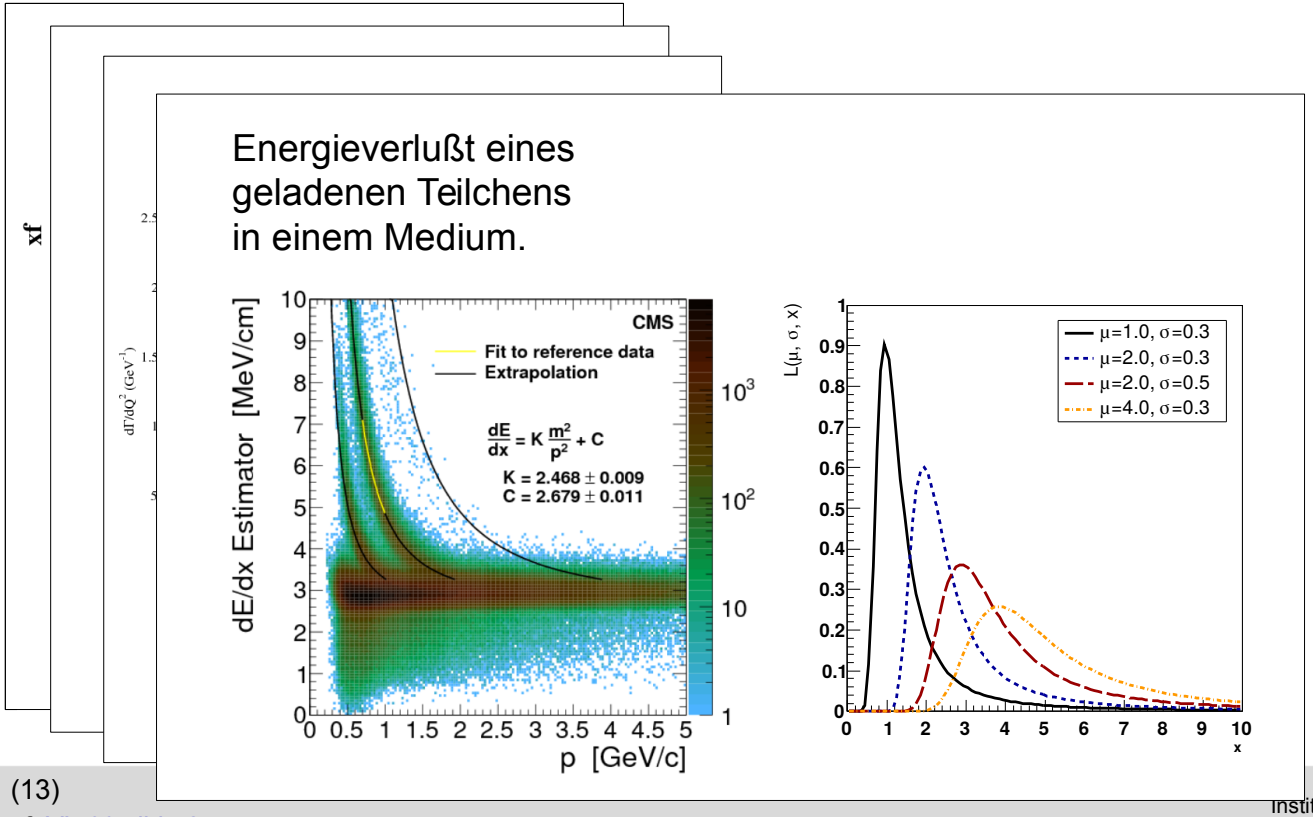



(13)
cf. VL-09 slide 2.

# Monte Carlo Methoden in der Teilchenphysik (13)

- Monte Carlo Methoden spielen eine zentrale Rolle bei der Simulation Teilchen-physikalischer Prozesse (z.B. Proton-Proton Kollisionen am LHC).

- Sie haben jetzt alles Rüstzeug in der Hand um eine Monte Carlo Simulation für die Beobachtung eines bestimmten Prozesses am LHC schreiben zu können:




(13)
cf. VL-09 slide 2.

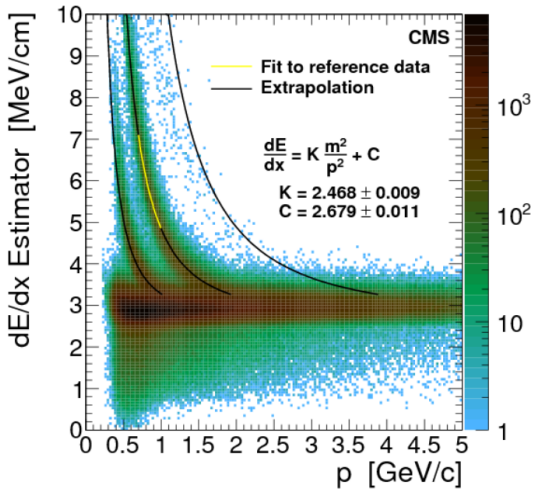
# Monte Carlo Methoden in der Teilchenphysik (13)

