FASTER FRACTAL PICTURES USING OPTIMAL SEQUENCES

by

Isobel McFarlane

A thesis presented to the University of Glasgow Faculty of Science
for the degree of Doctor of Philosophy.

Department of Mathematics,
October 1993.

Table of Contents

Preface

This thesis was submitted to the University of Glasgow in accordance with the requirements for the degree of Doctor of Philosophy.

I would like to express my gratitude to my supervisor, Doctor Stuart G. Hoggar for his advice, assistance and encouragement during the last three years. His comments and attention to detail were of particular value throughout.

I would like to thank the Head of Department, Professor R. W. Ogden for allowing me to study within this department.

I am grateful to my parents for their support and encouragement throughout my attendance at University. The support of the rest of my family and of my friends is also greatly appreciated.

I would like to thank the Science and Engineering Research Council for their financial support from October 1990 to September 1993.

Finally, I would like to thank Professor R. Bradley and Professor J. Gomatam for allowing me to prepare for this examination and to make the necessary corrections to this thesis while employed by them.

Abstract

Today a wide variety of images may be expressed as the attractor $\mathcal{A}$ of an iterated function system in the plane. An iterated function system or IFS is a finite collection of affine transformations w_1, ..., w_N, while the attractor $\mathcal{A}$ is the unique fixed point of the associated collage map W where $W(E) = \cup_i w_i(E)$ for any compact set E [4].

Since only 6N real numbers, known as the *code* of the IFS, are necessary to store this image, IFS's are being considered as a method of image compression [2, 5]. Moreover, algorithms which produce $\mathcal{A}$ quickly on a computer screen are being sought.

In this thesis, we study combinatorial ways of screening fractal pictures from the IFS code. We introduce the optimal sequence method and show it to be more accurate and faster than the widely used *Random Iteration Algorithm* (RIA for short) [1, 4, 16]. We also show it to be superior to the *Adaptive Cut Method* or ACM [13, 28] - one of the best non-RIA algorithms.

For uniform IFS, our investigations also lead to the expansion of the term *M-sequence* to include linear recurring sequences of period N^k - 1 over structures other than finite fields, and in particular over *far rings* which we define. We also study a new class of latin squares - *k-recurrent latin squares.* For non-uniform IFS, we initially restrict ourselves to a very simple model before extending the results obtained to more complicated models.

Chapter 1 Introduction

In this thesis, we introduce and study a new way of obtaining fractal images on the screen - the *optimal sequence method.*

In Chapter 2, we begin with a brief introduction to *iterated function systems* (IFS's). We note that provided efficient algorithms to determine the *IFS code* of images can be found, then fractal pictures (or *attractors*) may prove to be an efficient approach to image compression. We will study how the image can be re screened given the IFS code. We seek an algorithm which is fast and which does not waste time repeatedly hitting the same areas. We describe the *Collage algorithm* but show that it is not suitable for practical purposes. In Section 2.3, we consider the *Random Iteration Algorithm* (RIA) or *Chaos game.* This algorithm requires a random number generator (RNG) driver. Thus although the chaos game is usually an improvement on the collage algorithm, a poor choice of RNG can result in its failure. In order to appreciate this, we introduce an *addressing scheme.* This addressing scheme is crucial to the optimal sequence method. Finally, the *Adaptive Cut method* (ACM) is given. We are now in a position to introduce the optimal sequence method and to compare it with the above algorithms.

In Chapter 3, we consider the *optimal sequence method* for *uniform* IFS's. We define *optimal sequences* and discuss why the optimal sequence method guarantees an accurate approximation of the image. In fact, for any uniform IFS with N maps, an optimal sequence is simply a sequence of minimal length containing every k-digit word over {1, ..., N} exactly once. These sequences may be obtained using graph theory. However, this process is time consuming. If N is a prime power, then a kth-order M-sequence (with an additional initial zero) is an optimal sequence. We extend the definition of M-sequences to kth-order linear recurring sequences of period N^k-1, over any structure with N elements. In particular, we define *far rings* and show that *M-sequences* may be defined over some far rings. We also introduce *k-recurrent latin squares.* We compare the optimal sequence method (or M-sequence method) with the chaos game. We find that the M-sequence image shows much more detail and is produced in less time. In Section 3.4, we compare the M-sequence method and the ACM. The results highlight the speed and accuracy of the M-sequence method. We also note that the ACM does not always produce a true approximation of the attractor. This is investigated in Section 7.1.2.

Chapter 4 is concerned with the existence of M-sequences over structures other then finite fields. In particular, we give some results on far rings and the associated latin squares. We study the possible existence of M-sequences over two specific forms of latin squares. In Section 4.3, we give an example of a structure which is neither a far ring nor a finite field over which M-sequences exist. Finally, we list some conjectures and suggest some related problems.

We must then consider the optimal sequence method for *non-uniform* IFS. However, this is significantly more complicated than the uniform case. Consequently, we begin in Chapter 5, with a simple model non-uniform IFS. We give various results on the type and number of addresses. We show that the associated optimal sequences are cyclic. We study the structure of optimal sequences and find that certain addresses must occur in *runs* or *towers*. As a means of obtaining optimal sequences, we define the *dovetailing graph* of an IFS. By closely examining the possible paths through these graphs we develop algorithms which quickly produce an associated optimal sequence. We offer a template algorithm for this model. We compare the optimal sequence method and the chaos game for some non-uniform IFS. We note the superiority of the optimal sequence method. However, in its present form, the uses of the optimal sequence method are somewhat limited.

In Chapter 6, we extend the results of the previous chapter to a more general model so that images including a *fern* and a *tree* may now be produced. We describe how the algorithms of Chapter 5 may be adapted using M-sequences. Finally, we illustrate how sequences may be *spliced* together for more complicated images. For example, a *stem* may be added to the *fern*. The optimal sequence results maintain their high standard.

In Chapter 7, we compare in more detail the speed and accuracy of the chaos game, the ACM and the optimal sequence method. We discuss why the optimal sequence method produces better results than the ACM. Finally, we illustrate how the addresses may be used to improve the chaos game.

Chapter 8 is a brief discussion of the main results of this research and a look to the future.

Chapter 2 Iterated function systems - An introduction

In this chapter we introduce the necessary background and describe three previously established ways of producing fractal pictures. General references include [1, 3, 4, 16, 28].

2.1 Preliminaries

Notation 2.1 Let (X, d) be a complete metric space where d(x, y) denotes the distance between any points x, y of X. Further, let $\mathcal{H}(X)$ denote the collection of all non-empty compact subsets of X.

Definition 2.2 Let $x \in X$ and $B \in \mathcal{H}(X)$. Then the *distance from the point x to the set B*, denoted d(x, B), is defined as

$$d(x, B) = \min \{d(x, b) : b \in B\}.$$

Definition 2.3 For any A, $B \in \mathcal{H}(X)$, the *distance from the set A to the set B*, d(A, B) is

$$d(A, B) = \max \{d(x, B) : x \in A\}.$$

Definition 2.4 The *Hausdorff distance* between A and $B \in \mathcal{H}(X)$ is defined by

$$h(A, B) = \max \{d(A, B), d(B, A)\}.$$

In particular the Hausdorff distance, h(A, B), is zero if and only if A and B are identical.

Definition 2.5 The *diameter* of any $A \in \mathcal{H}(X)$, diam(A), is defined to be the maximum distance between any two points of A. Thus,

$$\mathrm{diam}(A) = \max \{d(x, y) : x, y \in A\}.$$

Note In the above definitions, the maximum and minimum exist because the sets A, B, X are compact [4, 16].

Definition 2.6 A transformation $f : X \rightarrow X$ is *contractive* provided there exists a non-negative number s, $0 \leq s < 1$, such that

$$d(f(x), f(y)) \leq s\, d(x, y),$$

for all points x and y in X. The smallest such constant s is called the *contractivity factor, Lipschitz constant* or simply the *ratio*, of f.

Definition 2.7 Let $f : X \to X$ be a transformation on (X, d). Then, $f^n : X \to X$, for $n = 0, 1, 2, \ldots$ is defined by $f^0(x) = x$, $f^1(x) = f(x)$, $f^{(n+1)} = f(f^n(x))$.

Theorem 2.8 (Banach's Fixed Point Theorem) Let $f : X \to X$ be a contraction mapping on the complete metric space (X, d) with ratio s where $0 \le s < 1$. Then f possesses a unique fixed point. That is, there exists exactly one point c of X such that $f(c) = c$. Moreover, for any point $x_0 \in X$, writing $x_n = f^n(x_0)$ for $n \ge 0$, the sequence $\{x_n\}$ converges to c. That is,

$$\mathrm{Lim}_{n\to\infty} f^n(x) = c$$

Finally, we have the following estimates for the distance from the fixed point c after n iterations.

$$d(x_n, c) \le \frac{s}{1-s} d(x_{n-1}, x_n) . \tag{2.1}$$

$$d(x_n, c) \le \frac{s^n}{1-s} d(x_0, x_1) \tag{2.2}$$

Proof Let $x_0 \in X$. Then for integers $m > n \ge 0$, we have

$$\begin{aligned} d(x_n, x_m) &= d(f^n(x_0), f^m(x_0)) \\ &= d(f^n(x_0), f^n(f^{(m-n)}(x_0)) \\ &\le s^n\, d(x_0, x_{m-n}) \\ &\le s^n\{d(x_0, x_1) + d(x_1, x_2) + \ldots + d(x_{m-n-1}, x_{m-n})\} \\ &\le s^n\, d(x_0, x_1)\{1 + s + s^2 + \ldots\} \\ &= s^n\, d(x_0, x_1) / (1 - s). \end{aligned}$$

Then, $d(x_n, x_m)$ can be made arbitrarily small by taking m, n sufficiently large. Hence $\{x_n\}$ is Cauchy and, since X is complete, this Cauchy sequence has a limit $c \in X$ such that $x_n \to c$ as $n \to \infty$. Since f is contractive and therefore continuous, we have

$$f(c) = f(\mathrm{Lim}_{n\to\infty} f^{(n)}(x_0)) = \mathrm{Lim}_{n\to\infty} (f^{(n+1)}(x_0)) = c,$$

showing that c is a fixed point. To show that there is no other fixed point, suppose there are two fixed points c and c'. Then $c = f(c)$ and $c' = f(c')$ so

$$d(c, c') = d(f(c), f(c')) \le s\, d(c, c'),$$

whence $(1 - s)\, d(c, c') = 0$. But since $(1 - s) \ne 0$, we have $d(c, c') = 0$, implying $c = c'$. Hence c is the unique fixed point of f.

To prove (2.1) consider,

$$d(x_n, x_m) \le d(x_n, x_{n+1}) + \ldots + d(x_{m-1}, x_m)$$

$$\leq d(x_{n-1}, x_n)\{s + s^2 + ...\}$$
$$\leq d(x_{n-1}, x_n)\, s/(1 - s).$$

Now let $m \to \infty$ so that $x_m \to c$. Then, since d is continuous in each variable, $d(x_n, x_m) \to d(x_n, c)$ while the right hand side remains unchanged, and we have proved (2.1). By noting $d(x_{n-1}, x_n) \leq s\, d(x_{n-2}, x_{n-1}) \leq ... \leq s^{n-1} d(x_0, x_1)$, we obtain (2.2) from (2.1). This completes the proof.

Definition 2.9 An *iterated function system*, or IFS for short, consists of a complete metric space (X, d) and a finite set of contractive mappings $w_i : X \to X$ with corresponding contractivity factors s_i $(1 \leq i \leq N)$. Such an IFS is often denoted $\{X; w_1,...,w_N\}$ or simply $\{X; w_{1\text{-}N}\}$.

Definition 2.10 The IFS $\{X; w_{1\text{-}N}\}$ is said to be *uniform* provided the corresponding ratios s_i $(1 \leq i \leq N)$ are all equal. Otherwise it is a *non-uniform* IFS.

Theorem 2.11 Associated with any IFS $\{X; w_{1\text{-}N}\}$, we define the *collage map*, $W : \mathcal{H}(X) \to \mathcal{H}(X)$ by

$$W(E) = \cup_i w_i(E)$$

for all $E \in \mathcal{H}(X)$. Then W is a contractive mapping with contractivity factor $s = \max\{s_1,..., s_N\}$ and W has a unique fixed point $\mathcal{A}$. That is,

$$\mathcal{A} = W(\mathcal{A}) = \cup_i w_i(\mathcal{A}).$$

In fact $\mathcal{A} = \operatorname{Lim}_{n\to\infty} W^n(\mathcal{A}_0)$ for all $\mathcal{A}_0 \in \mathcal{H}(X)$.

Note The map W is called the collage map to remind us of the fact that W(E) is formed as a union or collage of the N sets $w_1(E)$, ..., $w_N(E)$. W is also referred to as the *Hutchinson operator* after J. E. Hutchinson. He was the first to prove that W is contractive with respect to the Hausdorff distance and to apply the Fixed Point Theorem to W [17].

Definition 2.12 The fixed point $\mathcal{A} \in \mathcal{H}(X)$ of the collage map W, defined in Theorem 2.11, is called the *attractor* of the IFS $\{X; w_{1\text{-}N}\}$. The term attractor is used to suggest the movement of $\mathcal{A}_0$ towards $\mathcal{A}$ under successive applications of W. In contrast, since $\mathcal{A}$ is the unique set of $\mathcal{H}(X)$ which is unchanged by W, it is also referred to as the *invariant set* of the IFS.

Definition 2.13 An *affine transformation* $w : R^2 \to R^2$ is a composition of a linear mapping and a translation. It may be represented as

$$w(\mathbf{x}) = A\mathbf{x} + \mathbf{t}$$

where A is the (2 x 2) matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ (the linear transformation) and **t** the column vector $[e\ f]^T$ (the translation). Then an affine transformation is completely specified by the 6-tuple (a, b, c, d, e, f) known as the *code* of w.

In this work, we only consider the IFS $\{X; w_{1\text{-}N}\}$ where X is the plane R^2 or a compact subset. We will denote such an IFS simply by $\{w_{1\text{-}N}\}$ where each w_i is the affine transformation of ratio s_i and with code $(a_i, b_i, c_i, d_i, e_i, f_i)$ $(1 \leq i \leq N)$. Over the plane R^2, there are many different metrics which can be used to measure distance. Let $u = (x_1, y_1)$ and $v = (x_2, y_2)$ be arbitrary points of the plane. Then examples of metrics include the *Euclidean metric* which is defined as $d_2(u, v) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, the *maximum metric*, given by $d_\infty(u, v) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$ and the *lattice metric* $d_l(u, v) = |x_1 - x_2| + |y_1 - y_2|$. The ratio of a transformation depends on the choice of metric [28]. Consequently, a transformation may be contractive with respect to one metric and not with respect to another. For example, consider the map w with code (0.6, -0.6, 0.6, 0.6, 0, 0). Setting u = (0, 0) and v = (1, 1). It is easily checked that w is not contractive with respect to the maximum metric d_∞. Similarly, taking u = (0, 0), v = (0, 1), we find that w is not contractive over the lattice metric d_l. However, for arbitrary points u, v of the plane, $d_2(w(u), w(v)) \leq 0.6\sqrt{2}d_2(u\ v)$, showing that w is contractive with respect to the Euclidean metric. On the other hand, the map w' with code (0.85, -0.1, 0.85, 0.1, 0, 0) is contractive with respect to the maximum metric but not with respect to the Euclidean metric. We use the Euclidean metric throughout unless otherwise stated.

Today, a wide variety of images may be expressed as the attractor, $\mathcal{A}$, of an IFS of the plane [2, 5]. Further, as only 6N real numbers, known as the *code of the IFS*, are necessary to store this image, IFS's are being considered as a method of image compression. Moreover, algorithms which produce $\mathcal{A}$ on computer screen quickly are being sought. Below, we describe a few of these algorithms.

Note

(i) The affine transformation w with code (a, b, c, d, e, f) has the unique fixed point (x, y) given by

$$x = \frac{-e(d-1) + bf}{(a-1)(d-1) - bc}, \quad y = \frac{-f(a-1) + ce}{(a-1)(d-1) - bc}. \tag{2.3}$$

and has ratio s (with respect to the Euclidean metric),

$$s = \sqrt{\frac{p + \sqrt{p^2 - 4q}}{2}} \quad \text{where } p = a^2 + b^2 + c^2 + d^2 \text{ and } q = (ad - bc)^2 \qquad (2.4)$$

The ratio s is independent of e and f since translation preserves distance.

(ii) The code of the IFS $\{w_{1\text{-}N}\}$ is given in a table where the ith row is the code of the transformation w_i $(1 \le i \le N)$. This replaces the use of a large number of matrices.

Example 2.14 Table 2.1 is the IFS code for *Barnsley's Fern.* Using (2.3) and (2.4) it is easily checked that the fixed point of map w_2 is $(x_2, y_2) = (2.656, 9.959)$ and it is of ratio $s_2 = 0.851$. Further, since the ratio of each of the other three maps is smaller than s_2, we may conclude that the associated collage map W has ratio 0.851.

Attractor	map	a	b	c	d	e	f
Barnsley's Fern (Fig. 2.1, 2.2)	w_1	0	0	0	0.16	0	0
	w_2	0.85	0.04	-0.04	0.85	0	1.6
	w_3	0.2	-0.26	0.23	0.22	0	1.6
	w_4	-0.15	0.28	0.26	0.24	0	0.44

Table 2.1 The IFS code for *Barnsley's Fern.*

2.2 Deterministic IFS algorithm (Collage algorithm)

Let $\{w_{1\text{-}N}\}$ be an IFS of the plane with ratio s. Then the collage algorithm is given by

(i) Select any $A_0 \in R^2$,

(ii) Generate a sequence of collages $\{A_0, A_1, ...\}$ where

$$A_{n+1} = W(A_n),$$

or equivalently

$$A_n = W^n(A_0).$$

Then, by the Fixed Point Theorem 2.8, this process generates a (converging) sequence of sets which tends towards $\mathcal{A}$. After one iteration, A_1 is the union of N sets, after two iterations, A_2 is the union of N^2 sets, ... and so on. Further, Theorem 2.8 allows us to predict n, so that A_n is within a prescribed distance of $\mathcal{A}$. In fact,

$$d(A_n, \mathcal{A}) \le \frac{s^n}{1 - s} d(A_0, A_1) \, . \qquad (2.5)$$

Note This bound is reduced by a factor s < 1 after each iteration.

Example 2.15 *Barnsley's Fern.* We began with a small black square as our initial set A_0 and produced the converging sequence of collages $A_n = W^n(A_0)$ for Barnsley's fern. The sets A_n are given in Figure 2.1 for some n. The square was chosen since it illustrates clearly what happens to the set A_n when we apply the collage map W (for small n). Notice A_1 is the union of four sets, each characterising a different transformation w_i $(1 \leq i \leq 4)$. By A_{23}, we have reached the attractor $\mathcal{A}$ within the accuracy permitted by the screen resolution. We observed in Example 2.14 that the ratio of this IFS is 0.851. Then, by calculating $d(A_0, A_1)$ and using (2.5), the reader may verify that $d(A_{23}, \mathcal{A}) < 1$. To produce A_n from A_0, it is necessary to compute and draw $1 + 4 + 4^2 + ... + 4^n = (4^{n+1} - 1)/3$ rectangles. For example, to reach A_{23} below, we computed approximately 9.38×10^{13} rectangles. This illustrates the inefficiency of the collage algorithm.

We have observed that the collage algorithm is not fast enough for practical use. We need a more efficient algorithm, such as the one described in Section 2.3.

2.3 Random Iteration Algorithm (RIA), or chaos game

The *Random Iteration Algorithm* or RIA [1, 4, 11] produces a sequence of points $\{x_n\}$ given by

(i) Select $x_0 \in \mathcal{A}$ (a point x_0, such that $w_i(x_0) = x_0$ for some i, will do)

(ii) Plot points $x_0, x_1, ...$ where $x_{n+1} = w(x_n)$ and w is one of $w_1, ..., w_N$ which 'converge' to the attractor $\mathcal{A}$.

The *driver* of the RIA is the algorithm which produces a sequence $\sigma_1\sigma_2...\sigma_k$ on $\{1, ..., N\}$ where w_{σ_n} is the transformation performed at stage n.

Traditionally, each w_i occurs with a preassigned probability p_i $(1 \leq i \leq N)$ and a random number generator (RNG) is used to drive the RIA. We shall refer to such an RIA as the *chaos game*. However, to determine the best values for these probabilities can be difficult. Below, we describe the method popularised by Barnsley [4].

As usual, let $\mathcal{A}$ denote the attractor of the IFS $\{w_{1\text{-}N}\}$. Then the N sets $w_1(\mathcal{A})$, $w_2(\mathcal{A})$, ... , $w_N(\mathcal{A})$ form a covering of $\mathcal{A}$. That is, each point of $\mathcal{A}$ is in at least one of the sets $w_i(\mathcal{A})$, called *attractorlets*. To achieve uniform distribution, assuming no significant overlap between attractorlets, the number of points in each attractorlet should be proportional to the area of that attractorlet. Thus, since the factor by which an 'area' changes on undergoing an affine transformation is the absolute value of the determinant of the linear part of the transformation, we set

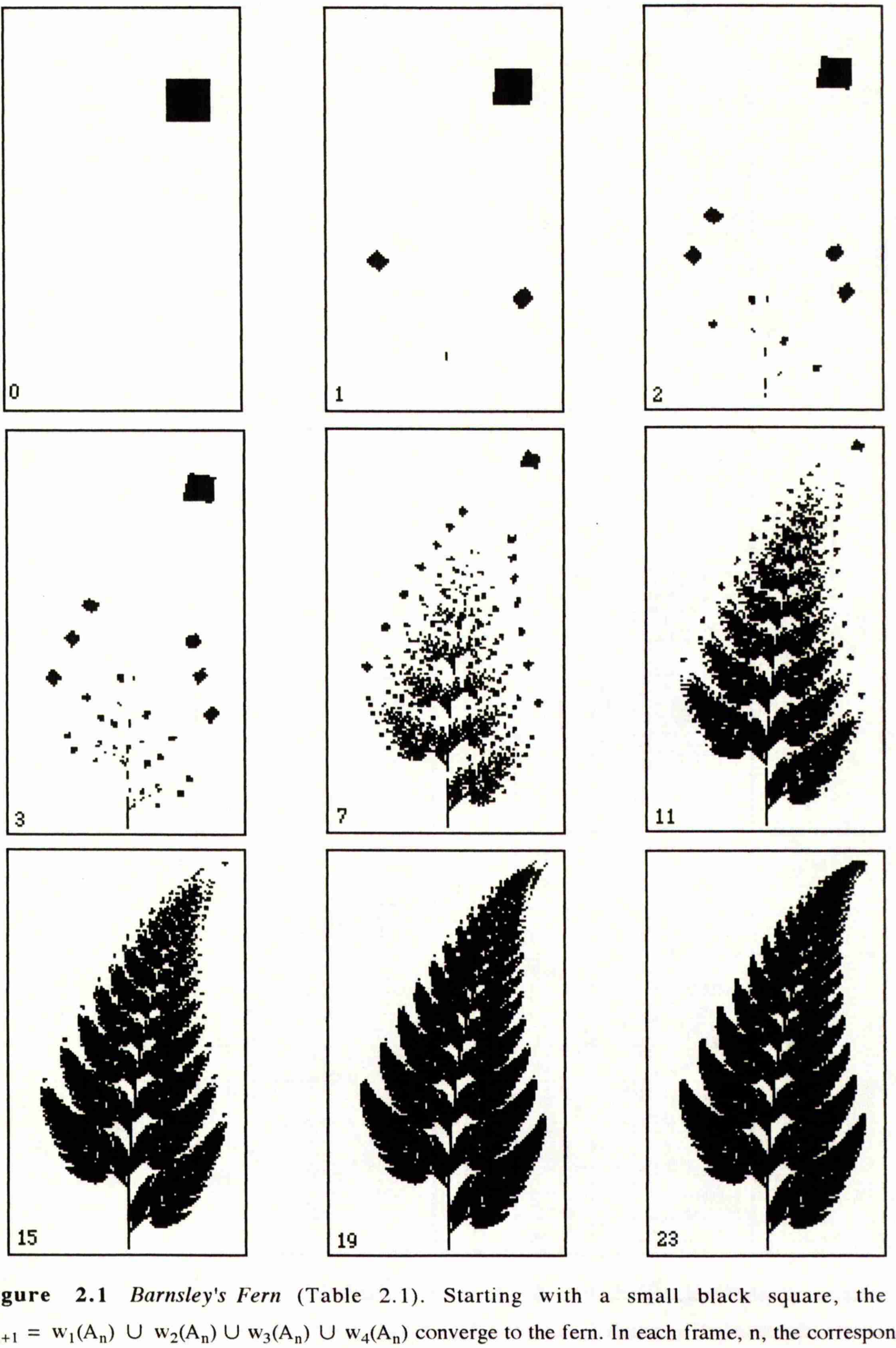


Figure 2.1 *Barnsley's Fern* (Table 2.1). Starting with a small black square, the sets $A_{n+1} = w_1(A_n) \cup w_2(A_n) \cup w_3(A_n) \cup w_4(A_n)$ converge to the fern. In each frame, n, the corresponding number of iterations is given in the bottom left hand corner. After [16].

$$p_i = \frac{|\det A_i|}{\sum |\det A_i|} \text{ where } w_i(\mathbf{x}) = A_i\mathbf{x} + \mathbf{t}_i. \tag{2.6}$$

This formula usually provides a reasonable estimate of the probabilities. However, if there are large areas of overlap then this method is less effective; or if the above formula results in $p_i = 0$ then the method fails. In the latter case, p_i would simply be set to a small positive number (cf Example 2.16). Unless otherwise stated, (2.6) will be used in our examples to determine the probabilities for the chaos game. In Section 7.2, we discuss an alternative way of determining the probabilities, which appears to be better than using (2.6).

Example 2.16 *Barnsley's Fern.* Since det $A_1 = 0$ for the map w_1 of Table 2.1, formula (2.6) would result in $p_1 = 0$ and the stem would never appear. Instead, we set $p_1 = 0.01$. Barnsley's fern was produced using the chaos game. The images produced at various stages are given in Figure 2.2. Even after only 500 iterations of the algorithm, a rough outline of the fern is visible.

In this thesis, we introduce a new class of drivers for the RIA - *optimal sequences.* In Chapters 3 and 5, optimal sequences are defined formally and we seek efficient ways of obtaining them. We illustrate that these optimal sequences drive the RIA much more efficiently than any RNG. Below, we introduce the concept of *addresses* crucial to optimal sequences and explain why the RIA works.

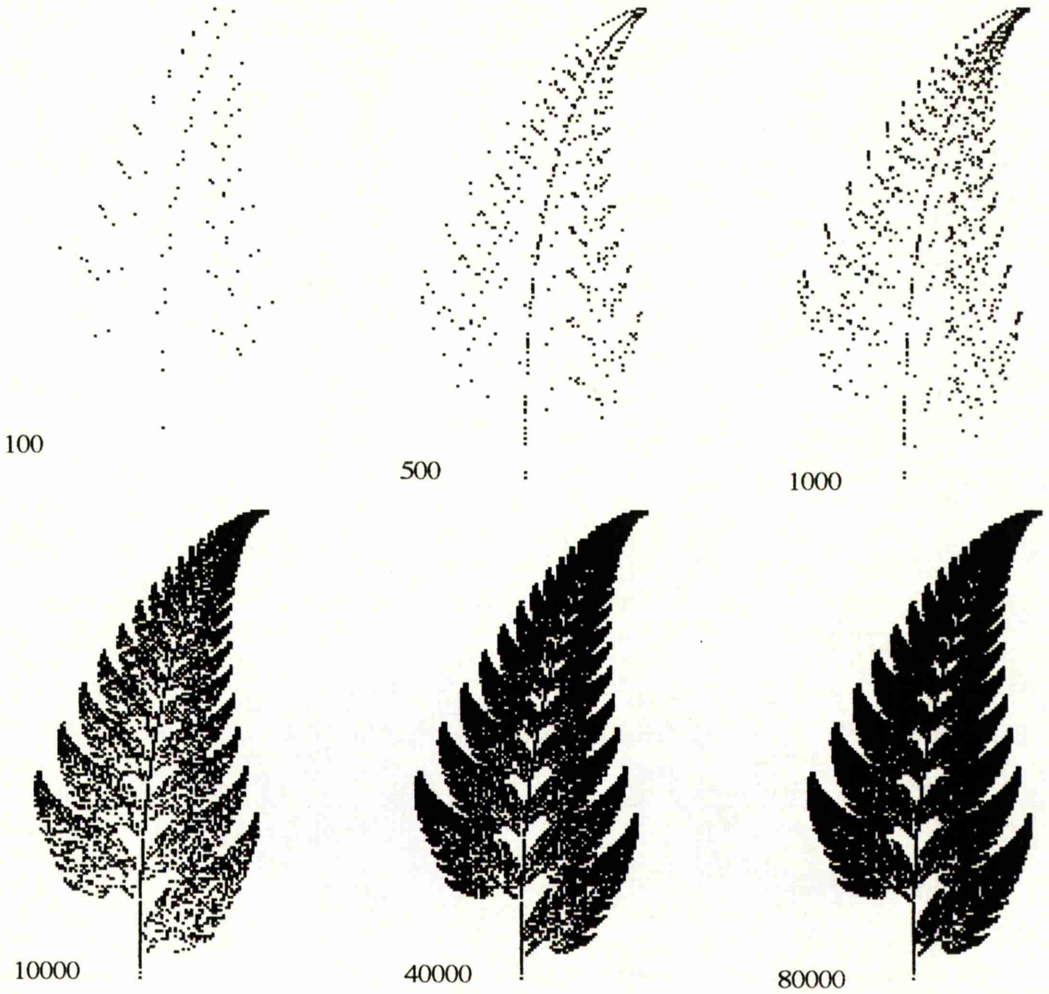


Figure 2.2 *Barnsley's Fern* after various stages of the chaos game. The number of points plotted for each image is given in the bottom left hand corner. After [16].

2.3.1 An addressing scheme

In section 2.3.2, we show that the RIA, driven by (i) a RNG and (ii) an optimal sequence, will in time produce an accurate approximation of the attractor $\mathcal{A}$. In order to do this, we must first introduce an addressing scheme for the IFS $\{w_{1\text{-}N}\}$ [4, 17, 16]. Before introducing a scheme for general N, we will consider a simple example.

Example 2.17 The *Sierpinski gasket* (for IFS code see Table 3.13 [24]) is the attractor of the IFS $\{\Delta;\ w_{1\text{-}3}\}$ where Δ denotes a solid equilateral triangle with vertices labelled 1, 2, 3, a centre of mass 0 and where w_i is the dilation with ratio 1/2 and centre i $(1 \leq i \leq 3)$. We seek some way to describe the point **x** of Figure 2.3.

