

## Aufgabe 11: *Simulationskette für Neutrinodetektor*

### a) Signal MC

Mit Hilfe der Transformationsmethode werden  $10^5$  Signalereignisse des Neutrinoflusses

$$\Phi = \Phi_0 \left( \frac{E}{\text{TeV}} \right)^{-\gamma}, \quad (1)$$

mit  $\gamma = 2.7$ , erzeugt. Für die Transformationsmethode wird der Fluss zunächst von 1 TeV bis  $E$  integriert und dann invertiert. Das Ziehen von gleichverteilten Zufallszahlen  $u$  im Intervall  $[0, \frac{\Phi_0 1 \text{ TeV}}{\gamma-1}]$  und Einsetzen in

$$\frac{E}{1 \text{ TeV}} = \left( 1 + \frac{u(1-\gamma)}{\Phi_0 1 \text{ TeV}} \right)^{\frac{1}{1-\gamma}} \quad (2)$$

liefert die gewünschten Signale. Diese werden in einem DataFrame mit dem Key Energy abgespeichert.

# Aufgabe 11a

Freitag, 10. November 2017 16:38

$$11. a) \quad \Phi = \Phi_0 \left( \frac{E}{\text{TeV}} \right)^{-\gamma}, \quad \gamma = 2,7$$

$$u = g(E) = \int_{1 \text{ TeV}}^E \Phi_0 \left( \frac{E'}{\text{TeV}} \right)^{-\gamma} dE' \quad \left( \begin{array}{l} \tilde{E} = \frac{E'}{\text{TeV}} \\ \frac{d\tilde{E}}{dE'} = \frac{1}{\text{TeV}} \end{array} \right)$$

$$g(E) = \int_1^{\frac{E}{\text{TeV}}} \Phi_0 \cdot 1 \text{ TeV} \tilde{E}^{-\gamma} d\tilde{E}$$

$$= \frac{1}{-\gamma+1} \Phi_0 \cdot 1 \text{ TeV} \left( \left( \frac{E}{\text{TeV}} \right)^{-\gamma+1} - 1 \right) = u$$

$$\Rightarrow u + \frac{\Phi_0 \cdot 1 \text{ TeV}}{1-\gamma} = \frac{\Phi_0 \cdot 1 \text{ TeV}}{1-\gamma} \left( \frac{E}{\text{TeV}} \right)^{-\gamma+1}$$

$$\Leftrightarrow \left( \frac{E}{\text{TeV}} \right)^{-\gamma+1} = \frac{u(1-\gamma)}{\Phi_0 \cdot 1 \text{ TeV}} + 1$$

$$\Leftrightarrow \frac{E}{\text{TeV}} = \left[ 1 + \frac{u(1-\gamma)}{\Phi_0 \cdot 1 \text{ TeV}} \right]^{\frac{1}{1-\gamma}}, \quad [\Phi_0] = \text{TeV}^3$$

$$u(1 \text{ TeV}) = \frac{\Phi_0 \cdot 1 \text{ TeV}}{1-\gamma} (1-1) = 0$$

$$u(\infty) = \frac{\Phi_0 \cdot 1 \text{ TeV}}{1-\gamma} (0-1) = \frac{\Phi_0 \cdot 1 \text{ TeV}}{\gamma-1}$$

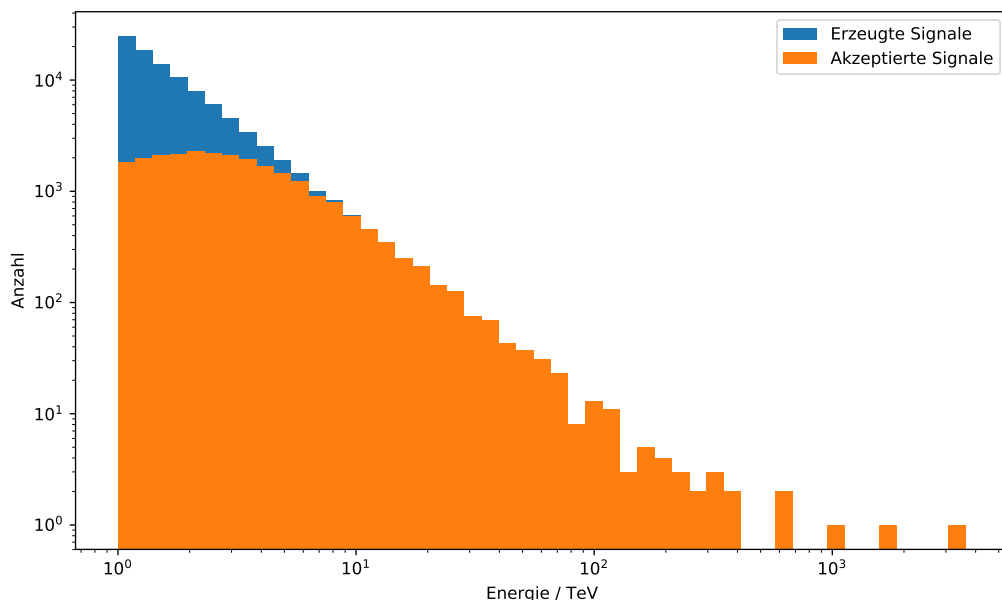
$$\Rightarrow u \in \left[ 0, \frac{\Phi_0 \cdot 1 \text{ TeV}}{\gamma-1} \right]$$

