Master Thesis

Diffusion on Fractals

Janett Balg
born 29th March 1983 in Zwickau

Tutor: Prof. Dr. K. H. Hoffmann

Chemnitz, the 21th March 2006
ACKNOWLEDGMENT

I want to thank Prof. K. H. Hoffmann for giving me the possibility to write my master thesis about “Diffusion on Fractals” and for many interesting and helpful discussions. I am especially grateful to Dr. S. Seeger and D. H. N. Anh for helpful discussions. I also want to thank F. Prehl and my family for their support.
**ABSTRACT**

We study anomalous diffusion on fractals with a static external field applied. We utilise the master equation to calculate particle distributions and from that important quantities as for example the mean square displacement $\langle r^2(t) \rangle$. Applying different bias amplitudes on several regular Sierpinski carpets we obtain maximal drift velocities for weak field strengths. According to $\langle r^2(t) \rangle \sim t^{\frac{2}{d_w}}$, we determine random walk dimensions of $d_w < 2$ for applied external fields. These $d_w$ corresponds to superdiffusion, although diffusion is hindered by the structure of the carpet, containing dangling ends. This seems to result from two competing effects arising within an external field. Though the particles prefer to move along the biased direction, some particles get trapped by dangling ends. To escape from there they have to move against the field direction. Due to the bias accelerated particles and the trapped ones the probability distribution gets wider and thus $d_w < 2$. 
CONTENTS

1. Introduction .................................................. 6

2. Modelling the diffusion phenomena ....................... 9
   2.1 Random walk method .................................. 9
   2.2 Fractals ................................................... 11
   2.3 Random walks on fractals .............................. 13
   2.4 The master equation approach ........................ 16
   2.5 The model of the external field ....................... 17

3. Implementation of the master equation .................... 20
   3.1 Randomised SIERPINSKI carpets ...................... 20
   3.2 Parallel implementation ............................... 25
   3.3 Implementation of the external field ................ 25
   3.4 Test cases ............................................... 26
      3.4.1 The EUCLIDEAN lattice ........................... 27
      3.4.2 The regular SIERPINSKI carpets ................ 27

4. Results ......................................................... 33
   4.1 Model parameters ...................................... 33
      4.1.1 The bias parameters ............................... 33
      4.1.2 The set of generators ............................. 34
   4.2 Numerical Results ..................................... 35
4.2.1 Mean Value 35
4.2.2 Mean square displacement 41
4.3 Discussion 43
4.3.1 Mean square displacement 43
4.3.2 The drift behaviour 50
5. Conclusion 52
Bibliography 53
Selbstständigkeitserklärung 58
1. INTRODUCTION

The purpose of this study is to show effects of a static external field on diffusion in disordered systems and to explain them. Such diffusive processes are physical transport processes. They are of great interest in the natural sciences and have many applications. Important examples are dialysis in medicine, diffusion pumps in vacuum technology, osmosis in biology, or the diffusion of atoms in crystalline materials as described in solid state physics.

Diffusion is the process of equalisation of concentrations caused by the motion of many small particles like atoms, molecules or ions in gases, liquids or solids. The particles move due to their kinetic energy, depending on the temperature. The resulting rectilinear motion is disturbed by collisions with other particles. This results in a trembling motion, that can also be observed for larger particles as for instance dust or clusters of molecules. This motion of particles in gases and liquids is called the BROWNIAN motion.

A simple model for simulating diffusive phenomena is the random walk approach. A random walker can be regarded as a diffusing particle, performing a random motion, similar to the BROWNIAN motion, on an appropriate discrete lattice in discrete time steps. However, diffusion then is a stochastic process of many moving particles. So we have to simulate not only one diffusing particle, but a large number of particles.

Both, the diffusive process and its simulation, can be characterised by the time development of their mean square displacement $\langle r^2(t) \rangle$. Here $r$ is the distance a particle has moved in time $t$ from its starting point. In the case of diffusion in a homogeneous medium, like in the above mentioned examples, the mean square displacement is proportional to $t$. This is referred to as normal diffusion.

But observing for example diffusion of water in biological tissues [01], hydrogen diffusion in amorphous metals [02], or diffusion in disordered systems [03, 04], such as porous rocks or cement, we encounter the so-called anoma-
1. Introduction

Anomalous diffusion is in general slower than normal one due to the fact that it occurs in structured media containing dangling ends, bottlenecks and backbends \[03\]. Particles move for example into such dangling ends and in order to escape they have to move backwards. By that the particles have delay times. So the motion is hindered. Nevertheless the time evaluation of \( \langle r^2(t) \rangle \) also satisfies a power law similar to normal diffusion. It is characterised by:

\[
\langle r^2(t) \rangle \sim t^\gamma
\]

(1.1)

with the anomalous diffusion exponent \( \gamma \neq 1 \). In the case of \( 0 < \gamma < 1 \) the diffusion process is called subdiffusive and for \( 2 > \gamma > 1 \) it is superdiffusive. However only subdiffusive processes and their simulation are discussed in this work.

So, in order to simulate subdiffusive processes, the complicated structure of the material has to be taken into account. As B. Mandelbrot expressed with the following words: “Clouds are not spheres, mountains are not cones, coastlines are not circles, bark is not smooth, nor does lightning travel in a straight line.” \[05\], classical geometric objects are not an appropriate way to describe such substrates. A more suitable method is the use of fractals to model these structures. In this study we consider a special class of two-dimensional fractals, the Sierpinski carpet family. They will be used to generate fractal patterns in order to capture the randomness of disordered materials \[06\].

The diffusion in disordered materials can be simulated in the same way as in uniform structures, by performing random walks on an appropriate lattice. However now the lattice also includes sites where the random walkers are not allowed to move, similar to real media. These lattices will be constructed on the basis of Sierpinski carpets.

For the characterisation the structure of disordered materials we can analyse a lot of structure properties, as porosity, the shortest path between two points or the properties of external boundaries. But how does structure influence diffusion? An important quantity to characterise the dynamics of random walkers on fractals is the so-called random walk dimension \( d_w \). It can be determined by the time scaling of the mean square displacement \( \langle r^2(t) \rangle \) as the anomalous diffusion exponent is related to \( d_w \) by \( \gamma = \frac{2}{d_w} \) \[03\].

To determine the random walk dimension, we can calculate the mean square
displacement $\langle r^2(t) \rangle$ of the distribution of diffusing particles. However for this we have to know the probability of a walker to be at site $i$ of the lattice at time $t$. The whole probability distribution will be approximated by frequencies, as we perform an ensemble of random walkers. But due to this approximation we obtain fluctuations, as random noise in the distribution. To overcome this handicap, we apply the master equation approach. With this method we obtain the complete time dependent probability distribution $P(i, t)$ for each time step iteratively.

The main focus of this study is the behaviour of anomalous diffusion influenced by an external field. Examples for such situations are impedance spectroscopy measurements as for polymer electrolytes [07], hopping electron conduction exposed to doped semiconductors in strong external electric fields [08, 09, 10], diffusion of particles in gels under high gravity or centrifugal force as in chromatographic columns [11]. Furthermore each diffusion on earth occurs in an external field, the gravity.

In chapter 2 we present the model to simulate the anomalous diffusion in external fields. For that we give a short introduction to the random walk method and we lead on to the master equation approach, that we apply. Afterwards we explain our model for the external field and show its association to the master equation. In section 2.2 we show the application of Sierpinski carpets for modelling complicate structures of real materials.

The implementation of the master equation and the external field is given in chapter 3. This has to be done effectively, in order to reach sufficiently large simulated times. The main problem is the huge amount of memory needed to calculate $P(i, t + 1)$. As for this we depend on $P(i, t)$ and all probabilities to move from one site $i$ of the lattice to a neighbouring site $j$.