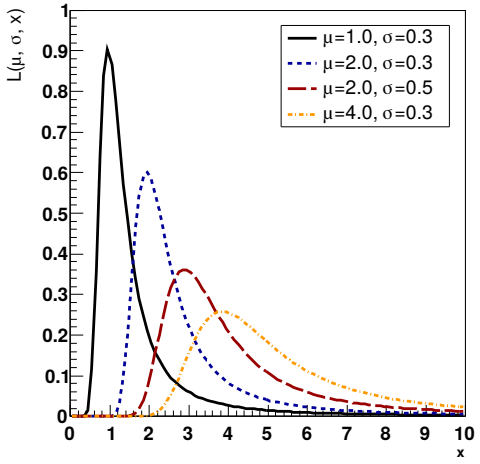
- Monte Carlo Methoden spielen eine zentrale Rolle bei der Simulation Teilchen-physikalischer Prozesse (z.B. Proton-Proton Kollisionen am LHC).

- Sie haben jetzt alles Rüstzeug in der Hand um eine Monte Carlo Simulation für die Beobachtung eines bestimmten Prozesses am LHC schreiben zu können:

Alles stochastisch unabhängige Prozesse!

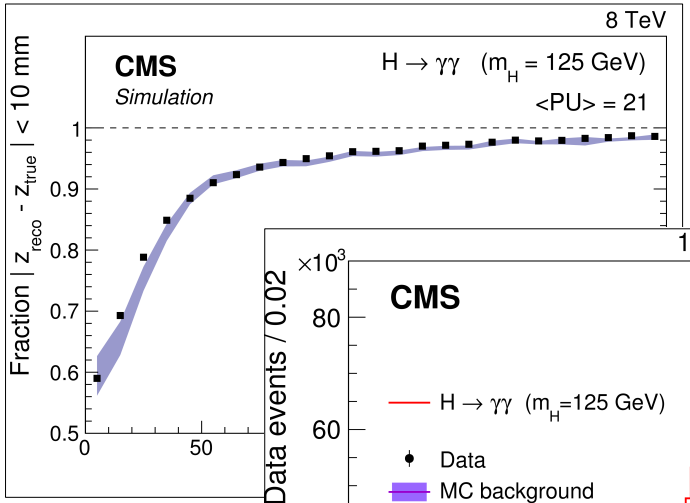
Energieverlußt eines geladenen Teilchens in einem Medium.

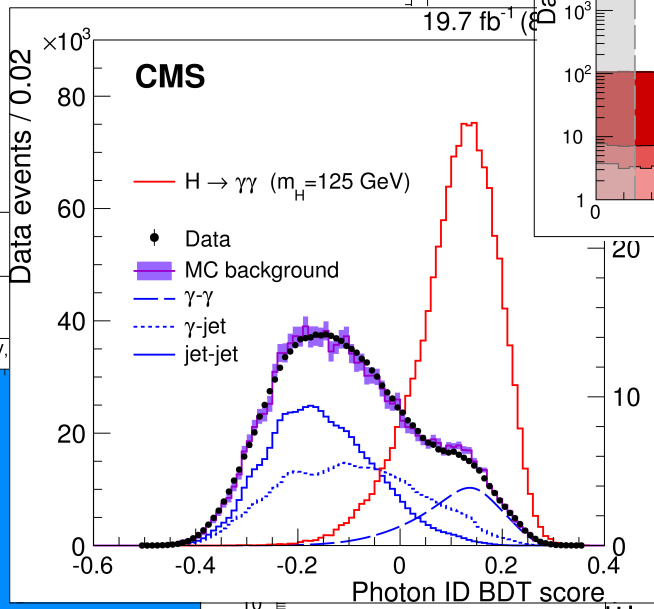


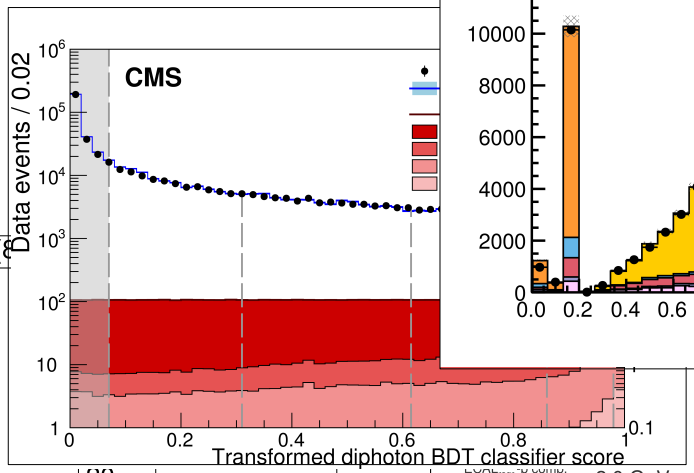



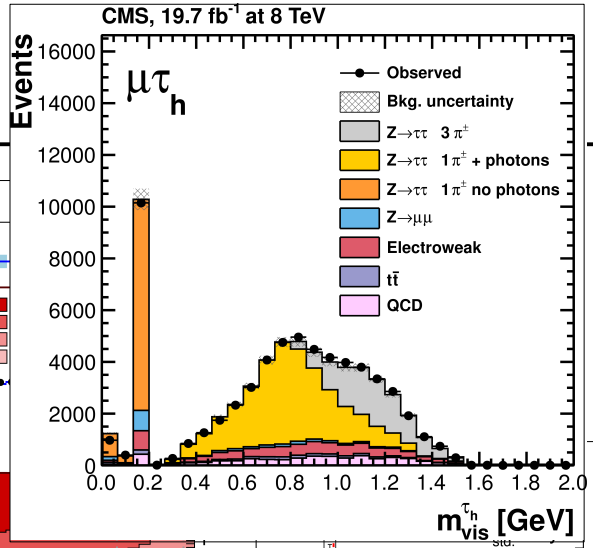

(13)
cf. VL-09 slide 2.

