

Masterarbeit
Jan Grzegorzewski

Reaktions-Diffusions Dynamik mit gebrochener Brownscher Bewegung

Sommersemester: 2016

Masterstudiengang: Physics
Abgabetermin : 1. Oktober 2016

Master Thesis
Jan Grzegorzewski

Reaction-diffusion dynamics with fractional Brownian motion

Summer term: 2016

Misc: See first title page

Contents

1 Theory	3
1.1 Brownische Bewegung	3
1.2 Gebrochen-rationale Brownische Bewegung	4
1.3 Algorithmus	4
1.3.1 Analyse des Algorithmuses	7
2 Reaktions-Diffusions-Dynamik	15
3 Appendix	16
3.0.2 Code:Gebrochen-rationale Brownische Bewegung	16
Bibliography	18

List of Figures

1.1	Mean Square Displacement of different α	7
1.2	Mean Square Displacement (Ensemble average of 4000 Trajectories)	8
1.3	Mean Square Displacement (Ensemble average fpr different Δt) . . .	8
1.4	Mean Square Displacement (time and ensemble avarge for different Δt and with standard deviation of the Mean)	9
1.5	Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.	10
1.6	Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.	11
1.7	rescaled function for different times	12
1.8	variere dt bei gleichem Faktor 1	12
1.9	variere dt bei gleichem Faktor 1, logarithmisch	13
1.10	variere Faktor bei gleichem dt	13
1.11	variere Faktor bei gleichem dt, logarithmisch	14

List of Tables

Erklärung

bla

Danksagung

bla

1 Theory

1.1 Brownische Bewegung

Gebrochene Brownische Bewegung ist eine Verallgemeinerung der Brownischen Bewegung. Sie kann benutzt werden bei der Simulation von Anomaler Diffusion.

Theorie von Felix, bisschen von Christoph und die Motivation könnte sein, dass man zwar anomale Diffusion simuliert werden kann wenn man alle Teilchen berücksichtigt. Mit diesem Ansatz könnte man konkret den Einfluss von Gebrochen-rationale Brownische Bewegung auf Reaktionen test.zb.als Vergleich zu normaler Brownian motion und Brownian Motion an sich ist auch schon eine Vereinfachung, welche eine zufällige Gaußzahl annimmt. Das heißt sämtliche Stöße des Teilchens mit anderen Teilchen innerhalb eines Zeitintervalls werden schon zu diesem Wert verallgemeinert. Als Unterschied zu tatsächlichen Simulation von allen Stößen (MD-Simulation). Die Motivation ist einen performanten Integrator für gebrochen-rationale Brownische Bewegung zu entwickeln. Die Anomale Diffusion, welche besonders in biologischen Systemen zu beobachten ist, wird durch Interaktion der Teilchen mit ihrer Umgebung verursacht. Für den im Verlauf verwendeten Integrator muss davon ausgegangen werden, dass keine weiteren Interaktion (Potentiale) auftreten. Da zur Erstellung der Trajektorie ihre sämtlichen Inkremente im voraus durch gewöhnliche Brownische Bewegung erstellt werden müssen. Der Vorteil von der Methode ist ihre Performance. Der hier verwendete Algorithmus leitet sich vom Davies-Haste Algorithmus [1] ab. Dabei werden anfänglich alle Inkremente (Geschwindigkeiten) für gewöhnliches Braunische Bewegung erzeugt $\eta_{Br}(t) = v(t)$. Für die Entfernung eines Teilchens zu seinem Ursprung gilt $\Delta R(t)_{Br} = R(t) - R(0) = \int_0^t dt' v(t')$. Es ergibt sich für die Mittlere Quadratische Verschiebung:
 $\delta r_{norm}^2(t) = <\Delta R(t)_{Br}> = 2dDt$ mit d = Anzahl der Dimension, D = Diffusionskonstante und t = Zeit. Da gewöhnliche Brownische Bewegung einer Gaußverteilung folgt und ein Markowischer Prozess ist, lässt sich für den Propagator folgende Gleichung aufstellen:

$$P(r, t) = [2\pi\delta r_{norm}^2(t)/d]^{-\frac{d}{2}} e^{\frac{-r^2 d}{2\delta r_{norm}^2(t)}} \quad (1.1)$$

