Rosenbluth Algorithm Studies of Self-avoiding Walks

Mandana Tabrizi

*A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE*

***GRADUATE PROGRAM IN MATHEMATICS AND STATISTICS
YORK UNIVERSITY
TORONTO, ONTARIO
August 2015***

Abstract

In this thesis we used self-avoiding walks as a model of linear polymers to study some of the most fundamental questions about polymers- namely the quantification of polymer entropy. We introduced scaling formulas for the number of walks and other polymer properties such as radius of gyration and end-to-end distance. Then, we calculated these quantities using a Monte Carlo simulation and estimated the critical exponents in the scaling formulas.

There is a pressure field in the vicinity of a polymer and a particle placed close to the polymer will accelerate away from it due to the pressure gradient. The scaling of the pressure as a function of distance from the polymer and length of the polymer is determined and tested numerically. Also, we modeled the relationship between velocity and the position of the particle in the 2-dimensional lattice and estimated the limiting speed.

Acknowledgment

I extend my sincere gratitude and appreciation to many people who made this thesis possible.

First, I would like to thank my supervisor, Dr. Janse van Rensburg for his support and guidance through this thesis-work. My thanks to my parents and especially my sister who gave me their unconditional support throughout my study years. Last but not least, I would like to thank the staff in the department, (especially Primrose) for being always helpful.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Random walks have been used to model different phenomena such as polymers, DNA structure and stock market returns. A random walk model of a polymer does not account for self-avoidance, but the introduction of self-avoiding walks (see, for example, reference [22]) improved the model. Self-avoiding walks (abbreviated as SAW) have been studied since the 1940s as the most basic model of a linear polymer [10, 22]. The field has advanced in major ways over the last 50 years, but the most basic question - the determination of the number of walks of given length, denoted by c_n - remains unresolved.

Another advantage of using self-avoiding model for polymers is that we can use it to formulate some polymer properties such as end-to-end distance and square radius of gyration. Also, the polymer entropy is related to the number of states, which in this model can be substituted by c_n.

There are different approaches to the numerical study of self-avoiding walks. Exact enumeration using series analysis techniques gives high quality estimates of critical exponents [44, 45]. Conformal field theory and Coulomb Gas methods have also been used to determine exact exponents in two dimensions [41, 49, 50]. Monte Carlo simulations which use numerical and statistical techniques are useful in the testing of scaling relations for walks for approximation of the number of walks (see for example [36]).

In this thesis, we choose Monte Carlo method to generate samples of walks. The Monte Carlo algorithm used to generate walks is the Rosenbluth algorithm in two and three dimensions. This algorithm works by recursively choosing the next vertex from the set of nearest neighbors that are not occupied by the walk. Using the Rosenbluth algorithm we estimated the number of walks of length n ($\langle W_n \rangle$), mean square end-to-end distance ($\langle R_e^2 \rangle$) and mean square radius of gyration ($\langle R_g^2 \rangle$).

Since the Monte Carlo method for calculation of the number of walks, radius of gyration and other properties for longer walks are computationally difficult, there are scaling formulas suggested to predict these quantities for any given length. All these quantities scale as a power law with the length of the walk. In this thesis, we estimated the metric exponents in scaling relations by fitting our numerical results of the Monte Carlo simulation to the scaling formulas.

Determining the metric exponents is one of the most basic reasons for developing numerical algorithms for walks. In addition to Monte Carlo a variety of different methods have been used to estimate the numerical value of metric exponents in the square and cubic lattices. Exact enumeration and series analysis of walks have verified many of these exact values to excellent accuracy and recent advances have extended series for walks to remarkable lengths in the square lattice [44], [45], [46] and [47] and in the cubic lattice [48].

Another problem examined in this thesis is a model for a particle near a polymer. The presence of the particle causes a change in pressure and therefore an acceleration of the particle from higher-pressure regions close to the polymer to lower pressure. In chapter 3 we study the relation between velocity and the position of the point in 2-dimensional lattice and also determine the terminal velocity of the particle. Our numerical results show that the terminal velocity is independent of the size of the polymer.

In chapter 4 we examine a model of entropic pressure in the vicinity of a polymer. In particular, we developed a scaling relation for the pressure. The scaling relation is tested by Monte Carlo simulation. The sampling is done by implementing the Rosenbluth algorithm for lattice polymers in 2-dimensional space. The estimated pressure shows scaling consistent with the derived scaling relation. For example, the numerical data show that the rescaled pressure is independent of the length of the polymer. Moreover, the pressure is found to be isotropic, decaying at the same rate in any direction with distance from the origin.

Chapter 2

The self-avoiding walk

A polymer is a molecule consisting of many monomers which are connected together by chemical bonds. A linear polymer is a polymer for which each of the monomers are joined to only two other monomers. Polymers can be very long molecules, some linear polymers have as many as 10^5 monomers.

Assume a linear polymer has n monomers numbered $0, 1, 2..., n$, starting from one endpoint. Let $v_i \in \mathbb{R}^3$ be the location of i^{th} monomer. The i^{th} monomer bond can be represented by the line segment between $\vec{v}_i$ and $\vec{v}_{i+1}$.

Given the assumptions above, one can see that a possible model for a linear polymer is a *random walk*. In general a random walk is a collection of vertices and edges and therefore can be put on a grid. The basic structure that satisfy this assumption is the d-dimensional hypercubic lattice $\mathbb{L}^d$. For example, a walk of length n in the two dimensional lattice $\mathbb{Z}^2$ is obtained by first setting the starting point at origin and then at each step, the next vertex is chosen from the nearest neighbor lattice sites (in 2-dimensions the choices are up, down, left and right). Continue to grow the walk in this fashion until a walk of length of n is obtained. See figure 2.1 for an example of a walk of length 117. In the construction of random walks of length n, each new edge can be chosen from $2d$ possible directions. Thus the number of random walks of length n is $(2d)^n$.

2.1 Self-avoiding walks

Since two segments of one molecule cannot occupy the same place in the space, the self-avoiding walk (abbreviated as SAW) seems more appropriate than random walks for modeling linear polymers. A self-avoiding walk is a random walk that cannot cross itself. A self-avoiding walk is defined by a sequence of vertices $\{\vec{v_0}, \vec{v_1}, ..., \vec{v_n}\}$ in a lattice together with a set of edges of the form (v_{i-1}, v_i), where $v_i \neq v_j$ if $i \neq j$ and the vertices of each edge are nearest neighbors in the lattice.

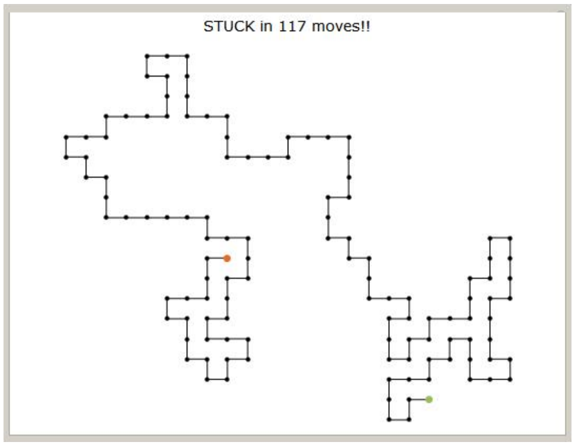

Figure 2.1: Picture of a self-avoiding walk that got trapped at length 117 [16]

In d-dimensions the Cartesian coordinates of a vertex $\vec{v_i}$ is denoted by a d-tuples $(X(\vec{v_i}), Y(\vec{v_i}), ..., Z(\vec{v_i}))$. We will consider walks from the origin in the d-dimensional hypercubic lattice $\mathbb{Z}^d$. In this thesis, we loosely refer to walks in d-dimensional lattice $\mathbb{L}^d$ as walks in d-dimensions.

It is convention to put the *zeroth vertex* $\vec{v_0}$ of the walk at the origin. The first and last vertices of a walk are its *end-vertices*. The walks are oriented; they have a direction from the first vertex at the origin to the last vertex at its endpoint. The length of a walk is the number of its edges.

The SAW model is a good model for a polymer in dilute solution (where polymers are far apart, so that there is not any considerable interactions between them) and in a good solvent (which minimizes attractive forces between monomers). For the rest of this thesis we assume that we have a dilute polymer solution in a good solvent.

The number of self-avoiding walks of length n starting at the origin is denoted by c_n. This number is known for $n \leq 71$ [46] in $\mathbb{Z}^2$ and for $n \leq 30$ in $\mathbb{Z}^3$ [48]. In a SAW of length n, there are $2d$ choices for the first step of the walk, and at most $2d - 1$ choices for the rest of the $n - 1$ steps, so

$$c_n \leq 2d(2d-1)^{n-1}. \tag{2.1}$$

The table below shows the values of c_n for different n in the square and cubic lattices $\mathbb{L}^2$ and $\mathbb{L}^3$.

n	c_n in Z^2	c_n inZ^3
1	4	6
2	12	30
3	36	150
4	100	726
5	284	3534
6	780	16926
7	2172	81390
8	5916	387966
9	16268	1853886
10	44100	8809878

Table 2.1: Table of number of SAWs, c_n, for $1 \leq n \leq 10$ in 2D [46] and 3D [48]

2.2 Polymer entropy

Self-avoiding walks can be used to quantify the entropy of a polymer. The physical definition is that entropy is (k=Boltzmann's constant)

$$S = k \log \left(density\ of\ states \right) \tag{2.2}$$

For a discrete model the density of states is just the number of states (or number of conformations). A polymer has many conformations, and these must be counted in order to compute the entropy. One way to do this is to use a model where there are a finite number of conformations, and then to count them. The self-avoiding walk is a good model for this – it is self-avoiding and linear like a polymer, and by counting its conformations, we have a function c_n which is the number of conformations.

For example, the entropy of all walks starting at origin and ending at lattice point R is related to the number of walks of length n and given by

$$S(R) = k \log(c_n^{end}(R)) \tag{2.3}$$

where $c_n^{end}(R)$ is the number of self-avoiding walks from the origin to R. We choose the units in a way that k=1 in the above.

2.2.1 Approximation of polymer free energy by random walks

Consider first a random walk approximation in one dimension for polymer entropy.

Let $c_n(x)$ be the number of the random walks of length n from 0 to x in1-dimension. Let n^+ represent the number of steps in the +x direction and n^- number of steps in the -x direction. Then $n^+ + n^- = n$ and $n^+ - n^- = x$, or $n^+ = \frac{n+x}{2}$. Therefore, $c_n(x)$ can be calculated using Stirlings formula.

$$c_n(x) = \binom{n}{\frac{n+x}{2}} \sim \sqrt{\frac{2}{\pi n}}\Big(1 - \frac{x^2}{n^2}\Big)^{-n/2-1/2} \left(\frac{1-\frac{x}{n}}{1+\frac{x}{n}}\right)^{x/2}$$

Assuming $< x^2 > \sim n$

$$c_n(x) \sim \sqrt{\frac{2}{\pi n}}\; e^{x^2/2n^2}\; e^{-x^2/2n}$$

Since $e^{x^2/2n^2} \to 1$ as $n \to \infty$, the probability that a random walk ends at the point x in 1-dimension is proportional to

$$n^{-\frac{1}{2}} e^{(\frac{-x^2}{2<x^2>})} \tag{2.4}$$

Assume $R = (x, y, z)$ in three dimensions,then the distributing function of R is

$$\begin{aligned} p(R) &\sim \frac{c_n(R)}{\sum_R c_n(R)} \\ &\sim constant\; n^{-\frac{1}{2}} e^{(\frac{-x^2}{2<x^2>})}\; n^{-\frac{1}{2}} e^{(\frac{-y^2}{2<y^2>})}\; n^{-\frac{1}{2}} e^{(\frac{-z^2}{2<z^2>})} \\ &\sim n^{-\frac{3}{2}}\; e^{(\frac{-3R^2}{2<R^2>})} \end{aligned} \tag{2.5}$$

Hence, the entropy of a chain of fixed length (n) ending at the point R is approximately given by

$$S(R) \sim \log p(R) \sim S(0) - \frac{3R^2}{2 < R^2 >} \tag{2.6}$$

This formula relates entropy to the size of a chain, as length increases, the entropy decreases. If we substitute 2.6 in the free energy formula $F(R) = U - TS$ and considering that internal energy of a walk is zero, we get to a predicted fundamental formula

$$F(R) \sim F(0) + \frac{3TR^2}{2 < R^2 >} \tag{2.7}$$

for the free energy of a polymer at temperature T.

2.3 Asymptotic properties of self-avoiding walks

Denote the numer of self-avoiding walks from the origin of length n by c_n. By equation 2.1, c_n grows exponentially fast with increasing length and soon it becomes difficult to determine it. Some values of c_n for small are listed in table 2.1.

In this section we study some properties of c_n. First a theorem:

Theorem 2.3.1 (Fekete's lemma)**.** *The basic subadditive theorem [9]: Suppose that $Z \subseteq \mathbb{N}$ is closed under addition and suppose the sequence a_n satisfies the subadditivity condition. That is,*

$$a_{n+m} \leq a_n + a_m \quad \forall n, m \in Z \tag{2.8}$$

Then

$$\lim_{n\to\infty} \frac{a_n}{n} = \inf_{1\leq n} \frac{a_n}{n} = \nu \tag{2.9}$$

Proof. Assume $n \in Z$ and $k \in Z$ is fixed. Set

$$A_k = \max\{a_l |\ l \leq k \text{ and } l \in Z\}. \tag{2.10}$$

Let $j = \left\lfloor \frac{n}{k} \right\rfloor$, then $n = jk + r$ where $0 \leq r \leq k$ (and so $a_r \leq A_k$ and $r \in Z$), since Z is closed under addition. Now by using the subadditivity property repeatedly we get

$$a_n = a_{jk+r} \leq a_{jk} + a_r \leq ja_k + A_k \tag{2.11}$$

Therefore,

$$\frac{a_n}{n} \leq \frac{ja_k}{jk+r} + \frac{A_k}{jk+r} \tag{2.12}$$

Take the limit superior on the left hand side as $n \to \infty$ in Z for fixed k. Then $j \to \infty$ in $\mathbb{N}$ and so

$$\limsup_{n\to\infty} \frac{a_n}{n} \leq \frac{a_k}{k} \quad \forall k \in \mathbb{N} \tag{2.13}$$

Taking the inf on the right hand side gives the result. Since a_n is finite-valued, then $\inf \frac{a_n}{n}$ is either finite or $-\infty$. □

2.3.1 The growth constant

Place a walk of length n in $\mathbb{L}^n$ with its first vertex at the origin. Place a second walk in $\mathbb{L}^n$ such that its first vertex is on the last vertex of the first walk. If the two walks do not intersect, then the result is a self-avoiding walk of length $n + m$. There are c_n choices for the first walk and c_m for the second, so

$$c_{n+m} \leq c_n c_m. \tag{2.14}$$

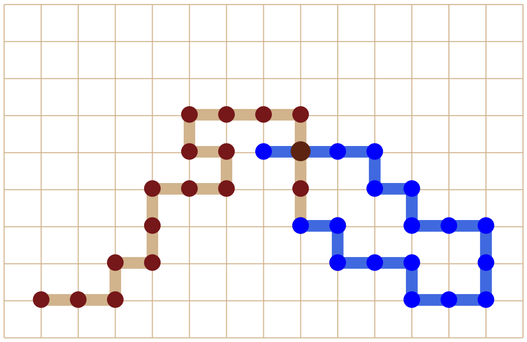

Figure 2.2: A walk of length n concatenated by a walk of length m crossing each other

Since each walk of length $n+m$ can be obtained this way, c_n is a submultiplicative function on $\mathbb{N}$ and therefore $\log c_n$ is a subadditive function on $\mathbb{N}$. Together with theorem 2.3.1 and equation 2.1, this implies that a connective constant $\kappa_d = \log \mu_d$ exists such that

$$\kappa_d = \lim_{n\to\infty} \frac{1}{n} \log c_n = \inf_{n>0} \frac{1}{n} \log c_n \tag{2.15}$$

where μ_d is called the growth constant. From the argument above we can conclude that the limit

$$\lim_{n\to\infty} [c_n]^{\frac{1}{n}} = \mu \tag{2.16}$$

exists but its not known that the $\lim_{n\to\infty}[\frac{c_{n+1}}{c_n}]$ exists. Also, Kesten's pattern theorem [18] shows that the limit

$$\lim_{n\to\infty} \frac{c_{n+2}}{c_n} = \mu^2 \tag{2.17}$$

exists.