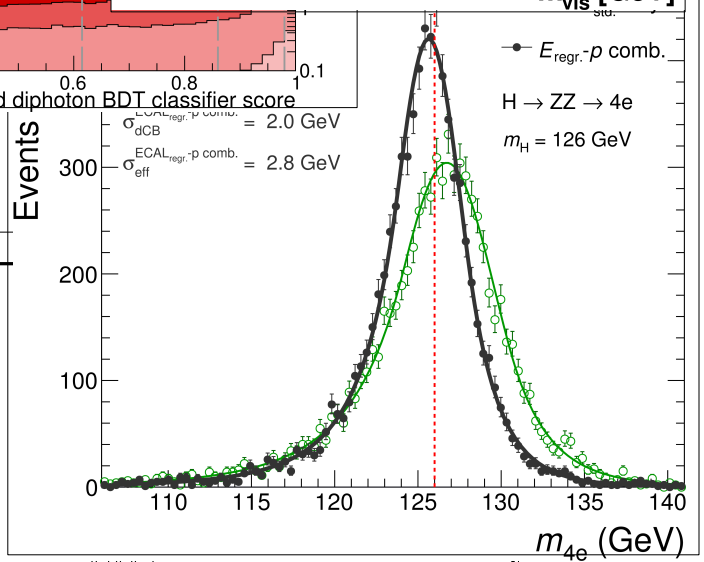
# Beispiele von Data-MC Vergleichen

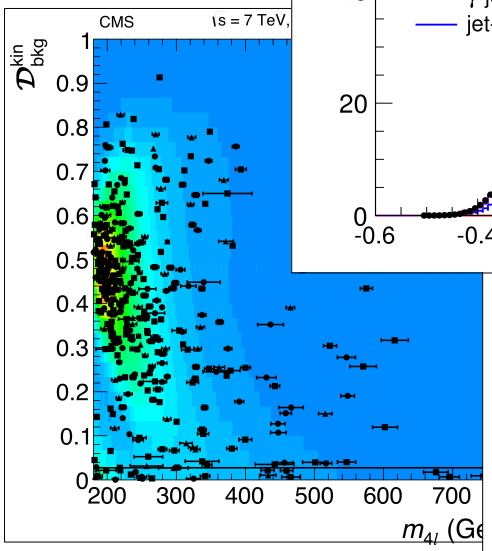


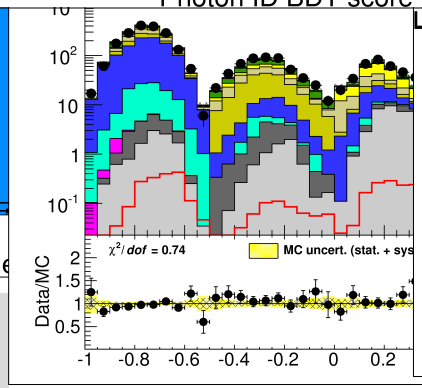



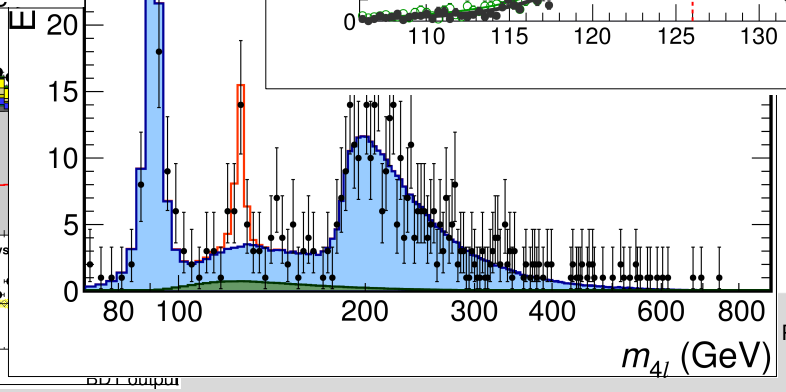



Kapiel 3.5:

# Monte Carlo Methoden

- Generatoren von (Pseudo-)Zufallszahlen (→ wichtige Eigenschaften, LCG).
- Transformation gleichverteilter Zufallszahlen auf beliebig verteilte Zufallszahlen (→ analytisch, rejection sampling, verbesserte Konvergenz).
- Monte Carlo als Integrationsmethode.
- Bedeutung in der (Teilchen-)Physik.