Theoretical Biophysics
A Computational Approach
Concepts, Models, Methods and Algorithms
Diffusion and Transport

Dieter W. Heermann

April 7, 2020

Heidelberg University

Table of Contents

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Introduction

Introduction I

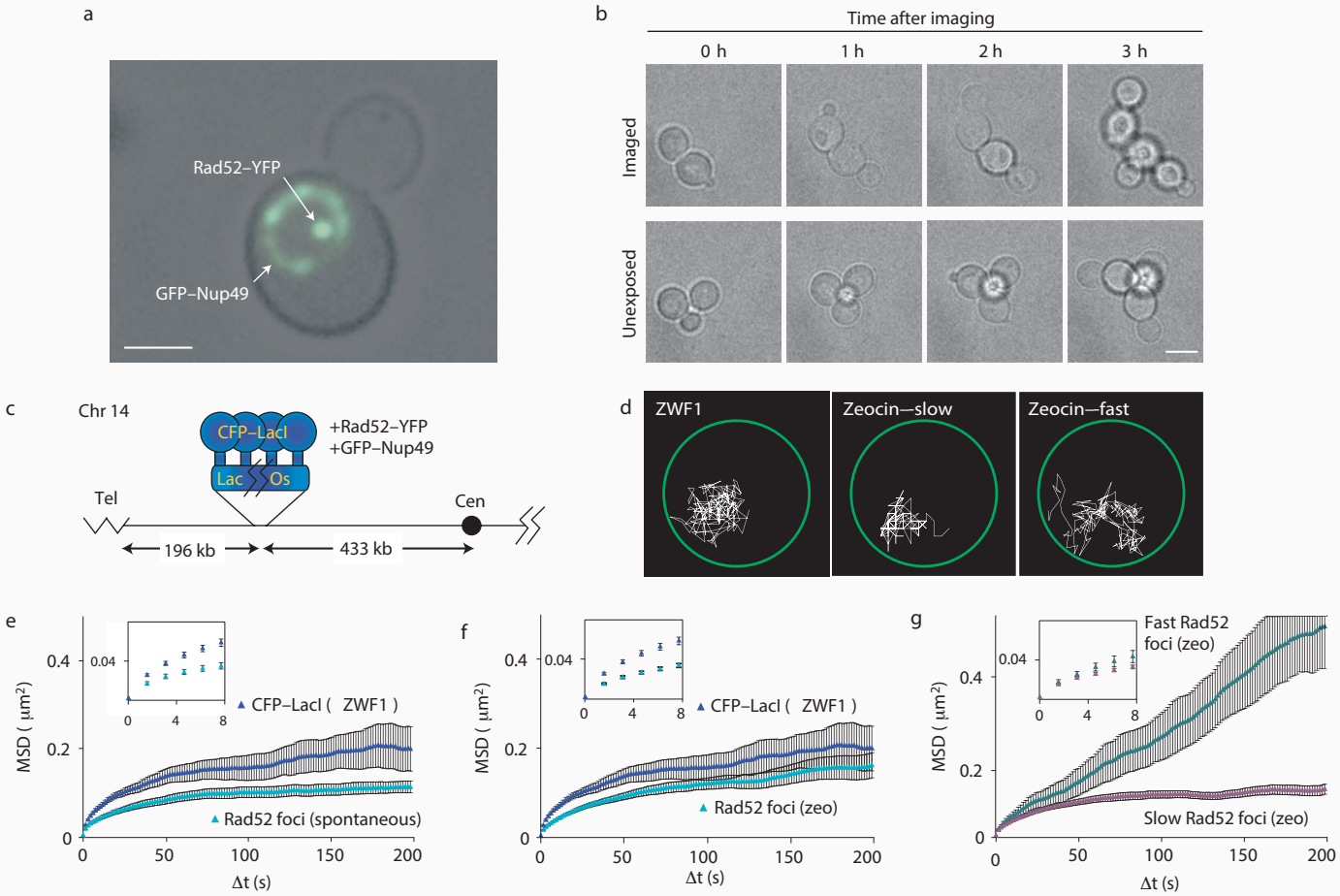


Figure 1: Diffusion of the ends of DNA after a double strand break. Image taken from [1].

Introduction II

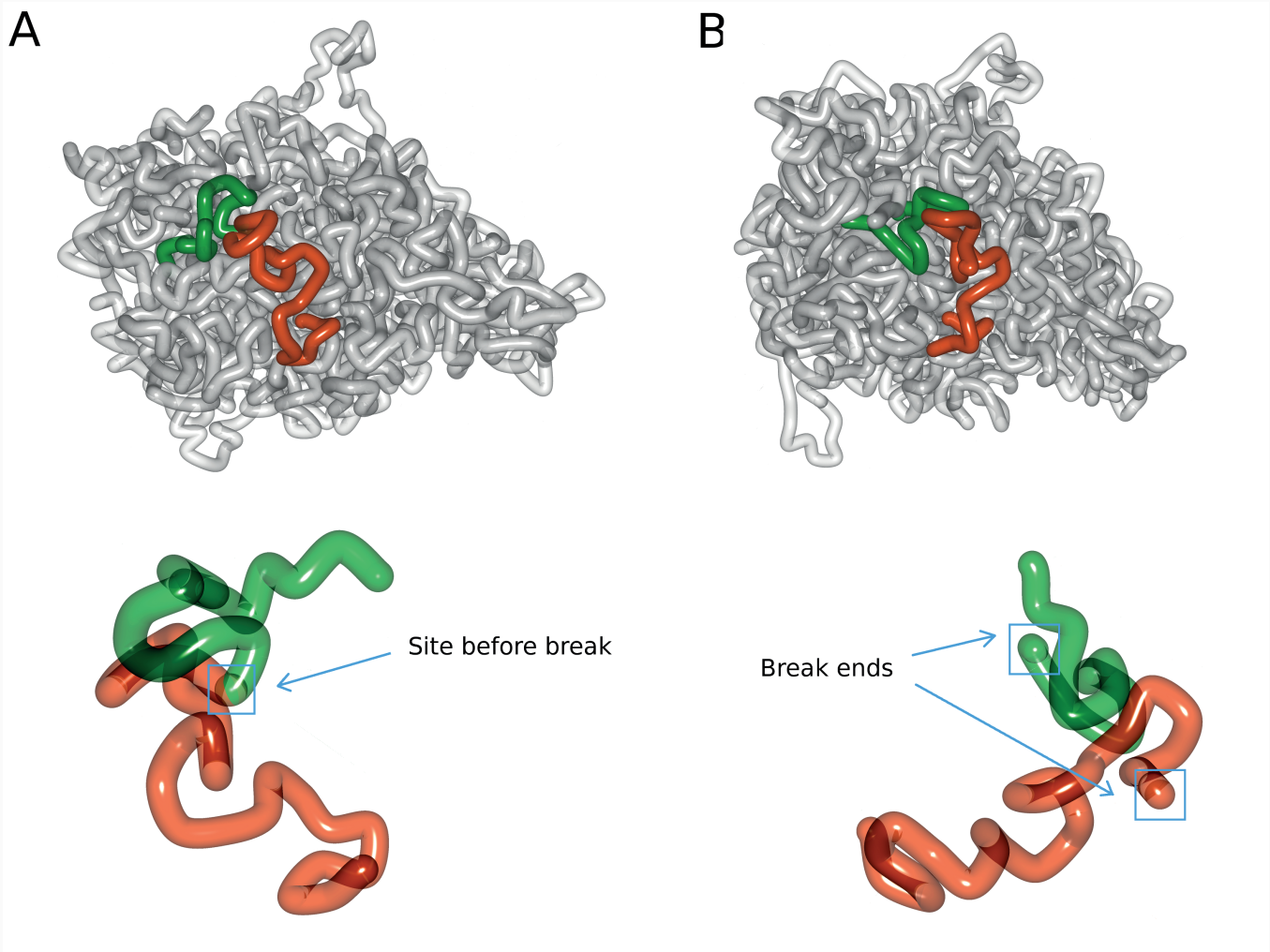


Figure 2: Model for the diffusion of the ends of DNA after a double strand break. Image taken from [2].

Introduction III

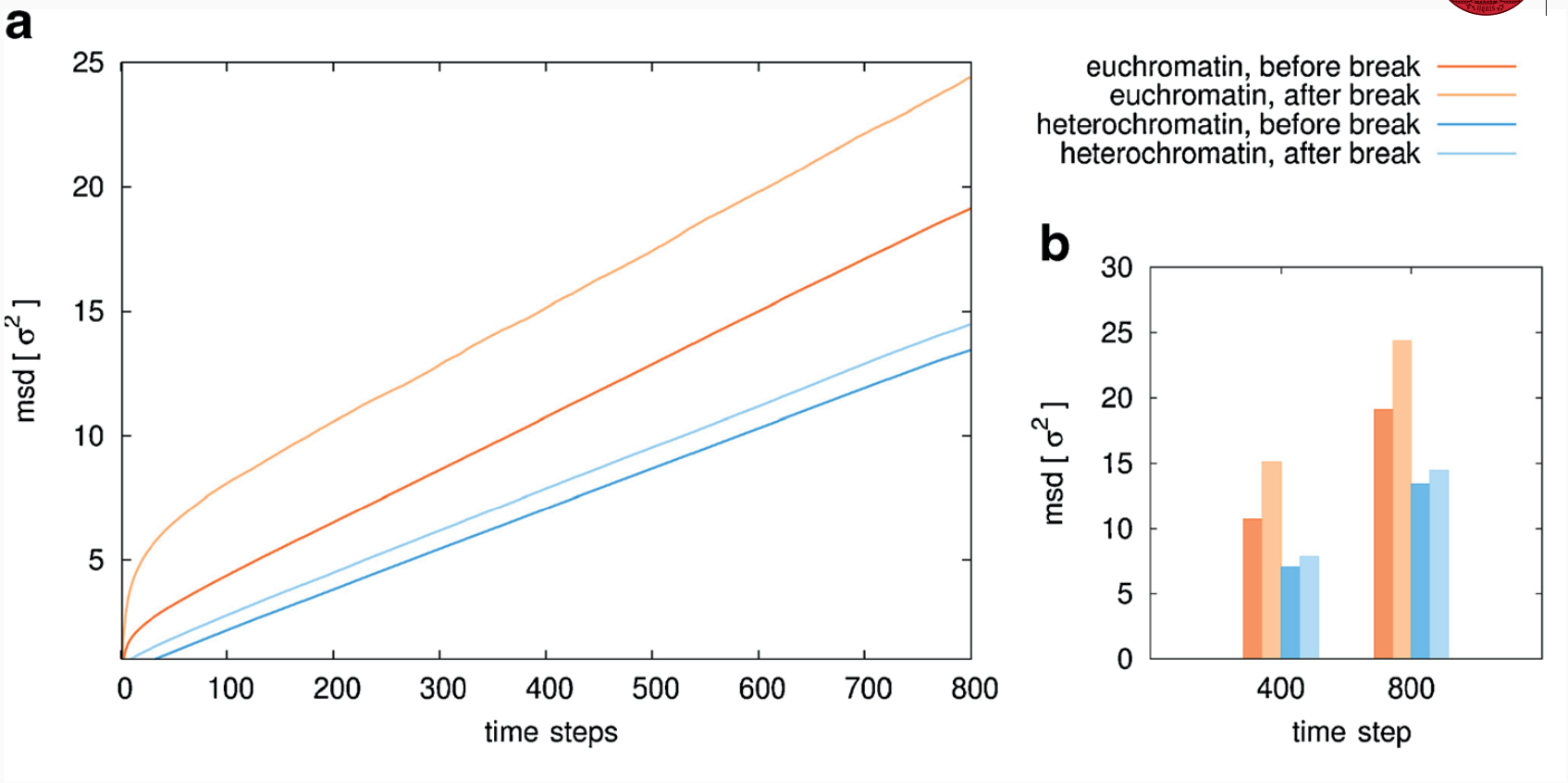


Figure 3: Results from the model for the diffusion of the ends of DNA after a double strand break. Image taken from [2].

Introduction IV

The diffusion and random walk problem was formulated as follows:

- A man starts from a point O and walks l yards in a straight line; he then turns through any angle whatever and walks another l yards in a second straight line. He repeats this process N times. I require the probability that after N of these stretches he is at a distance between r and $r + \delta r$. [3, 4].

A realization of such a process is shown below

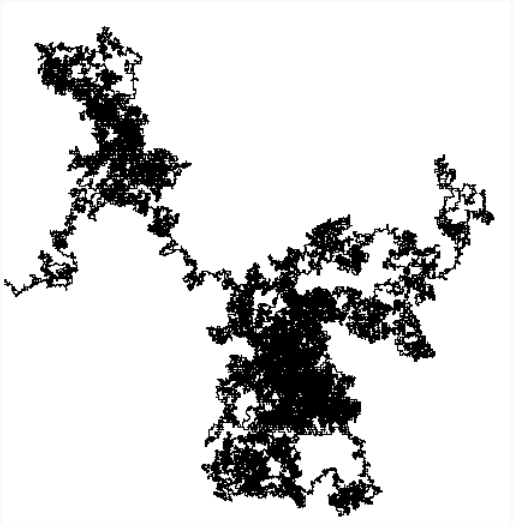


Figure 4: Snapshot of a random walk in two dimensions.

Introduction V

Lord Rayleigh pointed out the connection between this problem and an earlier paper of his published in 1880 concerned with sound vibrations. Rayleigh pointed out that, for large values of N, the answer is given by

$$\frac{2}{Nl^2} e^{-r^2/Nl^2} r\delta r \ .$$

where r is the distance and l sets the scale.

1919-21 the lattice random walk or Polya walk was introduced by George Polya [4]. Let

$$\mathbb{Z}^d = \{(x^1, x^2, ..., x^d) | x^i \in \mathbb{Z}\} \tag{1}$$

denote a d-dimensional integer lattice. Here the superscript denotes the components. Also the notation (x, y, z) etc. will be used. We will use subscripts to enumerate the elements of $\mathbb{Z}^d$: $x_1, x_2, ..., x_N$, where $x_i = (x_i^1, x_i^2, ..., x_i^d)$. Consider a starting point X on the lattice. The process described above for N steps could be described by

$$S_N = X + X_1 + X_2 + ... + X_N \tag{2}$$

where the X_i are elements of $\mathbb{Z}^d$ and independent and identically distributed (iid) random variables.

Let e_k be the vector where all elements are 0 except for the k-th element which is 1. Then $(e_1, ..., e_d)$ is a basis for $\mathbb{Z}^d$. With this we have

$$P(X_i = e_k) = P(X_i = -e_k) = \frac{1}{2d} \ . \tag{3}$$

We can define the random walk as a **time-homogeneous Markov chain** with state space $\mathbb{Z}^d$ and transition probability

$$p(x, y) := P(S_{n+1} = y | S_n = x) = \frac{1}{2d} \ , \ y - x \in \{e_1, ..., e_d\} \ . \tag{4}$$

Introduction VII

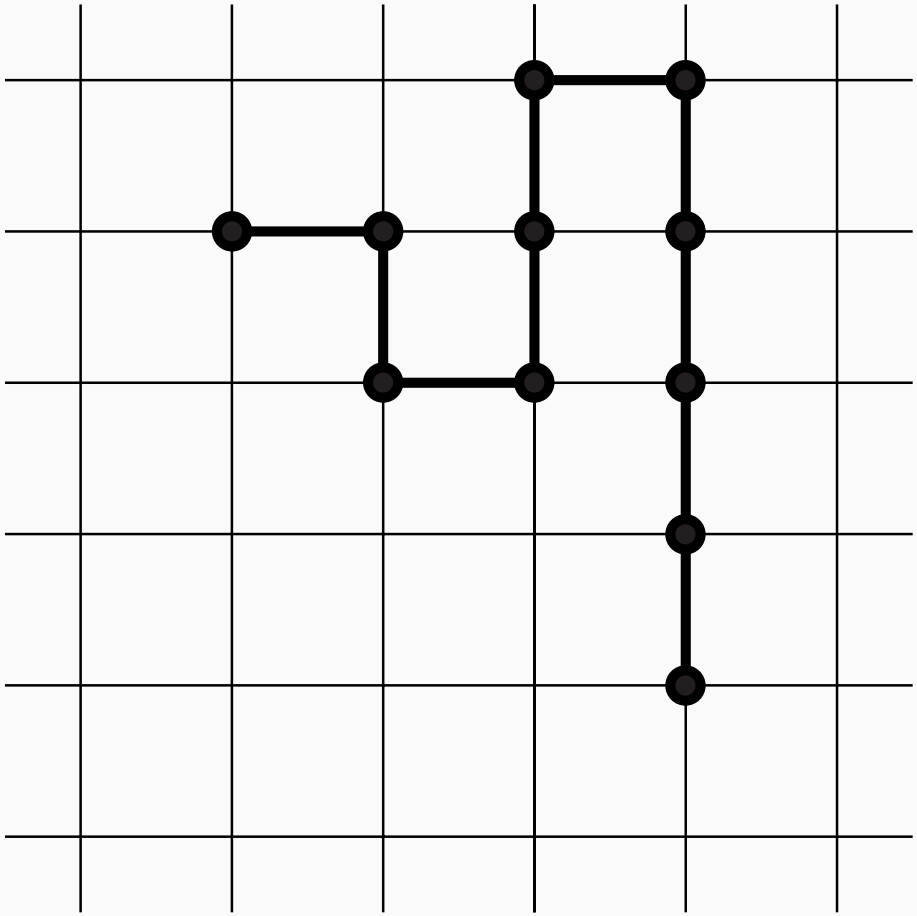


Figure 5: A random walk on a two dimensional lattice.