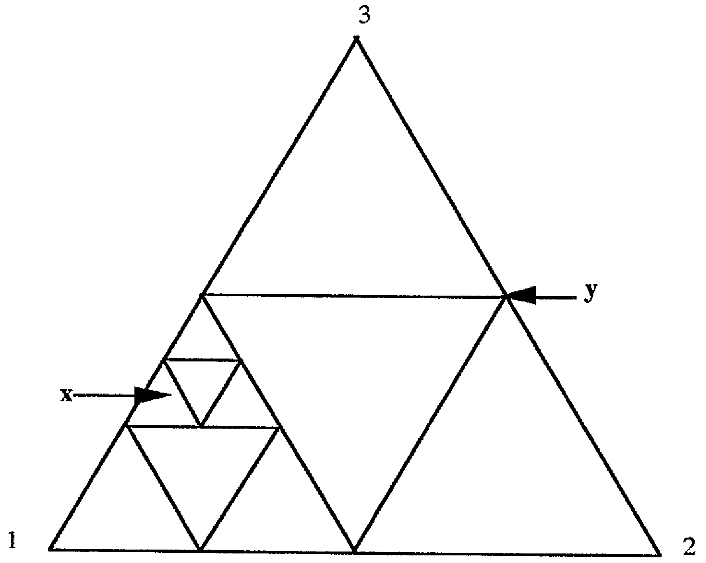


Figure 2.3 The addressing scheme for points of the *Sierpinski gasket.*

Initially, we subdivide Δ into four equilateral triangles. Each time a triangle is subdivided in this way, we will use 1 to identify the lower left triangle, 2 the lower right and 3 the uppermost. Further the middle triangle will be removed. Then the point **x** lies in Δ_1, the first level subtriangle with label 1. This triangle is then subdivided into smaller triangles and we note that **x** is in subtriangle 3 of subtriangle 1. We call this second level triangle Δ_{13}. Repeating this subdivision, we see **x** lies in the third level subtriangle Δ_{131} and so the address of **x** begins 131. To determine the address of **x** to a greater accuracy, we continue to subdivide and identify subtriangles. Some points may be identified by more than one address. For example the point **y** of Figure 2.3 has address 322... or 233... .

This address scheme is closely related to the application of the transformations w_i $(1 \leq i \leq 3)$. Evidently, $\Delta_1 = w_1(\Delta)$, $\Delta_2 = w_2(\Delta)$ and $\Delta_3 = w_3(\Delta)$. In fact, $\Delta_{\alpha\beta\chi} = w_\alpha w_\beta w_\chi(\Delta)$ where the transformations are applied from right to left (α, β, $\chi \in \{1, 2, 3\}$).

Notation 2.18 To extend the notation of Example 2.17 to the more general IFS $\{X;\ w_{1\text{-}N}\}$ with attractor $\mathcal{A}$, we express the subset with address beginning $\sigma = \sigma_1\sigma_2 \ldots \sigma_k$ (k finite) as

$$\mathcal{A}_\sigma = \mathcal{A}_{\sigma_1 \ldots \sigma_k} = w_{\sigma_1} \ldots w_{\sigma_k}(\mathcal{A}).$$

Then, applying the subdivision process of Example 2.17 to $\mathcal{A}$, we have $\mathcal{A} = \cup \mathcal{A}_{\sigma_i}$, $1 \leq \sigma_i \leq N$, $\mathcal{A} = \cup \mathcal{A}_{\sigma_i \sigma_j}$, $1 \leq \sigma_i$, $\sigma_j \leq N$, $\mathcal{A} = \cup \mathcal{A}_{\sigma_i \sigma_j \sigma_k}$, $1 \leq \sigma_i$, σ_j, $\sigma_k \leq N$ and so forth. Further any point $\mathbf{x} \in \mathcal{A}$ has an address beginning some sequence of length k, say

$\sigma_1\sigma_2\ldots\sigma_k$, where **x** is in the kth level partition $\mathcal{A}_{\sigma_1\ldots\sigma_k}$ and $\mathcal{A} = \cup\mathcal{A}_{\sigma_1\ldots\sigma_k}$, $\sigma_i \in \{1,\ldots,N\}$.

Theorem 2.19 We shall denote the ratio of the map w_σ by $\rho(\sigma)$. Then we define $\rho(\sigma_i\sigma_j)$,the ratio of the composite map $w_{\sigma_i}w_{\sigma_j}$ to equal the product of the ratio of w_{σ_i} and w_{σ_j}.
i.e.

$$\rho(\sigma_i\sigma_j) = \rho(\sigma_i)\rho(\sigma_j).$$

Remark 2.20 The above definition is valid since the ratio of the composite map $w_{\sigma_i}w_{\sigma_j}$ (over the Euclidean metric) is no more than the product of the ratio of w_{σ_i} and w_{σ_j}.
Although in some cases Definition 2.19 may significantly over estimate the ratio of composite transformations, it is quick to compute.

Definition 2.21 Let $\mathcal{A}$ be the attractor of the IFS $\{w_{1\text{-}N}\}$ of ratio s and let $\varepsilon > 0$ be given. Then, the corresponding list of *addresses* consists of all sequences $\sigma = \sigma_1\sigma_2\ldots\sigma_k$ over $\{1, \ldots, N\}$ where $\rho(\sigma_1\sigma_2\ldots\sigma_k) \le \varepsilon$ and $\rho(\sigma_1\sigma_2\ldots\sigma_{k-1}) > \varepsilon$. So the addresses divide the attractor into the subsets $\mathcal{A}_\sigma = w_{\sigma_1}\ldots w_{\sigma_k}(\mathcal{A})$, each of diameter not exceeding ε.

We may represent this construction by an *address tree*. Starting with a node labelled 1, we recursively branch at the node σ (representing the partition $\mathcal{A}_\sigma$) whenever, $\rho(\sigma) > \varepsilon$. We label such branches σ_1, ..., σ_N and their respective new nodes $\sigma\sigma_1$, ..., $\sigma\sigma_N$. When the tree is complete, each end node τ satisfies $\rho(\tau) \le \varepsilon$ and represents a unique address.

Note For a given $\varepsilon > 0$, a corresponding optimal sequence contains every address.

Example 2.22 The *Sierpinski gasket* of unit diameter. Let $\varepsilon = 1/8$. Then by Definition 2.21, the complete list of addresses is $\{\sigma_1\sigma_2\sigma_3 : \sigma_i \in \{1, 2, 3\}\ (1 \le i \le 3)\}$. The subsets associated with these addresses partition the attractor as shown in Figure 2.4.

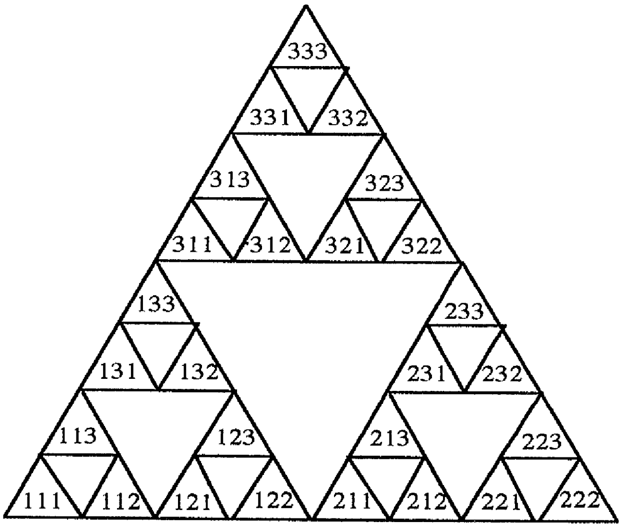


Figure 2.4 The subsets $\Delta_\sigma = w_{\sigma_1} w_{\sigma_2} w_{\sigma_3}(\Delta)$ where $\sigma_1\sigma_2\sigma_3$ is an address.

Example 2.23 Let $\mathcal{A}$ denote the Barnsley's fern attractor and recall $\mathcal{A} = w_1(\mathcal{A}) \cup w_2(\mathcal{A}) \cup w_3(\mathcal{A}) \cup w_4(\mathcal{A})$ (see Table 2.1 for IFS code). In fact $w_3(\mathcal{A})$, $w_4(\mathcal{A})$ are the lower left and right leaves respectively and $w_2(\mathcal{A})$ is the remainder of the fern apart from a small piece of the stem which is $w_1(\mathcal{A})$ (see Figure 2.5 where $w_1(\mathcal{A})$ is blue, $w_2(\mathcal{A})$ red, $w_3(\mathcal{A})$ yellow and $w_4(\mathcal{A})$ green). The address of each point in the lowest left leaf begins with 3. Similarly, every point in the lowest right leaf has address beginning with 4. Moreover, all point in the lowest left leaf of the lowest left leaf have addresses beginning 33 and so on.

The *tree* IFS of Figure 2.6 consists of only 5 affine transformations (see Table 7.3 for IFS code).

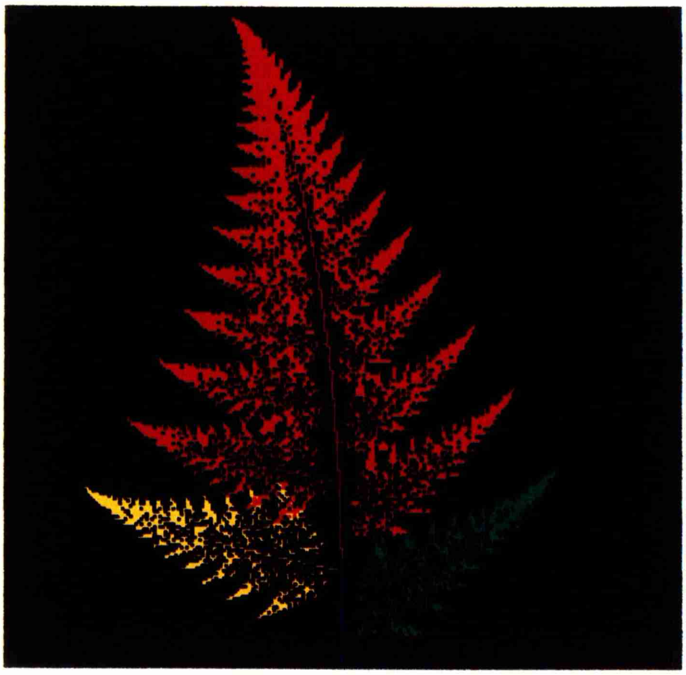


Figure 2.5 To illustrate how *Barnsley's fern* is formed, we have chosen a different colour for each area $w_i(A)$ $(1 \leq i \leq 4)$.

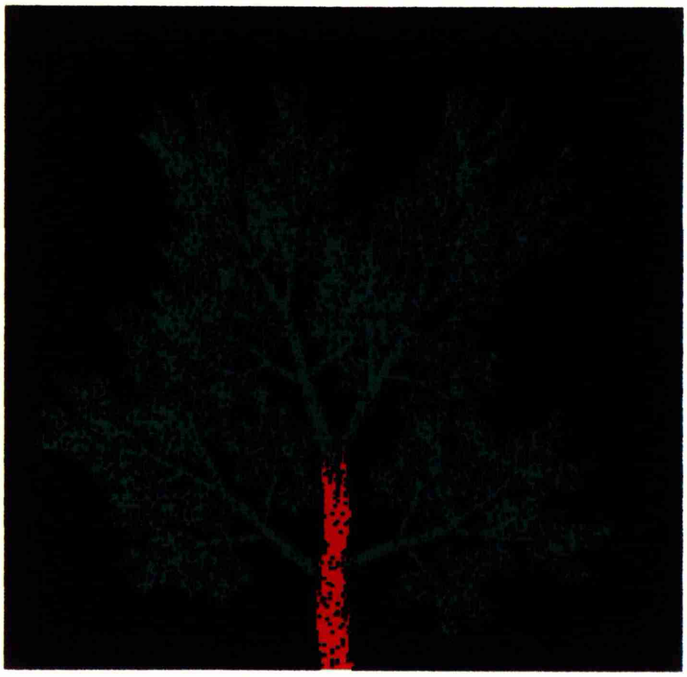


Figure 2.6 The *tree* IFS has only five maps.

2.3.2 Why the RIA works -

2.3.2.1 Using a RNG driver - The chaos game

Consider the IFS $\{w_{1\text{-}N}\}$ with corresponding probabilities p_i , ratios s_i and attractor $\mathcal{A}$ $(1 \le i \le N)$. Recall, the RNG driven RIA (i.e. the chaos game) produces a sequence of points $\{x_n\}$ where $x_{n+1} = w(x_n)$, w is randomly chosen from $\{w_1, \dots, w_N\}$ and $x_0 \in \mathcal{A}$. We wish to show that $\{x_n\}$ produces a reasonable approximation of $\mathcal{A}$. Clearly, due to the invariance property of the attractor $x_n \in \mathcal{A}$. Then it remains to show that $\{x_n\}$ fills out the attractor densely. Let $\alpha \in \mathcal{A}$ with address beginning $\tau_1\tau_2...$; $\tau_i \in \{1,...,N\}$ and let $\varepsilon > 0$. We will show that by playing the chaos game sufficiently long we will obtain a point $x_n \in \mathcal{A}$ within a distance ε from α. Recall,

$$\mathcal{A} = \cup w_{\sigma_1}\dots w_{\sigma_k}(\mathcal{A}) \text{ where } \sigma_i \in \{1, \dots, N\} \text{ for } k = 1, 2, \dots$$

We have $\alpha \in w_{\tau_1}\dots w_{\tau_k}(\mathcal{A})$ for $k = 1, 2,...$ where,

$$\operatorname{diam}(w_{\tau_1}\dots w_{\tau_k}(\mathcal{A})) \le s_{\tau_1}\dots s_{\tau_k}\operatorname{diam}(\mathcal{A}).$$

Thus by choosing k large enough, the diameter of $w_{\tau_1}\dots w_{\tau_k}(\mathcal{A})$ will be less than ε. Also any point of the attractor with address beginning $\tau_1\dots\tau_k$ will be in the kth-level partition $w_{\tau_1}\dots w_{\tau_k}(\mathcal{A})$ and so will have distance less than ε from the point α. Suppose the driver of the RIA produces the sequence $\sigma = \sigma_1\sigma_2...$. Then if $\tau_k\dots\tau_1$ appears as a subblock in σ, a point x_n will be plotted in $w_{\tau_1}\dots w_{\tau_k}(\mathcal{A})$. In other words we require that $\sigma = \sigma_1\dots\sigma_p\tau_k\dots\tau_1\sigma_{p+k+1}...$ for some positive integer p. The probability that any k-digit subsequence $\sigma_i\dots\sigma_{i+k-1}$ of σ equals $\tau_k\dots\tau_1$ is $p_{\tau_1}\dots p_{\tau_k}$; a positive number. Hence by playing the chaos game sufficiently long, we will obtain a point x_n of the attractor within a distance ε of α. We may conclude that we can indeed get arbitrarily close to any point of the attractor and so the chaos game fills out the attractor densely.

Note Above, we assume that the RNG is perfect. That is, that each symbol σ_i occurs with probability p_i and that every k-digit sequence $\sigma_1\dots\sigma_k$ occurs with positive probability $p_1\dots p_k$. But, in practice, it is essentially impossible to find a RNG which produces all or even most k-digit sequences, however small the value of k [20]. Thus, the chaos game often produces patchy images (see Section 2.3.2.1.1 below).

2.3.2.1.1 Addresses and RNG's

Recall, the chaos game requires a random sequence $\{\sigma_n\}$ over $\{1, ..., N\}$ such that i occurs with probability p_i and each symbol is independent of the previous. Below, we illustrate how one particular type of random number generator which was first introduced by D. H. Lehmer in 1949 [22], may be used to produce $\{\sigma_n\}$.

A *linear congruential sequence* $\{z_n\} = \{z_0, z_1, ...\}$ is given by

$$z_{n+1} = (az_n + c) \bmod m \tag{2.7}$$

with the *modulus* $m > 0$, the *multiplier* a, $0 \leq a < m$, the *increment* c, $0 \leq c < m$ and the *initial value* z_0, $0 \leq z_0 < m$. All linear congruential sequences are *periodic*. Further, since z_n may take at most m different values, the maximum period is m. Both the period and the 'randomness' of the sequence $\{z_n\}$ depend on the parameters m, a, c and z_0. For example, with $m = 10$, $a = c = z_0 = 9$, the sequence $z_n = \{9, 0, 9, 0,...\}$ defined by (2.7) has period two and is clearly not random. We seek values for these parameters so that $\{z_n\}$ is (i) of long period, (ii) random and (iii) fast to compute. Notice with $a = 0$, (2.7) becomes $z_{n+1} = c \mod m$, while $a = 1$ gives $z_{n+1} = (z_n + c) \bmod m$, neither of which will be random. Thus in practice, we are restricted to $2 \leq a < m$.

A long period is essential for any sequence that is to be used as a random number generator and, since the period of $\{z_n\}$ cannot exceed m, the modulus must be large. Notice that a long period does not ensure that $\{z_n\}$ is random. For example, $z_{n+1} = (z_n + 1) \bmod m$ has period m but is certainly not random. It is also desirable that $(az_n + c) \bmod m$ be fast to compute. For this reason, when $\{z_n\}$ is being produced by a t-bit binary computer, m is often set to the word size, 2^t (for some positive integer t). Then addition is automatically given mod m while multiplication mod m is simply the lower half of the product. However, there is a drawback. It is easily checked, if d is a divisor of m and $z_n' = z_n \bmod d$, then $z_{n+1}' = (az_n' + c) \bmod d$. Thus if $m = 2^t$, then the low-order k-bits are $z_n' = z_n \bmod 2^k$ so that $\{z_n'\}$ forms a congruential sequence of period 2^k or less. For some applications, the low-order bits are insignificant and so $w = 2^t$ is suitable. Otherwise, this situation may be prevented by taking $m = 2^t \pm 1$ or $m = p$ where p is the largest prime number less than 2^t. The Macintosh RNG, RandomX is $z_{n+1} = 7^5 z_n \bmod (2^{31} - 1)$ where 2^{31} is the word size of this computer and $2^{31} - 1$ is in fact prime.

For a given m, the next question to be answered is :- what values of a, c, z_0 (if any) result in period m? If m is the product of a number of distinct primes then only $a = 1$ can result in a full period, while if a higher power of some prime is a divisor of m, there are various possibilities. Knuth studies this in detail [20].

So, the sequence $\{z_n\}$ given by (2.7) generates a sequence of numbers in the range $\{1, ..., m\}$. The length and the randomness of this sequence depends on the choice of parameters a, c, x_0. From $\{z_n\}$, we may obtain a sequence $\{y_n\}$ of random real numbers between zero and one, by setting $y_n = z_n/m$. The random distribution of $\{y_n\}$ depends on the randomness of $\{z_n\}$.

From $\{y_n\}$, we get random integers σ_n between 0 and N - 1 by letting $\sigma_n = \lfloor N\ y_n \rfloor$. This gives each integer with equal probability 1/N. Alternatively, we may obtain the sequence $\{\sigma_n\}$ where the integers 1, ..., N occur with probabilities $p_1, \ldots, p_N$ respectively, as follows:-

$$\sigma_n = \begin{cases} 1 & \text{if} \quad 0 \le y_n < p_1 \\ 2 & \text{if} \quad p_1 \le y_n < p_1 + p_2 \\ \cdot & \\ \cdot & \\ \cdot & \\ N & \text{if} \quad p_1 + \ldots + p_{N-1} \le y_n < 1 \end{cases}$$

However, $\{\sigma_n\}$ will only contain every k-tuple over $\{1, \ldots, N\}$ provided $\{z_n\}$ was truly random, as the example below illustrates.

Example 2.24 The *square* (see Table 2.2 for IFS code) was produced using the chaos game driven by the linear congruential sequences (i) $z_{n+1} = (2^{28} + 3)z_n \bmod 2^{29}$ and (ii) $z_{n+1} = 7^5 z_n \bmod (2^{31} - 1)$ - the Macintosh RandomX. The results are given in Figure 2.7. The approximation produced by driver (i) is very strange. Although the points plotted all lie in the attractor, large parts appear to be missing. Further, even iterating the chaos game for 10,000 points or more does not fill in the missing areas. Initially, we might check that each map w_i occurs with the probability 0.25. For 10,000 iterations, we recorded the number of times each map is used and this confirmed that each map is performed with probability 0.25. So where does the problem lie? Looking again at Figure 2.7, we note that the chaos game with driver (i) is unable to produce many addresses. The following 3-digit words can never occur within the address of any point plotted using (i):-

223, 232, 233, 322, 323, 332, 230, 231, 320, 321, 201, 210, 301, 310, 023, 032, 123, 132, 012, 013, 102, 103, 001, 010, 011, 100, 101, 110.

We have $z_{n+1} = (2^{28} + 3)z_n \bmod 2^{29}$ with $z_0 = 1$. Then the corresponding driver σ_n where $\sigma_n = \lfloor 4z_n/2^{29} \rfloor$ produces random integers between 0 and 3 inclusive. For any z_n,

$$z_{n+1} = (2^{28} + 3)z_n \bmod 2^{29} \text{ and } z_{n+2} = (2^{28} + 3)z_{n+1} \bmod 2^{29} = 9z_n$$

so that

$$z_{n+2} = 3z_{n+1} - 2^{28}z_n \bmod 2^{29}.$$

Then, given σ_n, σ_{n+1},

$$\sigma_{n+2} = 3(\sigma_{n+1} + \varepsilon_1) - 2^{28}(\sigma_n + \varepsilon_2) \bmod 4$$

where $0 \leq \varepsilon_1, \varepsilon_2 < 1$. Thus the triples $(\sigma_n, \sigma_{n+1}, \sigma_{n+2})$ are restricted and non random. This accounts for the image produced.

Attractor	map	a	b	c	d	e	f
The Square (Figure 2.7)	w_0	0.5	0	0	0.5	0	0
	w_1	0.5	0	0	0.5	0.5	0
	w_2	0.5	0	0	0.5	0	0.5
	w_3	0.5	0	0	0.5	0.5	0.5

Table 2.2 IFS code for the *square* attractor.

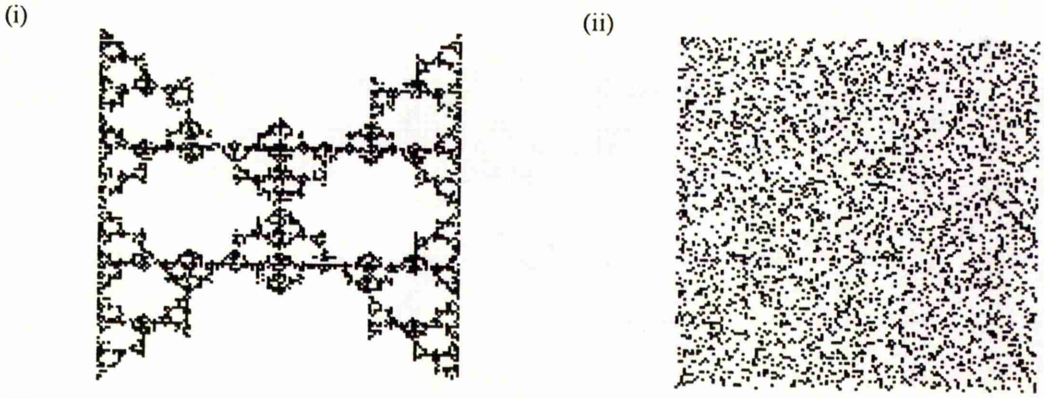


Figure 2.7 The *square* produced by the chaos game with drivers (i) and (ii) are given above. It is remarkable that (i) produces this image as an approximation of the *square*. The triples $(\sigma_n, \sigma_{n+1}, \sigma_{n+2})$ produced from (i) are restricted.

Note In general, it is unwise to take a ± δ to be divisible by high powers of 2 when δ is small and $m = 2^t$ for some positive integer t. Although the shortcomings of RNG's of this form were first illustrated by M. Greenberger [12] in 1965, they were still being widely used more than ten years later [20].

Then the success of the chaos game depends on the choice of RNG driver. It requires a RNG where each number produced is independent of the previous (so that all possible addresses may occur for a prescribed accuracy ε).

The example above illustrates how a poor choice of parameters a, c, x_0 for a given modulus m results in a non-perfect random integer generator, which in turn, results in the failure of the chaos game. In general, it is not possible to state whether or not a sequence is random. We apply a series of tests (both *empirical* and *theoretical*) to the number generator and if it passes them all, then we assume it to be random. Other methods to produce random number generators include combining two unrelated random number generators to produce a 'more' random sequence. Alternatively, z_{n+1} may depend on z_n and z_{n-1} so that the period can be as large as m^2. In fact this may be extended so that z_{n+1} depends on the previous i values. Again Knuth [20] covers this in more detail. However, it is essentially impossible to obtain a RNG which produces all or even most k-digit words over $\{1, ..., N\}$.

2.3.2.2 Using an optimal sequence

For completeness, we state the following. Recall, for a given $\varepsilon > 0$, the corresponding addresses partition $\mathcal{A}$ into subsets each of diameter not exceeding ε. It will be shown later that optimal sequences guarantee that the RIA plots at least one point in each of these subsets. Then, by setting ε sufficiently small (less than one pixel), the RIA with an optimal sequence will produce an accurate approximation of $\mathcal{A}$.

The following theorem holds for any driver (any sequence) which satisfies the stated condition and it gives a measure of the accuracy of the RIA.

Theorem 2.25 Consider the IFS $\{X; w_{1\text{-}N}\}$ with ratio s and attractor $\mathcal{A}$. Suppose the finite sequence $\sigma = \sigma_1\sigma_2\ldots\sigma_m$ contains all k-digit combinations over $\{1, ..., N\}$ and is used as the driver in the RIA to produce a sequence of points $\{x_n\}$ where $x_0 \in \mathcal{A}$ and $x_{n+1} = w_{\sigma_n}(x_n)$. Then, for every point $\alpha \in \mathcal{A}$, there is some point x_n within distance ε where ε satisfies

$$\varepsilon \le s^k \operatorname{diam}(\mathcal{A}).$$

We say $\{x_n\}$ is *ε-dense* in $\mathcal{A}$ [16].

Proof We shall prove the result by expressing $\mathcal{A}$ as a union of sets of diameter not exceeding $s^k \operatorname{diam}(\mathcal{A})$, each containing a point x_n. Recalling Notation 2.18, we have $\mathcal{A}_{\tau_1} = w_{\tau_1}(\mathcal{A})$, $\mathcal{A}_{\tau_1\tau_2} = w_{\tau_1}w_{\tau_2}(\mathcal{A})$, ..., $\mathcal{A}_{\tau_1\ldots\tau_k} = w_{\tau_1}\ldots w_{\tau_k}(\mathcal{A})$ and so on. Let $x, y \in \mathcal{A}_{\tau_1\ldots\tau_k}$. Then $x = w_{\tau_1}\ldots w_{\tau_k}(x')$ and $y = w_{\tau_1}\ldots w_{\tau_k}(y')$ for some $x', y' \in \mathcal{A}$. Further,

$$d(x, y) \le s^k\, d(x',y') \tag{2.8}$$

Write $X = \{x_n\}$ and let $\alpha \in \mathcal{A}$. Recall, $\mathcal{A} = \cup\mathcal{A}_{\sigma_1\ldots\sigma_k}$, $\sigma_i \in \{1,\ldots,N\}$ with $\operatorname{diam}(\mathcal{A}_{\sigma_1\ldots\sigma_k}) \le s^k \operatorname{diam}(\mathcal{A})$. Then $\alpha \in \mathcal{A}_{\tau_1\ldots\tau_k}$ for some $\tau_i \in \{1,\ldots,N\}$, $i = 1, \ldots, k$. But σ contains all k-digit combinations over $\{1, ..., N\}$, so in particular it must contain $\tau_1\ldots\tau_k$. Thus we may write $\sigma = \ldots\tau_k\ldots\tau_1\ldots$ where $\sigma_t = \tau_1$, $\sigma_{t-1} = \tau_2$, ..., $\sigma_{t+k-1} = \tau_k$ for some positive integer t. Then $x_t = w_{\tau_1} \ldots w_{\tau_k}(w_{\sigma_{t-k}} \ldots w_{\sigma_1}(x_0))$ is in the partition $\mathcal{A}_{\tau_1 \ldots \tau_k}$. Therefore,

$$\begin{aligned} d(\alpha, X) &\leq d(\alpha, x_t) \\ &\leq \text{diam } (\mathcal{A}_{\tau_1 \ldots \tau_k}) \text{ since } \alpha, x_t \in \mathcal{A}_{\tau_1 \ldots \tau_k} \\ &\leq s^k \,\text{diam}(\mathcal{A}) \text{ by } (2.8) \end{aligned}$$

Since this inequality holds for any $\alpha \in \mathcal{A}$, we have

$$h(\mathcal{A}, X) \leq s^k \,\text{diam}(\mathcal{A}).$$

Hence result.

2.4 Adaptive Cut Method (ACM)

The *Adaptive Cut Method* (ACM) is one of the best non-RIA algorithms [13, 28]. For a given $\varepsilon > 0$, the ACM plots one point in the partition $\mathcal{A}_\sigma$ for each address σ. Let $x_0 \in \mathcal{A}$, then the ACM provides the approximation

$$\mathcal{A}_\varepsilon = \{x : x = w_{\sigma_1} \ldots w_{\sigma_k}(x_0) : \sigma_1 \ldots \sigma_k \text{ is an address of the partition of } \mathcal{A}\}.$$

This algorithm may be described recursively as below. Let Id denote the identity affine transformation with code (1, 0, 0, 1, 0, 0) and let $x_0 \in \mathcal{A}$. Further let *subdivide* be a recursive function which plots the point $w(x_0)$ if $w = w_{\sigma_1} \ldots w_{\sigma_k}$ and $\rho(\sigma_1\sigma_2 \ldots \sigma_k) \leq \varepsilon < \rho(\sigma_1\sigma_2 \ldots \sigma_{k-1})$. Otherwise it calls *subdivide*(ww_1), ..., *subdivide*(ww_N). Then the adaptive cut method is simply *subdivide*(Id).

In later chapters, we compare the RIA and the ACM.

Chapter 3 Optimal sequences for uniform iterated function systems

Throughout this chapter we restrict our attention to the uniform IFS $\{w_{1\text{-}N}\}$ with ratio s and attractor $\mathcal{A}$. Recall, in a uniform IFS, all ratios s_i are equal $(1 \le i \le N)$. Then each transformation is responsible for covering the same proportion of $\mathcal{A}$ and thus the probability p_i is set to $1/N$ $(1 \le i \le N)$. We define a class of sequences called *optimal sequences*, look at ways of producing these sequences and introduce the *optimal sequence method.* We compare the optimal sequence method with both the chaos game and the ACM (We previously presented some of the following results in [14, 26].).

3.1 Addresses and optimal sequences

Using Definition 2.21, we may explicitly list the addresses of the attractor $\mathcal{A}$:-

Theorem 3.1 Let $\mathcal{A}$ be the attractor of the uniform IFS $\{w_{1\text{-}N}\}$ of ratio s and let $\varepsilon > 0$ be given. Then the corresponding addresses are all k-digit sequences over $\{1, ..., N\}$ where k is the least positive integer such that $s^k \operatorname{diam}(\mathcal{A}) \le \varepsilon$.

Example 3.2 Consider the uniform IFS $\{w_a, w_b\}$ of ratio $1/\sqrt{2}$. Suppose the corresponding attractor is of unit diameter and let $\varepsilon = 1/(2\sqrt{2})$ Then by Theorem 3.1, the addresses consist of all 3-digit sequences over $\{a, b\}$ and the address tree is given in Figure 3.1.

