

Midtherm Report
Jan Grzegorzewski

Reaction-diffusion dynamics with fractional Brownian motion

Summer term: 2016

Contents

1	Introduction	1
2	Theory	3
2.1	Brownian Motion	3
2.2	Fractional Brownian Motion	4
2.3	Algorythm	4
2.3.1	Analyse of Algorythm	7
3	Reaktions-Diffusions-Dynamics	15
4	Appendix	16
4.0.2	Code:Gebrochen-rationale Brownische Bewegung	16
	Bibliography	19

List of Figures

2.1	Mean Square Displacement of different α	7
2.2	Mean Square Displacement (Ensemble average of 4000 Trajectories)	8
2.3	Mean Square Displacement (Ensemble average fpr different Δt) . . .	8
2.4	Mean Square Displacement (time and ensemble avarge for different Δt and with standard deviation of the Mean)	9
2.5	Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.	10
2.6	Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.	11
2.7	rescaled function for different times	12
2.8	variere dt bei gleichem Faktor 1	12
2.9	variere dt bei gleichem Faktor 1, logarithmisch	13
2.10	variere Faktor bei gleichem dt	13
2.11	variere Faktor bei gleichem dt, logarithmisch	14

List of Tables

1 Introduction

Anomalous diffusion can be observed in many different areas of nature, in particular related to heterogeneous media (Crowded biological media, porous materials, ...). The most popular phenomenon of anomalous diffusion is a power-law behavior of the Mean-Square-Displacement (MSD) $MSD \propto t^\alpha$, which is violating the Einstein formula $MSD = 2dDt$ and thereby the central limit theorem. Various theoretical models try to encounter the power-law behavior of the MSD showing different origins for anomalous diffusion. These models have the main phenomenon, thus the power-law behavior of the mean square displacement, in common. But differ in some other observables, due to different origins of the anomalous power-law behavior. By measuring theses different quantities one might reveal the origin of the anomalous behavior in an experimental experiment.

The most general model which accounts this phenomenon so far is in my opinion Fractional Brownian Motion. This thesis is going to focus on Fractional Brownian Motion especially in respect to Reaction and Diffusion Dynamics. (Normal Diffusion Limited Smoluchowski equation ????? check it) is failing miserably in describing Reaction and Diffusion Dynamics while anomalous motion is present. The overall work aims not only to bring Reaction Diffusion and Fractional Brownian Motion together but also providing a Fractional Brownian Motion Integrator implemented into a **Reaction Diffusion Dynamics** software (ReaDDy). ReaDDy being a particle-based **Reaction Diffusion Dynamics** software is acting on a macromolecular level. It is coarse-graining molecules into spherical particles. Usually using Brownian or Langevin motion as the integrator. A Fractional Brownian Motion Integrator may be an elegant way of adding an approximation due to a crowded environment. The phenomenon of anomalous motion can also be studied with conventional integrators by actually building a crowded environment (for example: by adding obstacles). Nevertheless, the Fractional Brownian Motion approach could save computational time since it is not necessary to simulate each particle explicitly which builds up the crowded environment. Furthermore different studies show that heterogeneous environments have influences on particle segregation in the presents of reactions.

Having an Introduction set, Chapter 2 deals with the theoretical foundation for Fractional Brownian Motion. Subsequently, Due to computational reasons these

theoretical elaborations need to be transferred into discrete space and finally casted into an algorithm. The properties of the algorithm are studied. In chapter 3 theoretical foundation are going to be set. The final chapter 4 will deal with the present state of the thesis and give an outlook on further challenges.

2 Theory

2.1 Brownian Motion

Fractional Brownian motion is a generalized case of standard Brownian Motion. First I explain the behavior the behavior of standard Brownian Motion and than continue for the generalized case.

Standard Brownian Motion is the most important and best studied stochastic process.

(Felix skript + Roland Netz Beweis für $MSD = 2dDt$ angeblich auch gegen Ende der Vorlesung von Felix.

Erratic motion for mesoscopic particles have been first documented by Jan Ingenhousz in 1785, in particular of coal dust on the surface of alcohol. Later on in 1827 Robert Brown observed the erratic motion pollen grains.

Brownian motion is an stochastic process $\{W_t\}_{t \geq 0} : \Omega \rightarrow \mathbb{R}^n$ with $W_t(\omega)$ being the position of a particle with $\omega \in \Omega$ at time $t \in T$ in the observation time $T = [0, \infty)$.