## b) Akzeptanz

Da die Detektionswahrscheinlichkeit energieabhängig ist, müssen die in Aufgabenteil a) erzeugten Signale modifiziert werden. Dazu wird für jede zuvor bestimmte Energie die Detektionswahrscheinlichkeit nach der Gleichung

$$P(E) = (1 - e^{-E/2})^3 \quad (3)$$

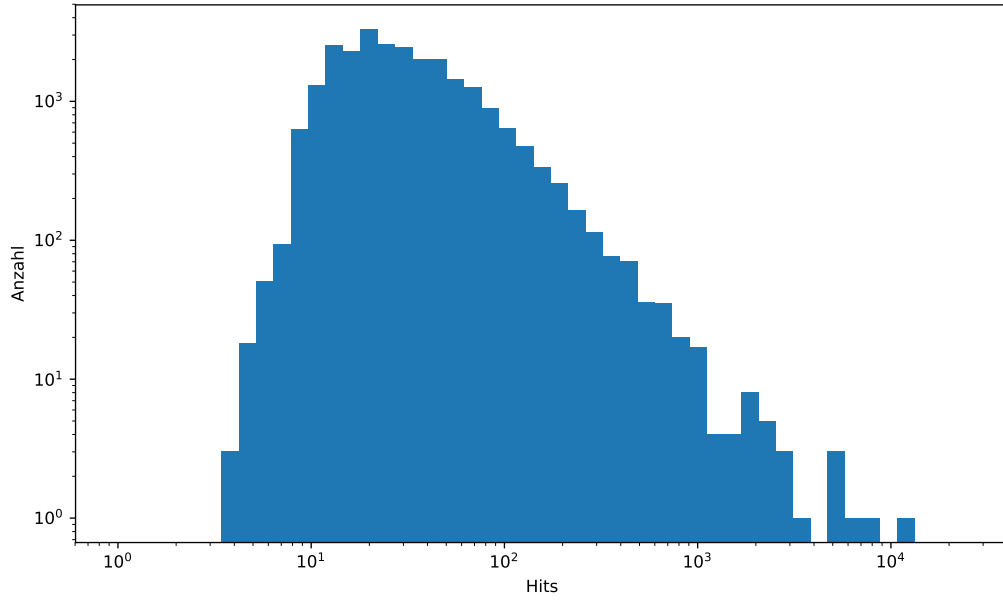
bestimmt. Dann werden  $10^5$  gleichverteilte Zufallszahlen  $r$  zwischen 0 und 1 mit `numpy.random.uniform()` gezogen und es wird eine Maske erstellt, indem elementweise verglichen wird, ob  $r < P(E)$  gilt. Diese Maske wird in dem DataFrame unter dem Key `AcceptanceMask` gespeichert. In Abbildung 1 sind die Histogramme der erzeugten und der akzeptierten Signale überlagert doppelt-logarithmisch dargestellt. Der sichtbare blaue Bereich entspricht dem Anteil des erzeugten Signals, der verworfen wird.