2.3.2 Numerical estimates of μ

Determining the growth constant μ is one of the most basic reasons for developing numerical algorithms for walks. The numerical value of μ has been estimated in the square and cubic and other lattices using a variety of different methods. Exact enumeration and series analysis of walks as well as Monte Carlo simulations [40] have traditionally been used to estimate μ.

Series analysis for polygons [21] gives μ in two dimensions to very high precision:

$$\mu = 2.63815853034 \pm 0.00000000010. \tag{2.18}$$

Monte Carlo estimates for μ have been made using grand canonical Monte Carlo algorithms which sample self-avoiding walks from a distribution over their lengths. A well-known algorithm is the Beretti-Sokal algorithm [19]. This algorithm was used to estimate

$$\mu = 2.638164 \pm 0.000014, \tag{2.19}$$

The error bar is a combined 95% statistical confidence interval and an estimated systematic error due to uncertainties in the model.

Less precise estimates for μ are available in three dimensions. Clisby et al [48] estimated that

$$\mu = 4.684043 \pm 0.000012, \tag{2.20}$$

by collecting series data on the cubic lattice self-avoiding walk using the lace expansion.

2.4 Scaling of the self-avoiding walk

Scaling arguments can be used to show that c_n and other mean observables of self-avoiding walks satisfy scaling laws. The parameters in the scaling formulas are called critical exponents. The relation 2.15 shows that $c_n = \mu_d^{n+o(n)}$. This suggests a scaling assumption for the number of n step SAWs, namely

$$c_n = A\mu_d^n n^{\gamma-1} \tag{2.21}$$

where γ is the *entropic exponent* of the self-avoiding walk. By theorem 2.3.1 above, $c_n \geq \mu^n$, and so $\gamma \geq 1$. In fact, conformal field theory gives the exact value $\gamma = \frac{43}{32}$ in two dimensions [7]. Numerical estimates give the approximation $\gamma = 1.1573$ [31] in three dimensions.

The mean field value $\gamma = 1$ is the exact value in four and higher dimensions. Also, in four dimensions, which is the upper critical dimension for self-avoiding walks, we have a logarithmic modification, so that $c_n = A\mu^n(\log n)^{1/4}$.

Assume that $\upsilon = \{\upsilon_0, \upsilon_1, ..., \upsilon_n\}$ is a self-avoiding walk in the lattice $\mathbb{L}^d$ and let the Cartesian coordinates of a vertex υ_i in d-dimension be denoted by the d-tuples $(X(\upsilon_i), Y(\upsilon_i), ..., Z(\upsilon_i))$. Metric quantities of a self-avoiding walk are observables which have units (length).

The mean-square end-to-end distance of SAWs of length n is a metric quantity defined by

$$R_e = \parallel \upsilon_0 - \upsilon_n \parallel_2 \tag{2.22}$$

where $\| \upsilon \|$ is the Euclidean norm of the vector υ defined by $\| \upsilon \|_2^2 = [X(\upsilon)]^2 + [Y(\upsilon)]^2 + ... + [Z(\upsilon)]^2$. The mean-square radius of gyration is a metric quantity defined by

$$R_g^2 = \frac{1}{n} \sum_{k=1}^{n} (\| \upsilon_k - \upsilon_{mean} \|_2^2) \tag{2.23}$$

where υ_{mean} is the center of mass of the vertices. The span of υ in the $\vec{e_1}$ direction is given by

$$S_1(\upsilon) = \max_{0 \leq n} \left\{ \left| \upsilon_i(1) - \min_{0 \leq n} \upsilon_i(1) \right| \right\} \tag{2.24}$$

The span $S_k(\upsilon)$ in other directions $\vec{e_k}$ are similarly defined. The total span is

$$S_T(\upsilon) = \sum_{i=1}^{d} S_i(\upsilon) \tag{2.25}$$

and average span of υ is defined by

$$S_a(\upsilon) = \frac{1}{d} S_T(\upsilon) \tag{2.26}$$

Finally, the maximal span is

$$S_m(\upsilon) = \max_k S_k(\upsilon) \tag{2.27}$$

The means of observables are obtained by taking the average over all walks of length n and denoted by $\langle . \rangle_n$. That is, if υ is a walk of length n and O is a property of this walk, then we have

$$\langle O \rangle_n = \frac{1}{c_n} \sum_{|\upsilon|=n} O(\upsilon) \tag{2.28}$$

The mean square end-to-end distance $\langle R_e^2 \rangle_n$, mean square radius of gyration $\langle R_g^2 \rangle_n$ and mean total span $\langle S_T \rangle_n$ can be calculated using 2.28 . Since $n^{1/d} \leq \langle S_T \rangle_n \leq n$ and $n^{2/d} \leq \langle R_g^2 \rangle_n \leq n^2$ these properties scale as power laws with n:

$$\langle R_e \rangle_n = B_0 n^{\nu}, \tag{2.29}$$

$$\langle R_e^2 \rangle_n = B n^{2\nu}, \tag{2.30}$$

$$\langle R_g^2 \rangle_n = C n^{2\nu}, \tag{2.31}$$

$$\langle S_T \rangle_n = D n^{\nu}. \tag{2.32}$$

ν is the metric exponent. In the above cases it is easily seen that $\nu \leq 1$. The lower bound $\frac{1}{d} \leq \nu$ is obvious for the mean square radius of gyration and the spans. However, it's not known for the mean end-to-end distance that $\frac{1}{d} \leq \nu$. The best lower bound $\frac{2}{3d} \leq \nu$ is due to an unpublished work by Neal Madras. These scaling assumptions are valid in all dimensions $d \neq 4$. In $4 \leq d$, ν takes its mean field value $\nu = \frac{1}{2}$.

In four dimensions, the scaling formula is modified by a logarithmic term; for example, in $d = 4$ it is expected that,

$$\langle R_g^2 \rangle_n = Cn(\log n)^{1/4}, \tag{2.33}$$

$$c_n = A\mu^n (\log n)^{1/4}. \tag{2.34}$$

There are corrections to the scaling in c_n and in metric quantities; it is expected that

$$c_n = A\mu^n n^{\gamma-1}(1 + \frac{a_1}{n} + ... + \frac{b_1}{n^{\Delta_1}} + \frac{b_2}{n^{\Delta_1+1}} + ...), \tag{2.35}$$

$$R_e^2 = Bn^{2\nu}(1 + \frac{c_1}{n} + ... + \frac{d_1}{n^{\Delta_1}} + \frac{d_2}{n^{\Delta_1+1}} + ...), \tag{2.36}$$

$$R_g^2 = Cn^{2\nu}(1 + \frac{e_1}{n} + ... + \frac{f_1}{n^{\Delta_1}} + \frac{f_2}{n^{\Delta_1+1}} + ...). \tag{2.37}$$

The corrections of the form $\frac{a_i}{n^i}$ are called analytic corrections and $\frac{b_i}{n^{\Delta_1+i}}$ are called confluent corrections [7]. The confluent correction exponent Δ_1 above is the first in series of confluent correction exponents $\Delta_1 \leq \Delta_2 \leq ....$ Least square fitting of R_g^2 and c_n to data to determine the exponents ν and γ may in some cases require regressions which include analytic and confluent corrections to scaling.

We can define other metric quantities, such as the volume of the smallest box containing the walk, the area of the image of the walk projected to lower dimensions and so on. All these quantities have dimensions length raised to an integer power, and they scale with the length of the walk as a power law with exponent ν. Another metric quantity of interest is the mean square distance between a vertex in a self-avoiding walk of length n, and the endpoints of the walk, denoted by $\langle R_m^2 \rangle_n$. That is, if υ is a walk from the origin and υ_i is a vertex in υ, then $\langle R_m^2 \rangle_n$ is the mean square distance of υ_i from origin, averaged over all choices of $\{\upsilon_i\}_{i=1}^n$.

Several dimensionless ratios can be formed using the metric quantities and each approaches a constant as $n \to \infty$

$$\lim_{n\to\infty} \frac{\langle R_g^2 \rangle_n}{\langle R_e^2 \rangle_n} = \frac{R_g^2}{R_e^2} \tag{2.38}$$

$$\lim_{n\to\infty} \frac{\langle R_m^2 \rangle_n}{\langle R_e^2 \rangle_n} = \frac{R_m^2}{R_e^2} \tag{2.39}$$

The limiting ratios are universal quantities (independent of the lattice). Conformal field theory predicts that (Cardy and Saleur [23]) these limiting ratios relate with each other in two dimensions as follows:

$$\frac{123}{91}\frac{R_g^2}{R_e^2} - \frac{R_m^2}{R_e^2} + \frac{1}{4} = 0. \tag{2.40}$$

In two dimensions we have [36],

$$\frac{R_g^2}{R_e^2} = 0.140264, \tag{2.41}$$

while for d=3 [36],

$$\frac{R_g^2}{R_e^2} = 0.1599. \tag{2.42}$$

2.5 Numerical testing of scaling

The calculation of critical exponents such as the entropic exponent (γ) and the growth constant (μ), and testing of scaling relations, are important motivations for inventing of Monte Carlo algorithms for numerical simulation of walks. These algorithms generate the walks for the purpose of enumeration and calculating properties such as radius of gyration and end-to-end distance. Fitting the simulated data to the scaling formulas mentioned in section 2.4 (such as 2.21, 2.30 and 2.31) may give estimates for the critical exponents and growth constant.

The scaling formulas for c_n (equation 2.21) and $\langle R_g^2 \rangle$ (equation 2.31) are modified by adding analytic and confluent terms in equations 2.35 and 2.37. These corrections should be considered when data analysis is done in order to extract the values of exponents, and all the scaling laws in the simulations are subject to such corrections.

In addition to Monte Carlo simulation, there are other methods available for extracting the critical exponents in the literature. The generation of exact series for lattice walks has produced remarkably accurate results for critical exponents and connective constants [42] and [43]. This approach has proven to be a superior method for obtaining numerical data on walks, and recent advances have extended series for walks to remarkable lengths in the square lattice [44], [45], [46] and [47] and in the cubic lattice [48] with the result that Monte Carlo simulations have lagged considerably in accuracy when used to verify series results.

The results of series analysis should be considered against the backdrop that exact (but non-rigorous) numerical values for certain scaling exponents of the self-avoiding walk have been obtained in the square lattice by Coulomb gas [49] and conformal field theory [50] methods. The series data have verified many of these exact values to remarkable accuracy and perhaps even to a degree that cannot be obtained by Monte Carlo simulations as a matter of principle. This may be so even in the cubic lattice, where series analysis (using a lace expansion technique) has been used to provide good to excellent estimates for three-dimensional self-avoiding walk exponents and the cubic lattice connective constant [48].

	$d = 2$	$d = 3$	Mean field
γ	43/32 [7]	1.1608 [12]	1
ν	3/4 [11]	0.5877 [31]	1/2
μ	2.638 [21]	4.684 [48]	–
Δ_1	3/2 [49]	0.47 ± 0.025 [38]	–

Table 2.2: Table of self-avoiding walks exponents in 2D and 3D

In table 2.2, some exact values of critical exponents calculated by conformal field theory and Coulomb gas techniques are given in two dimensions. The estimates in three dimensions were computed by Monte Carlo simulations for γ and ν.

In the square lattice the entropic exponent γ was estimated using exact enumeration of walks [33]

$$\gamma = 1.343745 \pm 0.000015 \tag{2.43}$$

Monte Carlo simulations are not as accurate, [32]

$$\gamma = 1.345 \pm 0.004. \tag{2.44}$$

In three dimensions, the entropic exponent was estimated using field theory and is given in table 2.2. Monte Carlo method gave an accurate estimate

$$\gamma = 1.1573 \pm 0.0002 \tag{2.45}$$

using the PERM algorithm on the Domb-Joyce model [31].

The metric exponent ν has also been estimated in two dimensions by Monte Carlo simulation [36]:

$$\nu = 0.74963 \pm 0.00008 \tag{2.46}$$

In three dimensions many more Monte Carlo studies have been done to measure ν. For example, Hsu [31] used the PERM algorithm on the DombJoyce model to obtain the high accuracy estimate

$$\nu = 0.58765 \pm 0.00020 \tag{2.47}$$

Series enumeration in three dimensions [44] gave the approximation

$$\nu = 0.592 \pm 0.003. \tag{2.48}$$

The confluent correction Δ_1 in scaling formula 2.35 was predicted to have values either $\Delta_1 = 3/2$ by Coulomb gas methods [49] or $\Delta_1 = 11/16$ by conformal invariance methods [37]. Exact enumeration studies as well as Monte Carlo simulations strongly supports that $\Delta_1 = 3/2$ [46]. In three dimensions only effective values of Δ_1 have been used in Monte Carlo simulations to estimate other exponents. Numerical estimates for Δ_1 can be found in [36]: $\Delta_1 = 0.56 \pm 0.03$,

2.6 The Flory argument for metric scaling

Flory introduced a scaling argument [22] and [10] for estimating the metric exponent ν. Suppose a polymer or self-avoiding walk of length N monomers occupies a spherical volume of space of mean diameter R. Since the molar concentration of mixture, c is defined as the amount of each constituent n divided by the volume, the monomer concentration in d dimensions is

$$c \sim \frac{N}{R^d} \tag{2.49}$$

Assume the volume of a single monomer is denoted by $v(T)$ i.e it's a function of temperature. The total number of pairs of interacting monomers has concentration $\frac{1}{2}c^2$. So the total energy due to monomer-monomer interaction is

$$f_{rep} = \tfrac{1}{2}\, T\, v(T)\, c^2. \tag{2.50}$$

Replacing c and then integrating over a volume R^d gives an approximation of the total energy of the polymer

$$F_{rep|tot} = \tfrac{1}{2}\, T\, v(T)\, \frac{N^2}{R^{2d}} \times R^d = \tfrac{1}{2}\, T\, v(T)\, \frac{N^2}{R^d}. \tag{2.51}$$

For large distortion we have small entropy, so Flory added an entropic elastic energy

$$F_{el} = BT\frac{R^2}{N} \tag{2.52}$$

which is obtained as follows: Assuming that the end-to-end distribution of the walk has a Gaussian distribution given by equation 2.4, the entropy of the walk is approximated by equation 2.6:

$$S(R) = A + B\frac{R^2}{N} \tag{2.53}$$

for some constants A and B. The second term is an elastic energy giving F_{el} above. Adding equations 2.51 and 2.52 we get the free energy of the walk

$$F = F_{el} + F_{rep|tot} = BT\frac{R^2}{N} + \tfrac{1}{2}\, T\, v(T)\, \frac{N^2}{R^d} \tag{2.54}$$

The statistics of the walk is dominated by conformations of minimum free energy, so minimizing the above gives an expected value R_F for the radius of the volume occupied by the walk by

$$R_F \simeq C_0 N^{3/(d+2)}. \tag{2.55}$$

Comparison to equation 2.30 gives

$$\nu = \frac{3}{d+2} \tag{2.56}$$

So for $d = 1$ we get the exact value. The estimation for $d = 2$ is exact [11] and d = 3 is almost accurate compared to recent high accuracy estimates (see for example [31]).

Chapter 3

Monte Carlo methods for self-avoiding walks

There are two general approaches to enumerate self-avoiding random walks. One is exact enumeration using series analysis. The other is Monte Carlo simulations which use numerical and statistical techniques for approximate enumeration of walks.

Monte Carlo sampling of walks generates a sample of states from a distribution over state space to calculate the expected values of observables. In the case of walks, the state space S can be defined in several ways, depending on the type and the implementation of the Monte Carlo algorithm. In this thesis, S is the state space of all different conformations of walks rooted at the origin.

The distribution over S gives the probability that a given state is sampled from S. This distribution may not be uniform. For example, if the state space is infinite, then the probability of a given walk will be dependent on variables such as length. One way of categorizing Monte Carlo algorithms for walks is based on the distribution of ensemble. If the distribution is uniform for walks of fixed size then the algorithm is called a *canonical Monte Carlo algorithm.* If the sampling algorithm creates walks of arbitrary length (size) from a distribution that depends on the size, then the algorithm is a *grand canonical Monte Carlo algorithm.*

Finally, Monte Carlo methods are divided into dynamic and static methods; depending on whether the sampling is done directly from the state space S along a Markov chain in S. There are several algorithms in the literature for each of these two classes.