1.2 Gebrochen-rationale Brownische Bewegung

Als Ausgangspunkt wird weiterhin die Differentialgleichung $\partial \mathbf{R}(t) = \boldsymbol{\eta}(t)$ betrachtet. $\eta_i(t)$ ist in dem Fall aber nicht mehr delta korreliert in der Zeit, wie für einfache Brownische Bewegung. Für Gebrochen-rationale Brownische Bewegung wird eine dauerhaftes Korrelation angenommen, sodass die Mittlere Quadratische Verschiebung nicht mit der für Brownische Bewegung übereinstimmt, aber stattdessen eine exponentielle Abhängigkeit zur Zeit aufweist $\delta r_{fBr}^2(t) = <\Delta R(t)_{fBr}> = 2dK_\alpha t^\alpha$. Die Geschwindigkeitsautokorrelationsfunktion für Gebrochen-rationale Brownische Bewegung kann wie folgt definiert werden:

$$Z(\omega) = K_\alpha \Gamma(1 + \alpha)(iz)^{1-\alpha} \text{ für } \quad (1.2)$$

- K_α = generalisierter Diffusions-Koeffizient
- ω = Frequenz

[2]

1.3 Algorithmus

Für den im Verlauf verwendeten Integrator muss davon ausgegangen werden, dass keine weiteren Interaktion (Potentiale) auftreten. Da zur Erstellung der Trajektorie ihre sämtlichen Inkremente im voraus durch gewöhnliche Brownische Bewegung erstellt werden müssen. Der Vorteil von der Methode ist ihre Performance. Der hier verwendete Algorithmus leitet sich vom Davies-Haste Algorithmus [1] ab. Die theoretischen Überlegung aus dem Theorienteil müssen für den Algorithmus in eine diskrete Form gebracht werden.

Das Inkrement ist:

$$\boldsymbol{\eta}(t) \longrightarrow \boldsymbol{\eta}_j \text{ mit } j = (0, 1, 2, \dots, n) \text{ und } n = \text{ Schrittanzahl} \quad (1.3)$$

Die dickgeschriebenen Inkremente $\boldsymbol{\eta}_j$ sind als Vektoren zu verstehen. Die Schreibweise wird im Verlauf so belassen. Es ergibt sich für die Länge der Trajektorie:

$$\Delta \mathbf{R}_n = \sum_{j=0}^n \boldsymbol{\eta}_j \Delta t \quad (1.4)$$

Der Algorithmus hat die Motivation die Inkremente $\boldsymbol{\eta}_j$ so zu generieren, dass sie die Eigenschaften der Gebrochen-rationale Brownische Bewegung wiederspiegeln und sie

insbesondere dem potentiellen Abklingen der Mittleren Quadratischen Verschiebung folgen.

$$\langle \Delta R_{fbr} \rangle = 2dK_\alpha t^\alpha \quad (1.5)$$

Der verwendete Algorithmus geht wie folgt:

1. Es werden $2n$ unabhängige Normalverteilte, mit dem Mittelwert $\langle \eta_j \rangle = 0$ und der Standardabweichung $\delta\eta_j = \sqrt{\Delta t}$, Zufallszahlen erstellt:

$$\eta_j = (\eta_0, \eta_1, \eta_2, \dots, \eta_{2n}) = \mathcal{N}(0, \sqrt{\Delta t}) \text{ mit } j = (0, 1, 2, \dots, 2n) \quad (1.6)$$

Im Verlauf des Algorithmus wird die Motivation für die doppelte Anzahl an Zufallszahlen gegenüber der resultierenden Trajektorienlänge für gebrochen-rationalen Brownische Bewegung erläutert.