Brownian Motion has several properties: (independet inceremnts – ζ vanishing covaraince) self-similarity fractal path, Gaussian process, gaussian increments, stationary, ergotic, Brownian scaling.

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:

$$P(r, t) = [2\pi\delta r_{norm}^2(t)/d]^{-\frac{d}{2}} e^{\frac{-r^2 d}{2\delta r_{norm}^2(t)}} \quad (2.1)$$

2.2 Fractional Brownian Motion

Als Ausgangspunkt wird weiterhin die Differentialgleichung $\partial R(t) = \eta(t)$ 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:

$$Z(\omega) = K_\alpha \Gamma(1 + \alpha)(iz)^{1-\alpha} \text{ für} \quad (2.2)$$

- K_α = generalisierter Diffusions-Koeffizient
- ω = Frequenz

[2]

2.3 Algorythm

Für den im Verlauf verwendeten Integrator muss davon ausgegangen werden, dass keine weiteren Interaktion (Potentiale) auftreten. Da zur Erstellung der Trajekto-

rie 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 Theorieteil müssen für den Algorithmus in eine diskrete Form gebracht werden.

Das Inkrement ist:

$$\boldsymbol{\eta}(t) \longrightarrow \boldsymbol{\eta}_j \text{ mit } j = (0, 1, 2, \dots, n) \text{ und } n = \text{ Schrittanzahl} \quad (2.3)$$

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:

$$\Delta \mathbf{R}_n = \sum_{j=0}^n \boldsymbol{\eta}_j \Delta t \quad (2.4)$$

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.

$$< \Delta \mathbf{R}_{fbr} > = 2dK_\alpha t^\alpha \quad (2.5)$$

Der verwendete Algorithmus geht wie folgt:

1. Es werden $2n$ unabhängige Normalverteilte, mit dem Mittelwert $< \boldsymbol{\eta}_j > = 0$ und der Standardabweichung $\delta \boldsymbol{\eta}_j = \sqrt{\Delta t}$, Zufallszahlen erstellt:

$$\boldsymbol{\eta}_j = (\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \dots, \boldsymbol{\eta}_{2n}) = \mathcal{N}(0, \sqrt{\Delta t}) \text{ mit } j = (0, 1, 2, \dots, 2n) \quad (2.6)$$

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

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

$$\boldsymbol{\eta}_g = \sum_{j=0}^{2n-1} \boldsymbol{\eta}_j e^{-\frac{i2\pi j g}{2n}} \text{ mit } g = (0, 1, 2, \dots, 2n) \quad (2.7)$$

$$(2.8)$$

Die entsprechende analytische Fouriertransformation ist:

$$\boldsymbol{\eta}(\omega) = \int_0^\infty e^{-i2\pi\omega t} \boldsymbol{\eta}(t) dt \quad (2.9)$$

$$\text{mit } \omega = g\Delta\omega, \Delta\omega = \frac{1}{2n\Delta t} \text{ und } t = j\Delta t \quad (2.10)$$

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

$$\boldsymbol{\eta}_{fbr_g} = \boldsymbol{\eta}_g \sqrt{2\operatorname{Re}(Z_g)} \quad (2.11)$$

$$Z(\omega) = K_\alpha \Gamma(1 + \alpha)(-\iota\omega)^{1-\alpha} \quad (2.12)$$

$Z(\omega)$ 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 2.9), 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 2.10) in eine diskrete Form gebracht.

$$\rightarrow Z_g = K_\alpha \Gamma(1 + \alpha)(i2\pi g\Delta\omega)^{1-\alpha} = K_\alpha \Gamma(1 + \alpha)(ig \frac{\pi}{n\Delta t})^{1-\alpha} \quad (2.13)$$

4. Für gebrochen-rationale Brownischen Bewegung ist die Auto-Korrelations Funktion an der Stelle 0: $Z_{g=0} = 0$. Daraus folgt nach Formel 2.11, dass auch das Nullte Inkrement im Frequenzraum $\boldsymbol{\eta}_{fbr_{g=0}} = 0$ ist. Das Nullte Inkrement im Frequenzraum ist mit Formel 2.4 aber auch:

$$\boldsymbol{\eta}_{fbr_{g=0}} = \sum_{j=0}^{2n-1} \boldsymbol{\eta}_j e^0 = \frac{\Delta \mathbf{R}}{\Delta t} \quad (2.14)$$

$\Delta \mathbf{R}$ 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:

$$\boldsymbol{\eta}_{fbr_{g=0}} = \mathcal{N}(0, \sqrt{2K_\alpha(2n\Delta t)^\alpha}) \quad (2.15)$$

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:

$$\eta_{fbr_j} = \frac{1}{2n} \sum_{g=0}^{2n-1} \eta_g e^{\frac{2\pi i j g}{2n}} \quad (2.16)$$

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.