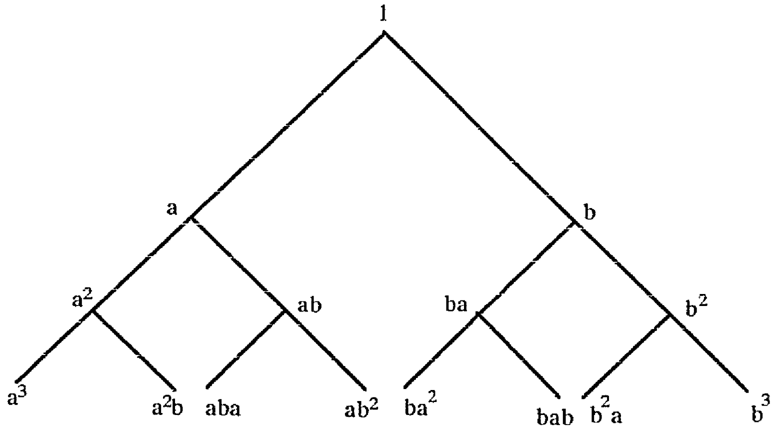


Figure 3.1 The address tree for the uniform IFS $\{w_a, w_b\}$ of Example 3.2.

Notice the address tree has three levels below the starting node and the ith level nodes are the 2^i i-digit words over $\{a, b\}$.

More generally, the address tree for the uniform IFS $\{w_{\sigma_1 - \sigma_N}\}$ has k levels below the starting node 1 and the ith level ($1 \leq i \leq k$) nodes are the N^i i-digit words over $\{\sigma_1, \ldots, \sigma_N\}$.

To achieve uniform distribution when producing an approximation of $\mathcal{A}$, it is reasonable to divide $\mathcal{A}$ into subsets of diameter not exceeding ε and impose the condition that at least one point is plotted in each subset. In fact, for a given $\varepsilon > 0$, the associated addresses provide a natural subdivision of $\mathcal{A}$, namely $\mathcal{A}_\sigma = w_{\sigma_1} \ldots w_{\sigma_k}(\mathcal{A})$ for each address $\sigma = \sigma_1 \ldots \sigma_k$. Let $x_0 \in \mathcal{A}$. Then one method to guarantee a point in each subset $\mathcal{A}_\sigma$, is to plot $w_{\sigma_1} \ldots w_{\sigma_k}(x_0)$ for each address σ. This is the ACM (see Sections 2.4, 3.4 and 7.1). But is it possible for the RIA to meet this condition? Recall, we observed in Section 2.3.2.1 that provided the driver produces a sequence which contains the subblock $\sigma_k \ldots \sigma_1$, then a point will be plotted in the kth-level subset $\mathcal{A}_{\sigma_1 \ldots \sigma_k}$. The example below illustrates this for the *Sierpinski gasket* with $k = 2$.

Example 3.3 Consider again the *Sierpinski gasket* with vertices 1, 2, 3 and suppose the driver produces the following sequence, written from right to left for convenience,

$$\ldots 1 2 3 2 2 1 3$$

We will illustrate what is required of the driver so that every 2nd-level subtriangle Δ_{ij} contains at least one point ($1 \leq i, j \leq N$). Applying the first two transformations, $w_1 w_3(\Delta)$ takes us to Δ_{13}. On applying the next transformation w_2, we land in partition Δ_{213} which is contained in Δ_{21}. Moreover, no matter what our first transformation had been, this would have taken us to Δ_{21}. Applying the other transformations of the sequence, we then visit Δ_{22}, Δ_{32}, Δ_{23}, Δ_{12} etc. Thus, the 2nd-level subtriangle visited after the jth transformation is $\Delta_{\sigma_{j-1}, \sigma_j}$. It is determined by the (j - 1)th and the jth transformation and is independent of all other previous transformations. Then to guarantee that a point is plotted in each of the 2nd-level triangles, the driver must produce a sequence which contains every 2-digit combination on $\{1, 2, 3\}$.

Then obviously, to ensure that at least one point is plotted in each of the kth-level subsets of the attractor of the IFS $\{w_{1-N}\}$, the driver of the RIA must produce a sequence which contains every k-digit combination over $\{1, \ldots, N\}$ [11, 16, 28]. Further, Theorem 2.25 gives us an estimate of how large k must be in order to obtain an accurate approximation of $\mathcal{A}$.

Thus by choosing a sufficiently large k and provided we can find a sequence containing all k-digit combinations over $\{1, \ldots, N\}$, we can use the RIA to obtain a sequence of points $\{x_n\}$ which is arbitrarily close to $\mathcal{A}$. In fact any number generator satisfying this condition will do. The key to speeding up the RIA is finding sequences where every k-digit word appears very quickly. Indeed, these sequences need not even be random! Moreover, it is essentially impossible for a RNG to produce sequences which contain all k-tuples (see Section 2.3.2.1.1) [20].

Theorem 3.4 Over a set of N elements, the shortest possible sequence containing every k-digit word exactly once has length $N^k + k - 1$.

Proof Suppose sequence σ is such a sequence. Then each new position of the sequence must be the start of a unique k-digit word. There are exactly N^k different k-digit words over $\{1, ..., N\}$. Then σ must have N^k digits plus an additional k - 1 to complete the last k-digit word. Hence length(σ) = $N^k + k - 1$.

Our approach is to dovetail the addresses into an *optimal sequence* S which is then used to drive the RIA - the *optimal sequence method.* By an optimal sequence, we mean a sequence of minimal length containing all addresses of $\mathcal{A}$. Consequently, S has length $N^k + k - 1$(Theorem 3.4). Further every address (i.e. every k-digit sequence over $\{1, ..., N\}$) starts in exactly one place and all but the last k - 1 digits start an address. Also, however produced, S is cyclic, that is the last k - 1 symbols are identical to the first k - 1, as proved below.

Theorem 3.5 For uniform IFS's, optimal sequences are cyclic.

Proof Let $a_ib_ic_i$ ... denote the ith address in an optimal sequence S. Write the successive addresses as columns, say :-

Row 1	a_0	a_1	a_2	...	a_m	where $b_0 = a_1$
Row 2	b_0	b_1	b_2	...	b_m	$c_0 = b_1 = a_2$
Row 3	c_0	c_1	c_2	...	c_m	and so on.
	.	.	.	...	.	

We claim that any fixed symbol $\tau \in \{1, ..., N\}$ appears in each row the same number of times. Consider for example rows 1, 2. Since permuting the symbols of an address gives another address, we have a bijection defined by xy... $\leftrightarrow$ yx... between addresses with τ in the first position (τ in row 1) and those with τ in the second position (τ in row 2). Thus τ appears the same number of times in rows 1 and 2. Similarly for any other two rows. Now consider again rows 1, 2. We have $b_0b_1...b_{m-1} = a_1a_2...a_m$ and hence $b_m = a_0$. Similarly $c_m = b_0 = a_1$ and so on. Thus S is cyclic.

Lemma 3.6 If S is an optimal sequence for the uniform IFS $\{w_{1-N}\}$ then so is its reverse.

Proof Reversing the entries of an address gives an address.

In view of Theorem 3.5, we may now define an optimal sequence formally as:-

Definition 3.7 For a given $\varepsilon > 0$, there exists a least positive integer k such that s^k diam($\mathcal{A}$) $\leq \varepsilon$, and an *optimal sequence* of the uniform IFS $\{w_{1-N}\}$ is a sequence of period length N^k which when continued cyclically to $N^k + k - 1$ terms, contains every k-digit word over $\{1, ..., N\}$ exactly once.

Lemma 3.8 Consider the uniform IFS $\{w_{1-N}\}$ and associated optimal sequence of Definition 3.7. Suppose this optimal sequence is used to drive the RIA. Then, after $N^k + k - 1$ iterations, the points $\{x_n\}$ will be ε-dense in $\mathcal{A}$. We shall refer to this as the *optimal sequence method* for uniform IFS.

Proof The result follows from Theorem 2.25, since any $N^k + k - 1$ consecutive terms of an optimal sequence contain every k-digit word exactly once.

Example 3.9 Then the sequence $S = a^3bab^3a^2$ is an optimal sequence for the uniform IFS $\{w_a, w_b\}$ of Example 3.2. Notice that the first two digits are equal to the last two (Theorem 3.5), indicating that the sequence is cyclic. S is of period length $2^3 = 8$ and its reverse is also a valid optimal sequence.

In Sections 3.2, 3.3, we study ways of obtaining optimal sequences for any positive integers N and k. We also compare the results of the chaos game and the optimal sequence method.

3.2 Obtaining optimal sequences from graphs

We seek ways of finding sequences of minimal length containing every k-digit word exactly once. One method involves graph theory. We first introduce the required terms [6, 7, 19].

Definition 3.10

(i) A *graph* G = (V, E) is defined to be a set of points, V, (called *vertices*) together with a set of lines, E, (called *edges*) joining certain vertices.

(ii) It is *directed* if there is a direction associated with each edge. i.e. let e = (v_i, v_j) be an edge in a directed graph.Then v_i is the *initial vertex* of e while v_j is the *final vertex.*

(iii) The *in (out)-degree* of a vertex $v_i \in V$ is the number of edges where v_i is the final (initial) vertex.

(iv) A directed graph is *balanced* provided each $v_i \in V$ has in-degree(v_i) = out-degree(v_i).

(v) A *path p*, from vertex v_{i_1} to vertex v_{i_k}, is a finite sequence of edges of the form

$$p = (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots, (v_{i_{k-1}}, v_{i_k}).$$

(vi) A directed graph is *connected* if and only if for each distinct pair of vertices v_i, v_j there exists a path from v_i to v_j or from v_j to v_i.

(vii) A vertex v of a directed graph is *isolated* if in-degree(v) = out-degree(v) = 0.

(viii) An *Eulerian circuit* in a directed graph is an oriented path such that every edge in the directed graph occurs exactly once and the initial vertex of the first edge is equal to the final vertex of the last edge.

Theorem 3.11 A finite directed graph G = (V, E) with no isolated vertices possesses an Eulerian circuit if and only if it is connected and balanced [19].

Proof We shall assume G is balanced and let P = $(e_1, \ldots, e_m)$ be an oriented path of longest possible length that uses no edge more than once where e_j denotes the edge (v_j, v_{j+1}) (for j = 1, ..., m). Then v_{m+1} is the final vertex of the path, and if k is the out-degree of v_{m+1}, then all k edges e with initial vertex v_{m+1} must appear in P. For otherwise, we could add e to P and obtain a longer path. But if the initial vertex of the edge e_j with $j > 1$ equals v_{m+1}, then the final vertex of edge e_{j-1} must also equal v_{m+1}. Hence since G is balanced, we must have $v_1 = v_{m+1}$, for otherwise the in-degree of vertex v_{m+1} would be at least k + 1. Now, by cyclic permutation of P, it follows that any edge e not in the path P, has neither initial or final vertex in common with any edge e_j $(1 \le j \le m)$ in the path. Thus if P is not Eulerian then G is not connected.

Recall, we wish to find a sequence of minimal length which contains every k-digit combination of {1, ..., N} exactly once. Let $G_k = (V_k, E_k)$ be the directed graph where

$$V_k = \{v_i : v_i = \sigma_{i_1}\ldots\sigma_{i_{k-1}}; \sigma_{i_j} \in \{1,\ldots, N\}\}$$

and

$$E_k = \{(v_i, v_j) : v_i = \sigma_{i_1}\ldots\sigma_{i_{k-1}}, v_j = \sigma_{i_2}\ldots\sigma_{i_{k-1}}\sigma_{i_k} ; \sigma_{i_j} \in \{1,\ldots,N\}\}.$$

Then G_k has N^{k-1} vertices each one corresponding to a different (k - 1)-digit word on $\{1, ..., N\}$ and in-degree(v_i) = out-degree(v_i) = N for all $v_i \in V_k$. Hence, G_k is balanced and has no isolated vertices. Further, it is easily checked that G_k is connected. Thus, by Theorem 3.11, G_k has an Eulerian circuit.

Theorem 3.12 From any Eulerian circuit of the graph $G_k = (V_k, E_k)$ above, construct the finite sequence $\sigma = \sigma_1\ldots\sigma_{N^k}$ as follows. Let

σ_1 = first digit of the label of the initial vertex,

σ_2 = first digit of the label of next vertex visited,

.

.

.

and σ_{N^k} = first digit of the second last vertex visited.

Then the sequence σ is a cyclic sequence of period N^k, which when continued to $N^k + k - 1$ terms contains every k-digit combination of $\{1, ..., N\}$ [6], and so may be used in the optimal sequence method for the uniform IFS $\{w_{1-N}\}$.

Proof Label the edge from vertex $\sigma_{i_1}\ldots\sigma_{i_{k-1}}$ to vertex $\sigma_{i_2}\ldots\sigma_{i_k}$ by $\sigma_{i_1}\ldots\sigma_{i_k}$ for every edge of G_k. Then, in any Eulerian circuit, the (k - 1) consecutive edges following the arbitrary edge $\sigma_{i_1}\ldots\sigma_{i_k}$ necessarily have the form

$$\sigma_{i_2}\sigma_{i_3}\ldots\sigma_{i_{k-1}}\sigma_{i_k}\tau_1$$
$$\sigma_{i_3}\sigma_{i_4}\ldots\sigma_{i_k}\tau_1\tau_2$$
$$\ldots$$
$$\sigma_{i_k}\tau_1\ldots\tau_{k-2}\tau_{k-1}$$

for appropriate letters $\tau_1, \tau_2, \ldots, \tau_{k-1}$. The corresponding terms of the finite sequence σ are $\sigma_{i_1}\ldots\sigma_{i_k}$ which is the edge (that is, the k-digit word) with which we arbitrarily started. Thus σ clearly contains every k-digit word and is of length N^k. Hence result.

Example 3.13

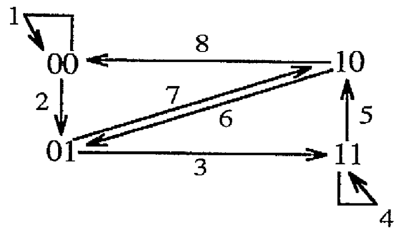


Figure 3.2 G_2 where N = 2.

Note Numbered edges correspond to an Eulerian circuit.

For N = 2, k = 2, consider the graph $G_k = (V_k, E_k)$ of Figure 3.2. The edges are numbered 1 to 8 to indicate an Eulerian circuit and from Theorem 3.12, the corresponding optimal sequence is S = 0001110100.

We have stated a method for finding optimal sequences for any positive integers N, k. However, finding Eulerian circuits can prove to be a 'slowish' process particularly for larger values of N and k. We seek a simpler way to produce optimal sequences. In the next section, we consider linear recurrence relations.

3.3 Obtaining optimal sequences from linear recurrence relations

In this section, we show how linear recurrence relations produce optimal sequences. We begin by considering the uniform IFS $\{w_{1\text{-}N}\}$ where N is a prime power.

3.3.1 IFS with $N = p^r$

Recall, we wish to obtain a sequence of length $N^k + k - 1$ containing every k-digit combination over N elements. Provided $N = p^r$, for some prime p, then the answer lies in finite field theory.

Informally, a field is a set F together with '+' and '.', two binary operations defined over F. It contains a zero, and an identity, usually denoted by 0 and 1 respectively, and satisfies the usual rules of arithmetic. Further, every non zero element, a, of F has an inverse, $a^{-1} (\in F)$, such that $a.a^{-1} = a^{-1}.a = 1$. If $a.b = 0$ $(a, b \in F)$, then we must have $a = 0$ or $b = 0$. Real, rational and complex numbers are all examples of infinite fields. However, finite fields also exist. The necessary definitions and results are given below but for more background on finite fields refer to [23] or [25]. Unless otherwise stated, let $N = p^r$, throughout Section 3.3.1.

Let F be an arbitrary field. Then F[x] consists of all polynomials in x with coefficients in F (or more briefly all polynomials over F). If f(x) is an irreducible polynomial over F (i.e. f(x) cannot be expressed as the product of some non-constant polynomials over F), then F[x] (mod f(x)), the set of polynomials in F[x] with the condition f(x) = 0 imposed, is a field. Let F_N denote a finite field with N elements.

Definition 3.14

(i) A *ring* is a system (R, +, .) consisting of a non-empty set R on which are defined the binary operations '+', '.' satisfying (for all a, b, c $\in$ R):-

(A0) R is closed under '+'

(A1) a + b = b + a

(A2) (a + b) + c = a + (b + c)

(A3) There is a *zero* in R, usually denoted 0, such that 0 + a = a + 0 = a

(A4) For all a in R, there is -a in R such that a + (-a) = (-a) + a = 0.

(M1) R is closed under '.'

(M2) (a.b).c = a.(b.c)

(D) a.(b + c) = a.b + a.c and (a + b).c = a.c + b.c

It is usually denoted simply by R.

(ii) In a ring R, an *identity*, denoted 1, is an element such that 1.a = a.1 = a, for all a $\in$ R.

(iii) A ring R is *commutative* provided x.y = y.x for all x, y $\in$ R.

(iv) Let a $\in$ R where R is a ring with identity 1. If there exists $a^{-1} \in$ R such that $a.a^{-1} = a^{-1}.a = 1$, then a is a *unit*. Further, R is a *division ring* if every non-zero element is a unit.

(v) Let R be a ring with a zero, 0. Then a non-zero element a $\in$ R is said to be a *divisor of zero* if there exists a non-zero b $\in$ R such that a.b = 0 or b.a = 0.

(vi) A *field* is a commutative division ring.

Theorem 3.15 (Existence & Uniqueness) For every prime p and every positive integer r there is (up to isomorphism) exactly one finite field with N = p^r elements. It may be constructed as follows. Let f(x) be an irreducible polynomial of degree r over F_p. Then

$$F_N = F_p[x] \ (\mathrm{mod}\ f(x)) \qquad (3.1)$$

is a finite field of order N.

The above uniqueness property permits us to talk of F_N, the finite field of order N. When N = p, the elements of F_p are often represented by {0, 1, ..., p-1}, the set of integers reduced modulo p (i.e. with the condition p = 0 imposed) and F_p is called Z_p. The example below illustrates how (3.1) may be applied in practice.

Example 3.16 It is easily checked that $f(x) = x^3 + x + 1$ is a 3rd degree irreducible polynomial over F_2. Then $F_8 = F_2[x]$ (mod f(x)), and every element of F_8 may be expressed uniquely as a polynomial $a\alpha^2 + b\alpha + c$ where a, b, c $\in F_2$ and α is a root of f in F_8. In fact, $F_8 = \{0, 1, \alpha, \alpha^2, \alpha+1, \alpha^2+\alpha, \alpha^2+\alpha+1, \alpha^2+1\}$.

Definition 3.17 Let k be a positive integer and let $a_1, \ldots, a_k$ be elements of the finite field F_N. Then a sequence of elements $\sigma_0, \sigma_1, \ldots$ of F_N satisfying the relation

$$\sigma_n = a_1\sigma_{n-1} + a_2\sigma_{n-2} + \ldots + a_k\sigma_{n-k} \tag{3.2}$$

is called a *(kth-order) linear recurring sequence.* The relation is a *(kth-order) homogenous linear recurrence relation.* Associated with this recurrence relation is the *characteristic polynomial*

$$f(x) = x^k - a_1x^{k-1} - \ldots - a_{k-1}x - a_k. \tag{3.3}$$

Note Each digit σ_n is determined recursively from the previous k digits by (3.2). Thus, if a sequence of k successive terms recurs then so do all subsequent terms, and we say σ is *periodic.*

Example 3.18 Over Z_3, let $f(x) = x^3 - x - 1$ be the characteristic polynomial of the 3rd-order linear recurring sequence σ. Then, by (3.2) $\sigma_n = \sigma_{n-2} + \sigma_{n-3}$, or more simply σ_n is the sum of two digits back and three digits back mod 3. Suppose, we set $\sigma_1\sigma_2\sigma_3 = 001$, then σ = 0010111220120010... .

After 13 digits, the 3-bit word 001 reappears, followed by 010, ... and so forth. Then σ has period 13. Further, no matter how long we continue σ, it will only ever include the following 3-digit words over {0,1,2}:- 001, 010, 101, 011, 111, 112, 122, 220, 201, 012, 121, 210, 100. We require a sequence of period $N^k = 27$.

Definition 3.19 Let k be a positive integer. Further, let σ be a kth-order linear recurring sequence over F_N with corresponding characteristic polynomial f(x). The greatest possible period of such a sequence is $N^k - 1$. If σ has period $N^k - 1$, then, σ is called a *Maximal sequence (M-sequence).* In this case, f is said to be *primitive* over F_N.

Example 3.20 Over Z_2, let $f(x) = x^4 - x - 1$, be the characteristic polynomial of σ. Then $\sigma_n = \sigma_{n-3} + \sigma_{n-4}$. Setting $\sigma_1 \ldots \sigma_4 = 0001$, we obtain σ = 0001001101011110001... . Then σ, having period $2^4 - 1 = 15$, is a M-sequence.

Further, $\sigma_1 \ldots \sigma_{18}$ contains every non-zero 4-digit binary word exactly once. Hence, by setting $\sigma_0 = 0$, we have that $\sigma_0 \ldots \sigma_{18}$ is a sequence of minimal length (see Theorem 3.4) containing every 4-digit word over $\{0, 1\}$ exactly once.

Note The maximum possible period of a kth-order linear recurring sequence over F_N is $N^k - 1$. Then an M-sequence is so called because it has period $N^k - 1$. On the other hand, an M-sequence gives a sequence of minimal length containing every possible non-zero k-tuple over F_N.

So, by choosing a primitive polynomial of degree k over F_N to be the characteristic polynomial of the sequence σ and setting $\sigma_0 \ldots \sigma_k = 0 \ldots 01$, we can easily obtain a sequence of period length N^k from the associated linear recurrence relation. In other words, an M-sequence with an additional initial zero yields an optimal sequence. Moreover, by associating a different transformation w_i $(1 \le i \le N)$ with each of the elements of F_N, these sequences can be used in the optimal sequence method to produce the attractor of the uniform IFS $\{X; w_{1\text{-}N}\}$.

Note If the optimal sequence method involves a kth-order linear recurrence, we generally refer to it as the *M-sequence method.*

Definition 3.21 For any non-zero element $\alpha \in F_{N^k}$ $(k \ge 1)$, the *order* of α is the least positive integer t such that $\alpha^t = 1$. An element of order $N^k - 1$ is called *primitive.*

Definition 3.22 Associated with each $\alpha \in F_{N^k}$ is a unique irreducible polynomial $M_\alpha(x) \in F_N$, with leading coefficient one, of least degree such that $M_\alpha(\alpha) = 0$. It is called the *minimal polynomial* of α.

Theorem 3.23 For every integer $k \ge 1$, there exists a primitive polynomial of degree k over F_N. Moreover, it is the minimal polynomial of a primitive element $\alpha \in F_{N^k}$ and may be determined by

$$M_\alpha(x) = \prod_{i=1}^{k}(x - \alpha^{N^i}).$$

Example 3.24 Let $F_4 = \{0, 1, b, b^2\}$ where $b^2 = b + 1$. We wish to find a primitive polynomial of degree 3 over F_4. The polynomial $f(x) = x^3 - x - 1$ is irreducible over F_4. Let $\alpha \in F_{4^3}$ be a root of f (i.e. $\alpha^3 = \alpha + 1$). Then $F_{4^3} = F_4[x]$ (mod f(x)). It is easily checked that $\beta = b\alpha + 1$ has order 63 and so is a primitive element of F_{4^3}. Further,

$M_\beta(x)$, the minimal polynomial of β, is a primitive polynomial of degree 3 over F_4. In fact, $M_\beta(x) = x^3 - x^2 - bx - b - 1$. Further, the corresponding linear recurrence relation will produce a sequence of period $4^3 - 1$.

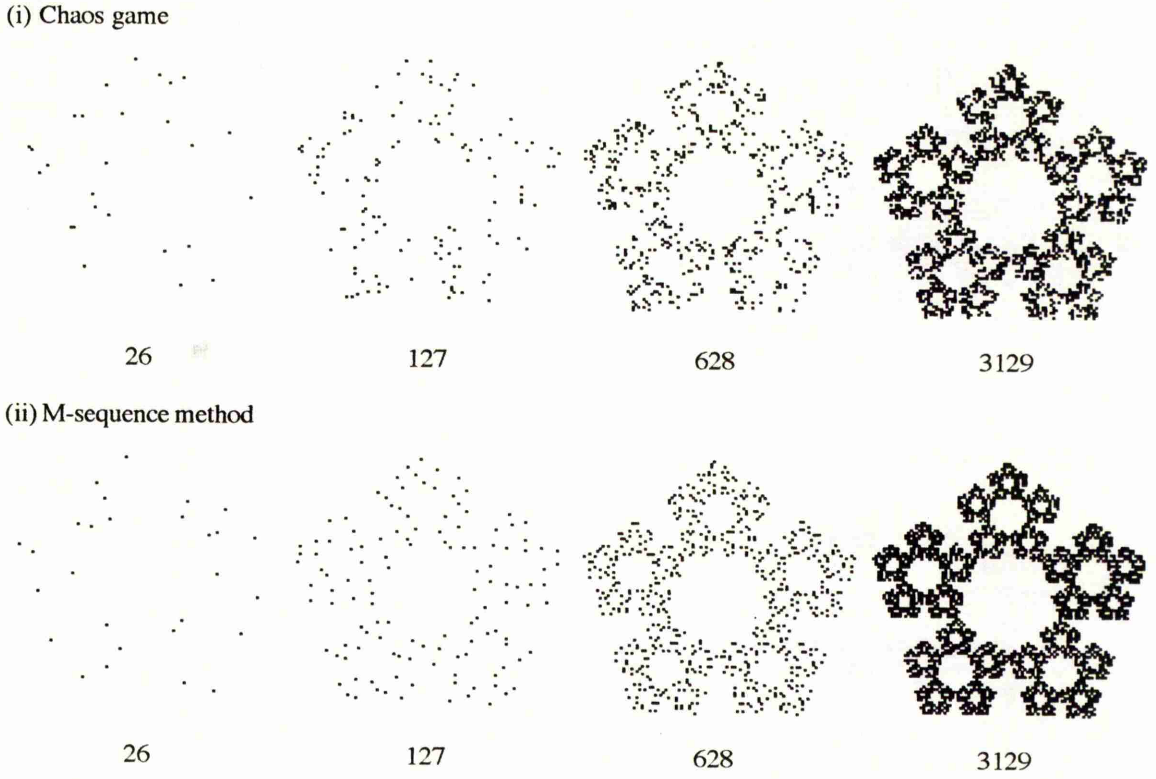
Then provided $N = p^r$, it is possible to find a kth-order linear recurring sequence of period $N^k - 1$. Goodman [11] and Hoggar [16] independently illustrated the superiority of the M-sequences method over the chaos game when producing the Sierpinski gasket. In the examples below we consider IFS's with N = 5, 8, 9. We compare the results of (i) the chaos game and (ii) the M-sequence method. Our RNG driver is based on the Macintosh RandomX.

Example 3.25 The *crystal* is the attractor of an IFS with five transformations, each responsible for 1/5th of the final area. The IFS code is given in Table 3.1. It is possible to find M-Sequences of degree k over Z_5 (see below Theorem 3.15). The crystal attractor was produced using (i) the chaos game and (ii) the M-sequence method.

The maximum distance between any two points of this attractor, diam ($\mathcal{A}$), is 101 pixels and the crystal IFS has contractivity ratio 0.382. The linear recurring relation corresponding to a kth degree primitive polynomial over Z_5, will produce an M-sequence. Then, by Lemma 3.8, the M-sequence method approximation, $\{x_n\}$, will be ε-dense in the attractor where $\varepsilon \leq (0.383)^k\ 101$. For k = 2,..., 5, Table 3.3 below gives the required information.

Attractor	map	a	b	c	d	e	f
Crystal (Figure 3.3)	w_1	0.382	0	0	0.382	0.3072	0.6190
	w_2	0.382	0	0	0.382	0.6033	0.4044
	w_3	0.382	0	0	0.382	0.0139	0.4044
	w_4	0.382	0	0	0.382	0.1253	0.0595
	w_5	0.382	0	0	0.382	0.4920	0.0595

Table 3.1 The IFS code for the *crystal.*



(i) Chaos game

(ii) M-sequence method

Figure 3.3 Comparison of the approximations of the *crystal* produced by (i) the chaos game and (ii) the M-sequence method (The corresponding number of iterations is given below each image.).

Number of Iterations (k)	Chaos game		M-sequence method	
	% Original Points	Time Taken (s)	% Original Points	Time Taken (s)
26 (2)	100.0	0.5	100.0	0.5
127 (3)	99.2	2	100.0	1
628 (4)	89.5	12	99.4	9
3129(5)	62.9	62	78.2	46

Table 3.2 Percentage of original points and the time taken for each approximation of the *crystal.*

k	Coeffs. of Primitive Poly $a_1 \ldots a_k$	Number of Iterations = 5^k+k-1	$\varepsilon \le (0.382)^k.\mathrm{diam}(\mathcal{A})$ and $\mathrm{diam}(\mathcal{A}) = 137$pixels
2	22	26	15 pixels
3	012	127	7
4	1012	628	2
5	00012	3129	1

Table 3.3 Primitive polynomials, length of corresponding M-sequences and accuracy for the M-sequence approximation of the *crystal* attractor for k = 2,...,5.

Although for a small number of iterations, both algorithms produce a high percentage of original points, even at this stage the image obtained by the M-sequence method is superior since it gives a better overall impression of the final attractor.

As the number of iterations increases, the chaos game attractor sees a significant drop in the number of original points while even after 3129 iterations using the M-sequence method, 78.2 % of the points plotted are original.

Clearly, the M-sequence method does indeed ensure the points are well distributed. Moreover, the M-sequence method is also faster than the chaos game with RandomX (see Section 7.1.1).

Example 3.26 The *Sierpinski carpet* requires eight affine transformations $w_1, \ldots, w_8$ each with associated probability $p_i = 1/8$ while the *Peano curve* is described by nine (equal probability) affine transformations (see Table 3.13 for IFS code). Let α denote a primitive element of F_8 (where $\alpha^3 = \alpha^2 + 1$). Recall, from Example 3.16, each element of F_8, may be expressed in terms of α. It is easily checked that $x^4 - x - \alpha^2$ and $x^5 - \alpha x^4 - x^2 - (\alpha^2 + \alpha + 1)x - (\alpha^2 + \alpha)$ are primitive polynomials of degree 4 and 5 respectively over F_8. Similarly, let β denote a primitive element of F_9 (where $\beta^3 = 2\beta + 1$). Then $x^3 - x - \beta$ and $x^5 - (2\beta + 2)x^4 + (\beta + 2)x^3 - 2\beta x^2 + 2x - \beta$ are primitive polynomials over F_9. The associated M-sequences were used in the optimal sequence method and the results produced were superior to the chaos game. The corresponding Peano curve images can be found in Section 3.4 where we compare the M-sequence method and the ACM.

For any IFS with p^r transformations, it is always possible to define a kth-order linear recurrence relation which will produce an M-sequence. However, we have not considered what happens when $N \neq p^r$. In the next section we set out to see if there are any kth-order linear recurrence relations over $N \neq p^r$ elements with period $N^k - 1$, or N^k.

3.3.2 IFS with $N \neq p^r$

Ideally, we would like to be able to find a kth-order linear recurring sequence of period $N^k - 1$, over something with $N \neq p^r$ elements. We again refer to these sequences as *M-sequences.*

For $N \neq p^r$, consider $Z_N = \{0, 1, ..., N-1\}$; the set of integers reduced modulo N. Then $(Z_N, +, .)$ is a ring with a zero where multiplication and addition are performed mod N. Further, $(Z_N, +, .)$ contains some divisors of zero. For example over Z_6, although $2, 3 \neq 0$ we have $2.3 = 0$. Thus 2, 3 are divisors of zero. The following lemma shows that M-sequences cannot be produced by linear recurrence relations over the ring $(Z_N, +, .)$.

