Versuch 223 - Brownsche Bewegung

Felix Fleischle

22.11.2021

Einleitung

Abbildung 1: Bild des Versuchsaufbaus aus der Praktikumsanleitung

Bei diesem Versuch ist es das Ziel, die Boltzmann-Konstante und die Diffusionskonstante aus der mittleren quadratischen Verschiebung von kugelförmigen Partikeln suspendiert in Wasser zu bestimmen.
Bei der Brownschen Bewegung handelt es sich um die Zickzackartige Bewegung von kugelförmigen Partikeln in Wasser, die durch das Stoßen der Wassermoleküle mit den Partikeln verursacht wird:

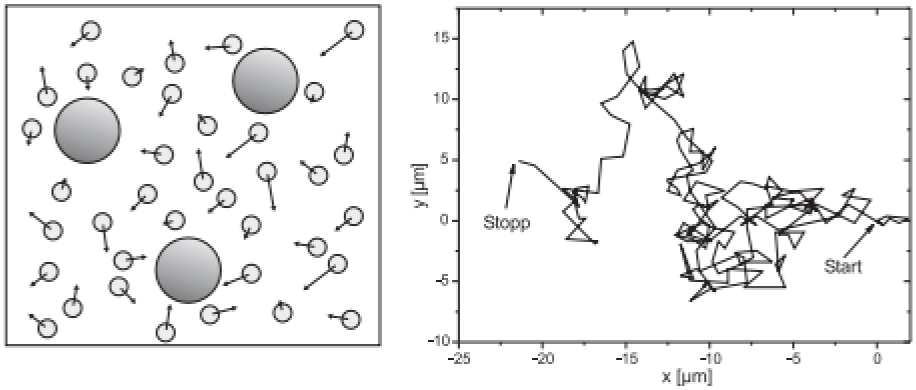


Abbildung 2: Modell der Brownschen Bewegung

Wenn man die Brownsche Bewegung nur in einer Dimension betrachtet, kann man sie mit einem Random-Walk beschreiben:

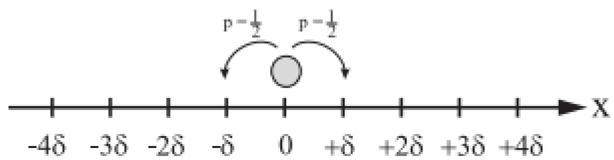


Abbildung 3: Eindimensionaler Random-Walk

Das Partikel befindet sich zum Zeitpunkt $t = 0$ am Punkt $x = 0$ und erfährt alle τ Sekunden einen Stoß. Bei jedem Stoß wird das Partikel um die selbe Distanz δ, mit der Wahrscheinlichkeit 1/2 nach links oder rechts gestoßen. Außerdem werden Wechselwirkungen zwischen den Partikeln vernachlässigt. Damit sich das Teilchen an der Postion $x = m\delta$ befindet, muss es $(n + m)/2$-mal nach rechts, und $(n - m)/2$-mal nach links gestoßen worden sein, mit $n = t/\tau$. Dies ergibt eine Binomialverteilung mit

$$P = \binom{n}{\frac{1}{2}(n+m)} p^{(n+m)/2}(1-p)^{(n-m)/2} = \frac{n!}{(\frac{1}{2}(n+m))!(\frac{1}{2}(n-m))!}\frac{1}{2}^{n} \tag{1}$$

Da n in der Regel sehr groß ist, können wir $n!$ und $m!$ mit der Stirlingschen

Formel nähern:

$$n! = \sqrt{2\pi n} n^n e^{-n} \tag{2}$$

Daraus folgt

$$P = \sqrt{\frac{2}{\pi n}} e^{-\frac{m^2}{2n}} \tag{3}$$

Da m immer entweder gerade oder ungerade ist, gilt $\Delta m = \pm 2$ und somit

$$P(m, n)\frac{\Delta x}{2\delta} = P(x, n)\Delta x \tag{4}$$

Mit $n = \frac{t}{\tau}$, $m = \frac{x}{\delta}$ und $D = \frac{\delta^2}{2r}$ folgt

$$P(x, t)\Delta x = \frac{\Delta x}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}} \tag{5}$$

Dies entspricht einer Gaußverteilung mit $\langle x \rangle = 0$ und $\sigma^2 = 2Dt$:

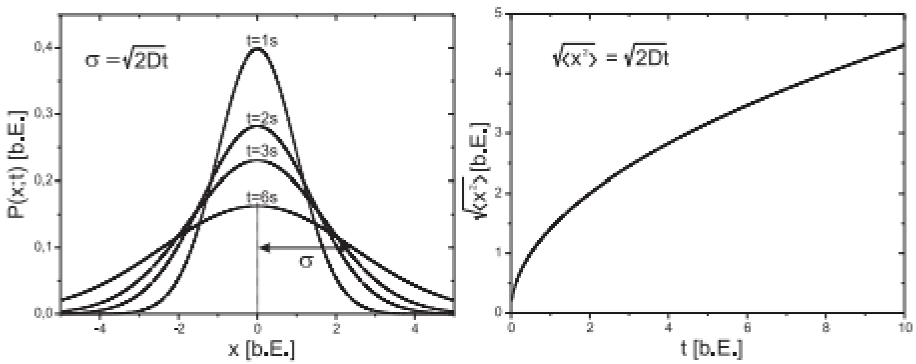


Abbildung 4: Gaußverteilung der 1d-Brownschen Bewegung

Im zweidimensionalen Fall gilt

$$\sigma^2 = \langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle = 4Dt \tag{6}$$