2.3.1 Analyse of Algorythm

Hier kommen die Plots aus dem Analyse Skript rein.

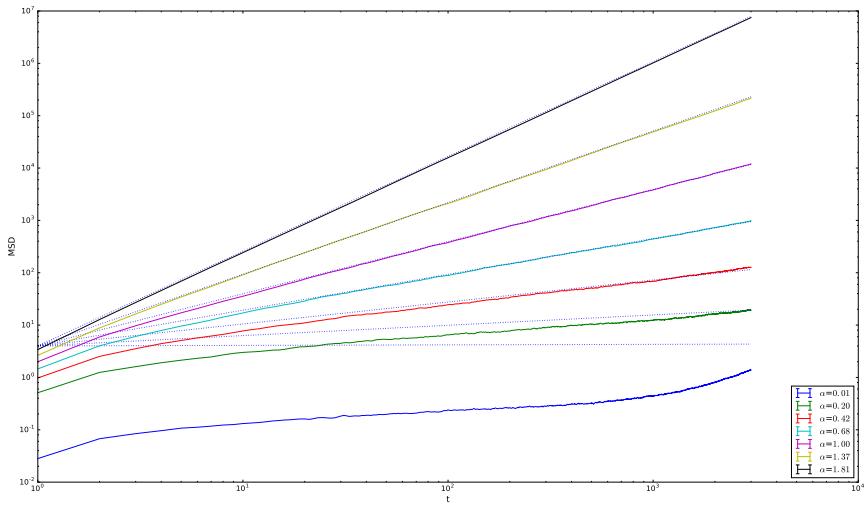
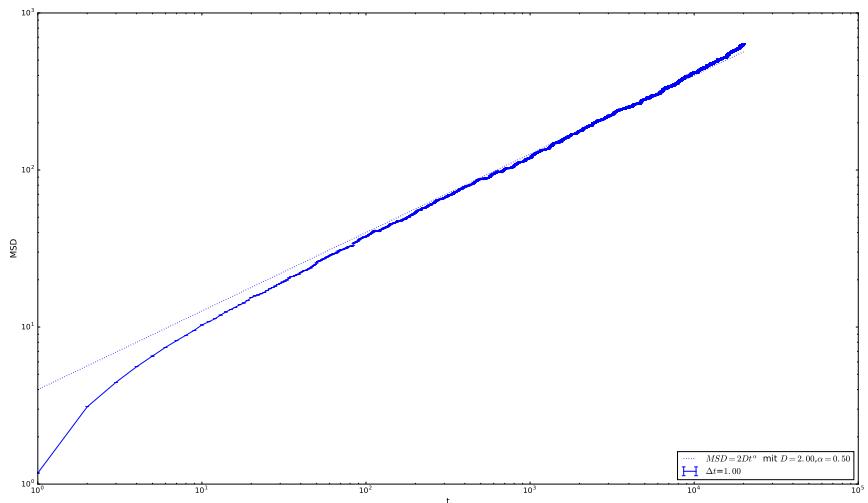
Figure 2.1: Mean Square Displacement of different α 

Figure 2.2: Mean Square Displacement (Ensemble average of 4000 Trajectories)