In chapter 4 we analyse different types of Sierpinski carpets in different external fields. We want to show reasons for the slowing down of diffusion and its dependence on certain field parameters. Thereby we will discuss the drift behaviour of the diffusing particles by the mean values in the different space directions. The changes of the dynamical properties will be observed by the mean square displacement $\langle r^2(t) \rangle$ and the random walk dimension $d_w$. That we will discuss at the end. Finally we conclude our results in chapter 5.
2. MODELLING THE DIFFUSION PHENOMENA

To simulate diffusion in disordered materials we use the model of the random walker. For this we apply the method of the master equation approach. Due to this we introduce the random walk method for the Euclidean space and then we show the relations to the master equation approach. To model disordered structures for our simulations, fractal geometries, such as the Sierpinski carpet are employed to generate the appropriate lattices. Finally we explain our model of the external field $b(t)$ and how it enters the master equation.

2.1 Random walk method

Observing a small particle under the microscope in a liquid, we find that it seems to perform a ‘random’ motion through the space. This trembling motion results on the one hand from the kinetic energy a particle inevitably possesses at a certain temperature and on the other hand by the interaction with other particles. It is the so-called Brownian motion. However, watching many particles this microscopical motion leads to macroscopic phenomena, as for example to pressure and to diffusion. Thus a simple model for diffusion is the random walk method. A random walker moves on a discrete lattice in discrete time steps randomly from one point to another. So it performs such a “trembling” motion. In order to simulate the diffusive process we have to run many of those random walkers. At the end we make a statistic over the random walkers and so we can analyse diffusion numerically.

Diffusion in homogeneous materials, represented by the Euclidean space, can be simulated by the use of a discrete regular lattice. This lattice consists of an (infinite) number of tiles, where each tile has four nearest neighbours,
2. Modelling the diffusion phenomena

Fig. 2.1: On the Euclidean lattice the random walker (•) can move to each neighbour with a probability $\Gamma_{ji}$. Thereby it hops from midpoint to midpoint, depicted by the smaller dots (•).

shown in Fig. 2.1. Two tiles are termed neighbours if they coincide in one edge. The random walker moves in each time step from the midpoint of a tile $i$ to one of its neighbours $j$. It can choose each neighbour with a probability of $\Gamma_{ji}$ or it can stay on its current position with $\Gamma_{ii}$. The sum of the transition probabilities over all neighbours is $\sum_j \Gamma_{ji} = 1 - \Gamma_{ii}$.

Simulating many random walkers, all starting from the same position, we get a frequency distribution after each time step, where the random walkers have moved to. This frequency distribution becomes the probability distribution $P(i, t)$, if we perform an infinite number of random walks. Thus $P(i, t)$ represents the probability of a walker to be at time $t$ on tile $i$. As it is not possible to simulate an infinite number of random walkers we run just a huge number of them. This allows to take the frequencies as a good approximation for the probability distribution $P(i, t)$. However, due to this there might be fluctuations which becomes smaller the more walkers we simulate.

Analysing the time development of the probability distribution, in the case of uniform materials we obtain a Gaussian distribution “spreading with time”. Its mean square displacement is

$$\langle r^2(t) \rangle \sim t. \quad (2.1)$$

In comparison with equation (1.1) the anomalous diffusion exponent $\gamma$ equals 1, that means normal diffusion is given. In this study we want to analyse the diffusion in disordered materials. Therefore we have to construct inhomogeneous lattices, that represent the disordered structures of real materials. So in the following we illustrate our model of generating such lattices.
2. Modelling the diffusion phenomena

2.2 Fractals

As pointed out in the introduction, classical geometries are not an appropriate way of describing the structure of disordered materials. Despite their random nature, natural structures exhibit the property of self-similarity at certain length scales \[12\] and they contain structures like dangling end, back-bends and bottlenecks \[03\]. So we can use fractals for constructing these complicate structures of real materials.

In this work we apply the Sierpinski carpet family as a structural model for real materials. This special kind of fractal is constructed from a so-called generator (see Fig. 2.2(a)), which is two dimensional in the present work. A generator is a square or tile divided into \(n \times n\) congruent subsquares, where \(m\) of these ones are “black” and the remaining \((n^2 - m)\) are “white”, that means removed. Generating a regular Sierpinski carpet we apply the following construction algorithm. First we start with a tile in two-dimensional space. Then we divide it in \(n \times n\) smaller subtiles and label \(m\) of them “black” according to the generator and the rest \((n^2 - m)\) “white” ones are removed. Each of the remaining tiles is again divided according to the fractal pattern. If this construction procedure or so-called iteration is repeated ad infinitum the limit object is a regular Sierpinski carpet. In Fig. 2.2 the first two iteration steps are shown.

To describe the structure of fractals, like Sierpinski carpets, we can apply the so-called Hausdorff-Besicovitch or fractal dimension \(d_f\). The fractal dimension \(d_f\) expresses the scaling of the mass by a factor of \(m\) of the fractal...
pattern when its linear length is scaled by a factor of $n$

$$d_f = \frac{\ln m}{\ln n}.$$  \hfill (2.2)

In the case of the Sierpinski carpet the linear length is the length of one side of a square within the fractal pattern. Furthermore we can interpret the number of black squares within the generator as mass $m$ of this pattern. Now we calculate the fractal dimension $d_f$ for the generator shown in Fig. 2.2. We start with a black square, that we scale by the factor $n = 3$. Applying the generator, we have only seven black squares left. So the mass scales by the factor $m = 7$. Thus the fractal dimension is $d_f = \frac{\ln 7}{\ln 3} = 1.77$.

So far we considered fractals as objects of infinite depth. That means, if we enlarge a subset of the fractal we find the fractal pattern at each length scale. However, real materials are self-similar only at certain length scales. For large length scales we can observe that the structure of real materials is rather homogeneous. To generate this effect, we are not constructing a lattice consisting just of one iterator of a given depth, but we create several iterators that will be combined to one lattice. But we also regard a smallest length scale with is at least limited by the size of atoms and molecules. This will be modelled by a maximum iteration depth $l$. Such finite Sierpinski carpets are named pre-carpets or iterators of level $l$. So finally we have a so-called repeated Sierpinski carpet of an finite depth. It exhibits a structure that is self-similar or scale invariant up to the linear length of such an iterator and at larger length scales the structure is translationally invariant \cite{13, 14}.

Taking the complicate structure of natural materials into account, we can use disordered fractal lattices for the simulations \cite{06}. The construction differs only a little from regular fractals. At first a square is divided into $n \times n$ congruent squares, where $m$ squares are “black” and $(n^2 - m)$ are “white”. Then the “white” ones will be removed, corresponding to the generator that was taken. Each remaining “black” one, will be divided into $n \times n$ subsquares again. But now different, randomly chosen generators are the basis of labelling the $n^2$ squares “black” or “white”. The whole procedure is again repeated ad infinitum or up to a desired iteration depth, as depicted in Fig. 2.3.

Furthermore Sierpinski carpets have the property of being finitely or infinitely ramified. That means, if a subset of a carpet can be separated from the rest just by cutting a finite number of connections, then the carpet will be
2. Modelling the diffusion phenomena

\[ p_1 = \frac{1}{3} \]
\[ p_2 = \frac{1}{4} \]
\[ p_3 = \frac{1}{5} \]

\((a)\) \hspace{1cm} \((b)\) \hspace{1cm} \((c)\)

Fig. 2.3: The first two construction steps ((b) and (c)) of a random Sierpinski carpet are pictured. The carpet is constructed by three different generators (a) chosen with equal probability.

... finitely ramified. Otherwise it would be an infinitely ramified one. Whether the emerging regular carpet is finitely or infinitely ramified can be seen directly from the generator. If the first and the last column coincide in exactly one black square the carpet will be finitely ramified. The same must hold true for the first and the last row. But combining different generators it is not possible to predict in any case the ramification of the resulting carpet.

... characterising fractals and by this the materials more detailed, we can introduce further structural properties of fractals. They can be characterised for example by new dimensions, like the fractal dimension \( d_f \). Amongst others there are the chemical dimension \( d_l \) \([13, 15]\) and the porosity \( d_p \) \([16]\). \( d_l \) describes the scaling of the mass with the shortest path, the chemical distance \( l \), whereas \( d_p \) specifies the distribution of holes with different areas and perimeters and expresses the scaling of the hole area (or perimeter) with the linear length. Furthermore we can compute the fractal dimension of the external boundary and the boundaries of internal cavities \([17]\).