Lemma 3.27 If N is not the order of a finite field, say $N = N_1N_2...N_s$, $s \geq 2$, where each N_i is a power of a distinct prime. Then the period of the linear recurring sequence (3.2) over Z_N falls short of N^k by an amount at least

$$s \prod_{i=1}^{k}(N_i^k - 1)^{1-1/s}.$$

Proof The general case is clear from the proof for $s = 3$. For positive integers a, b, c we have $(a + 1)(b + 1)(c + 1) = a(b + 1)(c + 1) + (b + 1)(c + 1) \geq ... \geq abc + ab + bc + ca \geq abc + 3(abc)^{2/3}$, since the arithmetic mean of ab, bc, ca equals at least its geometric mean. From Knuth [20], the period of (3.2) mod N equals the lcm over i of its periods mod N_i. These cannot exceed $N_i^k - 1$, by exclusion of the all 0's case. Setting $N_1^k = a + 1$, $N_2^k = b + 1$, $N_3^k = c + 1$ above we have (since the lcm of numbers does not exceed their product), that the excess of N^k over the actual period of (3.2) is at least $3[(N_1^k - 1)(N_2^k - 1)(N_3^k - 1)]^{2/3}$.

Example 3.28 By Lemma 3.27, the period of a sequence produced by a 3rd order linear recurrence over the ring $(Z_6, +, .)$ can be no more than 189. In practice, the maximum period of a 3rd order linear recurring sequence over the ring $(Z_6, +, .)$ is 182.

Thus conventional modular addition and multiplication over Z_N fails to give sequences of large enough period. In an attempt to obtain M-sequences we shall redefine the multiplication '.' over the additive group $(Z_N, +)$.

Definition 3.29 A set Q is a *quasigroup* if there is a binary operation, '.', defined on Q and if when any two elements $a, b \in Q$ are given, the equations $a.x = b$ and $y.a = b$ each have exactly one solution.

A *loop* L is a quasigroup with an identity. That is, a quasigroup in which there exists an element $e \in L$ such that $e.x = x.e = x$ for all $x \in L$ [8].

Definition 3.30 For any positive integer N, an (N x N) *latin square* on the numbers $\{1, 2, ..., N\}$ is an array of N rows and N columns where each row and each column contains each of 1, 2, ..., N exactly once [8].

Note We shall use the convention of denoting N by 0.

Theorem 3.31 The multiplication table of a quasigroup (and a loop) is a latin square [8].

Proof Let $a_1, \ldots, a_N$ denote the elements of the quasigroup Q with associated binary operation '.'. Further, let $a_i.a_j = a_{ij}$. Then, the (i, j)th entry of the multiplication table for '.' over Q is the product of the elements a_i and a_j. Suppose an element occurs twice in the same row so that $a_{ij} = a_{ik} = a$, say. But this gives two solutions to the equation $a_i.b = a$, contradicting the quasigroup axiom. A similar argument may be applied to each column. Hence each element occurs in every row and every column exactly once. It follows immediately that the (unbordered) multiplication table is a latin square.

Definition 3.32 The structure $(Z_N, +, .)$ is a *far ring* provided (i) $(Z_N, .)$ is a loop and (ii) $(Z_N, +)$ is the additive group of integers mod N.

The next question to be answered is :- Do M-sequences exist over far rings? Since we are particularly interested in the case $N \neq p^r$, we begin by considering a far ring with six elements; six being the smallest non-trivial positive integer which is not a prime power. Let FR_1 be the far ring $(Z_6, +, .)$ with elements denoted {1, ..., 5, 0} and with multiplication '.' given by Table 3.4(i).

(i)

.	1	2	3	4	5	0
1	1	2	3	4	5	0
2	2	1	4	5	0	3
3	3	5	2	0	1	4
4	4	0	1	3	2	5
5	5	4	0	1	3	2
0	0	3	5	2	4	1

(ii)

*	1	2	3	4	5	0
1	1	2	3	4	5	0
2	2	4	1	0	3	5
3	3	1	2	5	0	4
4	4	5	0	3	2	1
5	5	0	4	2	1	3
0	0	3	5	1	4	2

Table 3.4 Multiplication table for the far rings (i) FR_1 $(Z_6, +, .)$ (ii) FR_2 $(Z_6, +, *)$.

For k = 2, it is found that there are seven 2nd-order linear recurring sequences over the far ring FR_1 of maximal period $6^2 - 1 = 35$. In other words M-sequences of period 35 exist over FR_1. This is very promising. The periods of the other 2nd-order linear recurring sequences include 3, 8, 27, 29 and 34.

On considering k = 3, there are no linear recurring sequences of period $6^3 - 1 = 215$ over FR_1. There are, however three 3rd-order linear recurring sequences of period 214. With k = 4, there are two M-sequences of period $6^4 - 1$.

Now consider the far ring FR_2 $(Z_6, +, *)$ with multiplication table 3.4(ii). It is found for k = 2, 3, 4 that M-sequences of period $6^k - 1$ do exist over FR_2.

For $k = 2, 3, 4$, Table 3.5 (Table 3.6) gives the coefficients of all kth-order linear recurrence relations which define sequences of the maximum period possible over the far rings FR_1 (FR_2). If this maximum period is $6^k - 1$ then the corresponding sequences are M-sequences. Otherwise, although these sequences are not actually M-sequences, they may be used in the optimal sequence method since the difference in period is very small (e.g. for $k = 3$ over FR_1, the period is 214 rather than 215).

a_1a_2	Period	M-Seq	$a_1a_2a_3$	Period	M-Seq	$a_1a_2a_3a_4$	Period	M-Seq
01	35	√	130	214	×	1022	1295	√
04	"	√	150	"	×	1324	"	√
10	"	√	541	"	×			
20	"	√						
41	"	√						
51	"	√						
52	"	√						

Table 3.5 Coefficients and period of all linear recurring sequences $\sigma_n = a_1\sigma_{n-1} +...+ a_k\sigma_{n-k}$ with maximum period possible over the far ring FR_1 for $k = 2, 3, 4$.

a_1a_2	Period	M-seq	$a_1a_2a_3$	Period	M-Seq	$a_1a_2a_3a_4$	Period	M-Seq
12	35	√	135	215	√	0243	1295	□√
21	"	√	513	"	√	1524	"	√
23	"	√	523	"	√	2133	"	√
31	"	√				2305	"	√
						2312	"	√
						3321	"	√
						5454	"	√

Table 3.6 Coefficients and period of all linear recurring sequences $\sigma_n = a_1\sigma_{n-1} +...+ a_k\sigma_{n-k}$ with maximum period possible over the far ring FR_2 for $k = 2, 3, 4$.

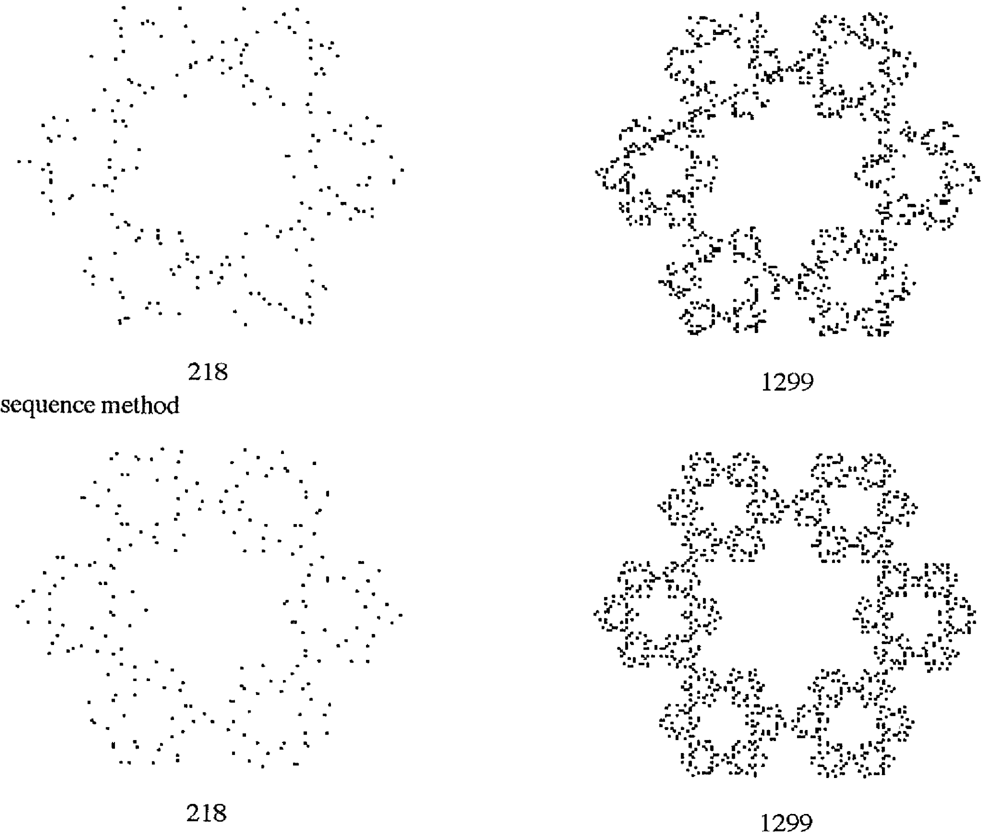
Remark 3.33 The period of any kth-order linear recurring sequence defined over the finite field F_N where $N = p^r$ is always a divisor of $N^k - 1$, provided the characteristic polynomial is irreducible. Can we make an analogous statement for far rings? - Observe, multiplication '.' is not distributive over addition in the far ring $(Z_N, +, .)$. Indeed, for any a, b in Z_N, $a.b = a.(b + 0) = a.b + a.0$ implies $a.0 = 0$, contradicting the definition of a quasigroup. Then the operations '.' and + of polynomials over the far ring $(Z_N, +, .)$ are not well defined. For example $a_1x + b_1x \neq (a_1 + b_1)x$ and $(a_0 + a_1x).(b_0 + b_1x) \neq a_0b_0 + (a_0b_1 + a_1b_0)x + a_1b_1x^2$ (for any a_0, a_1, b_0, b_1 in Z_N). Hence it is not possible to define characteristic polynomials for linear recurrences over far rings.

Definition 3.34 For any positive integer N, let $(Z_N, +, .)$ be a far ring with N elements over which a kth-order linear recurring sequence of period $N^k - 1$ exists. Then the latin square corresponding to the multiplication over this far ring is defined to be an *(N x N) k-recurrent latin square.*

Example 3.35 In this example we use M-sequences over the far ring FR_2 in the M-sequence method to produce a *hexagonal gasket*. The results are compared with those of the chaos game.

As in the previous example, the M-sequence method plots a higher percentage of original points than the chaos game. Further, the points, being well distributed, give a clearer view of the final attractor. Again, the M-sequence method is faster than the chaos game.

(i) Chaos game



218

1299

(ii) M-sequence method

218

1299

Figure 3.4 Comparison of the approximations of *hexagonal gasket* produced by (i) the chaos game and (ii) the M-sequence method (Again the corresponding number of iterations are given below each image.).

Attractor	map	a	b	c	d	e	f
Hexagonal gasket (Figure 3.4)	w_1	1/3	0	0	1/3	0	0
	w_2	1/3	0	0	1/3	1/3	$1/\sqrt{3}$
	w_3	1/3	0	0	1/3	1/3	$-1/\sqrt{3}$
	w_4	1/3	0	0	1/3	1	$1/\sqrt{3}$
	w_5	1/3	0	0	1/3	1	$-1/\sqrt{3}$
	w_6	1/3	0	0	1/3	4/3	0

Table 3.7 The IFS code for the *hexagonal gasket* [18].

Number of Iterations (k)	Chaos game		M-sequence method	
	% Original Points	Time Taken (s)	% Original Points	Time Taken (s)
218 (3)	98.1	5	98.6	3
1299 (4)	85.7	27	96.8	22

Table 3.8 Percentage of original points and the time taken for each approximation of the *hexagonal gasket.*

Note The M-sequence method produces an image which is ε-dense in the attractor where $\varepsilon \leq 5.3$ pixels for $k = 3$ and $\varepsilon \leq 1.8$ pixels for $k = 4$.

The discovery that M-sequences may be defined over far rings with six elements is very exciting. It provides a simple way to produce optimal sequences when $N \neq p^r$. A method to produce k-recurrent latin squares and the corresponding sequences must be sought. The new mathematics arising from this study will be of interest to people of different research backgrounds and results obtained may be pursued elsewhere. We study far rings and k-recurrent latin squares in Chapter 4.

3.4 A comparison

For uniform IFS's, we have illustrated the improvement in the RIA when it is driven by an optimal sequence (in particular an M-sequence) rather than a RNG. However, it is only fair that we compare the M-sequence method to another efficient algorithm. Consider the IFS $\{X; w_{1\text{-}N}\}$ with uniform probabilities, ratio s and attractor $\mathcal{A}$. Both the optimal sequence method and the ACM ensure that a point is plotted in each kth-level subset $\mathcal{A}_{\sigma_1 \ldots \sigma_k}$ (corresponding to the address $\sigma_1 \ldots \sigma_k$) to produce an image which is ε-dense in $\mathcal{A}$ ($\varepsilon \leq s^k \operatorname{diam}(\mathcal{A})$). One difference between the two algorithms is the precise choice of the point plotted in each subset. Let $x_0 \in \mathcal{A}$, then the ACM provides the attractor approximation $\mathcal{A}_\varepsilon = \{x : x = w_{\sigma_1}\ldots w_{\sigma_k}(x_0) : \sigma_i \in \{1,\ldots, N\}\}$. After essentially the same number of points have been calculated, each algorithm produces an approximation which is ε-dense in $\mathcal{A}$ where $\varepsilon \leq s^k\text{diam}(\mathcal{A})$. However, to reach this kth-level approximation using the ACM involves much more work than using the M-sequence method, since the code, (a, b, c, d, e, f), for each of N^k composite affine transformations $w_{\sigma_1}\ldots w_{\sigma_k}$ must be computed. Indeed this may be done recursively to reduce work (Algorithm 7.1) but it could never be as fast as the M-sequence method which simply involves computing σ_n from a kth-order linear recurrence relation and then performing w_{σ_n} (see Section 7.1.1 for further analysis of the speed).

Below we compare the ACM, and the M-sequence method, for various IFS (The code for each can be found in Table 3.13.). The time (in seconds), the number of original points and the image produced must be considered in each case. In some of the more interesting cases, the chaos game results are also given.

	ACM (Number of Iterations = 3^k)		M-sequence method (Number of Iterations = $3^k + k - 1$)	
k	% Original Points	Time Taken (s)	% Original Points	Time Taken (s)
6	100	30	95	11
7	75	91	75	33

Table 3.9 Percentage of original points and the time taken for (i) the ACM and (ii) the M-sequence method approximation of the *Sierpinski gasket.*

	ACM (Number of Iterations = 8^k)		M-sequence method (Number of Iterations = $8^k + k - 1$)	
k	% Original Points	Time Taken (s)	% Original Points	Time Taken (s)
4	64	144	75	74
5	14	1159	13	690

Table 3.10 Percentage of original points and the time taken for (i) the ACM and (ii) the M-sequence method approximation of the *Sierpinski carpet.*

	ACM		Chaos game		M-sequence method	
	(No. of Its. = 2^k)		(No. of Its. = 2^k)		(No. of Its. = $2^k + k - 1$)	
k	% Orig. Pts.	Time (s)	% Orig. Pts.	Time (s)	% Orig. Pts.	Time (s)
12	32	215	51	78	54	64
13	31	427	32	156	33	129

Table 3.11 Percentage of original points and the time taken for (i) the ACM, (ii) the chaos game and (iii) the M-sequence method approximations of the *dragon*.

	ACM		Chaos game		M-sequence method	
	(No. of Its. = 9^k)		(No. of Its. = 9^k)		(No. of Its. = $9^k + k - 1$)	
k	% Orig. Pts.	Time (s)	% Orig. Pts.	Time (s)	% Orig. Pts.	Time (s)
3	54	25	91	16	99	13
5	9.4	2070	9.3	1380	9.3	1350

Table 3.12 Percentage of original points and the time taken for (i) the ACM, (ii) the chaos game and (iii) the M-sequence method approximations of the *Peano curve*.

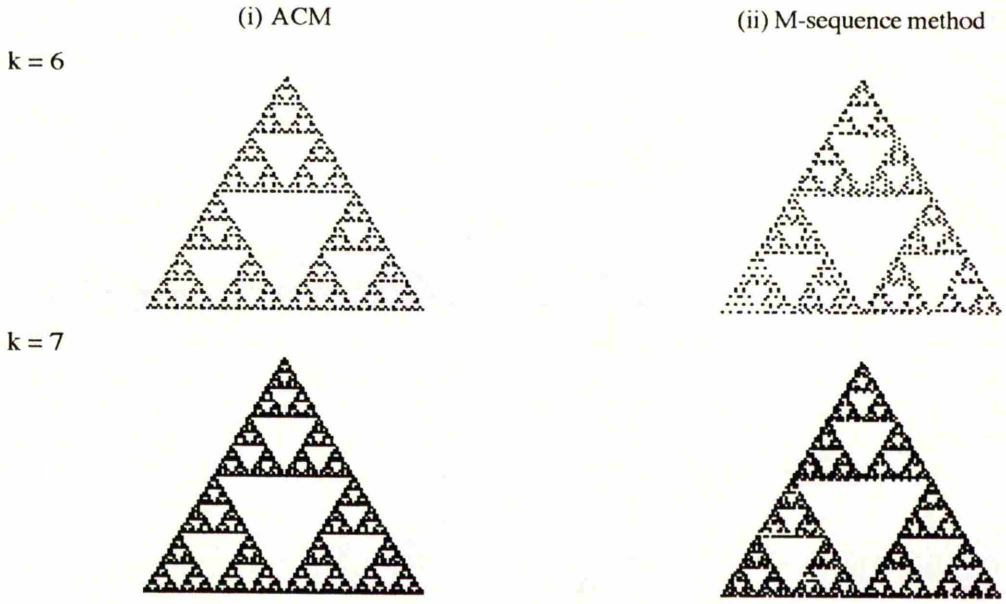


Figure 3.5 Comparison of the approximations of the *Sierpinski gasket* produced by (i) the ACM and (ii) the M-sequence method.

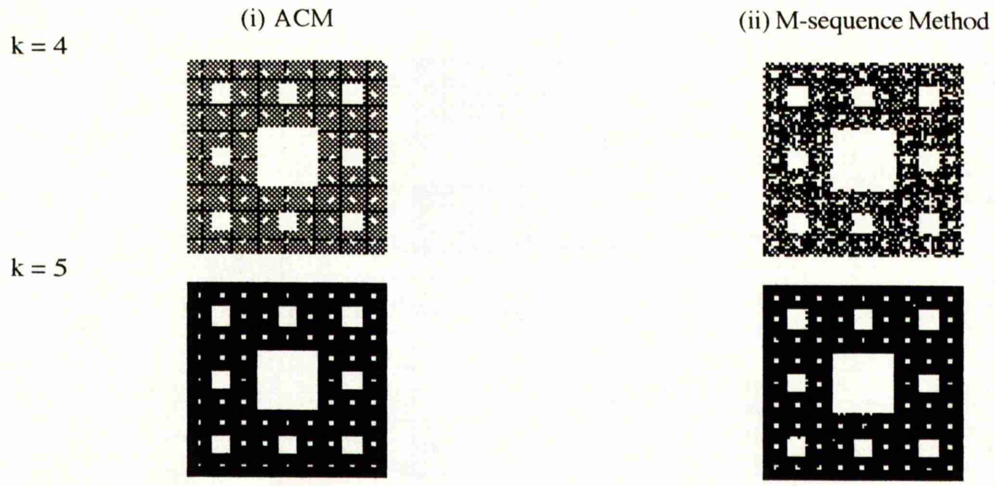


Figure 3.6 Comparison of the approximations of the *Sierpinski carpet* produced by (i) the ACM and (ii) the M-sequence method.

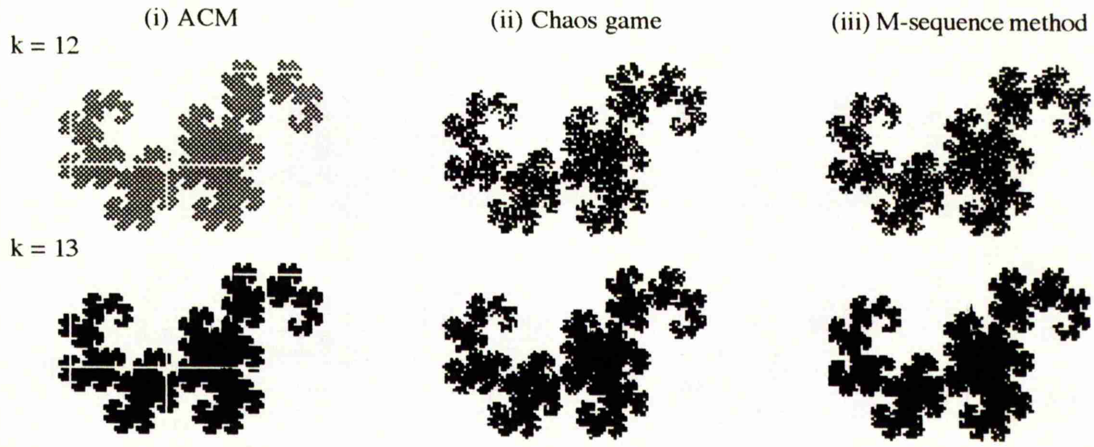


Figure 3.7 Comparison of the approximations of the *dragon* produced by (i) the ACM, (ii) the chaos game and (iii) the M-sequence method.

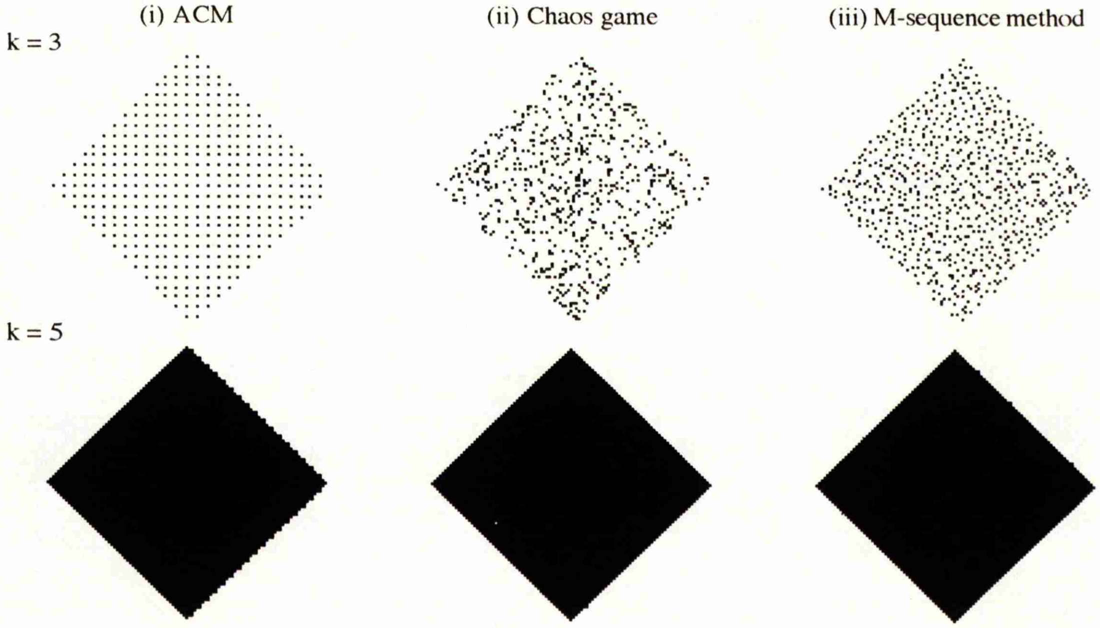


Figure 3.8 Comparison of approximations of the *Peano curve* produced by (i) the ACM, (ii) the chaos game and (iii) the M-sequence method.

The above results confirm that the M-sequence method is significantly faster than the ACM. For example, in less than the time taken for the ACM to produce an approximation of the dragon, $\mathcal{A}_\varepsilon$, where $\varepsilon \leq (1/\sqrt{2})^{12}.\mathrm{diam}(\mathcal{A})$, the M-sequence method can produce an approximation which is δ-dense in $\mathcal{A}$ where $\delta \leq (1/\sqrt{2})^{13}\mathrm{diam}(\mathcal{A})$.

For IFS with only a few affine transformations (N = 2, 3), the ACM produces the highest percentage of original points for small values of k. However, as k is increased, the difference between the ACM and the M-sequence method decreases and they produce a similar percentage of original points. In fact for large enough k, the M-sequence method plots significantly more original points (cf dragon k = 12). Even with relatively small k, the M-sequence method is clearly superior to the ACM for IFS with several affine transformations (N = 8, 9). The Sierpinski carpet (Peano curve) is made up of approximately 4500 (5550) different pixels. Hence, as k = 5 involves over 32,700 (59,000) iterations, it is inevitable that all three methods will produce a very small percentage of original points.

Finally, the images produced must be compared. For the Sierpinski gasket, the ACM and the M-sequence method produce results of the same standard while, as expected, the chaos game approximation (not shown) is very 'fuzzy'. The adaptive cut approximation of the dragon has two horizontal and two vertical white lines. This is not a fault of the printer - the equivalent pixels are unlit on the screen. These lines could be wrongly interpreted as a feature of the dragon attractor. Using ACM with k = 3, the Peano image produced is very disappointing. In fact the chaos game produces a higher percentage of original points than the ACM for the Peano curve (k = 3, 4, 5) and the dragon (k = 12, 13). Moreover, the ACM approximation of the Peano curve with k = 4 (see Section 7.1.2) has white horizontal and vertical lines running through it. The M-sequence method and the chaos game produce images which correctly approximate the dragon and the Peano curve. Recall, we stated previously that in general well-distributed points provide a better approximation of the attractor. On first inspection of the Sierpinski carpet (k = 4), it is tempting to say the adaptive cut image is superior for just this reason. However, on closer examination, there are a number of horizontal and vertical lines on the image. Again, this gives us an incorrect impression of the final image. The performance of the ACM appears to depend on the choice of x_0. For example, in the case of the dragon, using a different x_0, the ACM produces an approximation of the attractor without the white lines. This is investigated further in Chapter 7. We note that taking k sufficiently large ensures that the ACM approximation is correct. For example, with k = 5, the ACM approximation of the Peano curve is correct.

The ACM computes the code of N^k composite affine transformations and this is time consuming. In contrast the M-sequence method is very fast. Further, M-sequences as opposed to RNG's, ensure that the RIA plots well distributed points. This reduces redundancy. So the M-sequence method combines the speed of the chaos game with the accuracy of the ACM.

Attractor	map	a	b	c	d	e	f
Sierpinski gasket (Fig. 3.5)	w_1	1/2	0	0	1/2	0	0
	w_2	1/2	0	0	1/2	1/4	$1/4\sqrt{3}$
	w_3	1/2	0	0	1/2	1/2	0
Sierpinski carpet (Fig. 3.6)	w_1	1/3	0	0	1/3	0	0
	w_2	0	-1/3	1/3	0	1/3	1/3
	w_3	1/3	0	0	1/3	0	2/3
	w_4	0	1/3	-1/3	0	1/3	1
	w_5	0	1/3	-1/3	0	1/3	1/3
	w_6	1/3	0	0	1/3	2/3	0
	w_7	0	-1/3	1/3	0	1	1/3
	w_8	1/3	0	0	1/3	2/3	2/3
Dragon (Fig. 3.7)	w_1	1/2	1/2	-1/2	1/2	0	0
	w_2	-1/2	1/2	-1/2	-1/2	1	0
Peano curve (Fig. 3.8)	w_1	1/3	0	0	1/3	0	0
	w_2	0	-1/3	1/3	0	1/3	0
	$w_{\pounds}$	1/3	0	0	1/3	1/3	1/3
	w_4	0	1/3	-1/3	0	2/3	1/3
	w_5	-1/3	0	0	-1/3	2/3	0
	w_6	0	1/3	-1/3	0	1/3	0
	w_7	1/3	0	0	1/3	1/3	-1/3
	w_8	0	-1/3	1/3	0	2/3	-1/3
	w_9	1/3	0	0	1/3	2/3	0

Table 3.13 Code for various uniform IFS.

In this Chapter we considered only IFS's with uniform probabilities. The main objects were to discuss a sufficient requirement of good drivers and define optimal sequences and the optimal sequence method.

The first method of obtaining optimal sequences was related to graph theory and although it always guarantees a solution it is computationally expensive. We extended our study to the finite field F_N $(N = p^r)$ and illustrated that M-sequences (with an additional initial zero) are in fact optimal sequences. We illustrated that M-sequences exist over structures other than finite fields. These were used in the M-sequence method for uniform

IFS with $N \neq p^r$ transformations. We defined a new class of latin squares. In the future we hope to prove that for any $N \neq p^r$ and any k, an (N x N) k-recurrent latin square exists. Such results will be of interest in many other areas of research (see Chapter 4).

Finally, several examples illustrate the superiority of the M-sequence method compared to the chaos game. The M-sequence method is shown to be in some respects better than the ACM, one of the best non-RIA methods. This comparison is studied further in Chapter 7.

This M-sequence method need not be restricted to uniform IFS. As suggested by Hoggar [16], it may be possible to adapt the M-sequence method for non-uniform probabilities. Alternatively, optimal length sequences containing the required addresses for any IFS may be able to be generated. This is studied in Chapters 5 and 6.

Chapter 4 Some results on the existence of M-sequences

In Section 3.3.2, we introduced far rings. To recap, the structure $(Z_N, +, .)$ is a *far ring* provided

(i) $(Z_N, .)$ is a loop. i.e. conditions (4.1) and (4.2) hold for all a, b in Z_N.

Equations a.x = b and y.a = b, each have exactly one solution (4.1)

1 is the identity : 1.a = a = a.1. (4.2)

(ii) $(Z_N, +)$ is the additive group of integers modulo N.

In fact, the multiplication table for the far ring $(Z_N, +, .)$ is a latin square. In this chapter, we study far rings and the associated latin squares. We again use {0, 1, ..., N-1} to denote the elements of $(Z_N, +, .)$. But zero is not an annihilator. Indeed, 0.x = 0 for all x in Z_N, contradicts condition (4.1). Further, the multiplication '.' is not distributive over addition. For a.b = a.(b + 0) = a.b + a.0, implies 0 = a.0 (for all a). Notice (4.1) gives the cancellation law: a.x = b = a.z implies x = z.

Consider the kth-order linear recurrence

$$\sigma_n = a_1.\sigma_{n-1} + a_2.\sigma_{n-2} + \ldots + a_k.\sigma_{n-k} \mod N \qquad (4.3)$$

defined over the far ring $(Z_N, +, .)$. For N = 6, we previously gave examples of far rings and the associated coefficients a_1, ..., a_k so that the sequence $\{\sigma_n\}$ has period $N^k - 1$ (see Section 3.3.2). We extended the use of the term *M-sequence* to mean any linear recurring sequence meeting this bound, over whatever structure (originally over finite fields only). We said in the case of far rings that the corresponding latin squares were *k-recurrent* (see Definition 3.34).