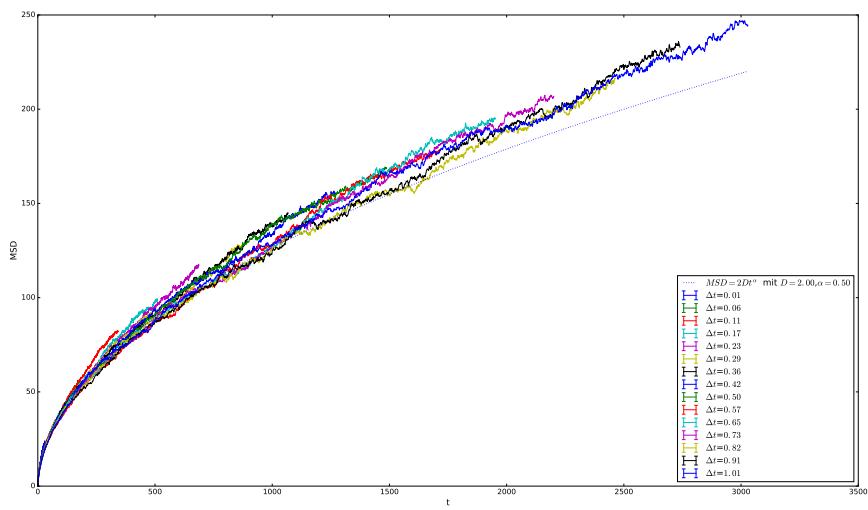


Figure 2.3: Mean Square Displacement (Ensemble average fpr different Δt)

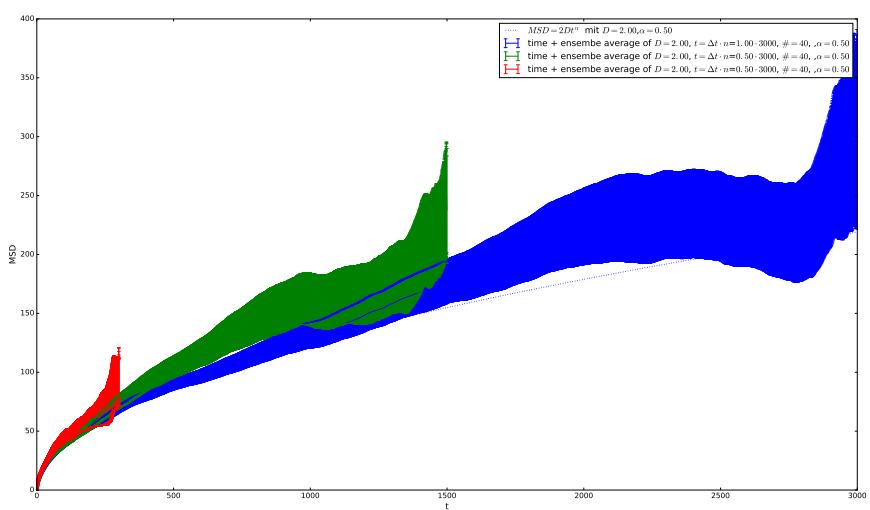


Figure 2.4: Mean Square Displacement (time and ensemble avarge for different Δt and with standard deviation of the Mean)

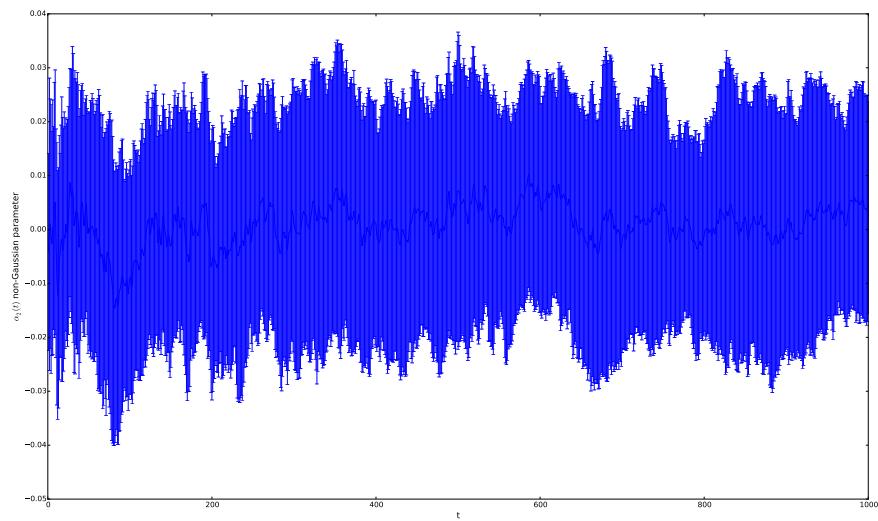


Figure 2.5: Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.