D wird auch als Diffusionskoeffizient bezeichnet und ist gegeben durch

$$D = \frac{kT}{f} \tag{7}$$

mit der Boltzmann-Konstante k, der Temperatur T und dem Reibungskoeffizienten f. Der Reibungskoeffizient ist nach Stokes bestimmt durch

$$f = 6\pi\eta a \tag{8}$$

mit der Viskosität η und dem Partikelradius a. Damit folgt

$$\langle r^2 \rangle = \frac{4kT}{6\pi\eta a} t \tag{9}$$

Durchführung

Wir beginnen mit der Probenpräparation. Dazu schneiden wir ein Stück doppelseitiges Klebeband auf die Größe des Deckglases zu, und stanzen ein Loch zentrisch in das Klebeband. Dann kleben wir das Klebeband auf den Objektträger und geben $250\mu m$ in das Loch. Anschließend kleben wir das Deckglas über das Loch. Dabei sollten sich im Idealfall keine Luftblasen bilden. Zum Schluss geben wir einen tropfen Immersionsöl auf das Deckglas, und spannen die Probe auf den Mikroskoptisch. Wir versuchen dann, mit dem Mikroskop einzelne Partikel zu erkennen.
Anschließend benutzen wir eine Software, um jede Sekunde ein Bild der Probe zu machen, für insgesamt 150 Bilder. Dabei sollte ein Teilchen ungefähr in der Mitte sein, welches sich über 150 Bilder verfolgen lässt.
Danach notieren wir die Zimmertemperatur, und beginnen mit der Eichung. Dazu spannen wir das Objektmikrometer auf den Objekttisch, und stellen scharf, bis wir die Skala gut erkennen können, und nehmen ein Bild auf.
Anschließend starten wir die Auswertungssoftware und laden das Eichbild. Wir bestimmen mit dem Cursor den Pixelabstand von $20\mu m$ und tragen diesen in die Software ein. Dann laden wir alle 150 Bilder hintereinander und setzen den Marker auf die Position des zu verfolgenden Partikels, womit dessen Position abgespeichert wird.

Messprotokoll - Brown'sche Bewegung

22.11.21

Furkan Cetin, Felix Fleischle, Laura Scholl

Gruppe 2, nachmittags

Geräte

- Durchlichtmikroskop Motic B1 mit CCD-Kamera
- Partikel suspendiert in Wasser
- PC mit Drucker
- Thermometer
- Objektmikrometer

Zimmertemperatur: $T = (21{,}3 \pm 0{,}1)\,°C$

Ø Partikel: $d = (755 \pm 30)\,nm$

Eichung: $20\,\mu m \rightarrow 206\,px$

Auswertung und Zusammenfassung

Die Auswertung wurde in Python durchgeführt, daher hier nur eine Zusammenfassung, getrennt von der Diskussion. Der Code ist am Ende des Dokuments zu finden.

Graphische Darstellung

Zuerst haben wir unsere Messdaten importiert und die x-y-Wertepaare graphisch dargestellt. Dabei ergab sich dieses Diagramm:

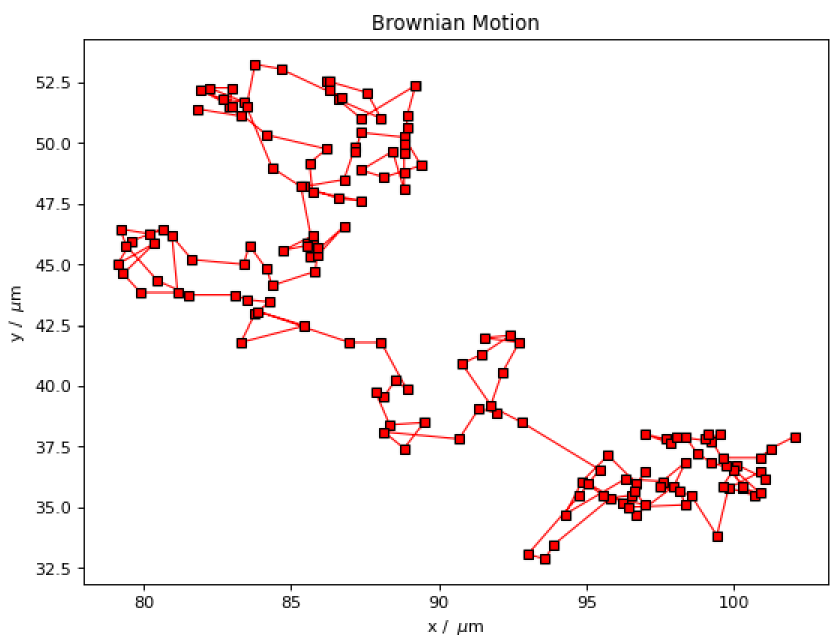


Abbildung 5: Graphische Darstellung der Brownschen Bewegung

Die Darstellung sieht aus wie erwartet.