2.3 Random walks on fractals

Knowing how to model disordered materials we now turn to the simulation of the diffusive processes on fractal lattices. The same idea as for the Euclidean space can be applied straightforward to disordered systems. In contrast to uniform materials, where the walker can move to each tile, now
2. Modelling the diffusion phenomena

Fig. 2.4: Here two different models for the random walker are shown. In (a) we see the model of the blind ant and in (b) the model of the myopic ant is depicted.

There exist ‘forbidden’ tiles, that are labeled “white”. These “white” tiles represent barriers of the real material.

As mentioned in section 2.1, the random walker on a uniform lattice moves with a transition probability of $\Gamma = \frac{1}{4}$ in each direction. However, on fractal lattices the walker can only hop to allowed neighbouring tiles. So, what happens if the random walker cannot move in a certain direction? In the literature there are basically two different models. They are known as the model of the blind ant (Fig. 2.4(a)) and of the myopic ant (Fig. 2.4(b)), first mentioned by de Gennes in 1976 [18]. He imagined a random walker on structured lattices as little ant in a labyrinth. In the first mentioned model the ant is blind. That means each neighbouring tile $j$ of tile $i$ will be selected with the same probability of $\Gamma_{ji} = \frac{1}{4}$. If the chosen tile belongs to the black ones in the underlying generator the walker moves to this tile. Otherwise it would stay at its current position. In the model of the myopic ant, the walker chooses the next direction only among the allowed neighbours. That means, if a tile has for example, three neighbours shown in Fig. 2.4(b) the probability to hop in one of these three directions is $\Gamma_{ji} = \frac{1}{3}$. Therefore the probability to stay or to move to the forbidden tile equals zero, as again $\sum_j \Gamma_{ji} = 1$ is given.

The dynamical property of diffusion on fractals can be described by the
random walk dimension $d_w$ that is defined via \([1,1]\) as

$$d_w = \frac{2}{\gamma}, \quad (2.3)$$

where $\gamma$ is the mentioned anomalous diffusion exponent. On an Euclidean lattice $d_w = 2$ is given, independent of the dimension of the Euclidean space \([0,3]\). But on fractals normally $d_w > 2$ corresponding to a slowed down diffusion. This is correlated to the delay of the diffusion particles in the dangling ends, bottlenecks and backbends existing in disordered structures.

There are different methods to compute $d_w$. On the one hand it can be calculated analytically either through a renormalisation scheme \([0,3,19]\) or via the Einstein relation \([20]\)

$$d_w = d_f + \zeta. \quad (2.4)$$

Here $\zeta$ is the resistance scaling exponent of the fractal. Equation \((2.4)\) follows from the strong connection between diffusive processes on some substrate and the current flow through an adequate resistor network \([21,22]\). But for these methods fractals in general need to be finitely ramified and as mentioned earlier, it is not sure to predict this for random fractals. Furthermore for the complex structure of fractals, above all randomly mixed ones, it is mostly computationally too extensive to calculate $\zeta$ or $d_f$ by this way.

Another method to determine $d_w$ is to simulate many random walkers in order to approximate the probability distribution $P(i,t)$ by a frequency distribution as mentioned in section 2.1. With given probability $P(i,t)$ to be at time $t$ at the position $i$, we can calculate the corresponding mean square displacement $\langle r^2(t) \rangle$ as

$$\langle r^2(t) \rangle = \sum\|r_i\|^2 P(i,t) \quad \text{with} \quad \|r_i\|^2 = (x_i - x_{r_0})^2 + (y_i - y_{r_0})^2,$$

\((2.5)\)

where $\|r_i\|$ is the distance of one random walker between its starting point $r_0$ to tile $i$, where it moved to after time $t$. $(x_{r_0}, y_{r_0})$ are the coordinates of $r_0$ and $(x_i, y_i)$ of tile $i$.

Plotting $\langle r^2(t) \rangle$ against time $t$ on logarithmic scales we get a straight line. So we find that a power law for the time dependence of the mean square displacement is given. Approximating the slope of the graph by linear regression
we obtain \( d_w \). Interestingly it appears that we get the same \( d_w \), whether we use the model of the blind or of the myopic ant.

Summarising the method of the random walk approach, we observe single trajectories of many particles in order to simulate the diffusion phenomena. However the number of possible paths for a particle increases exponentially for longer time scales. So we need a huge number of random walkers to obtain a good approximation for the important quantities, like the mean square displacement \( \langle r^2(t) \rangle \) or the random walk dimension \( d_w \). This handicap can be overcome by the exact enumeration method, also known as the master equation approach.

### 2.4 The master equation approach

The master equation gives the complete time-dependent probability distribution \( P(i, t) \) of all possible trajectories for each time step. Using this \( P(i, t) \) for further calculation, we can be sure to avoid fluctuations caused by a limited number of random walks.

The master equation is an iterative way of calculating \( P(i, t + 1) \) by a given distribution \( P(i, t) \). For the determination we require the probability distribution at time \( t \) and all transition probabilities \( \Gamma \) between two neighbouring tiles. That includes the transition probabilities to move from tile \( i \) to \( j \) and vice versa, but also the probability \( \Gamma_{ii} \) to stay at the current position. For the model of the blind ant \( \Gamma_{ii} \geq 0 \), whether for the myopic ant it is \( \Gamma_{ii} = 0 \) given.

All together the master equation has the following structure:

\[
\begin{align*}
P(i, t + 1) &= P(i, t) + \sum_{j \neq i} \left[ \Gamma_{ij} P(j, t) - \Gamma_{ji} P(i, t) \right] \\
&= (1 - L_i) P(i, t) + \sum_{j \neq i} (G_{ij} P(j, t))
\end{align*}
\]  

The factor \( L_i \) represents the overall loss of tile \( i \), that means \( L_i = \sum_j \Gamma_{ji} \) and \( G_{ji} \) are its gain factors. As all variables of \( P(i, t), L_i \) and \( G_{ij} \) are probabilities, \( P(i, t), L_i \) and \( G_{ij} \in [0, 1] \). To obtain a solution an initial distribution for a starting time \( t = t_0 \) has to be defined. We used for example the delta
distribution $\delta_{ir_0}$ with some starting point $r_0$ with

$$
\delta_{ir_0} = \begin{cases} 
1 & \text{for } r_0 = i \\
0 & \text{for } r_0 \neq i 
\end{cases}.
$$

(2.8)

Although we need a lot of information to calculate the probability distribution, the master equation has the advantage that a statistical average over many random walkers is not necessary any more. Applying equation (2.7) and (2.5) we can calculate an accurate mean square displacement and the random walk dimension $d_w$. Furthermore the movement of the mean of the distribution can be observed for every time step. Additionally the probability distribution function contains much more information than just $d_w$. Among other things $P(i, t)$ can be compared with different anomalous diffusion descriptions like fractional differential equations [23, 24, 19].

In order to implement the master equation approach different difficulties have to be improved. Computing the new probability distribution $P(i, t + 1)$ we depend on the entire probability distribution and on all neighbouring relationships at time $t$. Therefore we have a huge demand of memory. Furthermore for each tile the master equation has to be solved. So an effective and efficient implementation has to be done for storing and calculating all necessary information. We apply a parallel implementation, as explained in [25] for regular Sierpinski carpets. In chapter 3 we present the extensions for the randomised Sierpinski carpet [26] and for the external field.

### 2.5 The model of the external field

If we apply an external field $b(t)$ the particles have a certain biased direction in which they prefer to move. That is caused by a higher probability to move along the field than in the opposed direction. Applying a constant bias in a uniform and ordered medium this results in a drift velocity in direction of the field. This increases proportional to the field strength. In the case of disordered materials the diffusion is influenced by the structure of the lattice. So if we obtain a drift velocity $v_{dr}$ it might respond nonmonotonically to the corresponding external field [27].

The effect of an external field $b(t)$ can be taken care of by modified transition probabilities. For the determinations we split the bias vector $b(t)$ into its $x$-
2. Modelling the diffusion phenomena

and $y$-components $b_x$ and $b_y$. With these parameters we want to calculate the modified transition probabilities for each directions independent from other directions for all neighbouring relationships. Therefore we restrict to the model of the blind ant, because there this condition can be realised relatively easy.