**Abbildung 1:** Histogramm der erzeugten und akzeptierten Signale. Aufgetragen ist die Energie gegen die Häufigkeit.

## c) Energiemessung

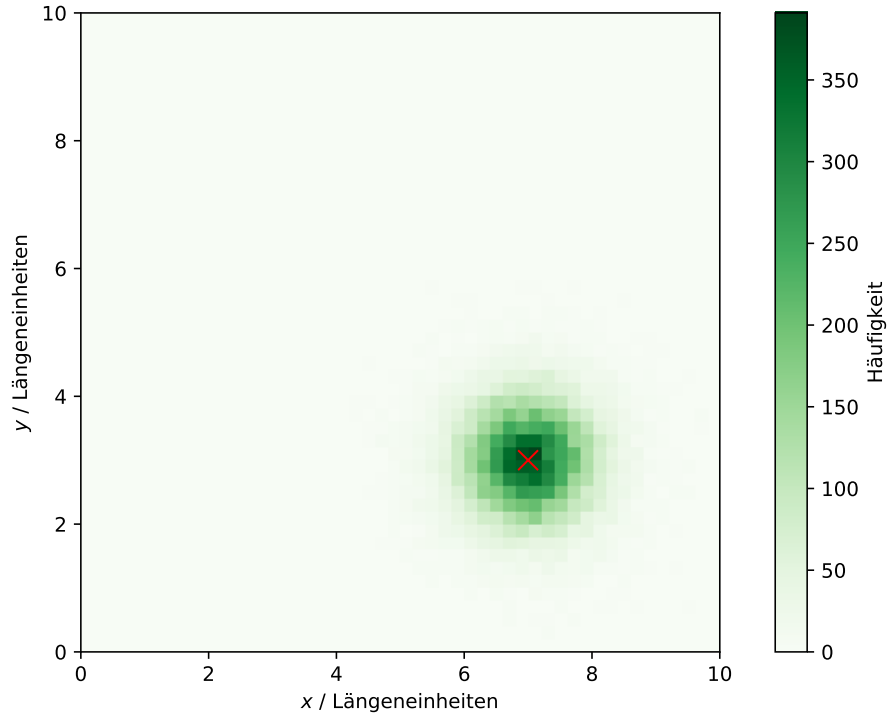
In diesem Aufgabenteil wird die Anzahl der Hits, also die Anzahl der Photomultiplier, die angesprochen haben, für die akzeptierten Energien simuliert. Dazu werden mit der Polarmethode eine normalverteilte Zufallszahlen mit dem Mittelwert  $10E$  und der Standardabweichung  $2E$  generiert. Da die Anzahl der Hits eine ganze, nicht negative Zahl sein muss, werden alle negativen Zufallszahlen verworfen und die Anzahl wird auf eine ganze Zahl gerundet. Die Anzahl Hits wird in dem DataFrame unter dem Key `NumberOfHits` gespeichert. Des Weiteren wird ein Histogramm für die Hits erstellt, das in Abbildung 2 zu sehen ist.



**Abbildung 2:** Histogramm für die Hits in Abhängigkeit der akzeptierten Energien.

#### d) Ortsmessung

Es wird ein quadratischer Flächendetektor mit der Kantenlänge 10 Längeneinheiten simuliert, bei dem im Punkt (7,3) das Signal auftritt. Die energieabhängige Ortsauflösung wird durch eine Normalverteilung in x-Richtung und eine Normalverteilung in y-Richtung mit der Standardabweichung  $\sigma = (\log_1 0(N + 1))^{-1}$  simuliert. Es gilt  $\mu_x = 7$  und  $\mu_y = 3$ . Gezogene Ergebnisse, die außerhalb der Fläche des Detektors liegen werden verworfen und neu gezogen. Die Koordinaten der Orte werden mit den Keys x und y im DataFrame gespeichert. Aus den erhaltenen Orten wird ein zweidimensionales Histogramm erstellt, das in Abbildung 3 dargestellt ist.