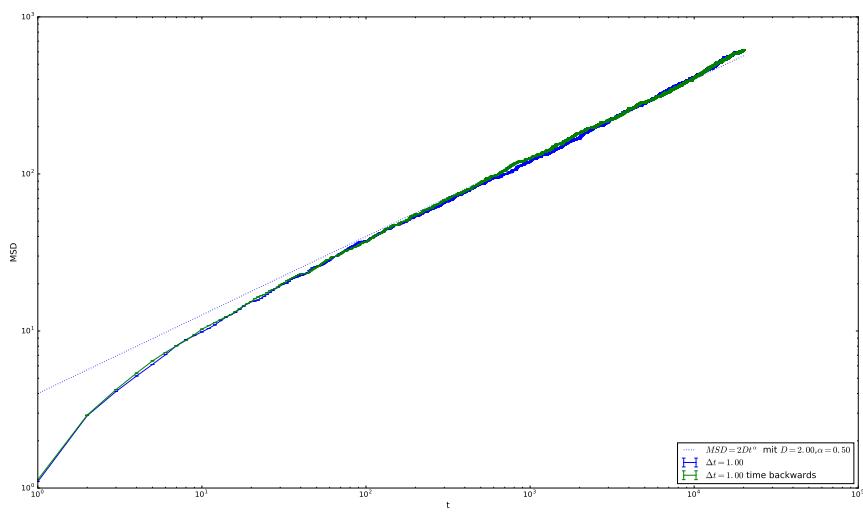


Figure 2.6: Non-gaussian parameter of $\alpha = 0.5$, $D = 2$ and 5000 trajectories ensemble.