3.1 Static vs dynamic Monte Carlo algorithms

In a static Monte Carlo algorithm states are sampled independently from a distribution over S. We can find the weighted average of the samples and take it as an approximation of an observable over S. Static Monte Carlo algorithms including Rosenbluth, GARM and GAS algorithms are called approximate enumeration algorithms (see [27] and [28]).

Dynamic Monte Carlo algorithms involve the implementation of dynamic rules for sampling along a Markov chain. In each step, the next state is obtained by applying an elementary move to the current state. The next state is then accepted by using a probabilistic rule. The Metropolis algorithm as well as its implementation using Umbrella sampling or multiple Markov Chain sampling are examples of dynamic Monte Carlo algorithms [26].

3.2 Rosenbluth sampling

The Rosenbluth algorithm is a static algorithm which is implemented as follows:

Set the origin as the starting vertex, and recursively choose the next vertex from the set of nearest neighbors that are not occupied by the walk. If there are no nearest neighbors available, then the walk is rejected and a new walk is initiated at the origin.

3.2.1 Rosenbluth algorithm

The steps in the algorithm are

1. Set $W = 1$ and let v_0 be the origin. Determine v_1 by choosing one of the nearest neighbors of v_0.

2. Recursively determine v_i by choosing one of the unoccupied neighbor vertices of v_{i-1}. Assume the number of these vertices is σ_{i-1}, then the probability of choosing v_i is $1/\sigma_{i-1}$.

3. If $\sigma_{i-1} = 0$ then the walk is rejected. Start a new walk from step (1).

4. Update the weight $W \rightarrow W\sigma_{i-1}$.

5. If $i < n$, then add a new vertex by going to step (2). If $i = n$ then a walk of length n and weight W has been generated. Start a new walk at step (1) until N walks are generated.

The Rosenbluth weight of a walk s of length n is

$$W(s) = \prod_{i=1}^{n} \sigma_i(s) \tag{3.1}$$

The probability $P(s)$ that a walk s is generated is $1/W(s)$. Note that walks with $\sigma_i = 0$ for some i are also included in sampled walks, but with zero weights.

Consider the average weight for walks of length n

$$\langle W_n \rangle = \sum_{|s|=n} P(s)W(s) \tag{3.2}$$

Since the sum is over all walks of length $|s| = n$ and $P(s) = 1/W(s)$, it follows that $\langle W_n \rangle = \sum_{|s|=n} 1 = c_n$, where c_n is the number of walks of length n from origin.

Consider a sample of N self-avoiding walks of length n, say $(s_1, s_2, ..., s_N)$ with associated weights $(W(s_i))$ and observables (O_i) obtained by Rosenbluth sampling. Then $\langle W_n \rangle$ is estimated by

$$\langle W_n \rangle_N = \frac{1}{N} \sum_{i=1}^{N} W(s_i) \tag{3.3}$$

As $N \to \infty$, $\langle W_n \rangle_N \to c_n$ where c_n is the number of walks of length n. Therefore, the Rosenbluth algorithm is a method for approximate enumeration of self-avoiding walks.

3.2.2 Data analysis of numerical results

Suppose we divide a sample of N walks obtained by Rosenbluth sampling into M blocks. For each block, we compute the mean values of observables using weighted averages:

$$\langle O_i \rangle_n^{block} = \frac{\sum_{i=1}^{\frac{N}{M}} O(s_i)W(s_i)}{\sum_{i=1}^{\frac{N}{M}} W(s_i)} \tag{3.4}$$

then we estimate the ordinary average of the blocks by:

$$\langle O \rangle_n = \frac{1}{M} \sum_{i=1}^{M} \langle O_i \rangle_n^{block} \tag{3.5}$$

and the standard error of the sample is calculated by:

$$std.err. = \sqrt{\frac{\langle O_i^2 \rangle_n - \langle O_i \rangle_n^2}{M-1}} \tag{3.6}$$

We generate SAWs of maximum length 70 using the Rosenbluth algorithm with 10^8 samples which is divided into 20 blocks. For each walk of length n in the sample we associate a weight using formula 3.1 and calculate $R_e(s)$ and $R_g(s)$ from formulas 2.22 and 2.23.

The number of walks of length n, mean square end-to-end distance and mean square radius of gyration; denoted by $\langle W_n \rangle$, $\langle R_e \rangle$ and $\langle R_g^2 \rangle$; are obtained by taking the block averages of $W(s)$, $R_e(s)$ and $R_g^2(s)$ over all walks in that block and then taking the ordinary average of block averages.

The table below shows the estimates for $\langle W_n \rangle$,$\langle R_e \rangle$ and $\langle R_g^2 \rangle$ and their corresponding standard error. Tables 3.2 and 3.4 show these approximations for 3-dimension and 2-dimension respectively.

n	$\langle W_n \rangle$	std.err.	$\langle R_e \rangle$	std.err.	$\langle R_g^2 \rangle$	std.err.
1	6	0	1	0	0.25	0
2	30	0	1.5313	$2e-005$	0.488861	$6e-006$
3	150	0	1.90743	$5e-005$	0.72995	$1e-005$
4	725.9995	0.002037636	2.27586	$7e-005$	0.98977	$2e-005$
5	3534	0.012576563	2.5777	$7e-005$	1.25519	$3e-005$
6	16926.06111	0.093633582	2.8849	0.0001	1.53771	$6e-005$
7	81390.38889	0.439911525	3.1497	0.0001	1.82453	$7e-005$
8	387966.1667	2.105470621	3.418	0.0001	2.12458	$7e-005$
9	1853882.222	11.47761549	3.657	$9e-005$	2.42802	$8e-005$
10	8809842.778	64.24425182	3.8997	0.0001	2.74255	$9e-005$
11	41933666.67	365.3273221	4.1205	0.0002	3.0599	0.0001
12	198840111.1	2561.516209	4.3444	0.0002	3.3868	0.0001
13	943956166.7	11942.4683	4.5511	0.0002	3.7161	0.0001
14	4468815000	63648.59614	4.7603	0.0002	4.0537	0.0002
15	21174705556	308747.4402	4.9555	0.0002	4.3934	0.0002
16	$1.0012E+11$	1260612.179	5.1528	0.0002	4.7405	0.0002
17	$4.73717E+11$	5701816.526	5.3387	0.0002	5.0894	0.0002
18	$2.23765E+12$	29823068.18	5.5264	0.0002	5.4453	0.0002
19	$1.05757E+13$	150989233.8	5.7036	0.0002	5.8025	0.0002
20	$4.99156E+13$	781035428.1	5.8824	0.0002	6.166	0.0002
21	$2.35701E+14$	3852584981	6.0529	0.0002	6.5308	0.0002
22	$1.11174E+15$	18580723412	6.2246	0.0002	6.9013	0.0002
23	$5.24579E+15$	93109753995	6.3887	0.0002	7.2729	0.0002
24	$2.47292E+16$	$4.76202E+11$	6.5545	0.0002	7.6501	0.0002
25	$1.16614E+17$	$2.61344E+12$	6.7134	0.0003	8.0284	0.0003
26	$5.49473E+17$	$1.34595E+13$	6.8736	0.0003	8.4118	0.0003
27	$2.58977E+18$	$6.91426E+13$	7.0277	0.0003	8.7962	0.0004
28	$1.21977E+19$	$3.4869E+14$	7.183	0.0004	9.1853	0.0004
29	$5.74653E+19$	$1.77326E+15$	7.3327	0.0004	9.5753	0.0005
30	$2.70564E+20$	$8.29057E+15$	7.4833	0.0004	9.9698	0.0006
31	$1.27417E+21$	$4.10344E+16$	7.629	0.0004	10.3651	0.0006
32	$5.99731E+21$	$2.03168E+17$	7.7754	0.0004	10.7643	0.0006
33	$2.82336E+22$	$9.15679E+17$	7.9173	0.0004	11.1643	0.0007
34	$1.32854E+23$	$4.31202E+18$	8.0602	0.0004	11.5687	0.0006
35	$6.25243E+23$	$2.06892E+19$	8.1987	0.0004	11.9736	0.0007

Table 3.1: Table of estimates of c_n, R_e and R_g^2 and corresponding standard error for $1 \leq length \leq 70$ in 3D

n	$\langle W_n \rangle$	std.err.	$\langle R_e \rangle$	std.err.	$\langle R_g^2 \rangle$	std.err.
36	$2.94134E+24$	$9.65025E+19$	8.3383	0.0004	12.3826	0.0007
37	$1.38391E+25$	$4.34764E+20$	8.4739	0.0005	12.7924	0.0007
38	$6.50892E+25$	$2.18673E+21$	8.6099	0.0004	13.2053	0.0008
39	$3.06176E+26$	$1.02842E+22$	8.7423	0.0004	13.6191	0.0008
40	$1.43976E+27$	$4.83258E+22$	8.8752	0.0004	14.0361	0.0008
41	$6.77109E+27$	$2.2942E+23$	9.0049	0.0004	14.4539	0.0009
42	$3.18341E+28$	$1.13434E+24$	9.1351	0.0004	14.8749	0.0009
43	$1.49685E+29$	$5.3851E+24$	9.2623	0.0004	15.2967	0.0009
44	$7.03628E+29$	$2.48608E+25$	9.3899	0.0005	15.721	0.001
45	$3.30791E+30$	$1.15714E+26$	9.5147	0.0005	16.147	0.001
46	$1.55473E+31$	$5.61627E+26$	9.6403	0.0006	16.575	0.001
47	$7.30794E+31$	$2.71255E+27$	9.7629	0.0007	17.004	0.001
48	$3.43424E+32$	$1.43816E+28$	9.8861	0.0007	17.435	0.001
49	$1.61401E+33$	$7.05082E+28$	10.0066	0.0008	17.867	0.001
50	$7.58382E+33$	$3.38112E+29$	10.1278	0.0008	18.302	0.002
51	$3.56375E+34$	$1.65545E+30$	10.2462	0.0008	18.737	0.002
52	$1.67431E+35$	$7.18241E+30$	10.3655	0.0009	19.176	0.002
53	$7.8668E+35$	$3.59204E+31$	10.4823	0.0009	19.615	0.002
54	$3.69556E+36$	$1.69658E+32$	10.6	0.001	20.056	0.002
55	$1.73618E+37$	$8.1369E+32$	10.714	0.001	20.499	0.002
56	$8.15512E+37$	$3.72405E+33$	10.83	0.001	20.944	0.002
57	$3.83091E+38$	$1.71852E+34$	10.943	0.001	21.389	0.002
58	$1.79924E+39$	$7.8658E+34$	11.057	0.001	21.837	0.003
59	$8.45105E+39$	$3.86917E+35$	11.169	0.001	22.285	0.003
60	$3.96889E+40$	$1.76586E+36$	11.281	0.001	22.736	0.003
61	$1.86405E+41$	$8.61747E+36$	11.391	0.001	23.187	0.003
62	$8.75344E+41$	$4.22837E+37$	11.502	0.001	23.641	0.003
63	$4.11077E+42$	$2.05964E+38$	11.611	0.001	24.094	0.004
64	$1.93024E+43$	$9.70462E+38$	11.72	0.001	24.551	0.004
65	$9.06399E+43$	$4.72366E+39$	11.828	0.001	25.007	0.003
66	$4.25579E+44$	$2.16969E+40$	11.936	0.001	25.467	0.004
67	$1.99827E+45$	$1.05913E+41$	12.042	0.002	25.926	0.004
68	$9.3817E+45$	$5.1811E+41$	12.149	0.002	26.388	0.004
69	$4.40475E+46$	$2.56772E+42$	12.253	0.002	26.85	0.004
70	$2.06785E+47$	$1.23524E+43$	12.359	0.001	27.315	0.004

Table 3.2: Table of estimates of c_n, R_e and R_g^2 and corresponding standard error for $1 \leq length \leq 70$ in 3D

n	$\langle W_n \rangle$	std.err.	$\langle R_e \rangle$	std.err.	$\langle R_g^2 \rangle$	std.err.
1	4	0	1	0	0.25	0
2	12	0	1.60952	0	0.518533	0
3	36	0	2.04623	0	0.805545	0
4	99.99947778	0.000599758	2.55692	0	1.15516	0
5	283.9967222	0.002101673	2.95105	0	1.52067	0
6	779.9897222	0.007938821	3.39049	0	1.93699	0
7	2171.991111	0.032001725	3.74767	0	2.36425	0
8	5916.001667	0.104294236	4.14988	0.0001	2.83976	0
9	16267.95	0.34463678	4.48719	0.0001	3.32354	0.0001
10	44100.05556	1.122998502	4.86116	0.0001	3.85066	0.0001
11	120291.5	3.058524998	5.18447	0.0001	4.38581	0.0001
12	324939.2778	9.333615386	5.53734	0.0001	4.9602	0.0001
13	881522.8333	24.25103596	5.84925	0.0001	5.54226	0.0001
14	2374493.333	64.11284985	6.18574	0.0001	6.16049	0.0001
15	6416728.889	166.7350622	6.48786	0.0001	6.78575	0.0001
16	17246011.11	440.5054204	6.81131	0.0001	7.44504	0.0001
17	46468272.22	1164.165542	7.10508	0.0002	8.11082	0.0002
18	124662000	3131.121455	7.41745	0.0002	8.80872	0.0002
19	335132833.3	11934.25618	7.70375	0.0003	9.51244	0.0003
20	$8.977E+08$	32930.02473	8.00668	0.0003	10.2465	0.0004
21	2408947778	101016.2595	8.28651	0.0003	10.9861	0.0004
22	6444998889	292580.2375	8.58141	0.0003	11.7545	0.0004
23	17267666667	826718.8472	8.8554	0.0003	12.5278	0.0004
24	46149977778	2620651.463	9.14313	0.0003	13.329	0.0005
25	$1.23491E+11$	7848742.55	9.41201	0.0003	14.135	0.0005
26	$3.29739E+11$	21644667.9	9.69366	0.0003	14.9684	0.0006
27	$8.81382E+11$	54432753.51	9.95831	0.0003	15.8071	0.0006
28	$2.35155E+12$	155138153.8	10.2348	0.0003	16.6728	0.0007
29	$6.27979E+12$	379595588	10.4954	0.0002	17.5437	0.0006
30	$1.67435E+13$	1107524440	10.767	0.0002	18.4409	0.0007
31	$4.46793E+13$	3201811908	11.024	0.0002	19.3433	0.0008
32	$1.19048E+14$	8853985805	11.2913	0.0003	20.2716	0.0008
33	$3.17438E+14$	25834004489	11.545	0.0003	21.2048	0.0007
34	$8.45396E+14$	67366610179	11.8078	0.0002	22.1624	0.0007
35	$2.25286E+15$	$2.11541E+11$	12.0583	0.0002	23.1252	0.0007