2. Die Inkremente werden dann mit Hilfe einer diskreten Fouriertransformation (numpy.fft) in den Frequenzraum transformiert.

$$\eta_g = \sum_{j=0}^{2n-1} \eta_j e^{\frac{-i2\pi j g}{2n}} \text{ mit } g = (0, 1, 2, \dots, 2n) \quad (1.7)$$

$$(1.8)$$

Die entsprechende analytische Fouriertransformation ist:

$$\eta(\omega) = \int_0^\infty e^{-i2\pi\omega t} \eta(t) dt \quad (1.9)$$

$$\text{mit } \omega = g\Delta\omega, \Delta\omega = \frac{1}{2n\Delta t} \text{ und } t = j\Delta t \quad (1.10)$$

3. Daraufhin wird die Auto-Korrelation Funktion [2] angewendet, wodurch die Inkremente jetzt einen gebrochen-rationalen Brownischen Charakter bekommen:

$$\eta_{fbg} = \eta_g \sqrt{2\operatorname{Re}(Z_g)} \quad (1.11)$$

$$Z(\omega) = K_\alpha \Gamma(1 + \alpha) (-i\omega)^{1-\alpha} \quad (1.12)$$

$Z(\omega)$ wurde durch die analytische Fouriertransformation $Z(\omega) = \int_0^\infty e^{i\omega t} Z(t) dt$ bestimmt. Da Numpy FFT eine andere Definition der Fouriertransformation

benutzt (siehe Formel 1.9),
wird $Z'(\omega) = \int_0^\infty e^{-i2\pi\omega t} Z(t) dt = K_\alpha \Gamma(1 + \alpha)(i\omega 2\pi)^{1-\alpha}$ so umgeschrieben.
Im Anschluss wird $Z(\omega) \rightarrow Z_g$ (nach Formel 1.10) in eine diskrete Form
gebracht.

$$\rightarrow Z_g = K_\alpha \Gamma(1 + \alpha)(i2\pi g \Delta\omega)^{1-\alpha} = K_\alpha \Gamma(1 + \alpha)(ig \frac{\pi}{n\Delta t})^{1-\alpha} \quad (1.13)$$

4. Für gebrochen-rationale Brownischen Bewegung ist die Auto-Korrelations Funktion an der Stelle 0: $Z_{g=0} = 0$. Daraus folgt nach Formel 1.11, dass auch das Nullte Inkrement im Frequenzraum $\eta_{fbr_{g=0}} = 0$ ist. Das Nullte Inkrement im Frequenzraum ist mit Formel 1.4 aber auch:

$$\eta_{fbr_{g=0}} = \sum_{j=0}^{2n-1} \eta_j e^0 = \frac{\Delta R}{\Delta t} \quad (1.14)$$

ΔR ist in dem Fall die zurückgelegte Entfernung nach $2n$ Zeitschritten. Dadurch würde das Teilchen nach $2n$ Schritten wieder zurück an seinen Ursprung kehren. Stattdessen wird Das Nullte Inkrement im Frequenzraum folgendermaßen berechnet:

$$\eta_{fbr_{g=0}} = \mathcal{N}(0, \sqrt{2K_\alpha(2n\Delta t)^\alpha}) \quad (1.15)$$

gerechnet. Dies wäre korrekt, wenn Gebrochen-rationale Brownische Bewegung ein Markowischer Prozess wäre. Da dies nicht der Fall ist, entsteht an dieser Stelle eine Approximation. Um sie genauer zu verstehen muss ich mich damit noch weiter beschäftigen. Um den Einfluss der Approximation zu verringern wurde in Schritt 1 die doppelte Menge an Inkrementen erstellt. Die Hoffnung ist, dass der Einfluss der Approximation mit zunehmender Entfernung von $2\Delta R_{fbr}$ immer geringer wird und bei ΔR_{fbr} bereits vernachlässigbar ist.

5. In der Zeitdomäne ergeben sich die Inkremente durch die Rücktransformation als:

$$\eta_{fbr_j} = \frac{1}{2n} \sum_{g=0}^{2n-1} \eta_g e^{\frac{2\pi i j g}{2n}} \quad (1.16)$$

Es wird nur die erste Hälfte der Inkremente, η_{fbr_j} für $j = (0, 1, \dots, n)$, weiter verwendet.

Für die drei dimensionale Gebrochen-rationale Brownische Bewegung kann für jede kartesische Komponente der soeben beschriebene Algorithmus verwendet werden, da davon ausgegangen wird, dass die Kartesischen Komponenten der Inkremente untereinander nicht korreliert sind.