**Abbildung 3:** Zweidimensionales Histogramm für die Orte, an denen das Signal detektiert wird.

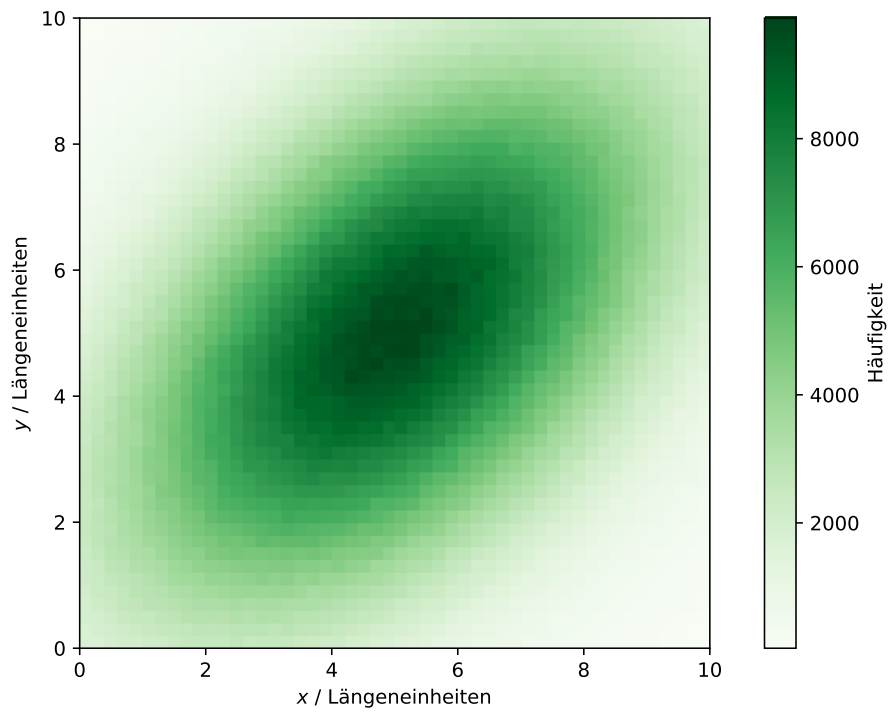
#### e) Untergrund MC

Zuletzt werden noch  $10^7$  Untergrundereignisse simuliert. Der Zehner-Logarithmus der Anzahl der Hits folgt dabei einer Normalverteilung mit  $\mu = 2$  und  $\sigma = 1$ , die x- und y-Koordinate sind um den Mittelpunkt (5,5) normalverteilt und zwischen x und y besteht eine Korrelation von  $\rho = 0.5$ :

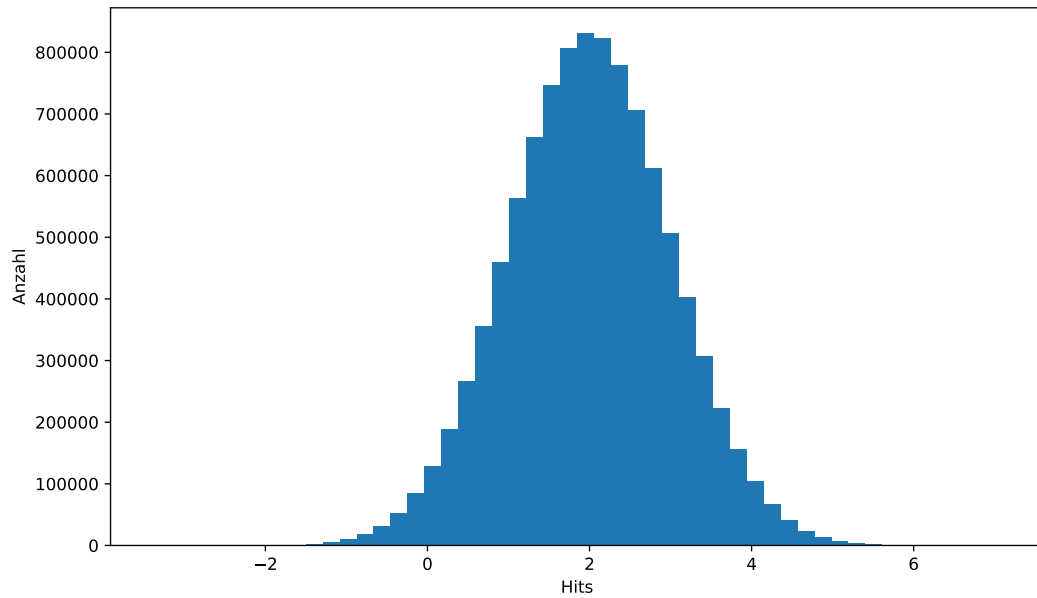
$$x = \sqrt{1 - \rho^2} \cdot \sigma \cdot x^* + \rho \cdot \sigma y^* + \mu \quad (4)$$

$$y = \sigma \cdot y^* + \mu. \quad (5)$$

Abbildung 4 zeigt ein zweidimensionales Histogramm der Orte der Untergrundereignisse und der Logarithmus der Anzahl der Hits ist in einem eindimensionalen Histogramm in Abbildung 5 dargestellt.



**Abbildung 4:** Zweidimensionales Histogramm der Orte der Untergrundereignisse.



**Abbildung 5:** Histogramm des Logarithmus der Anzahl der Hits des Untergrunds.