Table 3.3: Table of estimates of c_n, R_e and R_g^2 and corresponding standard error for $1 \leq length \leq 70$ in 2D

n	$\langle W_n \rangle$	std.err.	$\langle R_e \rangle$	std.err.	$\langle R_g^2 \rangle$	std.err.
36	$5.99652E + 15$	$6.02516E + 11$	12.3176	0.0002	24.1122	0.0005
37	$1.59705E + 16$	$1.78787E + 12$	12.5649	0.0002	25.1037	0.0006
38	$4.24922E + 16$	$4.74971E + 12$	12.8206	0.0002	26.1191	0.0006
39	$1.13118E + 17$	$1.32742E + 13$	13.0654	0.0002	27.139	0.0008
40	$3.00844E + 17$	$3.81861E + 13$	13.3183	0.0001	28.1819	0.0011
41	$8.00508E + 17$	$1.11373E + 14$	13.5609	0.0001	29.2299	0.0011
42	$2.12822E + 18$	$3.19491E + 14$	13.8113	0.0003	30.3008	0.0012
43	$5.66065E + 18$	$8.75149E + 14$	14.052	0.0005	31.3765	0.0013
44	$1.50453E + 19$	$2.61542E + 15$	14.3003	0.0005	32.4748	0.0013
45	$4.00008E + 19$	$6.91281E + 15$	14.5399	0.0006	33.5784	0.0017
46	$1.06281E + 20$	$2.05048E + 16$	14.787	0.0005	34.7055	0.0017
47	$2.82499E + 20$	$5.55165E + 16$	15.0261	0.0005	35.8391	0.0018
48	$7.50399E + 20$	$1.44605E + 17$	15.2728	0.0006	36.996	0.0018
49	$1.99379E + 21$	$3.91206E + 17$	15.5115	0.0006	38.1584	0.0023
50	$5.29446E + 21$	$1.02995E + 18$	15.7572	0.0008	39.3448	0.0023
51	$1.40643E + 22$	$2.88629E + 18$	15.9946	0.0007	40.5352	0.0023
52	$3.73376E + 22$	$8.18574E + 18$	16.2394	0.0007	41.7506	0.0024
53	$9.91609E + 22$	$2.21437E + 19$	16.4764	0.0011	42.9731	0.0036
54	$2.63196E + 23$	$6.2062E + 19$	16.7192	0.001	44.2182	0.0031
55	$6.98757E + 23$	$1.62541E + 20$	16.9543	0.0009	45.4684	0.0028
56	$1.85421E + 24$	$4.26605E + 20$	17.1957	0.0009	46.7415	0.0037
57	$4.9221E + 24$	$1.11493E + 21$	17.4303	0.0012	48.022	0.0049
58	$1.30589E + 25$	$2.92148E + 21$	17.6698	0.0013	49.323	0.0053
59	$3.46544E + 25$	$7.41619E + 21$	17.9023	0.0013	50.6308	0.0053
60	$9.1919E + 25$	$1.93138E + 22$	18.1404	0.0011	51.9605	0.0052
61	$2.43878E + 26$	$4.8987E + 22$	18.3719	0.0013	53.2956	0.0049
62	$6.4673E + 26$	$1.20052E + 23$	18.609	0.0007	54.6548	0.0024
63	$1.71578E + 27$	$3.16853E + 23$	18.84	0.0011	56.019	0.0037
64	$4.54989E + 27$	$8.56803E + 23$	19.0765	0.0015	57.4058	0.0037
65	$1.20688E + 28$	$2.3231E + 24$	19.3057	0.001	58.7968	0.0046
66	$3.2004E + 28$	$6.03022E + 24$	19.5394	0.0014	60.2076	0.0061
67	$8.48887E + 28$	$1.59758E + 25$	19.7679	0.0019	61.6241	0.0063
68	$2.25035E + 29$	$4.60035E + 25$	20.0003	0.0021	63.0608	0.0065
69	$5.9676E + 29$	$1.38038E + 26$	20.2266	0.0026	64.5005	0.007
70	$1.58173E + 30$	$4.21616E + 26$	20.4572	0.0032	65.9633	0.0078

Table 3.4: Table of estimates of c_n, R_e and R_g^2 and corresponding standard error for $1 \leq length \leq 70$ in 2D

3.3 Estimating the metric exponents

Monte Carlo methods for enumeration and calculating properties such as radius of gyration and end-to-end distance are computationally intense. There are scaling formulas suggested to estimate these quantities for any given length (for example see equations 2.21, 2.30 and 2.31 in section 2.4.) In this section we fit our calculated data of the previous section to the scaling relations to estimate the critical exponents and growth constant.

Data obtained by approximate enumeration may be evaluated using series analysis techniques, provided that the data are accurate enough to dampen out random noise and sampling biases. As mentioned in (2.30), end-to-end distance is approximated by$R_e = Bn^{\nu}$. In approximate enumeration R_e is replaced by approximated values of the simulated 2-dimensional data from table (3.4). We can use the Flory value of ν in 2D and calculate $< R_e > /n^{\nu}$ to find an estimation for B in the equation (2.30). This is done in figure (3.1) (the right plot), where $< R_e > /n^{\nu}$ is computed from approximations to R_e and then plotted against $1/n$.

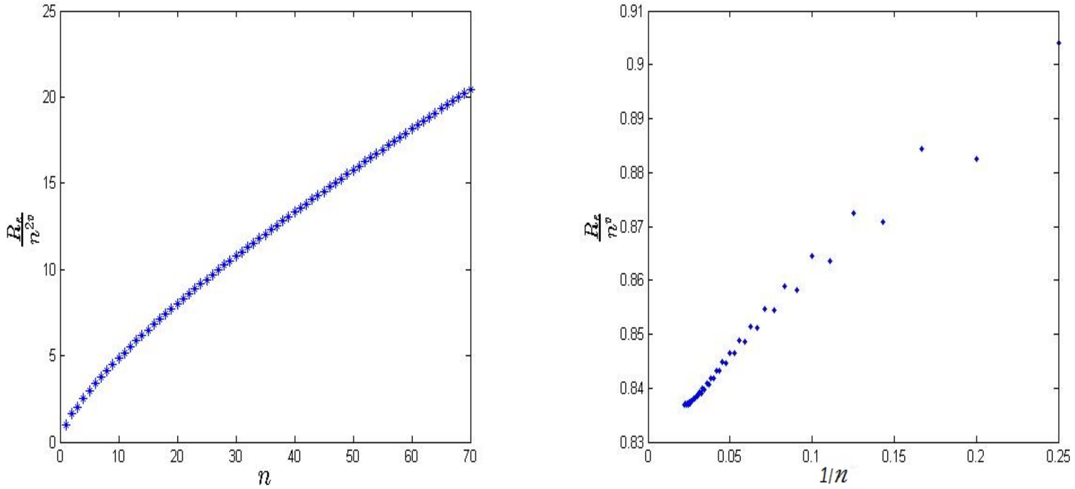

Figure 3.1: The plots for $< R_e > /n^{\nu}$ against n (left) and $1/n$ (right) in 2D

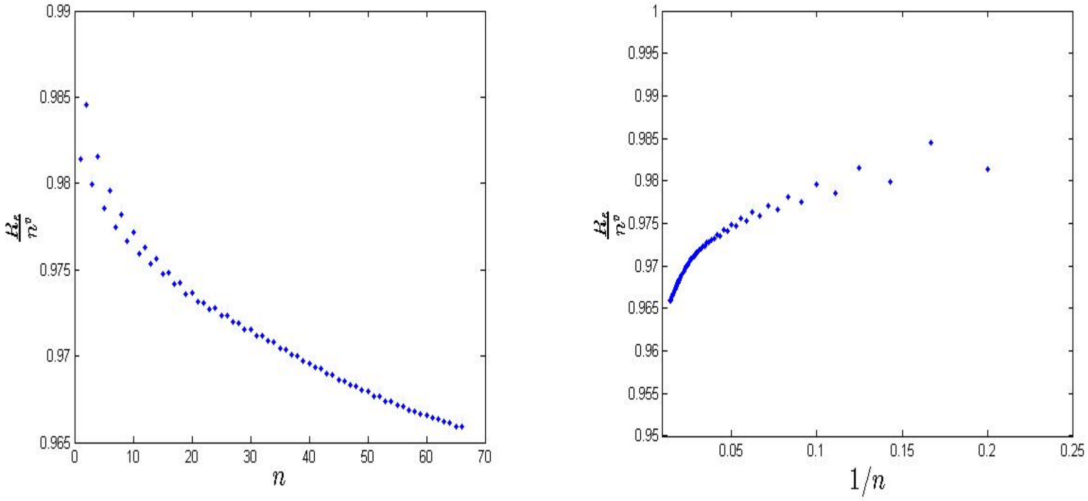

Figure 3.2: The plots for $< R_e > /n^\nu$ against n (left) and $1/n$ (right) in 3D

As it can be seen from the plot 3.1 as n increases $< R_e > /n^\nu$ converges to 0.84 which we take as an estimate for B in 2D. In a similar fashion, by using Flory value of ν in 3D and data from table (3.2) , we can plots $< R_e > /n^\nu$ vs n and $1/n$ (see figure 3.2). By extrapolating the ratio to the intercept with the Y-axis, we obtain an estimation $B = 0.97$ in 3D from figure (3.2).

We can also find an estimate for ν by using equation (2.30). This is done by plotting $\log(R_e)/\log(n)$ vs $1/n$, where we ignore the walks of small length (n starts from n_{min} =5 or 6.) As shown in figure (3.3), the sequence $\{(\log(R_e)/\log(n)\}_{n=n_{min}}^{70}$ approaches $\nu = 0.72$ in 2D and $\nu = 0.592$ in 3-dimension.

A more accurate way to estimate ν would be to use a least square method. Take logarithm of both sides of equation (2.30) to get:

$$\log(R_e) = \log B + \nu \log n \tag{3.7}$$

In order to take into account the error in the data, a weighted least square error is used. Minimize the error:

$$E_{Nmin}(B, \nu) = \sum_{i=Nmin}^{70} \frac{(\log(R_e)_i - (\log B + \nu \log i))^2}{(\sigma(R_e)_i/(R_e)_i)^2} \tag{3.8}$$

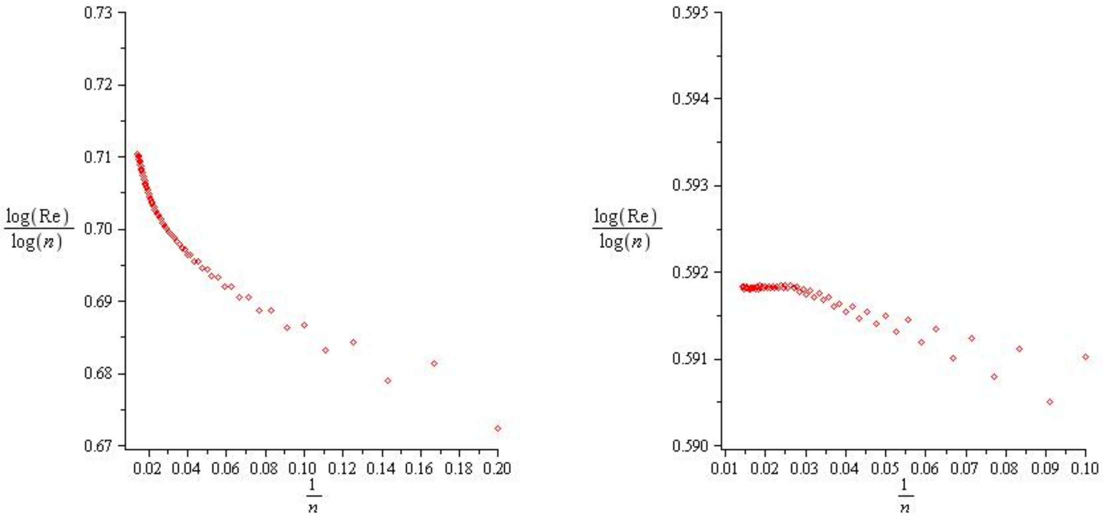

Figure 3.3: The plots for $log < R_e > /log(n)$ against $1/n$ in 2D (left) and 3D (right)

$Nmin$ is a cut-off $n_{min} \leq Nmin \leq n_{max}$ where $n_{min} = 4$ and $n_{max} = 35$ (The results for longer walks are convergent, so walks of length $35 \leq Nmin \leq 70$ are always counted in the calculations. On the other hand, walks of length $n \leq 4$ are less significant and discarded). The dataset for the walks of different length, i.e. $\{(R_e^2)_i\}_{i=Nmin}^{70}$ and $\{(\sigma(R_e^2)_i\}_{i=Nmin}^{70}$, is taken from Table (3.2) for 3D and Table (3.4) for 2D. If we take derivatives of 3.8 with respect to the parameters ν and B and equate them to zero and solve for B and ν, an estimate for ν is obtained. Tables 3.5 and 3.6 show the results for 3D and 2D respectively.

The estimates of ν is a function of $Nmin$, denoted by ν_n, $n = Nmin$. In figure 3.4 the values of $\{\nu_n\}_{n=n_{min}}^{n_{max}}$ is plotted against $1/n$, as it can be seen that this sequence approaches a limiting value. To obtain the best estimation of ν, we use a least square analysis, this time on the sequence $\{\nu_i\}_{i=n_{min}}^{n_{max}}$. Since the scatter plot suggests that there is a relationship between ν_n and $1/n$, we minimize:

$$\sum_{n=n_{min}}^{n_{max}} \left[\nu_n - (\nu_{best} + A/n + B/n^2)\right]^2 \tag{3.9}$$

to obtain estimates of ν_{best} for $n_{min} = 4$ and $n_{max} = 35$. The results are $\nu_{best} = 0.596$ in 3D and $\nu_{best} = 0.764$ in 2D. These results are acceptable if compared to the values given in the literature (for example see table 2.2.)

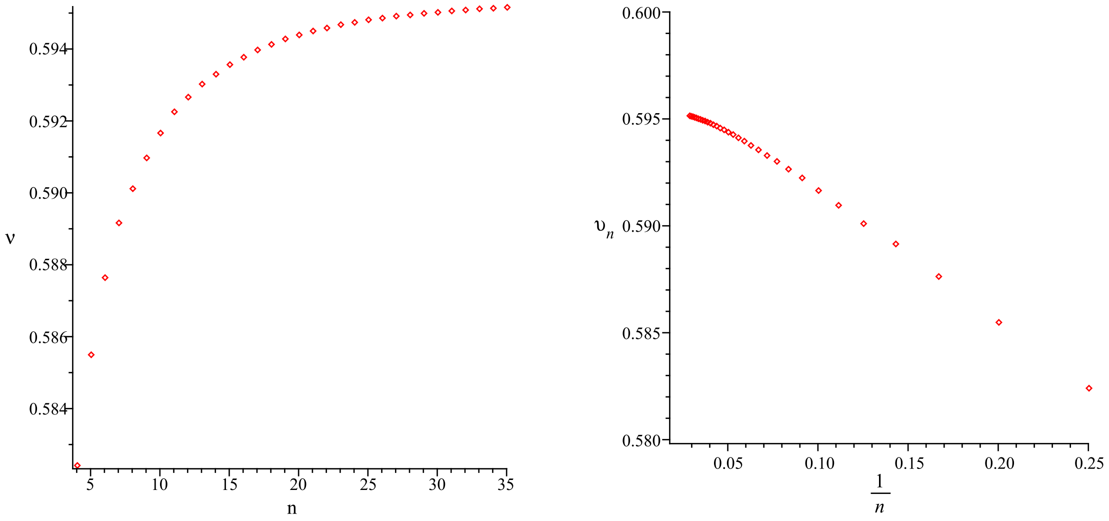

Figure 3.4: The plots for $\{\nu_n\}_{n=4}^{35}$ against n (left) and $1/n$ (right) in 3D

The growth constant can be similarly estimated. Use equation 2.21 to get

$$\frac{\log(c_n)}{n} = \frac{\log A}{n} + \log(\mu) + (\gamma - 1)\frac{\log(n)}{n} \tag{3.10}$$

Since $\log(n)/n \to 0$ as n increases and $logA/n$ is a small number,

$$\log(c_n)/n \sim \log(\mu) \tag{3.11}$$

Plot $\log(c_n)/n$ against $1/n$ where c_n is replaced by the estimates from Table (3.2). As it can be seen from Figure 3.5 the sequence $\log(c_n)/n$ is convergent to approximately 1.54. Therefore, we have the approximation $\mu = e^{1.54} = 4.66$ in 3D.

The exponents γ and μ can be simultaneously estimated using least square in a similar way as finding ν earlier. Take logarithm of both sides of scaling formula 2.21 to get:

$$\log(c_n) = \log A + n\log(\mu) + (\gamma - 1)\log(n) \tag{3.12}$$

Change variables $B = logA$, $C = log(\mu)$ and $D = \gamma - 1$. We obtain the summation

$$\sum_{n=Nmin}^{70} \frac{[\log(c_n) - (B + Cn + D\ \log(n))]^2}{(\sigma(c_n)/c_n)^2} \tag{3.13}$$

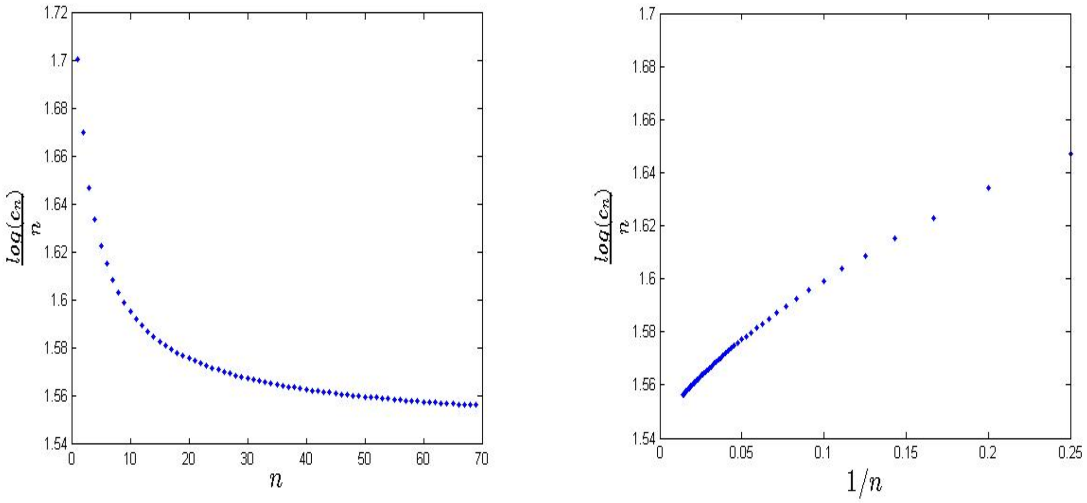

Figure 3.5: Plots for $log(c_n)/n$ against n (left) and $1/n$ (right) in 3D

$Nmin$ is between 4 and 35 and $\{(c_n)\}_{n=Nmin}^{70}$ and $\{(\sigma(c_n)\}_{n=Nmin}^{70}$ are taken from Table (3.2). We minimize 3.13 using Maple to estimate γ and μ. The results are shown in Tables 3.5 and 3.6 for 3D and 2D respectively.