1.3.1 Analyse des Algorithmuses

Hier kommen die Plots aus dem Analyse Skript rein.

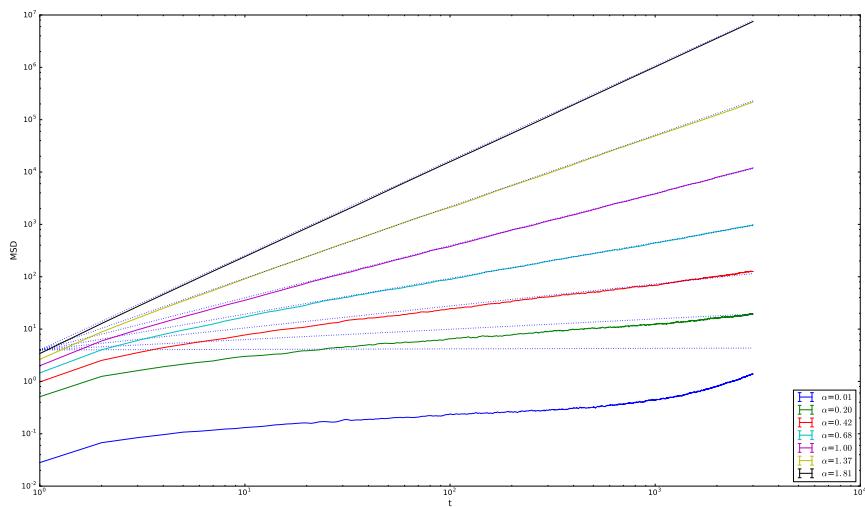


Figure 1.1: Mean Square Displacement of different α

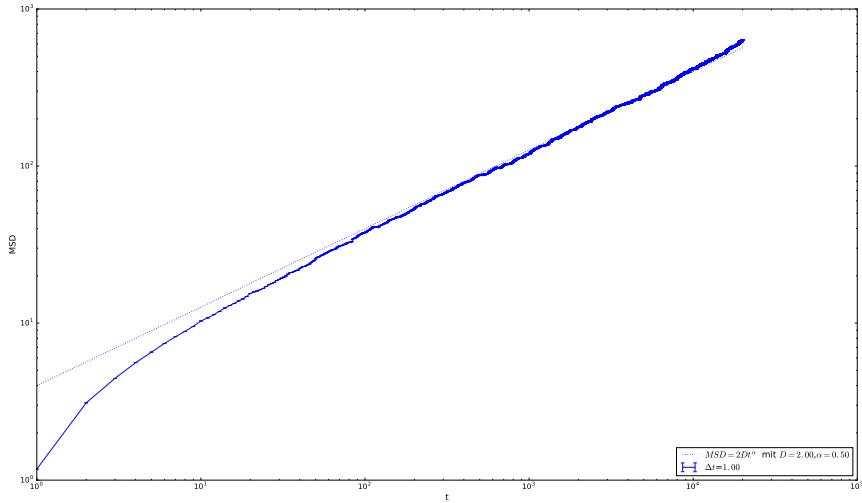
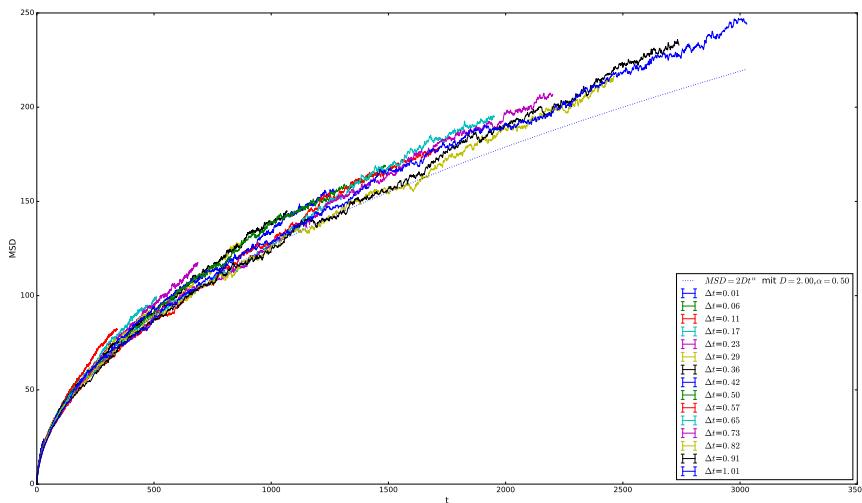