Diffusion

Diffusion of a Particle I

One approach to diffusion is to start with the master equation. Let us look at how the probability $P(x,t)$ for finding a particle at position x at time t evolves. We assume that the particles needs to change place. Thus the probability changes because there are jumps to the next position $x-a$ and $x+a$ as well was jumps from those positions (one dimensional). Hence there is a flow in and out of the position x

$$\begin{aligned}\frac{\partial}{\partial t}P(x,t) &= P(x+a,t)W(x+a\to x)+P(x-a,t)W(x-a\to x)\\ &- P(x,t)W(x\to x+a)-P(x,t)W(x\to x-a)\,, \end{aligned} \tag{5}$$

where

$$W(x\to x\pm a)=W(x\pm a\to x)=\frac{\Gamma}{2}\,, \tag{6}$$

with Γ being the number f jumps per unit time. Equation 5 is called the **master equation**.

To solve the above equation we can expand $P(x,t)$ in x up to second order

Diffusion of a Particle II

$$P(x \pm a, t) = P(x, t) \pm a \frac{\partial}{\partial x} P(x, t) + \frac{a^2}{2} \frac{\partial^2}{\partial x^2} P(x, t) + \mathrm{O}(\mathrm{a}^3) \ . \tag{7}$$

Inserting this in Equation (5) and retaining only terms up to second order in a we get

$$\frac{\partial}{\partial t} P(x, t) = \Gamma \frac{a^2}{2} \frac{\partial^2 P}{\partial x^2} \ . \tag{8}$$

Let

$$D = \Gamma \frac{a^2}{2} \ . \tag{9}$$

Assuming a symmetry of the lattice Λ we obtain

$$\frac{\partial}{\partial t} P(\mathbf{x}, t) = \frac{\Gamma}{6} a^2 \Delta P(\mathbf{x}, t) \ , \tag{10}$$

with the Laplace operator Δ. Note, since the dimensions are not coupled, we have generalized the approach.

Diffusion of a Particle III

We now calculate the second moment of the distribution P yielding the average distance squared that the random walker walked away from the origin, which we take to be at 0

$$\begin{aligned}
\langle x^2 \rangle &= \int_{-\infty}^{\infty} x^2 P(x,t) dx \\
\frac{\partial}{\partial t}\langle x^2 \rangle &= \int_{-\infty}^{\infty} x^2 \frac{\partial}{\partial t} P(x,t) dx \\
&= \frac{a^2}{2}\Gamma \int_{-\infty}^{\infty} x^2 \frac{\partial^2}{\partial x^2} P(x,t) dx \\
&= a^2 \Gamma \int_{-\infty}^{\infty} P(x) dx \\
&= 2D \, . \qquad (11)
\end{aligned}$$

Hence we get

$$\langle x^2 \rangle = 2Dt \, , \qquad (12)$$

i.e. the mean square displacement is proportional to time (**Einstein relation** [5]).

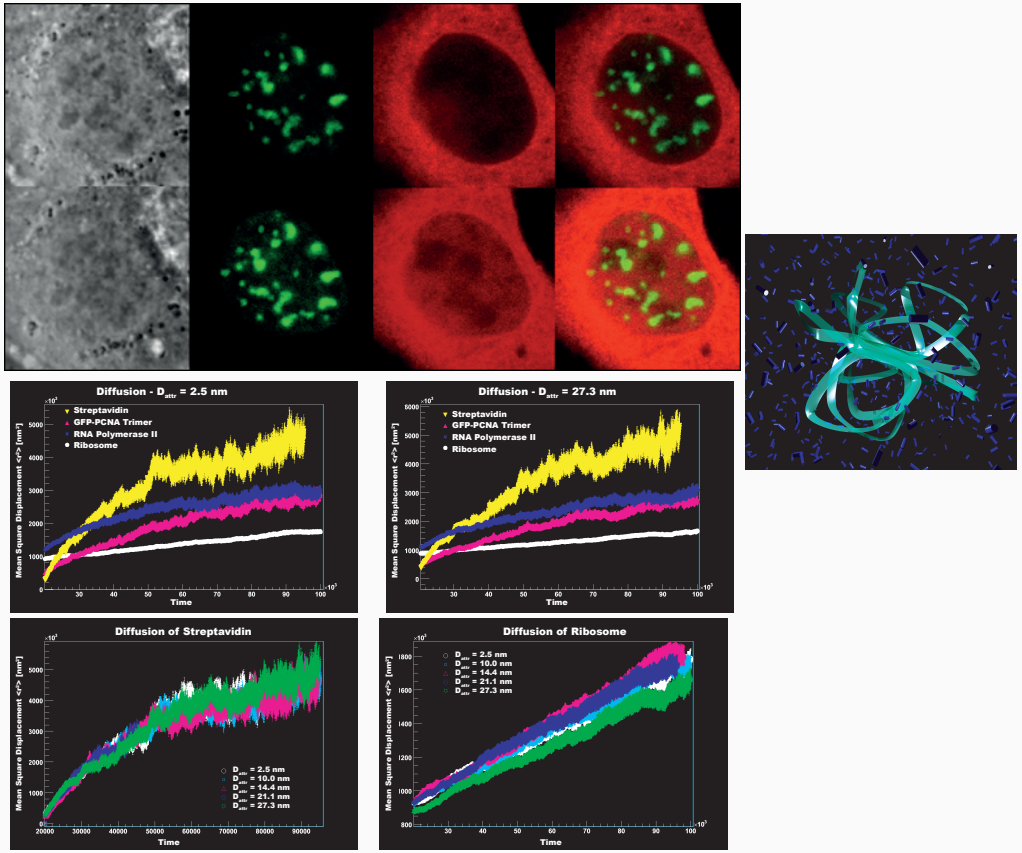


Figure 6: Accessibility and diffusion of transcription factor complexes into 1Mbp chromatin domains. NLS-streptavidin-Cy5 steady state distribution in mouse cells expressing MeCP2-GFP to label heterochromatin. TOP: Panels A) A cell immediately after microinjection, B) The same cell as in A) but 12 min later, the complexes have been transported into the nucleus ,C) A cell with aggregations of MeCP2 labelled centromeric heterochromatin structures. CENTER: Model with 1000 diffusing particles. BOTTOM: Simulation results. Taken from Odenheimer and Heermann, 2004.

Brownian Particle I

Let us now look at the Brownian particle problem [6, 7] and first consider time scales.

- Collision of solvent molecules with mass m and radius a with each other:

$$\tau_c \sim \frac{a}{\sqrt{kT/m}} \,. \tag{13}$$

- Momentum relaxation time τ_m for $t \gg \tau_c$:

  Assuming that the particle has an initial velocity v_0, what is the time for the particle to be at velocity 0 due to friction. The viscous force on the particle is $fv(t)$ thus (Stoke's law for a particle in fluid)

$$\frac{dp}{dt} = -fv(t) \tag{14}$$

  and hence

$$\tau_m = \frac{m}{f} \,. \tag{15}$$

The diffusive or Brownian time scale is thus $t \gg \tau_m$. Thus the net displacement becomes independent of its mass.

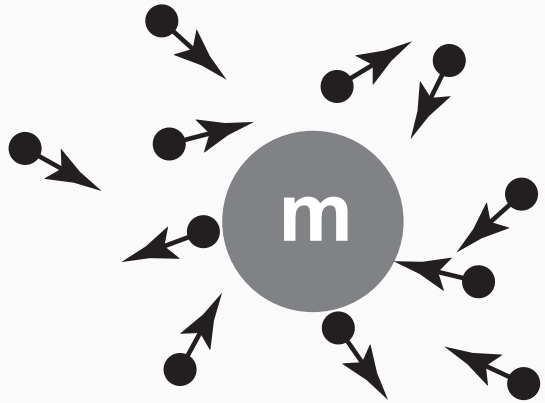


Figure 7: Brownian particle model visualization

The equation of a Brownian particle can hence be written as the effect of the systematic force F and a random force ξ acting of the particle by the collisions

$$m\frac{dv}{dt} = F + \sigma\xi(t) \tag{16}$$

with

$$\xi(0) = 0 \tag{17}$$
$$\langle \xi(t) \rangle = 0 \tag{18}$$
$$\langle \xi(t)\xi(t') \rangle = \delta(t - t') \, . \tag{19}$$

Hence the total force on the particle fluctuates with a typical autocorrelation time τ.

Considering again the special case of particle in a fluid we can write the above equation as the **Langevin equation**

$$\frac{dv(t)}{dt} = -\gamma v(t) + \sigma \xi(t) \tag{20}$$

with γ being the Stoke's coefficient and

$$\sigma^2 = \frac{2kT\gamma}{m} \tag{21}$$

the diffusion coefficient.

Let us rewrite Equation 20 as

$$\frac{dv(t)}{dt} = -\frac{\gamma}{m}v(t)dt + \frac{1}{m}dW(t) \tag{22}$$

where $dW(t) = \xi(t)dt$. Consider

$$\int_0^t f(s)dv(s) = -\frac{\gamma}{m}\int_0^t f(s)v(s)ds + \frac{1}{m}\int_o^t f(s)dW(s) \tag{23}$$

with arbitrary function $f(t)$. Assume $f(t) = 1$, then

$$\begin{aligned} v(t) - v(0) &= -\frac{\gamma}{m}\int_0^t v(s)ds + \frac{1}{m}[W(t) - W(0)] && (24)\\ &= -\frac{\gamma}{m}[x(t) - x(0)] + \frac{1}{m}[W(t) - W(0)] && (25) \end{aligned}$$

Divide the time t into intervals $[t_k, t_{k+1}]$, $t_0 < t_1 < \cdots < t_n$. Then

$$W(t) - W(0) = \sum_{k=1}^{n}(W(t_k) - W(t_{k-1}))\ . \tag{26}$$

Brownian Particle V

We may assume that W has the Markov property as the particle is subject to many random hits from the environment particles. Further, the average (see above) is assumed to be zero. Then

$$W(t) = W(0) + \int_0^t \xi(s)ds \tag{27}$$

will be a continuous function. Since we can make the intervals small due to the high frequency collisions, W(t) is a Markov process. Note further that

$$W(t_1) - W(t_0), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1}) \tag{28}$$

are independent, stationary with zero mean and iid. Hence W must be Gaussian (**Wiener process**).

What we need to numerically treat the Brownian particle problem is a random number generator delivering gaussian random numbers. One possibility is the Box-Muller algorithm. The foundation for this method is the transformation theorem

Theorem

Let X be a random variable with density function f and distribution F. Let

$$h : S \to B$$

a differentiable strong monotone function with $S, B \subseteq \mathbb{R}$.

- *$Y := h(X)$ is a random variable. The distribution of Y is*

$$F(h^{-1}(y)) \qquad \text{for monotonically increasing } h \tag{29}$$

$$1 - F(h^{-1}(y)) \qquad \text{for monotonically decreasing } h \tag{30}$$

- *If h^{-1} is absolutely continuous for almost all y then*

$$f(h^{-1}(y)) \left| \frac{d^{-1}(y)}{dy} \right|$$

is the density of $h(X)$.

As an application we choose $S = [0,1]^2$ and the density $f = 1$ on S. The transformation we choose as

$$\begin{cases} h_1(x_1, x_2) := y_1 & = \sqrt{-2\log x_1}\cos(2\pi x_2) \\ h_2(x_1, x_2) := y_2 & = \sqrt{-2\log x_1}\sin(2\pi x_2) \end{cases} \tag{31}$$

with the inverse h^{-1}

$$\begin{cases} x_1 = \exp\{-\frac{1}{2}(y_1^2 + y_2^2\} \\ x_2 = \frac{1}{2\pi}\arctan\frac{y_2}{y_1} \end{cases} \tag{32}$$

and thus

$$\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} = -\frac{1}{2\pi}\exp\left\{-\frac{1}{2}(y_1^2 + y_2^2)\right\} . \tag{33}$$

Algorithm 1 Box-Muller Algorithm

1: generate $X_1 \in [0,1]$ uniform and $X_2 \in [0,1]$ uniform
2: $\theta = 2\pi X_2$ and $q = \sqrt{-2\log X_1}$
3: $Y_1 = q\cos\theta$ and $Y_2 = q\sin\theta$

Below is an implementation of this algorithm in Python

```
1
  w = -math.log(random.random())
3 phi = (random.random()-0.5)*math.pi
  x = 2*math.sqrt(w) * math.sin(phi)
```

Code 1: Box-Muller algorithm for gaussian random numbers

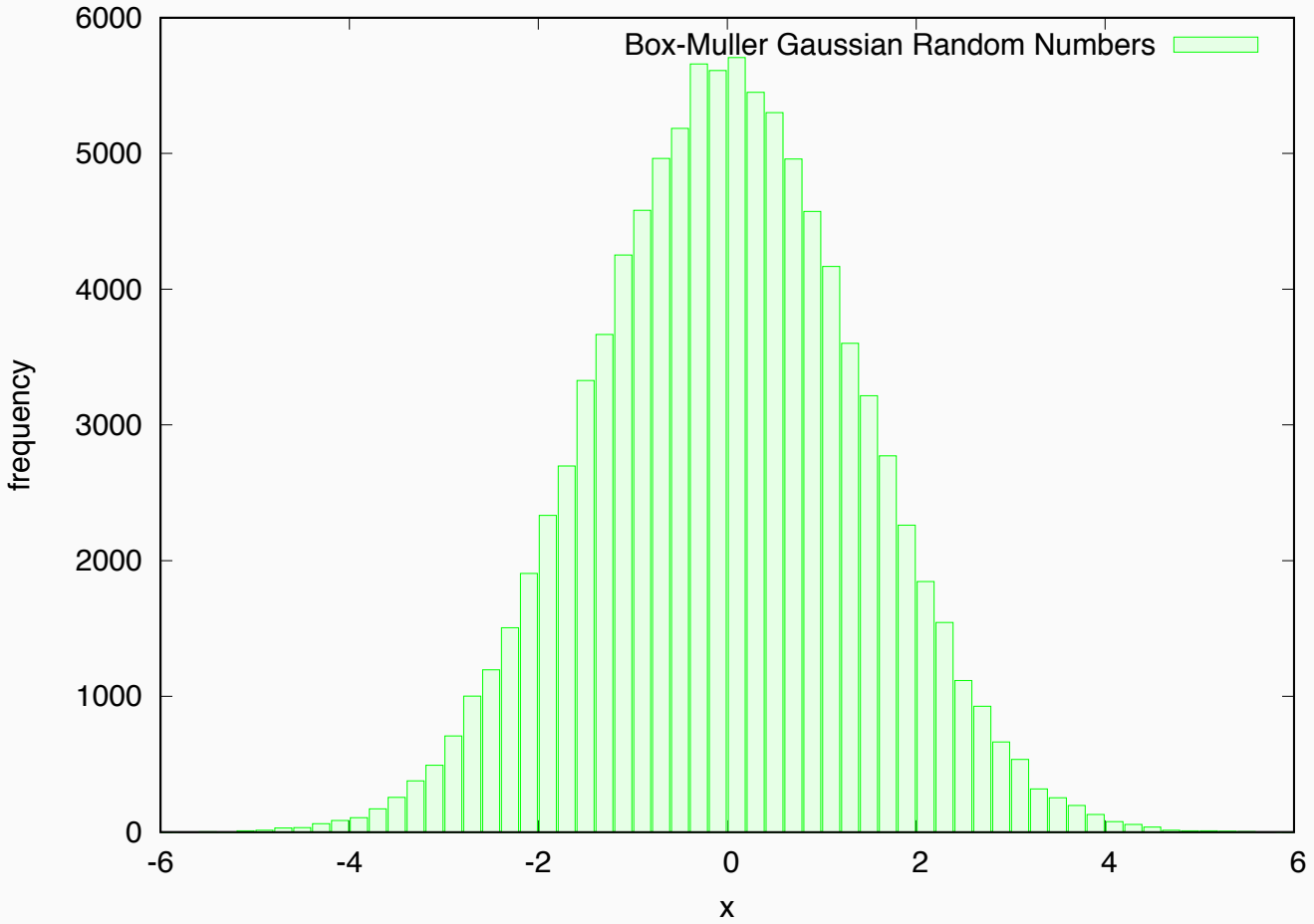


Figure 8: Distribution of the random numbers generated by the Box-Muller algorithm with 100000 numbers.