Below, we give some initial results on far rings and the associated loops (or latin squares) [14]. We study two specific forms of latin squares and try to determine for what values of k (if any) they are k-recurrent. We give an example of a structure which is neither a far ring nor a finite field over which M-sequences exist. In Section 4.4, we list some possible conjectures.

4.1 Far rings and k-recurrent latin squares

For the fixed set of coefficients a_1, ..., a_k and any set of initial values σ_0, ..., σ_{k-1}, the period of the sequence $\{\sigma_n\}$ defined by (4.3), can not exceed N^k - this being the maximum number of distinct k-tuples which can occur in such a sequence before a repetition is unavoidable. Further, since any k-tuple may be taken as the starting set, $\{\sigma_0, \ldots, \sigma_{k-1}\}$, every k-tuple is present in exactly one sequence (for a fixed set of coefficients) and the period lengths of the different sequences produced sum to N^k.

As our underlying interest in far rings is to determine the nature of the associated k-recurrent latin squares, we make the following definition.

Definition 4.1 For the far ring (Z_N, +, .), the *period profile* of the coefficient k-tuple (a_1, ..., a_k) is a list (in descending order) of the period lengths of the different sequences produced from (4.3) (as noted above these lengths sum to N^k).

The *period profile of the multiplication table* (or latin square) is a set of N^k period profiles ; one for each of the coefficient k-tuples (a_1, ..., a_k).

We computed the period profile of many different loops and now make the related conjecture.

Conjecture 4.2 Over any far ring (Z_N, +, .), the maximum possible period of the linear recurring sequence (4.3) is $N^k - 1$.

Example 4.3 There are exactly four far rings of order four. The corresponding latin squares are given in Table 4.1 below. All four define groups. A, B, C are the cyclic group of order 4 while D is K_4, the Klein 4-group. The corresponding period profiles for k = 2 are given in Table 4.2.

Note Since the multiplication table borders are in standard order 1, 2, ..., N - 1, 0, we may omit them. In other words, we give the equivalent latin squares.

A:	1	2	3	0	B:	1	2	3	0	C:	1	2	3	0	D:	1	2	3	0
	2	0	1	3		2	1	0	3		2	3	0	1		2	1	0	3
	3	1	0	2		3	0	2	1		3	0	1	2		3	0	1	2
	0	3	2	1		0	3	1	2		0	1	2	3		0	3	2	1

Table 4.1 The four multiplication tables (latin squares) of order 4.

We hope to prove the following conjecture.

Conjecture 4.4 For every positive integer $N \neq p^r$ and every integer $k \geq 2$, there exists an N x N k-recurrent latin square.

Proving Conjecture 4.4 should result in a method to obtain a k-recurrent latin square and the corresponding M-sequence coefficients a_1, ..., a_k for any $N \neq p^r$, $k \geq 2$.

Coefficients	Sequences over far ring of Table A (with period length in brackets) see Table 4.1	Period Profiles
00	110 (3); 123 (3); 132 (3); 200 (3); 330 (3); 2 (1)	3^5,1
01	120322 (6); 102300 (6); 133 (3); 1 (1)	6^2,3,1
02	1302 2320 (8); 112103 (6); 3 (1); 0 (1)	8,6,1^2
03	113 100 301 220 2 (13); 233 (3)	13,3
10	122100 (6); 232030 (6); 133 (3); 1 (1)	6^2,3,1
11	112310 (6); 130322 (6); 220 (3); 0 (1)	6^2,3,1
12	11312 00323 30102 (15); 2 (1)	15, 1
13	132 230 020 (9); 1103 (4); 12 (2); 3 (1)	9,4,2,1
20	131 032 230 (9); 11202 (5); 3 (1); 0 (1)	9,5,1^2
21	11323 30201 21003 (15); 2 (1)	15,1
22	122 030 023 133 2 (13); 110 (3)	13,3
23	12320 01022 13033 (15); 1 (1)	15,1
30	113003 (6); 10220 (5); 233 (3); 12 (2)	6,5,3,2
31	11030 13122 32002 (15); 3 (1)	15,1
32	123 033 100 (9); 1322 (4); 20 (2); 1 (1)	9,4,2,1
33	1120 3210 (8); 133023 (6); 2 (1); 0 (1)	8,6,1^2

Coefficients	Sequences over far ring of Table B (with period length in brackets) see Table 4.1	Period Profiles
00	1230 3320 (8); 110213 (6); 2 (1); 0 (1)	8,6,1^2
01	12002 33010 31322 (15); 1 (1)	15,1
02	133100 (6); 22030 (5); 112 (3); 23 (2)	6,5,3,2
03	113 003 210 (9); 1223 (4); 20 (2); 3 (1)	9,4,2,1
10	122 100 203 (9); 1330 (4); 23 (2); 1 (1)	9,4,2,1
11	112310 (6); 130332 (6); 220 (3); 0 (1)	6^2,3,1
12	120102 (6); 223003 (6); 113 (3); 3 (1)	6^2,3,1
13	11030 23313 20012 (15); 2 (1)	15,1
20	130 010 322 023 3 (13); 112 (3)	13,3
21	122320 (6); 100302 (6); 113 (3); 3 (1)	6^2,3,1
22	110 (3); 123 (3); 132 (3); 200 (3); 330 (3); 2 (1)	3^5,1
23	1203 1022 (8); 133230 (6); 1 (1); 0 (1)	8,6,1^2
30	11321 20030 22310 (15); 3 (1)	15, 1
31	11020 32300 13312 (15); 2 (1)	15,1
32	122 103 130 (9); 23320 (5); 1 (1); 0 (1)	9,5,1^2
33	112 322 010 021 3 (13); 330 (3)	13,3

Table C :- Each coefficient pair has profile 6^2,3,1

Table D :- Coefficients 00, 02, 20, 22 have profile 3^5,1; the remainder have profile 6^2,3,1

Table 4.2 Period profiles for 2nd order linear recurrences over the far rings with Tables A, B, C, D of Table 4.1 and the corresponding sequences for A, B.

2	1	3	0
1	3	0	2
0	2	1	3
3	0	2	1

Table 4.3

3-recurrent latin square.

Note None of the four possible latin squares of order 4, which possess identities, (i.e. A, B, C, D of Table 4.1) are 3-recurrent. However if we remove condition (4.2) so that (Z_N, .) is a quasigroup (but not a loop), then the latin square of Table 4.3 , with borders 0123, defining the multiplication '.', over the structure (Z_N, +, .) is 3-recurrent. For example, taking the coefficients $a_1a_2a_3 = 030$ results in an M-sequence.

Similarly, for $N \neq p^r$ and some integer $k \geq 2$, it may be that no (N x N) latin square corresponding to a loop is k-recurrent. Thus we might be forced to remove condition (4.2) in order to prove Conjecture 4.4.

Definition 4.5 The quasigroups (G, .) and (H, *) are *isotopic* if there exists an ordered triple of bijections $\theta, \phi, \psi : G \rightarrow H$ such that for all x, y in G,

$$\psi(x.y) = \theta(x) * \phi(y).$$

If $\theta = \phi = \psi$, then the quasigroups are *isomorphic*.

Definition 4.6

(a) We define the following operations which may be performed on a multiplication table or latin square.

(i) *transposition* - interchanging rows for columns,

(ii) taking the *row conjugate* - let J be the permutation sending each element of the column border to the element in the same row of column j. Then in the row conjugate the corresponding permutation map is J^{-1},

(iii) taking the *column conjugate* - as for (ii) above but with the roles of row and column reversed.

(b) *Conjugates* of a multiplication table (or latin square) are formed by performing any sequence of the operations (i), (ii), (iii) above.

Example 4.7

(i) Observe that the loops A, B, C, D of Table 4.1 are unchanged by transposition. Thus multiplication over each of the corresponding far rings is commutative.

(ii) It is easily verified that for the loop B of Table 4.1, the J maps are in order :- identity, (12)(03), (1320), (1023) so that the row conjugate has rows 1203, 2130, 3012, 0321.

(iii) Loop D, its row conjugate and its column conjugate (also its transpose - see (i)) are all identical.

A natural question to ask is :- Do far rings have the same period profiles if their associated loops are

(1) isotopic,

(2) isomorphic,

(3) conjugate?

By considering latin squares A, C of Table 4.1, we immediately see that isomorphism does not imply equal period profiles. Then, as isomorphism is a special case of isotopism, this gives NO to (1) and (2) above.

For each of loops A, B, C, we compared the period profile ($k = 2, 3$ only) of the loop itself, with its row and column conjugates. For $k = 2, 3$, A, its row conjugate and its column conjugate have equal profiles. Similarly for B. However, for C, only the loop and its column conjugate have equal profiles. This requires further investigation but it will not be pursued at present - the row and column conjugates of A, B, C being without an identity do not define multiplication over a far ring.

We wish to define an isomorphism of far rings, incorporating both addition and multiplication, such that (4.4) holds.

$$\text{Far rings are isomorphic} \Leftrightarrow \text{their period profiles are equal.} \tag{4.4}$$

To determine whether far rings are isomorphic, it is not desirable to have to compute their period profiles and so we seek a simple test. We make the following definition related to condition (4.4).

Definition 4.8 The far rings $(Z_N, +, .)$ and $(Z_N, +, *)$ are *isomorphic* if there exists a pair of permutations (ψ, ϕ) of Z_N such that,

(i) $\psi(1) = 1$,

(ii) $\phi(\alpha.a + \beta.b) = \psi(\alpha) * \phi(a) + \psi(\beta) * \phi(b)$ for all α, β, a, b in Z_N. (4.5)

Theorem 4.9 Isomorphic far rings have the same period profiles.

Proof We must prove if (ψ, ϕ) is an isomorphism between the far rings $(Z_N, +, .)$ and $(Z_N, +, *)$, then the coefficient k-tuples $a_1, ..., a_k$ and $\psi(a_1), ..., \psi(a_k)$ have the same period profiles for any integer $k \geq 2$.

We begin with $k = 2$, and suppose that the coefficient pair (a_1, a_2) with the multiplication'.' produce a sequence $\{\sigma_n\}$ of period t. That is, with addition mod N and subscripts mod t, we have $\sigma_n = a_1.\sigma_{n-1} + a_2.\sigma_{n-2}$. Then by (4.5),

$$\phi(\sigma_n) = \psi(a_1)*\phi(\sigma_{n-1}) + \psi(a_2)*\phi(\sigma_{n-2}),$$

and, since ϕ is bijective, the coefficients $\psi(a_1)$, $\psi(a_2)$ with the multiplication '*' give the sequence $\{\phi(\sigma_n)\}$ of period t. Moreover, the bijection ψ defines the bijection of pairs $(a_1, a_2) \rightarrow (\psi(a_1), \psi(a_2))$ so that the two far rings have the same period profile.

The result for $k \geq 2$ follows by induction. It suffices to illustrate by taking the step from $k = 2$ to $k = 3$. Consider the recurrence with coefficients a_1, a_2, a_3 and to simplify notation, let $\chi\beta\alpha\delta$ denote four successive terms of the sequence. Then

$\delta = a_1.\alpha + a_2.\beta + a_3.\chi$

$\Rightarrow \quad \delta = a_1.\alpha + 1.(a_2.\beta + a_3.\chi)$ 1 being the identity for '.',

$\Rightarrow \quad \phi(\delta) = \psi(a_1) * \phi(\alpha) + \psi(1) * \phi(a_2.\beta + a_3.\chi)$ by (4.5),

$\Rightarrow \quad \phi(\delta) = \psi(a_1) * \phi(\alpha) + \phi(a_2.\beta + a_3.\chi)$ since $\psi(1) = 1$, an identity for '*',

$\Rightarrow \quad \phi(\delta) = \psi(a_1) * \phi(\alpha) + \psi(a_2) * \phi(\beta) + \psi(a_3) * \phi(\chi)$ by (4.5).

Thus if the coefficients a_1, ..., a_k generate a sequence $\{\sigma_n\}$ of period t under the multiplication '.', then the coefficients $\psi(a_1)$, ..., $\psi(a_k)$ generate the sequence $\{\phi(\sigma_n)\}$ of period t under the multiplication '*'. We conclude that isomorphic far rings have the same period profiles for all $k \geq 2$.

Theorem 4.9 meets condition (4.4) in the forward direction. Theorem 4.11 below, concerns (4.4) in the opposite direction. However, since the result requires that we already have candidate maps ψ,ϕ, it is only a partial converse of Theorem 4.9.

Notation 4.10 Let ψ,ϕ be permutations of Z_N and let FR_1, FR_2 be far rings of order N. We say that (ψ,ϕ) *preserves k-profiles* from FR_1 to FR_2 provided ϕ maps a sequence defined over FR_1 by a kth-order linear recurrence with coefficients a_i $(1 \leq i \leq k)$ to the sequence over FR_2 given by a (kth-order) recurrence with coefficients $\psi(a_i)$ $(1 \leq i \leq k)$.

Theorem 4.11 Let ψ,ϕ permutations of Z_N and let FR_1, FR_2 be far rings of order N with respective multiplication '.', '*'. Suppose that $\psi(1) = 1$ and that (ψ,ϕ) preserves 2-profiles from FR_1 to FR_2. Then (ψ,ϕ) is a far ring isomorphism $FR_1 \rightarrow FR_2$.

Proof We have $\psi(1) = 1$. Thus to prove that (ψ,ϕ) is a far ring isomorphism, it remains to show that condition (4.5) holds for all ordered 4-tuples α, β, a, b in Z_N.

Consider a fixed 4-tuple (α,β,a,b) and the associated linear recurrence of order 2. Then since a, b must appear as successive terms of some sequence $\{\sigma_n\}$ with the coefficient pair α, β, we have $\alpha.a + \beta.b = c$ for some element c in Z_N. Further ϕ maps this sequence $\{\sigma_n\}$ to the sequence $\{\phi(\sigma_n)\}$ over FR_2 with coefficients $\psi(\alpha)$, $\psi(\beta)$ so that $\psi(\alpha)*\phi(a) + \psi(\beta)*\phi(b) = \phi(c) = \phi(\alpha.a + \beta.b)$ - this is condition (4.5).

Finally the above argument is valid for every 4-tuple over Z_N and so the proof is complete.

Example 4.12 By referring to Table 4.2, notice that the far rings FRA and FRB with multiplications '.', '*' given by latin squares A and B (of Table 4.1) respectively, have the same period profiles for $k = 2$. Is it possible to find an isomorphism (ψ,ϕ): FRA $\rightarrow$ FRB -

we suppose there is one and find out what it could be. For ease of calculation, we consider the period profiles which apply to as few coefficient pairs as possible and within these we pick out the sequences of least period.

(1) *Profile* $3^5,1$ - This is the period profile of the coefficient pairs 00 and 22 over FRA and FRB respectively and of no others. Thus $\psi(0) = 2$. Further, over both far rings, the corresponding period 1 sequence is 22..., so that $\phi(2) = 2$. Now by considering the sequences of period 3, we may conclude that the sequence 200... over FRA must be mapped, by ϕ, to the sequence 200... over FRB, giving $\phi(0) = 0$.

(2) *Profile* $9,5,1^2$ - This occurs for the coefficient pairs 20 over FRA, 32 over FRB and no where else. Thus $\psi(2) = 3$. Moreover, since ψ is bijective, we must have $\psi(3) = 0$. Hence $\psi = (1)(230)$. Comparing the corresponding sequences of period 1, we may deduce $\phi(3) = 1$. Finally, since the map ϕ is bijective, we have $\phi(1) = 3$ so that $\phi = (0)(2)(13)$.

(3) We conclude that if there is an isomorphism of far rings (ψ,ϕ): FRA → FRB, then $\psi = (1)(234)$ and $\phi = (0)(2)(13)$. Applying Theorem 4.11 systematically shows that (ψ,ϕ) is indeed an isomorphism. Further, by Theorem 4.9, the two far rings FRA, FRB have the same period profiles for $k = 3, 4, \ldots$.

Theorem 4.13 Let ψ,ϕ be permutations of Z_N with $\psi(1) = 1$ and $\phi(0) = 0$ and let FR_1, FR_2 be far rings of order N. If for some integer $m > 2$, (ψ,ϕ) preserves m-profiles from FR_1 to FR_2, then (ψ,ϕ) is a far ring isomorphism from FR_1 to FR_2.

Proof We may induct down from m to 2 to show that condition (4.5) holds for all α,β,a,b in Z_N. However, as before, it is sufficient to illustrate the reduction from 3 to 2. Again, let '.', '*' be multiplication over FR_1 and FR_2 respectively. Then, over FR_1, if (ψ,ϕ) preserves 3-profiles, we have

$\Rightarrow \quad c = \alpha.a + \beta.b$

$\Rightarrow \quad c = \alpha.a + \beta.b + 1.0$ — 1 is the identity of '.',

$\Rightarrow \quad \phi(c) = \psi(\alpha)*\phi(a) + \psi(\beta)*\phi(b) + \psi(1)*\phi(0)$ — (ψ,ϕ) preserves 3-profiles,

$\Rightarrow \quad \phi(c) = \psi(\alpha)*\phi(a) + \psi(\beta)*\phi(b)$ — $\psi(1) = 1, \phi(0) = 0$.

The result follows by Theorem 4.11.

Definition 4.14

(i) The *composition* of the far ring isomorphisms $(\psi,\phi) : FR_1 \to FR_2$, $(\psi',\phi') : FR_2 \to FR_3$, is the isomorphism $(\psi'\psi,\phi'\phi) : FR_1 \to FR_3$,

(ii) The *inverse* of the isomorphism (ψ,ϕ): $FR_1 \rightarrow FR_2$ is itself an isomorphism and it is given by $(\psi,\phi)^{-1} = (\psi^{-1},\phi^{-1}) : FR_2 \rightarrow FR_1$,

(iii) If the multiplications '.', '*' for the isomorphic far rings FR_1, FR_2 coincide then $FR_1 = FR_2$ and (ψ,ϕ) is an *automorphism* of FR_1. The automorphisms of FR_1 form a group, $Aut(FR_1)$.

Remark 4.15 If FR_1, FR_2 are isomorphic far rings, then the number of isomorphisms between them equals both $|Aut(FR_1)|$ and $|Aut(FR_2)|$.

Proof Suppose there is at least one isomorphism $f : FR_1 \rightarrow FR_2$. Then, we have the bijections $Aut(FR_1) \leftarrow$ {isomorphisms from FR_1 to FR_2} $\rightarrow Aut(FR_2)$, given by $g^{-1}f \leftarrow g \rightarrow fg^{-1}$.

4.2 A study of two different structures of far rings

Below, we consider two constructions of far rings and discuss whether the corresponding latin squares are k-recurrent.

4.2.1 The first category - FRL_N

Notation 4.16 Let L_N be the latin square of order N where the (i, j)th entry is (i + j - 1) mod N. Also, let FRL_N denote the far ring $(Z_N, +, .)$ where multiplication '.' is defined by L_N.

Observation 4.17 Consider the kth-order linear recurrence (4.3) defined over the far ring FRL_N. Since i.j = (i + j - 1) mod N (for all i, j $\in$ {0, 1, ..., N-1}), the recurrence may be rewritten as

$$\sigma_n = (a_1 + \sigma_{n-1} - 1) + (a_2 + \sigma_{n-2} - 1) + \ldots + (a_k + \sigma_{n-k} - 1) \bmod N$$
$$\sigma_n = \Sigma^k_{i=1}\sigma_{n-i} + \Sigma^k_{i=1}(a_i - 1) \bmod N$$
$$\sigma_n = \Sigma^k_{i=1}\sigma_{n-i} + \text{diff} \bmod N \qquad (4.6)$$

where diff $\in$ {0, 1, ..., N-1} and it may be determined by $\Sigma^k_{i=1}(a_i - 1)$.

Note For a given k and N, the far ring FRL_N will have at most N different period profiles - one corresponding to each value the term diff may take.

We give some elementary observations concerning the sequence (4.6) before making a conjecture based on the period profiles of various far rings FRL_N.

Lemma 4.18 Let N be any even integer greater than 2. Then the latin square L_N is not 2-recurrent.

Proof With $\sigma_0 = \sigma_1 = N/2 - \text{diff}$, the sequence $\{\sigma_n\}$ given by (4.6) has period 3 and so prevents the possibility of an M-sequence with any starting set.

Observations 4.19 Let N denote an even integer which is not a prime power. Then, we have the following.

(i) If diff = 0, (4.6) simplifies to $\sigma_n = \Sigma_{i=1}^{k}\sigma_{n-i}$ mod N which, as previously observed (Lemma 3.28), can never produce an M-sequence when $N \neq p^r$.

(ii) If diff is even and k = 3, then sequence (4.6) can never be an M-sequence. The recurrence will always produce two distinct sequences of period 1, namely -diff/2... and N/2 - diff/2... .

(iii) If diff = 1 - k and $k \geq 2$, then with $\sigma_0 = \ldots = \sigma_{k-2} = 1$, $\sigma_{k-1} = N/2 + 1$, a sequence of period k + 1 will be produced.

(iv) If diff = N/2 and k = 4 and with $\sigma_i = 0$ $(0 \leq i \leq 3)$, the sequence (4.6) has period 5.

Conjecture 4.20 For any positive integer $N \neq p^r$ and any $k \geq 2$, the latin square L_N is not k-recurrent.

Example 4.21 Recall, with diff = 0, the kth-order linear recurrence over FRL_N simplifies to $\sigma_n = \Sigma_{i=1}^{k}\sigma_{n-i}$ mod N. In fact, if $N = p^r$, this corresponds to the kth-order linear recurrence, $\sigma_n = \Sigma_{i=1}^{k}\sigma_{n-i}$, over the finite field F_N, with associated characteristic polynomial $f(x) = x^k - x^{k-1} - \ldots - x - 1$. Then if f(x) is a primitive polynomial over F_N, the recurrence will produce an M-sequence. For example, the polynomials $x^2 - x - 1$ and $x^6 - x^5 - x^4 - x^3 - x^2 - x - 1$ are primitive over the finite field Z_3 and so M-sequences of order 2, 6 will be obtained from (4.6) where diff is zero. This illustrates that we can not extend Conjecture 4.20 to include integers which are a prime power. We computed the period profiles of sequence (4.6) with N = 3 , k = 2, 6 and diff = 0, 1, 2. The period profile in each case was $N^k - 1, 1$.

Example 4.22 Below, we give the latin square L_6 and the period profiles (k = 2) for each possible diff. Notice that L_6 is not 2-recurrent (see Conjecture 4.20 above).

(i)

1	2	3	4	5	0
2	3	4	5	0	1
3	4	5	0	1	2
4	5	0	1	2	3
5	0	1	2	3	4
0	1	2	3	4	5

(ii)

diff	Sequences over far ring FRL_6 (with the period in brackets)	Period Profiles
0	123521341505543145325101 (24); 24044202 (8); 303 (3); 0 (1)	24,8,3,1
1	124102304544320342140500 (24); 35331511 (8); 522 (3); 5 (1)	24,8,3,1
2	305125343321523103545501 (24); 24220400 (8); 411 (3); 4 (1)	24,8,3,1
3	423221041205243445025401 (24); 53551311 (8); 300 (3); 3 (1)	24,8,3,1
4	541323345143503121105301 (24); 42440200 (8); 525 (3); 2 (1)	24,8,3,1
5	542504302122340324520100 (24); 35155313 (8); 414 (3); 1 (1)	24,8,3,1

Table 4.4 (i) Latin Square L_6 and (ii) the sequences and period profiles for k = 2 over the far ring FRL_6.

4.2.2 The second category - FRM_N

Notation 4.23 Let N be any positive integer such that $N + 1 = p^r$ for some prime p. Consider the multiplication over the loop $(Z_{N+1}, .)$ where the multiplication '.' is performed mod (N + 1). By deleting the row and column corresponding to multiplication involving zero, we obtain a latin square with elements 1, 2, ..., N such that 1 is the identity. We shall denote this latin square M_N. Further, let FRM_N be the far ring $(Z_N, +, .)$ with addition mod N and multiplication '.' defined by M_N.

Note Equivalently this far ring involves (i) addition mod N and (ii) multiplication mod (N + 1), where 1, 2, ..., N denotes the elements.

Example 4.24 Below we illustrate how the latin square M_6 may be obtained.

(i)

1	2	3	4	5	6	0
2	4	6	1	3	5	0
3	6	2	5	1	4	0
4	1	5	2	6	3	0
5	3	1	6	4	2	0
6	5	4	3	2	1	0
0	0	0	0	0	0	0

⇒ Remove the seventh row
and column to form ⇒

(ii)

1	2	3	4	5	6
2	4	6	1	3	5
3	6	2	5	1	4
4	1	5	2	6	3
5	3	1	6	4	2
6	5	4	3	2	1

Table 4.5 (i) Multiplication mod 7, (ii) M_6 obtained from Table (i) by deleting the appropriate rows.

At present, we have proved few results. However, below we give some of the observations which we have made.

Example 4.25 We computed the period profiles of every possible coefficient k-tuple over FRM_N for some integers $N = p - 1$, $k \geq 2$. The table below shows the number of M-sequences that exist for these cases.

k	Number of M-sequences over the far ring:-			
	FRM_4	FRM_6	FRM_{10}	FRM_{12}
2	4	0	6	4
3	0	10	8	2
4	6	10	*	*

Table 4.6 The number of M-sequences over the far ring FRM_N, N = 4, 6, 10 ,12 (* not computed).

Observation 4.26 For any $N = p^r - 1$, the kth-order recurring sequence (4.3) defined over the far ring FRM_N can never be an M-sequence if any one of the following holds for $i = 1, 2, \ldots, k$:-

(i) $a_i = 1$,

(ii) $a_i = N$,

(iii) k is even, N is even and $a_i = N/2$.

Proof (i) (4.3) simplifies to $\sigma_n = \Sigma^k_{i=1}\sigma_{n-i} \bmod N$ - this can never produce an M-sequence (Lemma 3.27, $N \neq 2$).

For (ii), (iii) set $\sigma_i = 1$ $(1 \leq i \leq k)$ to produce a sequence of period $k + 1$.

4.3 Other structures over which M-sequences exist

We originally defined far rings so that we could obtain M-sequences when $N \neq p^r$. However, as the example below illustrates, this does not prevent us from considering other possible structures over which M-sequences may be found.

Example 4.27 Consider the structure $(Z_6, +, .)$ with elements $\{1, 2, 3, 4, 5, 0\}$ where addition is performed mod 6 and the multiplication '.' is given by Table 4.7(i). Notice this has both an identity *and* a zero, so is not a far ring. For some coefficient k-tuples $(a_1, \ldots, a_k)$, the initial set $\sigma_0\sigma_1\sigma_2$ never reappears in $\{\sigma_n\}$. For example, if $(a_1, a_2) = (1, 0)$, the recurrence is simply $\sigma_n = \sigma_{n-1}$ and with $\sigma_0 = x$, $\sigma_1 = y$, $x \neq y$, we get the sequence xyyy... so that xy never reappears. However, there are coefficient k-tuples for which the corresponding kth-order recurring sequence is a valid M-sequence. These are given in Table 4.7(ii) for k = 2, 3, 4.

(i)

1	2	3	4	5	0
2	3	4	5	1	0
3	4	5	1	2	0
4	5	1	2	3	0
5	1	2	3	4	0
0	0	0	0	0	0

(ii)

k	2	3	4
M-sequence coefficients $a_1, ..., a_k$	31 41	NIL	4021 4051 4221 4551

Table 4.7 (i) Multiplication '.' and (ii) the M-sequence coefficients over (Z_6, +, .) for k = 2, 3, 4.

This example suggests a new area of study in the search for M-sequences.

4.4 Problems and conjectures

In addition to proving Conjectures 4.2, 4.4 and 4.20, we hope to investigate the following.

Problem 4.28 Find a short test for isomorphic far rings based on inspecting latin squares.

Problem 4.29 Reduce to a minimum the number of 4-tuples we must check to verify (4.5).

Problem 4.30 Investigate isomorphism and k-recurrence for far rings with $N \geq 6$. Also, find a simple test for k-recurrence and a way of obtaining the corresponding M-sequence coefficients $a_1, ..., a_k$.

Conjecture 4.31 If two far rings based on the same Z_N have the same period profiles, then they are isomorphic.

Problem 4.32 Investigate the relationship between the far ring (Z_N, +, .) and any structure (Z_N, +, *) where the latin square for '*' is a conjugate of the latin square for '.', (Z_N, *) is a quasigroup (but not necessarily a loop) and addition is performed mod N.

Problem 4.33 Study structures other than far rings over which M-sequences exist (see Example 4.27).

Problem 4.34 Discover whether amongst latin squares which produce M-sequences, there are characteristics reflected in the IFS attractors based on these sequences.

Chapter 5 Optimal sequences for non-uniform iterated function systems - The simplest case

We now turn our attention to non-uniform iterated function systems. That is, the IFS $\{w_{1-N}\}$ where the corresponding ratios s_i are not all equal. With non-uniform IFS, the shortcomings of the chaos game become more apparent. The main problem being how to calculate the associated probabilities p_i. Ideally the probabilities would be set to values which maximise the efficiency of algorithm. That is, so that the points plotted are well distributed over the attractor $\mathcal{A}$, with little wasted in going over the same areas more than once. This reduces the time taken to achieve an approximation of $\mathcal{A}$ to a given accuracy. However, in spite of refinements in the choice of p_i, much computation is still wasted [28] (See Section 2.8 for the usual method of computation for p_i [4], and Section 7.2 for an alternative method [28].).

In this chapter we give some initial results about addresses and optimal sequences. We study a simple model non-uniform IFS and suggest the format of a general algorithm to produce optimal sequences [15, 27]. We compare this optimal sequence method to both the chaos game and the ACM. In Chapter 6, we extend some of these results to more complicated models and discuss how these algorithms may be modified to produce associated optimal sequences for more complicated models.

5.1 Addresses and optimal sequences

To begin, we consider iterated function systems with just two maps w_a, w_b which scale uniformly in the ratios r_a, r_b respectively. That is, $|w_\tau(\mathbf{x}) - w_\tau(\mathbf{y})| = r_\tau|\mathbf{x} - \mathbf{y}|$ for $\tau \in \{a, b\}$ and vectors $\mathbf{x}$, $\mathbf{y}$. Further since we are only concerned with non-uniform iterated function systems we must have $r_a \neq r_b$. For uniform IFS's see Chapter 3.

Notation 5.1 Let $\sigma = \sigma_1 \ldots \sigma_k$, $\sigma_i \in \{a, b\}$ be a sequence of subscripts. Then $w_\sigma = w_{\sigma_1} \ldots w_{\sigma_k}$, the corresponding product of maps, scales uniformly in the ratio

$$\rho(\sigma) = \prod r_{\sigma_i} \text{ with } \rho(\sigma\tau) = \rho(\sigma)\,\rho(\tau),$$

where τ denotes any sequence of subscripts.

Note When applying w_σ to a point x, w_{σ_k} is applied first, followed by $w_{\sigma_{k-1}}$, ..., and lastly w_{σ_1}.

For completeness, we restate Definition 2.21 for the IFS $\{w_a, w_b\}$ above, using Notation 5.1.

