

中国科学技术大学

学士学位论文



T^n 格点上逾渗的几何结构

作者姓名： 朱沛俊

学科专业： 理论物理专业

导师姓名： 邓友金 教授

完成时间： 二〇一六年 五月

University of Science and Technology of China
A dissertation for bachelor's degree



Geometric Structure for Percolation in T^n Lattice

Author's Name: Peijun Zhu

Speciality: Theoretical Physics

Supervisor: Prof. Youjin Deng

Finished Time: May, 2016

摘 要

基于时间复杂度为 $O(n)$ 的对逾渗基团中的键分类的算法，实现了相关的分类程序。首先对比文章^[1]验证了二维在逾渗阈值附近的几何结构的相关分形维度。

在此之后对三维的逾渗模型进行了研究，得到了相关的分形维度数据，这个结果与二维情形一致：尽管枝叶所占的比例很大，去掉所有枝叶对分形维度并没有影响；而比例极低的枢纽剪除后对最大丛大小的分形维度影响非常大，使得分形维度从 2.52237 下降到 1.87246。

我们还研究了二维以及高维情形下缠绕类型的分类及其计算，在二维情况下由于一条线可以将平面分割为两部分，每个环的缠绕数只能是最简整数比。整个基团的缠绕情况是：一旦出现不同种的环，那么这个基团就等价于“田”型基团。高维情况下，环的缠绕数没有任何限制，而且一个基团中所有环的缠绕数可以“线性叠加”，组合成其他的缠绕可能性

关键词：逾渗模型 分形维度 键逾渗 几何结构 缠绕

Abstract

We implemented a program to classify bonds of a percolation configuration, based on a new $O(n)$ algorithm. We first compare our $2D$ result with article^[1].

Then we researched about the $3D$ percolation, and we get the fractal dimension. The result is similar to $2D$ condition: although the leaf take the main role in the overall quantity, its deletion does not cause a lower fractal dimension, whereas the deletion of junction affect the dimension greatly.

We also researched in the wrapping classification for $2D$ and higher dimension. The $2D$ is special, because a line can divide a plane to two parts. The wrapping vector for each circuit can only be irreducible. Superposition of different wrapping vector in a cluster cause a crossing. For $D > 3$ we will get all possible wrapping vector. And these vectors follow the superposition principle.

Key Words: Percolation Model, Fractal Dimension, Bond Percolation, Geometrical Structure, Wrapping

目 录

摘要	I
Abstract	II
第 1 章 简介	1
1.1 逾渗的基本知识	1
1.1.1 定义	1
1.1.2 逾渗的普适性	2
1.1.3 逾渗基团	2
1.2 T^n 上逾渗的缠绕拓扑结构	4
1.2.1 二维情形	5
1.2.2 高维情形	6
1.3 逾渗基团的几何结构	7
1.3.1 T^2 格点	8
1.4 逾渗的应用	9
第 2 章 算法及分析	10
2.1 剪枝	10
2.1.1 原理	10
2.1.2 算法、步骤与复杂度	11
2.2 回溯确定非桥	11
2.2.1 原理	11
2.2.2 算法、步骤与复杂度	12
2.2.3 链接过程的复杂度	14
2.3 稠化操作	15
2.4 缠绕的计算	16
2.4.1 原理	16
2.4.2 算法、步骤与复杂度	17
第 3 章 三维下数据结果与分析	18
3.1 运行相关的数据	18
3.2 各种组分的密度	18

3.3 最大基团的分形维度.....	20
3.4 最大的枝叶随尺度的变化.....	20
第 4 章 对未来工作的展望.....	23
参考文献.....	24
附录 A 论文与程序源代码下载.....	25
附录 B 致谢.....	26

图目录

1.1 $L = 64, p = 0.5 = p_c$ 时的键逾渗	1
1.2 $L = 24, p$ 不同时产生的逾渗	3
1.3 3D 情况下, 不同宽度的缠绕概率与置键概率的关系图	4
1.4 键的分类	8
2.1 $L = 64, p = 0.5 = p_c$ 时的键逾渗	10
2.2 N_1/N_2 的比值与置键概率 p 的关系	14
2.3 $L = 24, p = 0.48$ 稠化前后的图像对比	15
3.1 枝叶的比例随 $L^{-1.8596}$ 的变化关系	18
3.2 枢纽的比例随 $L^{-1.8596}$ 的变化关系	19
3.3 非桥的比例随 $L^{-1.8596}$ 的变化关系	19
3.4 去除枝叶后的分形维度	20
3.5 去除桥后的分形维度	21
3.6 最大枝叶随 $\ln(L + L_0)$ 的变化关系	22

算法索引

2.1 剪枝算法	11
2.2 通过回溯来标记非桥	12
2.3 回溯的详细过程	13
2.4 稠化	16
2.5 计算缠绕	17

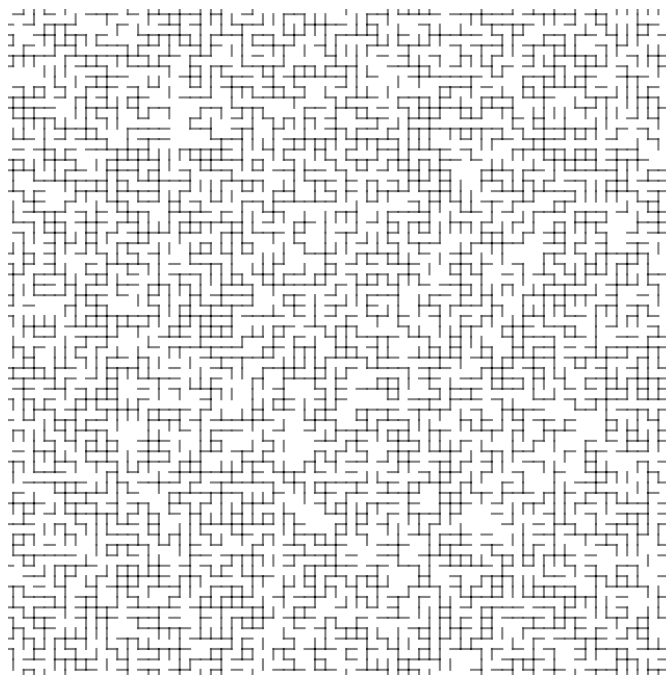
第 1 章 简介

1.1 逾渗的基本知识

在物理、化学和材料科学中，逾渗原本指流体穿过多孔材料的运动。设想将液体倒在一块多孔材料顶上，逾渗研究的是液体能否渗透过这些孔洞到达底部。Stauffer, et al.^[2] 给出了关于逾渗的详细介绍。

1.1.1 定义

逾渗理论中把多孔材料抽象为一些格子，最典型的格子是二维格子。设格子每个方向有 L 个格点，那么格子总共有 $L \times L$ 个格点 (Sites)。近邻格点之间的连线称为键 (Bonds) 或者边 (Edges)。如果把每个键看作一个通道，并假设所有键可以独立而随机的以概率 p 处于导通状态（或者以 $q = 1 - p$ 概率处于关闭状态）。



注：存在大尺度的缠绕基团，且在 x, y 两个方向都发生了逾渗。剪枝后约化为图2.1, 在这个约化后的图里面更容易看出来发生了缠绕。缠绕的定义参考1.2

图 1.1 $L = 64, p = 0.5 = p_c$ 时的键逾渗

对应每一对 (p, L) 我们能够得到液体能够渗透的概率 $P(p, L)$ 。当 L 很大的时候, $P(p, L)$ 会有一个极限

$$P(p) \stackrel{\text{def}}{=} \lim_{L \rightarrow \infty} P(p, L)$$

我们尤其对 L 很大的时候对应的逾渗概率、基团几何结构等各种性质感兴趣。当 L 足够大的时候, $P(p)$ 将会在某个点 p_c 产生一个很尖锐的突变, 这个点被称为逾渗阈值。研究键的导通概率对应的是否渗透以及相关的一些问题, 被称作键逾渗 (*Bond Percolation*)。它最早由 Broadbent, et al.^[3] 提出。随后许多数学家和物理学家对这个问题进行了大量研究。

如果每个键都认为是导通的, 但是格点 (Site) 有可能不导通, 那么我们称这种情况为座逾渗 (*Site Percolation*)。如果键与座 (格点) 都有可能不导通, 这种情况我们称之为键一座逾渗。广义的来说, 逾渗理论研究的对象是一个随机生成的图 (Graph) 的连通基团的性质与行为。

