Datenanalyse
WS 2001/00

Author(s):
Pruys, H.

Publication Date:
2001

Permanent Link:
https://doi.org/10.3929/ethz-a-004444215

4 Maxwell-Verteilung

Die Maxwellverteilung ist eine der bekanntesten Verteilungen einer physikalischen Größe. Als solche eignet sie sich gut zur Übung der Analyse 'gemessener' Daten und des Vergleichs dieser Daten mit der Theorie. Im einzelnen wollen wir in diesem Kapitel ein Matlab Programm entwerfen, das folgendes leistet:

1. Berechnung des Mittelwertes und seines Fehlers für gegebene Messdaten (Geschwindigkeiten von Xe-Atomen).
2. Histogrammieren der Messdaten.
3. Wiederum Berechnung des Mittelwertes ..., diesmal aus dem Histogramm.
4. Darstellung des Histograms mit Fehlerbalken.
5. Berechnung der Temperatur aus dem Geschwindigkeitsmittelwert
6. Darstellung der Maxwellverteilung zu dieser Temperatur in derselben Figur wie das Histogram.

Damit wir überhaupt das Programm schreiben können, müssen wir uns erstmal ein wenig mit der Theorie befassen. Die Maxwell-Verteilung hat folgende Form:

$$f(v) = \frac{dN(v)}{dv} = 4\pi v^2 (\frac{m}{2\pi kT})^{\frac{3}{2}} e^{\frac{-mv^2}{2kT}}$$

mit
v: Geschwindigkeit in m/s
m: Masse. z.B. Xe: m = 131.29 $\times$1.6605 $\times$ 10^{-27} kg
k: Boltzmannkonstante = 1.3807 $\times 10^{-23}$ Ws/K
T: Temperatur in K

Man kann dann zeigen, daß folgendes gilt:

$$\bar{v} = \sqrt{\frac{8kT}{\pi m}}, \qquad v_{max} = \sqrt{\frac{2kT}{m}}, \qquad \bar{v^2} = \frac{3kT}{m}, \qquad f(v_{max}) = \frac{4}{e}\sqrt{\frac{m}{2\pi kT}}$$

Hieraus folgt direkt der für Schritt 1 benötigte Zusammenhang zwischen der Temperatur und der durchschnittlichen Geschwindigkeit eines idealen Gases:

$$T = \frac{\pi m \bar{v}^2}{8k} \rightarrow \sigma_T = \frac{\pi m \bar{v}}{4k} \sigma_{\bar{v}}$$

Wir können uns jetzt also an die genauere Ausarbeitung bzw. Skizzierung des Programmes machen (siehe auch Programm 1.2 im Vorlesungsskript). Bei jedem Programmteil ist angegeben, mit welchen Befehlen man arbeiten soll:

- Definieren der Maxwell-Verteilung als Funktion
  Sinnvollerweise sollte man, wenn man mit der Maxwell-Verteilung arbeitet, diese auch als Matlab-Funktion zur Verfügung haben. Daher schreibe man eine Datei mit den Name Maxwell.m, in der eine Funktion namens Maxwell definiert wird, die für gegebene Geschwindigkeit und Temperatur $f(v)$ zurückgibt. Die erste Zeile unserer Funktion sollte also lauten: `function y = Maxwell(v,T)` Die Konstanten die man benoetigt, sollten innerhalb der Funktion als Variablen definiert werden (z.B. `k=1.3807e-23;`). Die letzte Zeile Der Funktion ist dann eine direkte Übertragung der Maxwellformel nach Matlab. Die Funktion sollte auch dann eine korrekte Antwort liefern, wenn `v` ein Vektor ist (also komponentenweise Operationen an einigen Stellen). Die einzige Matlab Funktion, die benötigt wird, ist `exp`.

- Einlesen der Daten aus maxwell5.dat
  Dies ist der Anfang des eigentlichen Programmes. Also macht man erstmal eine neue Datei mit dem Namen maxwell_prog.m auf. Und schreibe alles weitere dort hinein. Man benutze den Befehl `load` und speichere die Daten in einem Vektor `v`. Die Anzahl der Datenpunkte wird im nächsten Schritt noch benötigt und sollte daher als Variable `n` abgespeichert werden (Befehl: `length`).

- Berechnung des Mittelwertes und seines Fehlers.
  Man verwende die Befehle `mean` und `std`. Beachte, daß `std` die Standardabweichung $\sqrt{\frac{1}{n-1}\sum_1^n (v_i - \bar{v})^2}$ zurückgibt, welches NICHT der Fehler des Mittelwertes ist! Beide Werte sollten in sinnvoll benannten Variablen abgelegt werden.

- Ausgabe der Werte
  Hierzu verwende man möglichst `fprintf`.

- Füllen des Histograms
  Zuerst sollte man einen Geschwindigkeitsbereich und eine Binbreite definieren, die zu den Daten passen (ausprobieren!). Dafür verwende man eine Variable `x`, die die Geschwindigkeitswerte der Mittelpunkte der Bins enthält. Anschließend benutzt man den Befehl `hist`, um die Daten aus `maxwell5.dat` in das Histogramm zu füllen. Das Histogramm speichere man als Vektor `y` ab.