Figure 1.2: Mean Square Displacement (Ensemble average of 4000 Trajectories)

Figure 1.3: Mean Square Displacement (Ensemble average fpr different Δt)

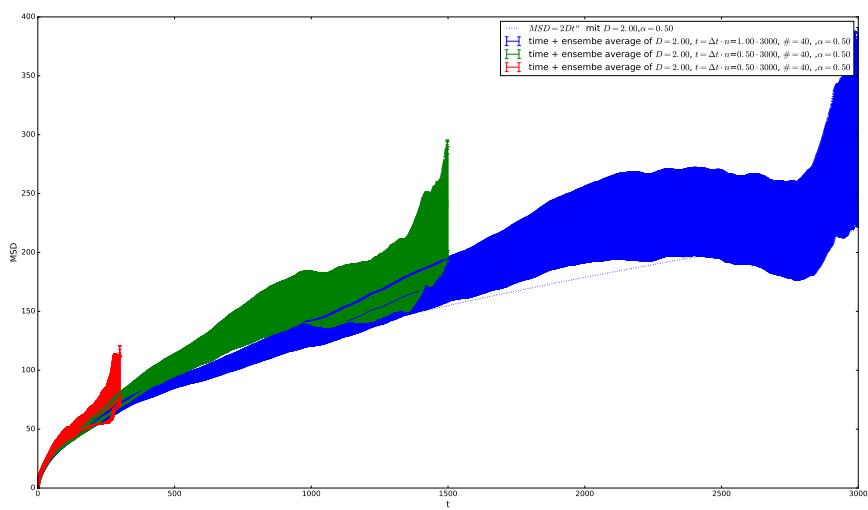


Figure 1.4: Mean Square Displacement (time and ensemble average for different Δt and with standard deviation of the Mean)

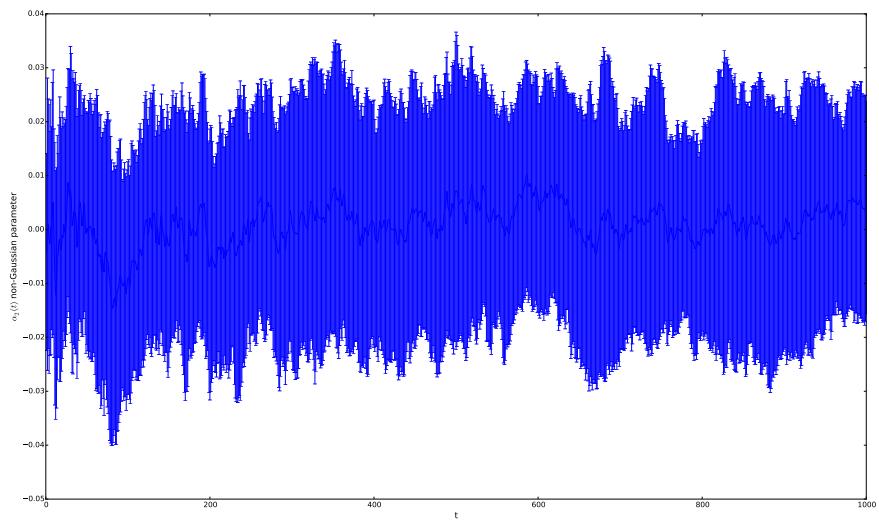


Figure 1.5: Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.