As Nmin ranges from $n_{min} = 4$ to $n_{max} = 35$, different estimations of μ and γ are obtained. Figures 3.6 and 3.7 plot 3-dimensional values of $\{\mu_i\}_{i=n_{min}}^{n_{max}}$ and $\{\gamma_i\}_{i=n_{min}}^{n_{max}}$ respectively. Similar to 3.9 we use a least square method to find a best estimate $\mu_{best} = 4.683$ and $\gamma_{best} = 1.160$ in 3D. The corresponding values in 2D are $\mu_{best} = 2.650$ and $\gamma_{best} = 1.184$.

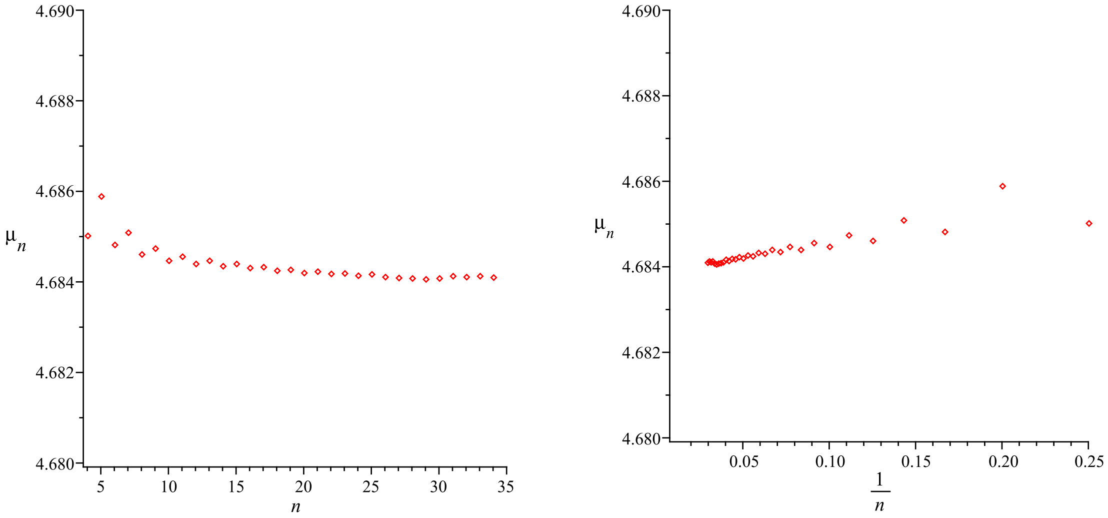

Figure 3.6: The plots for $\{\mu_n\}_{n=4}^{35}$ against n (left) and $1/n$ (right) in 3D

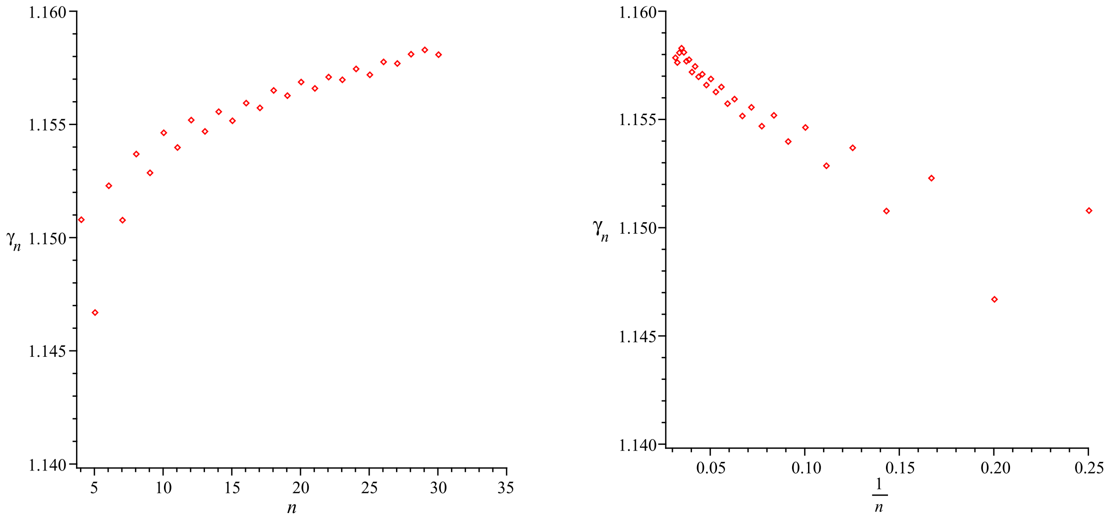

Figure 3.7: The plots for $\{\gamma_n\}_{n=4}^{35}$ against n (left) and $1/n$ (right) in 3D

n	ν	γ	μ
4	0.582436	1.15082	4.68503
5	0.585515	1.14672	4.6859
6	0.58766	1.15232	4.68483
7	0.589182	1.1508	4.6851
8	0.590134	1.15372	4.68462
9	0.59099	1.15289	4.68475
10	0.591681	1.15466	4.68448
11	0.592272	1.15401	4.68457
12	0.59268	1.15522	4.68441
13	0.593043	1.15472	4.68448
14	0.593318	1.15559	4.68436
15	0.593584	1.15519	4.68441
16	0.59379	1.15597	4.68432
17	0.593992	1.15576	4.68434
18	0.594147	1.15653	4.68426
19	0.594299	1.1563	4.68428
20	0.59441	1.1569	4.68421
21	0.594519	1.15662	4.68424
22	0.594602	1.15712	4.68419
23	0.594696	1.157	4.6842
24	0.594759	1.15748	4.68415
25	0.594831	1.15722	4.68418
26	0.594878	1.15779	4.68412
27	0.594933	1.15772	4.68413
28	0.594966	1.15813	4.68409
29	0.59501	1.15832	4.68407
30	0.595038	1.15811	4.68409
31	0.595077	1.15766	4.68414
32	0.595103	1.15788	4.68412
33	0.595136	1.15757	4.68414
34	0.595152	1.15787	4.68411
35	0.595177	1.15591	4.68429
35	0.595177	1.15591	4.68429

Table 3.5: Table of estimates of ν, γ and μ and corresponding standard errors in 3D

n	ν	γ	μ
4	0.7197126	1.2813708	2.6435642
5	0.7237568	1.2740622	2.64458
6	0.7244214	1.2896708	2.6426308
7	0.7276006	1.2824724	2.643455
8	0.728237	1.2913426	2.6425224
9	0.7303984	1.2841568	2.6432228
10	0.7309282	1.286987	2.6429614
11	0.7325878	1.2799462	2.6435772
12	0.7331892	1.2788122	2.643672
13	0.7345222	1.2701352	2.6443738
14	0.7350864	1.2656734	2.6447234
15	0.7364496	1.2600402	2.6451518
16	0.7371164	1.2548722	2.6455346
17	0.7385154	1.2515298	2.6457752
18	0.739299	1.2475132	2.646058
19	0.7405318	1.2460748	2.6461564
20	0.741333	1.2418578	2.64644
21	0.7425714	1.2362876	2.6468098
22	0.7434106	1.2364132	2.6468004
23	0.7446762	1.2301836	2.6471992
24	0.7455492	1.2418296	2.6464666
25	0.7468438	1.242826	2.6464034
26	0.7477474	1.2340808	2.6469274
27	0.749018	1.2273518	2.6473228
28	0.7499608	1.2260838	2.6473936
29	0.7512418	1.2282216	2.6472714
30	0.7522542	1.2304314	2.647147
31	0.7535404	1.2213018	2.647647
32	0.7545624	1.2173052	2.6478628
33	0.75593	1.2164586	2.6479102
34	0.7571098	1.2094716	2.6482792
35	0.7585594	1.2081368	2.648347

Table 3.6: Table of estimates of ν, γ and μ and corresponding standard errors in 2D

Chapter 4

A mathematical model for a particle near a polymer

A polymer confined by a hard wall or some other geometrical obstacle loses entropy. This loss of entropy causes a net force on the wall or obstacle. These forces have been observed experimentally [39] and modelled numerically by using self-avoiding walk models of polygons [34] and [35]

The problem we consider in this chapter is the situation where a test particle is placed near a linear polymer. The presence of the particle causes the polymer to lose conformational entropy and therefore results in a change in entropic pressure in the polymeric system. The gradient of the pressure induces a repulsive entropic force on the particle. The particle near a polymer will be expelled from its vicinity by these induced forces. (Since the polymer is much heavier compared to the particle the polymer will not be displaced.)

If we assume the particle can move freely without any friction and dissipation of energy and also assume that the polymer is kept at constant temperature, then the particle will accelerate from the higher-pressure region to the lower pressure. We suggest a model for this problem and simulate the model numerically to see the relation between velocity and the position of the particle on 2-dimensional lattice.

4.1 Entropic pressure near a walk

The canonical ensemble of a system S with states denoted by $s \in S$ is usually of fixed size, and at fixed temperature T and volume. The canonical partition function is given by

$$Z(T) = \sum_{s \in S} e^{\beta\, E_s}. \tag{4.1}$$

Here $\beta = \frac{1}{kT}$ and k is the Boltzmann constant. E_s is the total energy of the system when it is in state s. If there are $c_n(m)$ states of size n and energy m, then the above partition function is given by

$$Z_n(T) = \sum_m c_n(m)\, e^{\beta\, m}. \tag{4.2}$$

The exponent in the formula 4.1 is dimensionless; this makes β and E_s thermodynamical conjugate variables. In the models in this chapter the states will be walks, each of energy zero. That is, $E_s = 0$ so that the model may be considered to be at infinite temperature. In this case the partition function in equation 4.2 reduces to

$$Z_n(T) = \sum_{walks} e^{\beta\, E_{walk}} = \sum_{walks} 1 = c_n \tag{4.3}$$

where c_n is the number of walks of length (size) n.

If a system is at equilibrium, then the (extensive) free energy is given by $f_n(T) = k \log Z_n(T)$. Normally units are selected such that $k = 1$. That is, in the case of walks of zero energy the free energy is given by

$$f_n = \log Z_n(T) = \log c_n. \tag{4.4}$$

This is the extensive free energy, since as c_n grows bigger, f_n increases in proportion to n (note that c_n grows exponentially with n). Furthermore, the free energy per step is called the intensive free energy and is denoted by F_n. Thus

$$F_n = \frac{1}{n} \log Z_n(T) = \frac{1}{n} \log c_n. \tag{4.5}$$

If the limit $F = \lim_{n\to\infty} F_n$ exists, then the F is the limiting free energy, and the limit is also called the thermodynamic limit of the model. By equation 2.15 this limit exists and equals $\log \mu_d$.

In a system with energy $E_s = 0$, the total internal energy is also equal to zero, that is, $U = 0$. Since the free energy is given by $F = U - TS$, where U is the internal energy, T is temperature and S is the entropy of the system, comparison to equation 4.4 shows that the entropy is given by

$$TS = -\log c_n. \tag{4.6}$$

That is, since $f_n = -TS$ and assuming $T = 1$, the model has free energy purely determined by its entropy.

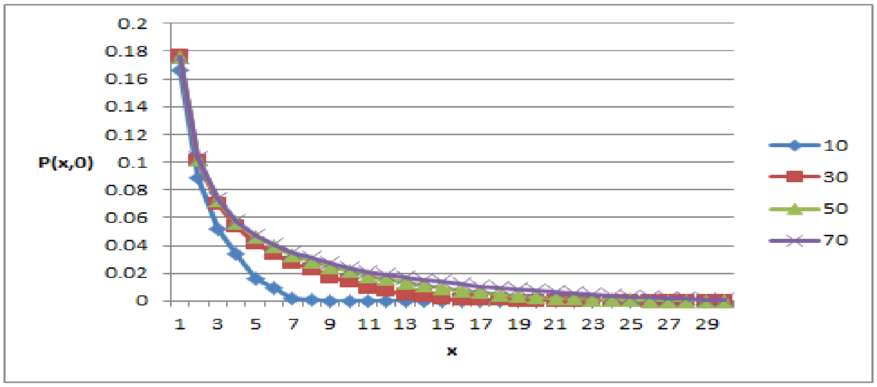

Figure 4.1: The pressure $P_n(x,0)$ along the X-axis at the points (x, 0) for x = 1; 2; 3;...; 70, and for walks of lengths n=10; 30; 50; 70.

Suppose that F_n is the free energy of a walk, and let $\Delta V(\vec{r})$ be a volume element centered at $\vec{r}$. Let $\overline{F_n(\vec{r})}$ be the free energy of the walk if it avoids the volume element $\Delta V(\vec{r})$. The change in free energy is $\Delta F_n = F_n - \overline{F_n(\vec{r})}$. In the lattice this will be the change in the free energy of the walk if the lattice site at $\vec{r}$ is excluded. This quantifies entropic pressure of the walk at $\vec{r}$, which is defined as

$$P_n(\vec{r}) = \frac{\Delta F_n}{\Delta V(\vec{r})} = -T\frac{\overline{\Delta S(\vec{r})}}{\Delta V(\vec{r})}, \tag{4.7}$$

where $\Delta\overline{S(\vec{r})}$ is the change in entropy of walks avoiding the site $\vec{r}$.

In terms of c_n the entropic pressure of walks is given as follows. Assume $c_n(\vec{r})$ is the number of SAWs of length n *passing* through the lattice site $(\vec{r})$, where $1 \leq k < n$. Let $\overline{c_n(\vec{r})}$ be the number of walks of length n excluding (avoiding) the lattice site $\vec{r}$. That is, $c_n - c_n(\vec{r}) = \overline{c_n(\vec{r})}$. Then using formula 4.6, the pressure at lattice point $\vec{r}$ is given by

$$P_n(\vec{r}) = -\log\frac{c_n - c_n(\vec{r})}{c_n} = -\log\left(1 - \frac{c_n(\vec{r})}{c_n}\right). \tag{4.8}$$

In figure 4.1 the pressure due to a particle near a walk for walks of lengths n=10; 30; 50; 70 is plotted against the position of the particle along the x-axis (i.e for the points $\vec{r} = (x,0)$ where $1 \leq x \leq 70$.) As we can see the pressure is high close to the origin and drops sharply with increasing distance. The pressure will be discussed in more details in chapter 4. Here, we consider the entropic force on a particle at a lattice point $\vec{r}$.