1.1.2 逾渗的普适性

普适性原理说明, 尽管不同的图的逾渗阈值 p_c 依赖于具体的局部的晶格结构, 逾渗基团在阈值附近的行为 (如分形维度等) 独立于图的局部晶格结构。而基团的行为在某种意义上比考虑 p_c 更自然。对于一个给定的维度, 在 p_c 附近, 基团的分形维度独立于晶格类型以及逾渗类型 (键座逾渗)。对相同的维数, 观测到的临界指数值对键逾渗过程和座逾渗过程是相同的。相关内容可以参考 Stauffer, et al.^[2] 的 2.5 与 2.6 节。

1.1.3 逾渗基团

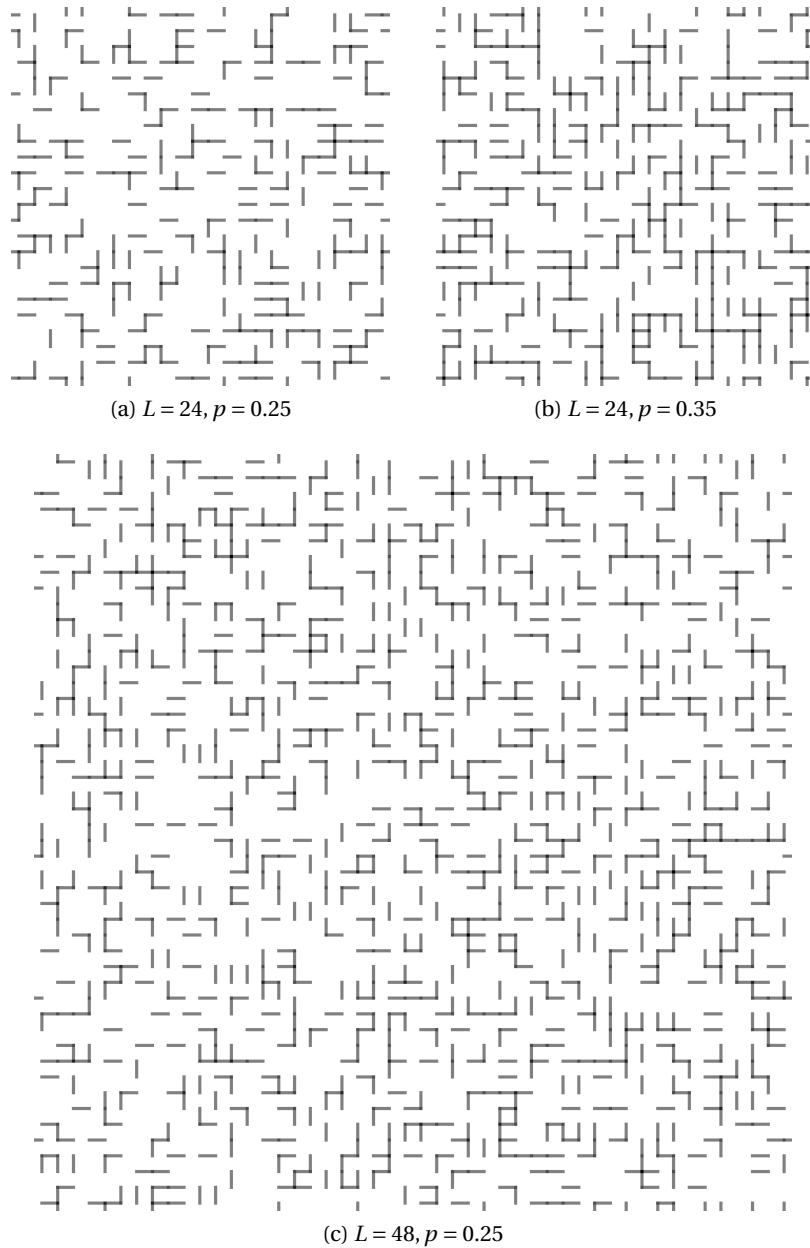
根据点与点之间的连通关系我们可以把具体的某个逾渗的格子划分为不同的基团, 使得每个基团内的点都是相互连通的, 而不同基团之间不存在任何通道。显然, 逾渗的基团的平均大小 S 与 p 和 L 有关系。

我们也可以对某个基团定义基团的特征长度

$$\xi = \langle r^2 \rangle - \langle r \rangle^2$$

设逾渗阈值为 p_c , 则 $\forall p < p_c$, 当 $L \rightarrow \infty$ 的时候, 平均尺寸 $S(p, L)$ 会有极限

$$S(p) \stackrel{\text{def}}{=} \lim_{L \rightarrow \infty} S(p, L)$$

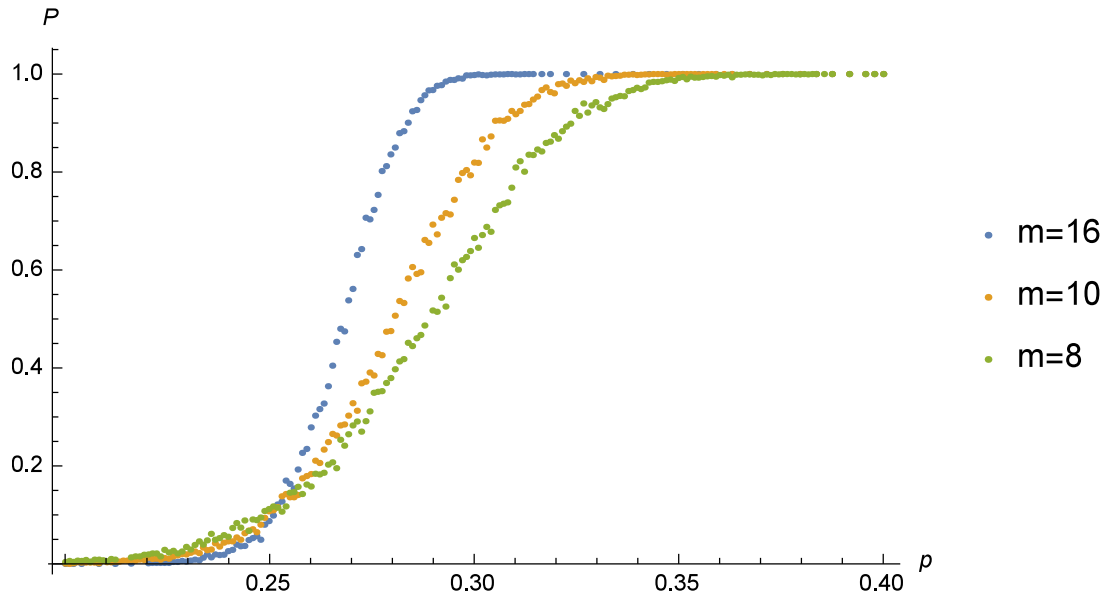


注：同样是 $L = 24$, $p = 0.25$ 的基团的特征长度 ξ 明显小于 $p = 0.35$ 。而同样 $p = 0.25$, 由于 L/ξ 比较大, 尺度为 $L = 24$ 或 $L = 48$ 对基团的特征长度 ξ 影响并不大。但是由于 $p < p_c$, 三者都不存在 $\xi \approx L$ 的大基团。这与产生了缠绕的图1.1形成对比。

图 1.2 $L = 24, p$ 不同时产生的逾渗

也就是说，当空间足够大（远远大于基团平均大小的时候），基团的平均大小是与 L 无关的。而 p 越接近于 p_c 对应的平均基团大小越大，平均大小 S 幂律增长

$$S(p) \propto 1/(p_c - p)^k, \quad p \rightarrow p_c^-$$



注：宽度 (m) 越大，展宽越窄

图 1.3 3D 情况下，不同宽度的缠绕概率与置键概率的关系图

显然 $\lim_{p \rightarrow p_c^-} S(p) = +\infty$ 。这与逾渗阈值定义相符合：如果能够逾渗就必须要求存在大尺度的基团，所以 $p \geq p_c$ 时，边长 L 越大必然导致 ξ 越大，因此 $\lim_{L \rightarrow \infty} S(L)$ 是发散的。