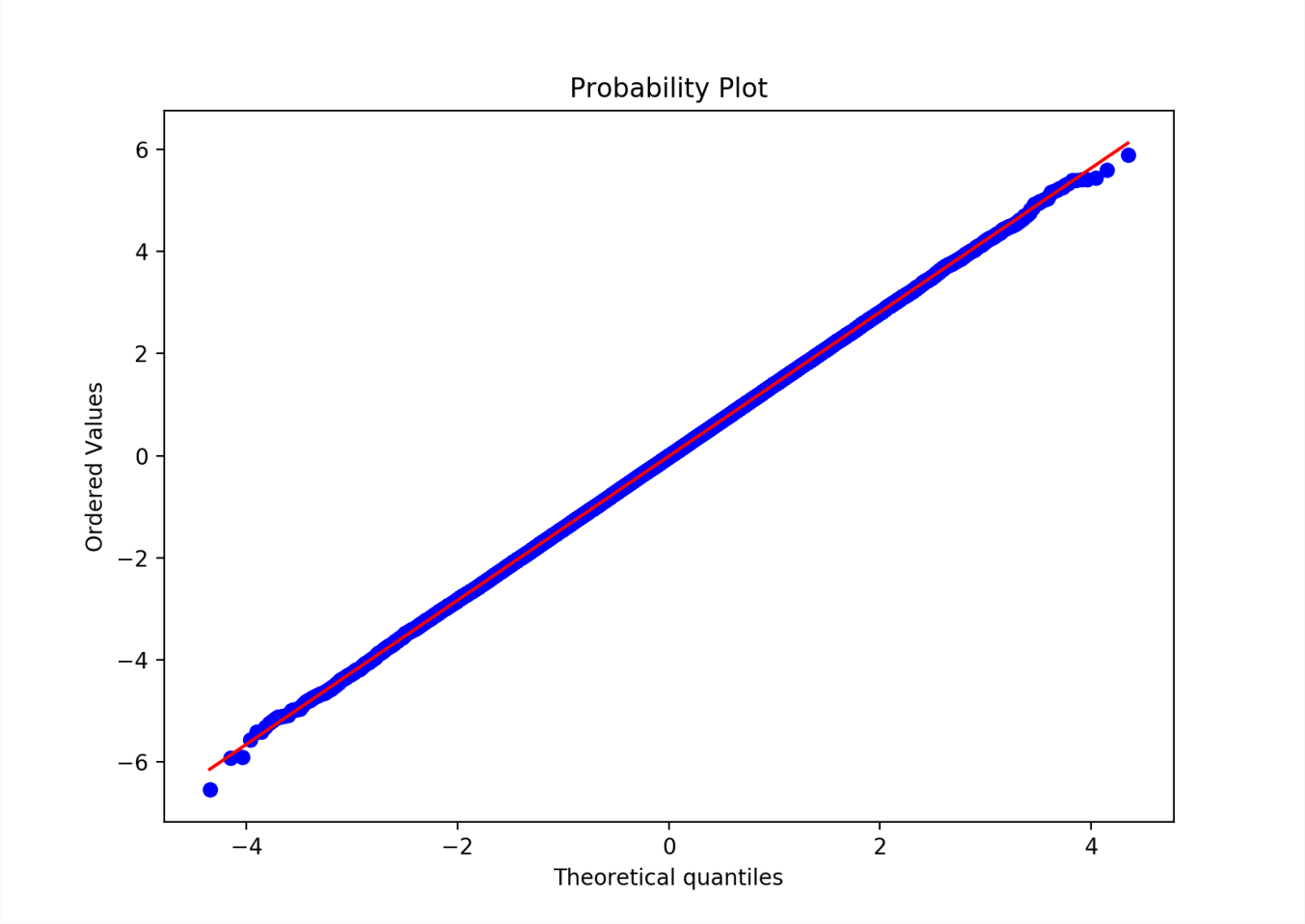


Figure 9: Distribution of the random numbers generated by the Box-Muller algorithm with 100000 numbers as a QQ-plot.

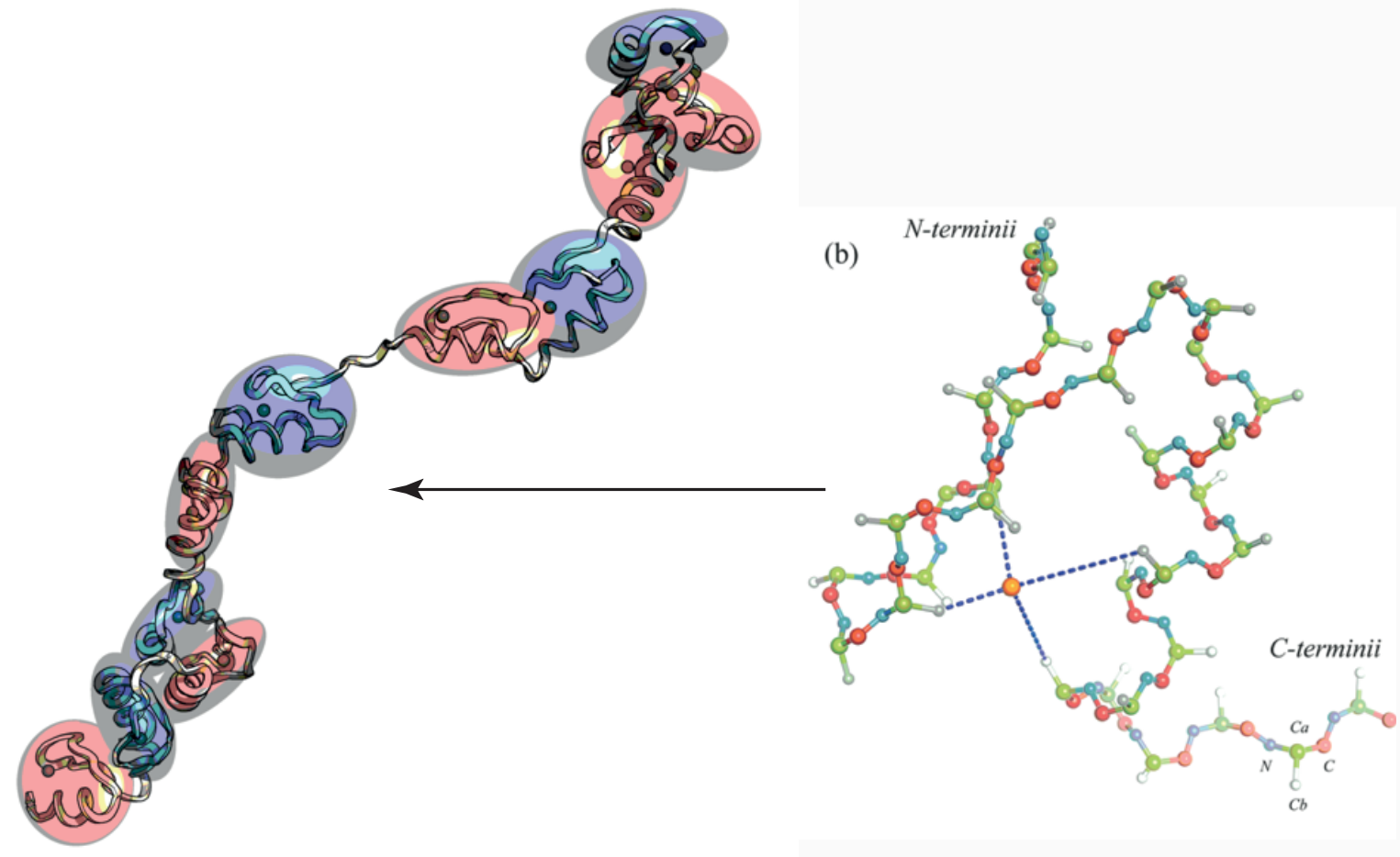


Figure 10: Polymer (macromolecule) model for the zinc finger protein.

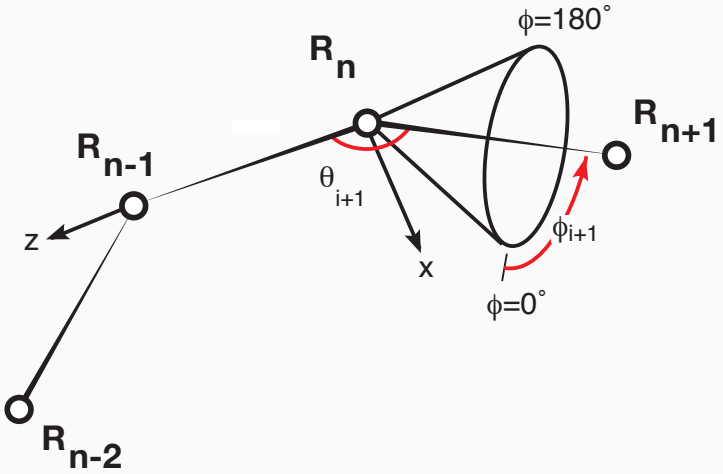


Figure 11: Polymer monomer and angle definition.

We can describe the interactions of a polymer using the Edwards-Hamiltonian which assumes the potential energy to be composed of two different terms

$$U(\{\mathbf{R}\}) = U_c(\{\mathbf{R}\}) + U_{ex}(\{\mathbf{R}\}) \tag{34}$$

where $U_c(\{\mathbf{R}\})$ is the entropic, harmonic term. It guarantees the connectivity of the chain. The term $U_{ex}(\{\mathbf{R}\})$ accounts for the excluded volume interaction. To describe the dynamics we can regard the monomers as classical Brownian particles which are

immersed in a solvent and interact via the potential (34). The solvent is described by the velocity field $\mathbf{u}(r,t)$ and assumed to be incompressible. The motion of the fluid couples to the trajectories $\mathbf{R}_n(t)$ of the monomers by dissipative forces. If the velocity of monomer n departes from the velocity $\mathbf{u}(\mathbf{R}_n(t),t)$ of the fluid at the monomer's position, it experiences a frictional force exerted by the solvent. This friction is characterized by a bare friction constant ζ.

The equation of motion for the monomers is then given by

$$\zeta \frac{d}{dt}\mathbf{R}_n(t) = \zeta \mathbf{u}(\mathbf{R}_n(t),t) - \frac{\partial U(\{\mathbf{R}\})}{\partial \mathbf{R}_n} + \xi_n \tag{35}$$

where $\xi_n(t)$ is the fluctuating Brownian force on monomer n exerted by the solvent molecules with the usual property

$$\langle \xi_n(t)\xi_m(t')\rangle = 2\zeta k_b T \delta_{n,m}\delta(t-t') \; . \tag{36}$$

The presence of N monomers give rise to additional stresses due to the presence of the forces $\mathbf{F}_n(t) = -\frac{\partial U(\{\mathbf{R}\})}{\partial \mathbf{R}_n}$ on monomer n at time t. The dynamics of the flow field is described by fluctuating hydrodynamics for an incompressible fluid. This can be

described by the Navier-Stokes equation. Let ρ_s denote the density and η_s the viscosity of the fluid then

$$\rho_s \left[\frac{\partial \mathbf{u}(\mathbf{r},t)}{\partial t} + \mathbf{u}(\mathbf{r},t)\nabla \mathbf{u}(\mathbf{r},t) \right] = \zeta_s \Delta \mathbf{u}(\mathbf{r},t) - \nabla p(\mathbf{r},t) - \sum_{m=1}^{N} \frac{\partial U(\{\mathbf{R}\})}{\partial \mathbf{R}_m} \delta(\mathbf{r} - \mathbf{R}_n(t)) + \mathbf{f}(\mathbf{r},t) \tag{37}$$

with p being the pressure. $\mathbf{f}(\mathbf{r},t)$ is a random force with

$$\langle \mathbf{f}_n(t)\mathbf{f}_m(t')\rangle = 2\eta_s k_b T \delta(\mathbf{r}-\mathbf{r}')\delta(t-t') \; . \tag{38}$$

We now allow for an external flow field $\mathbf{u}_{\text{ext}}(\mathbf{r},t)$. The field then can be written as a superposition

$$\mathbf{u}(\mathbf{r},t) = \mathbf{u}_{\text{ext}}(\mathbf{r},t) + \sum_{n=1}^{N} \Omega(\mathbf{r}\mathbf{R}_n(t))\mathbf{F}_n(t) + \mathbf{u}_{\text{f}}(\mathbf{r},t) \tag{39}$$

where Ω is the **Oseen tensor**

$$\Omega(\mathbf{r}) = \frac{1}{8\pi\eta_s r}(1 + \hat{\mathbf{r}}\hat{\mathbf{r}}^\dagger) \tag{40}$$

linked to the random force contribution $\mathbf{u}_\mathrm{f}(\mathbf{r}, t)$ via

$$\langle \mathbf{u}_\mathrm{f}(\mathbf{r}, t)\mathbf{u}_\mathrm{f}(\mathbf{r}', t')\rangle = 2k_b T\Omega(\mathbf{r} - \mathbf{r}')\delta(t - t') \; . \tag{41}$$

Define now the **mobility matrix**

$$\mathbf{M}_{n,m}(\mathbf{r}) \equiv \delta_{n,m}\mathbf{1} + \zeta(1 - \delta_{n,m})\Omega \; . \tag{42}$$

With this Equation 35 can be written as

$$\zeta(\frac{d}{dt}\mathbf{R}_n(t) - \dot{\gamma}(t)\mathbf{R}_n(t)) = -\sum_{m=1}^{N} \mathbf{M}_{n,m}(\mathbf{R}_n(t) - \mathbf{R}_m(t))\frac{\partial U(\{\mathbf{R}\})}{\partial \mathbf{R}_n} + \eta_n(\mathbf{R}_n, t) \tag{43}$$

where

Diffusion of Macromolecules VI

$$\eta_{\mathbf{n}}(\mathbf{r}_n, t) \equiv \xi_n(t) + \zeta \mathbf{u}_{\mathbf{f}}(\mathbf{r}, t) \ . \tag{44}$$

Rouse Model I

We start off our treatment of the dynamics of a polymer investigating the dynamics of the Gaussian chain model which is know as the **Rouse dynamics** [8]. That is, we are neglecting the effects coming from the excluded volume interaction and we assume no hydrodynamics interactions for the moment with zero shear. Under these assumptions our starting point is

$$\zeta \frac{d}{dt}\mathbf{R}_n(t) = -\sum_{m=1}^{N} \frac{\partial U(\{\mathbf{R}\})}{\partial \mathbf{R}_m} + \xi_n(t) \tag{45}$$

with

$$\langle \xi_n(t)\xi_m(t')\rangle = 2\zeta k_b T \delta_{n,m}\delta(t-t') \; . \tag{46}$$

Recall that within the gaussian model we have

$$P(\mathbf{R}_1,\ldots,\mathbf{R}_N) = \left(\frac{3}{2\pi b^2}\right)^{\frac{3}{2}N} \exp\left(-\sum_{n=1}^{N}\frac{3}{2b^2}(\mathbf{R}_n-\mathbf{R}_{n-1})^2\right) \; . \tag{47}$$

and obtain with Equation 45

$$\frac{d\mathbf{R}_n}{dt} = -\frac{3k_BT}{\gamma b^2}(2\mathbf{R}_n - \mathbf{R}_{n-1} - \mathbf{R}_{n+1}) + \xi_n \tag{48}$$

where we have assumed that $D = k_BT/\zeta$ is independent on $\mathbf{R}_n$. There are two special case: the two end monomers so that in total we have

$$\begin{aligned}
\frac{d\mathbf{R}_0}{dt} &= -\frac{3k_BT}{\zeta b^2}(\mathbf{R}_0 - \mathbf{R}_1) + \xi_0 && (49)\\
\frac{d\mathbf{R}_n}{dt} &= -\frac{3k_BT}{\zeta b^2}(2\mathbf{R}_n - \mathbf{R}_{n-1} - \mathbf{R}_{n+1}) + \xi_n && (50)\\
\frac{d\mathbf{R}_N}{dt} &= -\frac{3k_BT}{\zeta b^2}(\mathbf{R}_N - \mathbf{R}_{N-1}) + \xi_n && (51)\\
\langle \xi_n(t) \cdot \xi_m(t') \rangle &= 6D\delta_{n,m}\delta(t-t') \, . && (52)
\end{aligned}$$

Before we embark to solve these equations it is clear that for very large times the chain as a whole must diffuse. To describe this motion we use the radius of gyration for which we find

$$\frac{d\mathbf{R}_G}{dt} = \frac{1}{N+1}\sum_{n=0}^{N}\xi_n \tag{53}$$

with the formal solution

$$\mathbf{R}_G(t) = \mathbf{R}_G(0) + \frac{1}{N+1}\sum_{n}\int_0^t d\tau \xi_n(\tau)\,. \tag{54}$$

This can be used to calculate the mean square displacement

$$\begin{aligned}\langle(\mathbf{R}_G(t)-\mathbf{R}_G(0))^2\rangle &= \langle\int_0^t d\tau\int_0^t d\tau'\left(\frac{1}{N+1}\cdot\sum_n \mathbf{f}_n(\tau)\right)\cdot\left(\frac{1}{N+1}\sum_m \xi_m(\tau')\right)\rangle &(55)\\ &= \frac{6D}{N+1}t &(56)\\ &= 6D_G t\,. &(57)\end{aligned}$$

From this we recover the expected result that the chain as a whole diffuses just like a single particle centered around the radius of gyration

$$D_G = D/N = k_B T/N\zeta \ . \tag{58}$$

Note, that the diffusion constant is renormalized by the chain length.

