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

List of Figures

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 Gebrochenrationale 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 Zeitinvervalls 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 Quadratische Verschiebung:
$\delta r^2_{norm}(t) =< \Delta R(t)_{Br} >= 2dDt$ 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:

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

1.2 Gebrochen-rationale Brownische Bewegung

Es wurde eine festgestellt, dass für eine Systeme die Mittlere Quadratische Verschiebung nicht mit der für Brownische Bewegung übereinstimmt, aber statt dessen eine exponentielle Abhängigkeit zur Zeit besitzt. $\delta r^2_{fBr}(t) =< \Delta R(t)_{fBr} >= 2dK_\alpha t^\alpha$ Dieses Phänomen wurde als Gebrochen-rationale Brownische Bewegung definiert. Für dem mittlere quadratische Verschiebung gilt allgemein $\delta r^2_{fBr}(t) = \int_0^t dt' v(t')$ ... . Die Geschwindigkeitsautokorrelationsfunktion für Gebrochen-rationale Brownische Bewegung kann wie folgt definiert werden:

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

- K_α = generalisierter Diffusions-Koeffizient
- ω = Frequenz (die Variable nach der nach der Fouriertransformation)

[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 Theorieteil müssen für den Algorithmus in eine diskrete Form gebracht werden.
Das Inkrement ist:

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

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:

$$\Delta \boldsymbol{R}_n = \sum_{j=0}^{n} \boldsymbol{\eta}_j \tag{1.4}$$

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.

$$< \Delta \boldsymbol{R}_{fbr} >= 2dK_\alpha t^\alpha \tag{1.5}$$

Der verwendete Algorithmus geht wie folgt:

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

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

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

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

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

$$\tag{1.8}$$

   Die entsprechende analytische Fouriertransformation ist:

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

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

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

$$\boldsymbol{\eta_{fbr_g}} = \boldsymbol{\eta_g} \sqrt{2Re(Z_g)} \tag{1.11}$$

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

   $Z(\omega)$ 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 $Z(\omega) \longrightarrow Z_g$ (nach Formel 1.10) in eine diskrete Form gebracht.

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

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 $\boldsymbol{\eta_{fbr_{g=o}}} = 0$ ist. Das Nullte Inkrement im Frequenzraum ist mit Formel 1.4 aber auch:

$$\boldsymbol{\eta_{fbr_{g=o}}} = \sum_{j=0}^{2n-1} \boldsymbol{\eta_j} e^0 = \Delta \boldsymbol{R} \tag{1.14}$$

$\Delta\boldsymbol{R}$ ist in dem Fall die zurückgelegt 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:

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

gerechnet. Dies wäre korrekt, wenn gebrochen-rationalen 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 die Abweichung 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:

$$\boldsymbol{\eta_{fbr_j}} = \frac{1}{2n} \sum_{g=0}^{2n-1} \boldsymbol{\eta_g} e^{\frac{2\pi ijg}{2n}} \tag{1.16}$$

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

Für die drei dimensionale Gebrochen-rationale Brownische Bewegung kann für jede kartesische Komponente der soeben beschriebene Algorithmus verwendet werden.

1.3.1 Analyse des Algorithmuses

Hier kommen die Plots aus dem Analyse Skript rein.

2 Reaktions-Diffusions-Dynamik

3 Appendix

3.0.2 Code:Gebrochen-rationale Brownische Bewegung

```
# -*- coding: utf-8 -*-
__author__ = 'janek'
import numpy as np # module for scientific computing
from scipy import integrate
import matplotlib.pyplot as plt # module for plotting "a la" matlab

class Felix_Method():

    def __init__(self, D, particles,length,alpha,dt):
        self.D=D
        self.K_alpha=D
        self.particles=particles
        self.n=length
        self.alpha=alpha
        self.ki=self.n*2
        self.frq=np.fft.fftfreq(self.ki)*(np.pi*2./(dt))
        self.t=np.array(range(length))
        self.dt=dt

    def z(self):
        """
        :param D: Diffusionskoeffizient

        :param alpha: Anomalieparameter



        :return: z: Correlationsfunktion im Frequenzraum welche auf die
    gew hnliche diffusion angewendet wird.


        """
        z=((((+1j*self.frq))**(1-self.alpha))*self.K_alpha*np.math.
    gamma(1+self.alpha))*np.exp(np.pi/self.ki)
        return z
```

```
    def compute_trajectory(self):
        """
        :param D: Diffusionskoeffizient
        :param particles: Anzahl an Teilechen welche simuliert werden sollen.

        :param length: L nge der Trajektorien, welche simuliert werden sollen.

        :param alpha: Anomalieparameter

        :return: ( Particles x Length ) Array of Trajectories of all particles. Close to :cite:`Graigmile2003`


        """
        r_t_allparticles=[]
        #plt.plot(self.frq,"r")
        #plt.plot( np.arange(-np.pi/self.dt , np.pi/self.dt,(np.pi/(self.n*self.dt))))
        #plt.show()
        for particle in range(self.particles):
            v_t=(np.random.normal(0,np.sqrt(self.dt),size=self.ki)) # todo mit matrix.shape k nnen man eine zuf llige verteilung von allen daten generieren
            v_t=np.array(v_t)
            v_frq=np.fft.fft(v_t)
            v_ano_frq= np.sqrt(self.z().real*2.)*v_frq
            v_ano_frq[0]=np.random.normal(0,np.sqrt(2.*self.K_alpha*(self.ki*self.dt)**self.alpha))
            #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=integrate.cumtrapz(v_ano_t[:self.n].real,initial=0) # 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.