1.2 T^n 上逾渗的缠绕拓扑结构

缠绕 (Wrapping) 若给方格子加上周期性边界条件，方格子的拓扑结构变成了 n 维环 T^n 上的格点。如果基团中存在非平凡的环，我们称这种情况为缠绕 (*Wrapping*)，我们可以给缠绕进行分类。二维情况^[4]下，由于一条线可能将平面分割为两部分，所以我们需要单独讨论二维下的性质。以下的讨论中我们假定 L 足够大，以避免 L 的尺寸制约了绕圈的可能性。

1.2.1 二维情形

1.2.1.1 单个不自交的环的缠绕分类

设我们有某个不自交的环 C ，若它沿 x 方向缠绕了 p 圈而沿 y 方向缠绕了 q 圈，我们称之为 (p, q) 型环。由于缠绕的无向性 $(p, q) = (-p, -q)$ ，我们可以限定 $p \geq 0$ 来消除这个对称性。任何一个方向都不绕圈的平凡情况，对应于 $(0, 0)$ 型环。

为了方便分析，我们可以将 $L \times L$ 的格点空间周期性延拓。我们将坐标为 (mL, nL) 这种等价于原点的格点称为角点。我们选取曲线上任选一个点为原点 $(0, 0)$ ，那么 (p, q) 型缠绕的环等价于既不自交，也不与等价于自身的曲线相交的从 $(0, 0)$ 角点出发，中间不穿越其他角点最后到 (pL, qL) 角点的曲线段 $C(t)$ 。由于我们假定 L 足够大，所以 C 几乎可以忽略格点造成的效应，由于可以用格点的折线逼近曲线。

周期性延拓的过程中，产生一系列与 $C(x, y)$ 等价的曲线段

$$C_{mn} = (x, y) + (mL, nL), \quad \forall (m, n) \in \mathbb{Z} \times \mathbb{Z}$$

并且所有的曲线段 $C_{kp, kq}$ 都首尾相接连接起来达到无穷长。我们用 C'_{00} 代表 C_{00} 所在的无穷长的一系列首尾相接的曲线。也就是说

$$C'_{mn} \equiv C'_{m+kp, n+kq}$$

前面的不自交不仅仅要求 C'_{00} 本身不自交，也要求 C'_{00} 与任何除 $m:n = p:q$ 以外的 C'_{mn} 所在的无穷长曲线也不相交。

引理 1.1 二维情况下，若 $p:q$ 是最简整数比，则能够构造出来 (p, q) 型缠绕

此处“最简整数表示”的内涵与互质略有不同，因为包含了 $0:1, 1:0$ 这两种情况。 $p:q$ 是最简整数等价于 $(0, 0)$ 和 (p, q) 连线中，不存在任何格点。

证明 对于延拓后的格子，忽略格点的非连续性，构造连接 (pL, qL) 以及原点的线段为 C ， C 不会经过任何其他 $(p'L, q'L)$ 角点。而且 C'_{00} 这条直线平行于其他直线，因此也没有角点。 \square

引理 1.2 二维情况下，不存在非最简整数表示的 (p, q) 型缠绕

证明 不失一般性，我们选取 $(2, 0)$ 为例，用反证法来证明 $(2, 0)$ 型缠绕在二维不存在。周期性延拓后的 C'_{00} 把空间划分为两部分，且不经过 $(L, 0)$ 这个介于

原点与 $(2L, 0)$ 之间的角点。不妨假设 $(L, 0)$ 与 C'_{00} 下侧的空间相通。由于 C'_{10} 不与 C'_{00} 相交，因此 C'_{10} 中不包含任何上侧空间的点。假设曲线 C'_{00} 上的点纵坐标最大值为 y_{\max} ，则 C'_{10} 上任何点都是在下侧空间，所以纵坐标恒小于 y_{\max} 。然而平移的对称性要求 C'_{10} 对应的 $y'_{\max} = y_{\max}$ 。导出矛盾。□

定理 1.3 二维情况下，当且仅当 $p:q$ 是最简整数表示时，存在不自交的 (p, q) 型环。

1.2.1.2 逾渗基团的缠绕分类

对于一个逾渗基团来说，它可能存在多个环 $\{C_1, C_2, \dots, C_n\}$ 。考虑它的一些等价变化，如剪除树枝，把平凡环缩为一个点，可以得到一个逾渗基团缠绕性质的同伦等价关系。容易证明以下引理：

引理 1.4 二维若只存在一种环，多个同种环等价于一个环。

假设，“口”形格子内存在一个“十”字基团，使得横向纵向都贯通，我们称这种基团为“田”型基团 (Cross)。由于二维的特殊性质，我们可以证明：

引理 1.5 二维情况，如果存在两种以上的非平凡环，那么这个基团等价于“田”型基团。

根据引理1.4、1.5，立刻得出：

定理 1.6 二维情况下缠绕数 \bar{w} 一定是“田”型或 (p, q) 型。^[4]

1.2.2 高维情形

1.2.2.1 单个不自交的环的缠绕分类

高维情形下，定理1.3的前提不再成立。此时非最简缠绕的曲线可以用螺旋线的方式避免自交。因此缠绕类型的可能性约等于 T^n 的基本群^[5]。假设 T^n 中的一个环在第 k 个维度上缠绕了 p_k 圈，用向量 $\vec{p} = (p_1, p_2, \dots, p_n)$ 表示。那么代替1.3的是如下定理：

定理 1.7 在高维 ($D > 2$) 空间中，只要 \vec{p} 在格点上，就存在不自交的 \vec{p} 型环

1.2.2.2 逾渗基团的缠绕分类

在高维 ($D > 2$) 空间中, 缠绕数是一个高维的格点上的向量, 容易证明圈的环绕性的可数乘性:

引理 1.8 在高维 ($D > 2$) 空间中, 若存在 \vec{p} 型缠绕的环, 则对任意 $n \in \mathbb{Z}$ 能够构造出来任意的 $n\vec{p}$ 型缠绕的环。这里的乘法是普通的向量数乘。

可加性, 同样易证:

引理 1.9 在高维 ($D > 2$) 空间中, 若存在 \vec{p}^1 型环与 \vec{p}^2 型环, 则能够构造出来 $\vec{p}^1 + \vec{p}^2$ 型环。这里的加法是普通的向量加法。

综合上述引理 1.8 与 1.9, 我们得出最后的结论:

定理 1.10 在高维 ($D > 2$) 空间中, 若存在 $\vec{p}^1, \vec{p}^2, \dots, \vec{p}^k$ 型环, 则能够构造出来 $\sum_{i=0}^k c_i \vec{p}^i$ 型环, $\forall c_i \in \mathbb{Z}$ 。

1.3 逾渗基团的几何结构

近几十年来, 逾渗理论的一个目标是理解逾渗基团的几何结构。Herrmann, et al.^[6] 将逾渗基团分解为 Backbone 和 Dangling bonds, 再把 Backbone 更进一步分解为 Blobs 和 Red Bonds。这些分解手段加深了人们对逾渗的了解。

骨架 (Backbone)^[1,7] 通常是通过以下方法定义的: 固定准无限大格子中相隔很远的两个端点, 这两个点之间的所有路径占据的键的集合就是骨架。基团中去掉骨架后剩下的部分成为悬荡基团 (Dangling bonds)。

所有骨架中的桥组成了所谓的 Red bonds, 而骨架中剩余部分则被称为 Blobs。在临界点处, 逾渗基团的平均大小以 L^{d_F} 形式增长, 此处 L 是格子的宽度, d_F 是它的分形维度。类似的, 骨架的平均大小以 L^{d_B} 形式增长, Red-bond 数量以 L^{d_R} 形式增长。