The modified transition probabilities including the bias can be defined for our model as

$$\Gamma_{ij} = \frac{1}{2d} \left( 1 + e_{ji} b(t) \right) \quad \text{with} \quad b(t) \in [0, 1]. \quad (2.9)$$

The variable $d$ is the space dimension and $e_{ji}$ are the unit vectors pointing in the four direction to the neighbouring tiles $j$. If there is no neighbour in a certain direction, $\Gamma_{ij}$ and $\Gamma_{ji}$ are zero. The probability to stay at the current position $\Gamma_{ii}$ is determined as

$$\Gamma_{ii} = 1 - \sum_{j \neq i} \Gamma_{ji} = 1 - L_i \quad (2.10)$$

over all existing neighbours.

As shown in Fig. 2.4(a) for the blind ant the transition probability to move in a certain direction is $\Gamma_{ji} = \frac{1}{4}$ in the case without a bias. Applying the modified transition probabilities (2.9) and (2.10) it is always met that each $\Gamma_{ji}$ corresponding to tile $i$ in each direction $e_{ji}$ can be calculated independent from the other directions. For one neighbouring relationship the new transition probabilities are pictured in Fig. 2.5(a).

In the remainder of this work we investigate the influence of different bias amplitudes on the diffusion behaviour on fractals. In order to analyse the
effects of different structural properties the external field will be applied to several fractal generators and iteration depths. Important quantities that will be observed are besides others the mean value, the variance in the different space directions, and the mean square displacement of the probability distribution.
3. IMPLEMENTATION OF THE MASTER EQUATION

We want to utilise the master equation in order to analyse anomalous diffusion in presence of an external field $b(t)$. Therefore we need an efficient implementation to minimise the memory requirement of the calculation and the computing time. We have to cope with a huge memory demand due to the iterative structure of the master equation (2.7) because we depend on the entire probability distribution $P(i, t)$ and all transition probabilities $\Gamma$ to compute $P(i, t + 1)$ (see sec. 2.4). We employ the implementation described in [25]. This C-based programme presents an algorithm for regular SIERPINSKI carpets based on the dynamic storage of $P(i, t)$. This can also be applied straightforward to random carpets. Yet a lot of the simplifications which can be used for regular carpets do not work for randomised ones. Hence a short introduction to the implemented programme for randomised SIERPINSKI carpets [26] and to our extensions for the bias follows.

3.1 Randomised SIERPINSKI carpets

For the implementation different problems have to be solved. Inter alia we want to iterate the master equation in a simple loop without any branches. That is important to minimise the effective calculation time. Furthermore we want to apply a method to store the carpet dynamically to keep the data storage as small as possible. Using regular SIERPINSKI carpets the whole carpet structure is known by one single iterator of a certain depth. However employing randomised SIERPINSKI carpet, each iterator is different. So the connecting points between the iterators are unknown until all iterators are constructed. Furthermore we want to organise a data structure for each iterator so that the new propability values of each iterator can be computed.
3. Implementation of the master equation

locally. So we have the opportunity to parallelise the programme. Therefore we also need information about the boundary structure of neighbouring iterators in order to determine the transition probabilities correctly.

The starting point of the carpet is chosen within a pre-allocated iterator of level \( i \) (see sec. 2.2) that is initialised with a delta distribution \( P(i, t_0) = \delta_{i r_0} \). That is done in order to keep the memory demand as small as possible. Afterwards the whole carpet is described by a linked list, connecting the iterators one after another. Thereby every iterator stores topological information and probability values of the current time step and the one before. In order to iterate the master equation (2.7) the complete list is traversed once every time step.

If we want to calculate the new probability distribution of an iterator local, different types of allowed tiles have to be distinguished. First there are body tiles which are inside the iterator but not adjacent to a boundary. The tiles, that are inside the iterator and adjacent to a boundary are called boundary tiles. Finally the halo tiles are outside the iterator but adjacent to a boundary tile. Fig. 3.1 illustrates this situation by showing two iterators with their body (■), boundary (●) and halo (□) tiles.

The body tiles can be regarded as ‘uncritical’ due to the fact that all required information is stored in the corresponding iterator in order to update the probability distribution. So the transition probability values \( \Gamma_{ji} \) that we depend on can be easily calculated as all neighbouring tiles \( j \) and their probabilities are known locally. However, the boundary tiles are so-called ‘critical’ tiles. First for the boundary tiles the surrounding is not known at the time of initialisation. And second the probability value(s) at the appropriate halo tile(s) have to be given in order to perform the update for a certain tile. We

**Fig. 3.1:** Two adjacent iterators of level 3 with their body tiles (■), boundary tiles (●) and the first halo tile (□) ring.
initialise the distribution by a delta distribution, hence the probability values at the boundary and the halo tiles are zero. So the appropriate terms in (2.7) vanish initially. Only as soon as the random walkers reach the boundaries of the iterator, we have to assure that the neighbouring iterators are present in the next time step. Due to the fact that the probability values become non-zero we need the transition probabilities to calculate the new probability correctly. Therefore the appropriate transition probabilities are determined. Hence the boundary tiles are so-called ‘critical’ tiles. The halo tiles are just needed for computing the boundary tiles, so that we are able to distribute the iteration of the master equation among multiple processes. Therefore we do not have to update them by iterating (2.7). Instead the recent values are copied while updating the corresponding neighbouring iterator in every time step.

Furthermore we have to find an efficient way for enumerating the different kind of tiles by using the same scheme for all tiles. Thereby we can compute the master equation in a simple inner loop without any branches. Therefore we enumerate first the body tiles, then the boundary tiles and finally the halo tiles. For each class we sort the tiles first by their $y$-coordinate, and if those are equal, by their $x$-coordinate, both in ascending order. Concluding we have local tile indicies to each iterator, demonstrated in Fig. 3.2 for an iterator of level $l = 1$.

Afterwards we calculate for each tile a set of flags, which neighbour is present, the appropriate neighbouring tile indicies and the corresponding gain and loss factors of (2.7). If a neighbouring tile is not present its tile index is

---

**Fig. 3.2:** For a Sierpinski carpet generator (iterator level $l = 1$) the local tile indicies are pictured. The numbering starts with the body tile $(00)$, followed by the boundary and halo tiles. The index of non-existing tiles is set to $-1$. 

---
set to $-1$ and the appropriate gain factor to zero. The probability values of $P(i,t)$ and $P(i,t+1)$ are stored in two large pre-allocated onedimensional arrays $p0$ and $p1$, where the tile indicies correspond to the array indices. hen the master equation can be solved for each time step in a simple inner loop, if the gain factors corresponding to non-existent tiles are set to zero.

While iterating the master equation we want to calculate the contribution of each iterator to the mean square displacement of the probability distribution. Thus we have to store:

- the indices of the neighbours, gain and loss factors,
- the number of body, boundary and halo tiles,
- the index to the allocated memory in the $p0$ and $p1$ arrays,
- the tile coordinates local to the iterator origin,
- the iterator origin coordinates and
- administrative information required for a parallel implementation,

for every iterator of the carpet.

Up to now we have only discussed the storage of information regarding one single iterator. However the carpet has an overall structure. For this, it is important to know which iterators are present. Where are the connection points between the iterators? And also which iterators are side by side? So we need a bookkeeping method that handles the global structure of the carpet. From this we obtain the neighbour relationships of each iterator. Furthermore the bookkeeping method has to provide an algorithm to create new iterators, if the probability distribution reaches the boundaries of the current carpet structure, that means the probability values at the boundary becomes non-zero. The new iterators have to be integrated in the existing carpet structure. That means, the boundary topology of all adjacent iterators has to be updated, including neighbouring indicies, gain and loss factors.