Definition 5.2 Let $\mathcal{A}$ be the attractor of the IFS $\{w_a, w_b\}$ of ratio $s = \max\{r_a, r_b\}$. Then, the corresponding list of addresses consists of all sequences $\sigma = \sigma_1 \ldots \sigma_k$ over $\{a, b\}$ such that $\rho(\sigma_1 \ldots \sigma_k) \leq \varepsilon$ and $\rho(\sigma_1 \ldots \sigma_{k-1}) > \varepsilon$. The addresses divide $\mathcal{A}$ into subsets $\mathcal{A}_\sigma = w_{\sigma_1} \ldots w_{\sigma_k}(\mathcal{A})$ with $\mathrm{diam}(\mathcal{A}_\sigma) = \rho(\sigma)\, \mathrm{diam}(\mathcal{A}) \leq \varepsilon$.

To simplify notation, choose the units so that the *diameter* of $\mathcal{A}$ is one and let $\varepsilon > 0$. Then (analogously to the uniform IFS of Chapter 3), to obtain our approximation of $\mathcal{A}$, we aim to partition $\mathcal{A}$ into subsets of diameter not exceeding ε and to produce a sequence of points x_0, x_1, ... with a point in each subset. Again, these subsets are related to the addresses of $\mathcal{A}$. As noted in Definition 5.2, the subsets will be of the form $\mathcal{A}_\sigma = w_{\sigma_1} \ldots w_{\sigma_k}(\mathcal{A})$, with $\mathrm{diam}(\mathcal{A}_\sigma)$ equal to $\rho(\sigma)$. We achieve this partition as follows. Starting with $\mathcal{A} = \mathcal{A}_a \cup \mathcal{A}_b$ (Theorem 2.11), we recursively replace $\mathcal{A}_\sigma$ by $\mathcal{A}_\sigma = \mathcal{A}_{\sigma a} \cup \mathcal{A}_{\sigma b}$ whenever $\rho(\sigma) > \varepsilon$. Since w_a, w_b are both contractive, this process must eventually terminate and we will have a list of subsets

$$\mathcal{A}_\sigma = w_{\sigma_1} \ldots w_{\sigma_k}(\mathcal{A}) \text{ with } \mathrm{diam}(\mathcal{A}_\sigma) = \rho(\sigma) \leq \varepsilon \ .$$

We refer to $\mathcal{A}_\sigma$ as the subset with *address* $\sigma = \sigma_1 \ldots \sigma_k$.

Recall, we may represent this construction by an *address tree.* Starting with a node labelled 1 to represent $\mathcal{A}$ itself, we recursively branch at the node σ (representing partition $\mathcal{A}_\sigma$) whenever $\rho(\sigma) > \varepsilon$. We label such branches a, b and their respective new nodes σa, σb. When the tree is complete, each end node τ satisfies $\rho(\tau) \leq \varepsilon$. A simple example of an address tree is given in Figure 5.1

Example 5.3 The address tree for the IFS $\{w_a, w_b\}$ where the ratios r_a, r_b respectively, satisfy $r_a^2 = r_b$, $r_b^2 = \varepsilon$ is given in Figure 5.1. Observe that the sequence bb, usually written b^2, is an end node and hence an address, because $\rho(bb) = s_b^2 = \varepsilon$ whereas $\rho(b) = s_b > \varepsilon$. On the other hand the sequence ba is not an address since $\rho(ba) = r_b r_a > \varepsilon$ and so we must branch beyond this node. Then, reading off the end nodes we have the complete list of addresses:- a^4, a^3b, a^2b, aba, ab^2, ba^2, bab, b^2. We must accept apparent overkill in the sense that ab^2 and b^2 are both included. This is a logical consequence of the subsets covering the whole of $\mathcal{A}$ as well as being small enough.

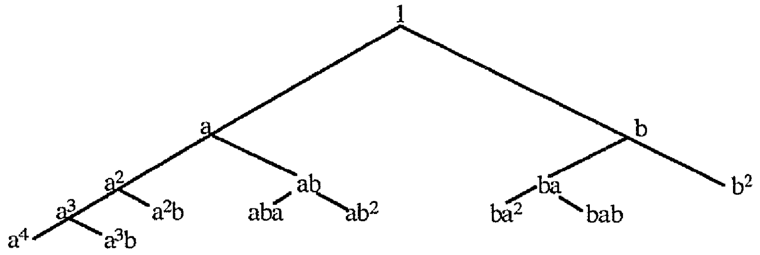


Figure 5.1 The address tree of Example 5.3 where $r_a^2 = r_b$ and $r_b^2 = \varepsilon$.

We now have the required partition of $\mathcal{A}$. But how can we guarantee to plot a point in each subset of this partition? Let $x_0 \in \mathcal{A}$. Then, as we observed in Chapter 3, by plotting $w_{\sigma_1} \ldots w_{\sigma_k}(x_0)$ for each address σ, the ACM meets this condition. Alternatively, for the uniform IFS $\{w_{1\text{-}N}\}$, the RIA when driven by a sequence of minimal length containing every address (i.e. every k-digit sequence over $\{1, ..., N\}$ for some integer k) meets this requirement - the optimal (or M-) sequence method. Can we implement a similar idea for non-uniform IFS? Again, our approach is to dovetail the addresses into an *optimal sequence*, that is a sequence of minimal length containing all the addresses, which we shall then use to drive the RIA. We begin with a simple example.

Example 5.4 We shall illustrate using addresses of Example 5.3. Consider the sequence S = babaaaabb. Notice that the address bab starts at the first digit, aba at the second, ..., and so on up to b^2 beginning at the eighth of the nine symbols. The last digit, b, does not start an address. However, its presence is essential so that the address b^2 appears. Further, if each address is to have a distinct starting point, they cannot dovetail into a sequence of less than nine digits. Hence, S is a sequence of minimal length containing all the addresses. Let $x_0 \in \mathcal{A}$ and suppose we use S (in reversed order) to drive the RIA for the IFS $\{w_a,\ w_b\}$. Then the following points will be plotted

$$w_b(x_0) \in \mathcal{A}_b,$$

$$w_bw_b(x_0) \in \mathcal{A}_{bb},$$

$$w_aw_bw_b(x_0) \in \mathcal{A}_{abb},$$

$$w_aw_aw_bw_b(x_0) = w_aw_aw_b(w_b(x_0)) \in \mathcal{A}_{aab} \text{ since } w_b(x_0) \in \mathcal{A},$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$w_bw_aw_bw_a{}^4w_b{}^2(x_0) \in \mathcal{A}_{bab} \text{ since } w_a4w_b2(x_0) \in \mathcal{A}.$$

Thus, we get a sequence of points x_1, ..., x_9, one from each line of the displayed argument, which includes a point in each set of the partition. The redundant point x_1 is a consequence of continuing S until the last address b^2 is completed. We now make the following definition.

Definition 5.5 An *optimal sequence* for the addresses of an IFS with attractor $\mathcal{A}$ is a sequence which contains every address and has length

$$\text{(total number of addresses)} + \text{(length of the shortest address)} - 1.$$

Theorem 5.6 Let the sequence S = ...rst of finite length k be an optimal sequence for the IFS attractor $\mathcal{A}$ and let $x_0 \in \mathcal{A}$. Then the RIA, driven by the reverse of S, produces the finite set of points x_1, ..., x_k where

$$x_1 = w_t(x_0),\ x_2 = w_s w_t(x_0),\ x_3 = w_r w_s w_t(x_0), \ldots$$

and a point is in each subset of the partition. If the diameter of each subset is no more than ε, then the set of plotted points $\{x_1, \ldots, x_k\}$ are ε-dense in $\mathcal{A}$.
Note This is the *optimal sequence method.*

We now give some results concerning addresses and optimal sequences which offer insight into the formation of optimal sequences. Then, a more specific model will be studied.

Lemma 5.7 (a) Every node in the address tree is reached by a unique path from 1,
(b) if $\sigma = \sigma_1 \ldots \sigma_k$ is a node, then so are σ_1, $\sigma_1\sigma_2$, ..., $\sigma_1 \ldots \sigma_{k-1}$,
(c) σ is a node if and only if $\sigma\tau$ is an address for some (possibly empty) sequence τ.

Proof (a) By construction the graph (address tree) we form is a *tree* in that (i) for every node σ there is a path from 1 to σ and hence a path between any two nodes via 1, and (ii) there are no 'cycles' i.e. no paths in which, as we proceed, we come to a node previously met. It follows from (ii) that the path from 1 to σ is unique since the existence of two such paths implies a cycle.
(b) The unique path from 1 to $\sigma = \sigma_1 \ldots \sigma_k$ contains the cited nodes.
(c) If $\sigma\tau$ is an address then it is of course a node and so by (b) σ is a node. Conversely, if σ is a node then the branching continues until end nodes $\sigma\tau$ are reached.

Theorem 5.8 We have the following equivalent facts.
(a) No point of a sequence can be the start of two addresses.

(b) No address is the rightwards extension of another.
(c) If σ is an address, then $\sigma\tau$ is not, for all non-empty sequences τ.

Proof We shall prove the result in form (c). Let σ be an address and consider a non-empty sequence τ. Then σ is an end node so by construction the address tree ends there and $\sigma\tau$ is not even a node. Notice that we are implicitly using Lemma 5.7(a) : the unique path from 1 to $\sigma\tau$, if it existed, would have to pass through σ.

Corollary 5.9 In an optimal sequence S, each address starts in exactly one place. The last address to appear is one of least length, say L and all but the last L - 1 digits start an address.

Lemma 5.10 Let σ, τ be sequences. Then $\sigma a\tau$ is an address if and only if $\sigma b\mu$ is an address for some sequence μ.

Proof Suppose $\sigma a\tau$ is an address. Then σ is a branching node and σb a node. Either σb is an end node or the construction continues and some $\sigma b\mu$ is an end node. On becoming an end node, it is an address. For the converse, simply reverse the argument.

We shall study the addresses and the consequent construction of optimal sequences of the non-uniform IFS $\{w_a, w_b\}$ (with respective ratios r_a, r_b), adding the following assumption.

Assumption 5.11 For some positive integers $n \geq 2$, $r \geq 1$, we have

$$r_a^n = r_b \qquad r_b^r = \varepsilon. \tag{5.1}$$

Note We omit n = 1, since uniform IFS's were covered in Chapter 3. The equalities (5.1) will be our working hypothesis to facilitate calculations and to find algorithms which produce optimal sequences. However, in practice, it is sufficient for (5.1) to hold with the equality sign replaced by the inequality '≤'. If they actually hold for real numbers lower than the integers r, n, the optimal sequence corresponding to model (5.1) will tend to result in a more accurate approximation of the attractor than expected. In any case we soon introduce an easier concept of *weights* and we will work with these.

Theorem 5.12 If b occurs r times in an address σ, then σ ends in b.

Proof Suppose that b occurs r times in σ and σ ends in a. Then $\sigma = \tau a$ for some sequence τ. Further b must occur r times in τ. Then $\rho(\tau) \leq \varepsilon$ - making σ an invalid address (Recall any sequence $\sigma = \sigma_1 \ldots \sigma_k$ is an address if and only if $\rho(\sigma) \leq \varepsilon$ and $\rho(\sigma_1 \ldots \sigma_{k-1}) > \varepsilon$ - see Definition 5.2).

Definition 5.13 The *weight* wt(σ) of a sequence σ is

$$\mathrm{wt}(\sigma) = n \times (\text{power of b in } \sigma) + (\text{power of a in } \sigma)\ .$$

By power, we mean the number of times a or b appears in σ. Thus wt(a) = 1, wt(b) = n, wt(bab) = 2n+1, $\mathrm{wt}(a^s b^t) = s + nt$ and $\mathrm{wt}(\sigma\tau) = \mathrm{wt}(\sigma) + \mathrm{wt}(\tau)$ if τ is a second sequence.

Note The fact that in terms of ratios n a's may be traded for one b suggests this definition. Deleting an 'a' from a sequence reduces the weight by 1, whereas deleting a 'b' reduces it by n.

We may now use weight to determine addresses.

Corollary 5.14 Let τ be a sequence and let x denote a or b. Then τx is an address if and only if

$$\mathrm{wt}(\tau x) \geq nr \text{ and } \mathrm{wt}(\tau) < nr.$$

Proof By Assumption 5.11, $\rho(\sigma) \leq \varepsilon \Leftrightarrow$ (power of a in σ)/n + (power of b in σ) $\geq r \Leftrightarrow$ (power of a in σ) + n x (power of b in σ) $\geq nr \Leftrightarrow \mathrm{wt}(\sigma) \geq nr$, for any sequence σ. So the two conditions of Definition 5.2 for an address become those of Corollary 5.14 when expressed in terms of weights.

Example 5.15 Consider the case r = 4, general $n \geq 2$. Then $\mathrm{wt}(a^{3n-1}b^2) = 5n\text{-}1 \geq 4n$ while $\mathrm{wt}(a^{3n-1}b) = 4n - 1$ and so by Corollary 5.14, $a^{3n-1}b^2$ is an address. However, after reordering the symbols to give the sequence b^2a^{3n-1}, we have $\mathrm{wt}(b^2a^{3n-1}) = 5n - 1$ and $\mathrm{wt}(b^2a^{3n-2}) = 5n - 2$, so that b^2a^{3n-1} is not an address.

Corollary 5.16 A sequence σ is an address if and only if it satisfies one of

(a) σ ends in a and has weight $\mathrm{wt}(\sigma) = nr$,

(b) σ ends in b and $nr \leq \mathrm{wt}(\sigma) \leq nr + (n\text{-}1)$.

Notation 5.17 Let,

(a) $(x_1, \ldots, x_k)$ denote the address $a^{x_1}ba^{x_2}b \ldots a^{x_{k-1}}ba^{x_k}$ ending in a where $x_j \geq 0$ for $j = 1, \ldots, k - 1$, $x_k \geq 1$ and $\sum_{j=1}^{k} x_j = n(r - k + 1)$,

(b) $(x_1, \ldots, x_k.)$ denote the address $a^{x_1}ba^{x_2}b \ldots a^{x_{k-1}}ba^{x_k}b$ ending in b where $x_j \geq 0$ for $j = 1, \ldots, k$ and $n(r - k) \leq \sum_{j=1}^{k} x_j \leq n(r - k) + (n - 1)$.

Example 5.18 Here we use Corollary 5.14, to list the addresses for the case r = 2, n = 5. We note the weight 'threshold' nr = 10, wt(a) = 1, wt(b) = 5. Then the addresses are:-

a^5b	a^6b	a^7b	a^8b	a^9b	a^{10}
a^4ba	a^3ba^2	a^2ba^3	aba^4	ba^5	
a^4b^2	a^3bab	a^2ba^2b	aba^3b	ba^4b	
a^3b^2	a^2bab	aba^2b	ba^3b		
a^2b^2	$abab$	ba^2b			
ab^2	bab				
b^2					

Using Notation 5.17, we relist the addresses for r = 2, n = 5 (see Lemma 5.19) :-

(5.)	(6.)	(7.)	(8.)	(9.)	(10)
(4, 1)	(3, 2)	(2,3)	(1,4)	(0,5)	
(4, 0.)	(3, 1.)	(2, 2.)	(1, 3.)	(0, 4.)	
(3, 0.)	(2, 1.)	(1, 2.)	(0, 3.)		
(2, 0.)	(1, 1.)	(0, 2.)			
(1, 0.)	(0, 1.)				
(0, 0.)					

Below, we prove some results on the type and number of addresses for the attractor $\mathcal{A}$ with $r \geq 1$, $n \geq 2$ - both crucial to the possible construction of optimal sequences.

Lemma 5.19 For any IFS satisfying Assumption 5.11 with $r \geq 1$, $n \geq 2$, the complete list of addresses is:-

(a) all addresses of the form $(x_1, \ldots, x_k)$; $k = 1, \ldots, r$,

(b) all addresses of the form $(x_1, \ldots, x_k.)$; $k = 1, \ldots, r$.

In order to derive an expression for the total number of addresses in the list of Lemma 5.19 and thus determine the length of a corresponding optimal sequence, we must first introduce some new notation and prove some auxiliary results.

Notation 5.20 For integers $k \geq 1$, $i \geq 0$, $\alpha \geq 0$ let,

(i) $\Sigma_k(\alpha) = \#\{(x_1, \ldots, x_k) : 0 \leq x_j \leq \alpha,\ \Sigma^k_{j=1} x_j = \alpha\}$,

(ii) $\Sigma^i{}_k(\alpha) = \#\{(x_1, \ldots, x_k) : 0 \leq x_j \leq \alpha,\ \Sigma^k_{j=1} x_j = \alpha,\ x_k = i\}$,

(iii) $\Sigma^{`}{}_k(\alpha) = \#\{(x_1, \ldots, x_k) : 0 \leq x_j \leq \alpha,\ \Sigma^k_{j=1} x_j = \alpha,\ x_k \neq 0\}$.

Example 5.21 For $\alpha \geq 0$, we have, $\Sigma_1(\alpha) = \#\{(x) : x = \alpha\} = 1$,
$\Sigma_2(\alpha) = \#\{(x_1, x_2) : 0 \leq x_1, x_2 \leq \alpha,\ x_1 + x_2 = \alpha\} = \alpha + 1$,
$\Sigma^i{}_2(\alpha) = \#\{(x_1, x_2) : 0 \leq x_1 \leq \alpha,\ x_2 = i,\ x_1 + x_2 = \alpha\} = 1$,
$\Sigma^{`}{}_2(\alpha) = \#\{(x_1, x_2) : 0 \leq x_1 \leq \alpha,\ 1 \leq x_2 \leq \alpha,\ x_1 + x_2 = \alpha\} = \alpha$.

The results of Lemma 5.22 (illustrated above) follow from the definitions of Notation 5.20 and by induction we have Lemma 5.23.

Lemma 5.22

(i) $\Sigma^i{}_k(\alpha) = \Sigma_{k-1}(\alpha - i)$, $\qquad (k \geq 2,\ i \geq 0,\ \alpha \geq 0)$

(ii) $\Sigma^0{}_k(\alpha) + \Sigma^1{}_k(\alpha) + \ldots + \Sigma^\alpha{}_k(\alpha) = \Sigma_k(\alpha)$, $\qquad (k \geq 1,\ \alpha \geq 0)$

(iii) $\Sigma_k(0) + \Sigma_k(1) + \ldots + \Sigma_k(\alpha) = \Sigma_{k+1}(\alpha)$. $\qquad (k \geq 1,\ \alpha \geq 0)$

Lemma 5.23

$$\sum_{i=1}^{n} (i)_k = \frac{(n)_{k+1}}{k+1} \quad \text{where } (n)_k = n(n+1) \ldots (n+k-1). \qquad (k \geq 1,\ \alpha \geq 0)$$

Theorem 5.24

$$\sum_{k+1} (\alpha) = \binom{k+\alpha}{k}. \qquad (k \geq 0,\ \alpha \geq 0) \qquad (5.2)$$

Proof The result is true for $k = 0$ and 1 by Example 5.21 above. Now assume that (5.2) holds for $k = w$ and consider $k = w + 1$,

$\Sigma_{w+2}(\alpha) = \Sigma^0{}_{w+2}(\alpha) + \Sigma^1{}_{w+2}(\alpha) + \ldots + \Sigma^{(\alpha)}{}_{w+2}(\alpha)$ $\qquad$ by Lemma 5.22(ii)

$\qquad = \Sigma_{w+1}(\alpha) + \Sigma_{w+1}(\alpha - 1) + \ldots + \Sigma_{w+1}(0)$ $\qquad$ by Lemma 5.22(i)

Hence the result follows by induction on k, after using the inductive hypothesis and Lemma 5.23.

Corollary 5.25 For $k \geq 1$, $i \geq 0$, $\alpha \geq 0$, we have

(i) $\displaystyle\sum^{(i)}_{k+1} (\alpha) = \binom{k+\alpha-i-1}{k-1}$, $\qquad$ (ii) $\displaystyle\sum^{`}_{k+1}(\alpha) = \binom{k+\alpha-1}{k}$.

Proof (i) follows from Lemma 5.22(i) and Theorem 5.24. For (ii) we have by Notation 5.20 that $\Sigma^{`}_{k+1}(\alpha) = \Sigma_{k+1}(\alpha) - \Sigma^{0}_{k+1}(\alpha)$, so the result follows from Theorem 5.24 and part (i).

Lemma 5.26 For $1 \leq k \leq r - 1$, the total number of addresses of the form $(x_1, \ldots, x_{k\bullet})$ is (see Notation 5.17(b))

$$\Sigma_{k+1}(n(r - k) + (n - 1)) - \Sigma_{k+1}(n(r - k) - 1).$$

Proof By Notation 5.17(b) and Notation 5.20(i), the number of addresses is $\Sigma_k(n(r - k)) + \Sigma_k(n(r - k) + 1) + \ldots + \Sigma_k(n(r - k) + (n - 1))$ (the arguments in the sum being consecutive integers), which equals $\Sigma_{k+1}(n(r - k) + (n - 1)) - \Sigma_{k+1}(n(r - k) - 1)$ by Lemma 5.22(iii).

Corollary 5.27 For any IFS with $r \geq 1$, $n \geq 2$ satisfying Assumption 5.11, the total number of addresses is

$$\sum_{k=0}^{r} \binom{n(r - k + 1) + k - 1}{k}. \qquad (5.3)$$

Proof We recall that, by Lemma 5.19, the addresses fall into two categories:-

(a) $(x_1, \ldots, x_k)$ for $k = 1, \ldots, r$. For $k = 1$ there is a unique address a^{nr} whilst for $2 \leq k \leq r$, the number of addresses is

$$\Sigma^{`}_{k}(n(r - k + 1)) = \binom{n(r - k + 1) + k - 2}{k - 1},$$

and the total number of addresses of type (a) may be written as,

$$1 + \sum_{k=1}^{r-1} \binom{n(r - k) + k - 1}{k}. \qquad (5.4)$$

(b) $(x_1, \ldots, x_{k\bullet})$ for $k = 1, \ldots, r$. By Lemma 5.26, the number of addresses of the form $(x_1, \ldots, x_{k\bullet})$ $(1 \leq k \leq r - 1)$ is $\Sigma_{k+1}(n(r - k) + (n - 1)) - \Sigma_{k+1}(n(r - k) - 1)$. When $k = r$ there are $\Sigma_{r+1}(n - 1)$ addresses. Hence the total number of type (b) addresses is

$$\sum_{k=1}^{r-1} \left[\binom{n(r - k + 1) + k - 1}{k} - \binom{n(r - k) + k - 1}{k}\right] + \binom{n + r - 1}{r}. \qquad (5.5)$$

Finally adding (5.4) and (5.5), we find the total number of addresses to be (5.3).

Recall, an optimal sequence, S, is a sequence of minimal length containing all addresses of an IFS attractor $\mathscr{A}$ (Definition 5.5). Then for every IFS satisfying model (5.1), the last address to appear in S is b^r and all but the last r - 1 digits start an address (see Corollary 5.9). Further, its exact length may be determined using equation (5.3) and Definition 5.5. We will now prove that such a sequence is cyclic, that is the last r - 1 symbols are identical to the first r - 1.

Theorem 5.28 Optimal sequences for model (5.1) are cyclic.

Proof Let $a_ib_ic_i$... denote the ith address in an optimal sequence S. Write the successive addresses as columns, say :-

Row 1	a_0	a_1	a_2	...	a_x	where $b_0 = a_1$
Row 2	b_0	b_1	b_2	...	b_x	$c_0 = b_1 = a_2$
Row 3	c_0	c_1	c_2	...	c_x	and so on.
.	.	.	.	...	.	

Let $\tau \in \{a, b\}$. Then we claim that τ appears in each of the first r rows the same number of times. This follows if switching any two of the first r entries of an address gives another address. Consider the three cases :-

(a) the address has length > r and ends in a which is not affected by the switch,

(b) the address has length > r and ends in b which is not affected by the switch,

(c) the address is b^r and so is unchanged .

By Corollary 5.16, the fact that the last symbol and the weight are unchanged, ensures that an address results in each case. Thus the symbol τ appears in each of rows 1, ..., r the same number of times.

Now consider rows 1, 2. We have $b_0 \ldots b_{x-1} = a_1 \ldots a_x$ and since τ appears the same number of times in each of the first r rows, it follows that $b_x = a_0$. Similarly, considering rows 1, 2, 3, we have $c_0 \ldots c_{x-1} = b_1 \ldots b_x$ and thus $c_x = b_0 = a_1$, and so on. Thus S is cyclic.

Note From now on, by an optimal sequence S, we mean a sequence of period length x (where x is equal to the total number of addresses), which when continued cyclically contains every address exactly once. i.e. $S = a_0 \ldots a_{x-1}$(repeated) where each symbol a_i is the start of a unique address $(0 \le i \le x - 1)$.

Example 5.29 For $r = 2$, $n = 5$ the sequence $S = a^{10}b^2a^4baba^3ba^2b$(repeated) has period length 26 and the reader is invited to verify that each of the first 26 symbols are the start of a unique address (see Example 5.18 for the list of addresses).

Later, we prove that if S is a cyclic optimal sequence then so is its reverse. However, we must firstly establish more about the construction of S.

5.2 The dovetailing graph

Initially, we used 'backtracking' to obtain a list of all optimal sequences for small values of r, n. This provided valuable insight into the possible construction of a general algorithm and allowed us to make checks on the inclusion of addresses. However, for larger values of r, n our backtracking program failed due to insufficient memory. We concluded that it was necessary to study theoretically the construction of optimal sequences.

Construction 5.30 Corresponding to type (a), (b) addresses of Corollary 5.16 there are two essential methods of dovetailing.

(a) *Runs* Let τ be a sequence such that $\mathrm{wt}(\tau) = nr - n$. Then the following *run* of addresses may be dovetailed as arrowed,

$$a^{n-1}\tau a \rightarrow a^{n-2}\tau a^2 \rightarrow \ldots \rightarrow a\tau a^{n-1} \rightarrow \tau a^n.$$

Note The weight of each address of the run is nr and they all end in a. They fit together into the sequence $a^{n-1}\tau a^n$ with distinct addresses starting at each of the first n symbols. These runs often form a sort of skeleton on which optimal sequences may be built by inserting other addresses in between.

(b) *Towers* An address $T = x_1 \ldots x_h\tau b$ such that each x_i $(1 \leq i \leq h)$ starts an address ending at the final b, τb is not itself an address, and $\mathrm{wt}(T) = nr + (n - 1)$, is called a *tower of height h.* There are two types of towers.

Type I $\quad a\ a \ldots a\ \tau\ b = a^n\tau b$, $(x_h = a)$ of height n

Type II $\quad a\ a \ldots a\ b\ \tau\ b = a^{h-1}b\tau b$, $(x_h = b)$ of height $h \leq n$.

In Example 5.18, the addresses where every symbol except the last is a, form the Type I tower a^9b with bottom address a^5b (here τ is empty). The other addresses ending in b, may be combined into the Type II towers a^4b^2, a^3bab, a^2ba^2b, aba^3b, ba^4b. Here τ is respectively empty, a, a^2, a^3, a^4. Notice ba^4b is the special case of a tower with a single address. To find the bottom of a tower, we repeatedly delete the leading 'a' whilst there is one and provided the weight still exceeds nr. The tower grouping is essential in any optimal sequence. That is, in every optimal sequence, the addresses of a tower appear within the tower and nowhere else.

Example 5.31 In the case r = 4, n = 5, constructions include :-
(a) the run $a^7ba^2ba \rightarrow a^6ba^2ba^2 \rightarrow \dots \rightarrow aba^2ba^7 \rightarrow ba^2ba^8$,
(b) the towers $a^{11}ba^3b$ and $a^{19}b$ with bottom addresses a^7ba^3b and $a^{15}b$ respectively.

We are now ready to prove :-

Theorem 5.32 If S is an optimal sequence for model (5.1) then so is its reverse.

Proof Notice that every symbol of S starts an address and every symbol has a successor and a predecessor (when S is continued cyclically). Recall, the addresses may be divided into :-
(a) those ending in a of weight nr exactly
(b) those ending in b of weight $nr \leq wt \leq nr + (n - 1)$.
Clearly every address of type (a) when reversed gives an address. Thus it remains only to consider addresses of type (b). Recall, these addresses are found in towers. We must show that within S, the type (b) addresses obtained on reversing S, themselves occur in a tower. We have,
(i) the tower $a^{nr-1}b$ - It is left as an exercise to prove that in any optimal sequence this tower must be found within the sequence $ba^{nr}b$. Hint :- consider what symbols can precede or follow the address a^{nr}. Then, since $ba^{nr}b$ is unchanged when reversed, the tower $a^{nr-1}b$ is still present.
(ii) a tower containing exactly k b's $(2 \leq k \leq r)$, namely $a^{x_1}ba^{x_2}b\dots a^{x_k}b$ where $\Sigma^k_{j=1}x_j = n(r - k) + (n - 1)$ - In any optimal sequence, the address $(x_1, \dots, x_{k\cdot})$ must occur in a tower where the top address has weight $nr + (n - 1)$. Thus to ensure the reversed sequence meets this condition, we must show that within S, the subsequence $a^{x_1}ba^{x_2}b\dots a^{x_k}b$ is followed by x_1 a's. Suppose to the contrary, we have within S,

$$a^{x_1}ba^{x_2}b\dots a^{x_k}ba^yb \qquad \text{where } y < x_1.$$

Then, this contains addresses $\beta = a^{x_2}b\dots a^{x_k}ba^yb$ with $\Sigma^k_{j=2}x_j + y < n(r - k) + (n - 1)$ duplicating β's appearance in the tower $a^xba^{x_3}b\dots ba^{x_k}ba^yb$ where $x = n(r - k) + (n - 1) - y - \Sigma^k_{j=3}x_j$. Hence result.

We now have some idea of how certain types of addresses must occur in an optimal sequence. To find such a sequence we must fit together these various addresses and subsequences. The *dovetailing graph* (see Definition 5.33) illustrates the possible constructions.

Definition 5.33 In seeking an optimal sequence for the attractor $\mathcal{A}$ of an IFS $\{w_a, w_b, w_c, ...\}$ we fit together various addresses, and partial sequences such as towers; the *dovetailing graph* Γ shows what may be so joined.

A *node* X of Γ is (represents) a sequence over {a, b, c,...} in which each of the first $x \geq 1$ symbols say and no others, starts an address contained in X. No address may appear twice in X and at least one must end at the last symbol of X.

A *directed branch* $X \to Y$ joins nodes X, Y if they may be dovetailed as for an optimal sequence. That is, so that the start of Y is one symbol on from the start of the last address in X. Then finding an optimal sequence is equivalent to finding a path through Γ which starts and ends at the same node (since optimal sequences are cyclic) and which includes every other node exactly once - an *optimal path.*

For every IFS satisfying model (5.1) with $r \geq 1$, $n \geq 2$, let $\Gamma_{r,n}$ denote the corresponding dovetailing graph.

In Sections 5.2.1, 5.2.2, we study in detail the dovetailing graph $\Gamma_{r,n}$ and the formation of optimal sequences for r = 2, 3 and general $n \geq 2$. Then we shall tackle the general case $r \geq 2$, $n \geq 2$ in Section 5.2.3. We omit the trivial case r = 1, with the unique optimal sequence $a^n b$.

5.2.1 The case $r = 2$, $n \geq 2$

5.34 The Dovetailing Graph $\Gamma_{2,n}$ With $r = 2$, $n \geq 2$, the dovetailing graph for model (5.1) is denoted $\Gamma_{2,n}$ and consists of the following.