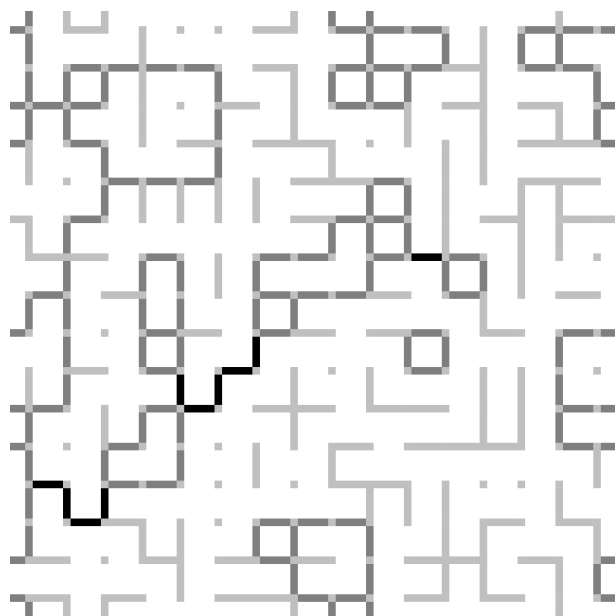
尽管二维的 d_F 和 d_R 的精确值我们已经知道了^[8,9], d_B 的精确值我们并不知道。^[1] 尽管如此, 这个值可以利用某个二阶 PDE^[10] 来计算。这个值在^[11] 中通过转移矩阵进行了估计。同时, 通过 Monte-Carol 模拟研究环上的一个恰当的关联函数, Deng, et al.^[12] 也得到了关于 d_B 的一些估计值。

我们考虑一种对逾渗生成的图的边的自然划分方法^[1], 然后再研究这种划分导致的基团的分形维度。具体来说, 我们把所有的边划分为三种类型: 枝叶、枢

纽和非桥。首先我们定义桥是删除后会影晌点之间的连通性的键。与它对应的非桥是删除后不影响点之间连通性的键。枝叶和枢纽是对桥的细分，删除后会把原来连通的一个基团变成两个基团：如果这两个基团枝叶存在一个基团是不含有环的树，那么我们称这个键为枝叶，反之则称这个键为枢纽。这里和前面的一些定义能够找到粗略的对应关系：

- 枝叶对应 **dangling**
- 枢纽对应 **red-bond**
- 非桥对应 **blobs**
- **leaf-free** 对应 **Backbone**

把逾渗产生的基团的枝叶删除后我们得到了 **leaf-free** 基团，在此基础上把枢纽删除我们就得到了 **bridge-free** 的基团，这些定义我们可以通过图1.4来说明。



注：二维边长 $L = 16, p = 0.5$ 时候随机产生的一个逾渗图及其键分类。图中键的颜色分三个亮度等级：黑色表示枢纽，深灰色表示非桥，浅灰色键代表枝叶，所有的格点的颜色也是浅灰色。

图 1.4 键的分类

1.3.1 T^2 格点

Xu, et al.^[1] 仔细研究了二维 T^2 方格子情况的键逾渗基团的几何结构。结果表明桥和非桥在系统尺寸很大的时候的比例都是系统大小的 $1/4$ 。尽管枝叶占据

的比例是所有的键中的 43%，删除枝叶对基团的分形维度并没有影响，依然保持 91/48。于此产生深刻对比的是，进一步删除比例仅为 7% 的枢纽却使得基团的分形维度下降到 1.64336(10)。

1.4 逾渗的应用

许多实际问题都可通过逾渗模型来研究，例如森林着火问题和果树疾病传播问题都可抽象成逾渗模型。以下列表给出了更多的实例。

- 多孔介质中流体的流动
- 群体中疾病的传播
- 通讯或电阻网络
- 螺旋状星系中恒星的随机生成
- 核物质中的夸克
- 表面上的液 He 薄膜
- 弥散在绝缘体中的金属原子
- 稀磁体
- 聚合物凝胶化
- 玻璃化转变
- 非晶态半导体中的变程跳跃

关于逾渗在森林火灾、油田、混合介质的导电性、聚合物凝胶化等问题中的详细应用可以参考 [Applications of Percolation Theory](#)，这些实例充分说明了逾渗问题的广泛性。

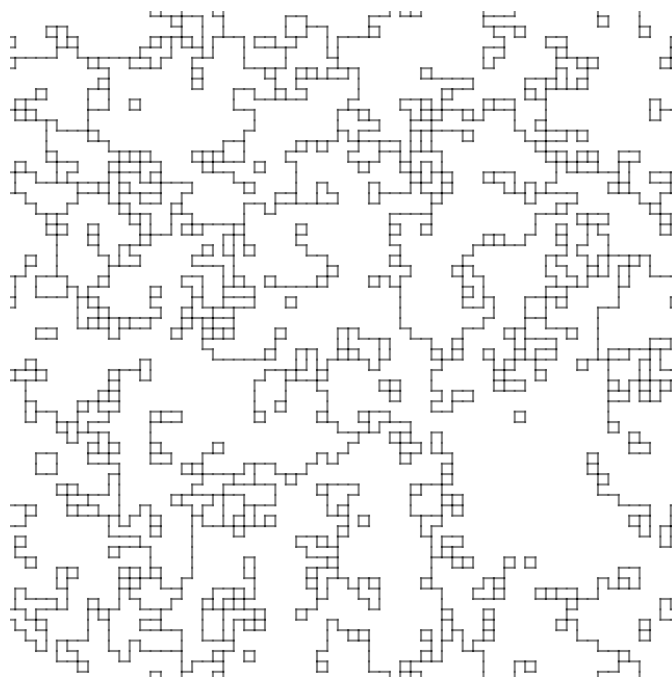
第 2 章 算法及分析

主要涉及到剪枝、标记非桥、稠化、缠绕等算法。应用这几个算法可以研究逾渗的几何结构、缠绕性质等。

2.1 剪枝

2.1.1 原理

剪枝就是把所有的枝叶全部剪除，对图1.1剪枝，得到图2.1



注：剪枝后，所有的点的度都大于等于二。剪枝前为图1.1

图 2.1 $L = 64, p = 0.5 = p_c$ 时的键逾渗

我们定义末梢是所有度为 1 的节点及其相连的键。对应每一个末梢，去除它后可能会有产生一个新的末梢，我们可以递归的剪除某个末梢及产生的新的末梢，直到遇到分岔点。在 1.3 节我们已经定义了枝叶，此处将枝叶划分为树叶与树枝：定义末梢对应的树叶是这一系列被递归剪除的末梢的集合，而树枝为枝叶去除所有树叶后的剩余部分。所以枝叶 = 树枝 + 树叶。如果把某个末梢对应的

树叶剪除，尽管可能会把某些树枝并入原来的某片树叶，容易证明不会有任何新的末梢产生。

2.1.2 算法、步骤与复杂度

如果依次遍历所有末梢并剪除对应的树枝，那么最后整个图将不存在任何末梢，因此不存在任何的树叶，所以也不存在枝叶。这个算法的伪代码实现如2.1所示。

Data: 逾渗过程产生的图
Result: 剪除并标记所有的枝叶

```

1 for  $i \in \text{all sites}$  do
2   current point  $C \leftarrow i$ ;
3   while  $C.\text{Degree}() == 1$  do
4     Set the only near point  $N = C.\text{Near}()$ ;
5     DeleteBond( $C, N$ );
6      $C \leftarrow N$ ;
7   end
8 end

```

算法 2.1: 剪枝算法

我们很容易分析这个算法的复杂度：

定理 2.1 剪枝过程时间复杂度 $O(N+L) = O(N)$ ，其中 N 为所有的点的数量， L 为所有的枝叶的数量。

证明 时间花费分为两部分：遍历所有点的消耗，这是 $O(N)$ 的；对所有末梢寻枝与剪枝的时间消耗，这一步是 $O(L)$ 的。 \square

2.2 回溯确定非桥

2.2.1 原理

BFS 中每次遍历键的时候，如果这个键指向未访问过的节点，那么会使得被遍历的点的数量增加。所有遍历时访问新点的键构成了一个遍历树，遍历树本身不存在任何环路。我们称某点在遍历树上距离根节点的层次数为遍历到该点的

时间。但是如果遍历的时候，发生了“碰撞”，也就是说键 $C \rightarrow N$ 指向的是已经访问过的节点，这意味着之前从另一条路径到达了 N 节点，且时间差为 0 或 1。这个键构成了某个环的一部分，也就是非桥。容易证明

引理 2.2 如果找到到 (N, C) 的层次数最大的共同祖先 $P = \text{Root}(N, C)$ ，那么显然从 C 或 N 回溯到根 P 所历经的所有键都是非桥。这些键加上 $C \rightarrow N$ 的键构成了本次碰撞对应的回路。

2.2.2 算法、步骤与复杂度

显然如果每次碰撞我们都通过回溯将对应的回路标记好，我们就能标记好所有的非桥。但是回溯本身花费了大量时间，我们可以对回溯进行路径压缩：每次回溯结束后把回溯中的分岔点作为中继节点直接链接到本次回溯的根节点。这样可以极大的减少同一条路径的重复回溯。具体的标记非桥的伪代码参见算法 2.2，回溯的伪代码参见算法 2.3。