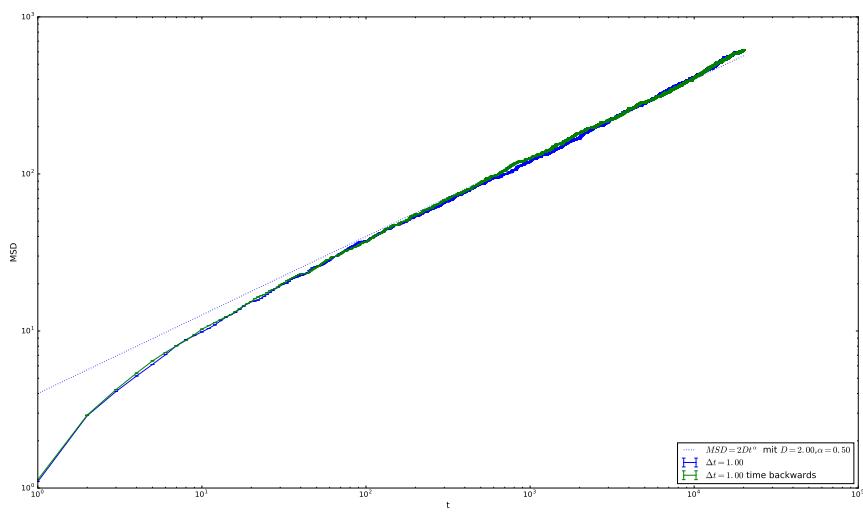


Figure 1.6: Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.