- Darstellen des Histogramms
  Hierzu verwende man die Befehle `subplot` (um das Histogram in der linken Hälfte einer Figur darzustellen), `hold off` und `bar` (um das Histogramm zu zeichnen).

- Berechnung des Mittelwertes und seines Fehlers aus dem Histogramm
  Der Befehl hierzu heißt `centroid`. Er liefert beide benötigten Werte. Die Zahlen sollten wiederum ausgegeben werden (Warum weichen sie von den direkt erhaltenen Zahlen ab?)

- Nachmals das Histogramm plotten, diesmal mit Fehlerbalken
  Hierzu soll der Befehl `plot_data` verwendet werden. Das Argumentformat ist etwas tricky: Es wird eine MATRIX erwartet! Die Anzahl der Einträge in jedem Bin ist natürlich Poisson-verteilt. Was folgt daraus für den Fehler? Der Plot soll diesmal die rechte Seite der Figur einnehmen (`subplot`).

- Berechnung der Temperatur (in Kelvin) und des Fehlers der Temperatur aus dem Mittelwert von `v`
  Auch hier macht man sich das Leben leichter, wenn man die Masse der Atome und die Bolzmannkonstante jeweils einer Variablen zuweist. So wird auch die Lesbarkeit der eigentlichen Formel im Programm erheblich erhöht. Die erhaltenen Werte sollen ausgegeben und in den Variablen `T` und `sigmaT` abgespeichert werden.

- Maxwell-Verteilung zu der errechneten Temperatur darstellen
  Die Darstellung soll auch in der rechten Hälfte der Figur landen, also am einfachsten vorher `hold on` setzen. Da es sich um eine Theoriefunktion handelt, will man eine glatte Kurve mit vielen (>200) Stützpunkten haben. Daher mußs man zuerst den Vektor `x` umdefinieren, bevor man mit dem Befehl `plot` die Kurve zeichnet. Hier wird auch endlich die funktion `Maxwell` verwendet, um die `y`-Werte auszurechnen. Eine kleine Hürde gilt es noch zu überwinden: Damit

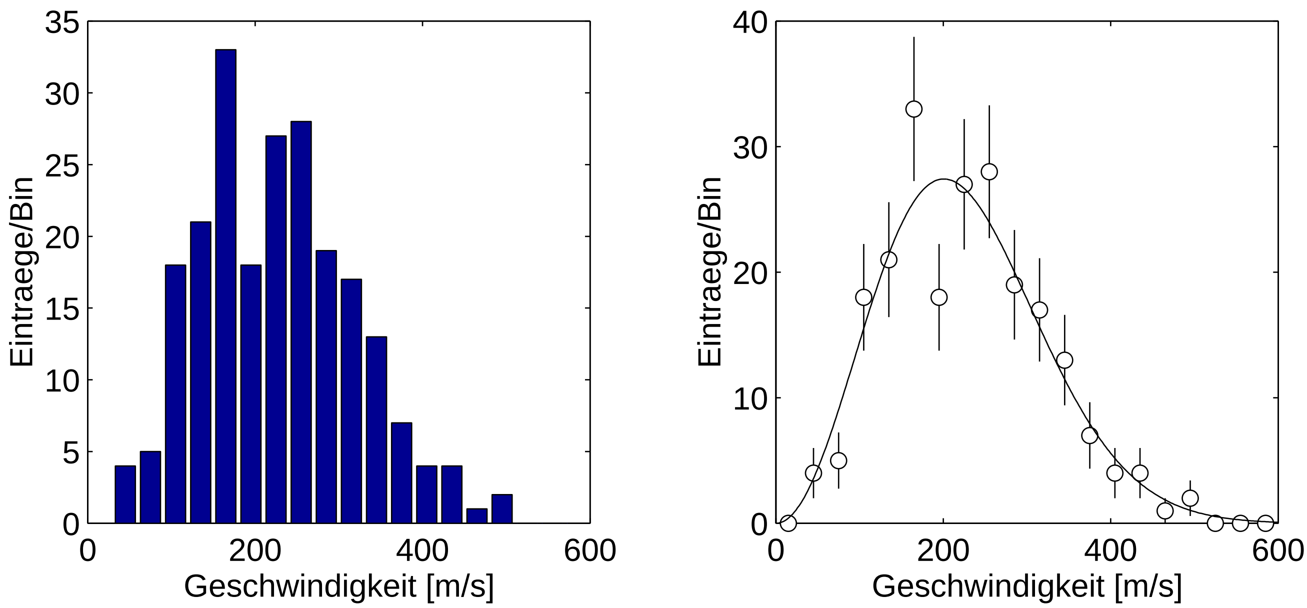

Abbildung 1: So oder ähnlich soll das Resultat aussehen

das Histogramm und die Theoriekurve vergleichbar sind, muß man die Theoriefunktion auf die Anzahl der Einträge im Histogramm normieren. Wie man sich leicht klarmachen kann, ist der Normierungsfaktor gleich der Binbreite im Histogramm multipliziert mit der Anzahl Einträge im Histogramm.

Ok, das wars! In einer der nächsten Kapitel werden wir dann etwas verfeinerte Methoden kennenlernen und programmieren, um gemessene Daten mit einer bekannten Theoriefunktion zu vergleichen.