Data: 剪枝过程产生的图

Result: 对剩余的键进行枢纽和非桥间的分类

```

1 set Time of initial point to zero;
2 for  $(C \rightarrow N) \in \text{BFS}$  do
3   if  $N.\text{IsVisited}()$  then
4     MarkNonBridge( $C, N$ );
5     Backtrace( $C, N$ );
6   else
7     Time[ $N$ ] = Time[ $C$ ] + 1;
8      $N.\text{SetBack}(C)$ ;
9   end
10 end

```

算法 2.2: 通过回溯来标记非桥

命题 2.3 回溯确定非桥过程时间复杂度 $O(N)$ ，其中 N 为所有的点的数量。

BFS 遍历整个图的消耗，这是 $O(N+B)$ 的。回溯与标记的过程中，由于每一次都只递减 (2.3-6) 到下一个分岔点并且在最后把记录 (2.3-5) 下来的分岔点重新

Input: Starting points (P, Q)

Data: Time,

Result:

- Bonds in the backtracing path are marked as non-bridge.
- All relays(stored by the stack R) are linked to root for (P, Q)

```

1 while  $P \neq Q$  do
2   if Time[ $P$ ] < Time[ $Q$ ] then
3     | Swap  $P \leftrightarrow Q$ ;
4   end
5    $R.append(P)$ ;
6   # Note: Decend to next diverging point;
7   do
8     | if  $P.IsLinked()$  then
9       |  $P \leftarrow P.Link()$ ;
10    | else
11    |   MarkNonbridge( $P, P.Back()$ );
12    |    $P \leftarrow P.Back()$ ;
13    | end
14  while Degree( $P$ ) == 1;
15 end
16 # Note:  $P = Q$  now is root for original  $(P, Q)$ ;
17 while  $P.IsLinked()$  do
18   |  $R.append(P)$ ;
19   |  $P \leftarrow P.Link()$ ;
20 end
21 for  $i \in R$  do
22   |  $i.SetLink(P)$ ;
23 end

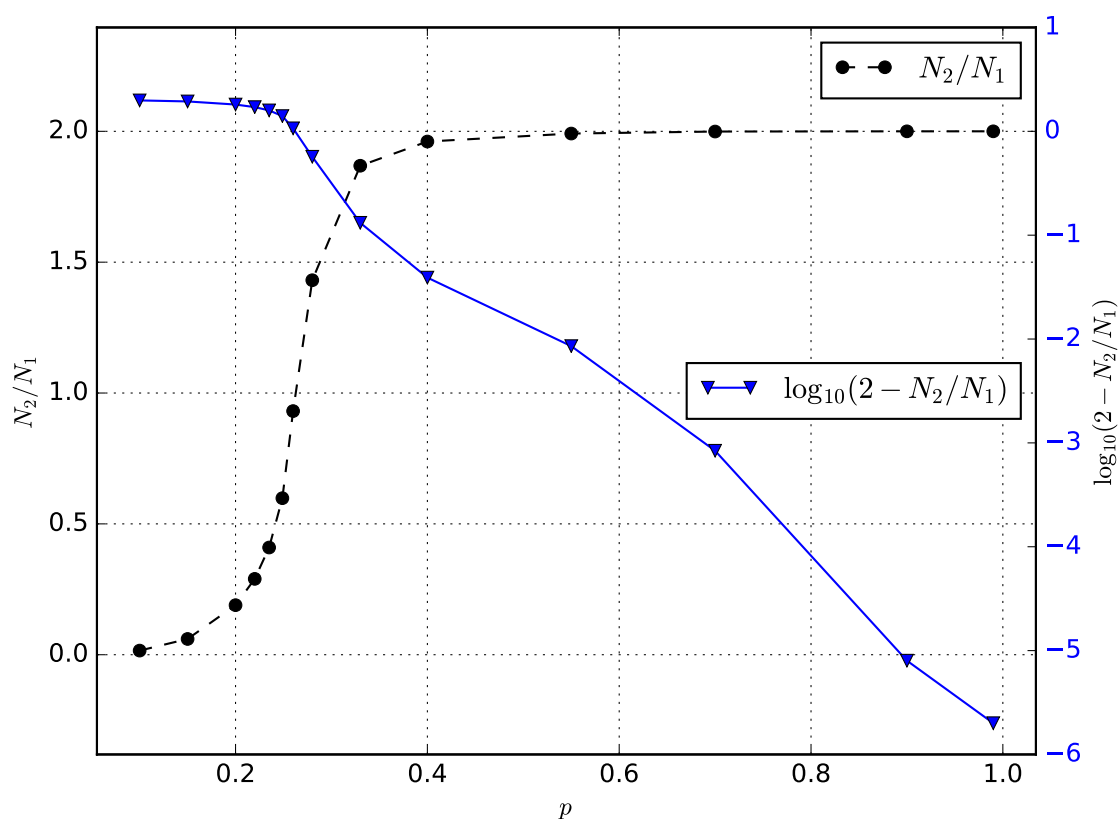
```

算法 2.3: 回溯的详细过程

链接 (2.3-22) 到此次递降的根节点。因此并不存在某个键被多次 $P.Back()$ 回溯的过程, 因此 $P.Back()$ 在整个程序中执行了 C 次。

2.2.3 链接过程的复杂度

我们设 $P.Link()$ 的执行次数为 N_1 , $Backtrace(C, N)$ 对应 N_2 。每次通过 $P.Link()$ 回溯到子树的根部的时间复杂度类似于并查集算法, 而并查集算法在使用了路径压缩与分层级两种优化后, 平均性能几乎是 $O(1)$ 的。每次回溯若涉及到基团之间的合并, 都会以新的根作为原来两个基团共同的根, 因此, 基本可以认为树是平衡的。所以 $P.Link()$ 的执行有理由认为是 $O(1)$ 的。



注: 三维边长 $L = 128$, 选取不同的概率, 每个进行模拟实验实验 20 次取平均值。结果表明平均比值一直随概率 p 严格增大, 最后收敛到 2 倍。需要强调的是, 任何单次实验中都没有观察到 N_1/N_2 大于两倍的情况, 因此我们可以大胆猜测, 这个比值严格小于两倍。2 这个比例值与维度并没有关系。对不同的维度 $D = 2, 3$ 实验均得到相同结果。

图 2.2 N_1/N_2 的比值与置键概率 p 的关系

即使不考虑平衡性，仅仅考虑路径压缩带来的效率提升，单次查找的效率也能够达到 $\ln(n)$ 。如果这个 $O(1)$ 结论成立，那么显然标记非桥的复杂度是 $O(N)$ 的。为了验证这个关于链接的复杂度的猜想，我们进行了大量实验计算 $P.Link()$ 的执行次数 N_1 与 $Backtrace(C, N)$ 执行次数 N_2 的比例，得到的结果如图2.2所示。

我们得到了一个有趣的命题：

命题 2.4 $N_1/N_2 \leq 2$

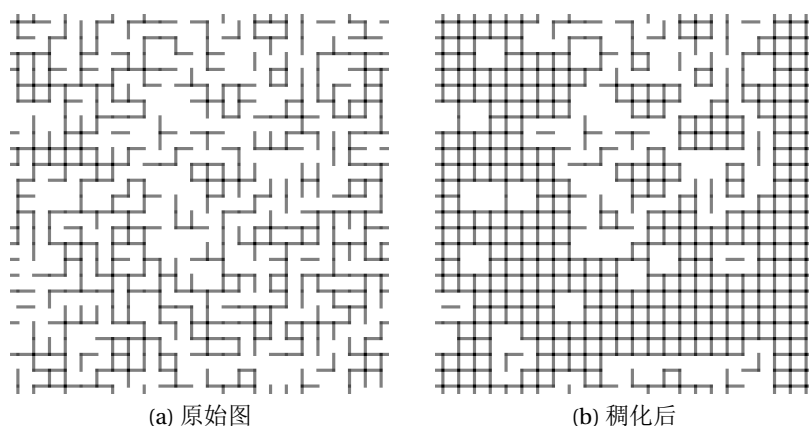
如果这个命题能够被理论上证明，那么显然整个算法的复杂度是 $O(N)$ 的。

算法中2.3-17部分的循环，在需要统计非桥丛的大小是时非常有用的。去除这一步后对 $P.Link()$ 的执行次数有微弱的影响——有时候更少有时候更多。

2.3 稠化操作

定义稠化 (*Densify*) 是把键逾渗中，所有属于同一个基团的最近邻点都加上连接的键，这个操作不改变任何基团的连通性，也不会给任何基团新增任何点。经过稠化后，添加任何一个键都会影响图的连通性。典型的稠化的例子是图2.3所示。