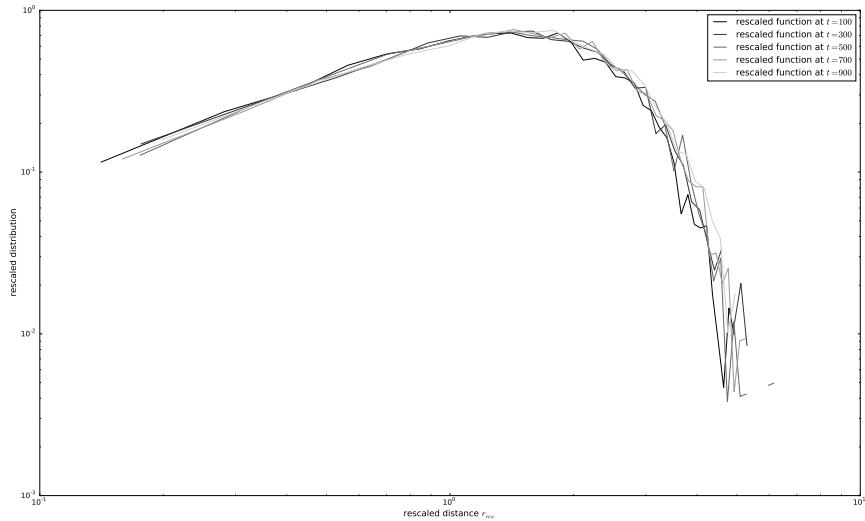


Figure 1.7: rescaled function for different times

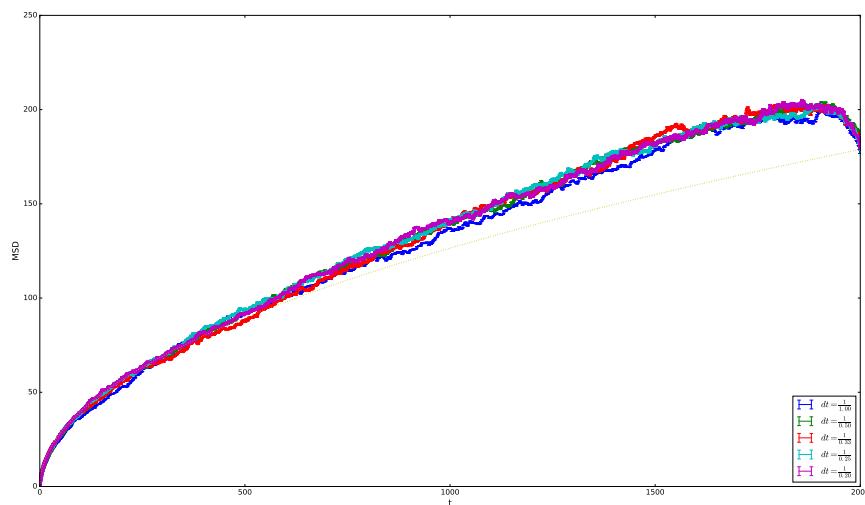


Figure 1.8: variere dt bei gleichem Faktor 1

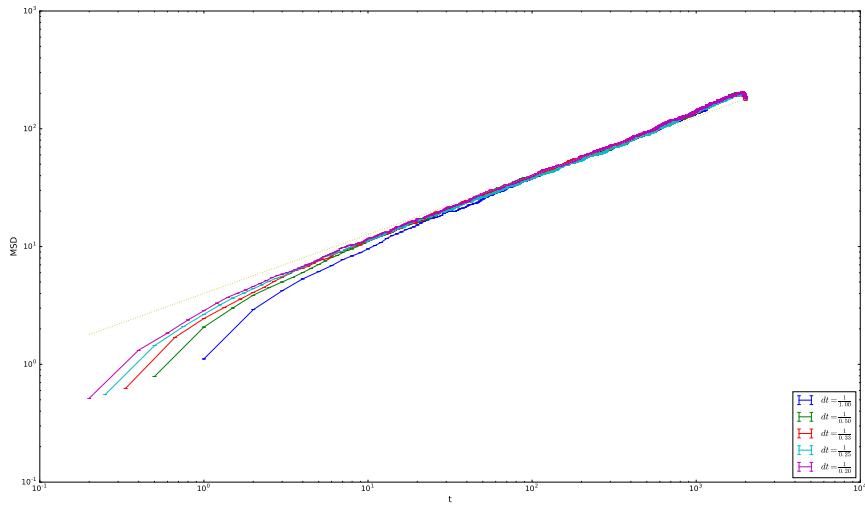


Figure 1.9: variere dt bei gleichem Faktor 1, logarithmisch

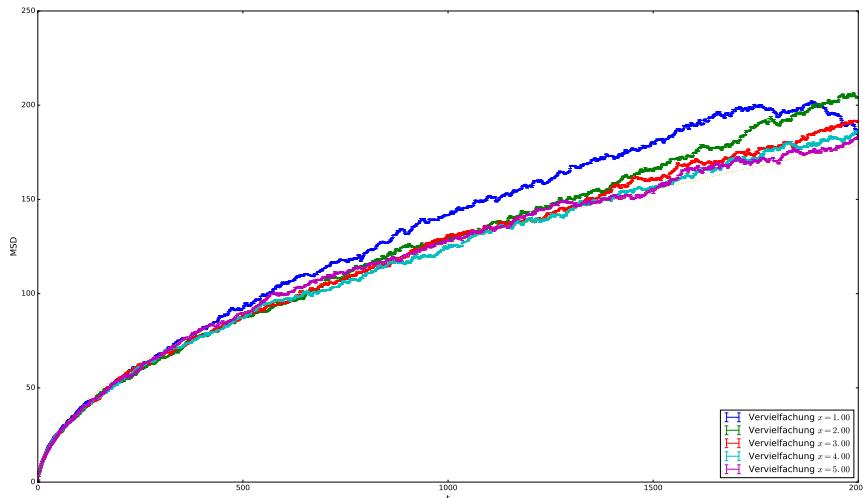


Figure 1.10: variere Faktor bei gleichem dt

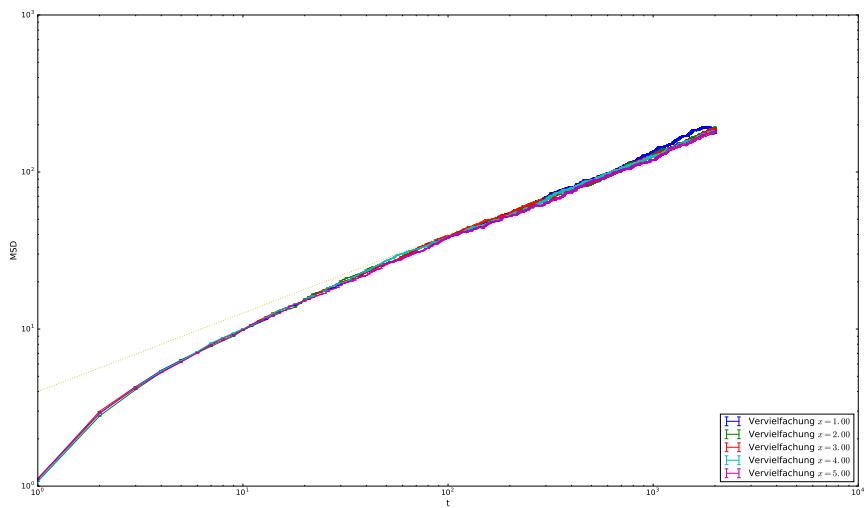


Figure 1.11: variere Faktor bei gleichem dt, logarithmisch

2 Reaktions-Diffusions-Dynamik

3 Appendix

3.0.2 Code: Gebrochen-rationale Brownische Bewegung

```
1 # -*- coding: utf-8 -*-
2 --author-- = 'janek'
3 import numpy as np # module for scientific computing
4 from scipy import integrate
5 import matplotlib.pyplot as plt # module for plotting "a la" matlab
6
7 class Felix_Method():
8
9     def __init__(self, D, particles, length, alpha, dt, x=2):
10         self.D=D
11         self.K_alpha=D
12         self.particles=particles
13         self.n=length
14         self.alpha=alpha
15         self.ki=self.n*x
16         self.frq=np.fft.freq(self.ki)*(np.pi*2. / (dt))
17         self.t=np.array(range(length))
18         self.dt=dt
19
20     def z(self):
21         """
22             :param D: Diffusionskoeffizient
23             :param alpha: Anomalieparameter
24
25
26             :return: z: Correlationsfunktion im Frequenzraum welche auf die
27             gew hnliche diffusion angewendet wird.
28
29
30         """
31         z=((((+1j * self.frq))** (1 - self.alpha)) * self.K_alpha * np.math.
32         gamma(1 + self.alpha)) * np.exp(np.pi / self.ki)
33         return z
```

```

35     def compute_trajectory(self):
36         """
37             :param D: Diffusionskoeffizient
38             :param particles: Anzahl an Teilechen welche simuliert werden
39             sollen.
40
41             :param length: Länge der Trajektorien, welche simuliert werden
42             sollen.