To solve the equations of motion (52) we assume periodic boundary conditions

$$\mathbf{R}_n = \mathbf{R}_{n+N} \tag{59}$$

which gives rise to the possibility to expand into a Fourier series

$$\mathbf{X}_k = \sum_{n=1}^{N} \mathbf{R}_n \exp(-ikn) \ . \tag{60}$$

That is in the absence of external forces, shear etc. the chain can be treated using a normal mode analysis. The 0 mode has already been treated above. Specifically we can use

$$\mathbf{R}_n(t) = \mathbf{X}(t) \cos(an + b) \ . \tag{61}$$

Rouse Model V

Recall that the following relation holds

$$\frac{1}{N+1}\sum_{n=0}^{N}\cos\left(\frac{k\pi}{N+1}(n+\frac{1}{2})\right)=\delta_{k,0}0\leq k<2(N+1) \tag{62}$$

and thus

$$\frac{1}{N+1}\sum_{n=0}^{N}\cos\left(\frac{k\pi}{N+1}(n+\frac{1}{2})\right)=\frac{1}{2(N+1)}\frac{\sin(k\pi)}{\sin\left(\frac{k\pi}{2(N+1)}\right)}\ . \tag{63}$$

The equations of motion for the Fourier coefficients are then given by

$$\frac{d\mathbf{X}}{dt}\cos b = -\frac{3kT}{\zeta b^2}\{\cos b-\cos(a+b)\}\mathbf{X} \tag{64}$$

$$\frac{d\mathbf{X}}{dt}\cos(na+b) = -\frac{3kT}{\zeta b^2}4\sin^2\left(\frac{1}{2}a\right)\cos(na+b)\mathbf{X} \tag{65}$$

$$\frac{d\mathbf{X}}{dt}\cos(Na+b) = -\frac{3kT}{\zeta b^2}\{\cos(Na+b)-\cos((N-1)a+b)\}\mathbf{X} \tag{66}$$

where we have used the addition theorems for the geometric functions

$$\begin{aligned} & 2\cos(na+b) - \cos((n-1)a+b) - \cos((n+1)a+b) \\ = \quad & \cos(na+b)\{2-2\cos a\} = \cos(na+b) 4\sin^2\left(\frac{1}{2}a\right) \ . \end{aligned} \tag{67}$$

Further, it is clear that the above introduced parameters a an b can not be independent. We must have

$$\cos b - \cos(a+b) = 4\sin^2\left(\frac{1}{2}a\right)\cos b \tag{68}$$

$$\cos(Na+b) - \cos((N-1)a+b) = 4\sin^2\left(\frac{1}{2}a\right)\cos(Na+b) \tag{69}$$

or

$$\cos(a-b) = \cos b \tag{70}$$
$$\cos((N+1)a+b) = \cos(Na+b)\,. \tag{71}$$

We find independent solutions from

$$a-b = b \tag{72}$$
$$(N+1)a+b = k2\pi - Na - b\,. \tag{73}$$

So, finally

$$a = \frac{k\pi}{N+1}, b = \frac{1}{2}a = \frac{k\pi}{2(N+1)}\,. \tag{74}$$

We can now write

$$\mathbf{R}_n = \mathbf{X}_0 + 2\sum_{k=1}^{N}\mathbf{X}_k \cos\left(\frac{k\pi}{N+1}(n+\frac{1}{2})\right). \tag{75}$$

and invert this to yield the Fourier coefficients

$$\mathbf{X}_k = \frac{1}{N+1} \sum_{n=0}^{N} \mathbf{R}_n \cos\left(\frac{k\pi}{N+1}(n+\frac{1}{2}) \right) \ . \tag{76}$$

The equations of motion are then given by

$$\begin{aligned} \frac{d\mathbf{X}_k}{dt} &= -\frac{3k_B T}{\zeta b^2} 4 \sin^2\left(\frac{k\pi}{2(N+1)} \right) \mathbf{X}_k + \xi_k & (77) \\ \langle \xi_0(t) \cdot \xi_0(t') \rangle &= \frac{6D}{N+1} \delta(t-t') & (78) \\ \langle \xi_k(t) \cdot \xi_{k'}(t') \rangle &= \frac{3D}{N+1} \delta_{k,k'} \delta(t-t') k \neq 0 & (79) \end{aligned}$$

with the abbreviation

$$\xi_k = \frac{1}{N+1} \sum_{n=0}^{N} \xi_n \cos\left(\frac{k\pi}{N+1}(n+\frac{1}{2}) \right) \ . \tag{80}$$

These are $3(N+1)$ independent stochastic differential equations. Coming back to the zero mode we have

$$\mathbf{R}_G = \mathbf{X}_0 \tag{81}$$

and

$$\frac{d\mathbf{X}_0}{dt} = \xi_0 \tag{82}$$

$$\langle \xi_0(t) \cdot \xi_0(t') \rangle = 6\frac{D}{N+1}\delta(t-t') \, . \tag{83}$$

To make further progress we assume that only the large wavelength modes will contribute significantly to our results so that we use

$$\frac{d\mathbf{X}_k}{dt} = -\frac{1}{\tau_k}\mathbf{X}_k + \xi_k \tag{84}$$

$$\langle \xi_k(t) \cdot \xi_k(t') \rangle = \frac{3D}{N+1}\delta_{kk'}\delta(t-t') \, . \tag{85}$$

with τ_k given by

$$\tau_k = \frac{\zeta b^2 (N+1)^2}{3\pi^2 k_B T} \frac{1}{k^2} = \frac{b^2 (N+1)^2}{3\pi^2 D} \frac{1}{k^2} \ . \tag{86}$$

We can now calculate the mean square displacement

$$\begin{aligned} \langle (\mathbf{R}_n(t) - \mathbf{R}_n(0))^2 \rangle &= \langle (\mathbf{X}_0(t) - \mathbf{X}_0(0))^2 \rangle \\ &\quad + 4 \sum_{k=1}^{N} \langle (\mathbf{X}_k(t) - \mathbf{X}_k(0))^2 \rangle \cos^2 \left(\frac{k\pi}{N+1} (n + \frac{1}{2}) \right) \qquad (87) \\ &= 6 D_G t \\ &\quad + \frac{4b^2}{\pi^2} (N+1) \sum_{k=1}^{N} \frac{1}{k^2} (1 - e^{-t k^2 / \tau_1}) \cos^2 \left(\frac{k\pi}{N+1} (n + \frac{1}{2}) \right) \qquad (88) \end{aligned}$$

We need to distinguish two cases. If t is very large, i.e. $t \gg \tau_1$, the first term will dominate, yielding

$$\langle (\mathbf{R}_n(t) - \mathbf{R}_n(0))^2 \rangle = 6 D_G t \quad t \gg \tau_1 \ . \tag{89}$$

Assume now $t \ll \tau_1$. In this case we can neglect the diffusion of the chain as a whole. Averaging over all monomers, and replacing the sum over k by an integral we get

$$\begin{aligned}
\frac{1}{N+1}\sum_{n=0}^{N}\langle(\mathbf{R}_n(t)-\mathbf{R}_n(0))^2\rangle &= \frac{2b^2}{\pi^2}(N+1)\int_0^\infty dk\,\frac{1}{k^2}(1-e^{-tk^2/\tau_1}) \\
&= \frac{2b^2}{\pi^2}(N+1)\int_0^\infty dk\,\frac{1}{\tau_1}\int_0^t dt' e^{-t'k^2/\tau_1} \\
&= \frac{2b^2}{\pi^2}\frac{(N+1)}{\tau_1}\frac{1}{2}\sqrt{\pi\tau_1}\int_0^t dt'\,\frac{1}{\sqrt{t'}} \qquad (90)
\end{aligned}$$

with the final result

$$\langle(\mathbf{R}_n(t)-\mathbf{R}_n(0))^2\rangle = \left(\frac{4k_BTb^2}{3\pi\zeta}\right)^{\frac{1}{2}} t^{\frac{1}{2}} \quad t \ll \tau_1\ . \qquad (91)$$

So, at short times the mean square displacement of a typical monomer goes like the square root of t. Essentially we have used the expansion

$$1 - e^{-x} \approx x \tag{92}$$

and for the second case

$$\int_{-\pi}^{\pi} dk \to \int_{-\infty}^{\infty} dk \tag{93}$$

With

$$\int_0^{\infty} dx \frac{1}{x^2}(1 - e^{x^2}) = \sqrt{\pi} \; . \tag{94}$$

The examination of the various mean-square displacements gives insight in the dynamic behaviour of polymer systems [9]. For this we define two different displacements: the mean-square displacement of the monomers in the center of the chains

$$g_1(t) \equiv \left\langle \left[\mathbf{R}_{N/2}(t) - \mathbf{R}_{N/2}(0) \right]^2 \right\rangle \; , \tag{95}$$

and the mean-square displacement of the center of mass of the chains,

$$g_3(t) \equiv \left\langle [\mathbf{R}_{\mathrm{CM}}(t) - \mathbf{R}_{\mathrm{CM}}(0)]^2 \right\rangle \ . \tag{96}$$

This is shown in Figure 12 for g_1

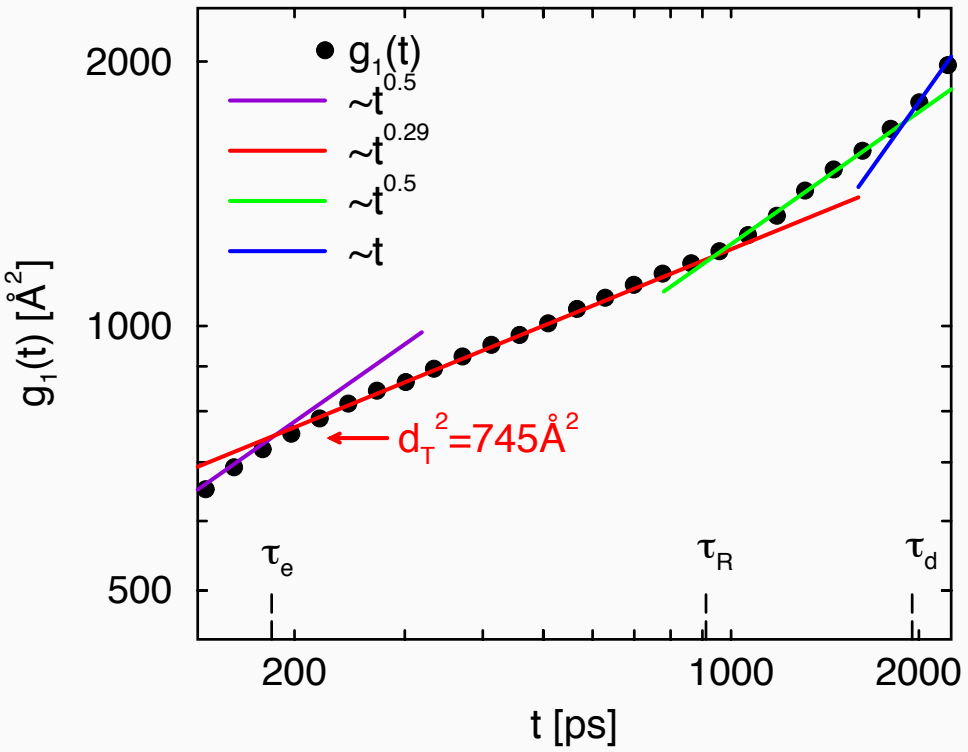


Figure 12: Diffusion of a polymer. Image taken from [10].

Rouse Model XIV

Within the treatment of the crossover from Rouse behaviour (with excluded-volume interaction) to reptation dynamics (see later) it is convenient to define certain relaxation times $\tau_1, \tau_2, \tau_3, \tau_4$ from intersection points of the curves of the several mean square displacements as follows:

$$g_1(\tau_1) = < R_G^2 > \quad (97)$$

$$g_2(\tau_2) = 2/3 < R_G^2 > \quad (98)$$

$$g_3(\tau_3) = g_2(\tau_3) \quad (99)$$

$$g_4(\tau_4) = < R_G^2 > \ . \quad (100)$$

Rouse Model XV

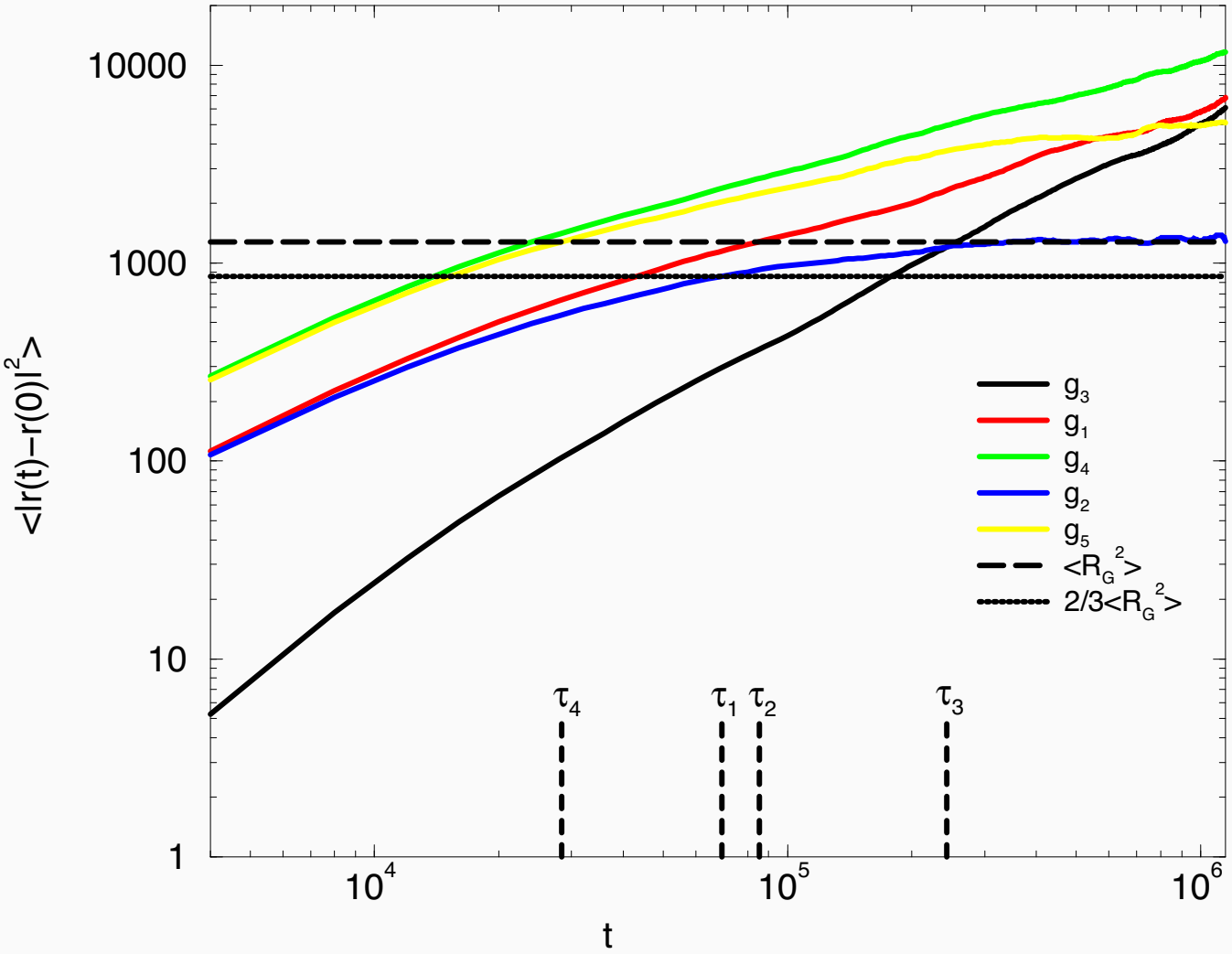


Figure 13: g1 to g4 as defined in 100. Image taken from [10].