In order to determine the loss factors of each tile we require the neighbour relationship of this tile. In order to calculate its gain factors we need to know the surrounding of the neighbouring tiles, because the gain factor of a tile $i$ are the loss factors of its neighbours $j$ which depend on the neighbour
3. Implementation of the master equation

Fig. 3.3: The initialisation of the gain (loss) factors of the generator depicted in Fig. 3.2 is shown for the first four tiles 00, 01, 02 and 03. The pull-initialisation is pictured by bold arrows (➡) according to the gain factor $G_{ij} = 0$ and the push-initialisation by the thin arrows (➞) corresponding to $G_{ji} = \frac{1}{4}$.

relationship of tile $j$. But for the boundary tiles we only have the data of the neighbours, because we just store the first halo tile ring. Therefore we create a bitmap to compute the correct gain factors of the boundary tiles. In this bitmap we store the pattern of the neighbours including the first and the second halo ring. Starting with a cleared bitmap we set the iterator pattern according to its construction rules into the bitmap. Then the bitmap is compared with the present neighbouring iterators, matching the halos with the corresponding bitmaps of the iterators. The so generated bitmap is suitable to calculate the boundary gain factors elegantly by a simple loop over all body, boundary and halo tiles.

For the initialisation of all loss and gain factors a ‘push-pull’ method is employed. Due to this method all tiles can be initialised independent of the neighbour relationship by a simple loop over all tiles. First the number of present neighbours $n$ for tile $i$ is determined. Then the corresponding overall loss factor is calculated as $L_i = \frac{n}{4}$ for the blind ant and $L_i = 1$ for the myopic ant algorithm. If the proper neighbour $j$ is present, we ‘push-initialise’ the gain factor $G_{ji}$, otherwise we apply the ‘pull-initialisation’. Pull-initialisation means that the gain factor $G_{ij}$ is set zero and the tile index for this direction to $-1$. However push-initialisation means that the gain factor $G_{ji}$ and the tile index for the opposite direction of the neighbour will be set. In the case of the myopic ant the gain factor has to be $G_{ij} = \frac{1}{n}$ and for the blind ant $G_{ij} = \frac{1}{4}$. Initialising a tile by this method means that we set the gains of all neighbours. Due to that all boundary and body tiles that we depend on for solving the master equation are set correctly if we are doing this for all...
tiles, starting with the body tiles, then the boundary tiles and finally the halo tiles. In Fig. 3.3 this push-pull initialisation is shown for the first four tiles of the corresponding iterator using the local tile indices.

3.2 Parallel implementation

As explained in section 3.1 we store all necessary information in such a structure that we can iterate the master equation for each iterator locally. Due to this fact we can parallelise the programme easily. Such a parallel implementation of the master equation obviously has several advantages. Especially for long calculations, that means long computing times, a huge amount of RAM is required. This requirement can be met by the larger number of processors. Additionally the computing power grows, which results in average in an overall speed up for the calculation.

The parallelisation is done by using a simple master-worker scheme. With this implementation each worker receives a number of iterator descriptions for the iterators they have to process. For that purpose the global list of iterators is split among the \( N \) workers and (2.7) is computed. If the probability values reach a non-zero value at a boundary tile, a message will be sent to the master to extend the carpet and the worker continues to process its iterator list.

The master is responsible for the overall control of the programme. Only there the global carpet structure is known. For each iteration step, the master sends a notice to the workers, to start their calculations and collects the messages about necessary carpet extensions. If the master receives such notices, it extends the global carpet topology after the workers finished their computations. This is indicated by each worker by a message labelled with “calculation done”. The master changes the carpet topology and informs the workers about new jobs. Finally it starts the boundary update between the workers, may optionally collect results or start a new iteration.

3.3 Implementation of the external field

In section 2.5 we already discussed that the effect of a bias can be integrated by modified transition probabilities \( \Gamma \). As we want to investigate constant
but also time dependent external fields, we have to update the new transition probabilities as in (2.9) defined for each time step. However up to now we only have constant gain and loss factors. They are determined just once during a run, when the corresponding iterator is initialised and are not stored for further calculations. In order to update the modified transition probabilities depend on the neighbour relationships of each tile and its neighbouring tiles mentioned in section 3.1. Therefore we store the local neighbourhood of a tile in order to determine the gains and losses effectively in each time step.

We see in (2.9) that the new transition probabilities $\Gamma_{ji}$ consist of two terms, a constant factor and a field dependent one. The constant factor corresponds to the used algorithm. As we apply the model of the blind ant this term is $\frac{1}{4}$. This factor is employed to initialise the gain factors, while creating the iterators. The second term depends on the external field $b(t)$. Due to that it is time dependent and has to be updated each time step. So we call the second factor dynamical gain. The update will be done before the probabilities of time step $t + 1$ are determined by (2.7).

### 3.4 Test cases

To test the correctness of the modified implementation we applied it to three different test cases. At first we consider the Euclidean space. For this case we know the results without a bias, but also in the presence of an external field. Furthermore we choose two different regular Sierpinski carpets each with a generator size of $3 \times 3$ tiles and an iteration depth $l = 5$. For these two fractal pattern we calculate the random walk dimension $d_w$ and compare them with reference values determined by our group.

For all calculations the so-called ‘Riesen’ cluster was used. Each of its 24 nodes contains 8 GB RAM and 2 CPUs with a clock rate of 2 GHz. However the virtual number of CPUs is 4, due to utilisation of hyperthreading.

Typically we need for example to iterate generator C (see fig. 4.1(a)) of size $5 \times 5$ tiles with an iteration depth $l = 4$ for $350000$ time steps $\sim 270$ h.
3. Implementation of the master equation

3.4.1 The Euclidean lattice

For the Euclidean lattice we applied the generator 3.4(a) with an iterator level \( l = 4 \). The run is performed over \( t = 10000 \) time steps with the starting point \( r_0 = (0, 0)^T \). We applied three different amplitudes of bias. First we apply a zero bias in order to show normal diffusion without a bias. Furthermore we employ a weak bias \( b = (0.2, 0.0)^T \) and a strong one with \( b = (0.6, 0.0)^T \), both are directed in \( x \)-direction.

In figure 3.5 the time development of the mean square displacement \( \langle r^2(t) \rangle - \langle r(t) \rangle^2 \) is shown for the different bias parameters. We can observe that all three curves are straight lines each with a slope of 1, thus \( d_w = 2 \). The movement of the probability distribution in \( x \)-direction is plotted in fig. 3.6(a). The mean value \( \langle x \rangle \) is depicted as a function of time \( t \). As expected a drift only occurs for \( b(t) > 0 \). Perpendicular to the field direction we observe no drift, as shown in fig. 3.6(b). So the implementation produces the expected results for the normal biased and unbiased diffusion, as the linear timedependence of the mean square displacement and the drift only along the bias direction.

3.4.2 The regular Sierpinski carpets

Here we compare the results of the implemented master equation with results achieved by the random walk method or resistance scaling for the case of no
Fig. 3.5: It is plotted the logarithmic \( \langle r^2 \rangle - \langle r \rangle^2 \) over the logarithmic time \( t \). The curves reach straight lines with \( d_w = 2 \).

The different generators were chosen as two complementary extremes for the boundary handling code. So the pattern of generator A has no corner tiles while B consist only of boundary tiles. Imagining the resulting iterator out of generator A, it is obvious that it has only four boundary tiles - for each direction one - and otherwise it contains only body tiles. Due to this we can test the correct iteration of the master equation in our implementation for fractal structures. Its fractal dimension \( d_f \) is \( d_f = \frac{\ln 5}{\ln 3} = 1.465 \). The U-shaped generator B is only composed of boundary tiles. So in each time step, we have to perform a lot of boundary updates. It provides to be a good way to check the correctness of the boundary updates between the different iterators. The fractal dimension of the resulting SIERPINSKI carpet is \( d_f = \frac{\ln 7}{\ln 3} = 1.77 \).

In fig. 3.7 and 3.8 the mean values in \( x \)- and \( y \)-direction over time \( t \) are shown for the two generators. As expected the curves according to generator A shows no drift in \( x \)- or \( y \)-direction. That is caused by the symmetry of the
3. Implementation of the master equation

In these two plots the mean values $\langle x \rangle$ in (a) and $\langle y \rangle$ in (b) over time $t$ are shown. Note that only along the the biased direction a drift occur.

generator and due to the fact, that we start our distribution in the middle of the cross. Applying generator B we start at the coordinates $r_0 = (0, 0)^T$. Here we can observe a little drift at the beginning, which is caused by the fractal structure.
3. Implementation of the master equation

Fig. 3.7: The mean value $\langle x \rangle$ over time $t$ for the generators shown in 3.4(b) and (c) with iteration depth $l = 5$.