Nodes There are $(n^2 + 5n + 2)/2$ addresses and these are completely represented by the following nodes :-

$(2n) = a^{2n}$,

$(n\text{-}1, 1), (n\text{-}2, 2), ..., (0, n) = a^{n-1}ba, a^{n-2}ba^2, ..., ba^n$,

$(2n\text{-}1.)$ = the tower $a^{2n-1}b$ to $a^n b$,

$(n\text{-}1,0.), (n\text{-}2, 1.), ..., (0, n\text{-}1.)$ = the towers $a^{n-1}b^2$ to b^2, $a^{n-2}bab$ to bab, ..., $ba^{n-1}b$.

Permitted branches Every node apart from (2n), (2n-1.) and (0, n) has exactly two possible successors and two possible predecessors.

(a) The compulsory branch :- $(0, n) \to (2n) \to (2n\text{-}1.)$. That is $ba^n \to a^{2n} \to a^{2n-1}b$.

(b) The possible successors to the node (x_1, x_2) depend on the value of x_1 ($x_1 + x_2 = n$, see Notation 5.17):-

(i) *Case $x_1 = 0$*

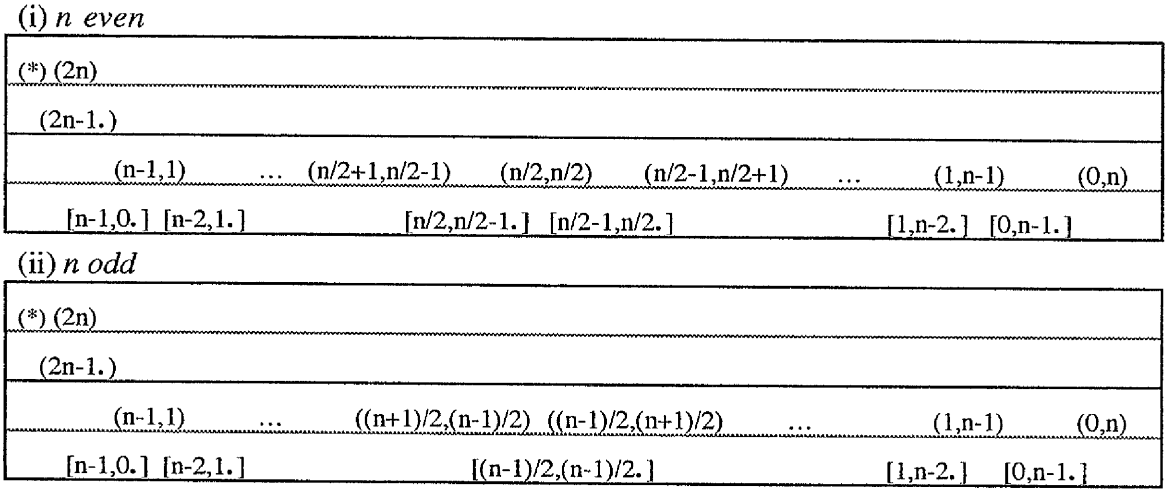
For node $(0, n)$ see Case (a) above

(ii) *Case $x_1 \geq 1$*

$$\begin{tikzcd}[row sep=small] & (x_1 - 1, x_2 + 1) \\ (x_1, x_2) \arrow[ur] \arrow[dr] & \\ & (x_1 - 1, x_2.) \end{tikzcd}$$

(c) The possible successors to the node $(x_1, x_2.)$ are :-

$$\begin{tikzcd}[row sep=small] & (x_2, x_1 + 1) \\ (x_1, x_2.) \arrow[ur] \arrow[dr] & \\ & (x_2, x_1.) \end{tikzcd}$$



(i) n even
(*) (2n)
(2n-1.)
(n-1,1) ... (n/2+1,n/2-1) (n/2,n/2) (n/2-1,n/2+1) ... (1,n-1) (0,n)
[n-1,0.] [n-2,1.] [n/2,n/2-1.] [n/2-1,n/2.] [1,n-2.] [0,n-1.]
(ii) n odd
(*) (2n)
(2n-1.)
(n-1,1) ... ((n+1)/2,(n-1)/2) ((n-1)/2,(n+1)/2) ... (1,n-1) (0,n)
[n-1,0.] [n-2,1.] [(n-1)/2,(n-1)/2.] [1,n-2.] [0,n-1.]

Figure 5.2 Dovetailing Graph $\Gamma_{2,n}$ for $n \geq 2$.

Definition 5.35 The dovetailing graph $\Gamma_{2,n}$ consists of two levels, each of which is itself split into two sublevels (see Figure 5.2).

Level 1 For reasons which become apparent when we study the general graph $\Gamma_{r,n}$, the top sublevel of level 1 contains only the node (2n) and is called the *1-run* while the second sublevel, called the *singleton*, consists of (2n - 1.)

Level 2 Again this level is divided into two, namely:-

(i) *2-runs* Following Construction 5.30(a), we group together the addresses (x_1, x_2) with common middle term b to form the 2-run

$$(n\text{-}1, 1) \rightarrow (n\text{-}2, 2) \rightarrow \ldots \rightarrow (1, n\text{-}1) \rightarrow (0, n).$$

(ii) *Pairs* The remaining addresses all end in b and must occur together in towers $(x_1, x_2.) = a^{x_1}ba^{x_2}b$ where $x_1 + x_2 = n - 1$ (Construction 3.30(b)). These towers may be dovetailed into pairs as :-

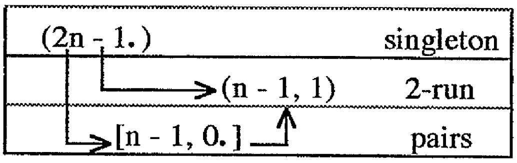
$$[x_1, x_2.] : (x_1, x_2.) \rightarrow (x_2, x_1.)$$

where [x_1, x_2.] is the sequence $a^{x_1}ba^{x_2}ba^{x_1}b$, dovetailing them. If n is odd, then we have the pair [x, x.] = a^xba^xb representing the unpartnered tower (x, x.) where x = (n - 1)/2.

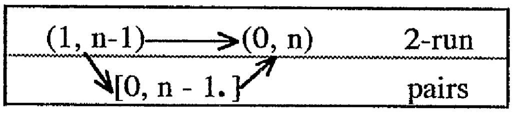
5.36 Constructing Optimal Sequences Recall, finding an optimal sequence is equivalent to finding an optimal path through the graph $\Gamma_{2,n}$. Since optimal sequences are cyclic, paths may begin at any node of $\Gamma_{2,n}$. However, for simplicity, we shall start at the 1-run node (2n). Then every optimal path we consider, must begin (2n) → (2n-1.), before going on to cover every node of level 2 exactly once and ending with (0, n) → (2n).

Theorem 5.37 After the singleton (2n - 1) of level 1, an optimal path must proceed along one of the two paths in (i) below. That is, straight to the first node of the 2-run or U-shaped inserting the pair [n - 1, 0.]. Then, from the 2-run node (1, n - 1), the path taken in (ii) must be of the opposite type to that taken in (i).

(i)



(ii)

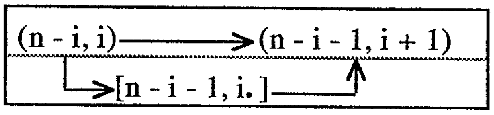


Proof It is easily verified that the paths in (i) and (ii) above are the only possible options which ensure that the towers occur in pairs. However, since an optimal path must include each node exactly once, the path in (ii) will involve the pair if and only if the pair (in reverse form) was not included in (i).

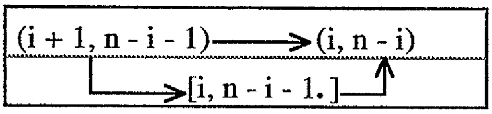
It is crucial to observe that (i) and (ii) in Theorem 5.37 involve the same tower nodes, (n - 1, 0.) and (0, n - 1.) but in reversed orders. If we work inwards from the path ends and use the same arguments as above, we have the following.

Corollary 5.38 When at the 2-run node (n - i, i) with 1 ≤ i ≤ (n - 1)/2, either one of the two paths in square (i) below must be taken. Moreover, the path in square (ii) following the 2-run node (i + 1, n - i - 1) must be of the opposite type to that taken in (i).

(i)



(ii)



Note If n is odd, then the pair [x, x.] = a^xba^xb where x = (n - 1)/2 has a unique position in any optimal path. That is:-

$$((n + 1)/2, (n - 1)/2) \rightarrow [(n - 1)/2, (n - 1)/2.\,] \rightarrow ((n - 1)/2, (n + 1)/2)$$

Example 5.39 Figure 5.3 shows the four different optimal paths through $\Gamma_{2,5}$. The corresponding optimal sequences are (i) $a^{10}b^2a^4baba^3ba^2b$, (ii) $a^{10}b^2a^4ba^2ba^3bab$, (iii) $a^{10}baba^3ba^2ba^4b^2$ (iv) $a^{10}ba^2ba^3ba^4b^2$.

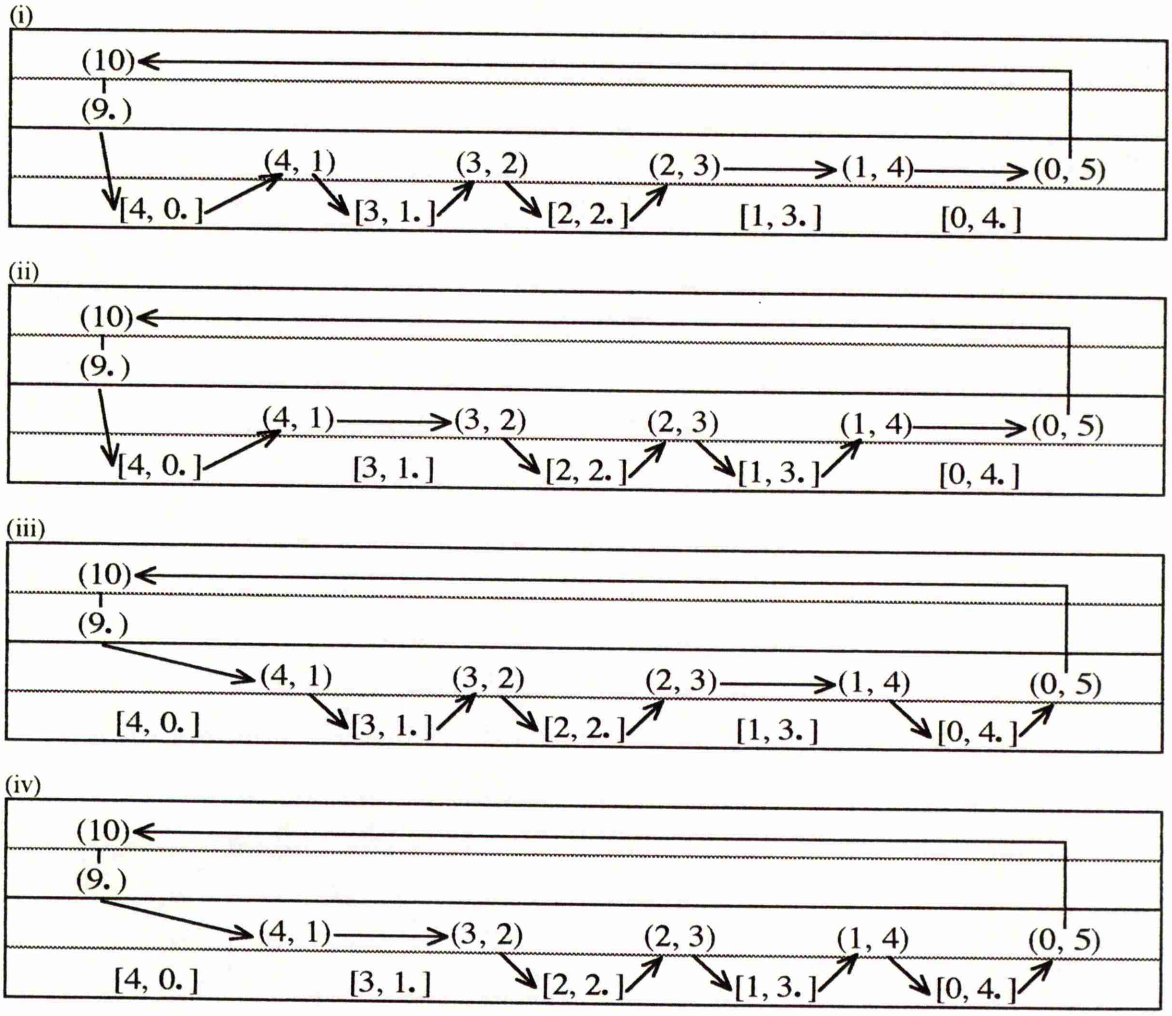


Figure 5.3 The four possible optimal paths through the dovetailing graph $\Gamma_{2,5}$.

Theorem 5.40 The number of distinct optimal sequences for model (5.1) with r = 2, n ≥ 2 is $2^{n \text{ div } 2}$.

Proof There are n div 2 distinct pairs of the form [x_1, x_2.] where $x_1 \neq x_2$, each of which must be inserted in exactly one of the two possible positions.

Note By distinct, we mean sequences which are not cyclically equivalent.

5.2.2 The case $r = 3$, $n \geq 2$

This case is much more complicated than $r = 2$ but the results and observations of the previous section provide a useful guide.

5.41 The Dovetailing Graph $\Gamma_{3,n}$ For this case the dovetailing graph $\Gamma_{3,n}$ consists of the following.

Nodes The $(n/6)(n + 2)(n + 13) + 1$ addresses are represented by the nodes listed below (using Notation 5.17).

$(3n) = a^{3n}$,

$(x_1, x_2) = a^{x_1}ba^{x_2}$ (with $x_1 + x_2 = 2n$ and $0 \leq x_1 \leq 2n - 1$),

$(x_1, x_2, x_3) = a^{x_1}ba^{x_2}ba^{x_3}$ (with $x_1 + x_2 + x_3 = n$ and $x_1, x_2 \geq 0$ but $x_3 \geq 1$),

$(3n - 1.)$ = the tower $a^{3n-1}b$ to $a^{2n}b$,

$(x_1, x_2.) = a^{x_1}ba^{x_2}b$ ($x_1 + x_2 = 2n - 1$ and $x_1, x_2 \geq 0$), a tower ending in $a^{n-x_2}ba^{x_2}b$ if $x_1 \geq n - 1$ (or equivalently $x_2 \leq n$) and ending in $ba^{x_2}b$ if $x_2 \geq n$.

$(x_1, x_2, x_3.) = a^{x_1}ba^{x_2}ba^{x_3}b$ ($x_1 + x_2 + x_3 = n - 1$ and $x_1, x_2, x_3 \geq 0$), a tower with bottom address $ba^{x_2}ba^{x_3}b$.

Permitted Branches Every node apart from $(3n)$, $(3n\text{-}1.)$ and $(0, 2n)$ has exactly two successors and two predecessors.

(a) The compulsory branch :- $(0, 2n) \rightarrow (3n) \rightarrow (3n\text{-}1.)$. That is, $ba^{2n} \rightarrow a^{3n} \rightarrow a^{3n-1}b$.

(b) The possible successors to the node (x_1, x_2) depend on the value of x_1 :-

(i) *Case $x_1 = 0$*

For node $(0, 2n)$ see (a) above

(ii) *Case $x_1 \geq 1$*

$$\begin{tikzcd} & (x_1 - 1, x_2 + 1) \\ (x_1, x_2) \arrow[ur] \arrow[dr] & \\ & (x_1 - 1, x_2.) \end{tikzcd}$$

(c) The possible successors to the node $(x_1, x_2.)$ are :-

(i) *Case $x_1 \leq n - 1$*

$$\begin{tikzcd} & (x_2, x_1 + 1) \\ (x_1, x_2.) \arrow[ur] \arrow[dr] & \\ & (x_2, x_1.) \end{tikzcd}$$

(ii) *Case $x_1 > n - 1$*

$$\begin{tikzcd} & (n - x_2 - 1, x_2, 1) \\ (x_1, x_2.) \arrow[ur] \arrow[dr] & \\ & (n - x_2 - 1, x_2, 0.) \end{tikzcd}$$

(d) There are two cases for the successors of the node (x_1, x_2, x_3) depending on the value of x_1:-

(i) *Case* $x_1 = 0$

$$\begin{tikzcd} & (x_2, x_3 + n) \\ (0, x_2, x_3) \arrow[ur] \arrow[dr] & \\ & (x_2, x_3 + n - 1.) \end{tikzcd}$$

(ii) *Case* $x_1 \geq 1$

$$\begin{tikzcd} & (x_1 - 1, x_2, x_3 + 1) \\ (x_1, x_2, x_3) \arrow[ur] \arrow[dr] & \\ & (x_1 - 1, x_2, x_3.) \end{tikzcd}$$

(e) The possible successors to the node $(x_1, x_2, x_3.)$ are :-

$$\begin{tikzcd} & (x_2, x_3, x_1 + 1) \\ (x_1, x_2, x_3.) \arrow[ur] \arrow[dr] & \\ & (x_2, x_3, x_1.) \end{tikzcd}$$

Definition 5.42 $\Gamma_{3,n}$ has three levels where the kth level is itself split into two sublevels, *k-runs* and *k-tuples* (k = 1, 2, 3) (see Figure 5.4).

Level 1 Analogously to the graph $\Gamma_{2,n}$, this level consists of the *1-run* (3n) and the *singleton* (3n-1.) and these nodes must occur in every optimal sequence as (3n) → (3n-1.).

Level 2 Again, following the pattern of $\Gamma_{2,n}$, we have:-

(i) *2-runs* We group together the addresses (x_1, x_2) to form the *2-run*:

$$(2n-1, 1) \rightarrow (2n-2, 2) \rightarrow \dots \rightarrow (1, 2n-1) \rightarrow (0, 2n).$$

(ii) *Pairs* All the towers with two b's are of the form $(x_1, x_2.)$ with $x_1 + x_2 = 2n - 1$. Then since none are of the form (x, x), each has a unique partner. Again, we write

$$[x_1, x_2.] : (x_1, x_2.) \rightarrow (x_2, x_1.),$$

where $[x_1, x_2.]$ represents the sequence $a^{x_1}ba^{x_2}ba^{x_1}b$ and insist that these towers occur only as pairs. However, if $x_1 > n-1$, the last address of the tower $(x_1, x_2.)$ is $a^{n-x_2}ba^{x_2}b$ and before the part $a^{x_2}b$ is reached there are an extra $n - x_2$ symbols, $a^{n-x_2-1}b$. We shall take these as the starting symbols of the 3-run $[x_2]$ defined below. The inclusion of this 3-run is essential to the dovetailing of towers.

Level 3 The two sublevels are:-

3-runs We group together the nodes (x_1, x_2, x_3) with common middle term $ba^{x_2}b$ as a 3-run in $[x_2] = a^{n-x_2-1}ba^{x_2}ba^{n-x_2}$, writing

$$[x_2] : (n-x_2-1, x_2, 1) \rightarrow (n-x_2-2, x_2, 2) \rightarrow \dots \rightarrow (0, x_2, n-x_2).$$

(ii) *Triples* The towers $(x_1, x_2, x_3.)$ may be dovetailed in triples as:-

$$[x_1, x_2, x_3.] : (x_1, x_2, x_3.) \rightarrow (x_2, x_3, x_1.) \rightarrow (x_3, x_1, x_2.)$$

where $[x_1, x_2, x_3.]$ represents the sequence $a^{x_1}ba^{x_2}ba^{x_3}ba^{x_1}ba^{x_2}b$ dovetailing them.

Level 1								
(3n)								
(3n-1.)								
Level 2	(2n-1,1)	...	(n+1,n-1)	(n, n)	(n-1,n+1)	...	(1,2n-1)	(0,2n)
	[2n-1,0.]	...	[n, n - 1.]	[n - 1, n.]		...	[0,2n-1.]	
Level 3	[0]	...	[n - 1]	[n - 1]		...	[0]	
	$[x_1,0,x_2.]$	...	$[x_1, n - 1, x_2.]$			...	$[x_1,0,x_2.]$	

Figure 5.4 Dovetailing Graph $\Gamma_{3,n}$ for $n \geq 2$.

5.43 Constructing Optimal Sequences We shall begin at the node (3n) and consider how an optimal path may be formed. Then the path begins (3n) → (3n-1.) before going on to lower levels. It will end with the branch (0, 2n) → (3n). From (3n-1.), an optimal path will go to the 2-run node (2n-1, 1) either directly or indirectly via the pair [2n - 1, 0.]. Observe, when at the 2-run node (2n - i , i) an optimal path will either go straight to the next 2-run node (2n - i - 1, i + 1) or insert the pair [2n - i - 1, i.] before proceeding to (2n - i - 1, i + 1) (i = 1, ..., n - 1, see Figure 5.5). However, since each pair must be included exactly once (either on the left or the right), the branch (2n - i, i) → (2n - i - 1, i + 1) can exist if and only if the pair [i, 2n - i - 1.] is inserted between (i + 1, 2n - i - 1) and (i, 2n - i). Then there are 2^n possibilities depending on which address of the pair is to be put first.

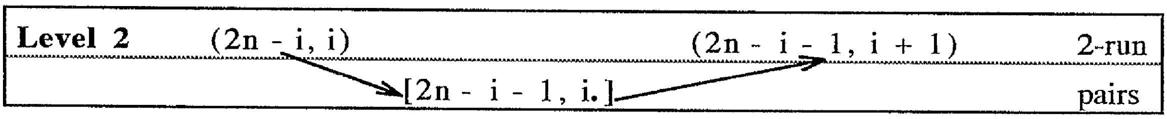


Figure 5.5 How to insert pairs in $\Gamma_{3,n}$.

But on inserting the pair $[x_1, x_2.]$ with $x_1 > n - 1$, we automatically include the 3-run $[x_2]$ (see Figure 5.9) , That is $[x_1, x_2.]$: $(x_1, x_2.)$ → $[x_2]$ → $(x_2, x_1.)$. While if the order is reversed to $[x_2, x_1.]$, still with $x_1 > n - 1$, then we have the dovetailing $(x_2 + 1, x_1)$ → $[x_2, x_1.]$ → $(x_2, x_1 + 1)$. Further, we may insert triples between the branches of the 3-run $[x_2]$ in level 3. For it is easily checked that the path $(x_1 + 1, x_2, x_3)$ → $[x_1, x_2, x_3.]$ → $(x_1, x_2, x_3 + 1)$ is the only way that the triple $[x_1, x_2, x_3.]$ can be proceeded or followed. However, just as the pairs fit together in either order, these triples fit in any of the three cyclically related orders, $[x_1, x_2, x_3.]$, $[x_2, x_3, x_1.]$ or $[x_3, x_1, x_2.]$. Then if $x_1 = x_2$ or $x_2 = x_3$, the triple $[x_1, x_2, x_3.]$ fits into three places of which two are in the same 3-run. While if $n = 3x + 1$ for some integer x, then the triple [x, x, x.] has a unique place in $\Gamma_{3,n}$.

5.2.3 The general case $r \geq 2$, $n \geq 2$

Following the patterns which emerged in Sections 5.2.1, 5.2.2, we now study the general dovetailing graph $\Gamma_{r,n}$ ($r \geq 2$, $n \geq 2$).

5.44 The Dovetailing Graph $\Gamma_{r,n}$ With integers $r \geq 2$, $n \geq 2$, the dovetailing graph $\Gamma_{r,n}$ for model (5.1) consists of the following :-

Nodes These may be divided into two types (using Notation 5.17):-

(a) $(x_1, \dots, x_k)$, $k = 1, \dots, r$.

(b) $(x_1, \dots, x_k.)$ where in particular $\sum_{j=1}^{k} x_j = n(r - k) + (n - 1)$, $k = 1, \dots, r$.

Note The nodes of type (a) are the addresses ending in a, while the type (b) nodes are the towers.

Permitted branches Every node apart from (nr), $(nr - 1.)$ and $(0, nr - n)$ has exactly two possible successors and two possible predecessors.

(a) The compulsory branch :- $(0, nr - n) \to (nr) \to (nr - 1.)$. That is, $ba^{nr-n} \to a^{nr} \to a^{nr-1}b$.

Note This covers the case $k = 1$ for node types (a) and (b) and $k = 2$ node type (a) with $x_1 = 0$.

(b) The possible successors to the node $(x_1, \dots, x_k)$ for $2 \leq k \leq r$ depends on the value of x_1 :-

(i) *Case $x_1 = 0$* $(3 \leq k \leq r)$

$$\begin{tikzcd} & (x_2, \dots, x_k+n) \\ (0, x_2, \dots, x_k) \arrow[ur] \arrow[dr] & \\ & (x_2, \dots, x_k+n-1.) \end{tikzcd}$$

(ii) *Case $x_1 \geq 1$* $(2 \leq k \leq r)$

$$\begin{tikzcd} & (x_1-1, x_2, \dots, x_k+1) \\ (x_1, x_2, \dots, x_k) \arrow[ur] \arrow[dr] & \\ & (x_1-1, x_2, \dots, x_k.) \end{tikzcd}$$

(c) The possible successors to the node$(x_1, \dots, x_k.)$ for $2 \leq k \leq r$ are:-

(i) *Case $x_1 \leq n - 1$*

$$\begin{tikzcd} & (x_2, \dots, x_k, x_1+1) \\ (x_1, x_2, \dots, x_k.) \arrow[ur] \arrow[dr] & \\ & (x_2, \dots, x_k, x_1.) \end{tikzcd}$$

(ii) *Case $x_1 > n - 1$*

$$\begin{tikzcd} & (y, x_2, \dots, x_k, 1) \\ (x_1, x_2, \dots, x_k.) \arrow[ur] \arrow[dr] & \\ & (y, x_2, \dots, x_k, 0.) \end{tikzcd}$$

where $y = n(r - k) - 1 - \sum_{j=2}^{k} x_j$.

Note case(ii) above can only occur when $k < r$.

Level 1	1-run
	singleton
Level 2	2-runs
	pairs
.	.
.	.
.	.
Level k	k-runs
	k-tuples
.	.
Level r	r-runs
	r-tuples

Figure 5.6 An Overview of Dovetailing Graph $\Gamma_{r,n}$, $r \geq 1$, $n \geq 2$.

Within all the optimal sequences we consider, we shall dovetail the nodes $(x_1, ..., x_{k\bullet})$ into *k-tuples* $(k = 1, ..., r)$ and insert them between

k -runs (see Definition 5.45). Then the dovetailing graph $\Gamma_{r,n}$ will have r levels where the kth level is itself split into 2 sublevels k-runs and k-tuples. Figure 5.6 gives an overview of the structure of $\Gamma_{r,n}$.

Definition 5.45 The kth level of $\Gamma_{r,n}$ $(1 \leq k \leq r)$ is split into two sub-levels namely :-

(i) *k-runs* We group together the addresses $(x_1, ..., x_k)$ $(1 \leq k \leq r)$ with common middle term $\tau = ba^{x_2}b ... a^{x_{k-1}}b$ as a run in the sequence $[x_2, ..., x_{k-1}] = a^{t-1}ba^{x_2}b...a^{x_{k-1}}ba^t$ where $t = n(r - k + 1) - \Sigma^{k-1}_{j=2} x_j$. To represent this grouping, we write,

$[x_2, ..., x_{k-1}]$:

$$(t - 1, x_2, ..., x_{k-1},1) \rightarrow (t - 2, x_2, ..., x_{k-1},2) \rightarrow ... \rightarrow (0,x_2, ..., x_{k-1},t) \qquad (5.6)$$

(ii) *k-tuples of towers* Recall that addresses ending b group into towers $(x_1, ..., x_{k\bullet}) = a^{x_1}b...a^{x_k}b$ where $\Sigma^{k}_{j=1} x_j = n(r - k) + (n - 1)$, $x_j \geq 0$ for $j = 1, ..., k$. These towers may be dovetailed into k-tuples $(1 \leq k \leq r)$ as :-

$$[x_1,...,x_{k\bullet}] : (x_1,...,x_{k\bullet}) \rightarrow (x_2,...,x_k,x_{1\bullet}) \rightarrow ... \rightarrow (x_k,x_1,...,x_{k-1\bullet}) \qquad (5.7)$$

where $[x_1, ..., x_{k\bullet}]$ is the sequence: $a^{x_1}b...a^{x_k}ba^{x_1}b...a^{x_{k-1}}b$, dovetailing them.

Within the structure (5.7), consider the branch

$$(x_i, x_{i+1}, ...,x_{i-1\bullet}) \rightarrow (x_{i+1},...,x_{i-1},x_{i\bullet}).$$

If $x_i > n - 1$, the last address of the tower $(x_i, x_{i+1}, ...,x_{i-1\bullet})$ is $a^xba^{x_{i+1}}b...a^{x_{i-1}}b$ where $x = n(r - k) - \Sigma^{k}_{j=1,j\neq i} x_j$. Moreover, before the tower $(x_{i+1},...,x_{i-1},x_{i\bullet})$ is reached there are an extra x symbols $a^{x-1}b$. We shall take these as the starting symbols of the (k + 1)-run $[x_{i+1},...,x_{i-1}]$. The inclusion of this (k + 1)-run is essential to the dovetailing of such towers (see Figure 5.7).

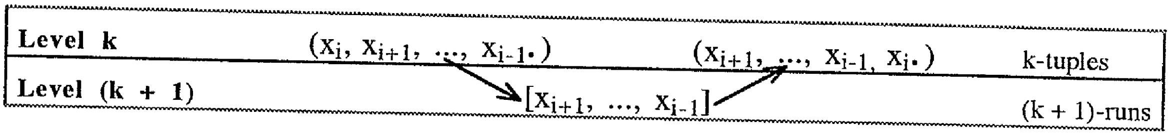

Level k
$(x_i, x_{i+1}, ..., x_{i-1}\bullet)$
$(x_{i+1}, ..., x_{i-1}, x_i\bullet)$
k-tuples
Level (k + 1)
$[x_{i+1}, ..., x_{i-1}]$
(k + 1)-runs

Figure 5.7 When $x_i > n - 1$, the (k + 1)-run is included to complete the branch.

Special Case In level 1, we have the 1-run, a^{nr}, and the 1-tuple or singleton, $a^{nr-1}b$. Further the unique path is $(nr) \rightarrow (nr - 1\bullet)$ and it is only possible to leave level 1 via the node $(nr - 1\bullet)$.

5.46 Constructing Optimal Sequences To obtain an optimal sequence from the dovetailing graph $\Gamma_{r,n}$, the basic idea is to move along the kth level k-runs inserting k-tuples (k = 1, ..., r) and ensuring that each k-tuple is inserted exactly once. As before, we shall start at node (nr) of the top level - level 1. Then the sequence path begins $(nr) \rightarrow (nr - 1\bullet)$ and ends $(0, nr - n) \rightarrow (nr)$.

Observe initially that when at any branch of the 2-run, we may insert a corresponding pair as shown in Figure 5.8. In other words, we can insert pairs between branches of the 2-run without affecting the 'flow' of the 2-run.

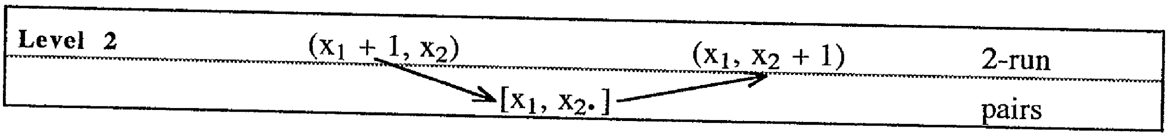


Figure 5.8 A pair may be inserted between branches of a 2-run in $\Gamma_{r,n}$.