In the ideal Rouse model for noninteracting Gaussian coils all these relaxation times are proportional to each other: $\tau_2/\tau_1 = 0.846, \tau_3/\tau_1 = 3.112, \tau_4/\tau_1 = 0.279$). These universal rations are reproduces nicely by the data in Figure 13.

We now introduce an effective chain length [10]

$$\tilde{N} = (N-1)\left[< l^2 >^{3/2} \Phi\right]^{-1/(3\nu-1)} \tag{101}$$

which leads to a crossover scaling law for relaxation times τ

$$\frac{W\tau}{N^{1-2\nu}} = \tilde{\tau}(\tilde{N}) \; . \tag{102}$$

Figures 14, 15 and 16, show the consistency of Equations 101 and 102.

Rouse Model XVII

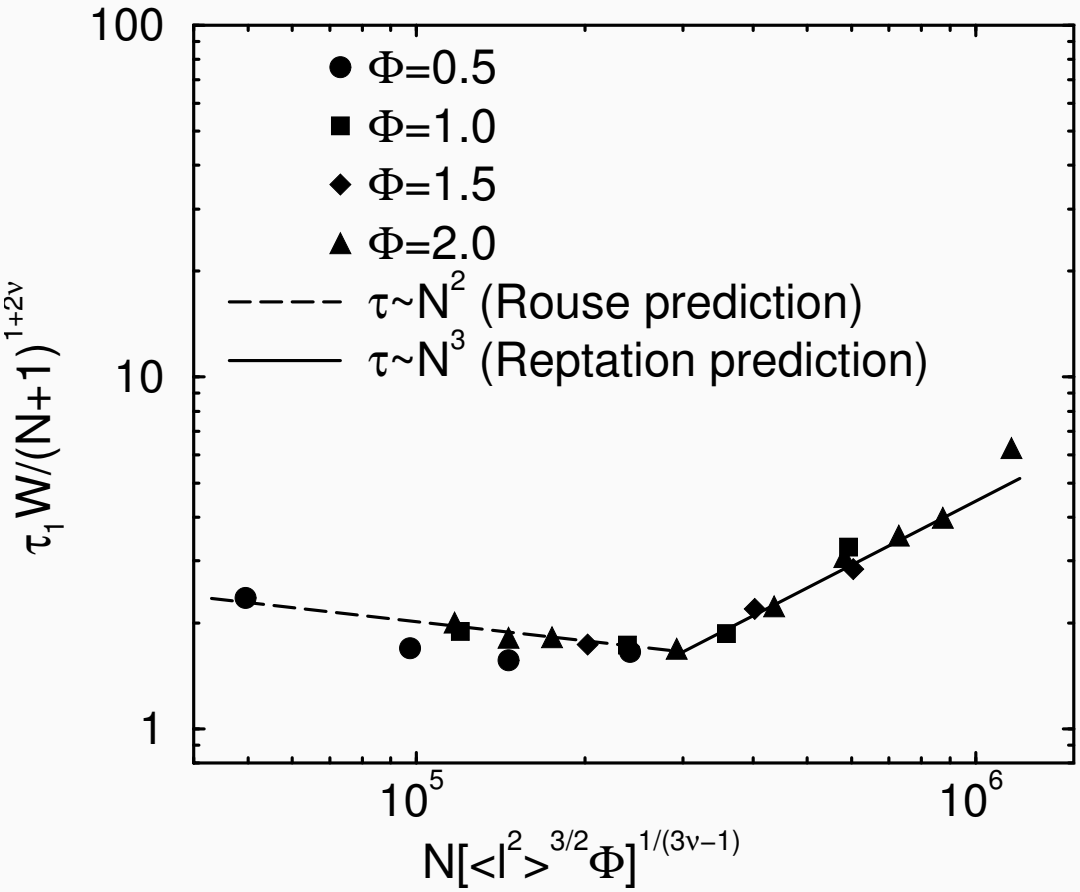


Figure 14: Log-log plot of the scaled relaxation times $\tau_1 W/N^{1+2\nu}$ vs. $\tilde{N}$. $N_p = N + 1$. Image taken from [10].

Rouse Model XVIII

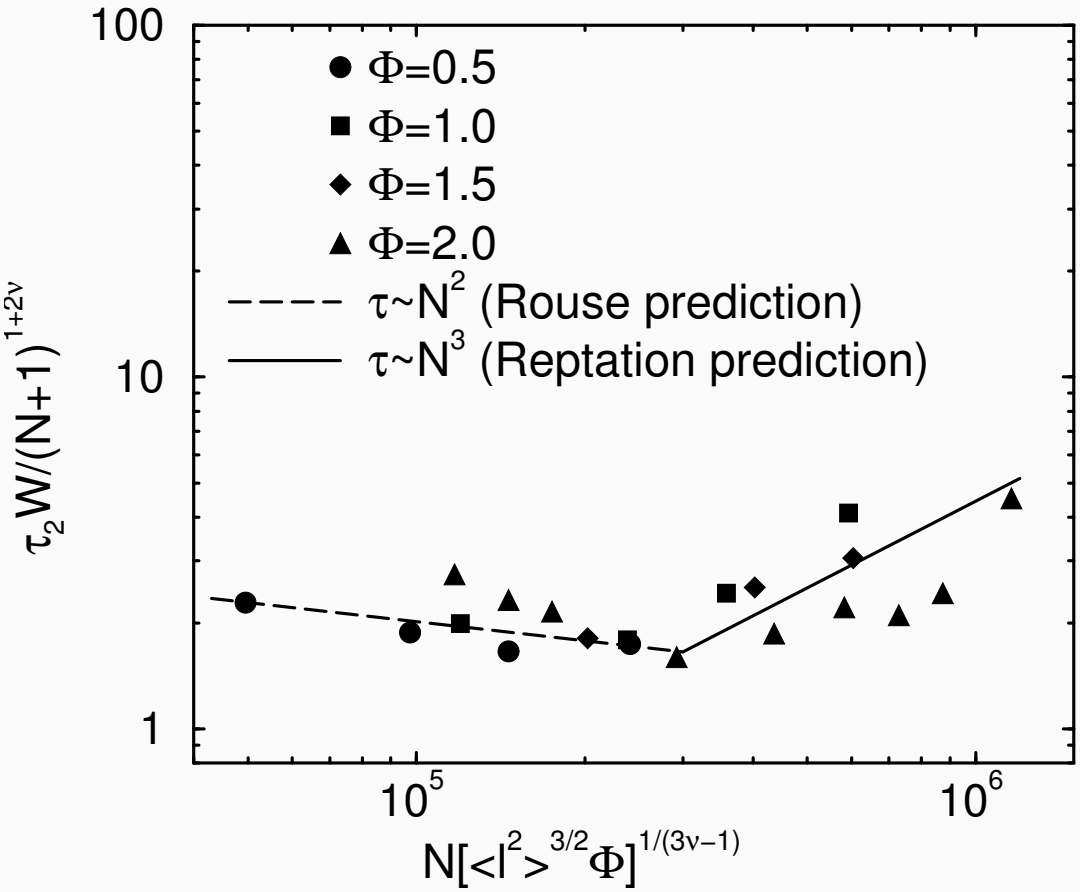


Figure 15: Log-log plot of the scaled relaxation times $\tau_2 W / N^{1+2\nu}$ vs. $\tilde{N}$. $N_p = N + 1$. Image taken from [10].

Rouse Model XIX

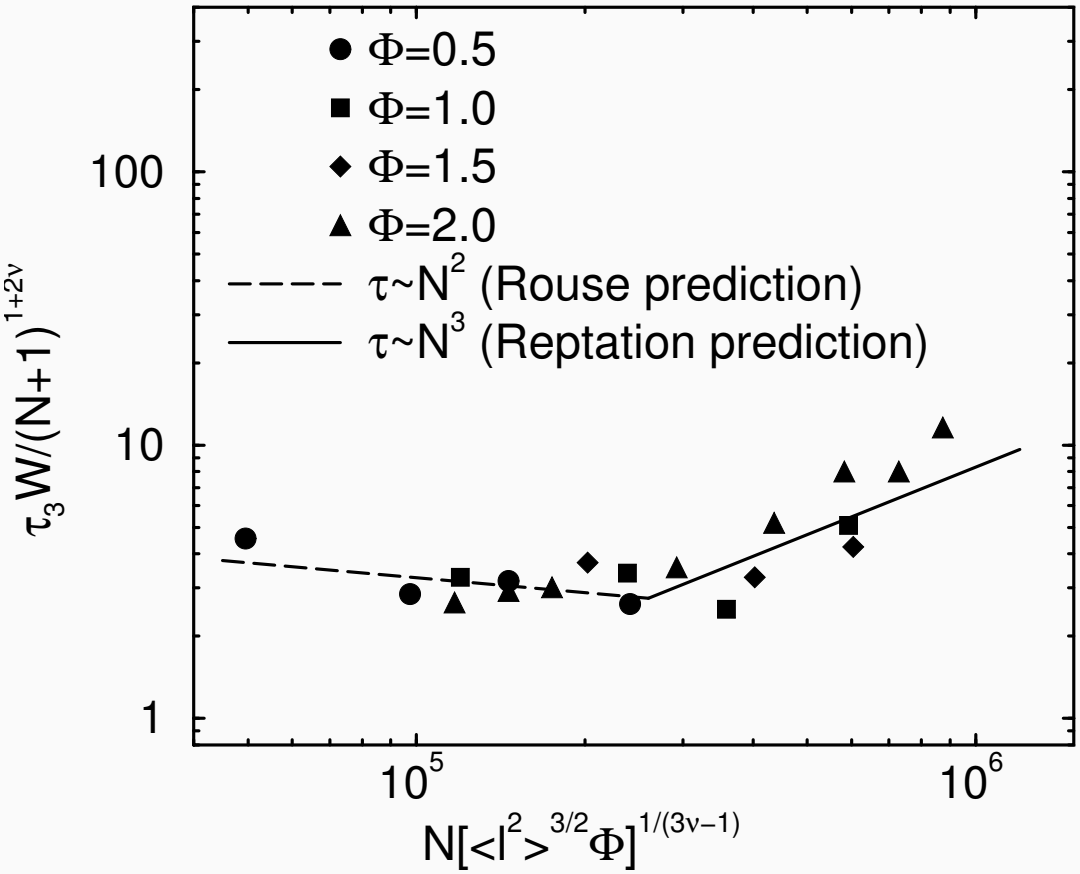


Figure 16: Log-log plot of the scaled relaxation times $\tau_3 W / N^{1+2\nu}$ vs. $\tilde{N}$. $N_p = N + 1$. Image taken from [10].

Rouse Model XX

The scaling plots (see Figures 14, 15, and 16) show a crossover from a Rouse to a reptation regime.

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Random Walks

Random Walk I

- We assume a lattice. For simplicity we take a simple square lattice [11, 12].
- On this lattice a particle or walker is placed. The walker regards this initial position as the origin.
- The walker draws a random number and decides, according to the drawn random number, to go to a new position on the lattice.
- The new position must be one of the nearest neighbours, and each of the neighbours has the same probability to be visited.
- Once the walker is at the new position, the walker regards this position as his new origin. In other words, he immediately forgets where he came from.
- Every step is made as if it is the first step.
- All steps are then independent of each other.
- It is assumed that in the array random are stored numbers which are uniformly distributed in the interval $(0, 1)$.
- A random number from the array is then multiplied by 4 and converted to an integer value. This integer value can either be $0, 1, 2$ or 3 labeling the four possible directions or nearest neighbours on the square lattice.

Random Walk II

- The numbers $0, 1, 2$ and 3 are uniformly distributed as long as the numbers in the array random are so distributed. Depending on the direction the random number points to, the walker occupies the appropriate position on the lattice by increasing or decreasing the x or y variable.
- The variables x_n and y_n hold the new position of the random walker.

```
/* ---- Choose a new nn site ---- */
i = floor(random[index++]* 4.0);
switch (i) {
   case 0: xn = x-1;
           yn = y;
           break;
   case 1: yn = y-1;
           xn = x;
           break;
   case 2: yn = y+1;
           xn = x;
           break;
   case 3: xn = x+1;
           yn = y;
           break;
} /* ---- switch i ---- */
```

1D Random Walk

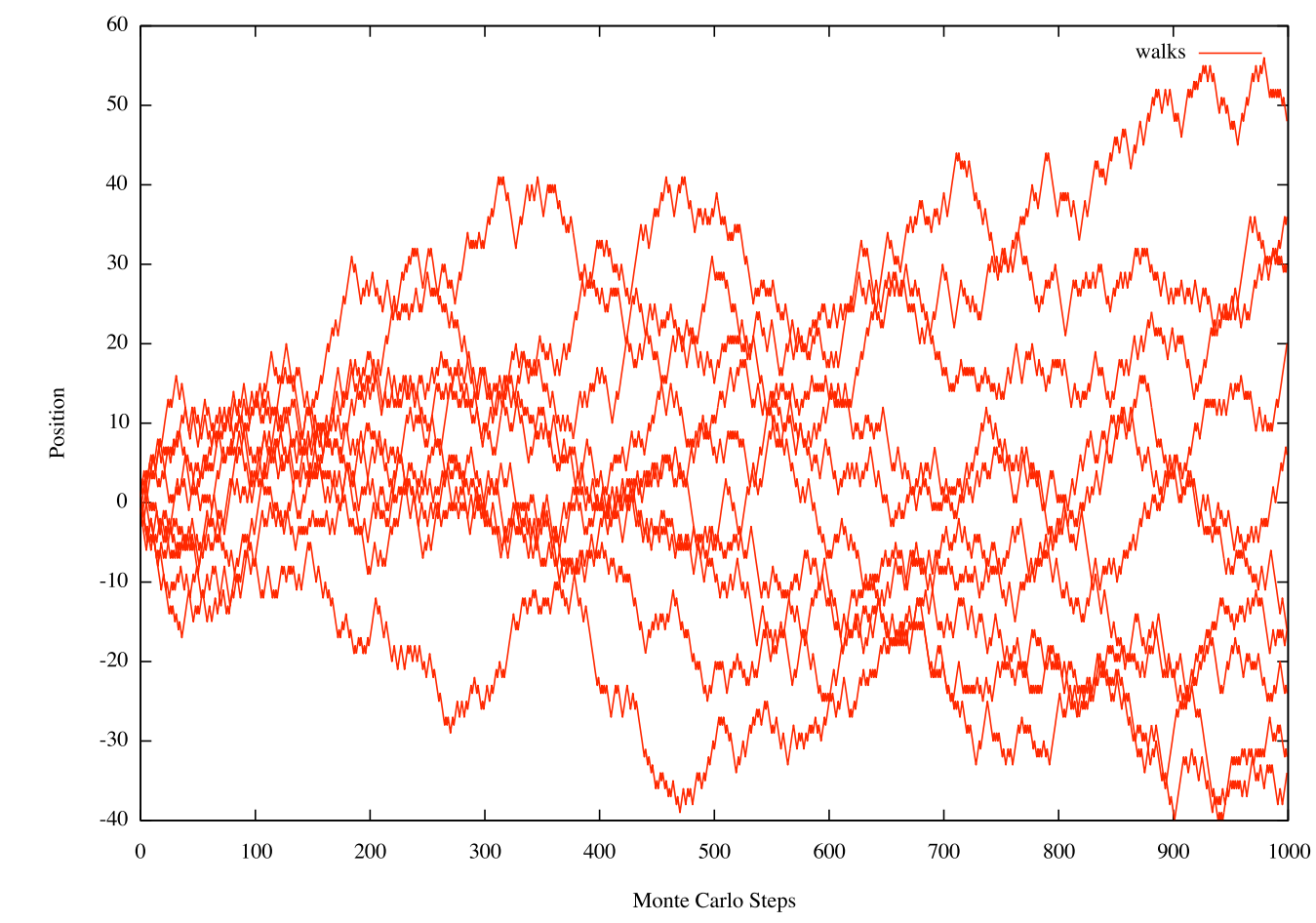


Figure 17: Examples of random walks in one dimension.