4.2 The entropic force and velocity of a test particle

In terms of equation 4.7 the pressure is a derivative of the free energy to unit volume: The total change in free energy is proportional to the volume of excluded sites, so that

$$\begin{aligned}\Delta F_n(\vec{r}) &= (\log c_n - \log \overline{c_n(\vec{r})})\Delta V(\vec{r}) \\ &= \log c_n - \log(c_n - c_n(\vec{r}))\Delta V(\vec{r}) \\ &= -\log(1 - \frac{c_n(\vec{r})}{c_n})\Delta V(\vec{r}).\end{aligned} \tag{4.9}$$

The pressure gradient between lattice sites $\vec{r}$ and $\vec{r}+\vec{e}$ (where $\vec{e} = (\pm 1, 0)$ or $\vec{e} = (0, \pm 1)$ causes a net force denoted by $\vec{f_n}(\vec{r}, \vec{r}+\vec{e})$. The direction of this force is on average along the steepest descent in the pressure field. The force also has components along lattice edges, and these may be computed by taking (discrete) derivatives of the pressure between adjacent lattice sites. For example, if $\vec{r}$ is a given lattice site, then

$$\vec{f_n}(\vec{r}, \vec{r}+\vec{e}) = \big(-P(\vec{r}+\vec{e}) + P(\vec{r})\big)\vec{e} \tag{4.10}$$

is the component of the force directed from $\vec{r}$ to $\vec{r}+\vec{e}$. By equation 4.8, this gives

$$\begin{aligned}|\vec{f_n}(\vec{r}, \vec{r}+\vec{e})| &= \log\frac{c_n - c_n(\vec{r}+\vec{e})}{c_n} - \log\frac{c_n - c_n(\vec{r})}{c_n} \\ &= \log(c_n - c_n(\vec{r}+\vec{e})) - \log(c_n - c_n(\vec{r})) \\ &= \log \overline{c_n(\vec{r}+\vec{e})} - \log \overline{c_n(\vec{r})} \\ &= -S_n(\vec{r}+\vec{e}) + S_n(\vec{r})\end{aligned}$$

where $S_n(\vec{r})$ is the entropy of walks if the site $\vec{r}$ is excluded by equation 4.6 (and where $T = 1$). That is, the force field is given by a change in entropy between excluding adjacent lattice sites, and so is also called an *entropic force*.

The entropic force is directed from the point of high entropic pressure to the point of low pressure. Putting test particle at a point $\vec{r}$ shows that it will experience an entropic force to points of lower pressure, and if it has mass, then it will accelerate. This occurs only if the walk is in contact with a large heat bath, and the total system is kept in equilibrium. This assumption is important because in this case the particle will take the energy from the polymer which in the absence of heat bath, will cool down.

The acceleration of a test particle due to the entropic pressure gradient give velocity $\vec{v}(\vec{r})$ to the particle at each point $\vec{r}$ along its path. In the lattice this may be quantified by noting that a change in velocity from point $\vec{r}$ to point $\vec{r}+\vec{e}$ is the acceleration $\vec{a} = (\vec{v}(\vec{r}+\vec{e}) - \vec{v}(\vec{r}))/\Delta t$ where Δt is the time interval that it takes for the particle to go from $\vec{r}+\vec{e}$ to $\vec{r}$. Assume that the *speed* $v(\vec{r}) = |\vec{v}(\vec{r})|$ is the magnitude of the velocity and $\Delta v(\vec{r}) = v(\vec{r}+\vec{e}) - v(\vec{r})$, therefore we can apply formula 4.10 to Newton's second law for a particle with unit mass to get

$$\frac{\Delta v(\vec{r})}{\Delta t} = f_n(\vec{r}, \vec{r}+\vec{e}) = -P(\vec{r}+\vec{e}) + P(\vec{r}) = -\Delta P(\vec{r}) \tag{4.11}$$

Notice that

$$\frac{\Delta v(\vec{r})}{\Delta t} = \frac{\Delta v(\vec{r})}{\Delta x}\ \frac{\Delta x}{\Delta t} \sim v(\vec{r})\ \frac{\Delta v(\vec{r})}{\Delta x} \tag{4.12}$$

The pressure gradient between lattice points $\vec{r}+\vec{e}$ and $\vec{r}$ is defined by $\Delta P(\vec{r}) = P(\vec{r}+\vec{e}) - P(\vec{r})$ and it is the pressure drop from point $\vec{r}+\vec{e}$ to $\vec{r}$. Choose $\Delta x = 1$, then equation 4.11 becomes

$$v(\vec{r})\ \Delta v(\vec{r}) \sim -\Delta P(\vec{r}) \tag{4.13}$$

Suppose that the particle is moving on x-axis, then $\vec{r} = (x, 0)$. Now, equation 4.13 can be approximated as a differential equation in x.

$$v(x,0)\ dv(x,0) = -dP(x,0) \tag{4.14}$$

Then we integrate 4.14 from y to x to obtain:

$$\frac{1}{2}\big(v^2(x,0) - v^2(y,0)\big) = P(y,0) - P(x,0) \tag{4.15}$$

Assuming that $y = 1$ we have

$$v(x,0) = \sqrt{v^2(1,0) + 2\big(P(1,0) - P(x,0)\big)} \tag{4.16}$$

So if we release at particle at (1,0) with zero velocity, then velocity at every point (x,0) where $1 \leq x$ can be calculated by

$$v(x,0) = \sqrt{2\big(P(1,0) - P(x,0)\big)} \tag{4.17}$$

For large x, P(x,0) = 0. Therefore the *terminal velocity* of particle at points far from the polymer can be determined by

$$v^{ter}(x,0) = \sqrt{2P(1,0)} \tag{4.18}$$

for a particle released at (1,0) at rest.

4.3 Numerical results

The speed of unit mass test particles released at $\vec{r} = (1, 0)$ at rest and accelerating at constant temperature along the x-axis can be calculated using equation 4.17. Furthermore, equation 4.18 gives the terminal velocity of such point. In figure 4.2 the velocity of the particle against its position on x-axis is plotted for two dimensional walks of lengths $10 \leq n \leq 70$.

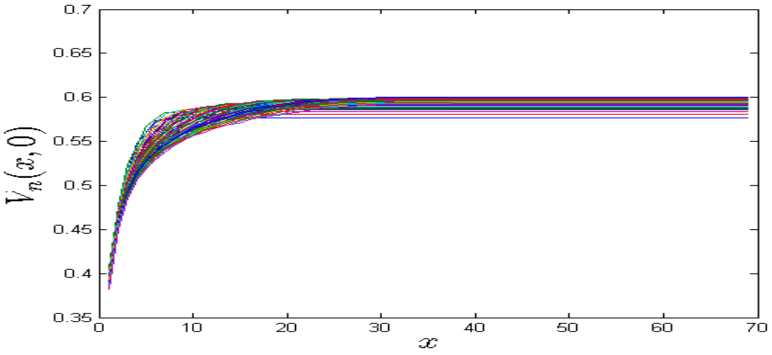

Figure 4.2: Plot of velocity of a point moving along the x-axis against its position in the square lattice. The particles were released at $(1, 0)$ and accelerated along the x-axis. The speed reaches a maximum of 0.59 independent of the length of walks. The walks were of lengths $10 \leq n \leq 70$

As we can see, the velocity increases as the point is going farther from the origin on the x-axis, until it reaches a maximum speed. Notice that the speed seems to reach a maximum independent of the length of walks considered, and for each case the terminal speed is about 0.59. For longer walks the rate at which the maximum speed is attained is smaller.

The velocity curve can be rescaled by multiplying distance by a factor of $n^{-\nu}$. In figure 4.3 we rescaled the data of figure 4.2. With increasing n, the data start to accumulate on the rescaled velocity curve. Since we chose to release the particle at (1,0), we can rescale the starting point as well. This is done in figure 4.4 by choosing the origin at x=2. As we can see the data for rescaled distance $n^{-\nu}(x - 2)$ fit better for small values of x; since the data cluster more tightly for walks of different lengths.

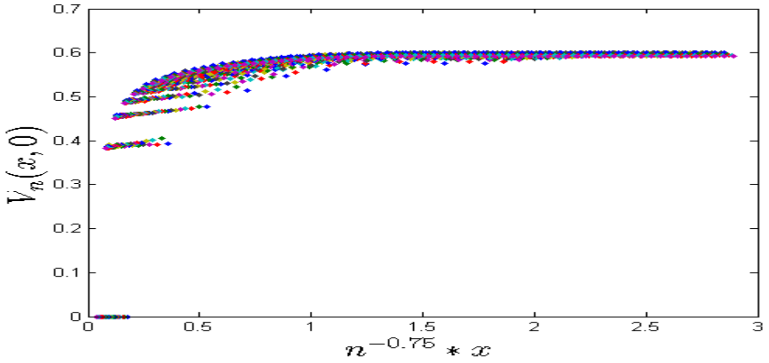

Figure 4.3: Plot of velocity of a point moving on the line x=0 against its rescaled position $n^{-0.75}(x)$ for x = 1; 2; 3;...; 70, in 2D. The walks were of lengths $10 \leq n \leq 70$

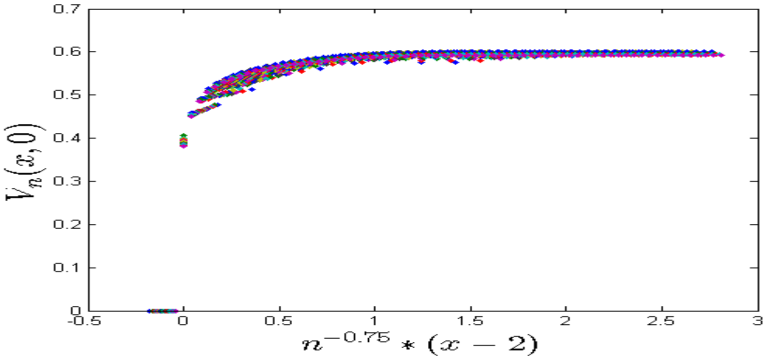

Figure 4.4: Plot of velocity of a point moving on the line x=0 against its rescaled position $n^{-0.75}(x-2)$ for x = 1; 2; 3;...; 70 in 2D. The walks were of lengths $10 \leq n \leq 70$

4.3.1 Velocity in other directions

The velocity of the particle along the diagonal line against its position ($\vec{r} = (x, x)$) is plotted in figure 4.5 for walks of lengths $10 \leq n \leq 70$. The curves follow a similar pattern as the speed along the x-axis in Figure 4.2, the difference is that the test particles gain less speed. The curves in figure 4.5 become better defined (accumulate more tightly) in figure 4.6 by rescaling the length by $\sqrt{2}\, n^{-\nu}(x-2)$. Terminal velocity converges to about 0.5 independent of length.

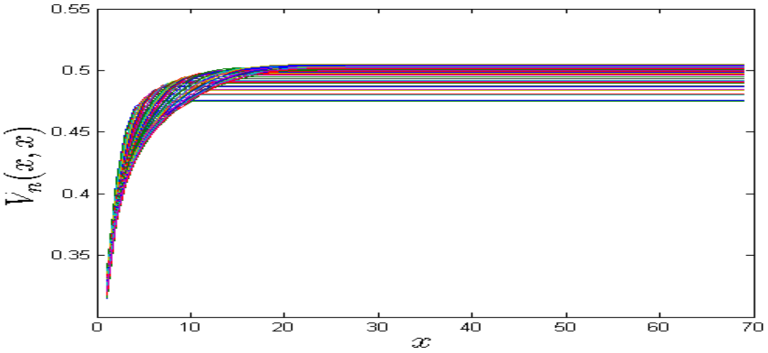

Figure 4.5: Plot of velocity of a point moving on diagonal line y=x against its position in the square lattice. The particles were released at $(1,0)$ and accelerated along the diagonal line $y = x$. The speed reaches a maximum of 0.5 independent of the length of walks. The walks were of lengths $10 \leq n \leq 70$

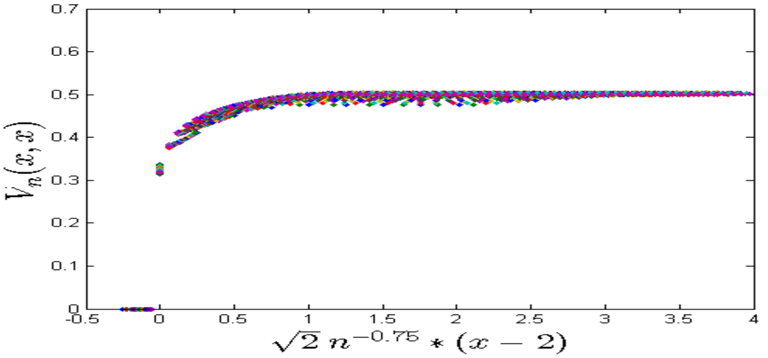

Figure 4.6: Plot of velocity of a point moving on the line y=x against its rescaled position $n^{-\nu}(x - 2)$ in 2D. The walks were of lengths $10 \leq n \leq 70$

Chapter 5

Rescaling pressure

In the previous section we mentioned that a particle placed close to a polymer will experience a pressure in the vicinity of the polymer. This pressure induces a repulsive net force on the particle. In this chapter we examine a model for entropic pressure close to a self-avoiding walk. In particular, a scaling relation is developed for the pressure.

The scaling is tested in section 5.4 using data generated by the Rosenbluth algorithm. Our numerical results verify the scaling formula (see section 5.3).

5.1 Modelling the entropic pressure near a walk

The purpose of this section is to introduce scaling formulas for walks in half-spaces and walks that end at a lattice point $\vec{r}$. These formulas will prove useful in the modelling of the entropic pressure near a walk.

Let L be a line through the origin in $\mathbb{R}^2$. This line cuts $\mathbb{R}^2$ into two half-lattices. This can be generalized in higher spaces, where L is a hyperplane of dimension $d-1$.

Denote the number of self-avoiding walks of length n from the origin confined to one of the half-spaces (on one side of L) by c_n^+. It is believed that c_n^+ has an asymptotic formula

$$c_n^+ \simeq C_1 n^{\gamma_1 - 1} \mu^n \tag{5.1}$$

where $\gamma_1 = 61/64$ in two-dimensions [41].

Define $c_n^{end}(\vec{r})$ to be the number of SAWs of length n rooted at $\vec{0}$ ending at $\vec{r}$ in the square lattice $\mathbb{L}^2$. Let $c_n^{end}(r) = \sum_{|\vec{r}|=r} c_n^{end}(\vec{r})$ be the number of walks of length n from $\vec{0}$ to a distance r from the origin. Since these walks end on a circle of radius r, we can assume that

$$c_n^{end}(r) \simeq A_0 r \; c_n^{end}(\vec{r}) \tag{5.2}$$

in $\mathbb{L}^2$ (where A_0 is a constant).

The average end-to-end distance of self-avoiding walks of length n is denoted by $R_e = \langle r \rangle_n$ and has a scaling relation introduced in equation 2.30. The expected distance of the endpoint of the walk from the origin can be computed using $c_n^{end}(r)$:

$$R_e = \langle r \rangle_n = \frac{1}{c_n} \sum_{r \geq 0} r \; c_n^{end}(r) \sim C n^{\nu} \tag{5.3}$$

where ν is the metric exponent of the self-avoiding walk and is known in two dimensions to be exactly $\nu = 3/4$.

The ratio $c_n^{end}(r)/c_n$ is the end-to-end distribution function $\Pr(n, r)$ of self-avoiding walks. It is the probability density of self-avoiding walks of n steps which reach the distance $r = |\vec{r}|$ [25]. In particular it is thought that for large n the distribution in d-dimensions approaches a limiting shape,

$$\Pr(n, r) \sim R_e^{-d} F_0\big(\frac{r}{R_e}\big) \tag{5.4}$$

The asymptotic expression above is the inverse of the volume occupied by the end point of the walk multiplied by a scaling function F_0 (which is a function of the distance of the walk from the origin scaled by R_e). The short- and long-range behaviors of $F_0(x)$ were determined in terms of γ and ν [24].

$$F_0(x) = \begin{cases} x^g & \text{if } x \text{ is small} \\ e^{-x^{\delta}} & \text{if } x \text{ is large} \end{cases} \tag{5.5}$$

where $g = \frac{\gamma - 1}{\nu}$ [24]and $\delta = \frac{1}{1-\nu}$ [25]. Thus in terms of $\Pr(n, r)$ we have

$$\begin{aligned} c_n^{end}(\vec{r}) &\sim \Pr(n, r) \; c_n \\ &= C_0 \frac{1}{R_e^d} F_0\big(\frac{r}{R_e}\big) c_n \\ &= C_0 r^{-d} \big(\frac{r}{R_e}\big)^d F_0\big(\frac{r}{R_e}\big) c_n \end{aligned} \tag{5.6}$$

Let $x^d F_0(x) = G(x)$, then we have

$$c_n^{end}(\vec{r}) \sim r^{-d} G\big(\frac{r}{R_e}\big) c_n \tag{5.7}$$

Substituting c_n from 2.21 gives:

$$c_n^{end}(\vec{r}) \sim n^{\gamma - 1} \mu^n r^{-d} G\big(\frac{r}{R_e}\big) \tag{5.8}$$

Note that $R_g \sim R_e \sim C n^{\nu}$, where R_g^2 is the mean square radius of gyration of SAWs.