Fig. 3.8: The mean value $\langle y \rangle$ over time $t$ for the generators shown in 3.4(b) and (c) with iteration depth $l = 5$. 
As mentioned in section 2.3, the random walk dimension can be determined by a linear fit of the mean square displacement over time in a log-log plot. For our two generators this is depicted in fig. 3.9. Besides the $\langle r^2(t) \rangle - \langle r \rangle^2$ plot we also present the best fit for each curve. Fitting the slope of the curves via gnuplot \[29\] we obtain $d_w = 2.4952$ for generator A and for generator B $d_w = 2.1370$.

In section 2.2, we explained that we employ the repeated Sierpinski carpet model because real materials get homogeneous at larger length scales. That means we have to pay attention that we do not include the crossover from anomalous to normal in our fittings. This crossover can be observed at long time scales. Hence we have to know how long it takes until the mass of diffusing particles have passed an iterator. According to (1.1) and (2.3) the average distance from the starting point after time $t$ is

$$\langle r(t) \rangle \sim t^{\frac{1}{d_w}}. \quad (3.1)$$

As we apply iterators of $243 \times 243$ tiles it takes $t \approx 760000$ time steps for 3.4(b) and $t \approx 135000$ for 3.4(c) till most of the particles have crossed one iterator. As we determine only 10000 time steps it has no influence for our
As we want to know how good our calculation of $d_w$ is, we require a reference value for comparison. For generator A we used $d_w = 2.465$ determined by the resistance scaling algorithm in [28]. The random walk dimension of the second pattern was computed with the random walk method by our group as $d_w = 2.150$. Comparing reference values with our results, we find that the error is within the accuracy of our fitting method. So the implemented programme passed all tests.
4. RESULTS

In this chapter we present the results of simulations with a set of different generators for different static bias strengths. We applied the method described in chapter 2 and 3. In order to explain the evaluations we first introduce the external field parameters and the employed generators. Then numerical results are presented and will be discussed.

4.1 Model parameters

4.1.1 The bias parameters

In this work we want to investigate the effects of different static external fields on diffusive processes. To start with we concentrate on static bias and investigate the amplitude dependences. Due to this we omit the time dependence and concentrate on influences of different bias amplitudes of $b$ which are constant in time.

In the following we want to present differences between anomalous diffusion within and without an external field. Consequently we also employ $b = (0.0, 0.0)^T$ besides different bias amplitudes. We also apply $b = (1.0, 0.0)^T$. In this case the particles have no possibility to move against the field direction. However, described in section 3.3, they can walk perpendicular to the direction of $b$. The bias amplitudes in between are chosen with regular distances. We introduce a bias along the $x$-direction with equidistance increasing of the bias strength. According to (2.9) as bias parameters $b$ we used:

$$b \in \left\{ \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 0.2 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 0.4 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 0.6 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 0.8 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} \right\}, \quad (4.1)$$
4. Results

34

Fig. 4.1: Here the four different fractal generators are presented for analysing the diffusion behaviour with an external field applied.

4.1.2 The set of generators

Diffusion depends on the structure on which it takes place. So we have to think about structural properties, that play an important role for anomalous diffusion in external fields. Therefore we search different fractal pattern in order to present different parameters. Nethertheless we want to compare the carpet structures. Thus we select pattern with same size (5 × 5 tiles) and same the fractal dimension ($d_f = 1.5936$). Nonetheless the dynamical properties of each pattern as characterised by the random walk dimension $d_w$ differ.

Due to the structure of the different generators we want to discuss several questions as for example: How do dead ends hinder diffusing particles? Can we also observe drifts perpendicular to the field direction? What is the reason for such a drift? Can the number of connection points in $y$-direction change the diffusion along the bias? Due to these questions we choose the generators depicted in fig. 4.1.

We tried to pick different structural properties. Generator C in fig. 4.1(a) includes no dead ends at iteration level 1. Small dead ends have the generators D and F in fig. 4.1(b) and (d). The largest dangling end is contained in generator E (fig. 4.1(c)). Although pattern D and F exhibit nearly the same structure, generator F has more connection points in $y$-direction. If we apply the regular SIERPINSKI carpet construction it results in an infinitely ramified carpet. Because we want to investigate the influences of dangling end, that trap the particles we choose the dead ends along the bias direction.

Generating a precarpet of iteration depth $l > 1$ we find dead ends in each carpet structure. So we apply apart from higher iteration depths also $l = 1$ to
4. Results

<table>
<thead>
<tr>
<th>Generator</th>
<th>ref. $d_w$</th>
<th>$r = 5$</th>
<th>$r = 25$</th>
<th>$r = 125$</th>
<th>$r = 625$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>2.487</td>
<td>55</td>
<td>2,993</td>
<td>164,065</td>
<td>8,981,598</td>
</tr>
<tr>
<td>D</td>
<td>2.489</td>
<td>55</td>
<td>3,016</td>
<td>165,657</td>
<td>9,097,988</td>
</tr>
<tr>
<td>E</td>
<td>2.599</td>
<td>66</td>
<td>4,298</td>
<td>281,753</td>
<td>18,471,075</td>
</tr>
<tr>
<td>F</td>
<td>2.498</td>
<td>56</td>
<td>3,105</td>
<td>173,013</td>
<td>9,640,694</td>
</tr>
</tbody>
</table>

Tab. 4.1: The average number of time steps to cross the distance $r$ corresponding to the linear size of the iterators at a certain depth

see the differences between dangling end structures and no ones. A further advantage of this iteration level is the corresponding small iterator size. So the time until the crossover from anomalous to normal diffusive behaviour is relatively short. Furthermore we construct iterators of level $l = 2$ to show the effects of fractal patterns for short length scales and $l = 4$ for longer ones. The iterators of $l = 4$ reach a size of $625 \times 625$ tiles. So the property of self-similarity can be found over three order of magnitudes.

At the end we will compare the diffusion on different carpet structures and their iteration levels. So it is helpful to know the average times to cross an iterator of a certain size. In Tab. 4.1 we present these crossover times for each fractal pattern for four distances. Furthermore we display the appropriate random walk dimension calculated by our group with the random walk approach.

Summarising the expressed properties we expect different diffusion behaviours for each generator.

4.2 Numerical Results

4.2.1 Mean Value

The mean value $\langle e \rangle$ describes the average distance the particles have moved in a direction $e \in \{x, y\}$ after a certain time $t$. By this the drift behaviour of the probability distribution $P(i, t)$ can be expressed. It will be determined for the different directions $e$ as

$$\langle e \rangle = \sum_{k \in e} \delta_e P(i, t) \quad \text{with} \quad e \in x, y,$$  \hspace{1cm} (4.2)
where \( i_e \) is the appropriate coordinate of the corresponding tile \( i \).

We will observe that we can have certain drifts even if no external field is present. This is caused by the inhomogeneous structure of the lattice around the starting point \( r_0 \). This can also occur without applying an external field. Applying an external field another kind of drift occurs, caused by the biased direction. This drift is along the external field, however in some cases we can also recognise perpendicular to the field a drift movement. That results from the bias influenced by the lattice structure.

Our results for the mean values for three of the four fractal pattern are shown in Fig. 4.2, 4.3, and 4.4. We plot the mean values \( \langle x \rangle \) and \( \langle y \rangle \) over time \( t \) for the generator C, E, and F for different iteration levels. The results for generator D are quite similar to generator F. The graphs (a), (c), and (e) show the mean values in \( x \)-direction and (b), (d), and (f) present the one in \( y \)-direction. Furthermore (a) and (b) show the behaviour for the iteration depth \( l = 1 \), beneath (c) and (b) for \( l = 2 \) and the last two ((e) and (f)) for \( l = 4 \).