稠化的步骤为，先 BFS 遍历图，在此过程中给所有点染色（颜色就是所在的丛的编号）。然后再一次顺序遍历所有的点，并判定它的邻点是否和它颜色相同，如果相同就设置相应的键。稠化的伪代码参考2.4。



注：稠化后，键的密度增大，但每个基团的连通性都保持不变。添加任何一个新的键都会影响图的连通性。

图 2.3 $L = 24, p = 0.48$ 稠化前后的图像对比

Data: 逾渗过程产生的图

Result: 对连通的邻点进行连接

```

1 BFS the Graph and set Color for every site;
2 for C ∈ all sites do
3   if C.IsColored() then
4     for N ∈ C.Nears() do
5       if Color[P] == Color[Q] then
6         AddBond(C, N)
7       end
8     end
9   end
10 end

```

算法 2.4: 稠化

定理 2.5 稠化操作时间复杂度是 $O(ND)$ 的。

证明 首先 BFS 染色过程的复杂度是 $O(N + B) = O(N)$ 的。后期连接相邻同色点需要对所有被染色点的所有键进行判断，这一步的复杂度是 $O(MD)$ ，此处 D 是逾渗的维度， M 是被染色的点数。每个点可能的度的上限是 $2D$ ，由于这个图是无向图所以事实上只有 $MD = (M \times 2D)/2$ 个可能的键需要被判定。每个点被占据的概率为 $1 - (1 - p)^{2D} = 1 - q^{2D}$ ，因此 $M \approx N(1 - q^{2D})$ \square

2.4 缠绕的计算

2.4.1 原理

在 2.2 节中，我们提到每次碰撞意味着新的环路。我们可以利用遍历树对应的路径计算到达某点时，相对于起点的位移。也就是第一次访问某个节点的时候，由它的父节点相对原点的位移与它相对父节点的位移，计算出它的位移。由于缠绕的存在性，“位移场”并非是保守的，而是依赖于路径，可能相差 $\Delta \vec{z}_L, \Delta \vec{z} \in \mathbb{Z}^D$ 的。

区数与区数差 按照周期性边界条件将原来的区延拓后，我们可以用区数矢量 \vec{z} 表示某个区相对于原来的区平移了 $L\vec{z}$ 的距离。若两条路径走到了不同区 \vec{z}^1 与 \vec{z}^2

中的同一个点的镜像点，那么区数差矢量 $\Delta z = z^1 - z^2$ 。容易看出此处的 Δz 等价于路径差别 $L\Delta z$ 。同时区数差矢量等价于原先的缠绕数（参考1.2.2.1小节）。

2.4.2 算法、步骤与复杂度

由于区数差与位移差在分析缠绕时候的等价性，我们可以不记录位移而是单纯记录每个点的区数来节省大量存储。仅当访问到新节点的键穿越边界时，才有相对父节点的区数变化。通过记录区数，我们可以更直接的计算区数差得到缠绕数。

由于位移等都依赖路径，我们仅在第一次访问的时候设定位移。一旦。遍历整个图，并且记录所有发生碰撞时的区数差 Δz ，我们就能根据定理1.6、1.10对该图的缠绕进行归类。具体伪代码请参考算法2.5。

显然，除了最后一步分析缠绕之外，缠绕的复杂度与 BFS 相同，都是 $O(N)$ 的。即使在基团很大的时候，最后一步的分析一般花的计算量也并不大，几乎是只跟维度有关。

Data: 逾渗过程产生的图

Result: 对图的缠绕进行归类

```

1 set Zone of initial point to zero;
2 for  $(C \rightarrow N) \in \text{BFS}$  do
3   if  $N.\text{IsVisited}()$  then
4      $D = \text{Zone}[N] - \text{Zone}[C] - \text{DeltaZone}(C, N)$ ;
5     if  $D \neq 0$  then
6        $Z.\text{Append}(D)$ ;
7     end
8   else
9      $\text{Zone}[N] = \text{Zone}[C] + \text{DeltaZone}(C, N)$ ;
10  end
11 end
12 Classify cluster type by analysing Z

```

算法 2.5: 计算缠绕

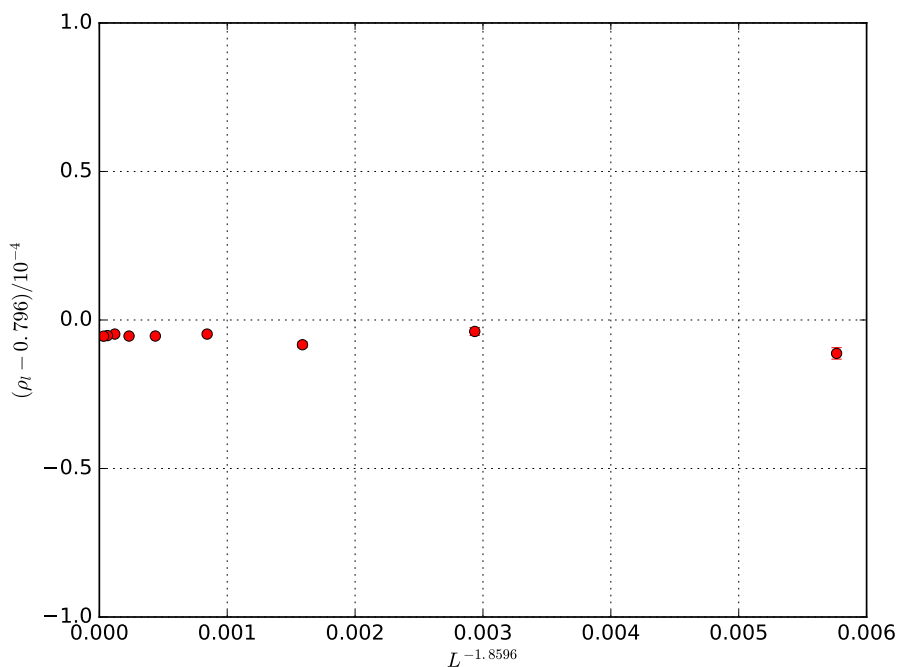
第 3 章 三维下数据结果与分析

3.1 运行相关的数据

为了研究三维情况逾渗基团的几何结构，选取三维宽度 $L = 16, 23, 32, 45, 64, 90, 128, 181, 256$ 的情况分别运行次数若干次。对于最大的 $L_{\max} = 256$ 的情况运行了 $N_{\max} = 8 \times 10^6$ 次，运行次数反比于宽度 $N = N_{\max} L_{\max} / L$ 。对于最小的 $L = 16$ 运行次数为 $N_{16} = 1.28 \times 10^8$