5.2 The number of walks passing through a point $\vec{r}$

In order to find the pressure due to a particle near a walk, we need to determine a scaling formula for the number of walks passing through a lattice point $\vec{r}$.

A walk of length n passing through $\vec{r}$ (denoted by $c_n(\vec{r})$) can be divided into two subwalks that do not cross each other (see figure 5.1). A subwalk of length k starting at $\vec{0}$ ending at $\vec{r}$ and a subwalk of length $n-k$ starting at $\vec{r}$ in a half-space. Since these two walks are assumed not to intersect, the second walk is modeled by growing a walk in half-space separated by a wall normal to and passing through $\vec{r}$. Therefore, we have

$$c_n(\vec{r}) \sim \sum_{k=0}^{n} c_k^{end}(\vec{r}) c_{n-k}^{+} \tag{5.9}$$

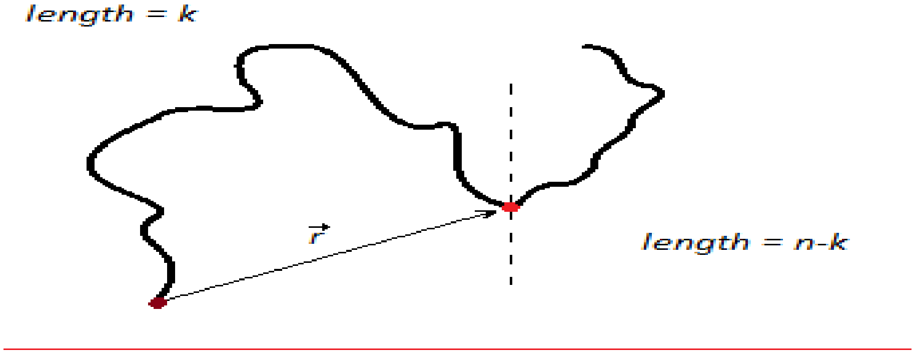

Figure 5.1: A walk of length n divided into two subwalks; a subwalk of length k ending $\vec{r}$ and a subwalk of length $n-k$ in half-space starting at $\vec{r}$.

Substitute the number of walks that ends at $\vec{r}$ and the number of walks in a half-space from scaling formulas 5.6 and 5.1 to obtain:

$$c_n(\vec{r}) \sim C_0 C_1 \mu^n \int_k r^{-2} (\frac{r}{R_g})^d F_0(\frac{r}{R_g}) k^{\gamma-1} (n-k)^{\gamma_1 - 1} dk \tag{5.10}$$

where the summation is approximated by an integral. Since d=2, the above formula simplifies to:

$$c_n(\vec{r}) \sim C_0 C_1 \mu^n \int_k R_g^{-2} F_0(\frac{r}{R_g})\ k^{\gamma-1} (n-k)^{\gamma_1 - 1} dk \tag{5.11}$$

Scale distance by $r = a\,R_g$, where $R_g \simeq Cn^{\nu}$ is the root of the mean square radius of gyration of the walk, and a is the scaling factor. Then $\vec{r} = (a\,R_g)\frac{\vec{r}}{|\vec{r}|}$ and $c_n(\vec{r})$ becomes a function of a. Let

$$\hat{c}_n(a) = c_n\Big((aR_g)\frac{\vec{r}}{|\vec{r}|}\Big) \tag{5.12}$$

The function $\hat{c}_n(a)$ is the number of walks passing through a point at distance aR_g from the origin and in the direction of $\vec{r}$.

F_0 in equation 5.10 becomes

$$F_0(\frac{r}{R_g}) = F_0\big(\frac{aCn^{\nu}}{Ck^{\nu}}\big) = F_0\big(a\big(k/n\big)^{-\nu}\big) \tag{5.13}$$

Replace F_0 and R_g in 5.11 and put $C_3 = C^{-2}C_0C_1$ to see that

$$\hat{c}_n(a) \sim C_3\mu^n \int_k k^{-2\nu}F_0\big(a\big(k/n\big)^{-\nu}\big)\; k^{\gamma-1}(n-k)^{\gamma_1-1}dk \tag{5.14}$$

In other words,

$$\hat{c}_n(a) \sim C_3 n^{-2\nu}n^{\gamma-1}n^{\gamma_1-1}n\;\mu^n \int_k \big(k/n\big)^{-2\nu}F_0\big(a\big(k/n\big)^{-\nu}\big)\;\big(k/n\big)^{\gamma-1}\big(1-k/n\big)^{\gamma_1-1}\;d(k/n). \tag{5.15}$$

Substituting $x = k/n$ gives:

$$\hat{c}_n(a) \sim C_3 n^{-2\nu+\gamma+\gamma_1-1}\mu^n \int_0^1 x^{-2\nu}F_0(ax^{-\nu})\;x^{\gamma-1}(1-x)^{\gamma_1-1}dx \tag{5.16}$$

Define the function

$$g(a) = \int_0^1 x^{-2\nu}F_0(ax^{-\nu})\;x^{\gamma-1}(1-x)^{\gamma_1-1}dx \tag{5.17}$$

where F_0 is defined in equation 5.5. Therefore, we have

$$\hat{c}_n(a) \sim C_3 n^{-2\nu+\gamma+\gamma_1-1}\mu^n g(a) \tag{5.18}$$

Notice that $\hat{c}_n(a)$ is only dependent on the distance from the origin and not on the direction. Dependence of $\hat{c}_n(a)$ on a is given by $g(a)$. By 5.5 as a increases g(a) decays quickly to zero. Plot 5.2 shows the dependence of $g(a)$ on a on a log-log scale. For small values of a the plot decrease slowly but decays to zero quickly as $\log(a)$ approaches to 0.

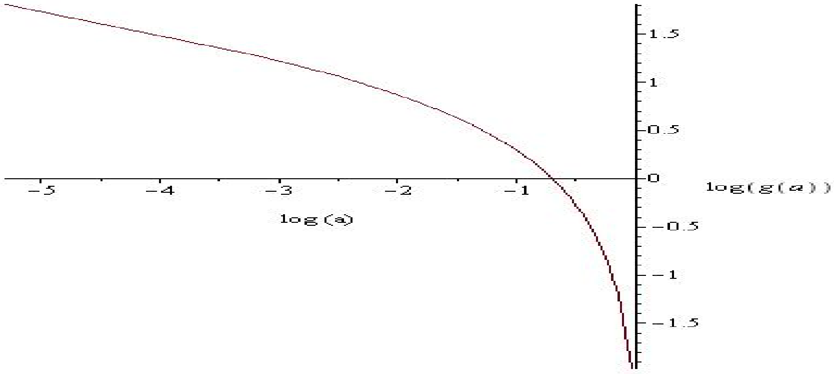

Figure 5.2: Plot of $\log(g(a))$ against $\log(a)$

5.3 The rescaled pressure

The pressure at the point $\vec{r}$ due to a polymer placed at $\vec{0}$ is given by formula 4.8. Let $r = |\vec{r}|$ and let $P_n(r)$ be the mean pressure at a distance r from the origin (that is the average over all directions). The scaling of $P_n(r)$ can be found by rescaling distance by $r = aR_g$, where R_g is the mean radius of gyration. Define the rescaled pressure by

$$\mathbf{P}_n(a) = P_n(aR_g) \tag{5.19}$$

where $0 \leq a$ and $R_g = Cn^\nu$ defines the length scale. Then, equation 4.8 gives the pressure of a polymer at a distance aR_g from the origin:

$$\mathbf{P}_n(a) = -\log\big(1 - \frac{\hat{c}_n(a)}{c_n}\big) \tag{5.20}$$

where c_n is the number of walks of length n and $\hat{c}_n(a)$ is the number of polymers passing through a point a distance aR_g of the origin. To find the scaling of the pressure we substitute for c_n and $\hat{c}_n(a)$ from equations 2.21 and 5.18. In particular, for a not too small and n large we have

$$\mathbf{P}_n(a) \sim -\log\Big(1 - \frac{C_3 n^{-2\nu+\gamma+\gamma_1-1}\mu^n g(a)}{Cn^{\gamma-1}\mu^n}\Big)$$

Let

$$A = \frac{C_3 n^{-2\nu+\gamma+\gamma_1-1}\mu^n g(a)}{Cn^{\gamma-1}\mu^n} \tag{5.21}$$

So that $\mathbf{P}_n(a) \sim \log(1-A)$. Since $-2\nu + \gamma + \gamma_1 - 1 \leq 0$ and since $g(a)$ decays to zero quickly as a increases, $A \to 0$. Thus $\mathbf{P}_n(a) \sim 1 - (1-A) + O(A^2) \sim A$, or

$$\mathbf{P}_n(a) \sim \frac{C_3 n^{-2\nu+\gamma+\gamma_1-1}\mu^n g(a)}{C n^{\gamma-1}\mu^n} \tag{5.22}$$

For some constant C_4 the above simplifies to

$$\mathbf{P}_n(a) \sim C_4 g(a) n^{-2\nu+\gamma_1} \tag{5.23}$$

Using the Flory value $\nu = 3/4$ and exact value $\gamma_1 = 61/64$ gives the following relation in d=2:

$$n^\rho \mathbf{P}_n \sim g(a) \tag{5.24}$$

where $\rho = 2\nu - \gamma_1$ which equals to $35/64$ in 2-dimensions. Note that right-hand side is independent of n.

Since the lattice is rescaled by a factor of $n^{-\nu}$, we can test the prediction in 5.24 numerically by plotting $n^\rho \mathbf{P}_n$ against $\vec{r}/n^\nu$. This should collapse data for different choices of $\vec{r}$ and n into a single curve which has similar shape to figure 5.2 and is a function of a only.

5.4 Numerical results

Monte Carlo simulations using the Rosenbluth algorithm generate a sample of 10^6 walks of length $1 \leq n \leq 70$ in two dimensions. From this sample we find the number of walks passing through the lattice sites of the form $(\pm x, \pm x)$ and $(\pm x, \mp x)$ for $1 \leq x \leq 70$. If we take the average of these four groups of data, we get an estimate for the number of walks in $\mathbb{L}^2$ passing through the points (x, x), where $1 \leq x \leq 70$.

We substitute these data in equation 4.8 to calculate the pressure at the point $\vec{r}$ in two-dimensions. For polymers of length n the pressure at lattice points with $\frac{n}{2} \leq |x|$ and $\frac{n}{2} \leq |y|$ is zero, since a polymer rooted at origin can not pass through such points. Table 5.1 shows the results for c_n and $\overline{c_n(1,1)}$ and the corresponding pressure at this point in 2D. Similarly, the pressure at points $(x, 0)$ with $1 \leq x \leq 70$ can be estimated numerically.

5.4.1 The pressure near a polymer

The pressure near walks is computed from the data similar to the data in Table 5.1. In Figure 5.3 the pressure of the polymer along points on the x-axis is plotted as a function of the distance from the origin; that is $P_n(x, 0)$ is plotted against x for $1 \leq n \leq 70$.

n	c_n	$\overline{c_n(1,1)}$	$P_n(1,1)$
27	$8.8e+011$	$2.17E+11$	0.1231
28	$2.35e+012$	$5.81E+11$	0.1232
29	$6.27e+012$	$1.56E+12$	0.1238
30	$1.67e+013$	$4.16E+12$	0.1244
31	$4.46e+013$	$1.11E+13$	0.1242
32	$1.19e+014$	$2.97E+13$	0.1246
33	$3.17e+014$	$7.92E+13$	0.1248
34	$8.46e+014$	$2.12E+14$	0.1251
35	$2.25e+015$	$5.63E+14$	0.1252

Table 5.1: Table of estimates of c_n , $\overline{c_n(1,1)}$ and $P_n(1,1)$ for walks of length $27 \leq n \leq 35$ in 2D

Figure 5.3 shows that the pressure increases quickly as we get closer to the origin in the xy-plane. Since the origin is always occupied, the pressure at this point is infinity.

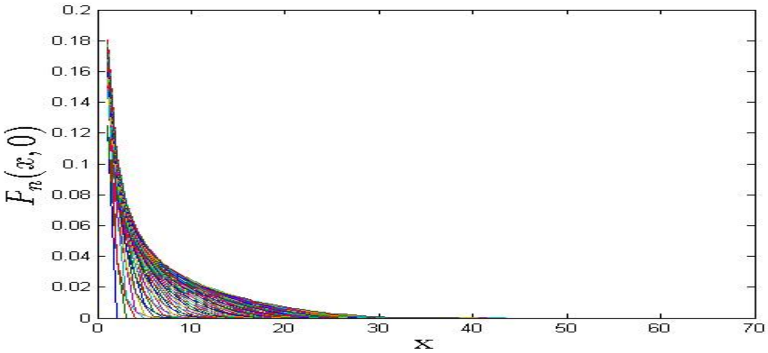

Figure 5.3: Plot of pressure on the points (x,0) on the x-axis against its position in 2D. The curves represent the pressure along the x-axis for polymers of length $1 \leq n \leq 70$

The scaling of pressure can be uncovered using equation 5.24. That is, rescale the distance by a factor of $n^{-\nu}$ and rescale the pressure by n^{ρ}. The scaling analysis in previous section predicts that plotting $n^{\rho}\mathbf{P}_n$ against $n^{-\nu}|r|$ will collapse data of all length on a single curve.

Using the exact estimates for $\nu = 3/4$ and $\gamma_1 = 61/64$ in two-dimensions, we get $\rho = 2\nu - \gamma_1 = 35/64$. Hence, the rescaled data $n^{35/64}\mathbf{P}_n$ should be plotted against $n^{-0.75}|\vec{r}|$. This is done in Figure 5.4 where data of Figure 5.3 is used. As it can be seen, the rescaled pressure $\mathbf{P}_n$ for all lengths $1 \leq n \leq 70$ aligns along a single curve. This is strong evidence in support of the scaling relation derived in 5.24.

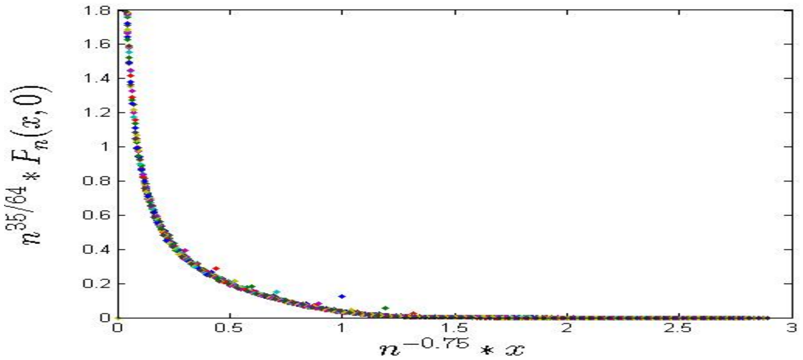

Figure 5.4: Testing the scaling prediction in equation 5.24. The rescaled pressure $n^{35/64}\mathbf{P}_n$ plotted as a function of $n^{-3/4}x$. These data include all the data points in figure 5.3. The data collapse to a single curve, uncovering the scaling function g(a) in equation 5.24

Equation 5.24 suggests that $n^{\rho}\mathbf{P}_n$ can be approximated by the function g(a). To test this proposition, we plot the numerical data of Figure 5.4 in log-log axes (see Figure 5.6) and compare it with the theoretically calculated plot of g(a) in Figure 5.2 . As it can be seen, Figures 5.6 and 5.2 have similar shape and this shows that the scaling approximation of pressure made in equation 5.24 is correct and can be determined as a function of distance. As expected, the pressure is high for lattice points close to the origin. But as we pass a distance of roughly $r \sim n^{\nu}$ (or $a \sim 1$) the pressure drops sharply to zero. This crossover happens at the *boundary layer* of the polymer, which may be considered a gas of monomers localised inside its mean radius (see picture 5.5.)