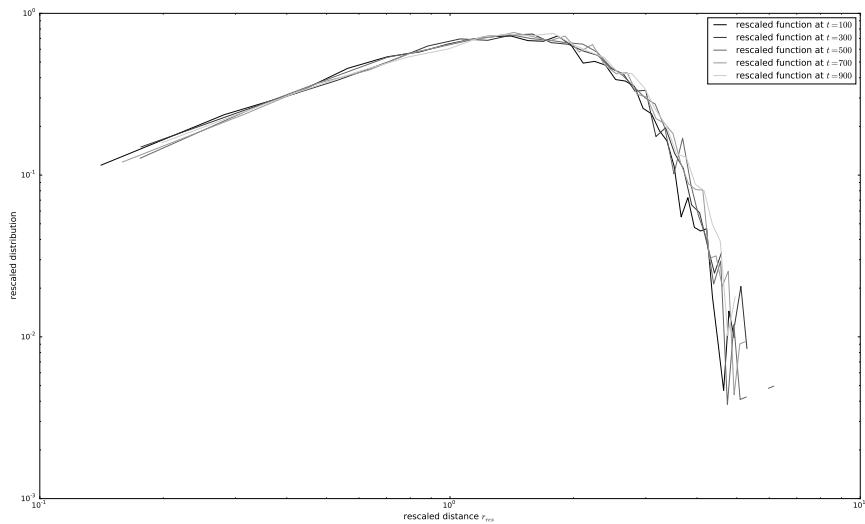


Figure 2.7: rescaled function for different times

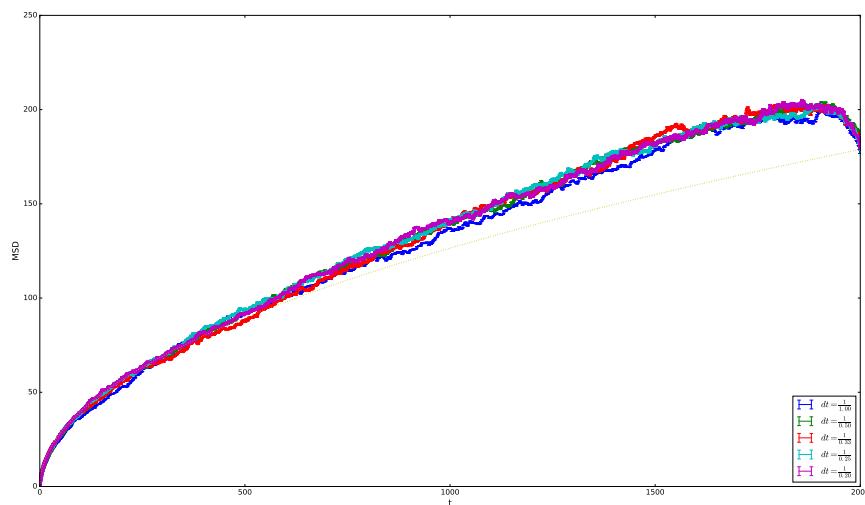


Figure 2.8: variere dt bei gleichem Faktor 1

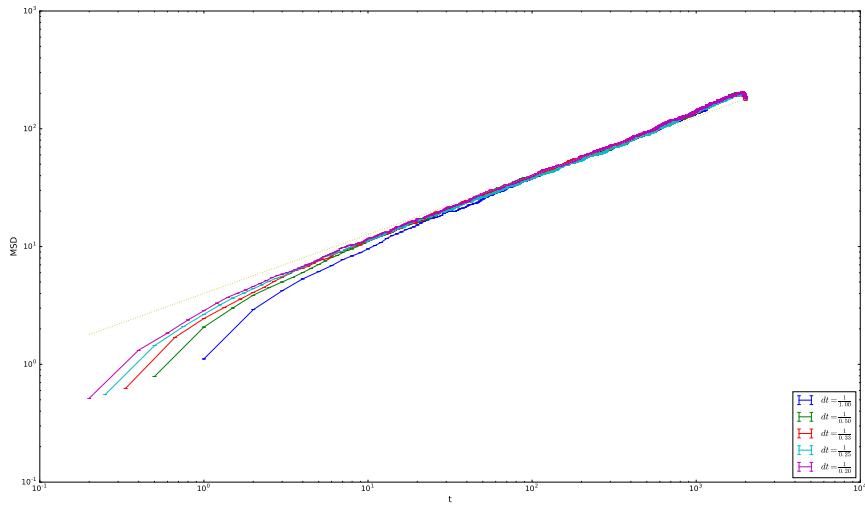


Figure 2.9: variere dt bei gleichem Faktor 1, logarithmisch

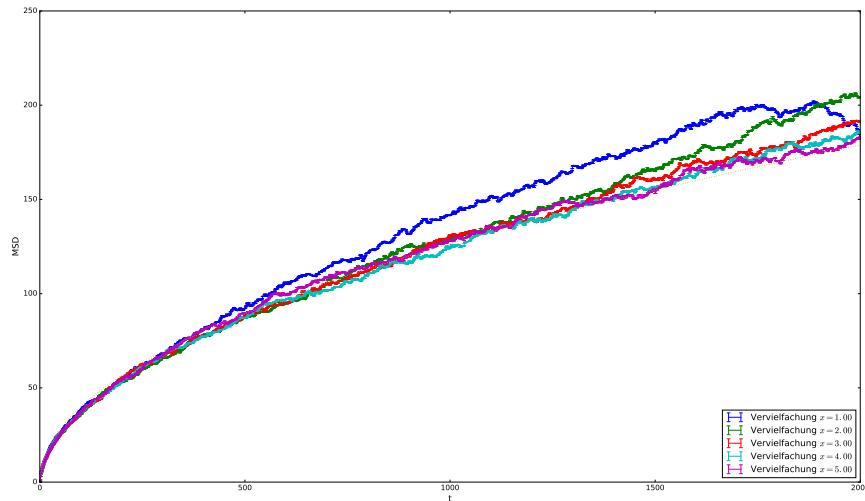


Figure 2.10: variere Faktor bei gleichem dt

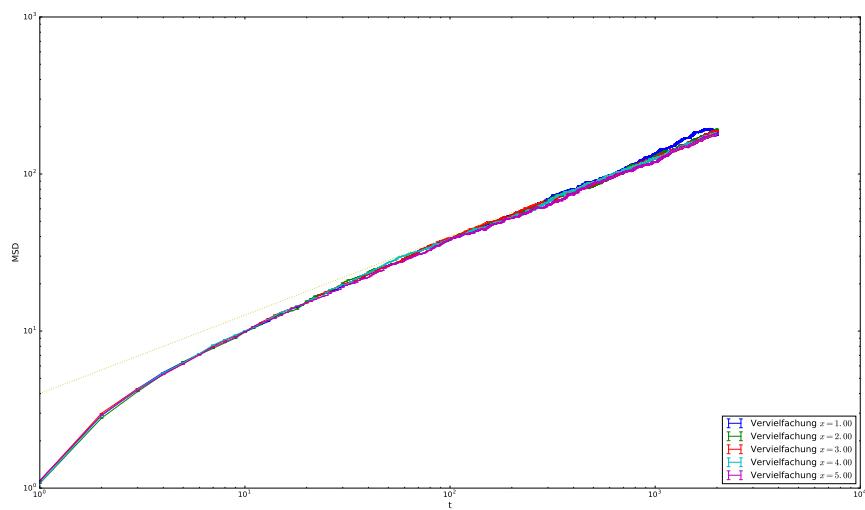


Figure 2.11: variere Faktor bei gleichem dt, logarithmisch

3 Reaktions-Diffusions-Dynamics

4 Appendix

4.0.2 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 import test_cython.genereatefracincrements as ginc
7
8
9 class Felix_Method():
10
11     def __init__(self, D, particles, length, alpha, dt, x=2, version="python"):
12         self.D=D
13         self.K_alpha=D
14         self.particles=particles
15         self.n=length
16         self.alpha=alpha
17         self.ki=self.n*x
18         self.frq=np.fft.freq(np.pi*2.)/(dt))
19         self.t=np.array(range(length))
20         self.dt=dt
21         self.version=version
22
23     def z(self):
24         """
25             :param D: Diffusionskoeffizient
26             :param alpha: Anomalieparameter
27
28
29             :return: z: Correlationsfunktion im Frequenzraum welche auf die
30             gew hnliche diffusion angewendet wird.
31
32
33
```

```

35     """
36     z=((((+1j* self . frq ))** (1- self . alpha ))* self . K_alpha *np . math .
37     gamma(1+ self . alpha ))
38     return z
39
40     def compute_trajectory ( self ):
41         """
42             :param D: Diffusionskoeffizient
43             :param particles: Anzahl an Teilechen welche simuliert werden
44             sollen .
45
46             :param length: L nge der Trajektorien , welche simuliert werden
47             sollen .
48
49             :param alpha: Anomalieparameter
50
51             :return: ( Particles x Length ) Array of Trajectories of all
52             particles. Close to :cite:`Graigmile2003`"