Random Walk Contact Map I

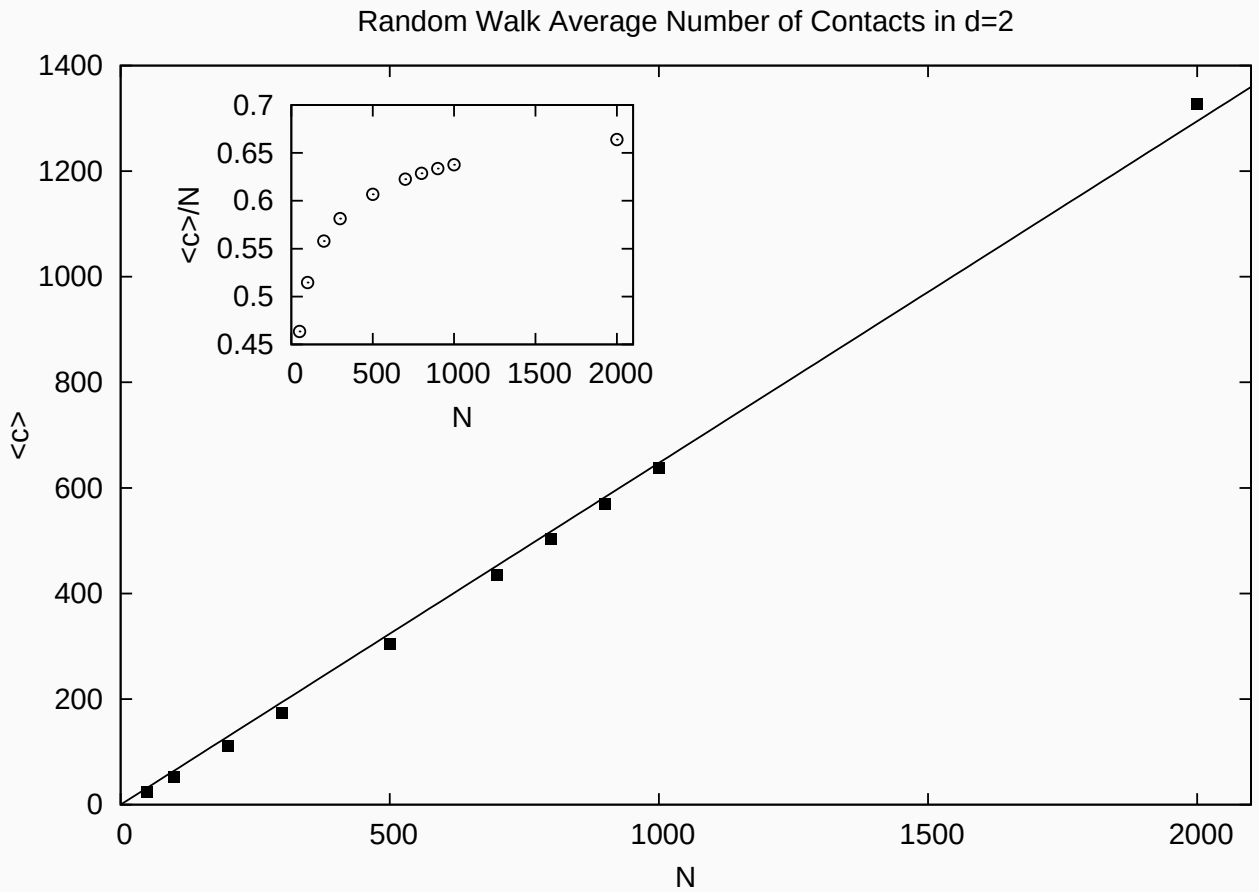


Figure 18: Number of self-contacts for the random walk.

Random Walk Contact Map II

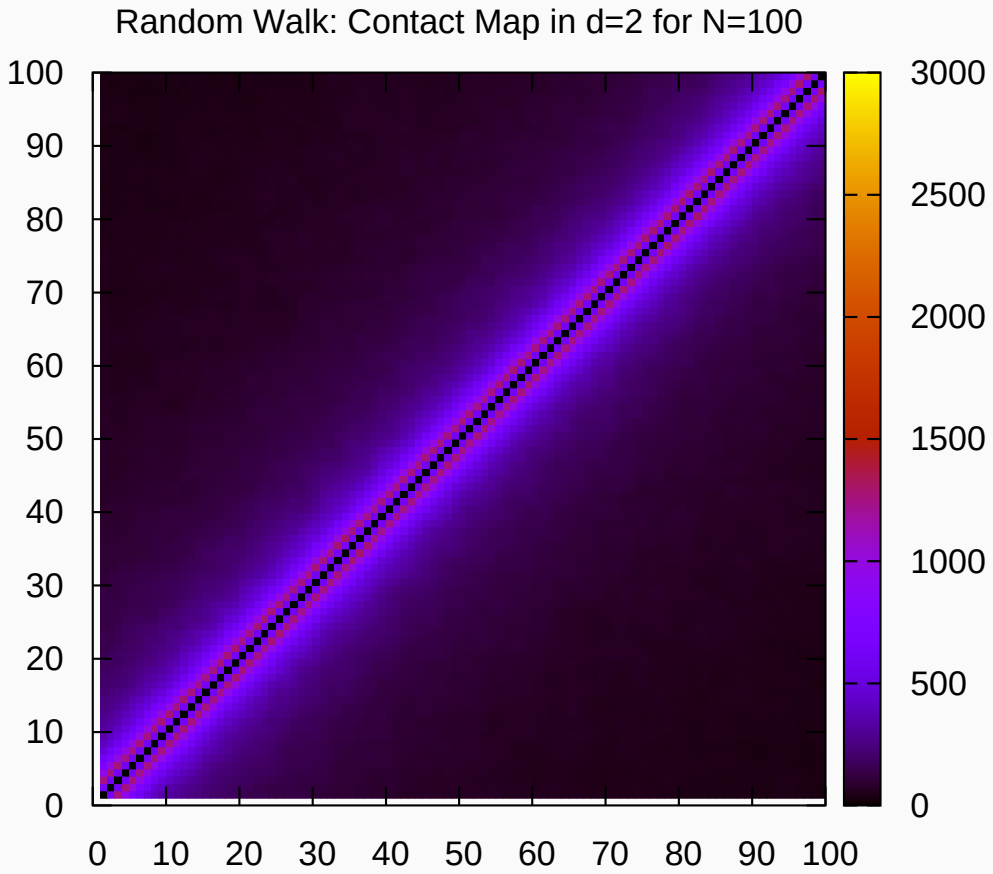


Figure 19: Contact map for the random walk.

- Let us assume that the walker performed N steps. This constitutes one realization of a random walk.
- We may now be interested in computing properties of such a walk. From just one realization we cannot draw any conclusion since the walk may be atypical. We need to generate many walks, calculate for every walk the desired property and then average over the results.
- The point which we want to make is that the generation of the samples, i.e., all the realizations of random walks are generated independently. Let A_i be the observable property computed for the i-th realization of a random walk. We define the average, or expectation value for the observable A, denoted by $<A>$, as the arithmetic mean over all A_i

$$<A> \quad = \frac{1}{n}\sum_{i=1}^{n} A_i \ . \tag{103}$$

Gaussian Walk

Algorithm 2 Gaussian Walk

1: **for** *n_cycles* **do**
2: choose initial site x_0
3: **for** *length_of_walk* **do**
4: Draw random displacement Δx from a normal distribution
5: $x_{i+1} = x_i + \Delta x$
6: **end for**
7: Compute property of the walk
8: **end for**

Levy Random Walks I

Levy flights (anomalous diffusion processes) are α-stable Levy processes. The hallmark of these processes is that they have an infinite variance (expect for the reduction to the gaussian case $\alpha = 2$). Furthermore they are scale-invariant and self-similar (for a review see [13]).

Gaussian and Levy Random Walks

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

```
import random
import math

def rng(alpha, beta, c, delta):
  # Validation of the parameters
  if (alpha < 0.1 or alpha > 2):
    x = float('nan')
    return x
  if (abs(beta) > 1):
    x = float('nan')
    return x
  w = -math.log(random.random())
  phi = (random.random()-0.5)*math.pi
  # Box-Muller Algorithm for the Gaussian case
  if (alpha == 2):
    x = 2*math.sqrt(w) * math.sin(phi)
    x = delta + c*x
    return x
  if (beta == 0.0):
    # This is Cauchy
    if (alpha == 1):
        x = math.tan(phi)
    else:
        # Levy case
        inva = float(1.0/alpha)
        x = math.pow((math.cos((1.0-alpha)*phi)/w),(inva - 1.0)) \
              * math.sin(alpha * phi) / math.pow(math.cos(phi),inva)
  return x
```

Gaussian and Levy Random Walks


```
random.seed(411)
# Levy Random Walk (flight) generation
alpha = 1
beta  = 0.0
delta = 0.0
c = 1
n = 100000

x = 0
y = 0

for r in range(n):
  x += rng(alpha,beta,c,delta)
  y += rng(alpha,beta,c,delta)

# Gaussian Random Walk generation
alpha = 2
beta  = 0.0
delta = 0.0
c = 1
n = 100000

x = 0
y = 0

for r in range(n):
  x += rng(alpha,beta,c,delta)
  y += rng(alpha,beta,c,delta)
```

Gaussian and Levy Random Walks I

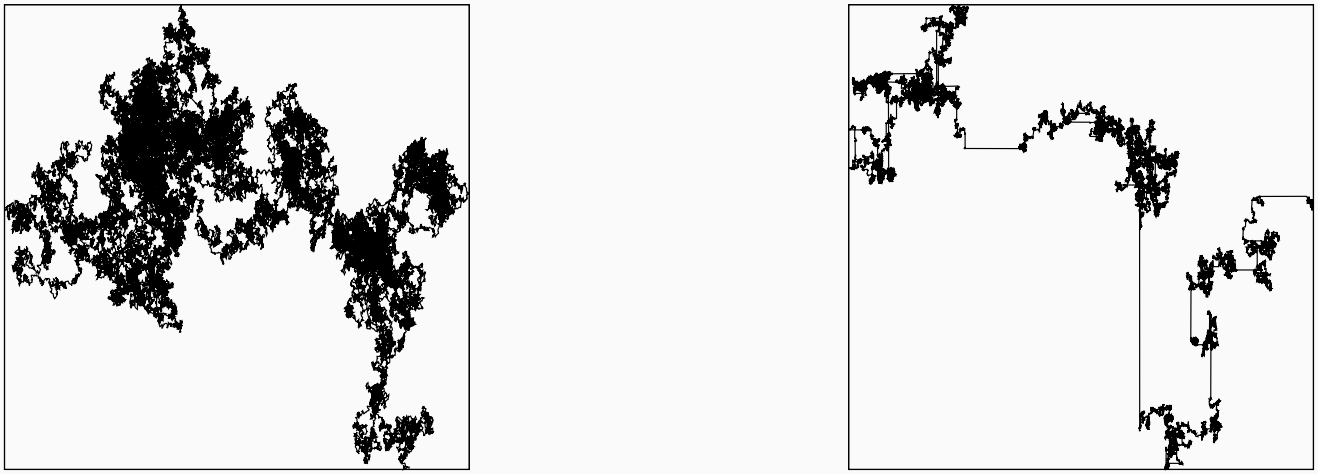
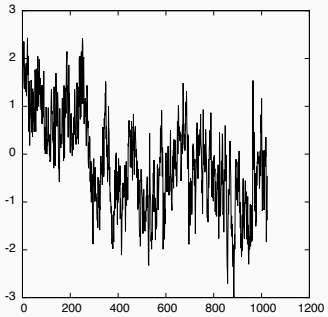
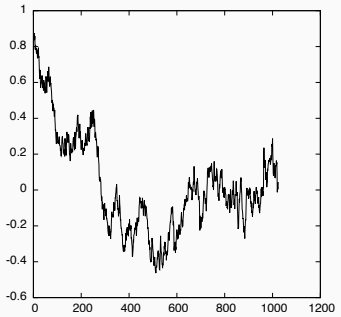


Figure 20: The left panel shows the gaussian random walk and the right panel a Levy walk with parameter $\alpha = 1.66$. Both walk are 100000 steps long

Gaussian and Levy Random Walks I





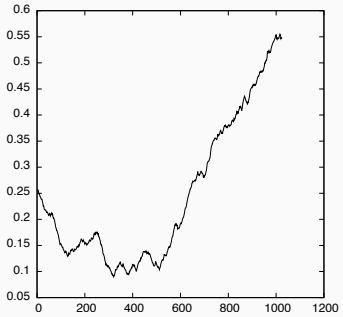


Figure 21: Fractional random walks in D = 2 for $\sigma = 1$, $\mu = 0$, $H = 0.1$, 0.5 and 0.9 respectively from left to right starting with the same seed for the random number generator.

Fractional Random Walk

```
    N = int(pow(2,max_steps))
    x = [0.0] * (N+1)
    x[N] = sigma * the_rng.generate_random_number()
    D = N
    d = int(D/2)
    level = 1
    while ( level <= max_steps ):
       dispi = sigma * pow(0.5,level*H)*sqrt(0.5)*sqrt(1-pow(2.0,2.0*H-2.0))
       for i in range(d,int(N-d)+1,D):
          x[i] = 0.5 * (x[i-d] + x[i+d])
       for i in range(0,int(N+1),int(d)):
          x[i] = x[i] + dispi * the_rng.generate_random_number()
       D = int(D/2)
       d = int(d/2)
       level += 1
```

Code 2: Fractional Random Walk

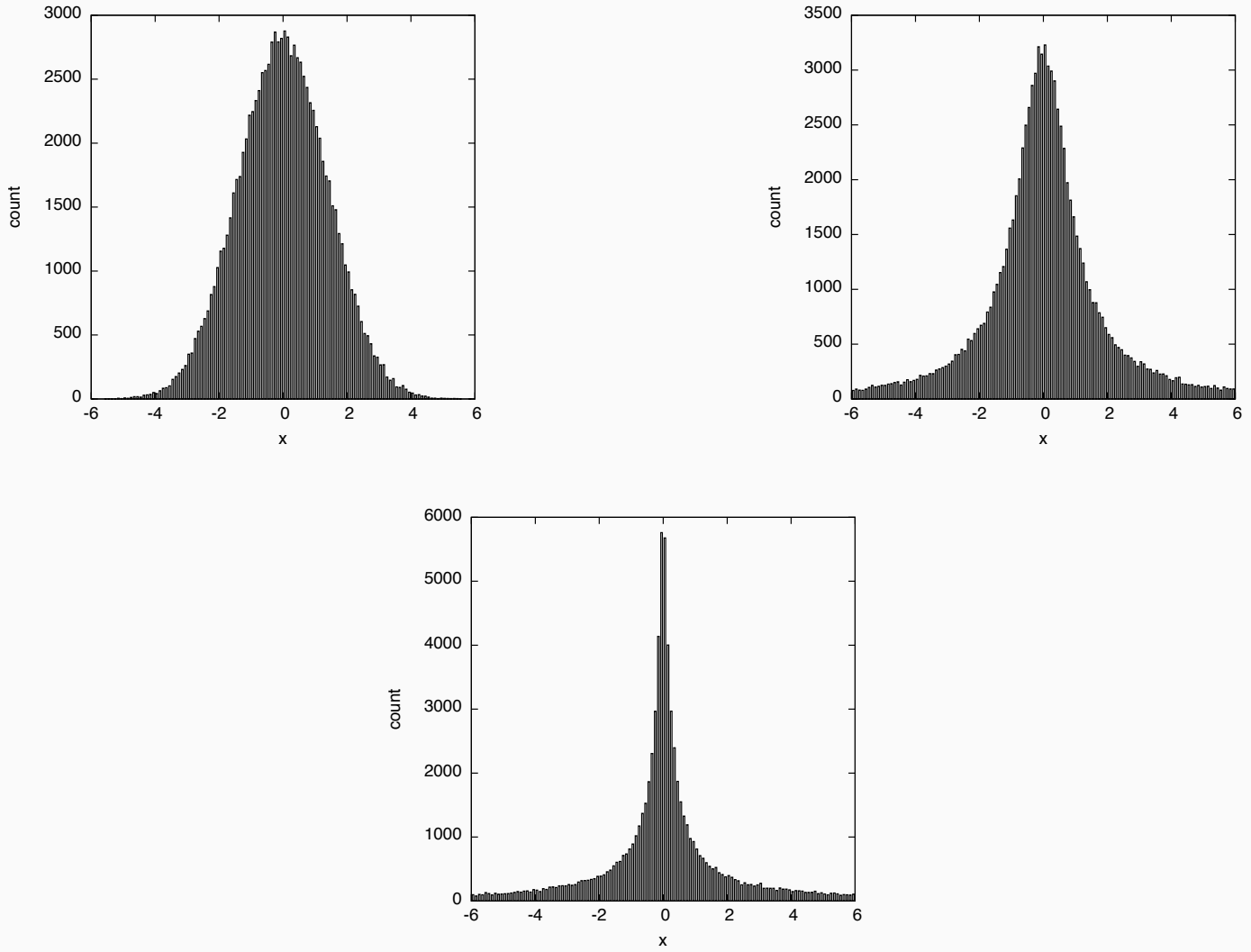


Figure 22: The left panel shows the result of the Box-Muller algorithm for the generation of gaussian distributed random numbers. The middle panel shows the case of generation of Cauchy distributed random numbers. The right panel shows the general case. Shown are the results for a sample size of 100000.