In general we can say that in the cases of no bias we observe only small drifts for all carpet structures at the beginning. As mentioned they seem to correspond to the lattice structure. Applying different external fields we find a long time drift behaviour for all pattern at least in \( x \)-direction. For generator C, D, and F there occurs not only a drift along the bias but also perpendicular to it. In detail we can see that each iterator exhibits another drift behaviour.

For generator C at iteration depth \( l = 1 \) we recognise that the slopes of the curves in Fig. 4.2(a) and (b) increases with increasing bias amplitudes. But for higher iteration levels, and also for the other fractal carpet structure another behaviour can be observed. There the curves corresponding to weak bias amplitudes exhibits the steepest slopes. We can also see that the curves for strong external fields have a very small slope. The extrem case of this behaviour is given by \( b = (1.0, 0.0)^T \). These curves stay nearly constant for \( t > 5000 \) for generator C or \( t > 100 \) for generator E. Just at the beginning a small drift movement can be seen, although the strongest bias amplitude is applied. The exception of this is generator C at iteration level \( l = 1 \), that reaches a straight line.

By the slopes we can determine the mean drift velocities \( \langle v_{dr} \rangle \) of the probability distribution for the different directions \( e \). For this we can also interpolate
the data points of the mean values over time by a function. Differentiating this function we obtain $\langle v_{\text{dr}} \rangle$. We have done this with Mathematica [31]. The results can be seen in Fig. 4.5. There the mean drift velocity $\langle v_{\text{dr}} \rangle$ for each direction is plotted over the bias amplitude $b$ for the different generators. From top to bottom the different iteration level $l = 1, 2, 4$ are depict. We see...
that the largest drift velocity does not occur for the strongest bias amplitude, excluding the mentioned exception for generator C at level $l = 1$. In the plots we can recognise that highest drift velocities can be observed for weak external fields.

Comparing the different graphs for one iterator we can see certain differences.
Fig. 4.4: The mean values of the probability distribution in $x$- and $y$-direction over time $t$ for generator F (Fig. 4.1(d)) for the iteration depth $l = \{1, 2, 4\}$. between the several iteration depths. At iteration depth $l = 1$ drift nearly starts from the beginning. For $l = 2$ we observe that it takes more time until curves reaches straight lines. We see that in the first time steps the drift is quite strong. Then the curves have little ‘break’ and we obtain the linear slope. In the case of $l = 4$ another drift behaviour seems to be present for the fractal pattern. However scaling the curves of $l = 2$ by an factor of two,
we find the similar behaviour of both processes. That means if we compute more time steps, we achieve the same drift behaviour for \( l = 4 \) as we obtain for \( l = 2 \) and \( l = 1 \).
4. Results

4.2.2 Mean square displacement

Now let us examine the curves for the variance of the probability distribution in \( x \)- and \( y \)-direction, but also the overall mean square displacement. In order to analyse the “spreading” of the distribution and not to include the drift movement we determined the central second moment. This is defined as

\[
D^2(X_e) = \langle e^2 \rangle - \langle e \rangle^2.
\]  

(4.3)

In Fig. 4.6 we display the central moments in \( x \)- and \( y \)-direction of generator C. That includes all iteration depths and all \( b \) according to (4.1). The overall mean square displacement is shown in Fig. 4.7. The vertical line represents the average number of time steps when the majority of particles left a certain length scale (for higher iteration level) or the first iterator. So the first line corresponds to the crossing the first generator at \( r = 5 \), the second one for \( r = 25 \) for iterator of level \( l = 2 \) and finally for level \( l = 3 \) the distance \( r = 125 \). We labelled the steps according Tab. 4.

For the mean square displacements as depicted in Fig. 4.6 and 4.7 for \( l = 1 \) we see that after \( t = 55 \) all six curves reaches a slope of 1, according to \( d_w = 2 \). The curves differ just by a constant prefactor that can be merged into the diffusion constant \( D \). The crossover to normal diffusion can be seen very well.

Except for generator C \( (l = 1) \) the mean square displacement of the probability distribution stays constant at larger than \( t > 5000 \) for a bias \( b = (1.0, 0.0)^T \). So it seems to correspond to the structure of the pattern, because only for generator C of level \( l = 1 \) contains no dangling ends. For the central moments of the fractal pattern for stronger field we find a similar behaviour as for \( b = (1.0, 0.0)^T \). For generator E this can be seen clearest, show in Fig. 4.8 for the \( x \)- and \( y \)-direction and from Fig. 4.9 for \( r \). There we also observe that the curve stays nearly constant for a certain time, but then their slope increases (slowly) again.

In \( y \)-direction the diffusion slows down with increasing bias. However for each bias of the bias parameter (except for \( b = (1.0, 0.0)^T \)) the distribution goes over to normal diffusion after leaving the fractal length scale. However in the overall mean square displacement and especially in \( x \)-direction the slope is smaller than 1. This can be observed for \( l = 1 \) but clearer of larger iteration levels. This is illustrated in Fig. 4.10. We picked two bias amplitudes.
Fig. 4.6: The variance of the probability distribution in x- and y-direction over $t$ is plotted for generator C (Fig. 4.1(a)) for the iteration depths $l = \{1, 2, 4\}$.

$b = (0.0, 0.0)^T$ and $\bar{b} = (0.4, 0.0)^T$ for generator F with $l = 2$ and plotted their logarithmic mean square displacement $D^2(X)$ over the logarithmic time $t$. Additionally we show the best fits of these curves for the different length scales. Having a slope $\gamma < 1$ means that $d_w < 2$ according to (2.3). So the diffusive process within an external field seems to be superdiffusive. On
4. Results

4.3 Discussion

4.3.1 Mean square displacement

At first we want to understand the “superdiffusive” behaviour of the biased diffusion. We investigate a so-called marginal distribution \( \tilde{p}(x, t) \) of the probability distributions in \( x \)-direction with MATHEMATICA [31] as

\[
\tilde{p}(x, t) = \sum_y P((x, y), t),
\]

Fig. 4.7: The log-log plot of the mean square displacement \( D^2(X) \) of the probability distribution over time \( t \) for generator \( C \) (Fig. 4.1(a)) are shown for the iteration depths \( l = \{1, 2, 4\} \).

the other hand we know that diffusion is hindered by dangling end and not enhanced. This strange effect we find for all fractal pattern.
Fig. 4.8: The variance of the probability distribution in $x$- and $y$-direction over time $t$ for generator E (Fig. 4.1(a)) is plotted for the iteration depths $l = \{1, 2, 4\}$.

with $P((x, y), t) = P(i, t)$. With respect to the results given in section 4.2 we see that the effect increases with iteration level. Therefore we show the marginal distribution in fig. 4.12 for generator E at the iteration level $l = 2$ for different time steps. With this generator we observed the strongest “superdiffusion”. In order to illustrate the crossover to normal diffusion we
Fig. 4.9: The log-log plot of the mean square displacement $D^2(X)$ of the probability distribution over time $t$ for generator E (Fig. 4.1(c)) are shown for the iteration depths $l = \{1, 2, 4\}$. Also show the marginal distribution for generator C with iteration level $l = 1$ in fig. 4.11.

The marginal distribution $\tilde{p}(x, t)$ for $b = (0.0, 0.0)^T$ is displayed by the red curve and $b = (0.4, 0.0)^T$ by the green one. On the left hand we show $\tilde{p}(x - \langle x \rangle, t)$ over $x$ where $\langle x \rangle$ is the mean value of $\tilde{p}(x, t)$ at time $t$. On the right hand the marginal distribution, directly determined by the probability distribution is given. The different time steps ($t = 0, 50, 100, 489$) are beneath each other.