53
54         """
55
56         #plt . plot ( self . frq , "r")
57         #plt . plot ( np . arange (-np . pi / self . dt , np . pi / self . dt ,( np . pi /( self . n* self . dt ))))
58         #plt . show ()
59         r_t_allparticles = []
60         if self . version == "cpp":
61             inc = ginc . pyIncrements ( self . n, self . particles )
62             inc . generateIncrements ( self . D, self . dt , self . alpha )
63             a = inc . returnIncrements ()
64             r_t = np . cumsum ( a [:, 0 ,:], axis = 1)
65             r_t [:, 0 ] = 0
66             return r_t
67         if self . version == "python":
68             for particle in range ( self . particles ):
69                 #r = np . random . RandomState (1234)
70                 v_t =(np . random . normal (0,np . sqrt ( self . dt ), size = self . ki ))
71                 #todo mit matrix . shape k nnen man eine zuf llige verteilung von
72                 #allen daten generieren
73                 v_t = np . array ( v_t )
74                 v_fraq = np . fft . fft ( v_t )
75                 v_ano_fraq = np . sqrt ( self . z () . real * 2.) * v_fraq
76                 v_ano_fraq [0] = np . random . normal (0,np . sqrt ( 2.* self . K_alpha
77                 *( self . ki * self . dt )** self . alpha ))

```

```

73         #v_ano_frq [ self .n]=np .sqrt ( self .z () [ self .n]. real* self .n
    *2)* v_t [ self .ki-1]
    #v_ano_frq [ self .n-1]=np .sqrt ( self .z () [ self .n]. real* self .
    .n*2)* v_t [ self .ki-1]
    v_ano_t=np .fft .ifft ( v_ano_frq )
    #r_t1=np .cumsum( v_ano_t [: self .n]. real) #Ort bei
    anomaler Diffusion in Abh ngigkeit von der zeit
    r_t=np .zeros( self .n)
    r_t [1:]=np .cumsum( v_ano_t [: self .n]. real) [1:] #Ort bei
    anomaler Diffusion in Abh ngigkeit von der zeit
    r_t_allparticles.append(r_t) # Trajektorie bei anomaler
    Diffusion f r alle teilchen
    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.