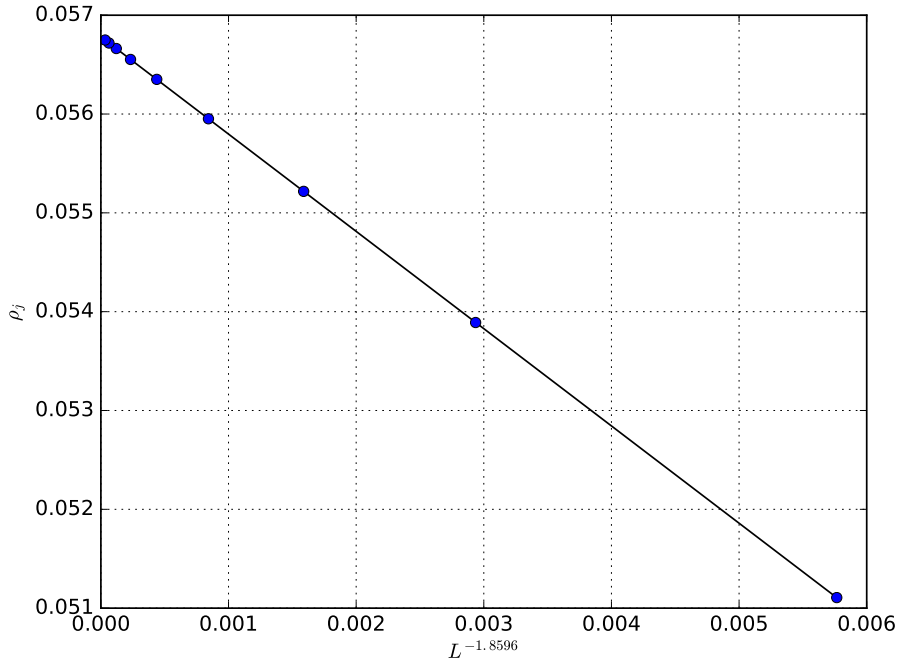
3.2 各种组分的密度

对枢纽以及非桥的比例进行拟合 $\rho = \rho_0 + aL^{-y}$ ，得到对应的两个 y 分别是 1.85891885 和 1.8603757，因此我们估计 $y \approx 1.8596(9)$ 。以 L^{-y} 为横轴，各种组分的比例为纵轴，得到了图 3.1、3.2、3.3，由于误差棒小于标记的记号的大小，三个图中误差几乎都未被显示出来。



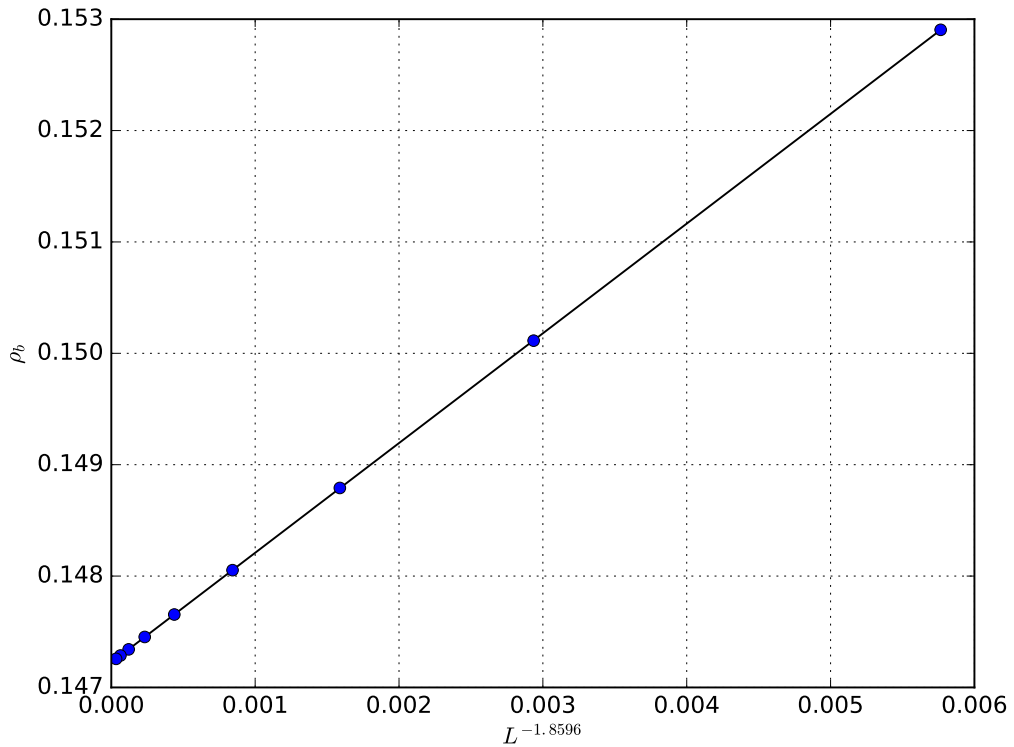
注：随着 L 增大，逐渐逼近于 0.7959952(2)

图 3.1 枝叶的比例随 $L^{-1.8596}$ 的变化关系



注：随着 L 增大，比例逐渐逼近 0.0567811(1)

图 3.2 枢纽的比例随 $L^{-1.8596}$ 的变化关系



注：随着 L 增大，比例逐渐逼近 0.1472237(2)

图 3.3 非桥的比例随 $L^{-1.8596}$ 的变化关系

3.3 最大基团的分形维度

研究了最大基团、最大去枝叶基团、最大去桥基团的分形维度。结果得出对应的分形维度分别为 2.52234679, 2.52239323, 1.8724634 (误差尚未进行计算)。在误差范围内我们可以认为, 去除枝叶后, 最大基团的分形维度没有发生变化, 都是 2.52237(3)。而进一步去除关键的非桥后, 分形维度则下降到 1.87246。去除枝叶与桥后分别对应于图 3.4、3.5。

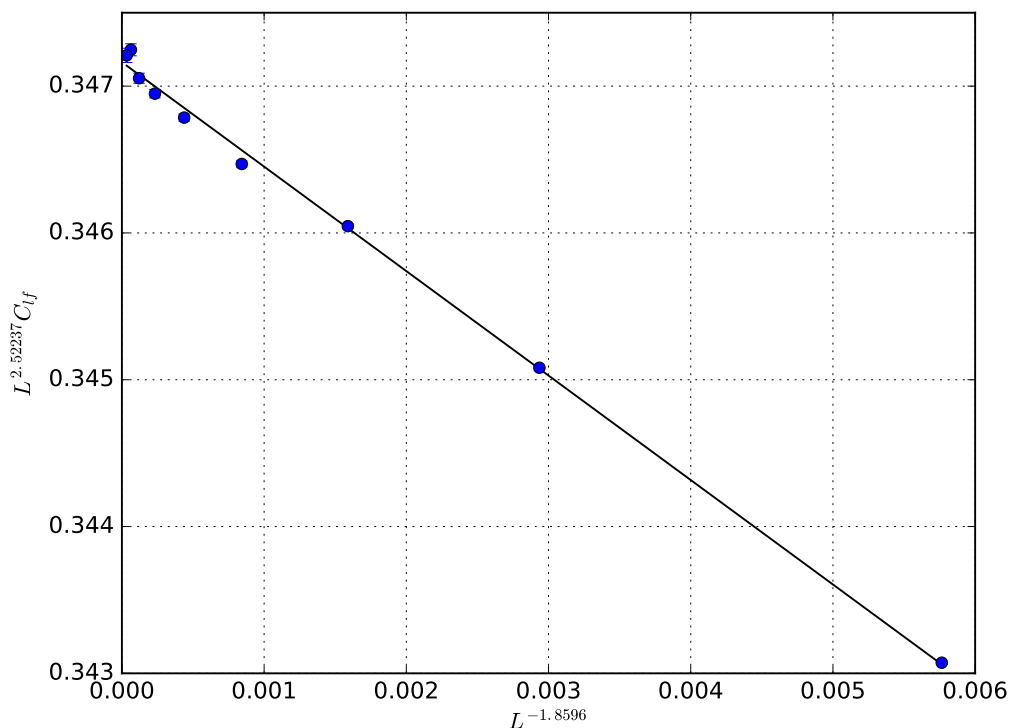


图 3.4 去除枝叶后的分形维度

3.4 最大的枝叶随尺度的变化

最大枝叶随尺度增大, 增长的越来越慢, 我们以 $y = k \ln(L + L_0) + b$ 形式的函数进行拟合, 结果发现当 $L_0 = 5.291$ 时, 以 $t = \ln(L + L_0)$ 与 y 进行线性拟合, 结果得到对应的 $r = 1 - 4.5 \times 10^{-7}$, 我们作出 y 与 t 的关系如图 3.6 所示。