Geometric Random Walk I

The essential idea underlying the random walk for real processes is the assumption of mutually independent increments of the order of magnitude for each point of time. However, economic time series in particular do not satisfy the latter assumption. Seasonal fluctuations of monthly sales figures for example are in absolute terms significantly greater if the yearly average sales figure is high. By contrast, the relative or percentage changes are stable over time and do not depend on the current level of

Analogously to the random walk with i.i.d. absolute increments $Z_t = X_t - X_{t-1}$, a geometric random walk $\{X_t;\ t \geq 0\}$ is assumed to have i.i.d. relative increments

$$R_t = \frac{X_t}{X_{t-1}}\,, \quad t = 1, 2, \ldots\,.$$

For example, a geometric binomial random walk is given by

$$X_t = R_t \cdot X_{t-1} = X_0 \cdot \Pi_{k=1}^{t} R_k \tag{104}$$

where $X_0, R_1, R_2, \ldots$ are mutually independent and for $u > 1$, $d < 1$:

$$P(R_k = u) = p\ ,\ P(R_k = d) = 1 - p\ .$$

Geometric Random Walk II

Given the independence assumption and $\mathbf{E}[R_k] = (u-d)p + d$ it follows from equation $\mathbf{E}[X_t]$ increases or decreases exponentially as the case may be $\mathbf{E}[R_k] > 1$ or $\mathbf{E}[R_k] < 1$

$$\mathbf{E}[X_t] = \mathbf{E}[X_0] \cdot (\mathbf{E}[R_1])^t = \mathbf{E}[X_0] \cdot \{(u-d)p + d\}^t.$$

If $\mathbf{E}[R_k] = 1$ the process is on average stable, which is the case for

$$p = \frac{1-d}{u-d}\,.$$

For a recombining process, i.e. $d = \frac{1}{u}$, this relationship simplifies to

$$p = \frac{1}{u+1}\,.$$

Taking logarithms in equation yields:

$$\ln X_t = \ln X_0 + \sum_{k=1}^{t} \ln R_k\ .$$

Geometric Random Walk III

$\tilde{X}_t = \ln X_t$ is itself an ordinary binomial process with starting value $\ln X_0$ and increments $Z_k = \ln R_k$ for which hold:

$$P(Z_k = \ln u) = p, \quad P(Z_k = \ln d) = 1 - p \ .$$

For t large, $\tilde{X}_t$ is approximately normally distributed, i.e. $X_t = \exp(\tilde{X}_t)$ is approximately lognormally distributed.

Geometric Random Walk

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

```
import numpy as np
import matplotlib.pyplot as plt

plt.title('Geometric Random Walk')
plt.xlabel('time')
plt.ylabel('displacement')

mu = 0.0001
sigma = 0.01
max_samples = 10

for sample in range(max_samples):

  g = np.random.normal(mu, sigma, size=500)
  total_return = (1+g).cumprod()
  plt.plot(total_return)

plt.show()
```

Code 3: Geometric Random Walk

Geometric Random Walk I

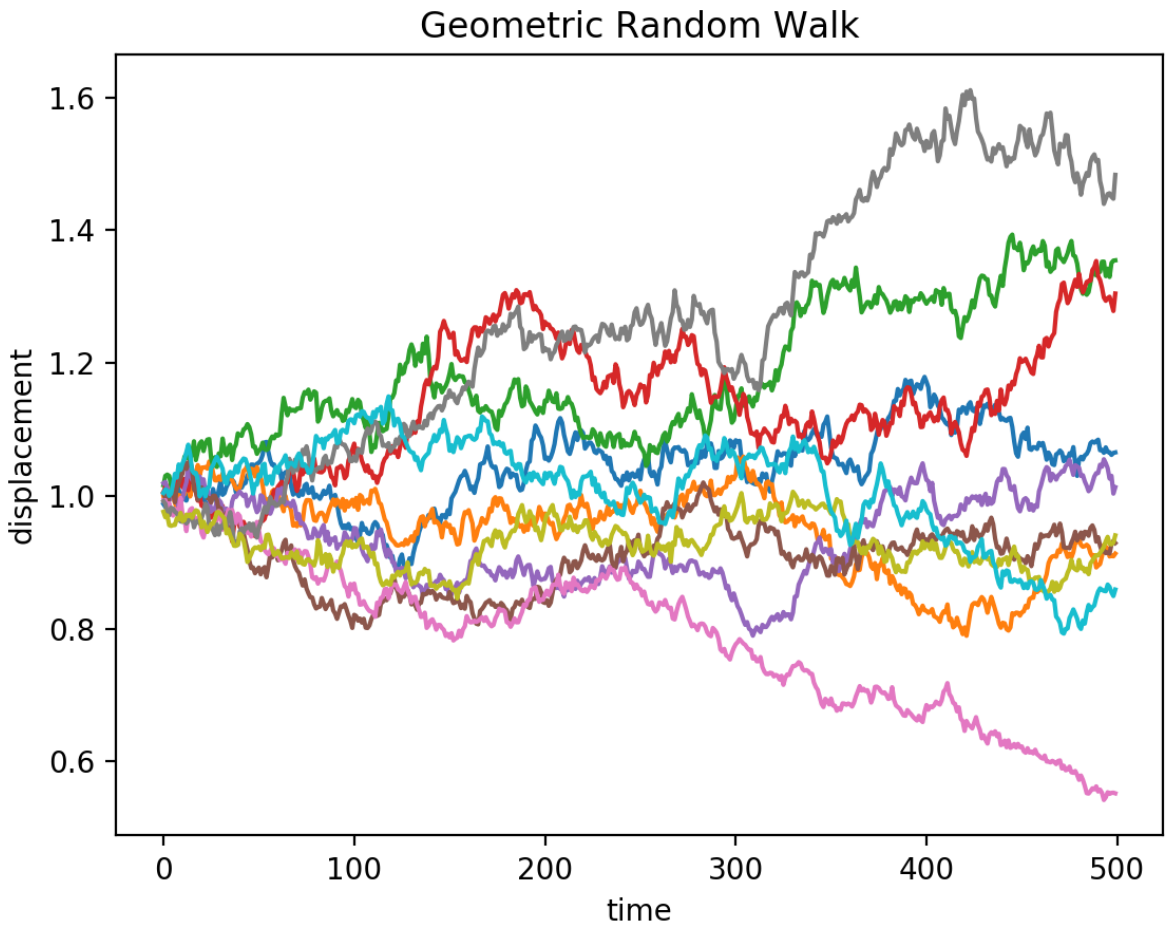


Figure 23: Geometric random walks in D = 1 for $\sigma = 0.01$, $\mu = 0.0001$

Self-Avoiding Random Walks

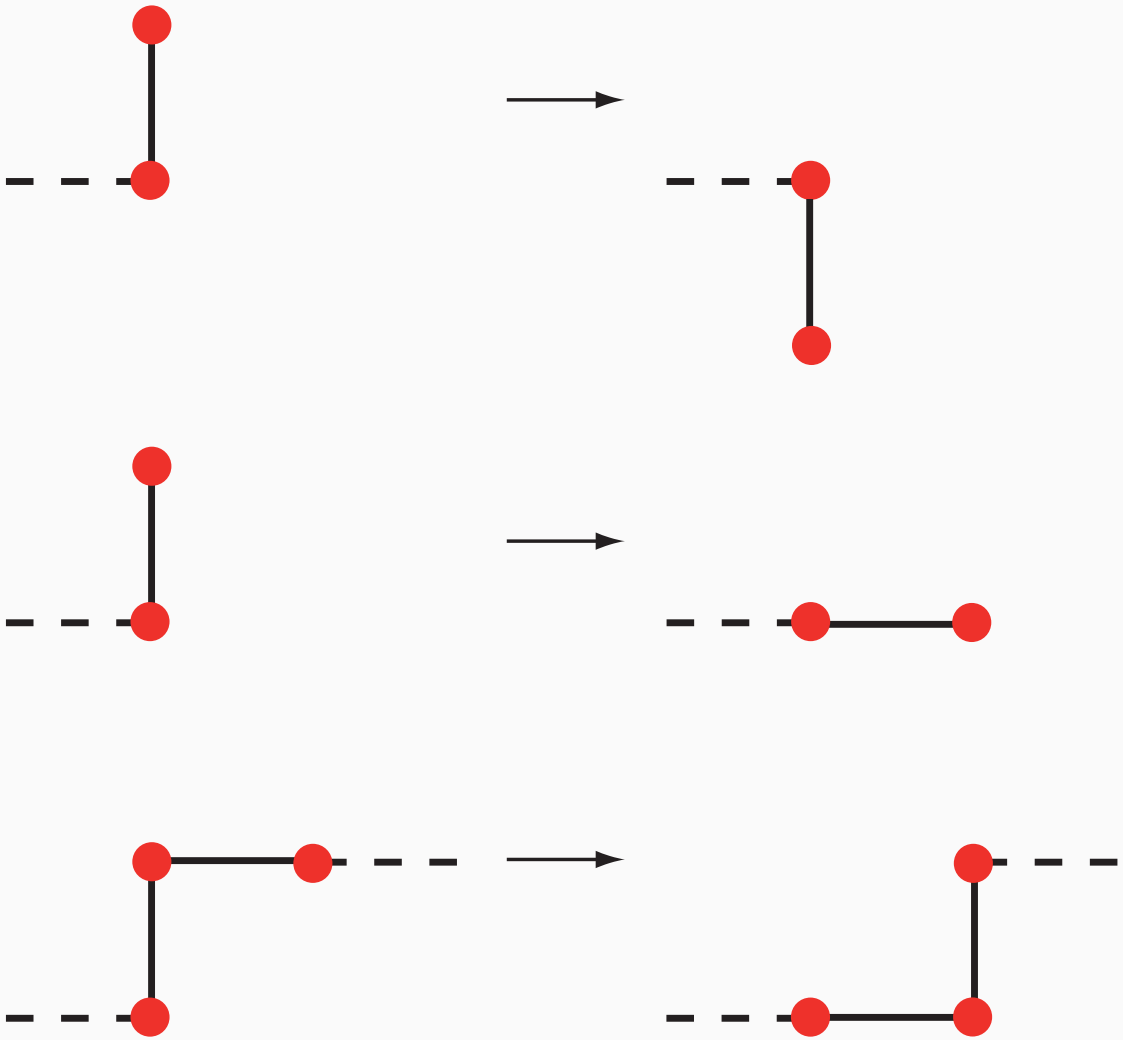
The code can be found here

```
while ( sample < sample_size ) {

  /* ==== Reset the walker to the origin ==== */

  w[0][0]   = xc;
  w[0][1]   = yc;
  x         = xc;
  y         = yc;
  l         = 0;
  occupancy = 0;
  walk++;

  return_code = r250( N,ran,mf);

  while ( (l < N)  &&  (occupancy == 0) )  {
    d  = ran[l] * 4;
    switch (d) {
        case 0:  x++;
                 break;

        case 1:  y++;
                 break;

        case 2:  x--;
                 break;

        case 3:  y--;
                 break;

    }

    if ( ( x < 0 ) || ( x == L ) || ( y < 0 ) || ( y == L ) ) {
        /* Random walker not on the lattice */
        exit(-1);
    }

    if ( g[x][y] < walk ) {
        g[x][y] = walk;
        l++;
        w[l][0] = x;
        w[l][1] = y;
        occupancy = 0;
    }
    else {
        occupancy = 1;
    }
  }

  /* ==== Now check if a SAW was generated. If yes, then ==== */
  /* ==== we do the analysis, else we must try again     ==== */

  if ( l == N ) {
    /* ---- we can compute the end-to-end distance etc. ---- */

    x = xc - w[N-1][0];
    y = yc - w[N-1][1];
    end_to_end += x*x + y*y;

    cmx = 0;
    cmy = 0;
```

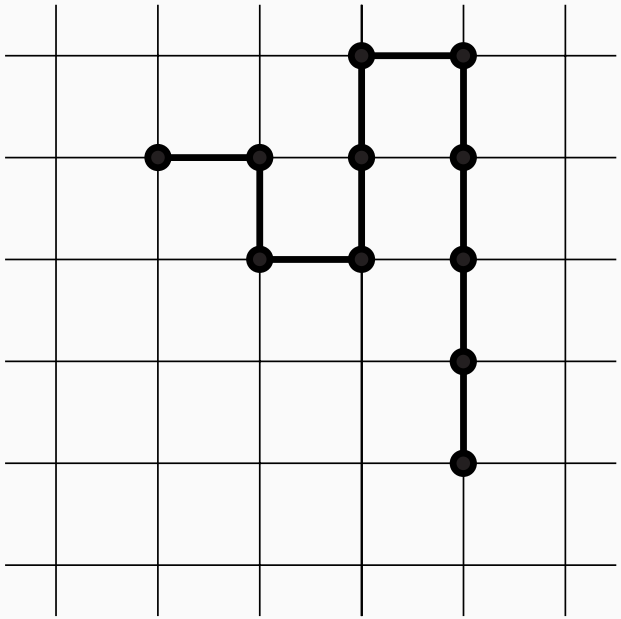
Self-Avoiding Random Walks: Simple Sampling

- Performing the simple sampling simulation it becomes immediately evident that we have a problem with the simple sampling technique for the self-avoiding random walk model.
- As we increase the number of steps the walker should travel, it becomes harder and harder to find a walk. In almost all cases the walk terminates earlier because there is a violation of the self-avoiding condition! (attrition problem).
- This shows that the simple sampling, even though being the simplest and perhaps even most powerful method has clear limitations.

Self-Avoiding Random Walks: Local Move Algorithms



Self-Avoiding Random Walks: Reptation Algorithm



Algorithm 3 Basic Algorithm: Reptation Algorithm

1: Assume that we have generated a random walk.
2: Choose one of the end points at random and delete this point.
3: Choose one the end points at random.
4: Add the delete point to the choosen end with a random direction.

Self-Avoiding Random Walks: Pivot Algorithm I

- Let W denote the set of self-avoiding walks of length N on a lattice λ.
- Further let $G(\lambda)$ be the group of lattice symmetries.
- The pivot algorithm [14] takes a self-avoiding random walk and pivots the walk to generate a new walk from the set W such the sequence of generated walks yields a Markov chain which is aperiodic and irreducible with uniform stationary distribution π.
- The sequence $\{\omega_t\}$ is aperiodic and irreducible with uniform stationary distribution π.
- The sequence further is reversible

$$\pi(\omega_i)P(\omega_i,\omega_j) = \pi(\omega_j)P(\omega_j,\omega_i) \quad . \tag{105}$$

Since π is uniform, we need to show that P is symmetric. Suppose there are m ways to move, with one pivot, from a self-avoiding walk ω to another self-avoiding walk $\bar{\omega}$. For $i = 1, 2, ..., m$, consider the pairs (x_i, g_i). Each pair gives a transition, using the pivot algorithm from ω to $\bar{\omega}$.