The red curve is a relatively symmetric similar to a Gaussian curve for each time step. However for the bias influenced marginal distribution that is not the case. In the graphs 4.12(a) and (b) the initial distribution is shown. As we initialise both distributions in same way the curve fall on each other. But for longer times we observe that the green curves get asymmetric. That
Fig. 4.10: In (a) the log-log plot of the mean square displacement $D^2(X)$ over $t$ is shown for generator C ($l = 2$) for $b = (0.0, 0.0)^T$ and $b = (0.0, 0.0)^T$. Further the corresponding best fits for $t \in [0 : 1000]$ and $t \in [10000 : 0]$ with the appropriate $d_w$ are plotted. In (b) the same is presented for the generator E ($l = 2$).

means $\tilde{p}(\langle x \rangle, t) \neq \max(\tilde{p}(x, t))$ or in other words the maximum of the curve
Fig. 4.11: The reduced probability distributions $\tilde{p}(x^*, t)$ with $x^* = x - \langle x \rangle$ and $\tilde{p}(x, t)$ is plotted over $x$ for generator C (Fig. 4.1(a)) ($l = 1$).
4. Results

Generator E – Boundary Distribution in x–direction – l=2

Fig. 4.12: The reduced probability distributions $\tilde{p}(x^*, t)$ with $x^* = x - \langle x \rangle$ and
$\tilde{p}(x, t)$ is plotted over $x$ for generator E (Fig. 4.1(c)) ($l = 2$).
is not in the middle of the distribution. It is on the left hand of $\langle x \rangle$. From the peak to positive $x$ values we see a strong decay of the probabilities value to $\tilde{p}(x, t) = 0$. For long time scales the distribution asymptotically becomes a symmetrical shaped curve. This can be seen in fig. 4.11.

The obtained curves can be explained in following way: Due to the bias the particles move along the bias direction, if this motion is not hindered by the lattice structure. Due to that movement along the bias the particles cross the iterator “faster” than without a bias. In general by (4.2) the drift motion is left out in the mean square displacement. However due to the structure of the fractal lattice particles also move into dead ends or against “a brick wall”, lying along the bias. In order to escape they have to move against the bias direction or they just keep staying on their current position. So a lot of particles experience a “delay”. This delay depends on the length of the dead ends and on the strength of the bias, because the probability to get out is bias dependent. In the case of a bias amplitude $\mathbf{b} = (1.0, 0.0)^T$ it is impossible to escape as the transition probabilities to walk opposed to the field is zero. Due to this the mean square displacement for carpet structures with dead ends stays constant for longer times applying $\mathbf{b} = (1.0, 0.0)^T$, because all particles are trapped in dead ends after a certain period of time.

So on the one hand we have the fast particles moving along the bias which do not get trapped. On the other hand there are particles running into dead ends. To escape they have to move against the bias. Due to these competing effects the probability distribution gets wider displayed by the marginal distribution. It seems to be that the maximum peak represents the huge number of trapped particles and the strong decay the free moving ones but we have not proof it up to now.

However for long time scales the distribution reaches local equalisation. Furthermore the mass of particles cross at least one iterator and so there is some averaging over different structures. That is why the marginal distribution achieves a symmetrical shape and diffusion becomes normal.

Summarising we know that the mean square displacement explains the behaviour of anomalous diffusion without a bias very well. Also for biased diffusion it gives a global description of the diffusion behaviour of the distribution. However due to the bias and the structured lattice it is not enough just to determine (4.2) to obtain the mean square displacement uninfluenced by the external field. So for further investigations we may search for other
4. Results

possibilities to describe the biased diffusion in a more proper way. Thereby it seems to be important to analyse the hopping of particles in dead ends and the resulting consequences in more detail. One possibility we can try a new approach by so-called fibres and clouds \cite{32,33,19}. This method uses the natural similarity group of a walk to find interesting structures in the random walk behaviour.

4.3.2 The drift behaviour

The drift in $x$-direction is understood as the direct result of the bias though the external field in this direction. However for the generators C, D, and F we also find a drift in $y$-direction. This can be seen as a direct result of the bias in combination with the fractal pattern. This observation is confirmed by comparing the generators with each other. All three generators have a triangle like structure. The basic pattern of all four generators is a L-shape, consisting of an “$y$” and “$x$” axes. In the pattern of C, D, and F all further tiles (not contained in the L-shape) are connected directly or by a neighbouring tile with this “$y$” axes. But in the generator E these four tiles are just connected with the “$x$” axes.

In the following we explain the $y$ drift for generator C with iteration level $l = 1$, where the same explanation holds true for the other two generators. The particles start in the left corner at the bottom of the iterator. Now caused by the bias, most of the particle walk along the bias. However, half of the particles can move in the two $y$-directions. The particles walking downwards, reach after two steps the four corner tiles. Caused by the bias, they move into these corner and prefer to leave from there the iterator along the bias through the right corner at the bottom than opposed to it. On the other hand, the particles moving upwards at the beginning hop also in the four corner tiles of the corresponding iterator, due to the bias. From there we have the same situation as before. Most of the particles prefer to leave the iterator along the bias. All together the particles leave (most of them) the iterator through the right corner at the bottom. So we can expect a drift in positive $x$- and negative $y$-direction.

For generator E only a long time drift in $x$-direction along to the bias appears, shown in fig. 4.3. That drift is also caused by the bias. The reason for no drift in $y$-direction seems to be that the pattern does not contain such corner
Results

like structures that leads the particles in certain \( y \)-direction. Just due to the
dead ends where particles get trapped we can see this little drift motion in
\( y \)-direction.

Up to here we proposed an explanation, why drift can also occur in \( y \)-
direction. The question remains open why we do see the observed respons of
the drift velocity to the bias amplitude, shown in fig. 4.5.

This effect indicates again influences of dead ends on the drift behaviour of
diffusing particles. During the diffusive process particles can move into dead
ends. Due to the external field particles walked into dead ends lying along the
bias. If they want to escape they have to hop opposed to the bias direction.
However, for this the transition probability decreases with increasing bias
amplitude. So it takes a long time to escape for the particles with a strong
external field applied. Thereby the overall motion slows down, as many
particles are trapped. For weak bias amplitudes the particle can leave dead
ends relatively easy. So more particles can walk along the external field.
That leads to higher mean drift velocities \( \langle \tilde{v}_{dr} \rangle \). The same explanation can
also be proposed for other random structures, so-called percolation clusters
\[34, 08\].
5. CONCLUSION

The implementation of the master equation presented in chapter 3 is the starting point for our numerical simulations. They are done in order to compute diffusion phenomena in different external fields in disordered materials. In this master thesis I wanted to get a first insight into anomalous diffusion under the influence of an external field. Therefore I analysed diffusion in time constant fields on regular SIERPINSKI carpets. For testing the extended implemented programme I applied three different lattice structures presented in section 3.4. At first I investigate the well know EUCLIDEAN lattice within and without a biased direction. I find that the results of the simulations are reasonable and correct. The same holds true for the calculation of the random walk dimensions $d_w$ for two different regular SIERPINSKI carpets.

In order to analyse diffusion on structured lattices influenced by a static external field we applied four different fractal pattern and six bias amplitudes, explained in section 4.1. Surprisingly the achieved random walk dimension $d_w$ exhibits that diffusion within an external field is a superdiffusive process. But the diffusive process is hindered by the fractal structure that contains dead ends. With the probability distribution obtained by the master equation approach we calculated the marginal distribution $\tilde{p}(x,t)$. This helped to understand the diffusive behaviour. We find that competing effects resulting in a faster spreading of the probability distribution are the reason for the obtained $d_w$. On the one hand I have fast moving particles, accelerated through the external field. On the other hand there are particles running into dead ends. To escape form these traps, the particles have to move opposite to the field. Due to these two effects the mean square displacement gets wider and the diffusion temporarily seems to be superdiffusive.

Furthermore I find that $\nu_{dr}$ does not increase proportional with increasing bias amplitude. The maximal drift velocity apprears in weak bias. The reason for this phenomena seems to be again dead ends. As mentioned particles walked into dead end have a delay there. With stronger amplitudes the
probability to leave gets smaller. So the particles are trapped. Caused by that we see that the motion slows down with higher bias amplitudes.

This thesis is just a first step in order to investigate anomalous diffusion in the presence of an external field. The presented implementation of the master equation in chapter 3 can be applied directly for further investigations. So in the future we want to apply time dependent external fields but also randomised fractals as an appropriate model for real materials.

In summary we see that the master equation approach provides a great tool in order to analyse anomalous diffusion on fractals, as we obtain the whole probability distribution for further investigations.


SELBSTSTÄNDIGKEITSERKLÄRUNG

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.


JANETT BALG