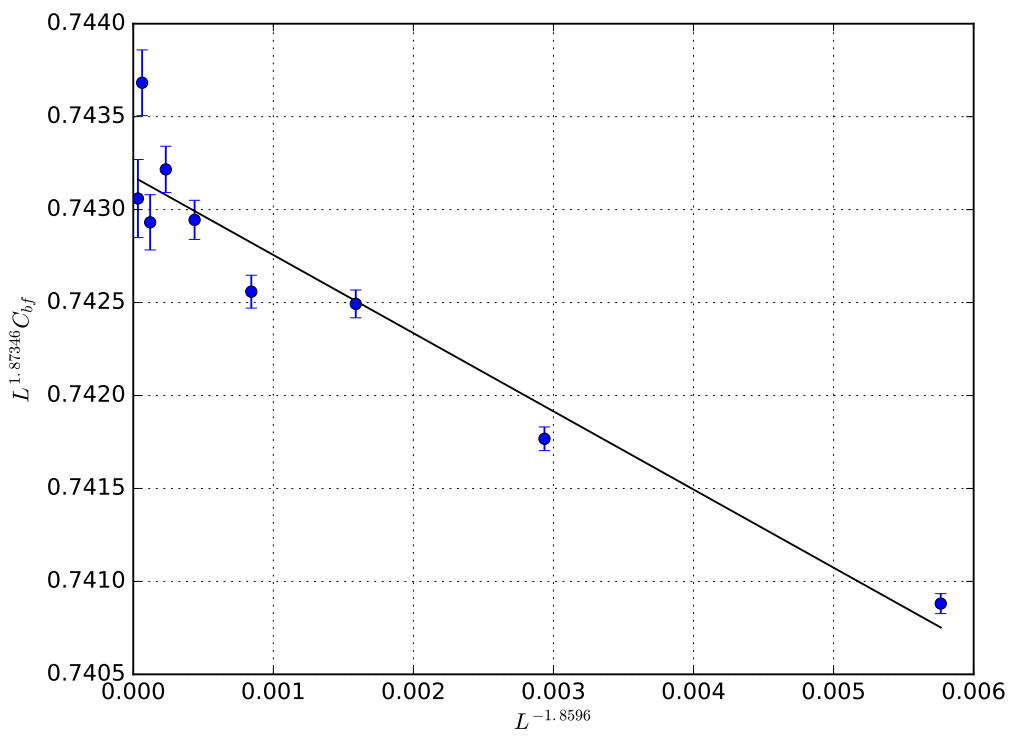
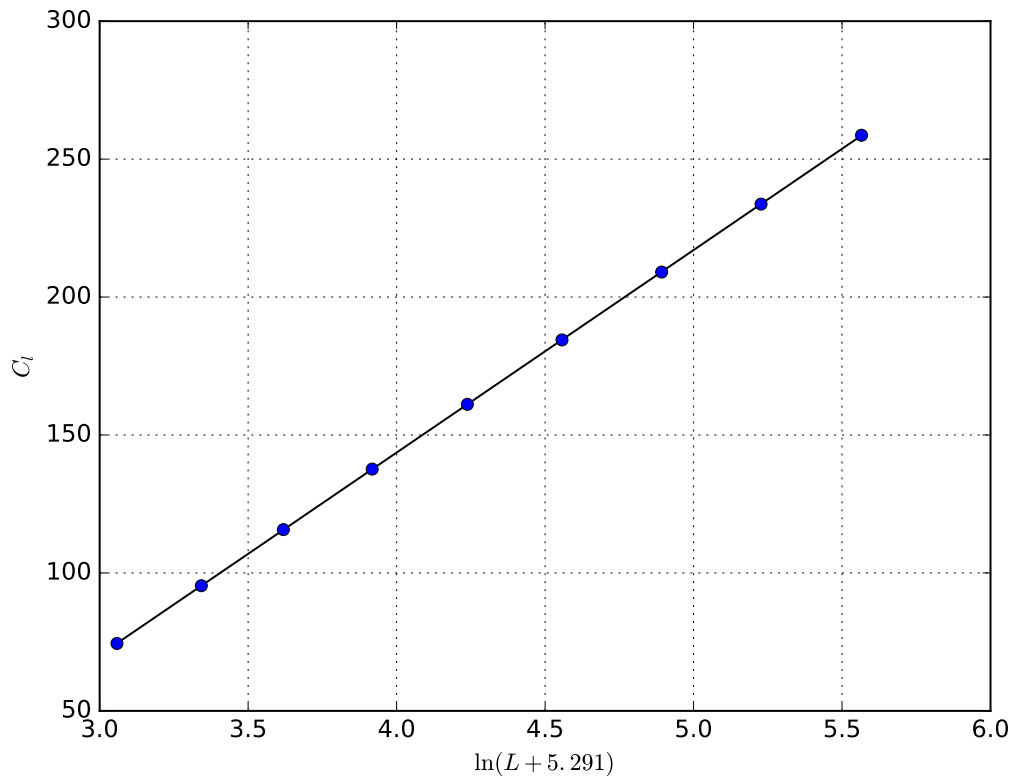


图 3.5 去除桥后的分形维度



注：随着 $\ln(L+L_0)$ 增大，最大树枝线性增加， $C_l = 73.409\ln(L+5.291) - 150.011$ ， $r = 0.999999553$ ，误差棒小于点的大小。

图 3.6 最大枝叶随 $\ln(L + L_0)$ 的变化关系

第 4 章 对未来工作的展望

以下这些内容还有待深入探究：

回溯确定非桥的复杂度证明

严格证明命题2.4，从而证明关于算法2.2的复杂性的命题2.3。

在临界概率附近基团的缠绕行为

可以研究的内容有：各种缠绕类型的比例、同一个逾渗出现多个缠绕的概率、有效分析缠绕的方法。

利用稠化操作进一步研究几何结构

稠化操作后，有可能把一些桥变成非桥，可以以此为基础研究逾渗基团的某些性质。

更高维下，剪枝与剪非桥对分形维度的影响

二维和三维下，已经验证了剪枝对分形维度影响不大而剪非桥却影响非常大。未来可以继续研究更高维情况下，剪除枝对分形维度的影响。最大枝的大小等也是可能的研究内容。此外，也许需要学习相关的逾渗理论的知识，从理论上研究这些结果出现的原因。

参考文献

- [1] Xu X, Wang J, Zhou Z, et al. Geometric structure of percolation clusters[J]. *Physical Review E*. 2014, 89(1):012120.
- [2] Stauffer D, Aharony A. Introduction to percolation theory[M]. [S.l.]: CRC press, 1994.
- [3] Broadbent S R, Hammersley J M. Percolation processes[C]// Cambridge Univ Press. Mathematical Proceedings of the Cambridge Philosophical Society: volume 53. [S.l.]: Cambridge Univ Press, 1957: 629–641.
- [4] Arguin L P. Homology of fortuin–kasteleyn clusters of potts models on the torus[J]. *Journal of statistical physics*. 2002, 109(1-2):301–310.
- [5] Nakahara M. Geometry, topology and physics[M]. [S.l.]: CRC Press, 2003.
- [6] Herrmann H J, Stanley H E. Building blocks of percolation clusters: Volatile fractals[J]. *Physical review letters*. 1984, 53(12):1121.
- [7] Herrmann H, Hong D, Stanley H. Backbone and elastic backbone of percolation clusters obtained by the new method of 'burning'[J]. *Journal of Physics A: Mathematical and General*. 1984, 17(5):L261.
- [8] Nienhuis B. Critical behavior of two-dimensional spin models and charge asymmetry in the coulomb gas[J]. *Journal of Statistical Physics*. 1984, 34(5-6):731–761.
- [9] Coniglio A. Fractal structure of ising and potts clusters: exact results[J]. *Physical review letters*. 1989, 62(26):3054.
- [10] Lawler G F, Schramm O, Werner W. One-arm exponent for critical 2D percolation[M]. [S.l.]: Université de Paris-Sud. Département de Mathématique, 2001.
- [11] Jacobsen J L, Zinn-Justin P. A transfer matrix for the backbone exponent of two-dimensional percolation[J]. *Journal of Physics A: Mathematical and General*. 2002, 35(9):2131.
- [12] Deng Y, Blöte H W, Nienhuis B. Backbone exponents of the two-dimensional q-state potts model: A monte carlo investigation[J]. *Physical Review E*. 2004, 69(2):026114.

附录 A 论文与程序源代码下载

论文的源代码请查看 [Bitbucket](#) 的 [zpj/Bachelor-thesis](#) 项目。本论文的 pdf 版本，以及涉及到的相关算法的具体实现的 C++ 代码请从这个项目 [下载](#)。

附录 B 致 谢

在毕设期间，我有幸获得了邓友金老师的关于逾渗模型的课题。在邓老师和黄伟师兄的悉心指导下，我和章彦博同学合作进行了逾渗相关的一些的研究。章彦博同学的和我的一些讨论与合作对项目进展帮助极大，极大的激发了我的一些灵感与想法，在此深表谢意。