Thus,

$$P(\omega, \bar{\omega}) = \sum_{i=1}^{m} P(g = g_i) \cdot P(x = x_i) \quad . \tag{106}$$

Notice that the pairs (xi, g_i^{-1}), for $i = 1, 2, ..., m$ give one-step transitions from $\bar{\omega}$ and that $P(g = g_i) = P(g = g_i^{-1})$ because g is chosen uniformly. Therefore

$$P(\omega, \bar{\omega}) = \sum_{i=1}^{m} P(g = g_i) \cdot P(x = x_i) = \sum_{i=1}^{m} P(g = g_i^{-1}) \cdot P(x = x_i) = P(\bar{\omega}, \omega) \quad . \tag{107}$$

Self-Avoiding Random Walks: Pivot Algorithm III

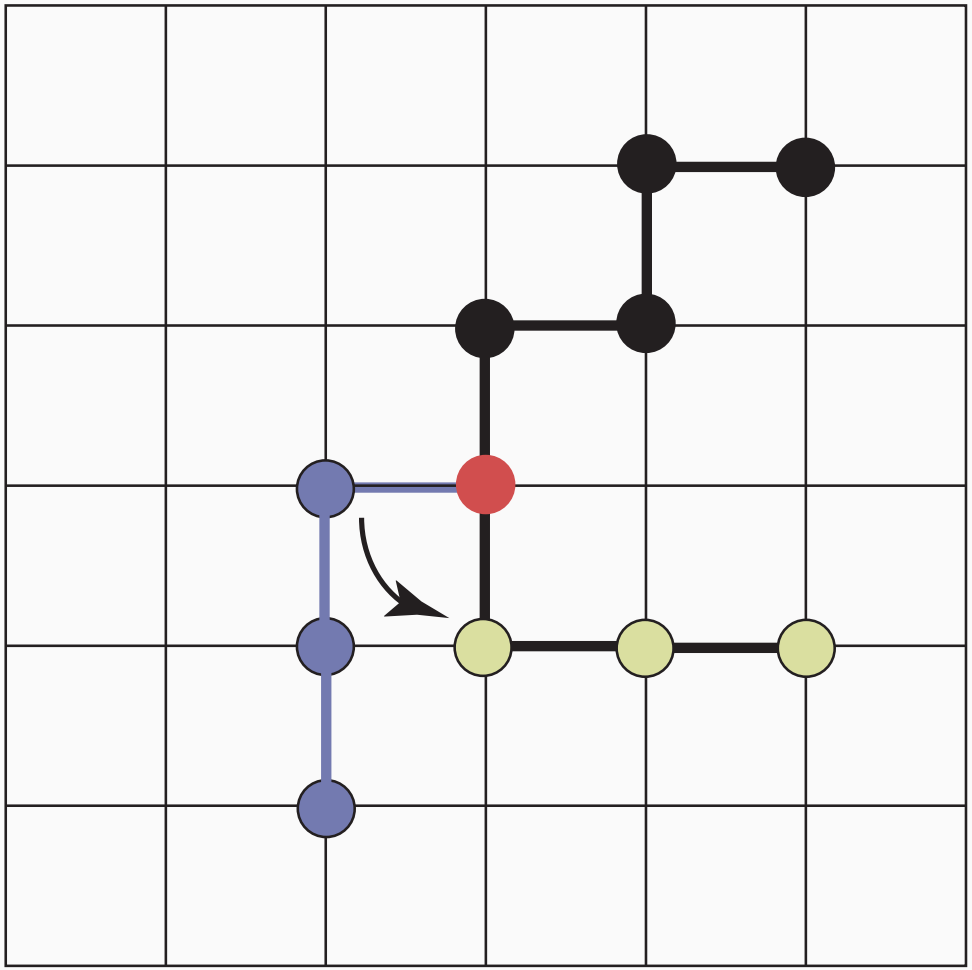


Figure 24: Pivot move: The rotation is made around the red monomer.

Algorithm 4 Basic Algorithm: Pivot Algorithm

1: Start with a self-avoiding walk $\omega_0 \in W$.
2: Next choose an integer i uniformly from the set $\{0, 1, 2, ..., N-1\}$. The site connected with this index is the pivot site $x = \omega_t(i)$.
3: Select a lattice symmetry g uniformly from the symmetry group G.
4: Set $\bar{\omega}(k) = \omega_t(k)$ for $k \leq i$, and $\bar{\omega}(k) = g(\omega_t(k))$ for $k \rangle i$.
5: **if** $\bar{\omega}$ is self-avoiding **then**
6: $\omega_{t+1} = \bar{\omega}$.
7: **else**
8: let $\omega_{t+1} = \omega_t$.
9: Goto 2. for the next generation $t := t + 1$.
10: **end if**

Diffusion in a Crowded Environment

Diffusion in a Crowded Environment I

The interior of biological cells represents a very dense and crowded environment with a specific molecular mobility. Intracellular diffusion is hindered by barriers consisting of large molecules sometimes even immobile tethered large molecules, binding and collisional interactions. One way of interpreting such a system is to view it as a disordered system. In general for random walks in media with disordered microscopic substructures one expects anomalous diffusion where the mean-square displacement $\langle \Delta \mathbf{r}(t)^2 \rangle$ no longer is proportional to the time t:

$$\langle \Delta \mathbf{r}(t)^2 \rangle = C_\alpha \; t^\alpha \tag{108}$$

with $C_\alpha > 0$. If $0 < \alpha < 1$, then we call the diffusion sub-diffusive, and if $\alpha > 1$, super-diffusive; normal diffusion has $\alpha = 1$.

In biological applications, the prohibited sites may be more or less mobile biomolecules. Their effect can be taken into account approximately by assuming that also a prohibited site i allows the walker to move through, with some probability q_i. The reciprocal probability $1/q_i$ then can be interpreted as the lifetime of the barrier, in the sense that about once during that lifetime the barrier moves away for one time step before returning to that site. Thus we have still a quenched disorder; with annealed disorder where all lifetimes are the same, we have normal diffusion, squared distance

proportional to time, with a diffusivity reduced by the (slowly) moving barriers. We now assume that the probability distribution function $f(q)$ for the q_i is a power law,

$$f(q) \propto 1/q^a \tag{109}$$

with some exponent a between zero and infinity. More quantitatively, for each prohibited site we determine, when it is visited for the first time by the walker, a random number r, homogeneously distributed between 0 and 1, and then fix q_i for that site i as

$$q_i = Ar^{1/(1-a)} \tag{110}$$

with some free parameter A.

Diffusion in a Crowded Environment III

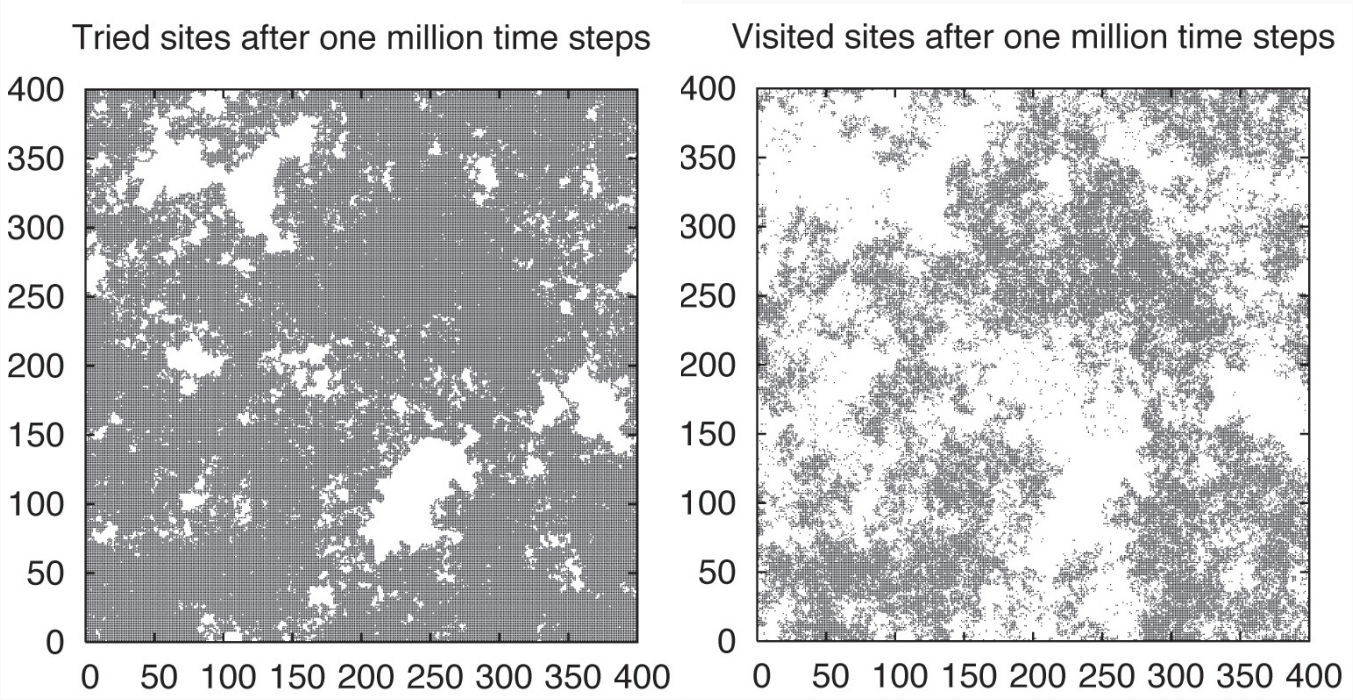
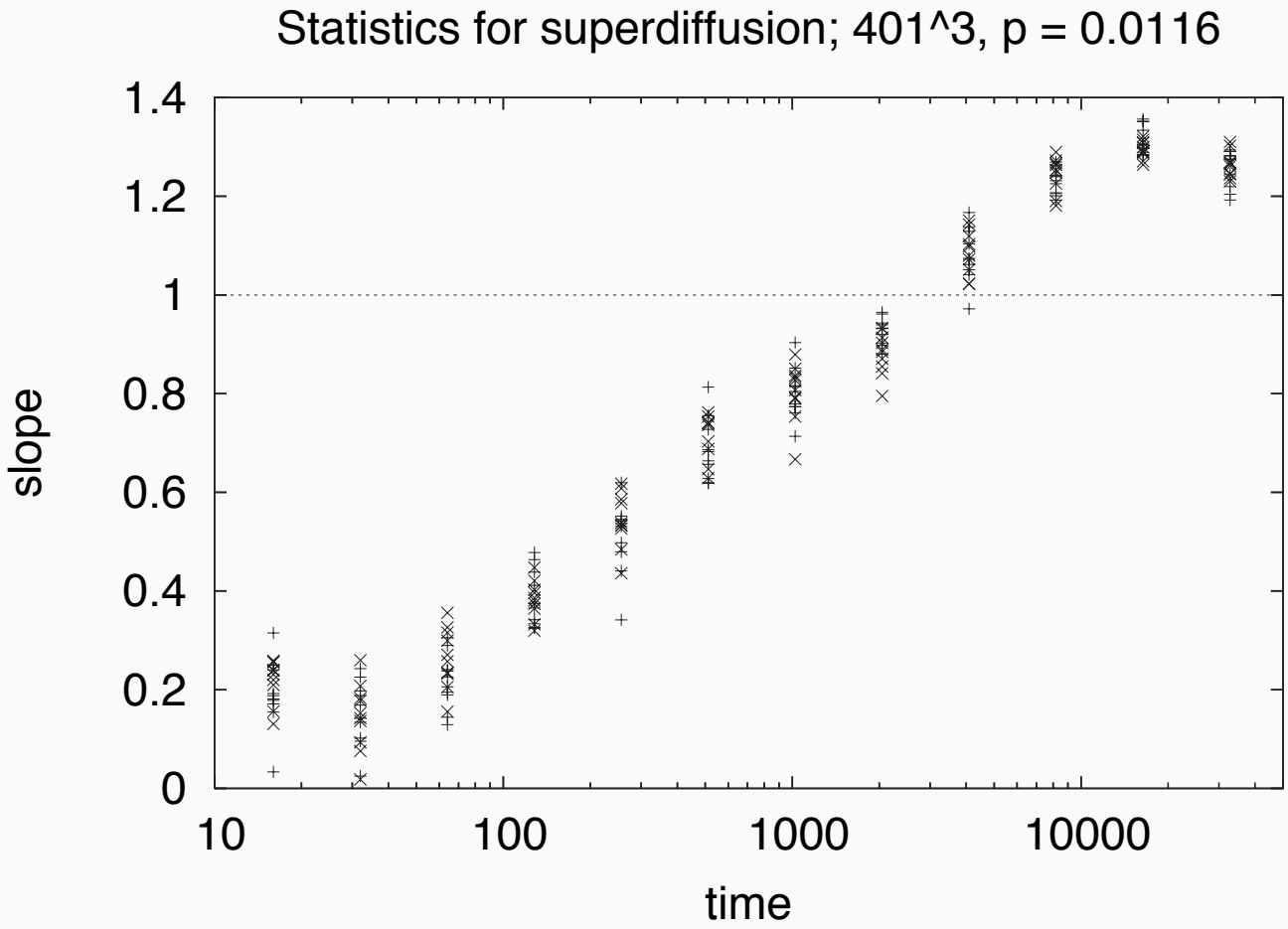


Figure 25: The figure illustrates for two dimensions at $p = p_c - 0.5 = 0.0927$ and exponent $a = 1/2$ the results of one walk after one million time steps. Part a shows the set of sites which have been tried at least once, and part b shows those sites which have actually been visited in spite of the barriers. After 8 million steps, all sites were tried, and after 64 million steps, all sites were visited. One can get anywhere, provided one has enough time.

Diffusion in a Crowded Environment IV



Diffusion in a Crowded Environment V

Figure 26: In three dimensions we have $p_c = 0.3116$. For very small $p = 0.0116$, squares in the figure (middle column bottom), we see an overshooting with an effective exponent α_{eff} above unity at intermediate times; this is not a statistical fluctuation and shows up in all 20 simulated samples. One may call this effect **super-diffusion** since for more than one order of magnitude the exponent is above unity. Basically, the positive probability of each barrier to move away and to let through the random walker means that for sufficiently long times we always get normal diffusion, $\alpha = 1$. For times which are not long enough to see this moving-away of the barriers, but long enough for the walker to explore the whole finite cluster for $p < p_c$ on which it started, we have $\alpha = 0$. For our moderately small $A = 0.01$ these different regimes cannot be reliably separated;

Excercises

Excercises I

Exercise 1: **Random Walk**

Given a Random Walk of length N. What is the probability that the walk returns to its origin, i.e. that a loop of length N has been formed and how does it scale with the length N? (Do not get into proving Polya's theorem! Give a back of the envelop estimation.)

Exercise 2: **Brownian motion in a harmonic potential**

Consider a Brownian particle of mass m in an harmonic potential with spring force k ($\omega_0^2 = k/m$) and random force ξ (Gaussian random process)

$$\frac{dx(t)}{dt} = v(t) \tag{111}$$

$$\frac{dv(t)}{dt} = \frac{\gamma}{m} v(t) - \omega_0^2 x(t) + \frac{1}{m}\xi(t)\,. \tag{112}$$

Develop an algorithm to solve the problem.

Exercise 3: **Geometric Bownian Motion**

The time-evolution of an observation $S(t)$ is said to follow the geometric Brownian motion if the stochastic differential equation holds

$$dS(t) = \mu S(t)dt + \sigma S(t)dB(t) \tag{113}$$

where μ is the drift and σ the standard deviation. Show that

$$S(t) = S(0)\exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B(t)\right)$$

is the solution to Equation 113. Examples for this is behaviour is the motion of pollen grains on still water.

Exercise 4: **Average Crossing Number**

The Average Crossing Number (ACN) of a random walk of length N is the expected value over all possible random walks of length N of the expected value of the crossing number of one of these confirmations over all possible planar projections. Show that an upper bound for the ACN is

$$\mathrm{ACN} \sim N^2 \,. \tag{114}$$

As a matter of fact it can be shown that [15]

$$\mathrm{ACN} \sim N \log N \,.$$

Write a program to compute the ACN.

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Bibliography

References

[1] Vincent Dion, Veronique Kalck, Chihiro Horigome, Benjamin D. Towbin, and Susan M. Gasser. Increased mobility of double-strand breaks requires mec1, rad9 and the homologous recombination machinery. *Nat Cell Biol*, 14(5):502–509, 05 2012.

[2] Yang Zhang and Dieter W. Heermann. Dna double-strand breaks: linking gene expression to chromosome morphology and mobility. *Chromosoma*, 123(1):103–115, 2014.

[3] Karl Pearson. The problem of the random walk. *Nature*, 72:294, 1905.

[4] G. Polya. über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im strassennetz. *Mathematische Annalen*, 84(1-2):149–160, 1921.

[5] A. Einstein. *Ann. Phys.*, 17:549, 1905.

[6] R. Brown. A brief account of microscopical observations made in the months of june, july, and august 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *Edinburgh New Philosophical Journal*, 5:358–371, 1828.

Bibliography II

[7] G. van Kampen. *Stochastic Processes in Physics and Chemistry*. North Holland, Amsterdam, 1981.

[8] P. Rouse. *J.Chem.Phys.*, 21:1272, 1953.

[9] Wolfgang Paul, Kurt Binder, Dieter W. Heermann, and Kurt Kremer. Dynamics of polymer solutions and melts. reptation predictions and scaling of relaxation times. *The Journal of Chemical Physics*, 95(10):7726–7740, 1991.

[10] D. W. Heermann W. Paul, K. Binder and K. Kremer. *J. Chem. Phys.*, 95:7726, 1991.

[11] William Feller. *An Introduction to Probability Theory and Its Applications, Volume 1, 3rd Edition*. John Wiley & Sons, Inc., 1968.

[12] R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford Univ. Press, Oxford, 2001. R. Grimmett, D.R. Stirzaker: P'.obabili'y Q]ld RQ]ldorn Processes (Oxford Univ. Press, Oxford 1987).

[13] Alexei V. Chechkin, Ralf Metzler, Joseph Klafter, and Vsevolod Yu. Gonchar. *Introduction to the Theory of Lévy Flights*, pages 129–162. Wiley-VCH Verlag GmbH & Co. KGaA, 2008.

[14] N. Madras and A.D Sokal. The pivot algorithm: a highly ecient monte carlo method for the self-avoiding walk. *J. Stat. Phys.*, 50:109–186, 1988.

Bibliography III

[15] M. P. Diesinger and W. D. Heermann. Average crossing number of gaussian and equilateral chains with and without excluded volume. *The European Physical Journal B*, 62(2):209–214, 2008.

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Index

Index I