Berechnung der Boltzmann-Konstante und der Diffusionskonstante aus dem mittleren Verschiebungsquadrat

In diesem Versuchsteil haben wir zuerst $\Delta x_i = x_{i+1} - x_i$ und $\Delta y_i = y_{i+1} - y_i$ aus unseren Daten berechnet, und daraus das Verschiebungsquadrat $r_i^2 = \Delta x_i^2 + \Delta y_i^2$

berechnet. Als dessen Mittelwert haben wir das Ergebnis

$$r^2 = (1,77 \pm 0,15) \cdot 10^{-12} m^2$$

Daraus haben wir die Boltzmann-Konstante berechnet zu

$$k = \frac{6\pi\eta a}{4Tt} r^2 = (1,04 \pm 0,10) \cdot 10^{-23} \frac{J}{K}$$

Wenn wir dies mit dem Literaturwert vergleichen, ergibt sich

$$\sigma_{k1} = \frac{1,380649 - 1,04}{0,1} = 3,4$$

Außerdem haben wir die Diffusionskonstante berechnet über

$$D = \frac{r^2}{4t} = (4,4 \pm 0,4) \cdot 10^{-13} \frac{m^2}{s}$$

Kontrollverteilung

Anschließend haben wir die Partikelverschiebungen in ein Histogramm eingetragen, sowohl die Δx_i als auch Δy_i. Das Ergebnis war das folgende Histogramm:

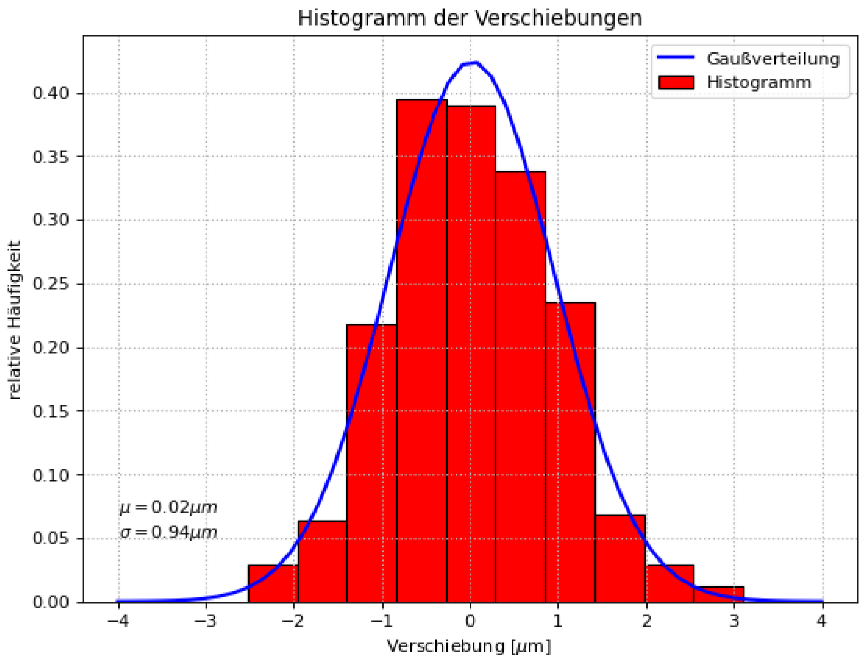


Abbildung 6: Histogramm der Partikelverschiebungen

In das Histogramm wurde zusätzlich eine Gaußverteilung mit den berechneten Mittelwerten der Verschiebung und Standardabweichung der Verschiebung eingezeichnet. Wir erkennen, dass das Histogramm die Gaußverteilung gut annähert.

Kummulative Verteilung der Verschiebungsquadrate

Wir plotten die kummulative Verschiebung als Funktion der Zeit:

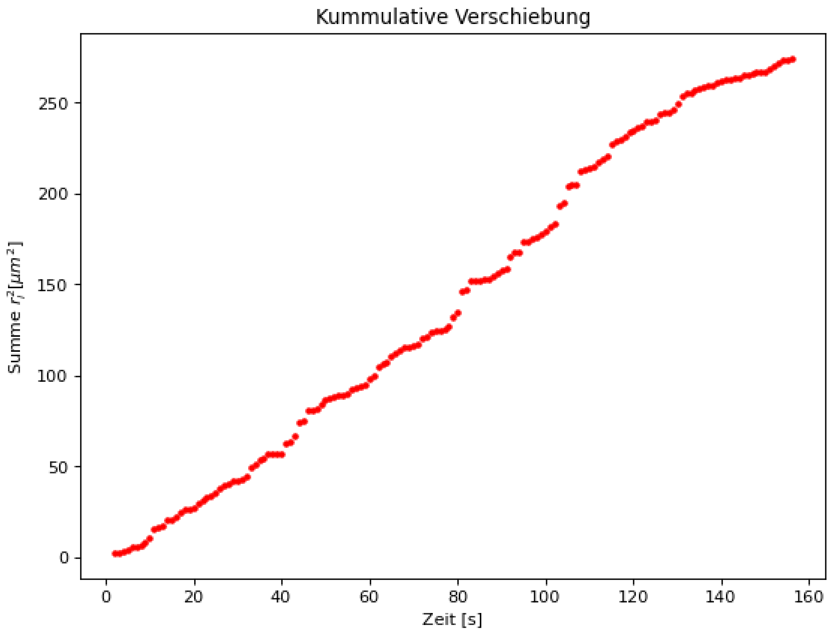


Abbildung 7: Kummulative Verschiebung als Funktion der Zeit

Wir haben dann einen fit durchgeführt, welcher uns eine Gerade mit der Steigung $s = (1,95023 \pm 0,00018) \cdot 10^{-12} \frac{m^2}{s}$ ergeben hat:

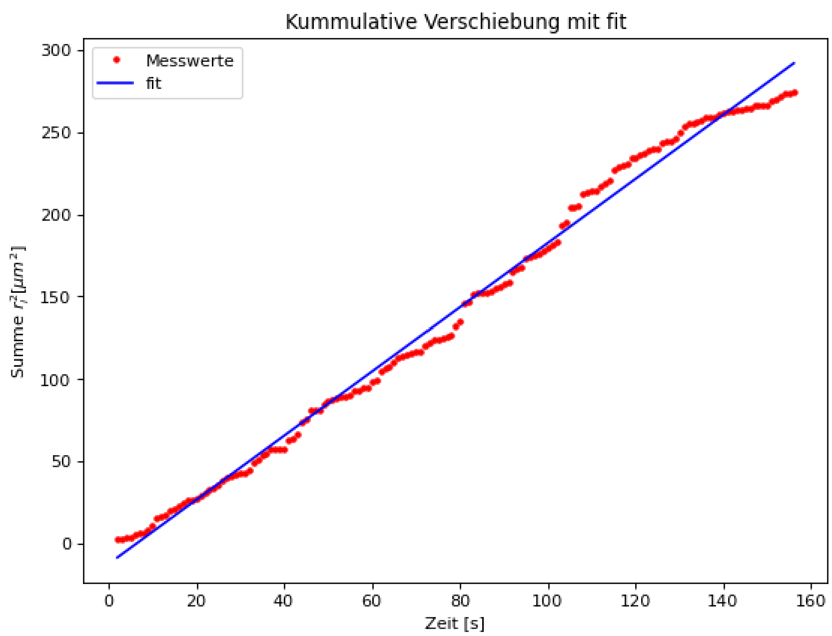


Abbildung 8: Kummulative Verschiebung als Funktion der Zeit mit fit

Aus der Steigung haben wir nun die Boltzmann-Konstante errechnet:

$$k_2 = \frac{6\pi\eta a s}{4T} = (1,15 \pm 0,05) \cdot 10^{-23} \frac{J}{K}$$

Dies entspricht einer Abweichung zum Literaturwert von

$$\sigma_{k2} = \frac{1,380649 - 1,15}{0,05} = 4,61$$

Wir haben außerdem erneut die Diffusionskonstante berechnet:

$$D_2 = \frac{s}{4} = (4,88 \pm 0,03) \cdot 10^{-13} \frac{m^2}{s}$$

Die Abweichung zwischen unseren beiden Diffusionskonstanten beträgt

$$\sigma_D = \frac{4,88 - 4,4}{\sqrt{0,03^2 + 0,4^2}} = 1,2$$

Diskussion

Insgesamt waren die meisten Ergebnisse bei diesem Versuch gemäß unseren Erwartungen.
Die Abweichung vom Literaturwert der Boltzmann-Konstante unserer Werte ist jedoch bei beiden Methoden größer als 3 sigma. Der Grund dafür könnte zum Beispiel die Luftblase sein, die nach dem Einspannen der Probe auf dem Objekttisch zu erkennen war. Wir haben versucht, uns bei der Durchführung so weit wie möglich von der Luftblase zu entfernen, trotzdem könnte diese die Bewegungsmuster der Partikel verändert haben.
Der zweite Wert ist näher dran am Literaturwert (auch wenn die Sigma-Abweichung etwas höher ist, da die Steigung durch Python einen sehr kleinen Fehler hat). Dies war auch zu erwarten, da die Methode über die Steigung deutlich genauer ist, als einfach den Mittelwert der Messwerte zu nehmen und in die Formel einzusetzen.