43
44             :return: ( Particles x Length ) Array of Trajectories of all
45             particles. Close to :cite:`Graigmile2003`
46
47             """
48         r_t_allparticles = []
49         #plt.plot(self.freq, "r")
50         #plt.plot( np.arange(-np.pi/self.dt, np.pi/self.dt,(np.pi/(
51             self.n*self.dt))))
52         #plt.show()
53         for particle in range(self.particles):
54             #r = np.random.RandomState(1234)
55             v_t=(np.random.normal(0,np.sqrt(self.dt),size=self.ki)) # todo mit matrix.shape k nnen man eine zufällige Verteilung von allen daten generieren
56             v_t=np.array(v_t)
57             v_frfq=np.fft.fft(v_t)
58             v_ano_frfq= np.sqrt(self.z().real*2.)*v_frfq
59             v_ano_frfq[0]=np.random.normal(0,np.sqrt(2.*self.K_alpha*(self.ki*self.dt)**self.alpha))
60             #v_ano_frfq[ self.n]=np.sqrt( self.z()[ self.n].real* self.n*2)*
61             v_t[ self.ki-1]
62             #v_ano_frfq[ self.n-1]=np.sqrt( self.z()[ self.n].real* self.n*2)*
63             v_t[ self.ki-1]
64
65             v_ano_t=np.fft.ifft(v_ano_frfq)
66             #r_t1=np.cumsum(v_ano_t[:self.n].real) #Ort bei anomaler Diffusion in Abhängigkeit von der zeit
67             r_t=integrate.cumtrapz(v_ano_t[:self.n].real, initial=0) # Ort bei anomaler Diffusion in Abhängigkeit von der zeit
68             r_t_allparticles.append(r_t) # Trajektorie bei anomaler Diffusion für alle Teilchen
69         return np.array(r_t_allparticles)

```

..../simulation.py

Bibliography

- [1] Peter F. Craigmile.
Simulating a class of stationary Gaussian processes using the Davies-Harte algorithm, with application to long memory processes.
Journal of Time Series Analysis, 24(5):505–511, 2003.
- [2] Felix Höfling and Thomas Franosch.
Anomalous transport in the crowded world of biological cells.
Reports on Progress in Physics, 76(4):046602, apr 2013.