The turnover at the *boundary* of the polymer defines a *surface layer*. Inside the surface layer a density of vertices is occupied by the lattice polymer, that causes the non-zero entropic pressure which decreases with increasing distance. Once we pass the surface layer, the pressure decreases quickly, and, as we go farther from the origin, the

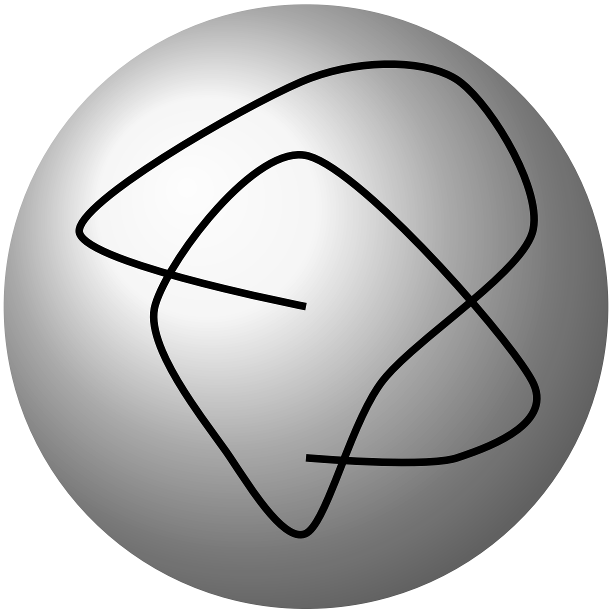

Figure 5.5: Schematic illustration of boundary layer of polymer. Inside the surface layer there is a density of vertices occupied by the lattice polymer. This underlies the entropic pressure which decreases with increasing distance.

pressure drops to zero. This can be seen in the sharp descent of $n^{\rho}\mathbf{P}_n$ with increasing distance in Figure 5.6 for $1 \leq a$.

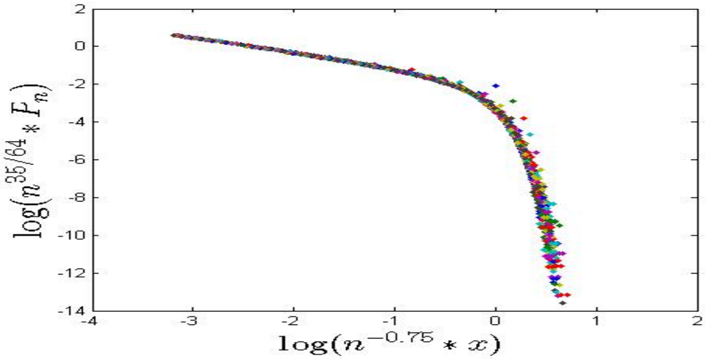

Figure 5.6: The same data as in Figure 5.4, but on a log-log scale. These data accumulate along a curve which is similar in shape to g(a) plotted in 5.2, but with rescaled axes.

5.4.2 Pressure in other directions

The scaling of $P_n(a)$ in equation (27) suggests that the pressure field is isotropic, that is, the same in all directions. This can be tested numerically by examining the rescaled pressure along other directions in the lattice.

Consider first the pressure $P_n(x, x)$ along the points on the diagonal direction in the lattice. The pressure is a function of the distance $|r| = ||(x, x)||_2 = \sqrt{2}\ x$ from the origin. Therefore, if we rescale the distance by $\sqrt{2}\ xn^{-0.75}$ and plot it against the rescaled pressure $n^{35/64}\mathbf{P}_n$, we should see that the data is collapsed on a single curve again. This is confirmed in Figure 5.7 for $1 \leq n \leq 70$.

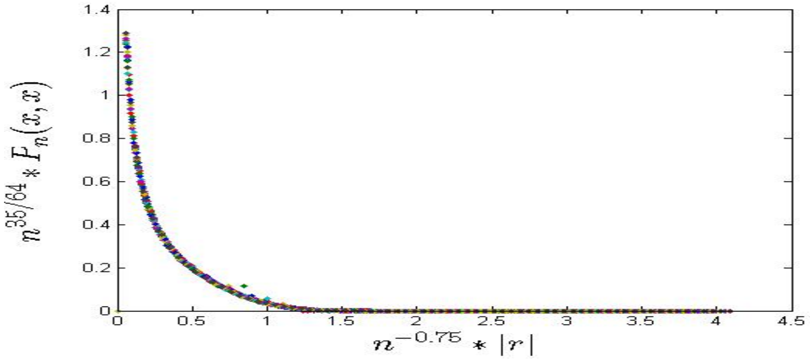

Figure 5.7: Testing the scaling prediction in equation 5.24. The rescaled pressure $N^{35/64}\mathbf{P}_n(x, x)$ plotted as a function of $\sqrt{2}xn^{-3/4}$. The data collapse to a single curve, uncovering the scaling function g(a) in equation 5.24

Comparing Figures 5.7 to 5.4 shows a similar pattern in the rescaled data. So in Figure 5.8 we plotted the data of figures 5.7 and 5.4 on the same scale and axes. Since the scaling along the diagonal is the same as along the x-axis, this supports the suggestion that the pressure field is isotropic, even at small distances in $\mathbb{L}^2$.

The data for pressure on vertices (x,0) and (x,x) from figures 5.7 and 5.4 are plotted in figure 5.9 on logarithmic axes. The result is a graph similar to Figure 5.6 with a well-defined surface layer. For a particle approaching the surface layer, the relative pressure gradient is small at first. As the surface layer is reached, the pressure grows sharply. The pressure gradient increases more gently as the surface layer is crossed towards the vertices closer to origin. This is seen in Figure 5.9 for small values of a, as the surface layer is crossed and we get closer to the origin.

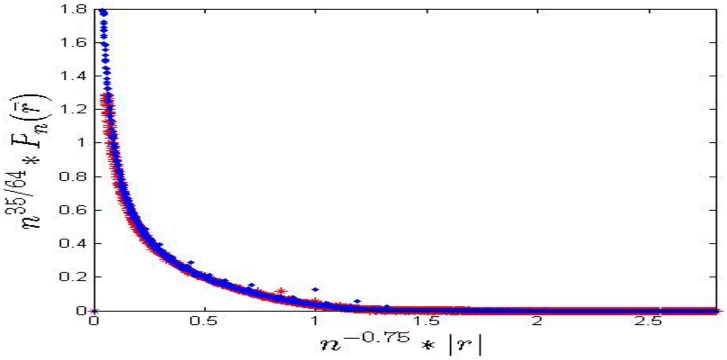

Figure 5.8: The data in Figures 5.7 and 5.4 plotted on the same scale and axes.

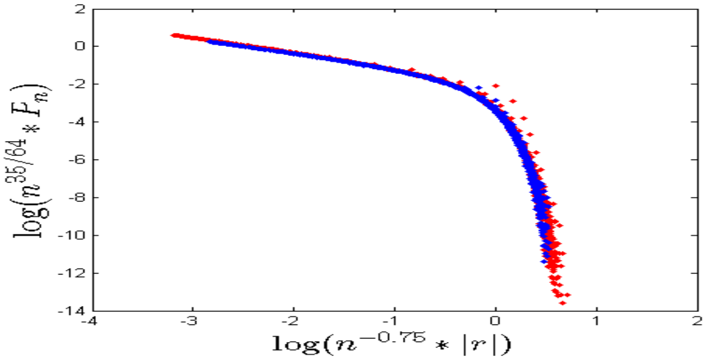

Figure 5.9: The data in Figures 5.7 and 5.4 are plotted on the same log-log scale. The blue data represent the pressure along $r = (x, x)$ and the red data is from figure 5.4. This data accumulate along a curve which is similar in shape to g(a) plotted in 5.2, but with rescaled axes.

Chapter 6

Conclusions

The first problem examined in this thesis is finding the number of different conformations of a polymer of a given length. We used the Rosenbluth algorithm for approximate enumeration of self-avoiding walks as a model of polymer entropy in two and three dimensions. In addition, we have calculated some of the polymer properties such as radius of gyration and end-to-end distance.

Monte Carlo methods for the calculation of the number of walks, radius of gyration and other properties walks produces data which can be used to examine the scaling formulas for walks.In this thesis we fitted our numerical results to the scaling relations to estimate critical exponents. The methods we used are approximate enumeration and linear regression. The results show acceptable fitness to those of published papers.

Improved computing power and better approximate enumeration data may make it possible to apply more sophisticated series analysis techniques such as Pade or differential approximants to Monte Carlo data in later work. Also using Monte Carlo simulation to find the correction-to-scaling exponent in scaling formulas would be an interesting question to consider.

We developed a model to determine the scaling of entropic pressure in the vicinity of a linear polymer rooted at the origin in the square lattice. The scaling relation was tested numerically by collecting data using Rosenbluth sampling of self-avoiding walks in $\mathbb{L}^2$.

The data gathered in figures 5.7 and 5.4 support the scaling relation 5.24. These data show that the pressure is independent of length of polymer. Moreover, the pressure is decaying at the same rate in every direction (pressure is isotropic.) In particular, figure 5.9 reveals a similar shape as in theoretically driven plot of g(a) in 5.2.

We also studied the speed of a unit mass particle accelerating away from the vicinity of the polymer due to pressure gradient. Our numerical results suggest that the terminal velocity is independent of the size of polymer - and only dependent on the initial position of the particle.

In future we can use similar models for a directed path in the vicinity of a hard wall. Such a path will exert pressure on the wall because of loss of entropy. The pressure at a particular point may be estimated by approximating the loss of entropy if the point is excluded from the path [34].

Bibliography

[1] Janse van Rensburg, E. J. and Prellberg, T. (2013)
The pressure exerted by adsorbing directed lattice paths and staircase polygons
J. Phys. A: Math. Theor.

[2] Flory P (1942)
Thermodynamics of High Polymer Solutions
J. Chem. Phys. 10, 51

[3] T, Alfrey, A, Bartovics, and H. Mark, (1942)
Thermodynamics of High Polymer Solutions
J. Am. Chern. Soc. 64, 1557

[4] Flory P (1949)
The configuration of real polymer chains
J. Chem. Phys. 17, 303

[5] Jensen I (2004)
Enumeration of self-avoiding walks on the square lattice
J. Phys. A: Math. Gen. 37 550324

[6] Clisby N, Liang R and Slade G (2007)
Self-avoiding walk enumeration via the Lace expansion
J. Phys. A: Math.Theor. 40 109731017

[7] Duplantier B (1986)
Polymer network of fixed topology: renormalization, exact critical exponent γ in two dimensions
Phys. Rev. Lett. 57 9414

[8] J. M. Hammersley (1962)
Generalization of the Fundamental Theorem on Subadditive Functions.
Mathematical Proceedings of the Cambridge Philosophical Society, 58 , pp 235-238
doi:10.1017/S030500410003646X

[9] Einar Hille, Ralph Saul Phillips (1957)
Functional Analysis and Semi-Groups
American Mathematical Soc., pp 243

[10] De Gennes, P. G. (1979)
Scaling concepts in polymer physics
Cornell university press. pp 43

[11] Duplantier B (1990)
Renormalisation and conformal invariance for polymers
Fundamental Problems in Statistical Mechanics VII ed H van Beijeren (Amsterdam: Elsevier) pp 171223

[12] Grassberger P (1993)
Monte Carlo simulations of 3D self-avoiding walks
J. Phys. A: Math. Gen. 26 276976

[13] M L Huggins(1942)
Thermodynamic properties of solutions of long-chain compounds
Ann. N. Y. Acad. Sci. 43, 1

[14] W. Kuhn, R. Pasternak, and H. Kuhn(1947)
Mechanische und optische eigenschaften von gequollenem kautschuk Helv. Chim. Acta, 30, 1705

[15] Maurice L. Huggins(1939)
The Viscosity of Dilute Solutions of Long-Chain Molecules. III. The Staudinger Viscosity Law
J. Appl. Phys. 10, 700

[16] http://sar.informatik.hu-berlin.de/teaching/2008-w/2008-w

[17] O'Brien(1990)
Monotonicity of the number of self-avoiding walks
J. Stat. Phys. Volume 59, Issue 3-4, pp 969-979

[18] Kesten(1964)
On the Number of Self-Avoiding Walks. II
J. Math. Phys. 5, 1128

[19] Nidras(1996)
Grand canonical simulations of the interacting self-avoiding walk model
J. Phys. A: Math. Gen. 29, 7929

[20] Clisby N, Liang R and Slade G (2007)
Self-avoiding walk enumeration via the Lace expansion
J. Phys. A: Math.Theor. 40, 10973

[21] Jensen I and GuttmannAJ(1999)
Jensen I and GuttmannAJ 1999 Self-avoiding polygons on the square lattice
J. Phys. A:Math. Gen. 32, 4867

[22] Flory(1971)
Principles of Polymer Chemistry
Cornell University Press, Chap XII

[23] Cardy J L and Saleur H (1989)
Universal distance ratios for two dimensional polymers
J. Phys. A: Math. Gen. 22 L601L604

[24] des Cloizeaux J (1974)
Lagrangian theory for a self-avoiding random chain
Phys. Rev.

[25] Fisher M E (1966)
Shape of a self-avoiding walk or polymer chain
J. Chem. Phys.

[26] Torrie G M and Valleau J P (1977)
Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling
J. Comput. Phys 23 18799

[27] Rechnitzer A and Janse van Rensburg E J (2008)
Generalized atmospheric Rosenbluth methods (GARM)
J. Phys. A: Math. Theor. 41 442002

[28] Rosenbluth M N and Rosenbluth A W (1955)
Monte Carlo calculation of the average extension of molecular chains
J. Chem. Phys. 23 3569

[29] Jensen I 2004
Enumeration of self-avoiding walks on the square lattice
J. Phys. A: Math. Gen. 37 550324

[30] Clisby N, Liang R and Slade G (2007)
Self-avoiding walk enumeration via the Lace expansion
J. Phys. A: Math. Theor. 40 109731017

[31] Hsu H-P, Nadler W and Grassberger P (2004)
Scaling of star polymers with 180 arms
Macromolecules 37 465863

[32] Rechnitzer A and Janse van Rensburg E J (2002)
Canonical Monte Carlo determination of the connective constant of self-avoiding walks
J. Phys. A: Math. Gen. 35 L60512

[33] Guttmann A J and Conway A R (2001)
Square lattice self-avoiding walks and polygons
Ann. Comb. 5 31945

[34] F Gassoumov and E J Janse van Rensburg (2013)
The entropic pressure of a lattice polygon
J. Stat. Mech

[35] E J Janse van Rensburg (2014)
The entropic pressure of lattice knots
J. Stat. Mech

[36] Li B, Madras N and Sokal A D (1995)
Critical exponents, hyperscaling, and universal amplitude ratios for two and three dimensional self-avoiding walks
J. Stat. Phys 80 661754

[37] Saleur H (1987)
Conformal invariance for polymers and percolation
J. Phys. A: Math. Gen. 20 45560

[38] Guida R and Zinn-Justin J (1998)
Critical exponents of the N-vector model
J. Phys. A: Math. Gen 31 810321

[39] Carignano M A and Szleifer I (1995)
On the structure and pressure of tethered polymer layers in good solvent
Macromolecules

[40] Metropolis N, et al. (1953)
Equation of state calculations by fast computing machines
The journal of chemical physics 21.6

[41] Cardy J L (1983)
Conformal invariance, Phase Transitions and Critical Phenomena
vol 11, ed C Domb and J L Lebowitz (New York: Academic) pp 55-126

[42] Hammersley J M and Morton K W (1954)
Poor mans Monte Carlo
J. R. Stat. Soc. B 16 2338

[43] Hammersley J M and Welsh D J A (1962)
Further results on the rate of convergence to the connective constant of the hypercubic lattice
Q. J. Math. Oxford 13 10810

[44] Guttmann A J (1989)
On the critical behaviour of self-avoiding walks: II
J. Phys. A: Math. Gen. 22 280713

[45] Jensen I (2003)
A parallel algorithm for the enumeration of self-avoiding polygons on the square lattice
J. Phys. A: Math. Gen. 36 573145

[46] Jensen I (2004)
Enumeration of self-avoiding walks on the square lattice
J. Phys. A: Math. Gen. 37 550324

[47] Torrie G M and Whittington S G (1975)
Exact enumeration of neighbour-avoiding walks on the tetrahedral and body-centred cubic lattices
J. Phys. A: Math. Gen. 5 117884

[48] Clisby N, Liang R and Slade G (2007)
Self-avoiding walk enumeration via the Lace expansion
J. Phys. A: Math. Theor. 40 109731017

[49] Nienhuis B (1982)
Exact critical point and critical exponents on O(n) models in two dimensions
Phys. Rev. Lett. 49 106265

[50] Duplantier B and Saleur H (1987)
Exact tricritical exponents for polymers at the point in two dimensions
Phys. Rev. Lett. 59 53942