Versuch 223 Brownsche Bewegung - Auswertung

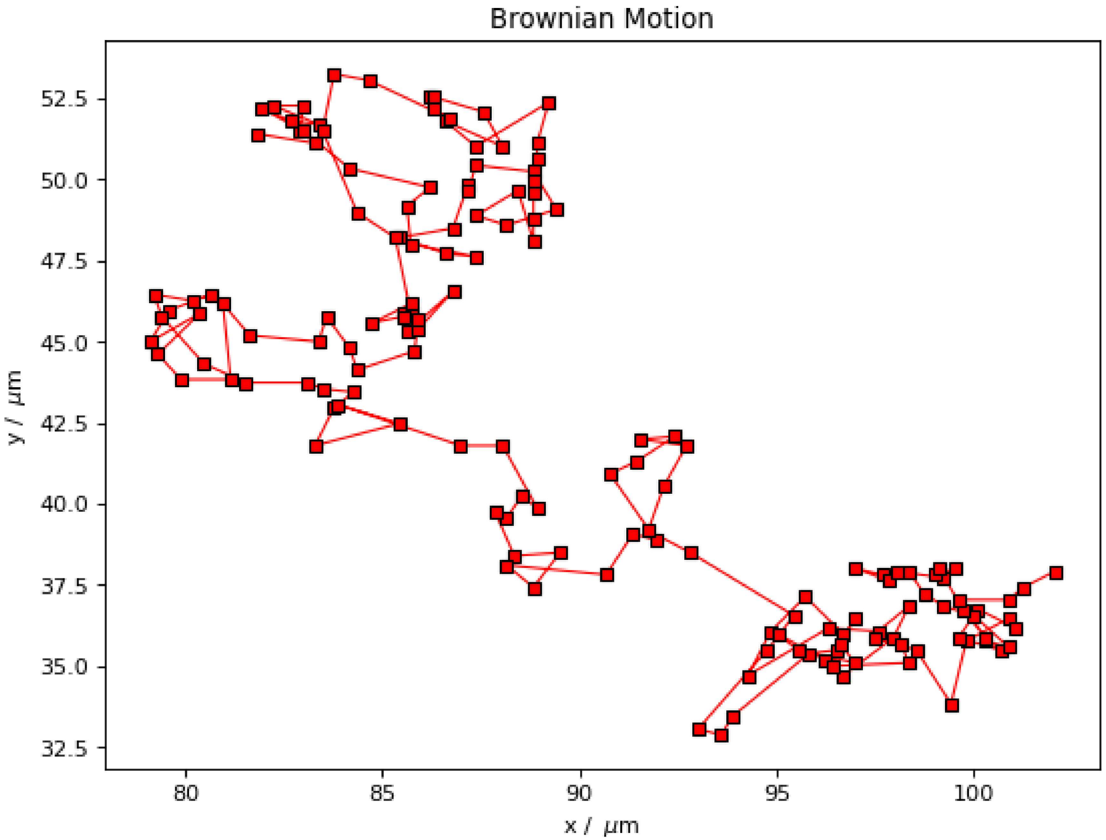
Felix Fleischle - 22.11.2021

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
from scipy.optimize import curve_fit
from scipy.stats import norm
```

In [2]:
```
def comma_to_float(valstr):
    return float(valstr.decode("utf-8").replace(",","."))
t,x,y = np.loadtxt("C:/Users/fexfl/Desktop/data/Messung.dat", skiprows = 1, usecols =(1,2,3), converters={1:comma_to_float, 2:comm
```

In [3]:
```
plt.figure(figsize=(8, 6), dpi=80)
plt.plot(x, y, marker='s', color='red', linewidth=1, markeredgecolor="black")
plt.xlabel('x / '+' $\mu$'+'m')
plt.ylabel('y / '+' $\mu$'+'m')
plt.title('Brownian Motion')
#plt.savefig('figures/brown1.pdf', format='PDF')
```

Out[3]:
```
Text(0.5, 1.0, 'Brownian Motion')
```



In [4]:
```
dt=np.array([])
dx=np.array([])
dy=np.array([])
i=0

while i < len(t)-1:
    dt=np.append(dt,t[i+1]-t[i])
    dx=np.append(dx,x[i+1]-x[i])
    dy=np.append(dy,y[i+1]-y[i])
    i = i + 1

r_squared=dx**2+dy**2
```

In [5]:
```
r_squared_mean=np.mean(r_squared) * 10**-12
print("r_squared_mean= " ,r_squared_mean)
```

```python
r_squared_mean_std=np.std(r_squared)/np.sqrt(len(r_squared)) * 10**-12
print("r_squared_mean_std= " ,r_squared_mean_std)
dt_mean=np.mean(dt)
print("dt_mean= ", dt_mean)
```

```
r_squared_mean=  1.7697191032258056e-12
r_squared_mean_std=  1.5387279956266915e-13
dt_mean=  1.000858064516129
```

In [6]:

```python
# Messwerte für a,T und eta aus Diagramm
eta = 0.98 * 10**(-3) # Pa s
eta_err = 0.01 * 10**(-3)
a = 0.5 * 755 * 10**(-9) # m
a_err = 0.5 * 30 * 10**(-9)
T = 21.3 + 273.15 # K
T_err = 0.1

k = 6 * np.pi * eta * a * r_squared_mean / (4 * T * dt_mean)

def errorFrac(x, x_err, p):
    return (x_err * p)/x

eta_errfrac = errorFrac(eta, eta_err, 1)
a_errfrac = errorFrac(a,a_err, 1)
T_errfrac = errorFrac(T,T_err, 1)
r_squared_mean_errfrac = errorFrac(r_squared_mean, r_squared_mean_std, 1)

print(eta_errfrac)
print(a_errfrac)
print(T_errfrac)
print(r_squared_mean_errfrac)

k_err = np.sqrt(eta_errfrac**2 + a_errfrac**2 + T_errfrac**2 + r_squared_mean_errfrac**2) * k

print("Boltzmann-Konstante:", k, "+-", k_err, "[J/K]")
```

```
0.010204081632653062
0.039735099337748346
0.0003396162336559688
0.08694758353582394
Boltzmann-Konstante: 1.046898194372693e-23 +- 1.0064933373080443e-24 [J/K]
```

In [7]:

```python
#D = k * T / (6*np.pi*eta*a)
D = r_squared_mean / (4 * dt_mean)
```

```
k_errfrac = errorFrac(k,k_err,1)

#D_err = np.sqrt(k_errfrac**2 + T_errfrac**2 + eta_errfrac**2 + a_errfrac**2) * D
D_err = r_squared_mean_std / (4 * dt_mean)

print("Diffusionskonstante:", D, "+-", D_err, "[m^2/s]")
```

```
Diffusionskonstante: 4.4205046798553485e-13 +- 3.8435219992222356e-14 [m^2/s]
```

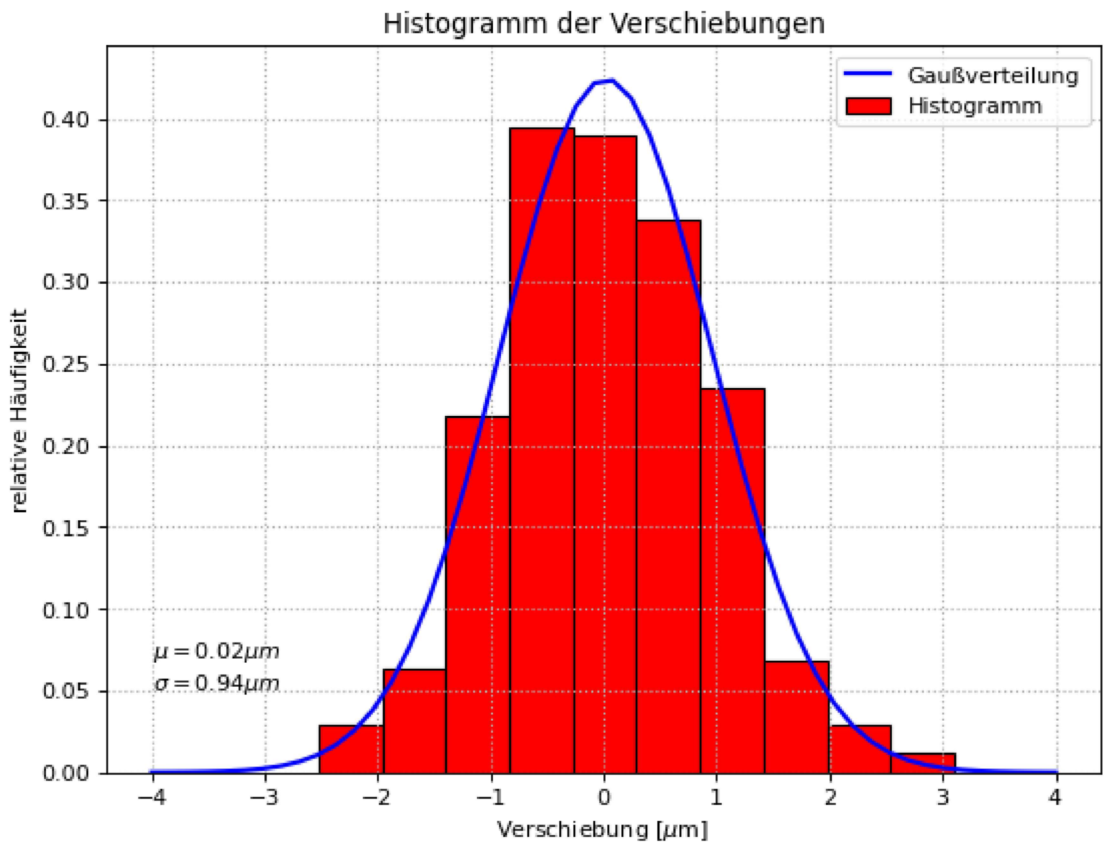
In [8]:

```
all_data = np.append(dx,dy)

mu = np.mean(all_data)
sigma = np.std(all_data)
gauss = norm.pdf(np.linspace(-4,4), mu, sigma)

print(mu)
print(sigma)

plt.figure(figsize=(8, 6), dpi=80)
plt.hist(all_data, density=True, color="red", histtype="bar", label = "Histogramm", edgecolor="black")
plt.plot(np.linspace(-4,4), gauss,"b-" ,linewidth=2, label="Gaußverteilung")
plt.legend()
plt.title("Histogramm der Verschiebungen")
plt.grid(linestyle=":", linewidth=1)
plt.xlabel("Verschiebung [$\mu$m]")
plt.ylabel("relative Häufigkeit")
plt.text(-4, 0.05, "$\sigma = 0.94 \mu m$")
plt.text(-4, 0.07, "$\mu = 0.02 \mu m$")
```

```
0.021609677419354854
0.940421487129433
```

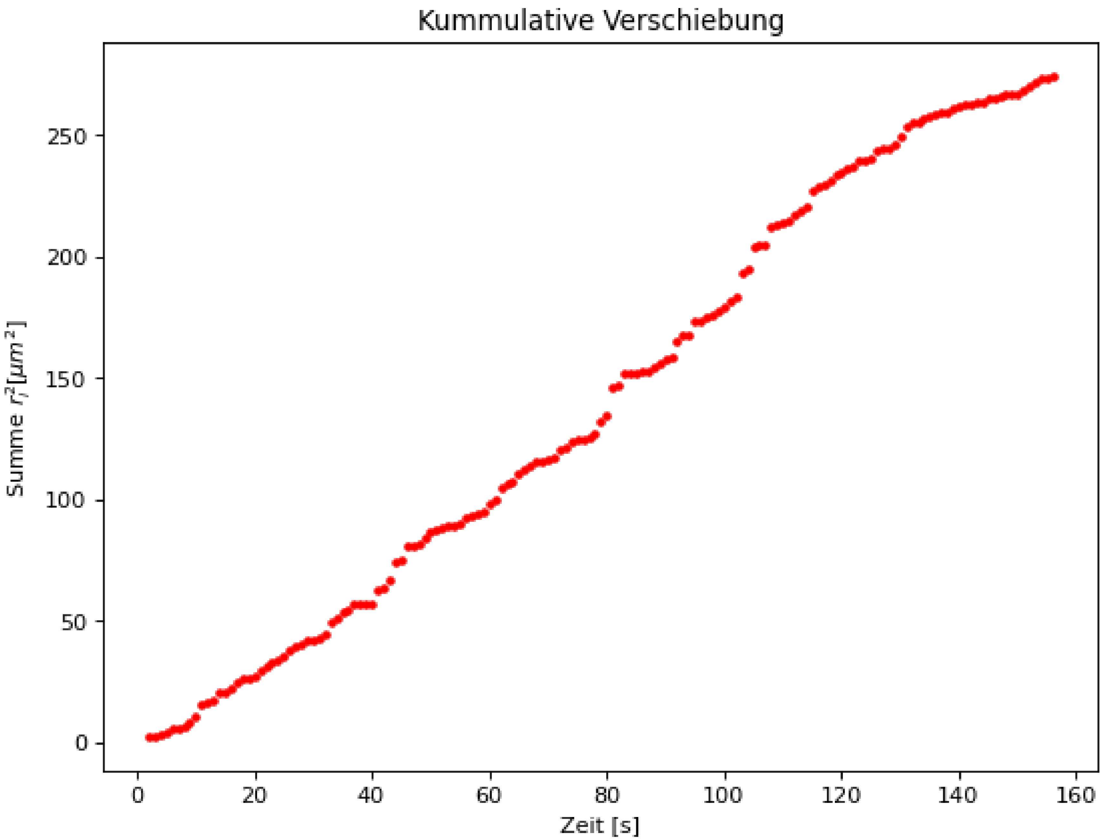
Out[8]:

```
Text(-4, 0.07, '$\\mu = 0.02 \\mu m$')
```



In [9]:
```
plt.figure(figsize=(8, 6), dpi=80)
r_kumm = np.cumsum(r_squared)
plt.plot(t[:-1], r_kumm, marker=".", color="red", linewidth=0)
plt.xlabel("Zeit [s]")
plt.ylabel("Summe $r_i^2 [\mu m^2]$")
plt.title("Kummulative Verschiebung")
```

Out[9]:
```
Text(0.5, 1.0, 'Kummulative Verschiebung')
```



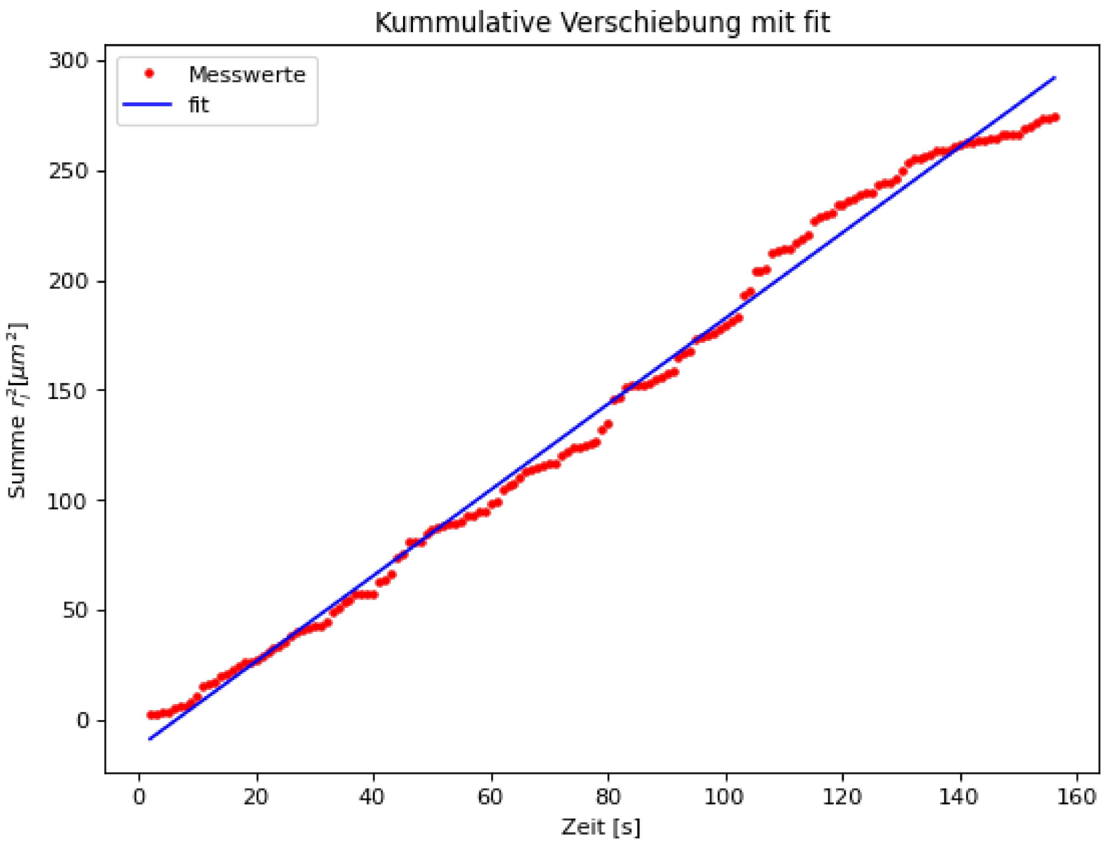
In [10]:
```
def linear(x,s,z):
    return s*x + z

popt, pcov = curve_fit(linear, t[:-1], r_kumm)

s = popt[0]
z = popt[1]
s_err = np.sqrt(pcov[0,0])
z_err = np.sqrt(pcov[1,1])

s_m = s * 10**-12
s_m_err = s_err * 10**-12
print("Steigung:", s_m, "+-", s_m_err, "[m^2/s]")

plt.figure(figsize=(8, 6), dpi=80)
r_kumm = np.cumsum(r_squared)
```

```
plt.plot(t[:-1], r_kumm, marker=".", color="red", linewidth=0, label="Messwerte")
plt.plot(t[:-1], linear(t[:-1], s, z), color="blue", label="fit")
plt.xlabel("Zeit [s]")
plt.ylabel("Summe $r_i^2 [\mu m^2]$")
plt.title("Kummulative Verschiebung mit fit")
plt.legend()
```

```
Steigung: 1.950229949035652e-12 +- 1.3433602608827583e-14 [m^2/s]
```

Out[10]:

```
<matplotlib.legend.Legend at 0x1c438c11160>
```



In [11]:

```
k_s = 6 * np.pi * eta * a * s_m / (4 * T)

s_m_errfrac = errorFrac(s_m, s_m_err, 1)

k_s_err = np.sqrt(eta_errfrac**2 + a_errfrac**2 + s_m_errfrac**2 + T_errfrac**2) * k_s

print("Boltzmannkonstante:", k_s, "+-", k_s_err, "[J/K]")
```

```python
#D_s = k_s * T /(6 * np.pi * eta * a)
D_s = s_m / 4

k_s_errfrac = errorFrac(k_s, k_s_err, 1)

#D_s_err = np.sqrt(k_s_errfrac**2 + T_errfrac**2 + eta_errfrac**2 + a_errfrac**2 ) * D_s
D_s_err = s_m_err / 4

print("Diffusionskonstante:", D_s, "+-", D_s_err, "[m^2/s]")
```

```
Boltzmannkonstante: 1.1546714459783034e-23 +- 4.803439102876003e-25 [J/K]
Diffusionskonstante: 4.87557487258913e-13 +- 3.3584006522068958e-15 [m^2/s]
```

In [ ]:

```python

```