However, if $x_1 > n - 1$, then in order to complete the pair $[x_1, x_2\bullet]$, we automatically include the 3-run $[x_2]$ (see Definition 5.45, Figure 5.9). Further, analogously to the structure of level 2, we may insert triples between branches of the 3-run $[x_2]$ in level 3. Moreover, these triples may include 4-runs of level 4 and so we may insert 4-tuples and so on until we reach the r-runs and r-tuples of the bottom level - level r.

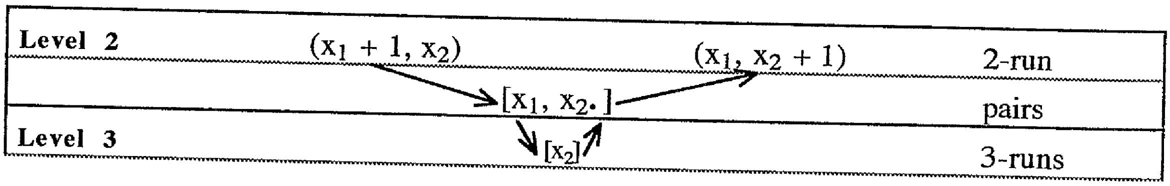


Figure 5.9 When $x_1 > n - 1$, the 3-run $[x_2]$ is essential to complete pair $[x_1, x_2\bullet]$ in $\Gamma_{r,n}$.

Note (i) Since $x_j \leq n\text{-}1$ $(1 \leq j \leq r)$, the r-tuple $[x_1, ..., x_r\bullet]$ contains only addresses of the corresponding towers.

(ii) Let $m \geq 1$ be the least integer such that the tower $(x_1, \ldots, x_{k\cdot})$ equals $(x_1, \ldots, x_m, \ldots, x_1, \ldots, x_{m\cdot})$. Then the k-tuple $[x_1, \ldots, x_{k\cdot}]$ fits between m different branches of k-runs corresponding to the m cyclically related orders (see Example 5.47 below).

Example 5.47

(i) $[x_1,x_2,x_{3\cdot}]$ where x_1, x_2, x_3 are not all equal - there are three possibilities for the insertion of the triple as shown below:-

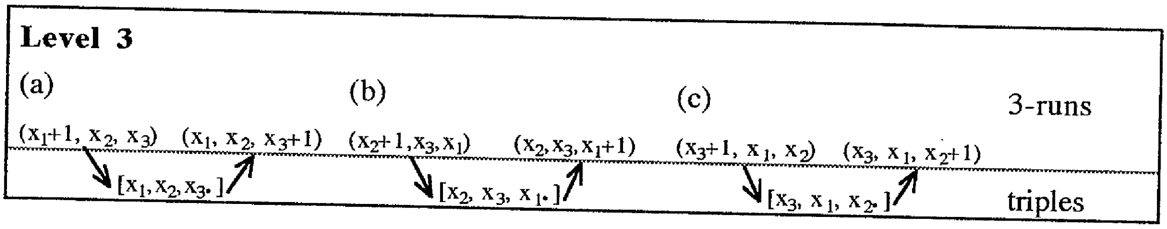


Figure 5.10 The triple $[x_1,x_2,x_{3\cdot}]$ may be inserted in three different ways.

(ii) $[x_1,x_2,x_1,x_{2\cdot}]$ where $x_1 \neq x_2$ - we are restricted to just two possibilities:-

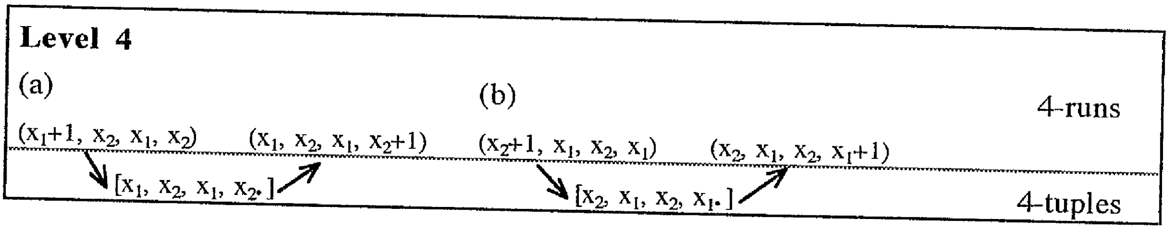


Figure 5.11 The 4-tuple $[x_1,x_2,x_1,x_{2\cdot}]$ may be inserted in only two ways.

Then, for $r \geq 2$, $n \geq 2$, we are now able to obtain an optimal sequence simply by moving along the 2-run inserting pairs, along the 3-runs inserting triples, ..., and along the r-runs inserting r-tuples. However, for larger values of r, n the graph $\Gamma_{r,n}$ becomes very complicated. Ideally, we require a computer algorithm which quickly produces an optimal sequence for any $r \geq 2$, $n \geq 2$. We investigate the possible structure of such an algorithm in the next section.

5.3 Implementation - The algorithms

Our aim is to design a computer algorithm which produces an optimal sequence for any $r \geq 2$, $n \geq 2$. At present, we have not reached this goal. We are however aware of the similarities between the structure of the algorithms for r and $(r + 1)$. But on increasing r even by just one, things do become significantly more complicated. In this section we do no more than state the algorithms for $r = 2, 3, 4$ and offer a template for the algorithm for general r. We are hopeful of a general algorithm in the near future.

5.3.1 Algorithm for $r = 2$, $n \geq 2$

The dovetailing graph $\Gamma_{2,n}$ $(n \geq 2)$ is shown in Figure 5.2. Notice the two cases, depending on whether or not n is even. Recall, when n is even there are no pairs of the form $[x, x.]$ whereas for n odd there is exactly one of this form, namely $[(n-1)/2, (n-1)/2.]$. As discussed previously our optimal path will start at $(2n)$. We impose the condition that each pair is inserted as soon as possible. Then our path begins $(2n) \rightarrow (2n - 1.) \rightarrow [n - 1, 0.] \rightarrow (n - 1, 1) \rightarrow [n - 2, 1.] \rightarrow \ldots$. Further, on leaving the nodes $[n/2, n/2 - 1.]$, $[(n - 1)/2, (n - 1)/2.]$ for n even, odd respectively, all the pairs have been included and the path now proceeds straight along the remainder of the 2-run ending with the branch $(0, n) \rightarrow (2n)$. We shall now show that Algorithm 5.50 produces one complete period of the optimal sequence corresponding to this optimal path through $\Gamma_{2,n}$. Following the algorithm, the sequence produced begins with a^n and with $j = 0$, the subsequence a^nb^2 is added so that the corresponding path begins

$$(2n) \rightarrow (2n - 1.) \rightarrow (n - 1, 0.).$$

For $j = 1, 2, \ldots, (n - 1)$ div 2, a further $a^{n-j}ba^jb$ is added. In particular when $j = 1$, the subsequence $a^{n-1}bab$ is added so that the path continues

$$(n - 1, 0.) \rightarrow (0, n - 1.) \rightarrow (n - 1, 1).$$

Then the pair $[n - 1, 0.]$ has been inserted in the required form.

Now consider the path formed when j is increased from $x - 1$ to x ($x = 1, \ldots, (n - 1)$ div 2). Immediately before j becomes x, the path is at the node $(n - x, x - 1.)$ while when $j = x$, the subsequence $a^{n-x}ba^xb$ is added and the path extends

$$(n - x, x - 1.) \rightarrow (x - 1, n - x.) \rightarrow (n - x, x),$$

so that every pair occurs in the form $[n - x , x - 1.]$ for $x = 1, \ldots, (n - 1)$ div 2 as required.

When $j = (n - 1)$ div 2, the path corresponds with

$$(n - (n - 1) \text{ div } 2, (n - 1) \text{ div } 2) \rightarrow (n - (n - 1) \text{ div } 2 - 1, (n - 1) \text{ div } 2.).$$

If n is odd, then this branch is

$$((n - 1) \text{ div } 2 + 1, (n - 1) \text{ div } 2) \rightarrow ((n - 1) \text{ div } 2, (n - 1) \text{ div } 2.)$$

so that the final pair $[(n - 1)$ div $2, (n - 1)$ div $2.]$ is included. However, if n is even this only includes one half of the final pair $[n - (n - 1)$ div $2 - 1, (n - 1)$ div $2.]$ and so the algorithm adds the subsequence $a^{n-(n-1)\text{div } 2}b$ to include its partner, namely $((n - 1)$ div $2, n - (n - 1)$ div $2.)$.

At this point, the algorithm has produced one complete period of the sequence. It is easily checked (by cyclically continuing the sequence) that the final branches are

$$((n - 1) \text{ div } 2, n - (n - 1) \text{ div } 2 + 1) \rightarrow \dots \rightarrow (0, n) \rightarrow (2n),$$

so that our optimal path is complete.

Notation 5.48 In the algorithms shortly to be described, various portions of the optimal sequence are generated by the routines listed below.

(i) *Power*('x', i) gives the subsequence x^i,

(ii) *SetChar*(x, 'γ') gives $a^x\gamma$,

(iii) *Set 1*(x) gives a^xb,

(iv) *Set 2*(x_1, x_2) gives $a^{x_1}ba^{x_2}b$,

(v) *Set k*(x_1, ..., x_k) gives $a^{x_1}b\dots a^{x_m}b$ where m is the least positive integer such that (x_1, ..., $x_{k\bullet}$) equals (x_1, ..., x_m, ..., x_1, ..., $x_{m\bullet}$) That is, the k-tuple (x_1, ..., $x_{k\bullet}$) consists of k/m copies of the m-tuple (x_1, ..., $x_{m\bullet}$) ($k \geq 3$),

(vi) *Insert k-tuples*(k, r) is the recursive subroutine ($k \geq 3$)

```
begin
    if $(n(r - k + 1) = kj + 1)$ and $((k = 3)$ or $((k > 3)$ and $(i_1 = \dots = i_{k-3} = j)))$ then
        Set 1(j)                                                  {k-tuple [j, ..., j. ]}
    for $i_{k-2}$:= up(k, r) downto down(k, r) do
        begin
            Set k($n(r - k + 1) - j - 1 - \sum_{s=1}^{k-2} i_s, i_{k-2}, \dots, i_1, j$)            {k-tuples}
            if $k < r$ then
                Insert k-tuples(k + 1, r)
        end
end
```

where *up*(k, r) and *down*(k, r) are suitably chosen to avoid duplicate insertions. Values for r = 3, 4 are given in Algorithms 5.51, 5.52.

Example 5.49 Optimal sequence for model (5.1) with r = 2, n = 4. Following the path of Figure 5.12 or using Algorithm 5.50 both yield the optimal sequence with complete cycle $a^8b^2a^3baba^2b$. Similarly, for r = 2, n = 5, Figure 5.3(i) is the path corresponding to Algorithm 5.50.

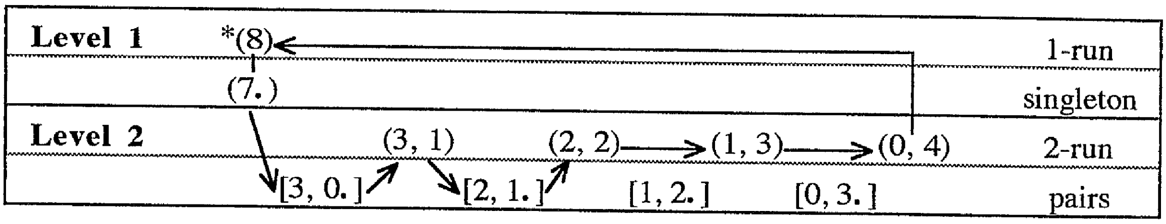


Figure 5.12 The optimal path through $\Gamma_{2,4}$ taken when each pair is inserted as soon as possible.

5.3.2 Algorithm for $r = 3$, $n \geq 2$

We now consider $r = 3$, $n \geq 2$. Notice that since the pairs are $[x_1, x_2.]$ with $x_1 + x_2 = 2n - 1$, there are none of the form $[x, x.]$. However, if $n = 3x + 1$ for some integer x, then the 'triple' $[x, x, x.]$ does exist and has the unique position $a^{x+1}ba^xba^x \rightarrow a^xba^xba^xb \rightarrow a^xba^xba^{x+1}$. Figure 5.4 shows the general structure of the dovetailing graph $\Gamma_{3,n}$.

As we hope eventually to produce an algorithm for any $r \geq 2$, $n \geq 2$, we must take a path analogous to that of $r = 2$ above. Thus, we impose the condition : every pair is inserted as soon as possible. However, this will involve 3-runs and so we impose the second condition : every triple is inserted as soon as possible. Consider this optimal path. Since the pairs are to be included as soon as possible, each occurs in the form $[2n - j - 1, j.]$ with $j = 0, ..., (2n - 1)$ div 2. Also, since $2n - j - 1 \geq n - 1$, this pairing must involve the corresponding 3-run [j]. That is,
$(2n - j - 1, j.) \rightarrow (n - j - 1, j, 1) \rightarrow (n - j - 2, j, 2) \rightarrow ... \rightarrow (0, j, n - j) \rightarrow (j, 2n - j - 1.)$.
Then clearly, the algorithm

```
        Power('a', n)
        for j:= 0 to (2n - 1) div 2 do
            begin
                Power('a', 1)
                Set 2(2n - 1 - j, j)
            end
```

will produce the sequence corresponding to the path
$(3n) \rightarrow (3n - 1.) \rightarrow [2n - 1, 0.] \rightarrow (2n - 1, 1) \rightarrow ... \rightarrow (n + 1, n - 1) \rightarrow [n, n - 1.]$.
But the optimal path must include the triples and consequently the triples must be inserted between branches of the 3-runs. The insertion of the triple $[i, j, n - j - i - 1.]$ in an optimal path is equivalent to the insertion of the subsequence $a^{n-i-j-1}ba^iba^jb$. Then, any algorithm which produces an optimal sequence would require *Set 3*$(n - j - i - 1, i, j)$ for each triple inserted. Recall, in our optimal sequence we wish each triple to occur in the first possible

position. Beginning with $j = 0$ (since the first 3-run is $[0]$), every triple with at least one zero entry will be found between branches of the 3-run $[0]$. When $j = 1$, of the triples not yet inserted, we now include those with at least one entry equal to one and so on until all the triples have been included. Triples of the form $[j, j, n - 1 - 2j.]$ where $j < n - 1 - 2j$ must occur in the first of the two possible positions within the 3-run $[j]$. While, if $n = 3j + 1$, the triple $[j, j, j.]$ has a unique position within the 3-run $[j]$ and is included by the command *Set 1*(j). With *up* $= n - 1 - j - (j + 1)$ and *down* $= j$, the subroutine

```
for $i_1$ = up down to down do
    Set 3( $n - 1 - i_1 - j, i_1, j$ )
```

inserts the required triples between the branch $(2n - 1 - j, j.) \rightarrow (j, 2n - 1 - j.)$ for $j = 0, ..., (2n - 1)$ div 2. Then, by inserting this subroutine within the for loop over j above, we obtain an algorithm which produces the sequence corresponding to the path which begins $(3n) \rightarrow (3n - 1.)$ and which includes every pair and every triple apart from $[n, n - 1.]$. To complete the final pair we add a further $a^n b$. The algorithm we have described above is Algorithm 5.51. Hence Algorithm 5.51 produces a complete cycle of the optimal sequence corresponding to this path ($n \geq 2$).

Even at this stage a pattern is beginning to emerge. However before we try to devise a template for the general algorithm, we will consider $r = 4$.

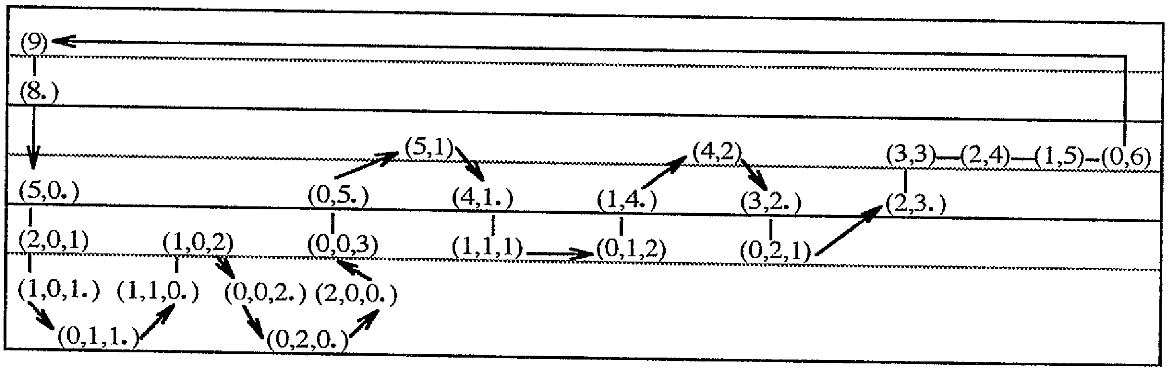


Figure 5.13 The optimal path through $\Gamma_{3,3}$ corresponding to the optimal sequence of Algorithm 5.51.

5.3.3 Algorithm for $r = 4$, $n \geq 2$

Each k-tuple ($k = 2, 3, 4$) has exactly k possible positions in the dovetailing graph $\Gamma_{4,n}$ unless it is one of the following.

(i) The pair $[x, x.]$ - exists if and only if $3n = 2x + 1$ - unique position in $\Gamma_{4,n}$,

(ii) the triple $[x, x, x.]$ - exists if and only if $2n = 3x + 1$ - unique position in $\Gamma_{4,n}$,

(iii) the 4 - tuples

(a)[x, x, x, x.] - exists if and only if n = 4x + 1 - unique position in $\Gamma_{4,n}$,

(b)[x, y, x, y.] - exists if and only if n = 2(x + y) + 1 - two possible positions in $\Gamma_{4,n}$.

Extending the conditions of the previous algorithms, we insert pairs, triples and quadruples as soon as possible and thus automatically include the 3-runs and 4-runs. By extending the explanation of Algorithms 5.50 and 5.51, it can be checked that Algorithm 5.52 produces a period cycle of the optimal sequence corresponding to this path ($n \geq 2$).

5.3.4 Algorithm for $r \geq 2$, $n \geq 2$ - a template

Following the pattern of the algorithms above, our general algorithm (see Template Algorithm 5.53) will produce the optimal sequence obtained by inserting pairs, triples, ..., r-tuples as soon as possible.

In the general algorithm, j increases in unit steps from zero to ((r - 1)n - 1) div 2. Then, so that every k-tuple is inserted as soon as possible, starting with j = 0, each k-tuple [x_1, ..., x_k.], with at least one entry $x_i = 0$, must be included before j is increased to one. Further, this k-tuple must be included by the command *Set k*(x_{i+1}, ..., x_{i-1}, 0) where [x_{i+1}, ..., x_{i-1}, 0.] is its maximum lexicographic cyclic form ending in zero. Similarly, when j becomes one, every k-tuple not yet included and with at least one $x_i = 1$, must be inserted by the command *Set k*(x_{i+1}, ..., x_{i-1}, 1) where [x_{i+1}, ..., x_{i-1}, 1.] is its maximum lexicographic form ending with one. Then j becomes two, and so on up to ((r - 1)n - 1) div 2. Thus to obtain the general algorithm, we require the functions *up*(k, r) and *down*(k, r) which determine the limits in *Insert k-tuples*(k, r) to ensure that the above condition is satisfied.

For example, with r = 5, n = 10, the 5-tuple [3, 1, 2, 2, 1.] will occur in any optimal sequence in one of its five cyclic forms :- [3, 1, 2, 2, 1.], [1, 2, 2, 1, 3.], [2, 2, 1, 3, 1.], [2, 1, 3, 1, 2.], [1, 3, 1, 2, 2.]. However, if the optimal sequence is produced using Template 5.53 (the algorithm promised in this section), then this 5-tuple must be included by the command *Set 5*(3, 1, 2, 2, 1) when j equals one.

Algorithm 5.50 - produces an optimal sequence for model (5.1) with $r = 2$, $n \geq 2$.

```
begin
    Power('a', n)
    for j := 0 to (n - 1) div 2 do
        begin
            Power('a', 1)
            Set 2(n - 1 - j, j)                {Pairs}
        end
    if n even then
    SetChar(n div 2, 'b')
end
```

Algorithm 5.51 - produces an optimal sequence for model (5.1) with $r = 3$, $n \geq 2$.

```
begin
    Power('a', n)
    for j := 0 to (2n - 1) div 2 do
        begin
            Power('a', 1)
            Set 2(2n - 1 - j, j)               {Pairs}
            Insert k-tuples(3, 3)
        end
    SetChar(2n - 1 - ((2n - 1) div 2), 'b')
end
```

where *up*(3, 3) = $n - 1 - j - (j + 1)$ and *down*(3, 3) = j.

Algorithm 5.52 - produces an optimal sequence for model (5.1) with $r = 4$, $n \geq 2$.

```
begin
    Power('a', n)
    for j := 0 to (3n - 1) div 2 do
        begin
            Power('a', 1)
            Set 2(3n - 1 - j, j)               {Pairs}
            Insert k-tuples(3, 4)
        end
    if n even then
        SetChar(3n - 1 - ((3n - 1) div 2), 'b')
end
```

and where *up*(3, 4) = $2n - 1 - j - (j + 1)$,
down(3, 4) = j,
up(4, 4) = $n - 1 - j - i_1 - (j + 1)$,
down(4, 4) = $j + ((i_1 + j)/((n + 1) \text{ div } 2))$.

Template Algorithm 5.53 - produces an optimal sequence for model (5.1) with $r \geq 2$, $n \geq 2$.

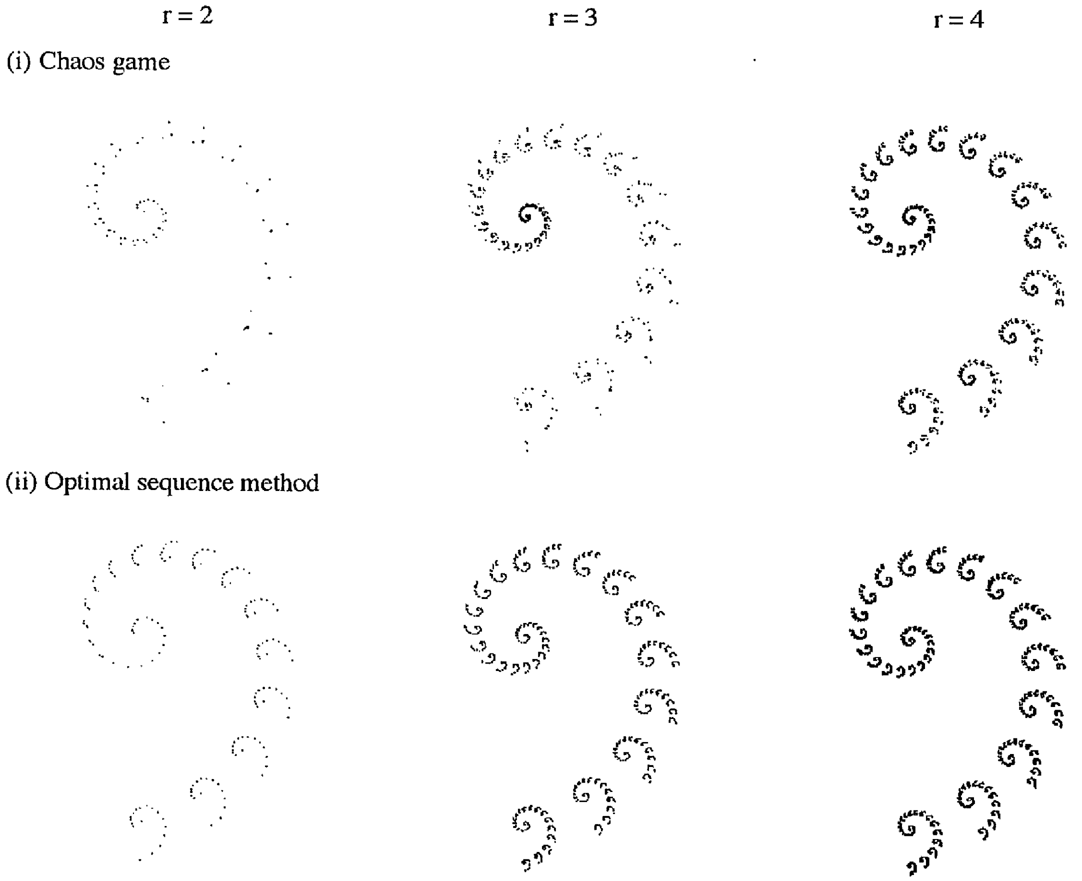
```
begin
    Power('a', n)
    for j:= 0 to ((r - 1)n - 1) div 2) do
        begin
            Power('a', 1)
            Set 2((r - 1)n - 1 - j, j)         {Pairs}
            Insert k-tuples(3, r)
        end
    if (r even, n even) or (r odd, any n) then
        SetChar((r-1)n - 1 - ((r - 1)n - 1) div 2,'b')
end
```

where *up*(3, r) = $n(r - 2) - 1 - j - (j + 1)$,
down(3, r) = j, *up*(4, r) = $n(r - 3) - 1 - j - i_1 - (j + 1)$
and *down*(4, r) = $j + ((i_1 + j)/((n(r - 3) + 1) \text{ div } 2))$.

5.4 The results

We now compare the optimal sequence method (using the Algorithms of Section 5.3) with the chaos game for some non-uniform IFS's satisfying model (5.1). Again, we use RandomX for the chaos game.

Example 5.54 The *Ear*. The ear is the attractor of the IFS $\{w_a, w_b\}$ where w_a consists of a rotation through 20 degrees followed by a uniform scaling in the ratio $r_a = 0.9$ and w_b is a uniform scaling in the ratio 0.2 then translation downwards (see Table 5.1). Thus, this IFS fits model (5.1) with $r \geq 1$ and $n = 16$. The images produced by the chaos game and the optimal sequence method for r = 2, 3, 4 are given in Figure 5.14.



r = 2 r = 3 r = 4

(i) Chaos game

(ii) Optimal sequence method

Figure 5.14 The *ear* produced by (i) the chaos game and (ii) an optimal sequence for r = 2, 3, 4.

For r = 2, the chaos game produced 70 repeated points out of 170 whereas the optimal sequence had no repetitions. Notice, the optimal sequence image has clearly outlined copies of the ear within itself while the chaos game approximation has successive clumps of about three points.

For r = 3, the optimal sequence method plots 77.2% original points compared with only 37.0% for the chaos game. The corresponding optimal sequence image shows greater detail.

Finally, for r = 4, the optimal sequence and the chaos game plot 20.4% and 14.3% original points respectively, Comparing the corresponding images, we see that the optimal sequence approximation is sharper.

Example 5.55 *The Spiral.* This IFS consists of two affine transformations w_a, w_b of respective ratios 0.95 and 0.25 and so fits model (5.1) with n = 28. Figure 5.15 gives the optimal sequence approximations for r = 2, 3. For comparison, the results of the chaos game for the same number of iterations are also given.

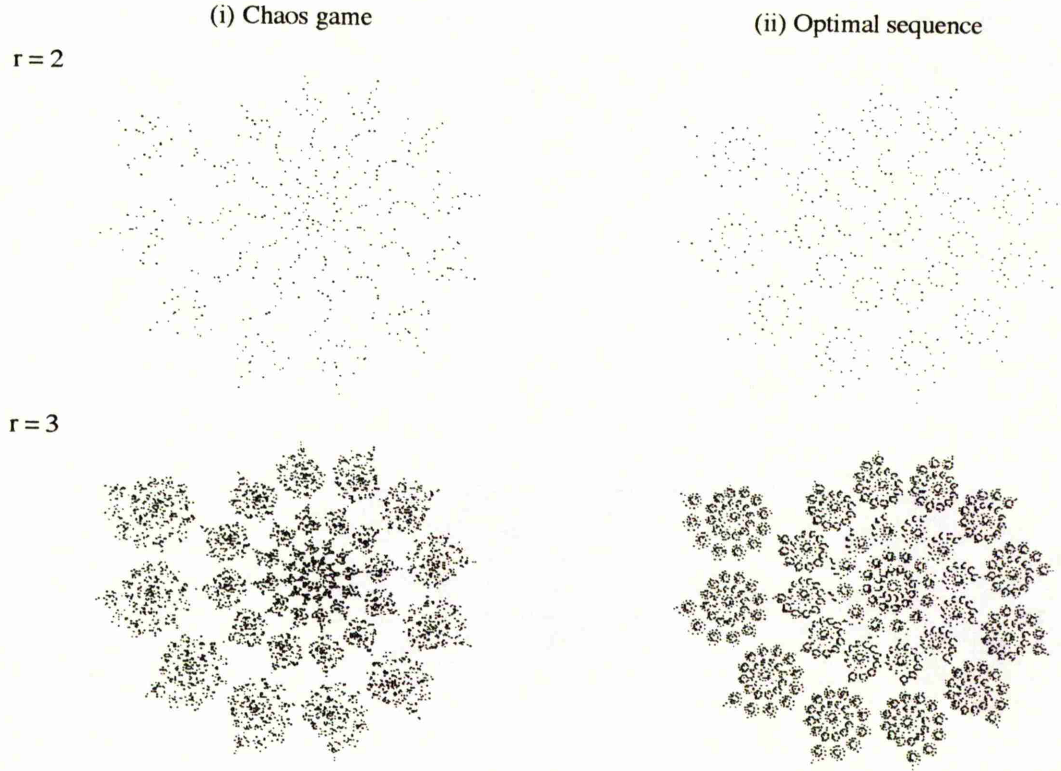


Figure 5.15 The *spiral* produced by (i) the chaos game and (ii) an optimal sequence for r = 2, 3.

For r = 2, neither method plots any duplicate points. However, the corresponding optimal sequence image shows more structure. For r = 3, the chaos game plots only 86.4%

original points while the optimal sequence plots 99.6%. Notice the finer detail in the optimal sequence image. In comparison, the chaos game image appears blurred. Also, the optimal sequence method is slightly faster.

Attractor	map	a	b	c	d	e	f	probability p
The	w_a	0.846	-0.308	0.308	0.846	0	0	0.953
Ear	w_b	0.200	0	0	0.200	0	-2	0.047

Table 5.1 The code and probabilities p for the *ear* [16].

Attractor	map	a	b	c	d	e	f	probability p
The	w_a	0.823	-0.475	0.475	0.823	0.301	-0.174	0.935
Spiral	w_b	0.25	0	0	0.25	0	0.5	0.065

Table 5.2 The code and probabilities p for the *spiral* [21].

In this chapter, we considered only the non-uniform IFS satisfying model (5.1). We studied the construction of the corresponding optimal sequences and compared their performance with the chaos game. We found that the optimal sequence method was faster than the chaos game and produced superior images. However, we must try to extend the optimal sequence method to other non-uniform IFS. This is studied in Chapter 6.

Chapter 6 Optimal sequences for non-uniform iterated function systems - A more general model

In Chapter 5, we illustrated the improvement possible when the optimal sequence method rather than the chaos game is used to produce the attractor of any non-uniform IFS satisfying model (5.1). However, if the optimal sequence method is to be useful in practice, it must be expanded to include non-uniform IFS besides those satisfying (5.1). In this chapter, we explain in detail how the algorithms of Chapter 5, may be extended to more complicated models.

6.1 Extending the simplest case

In order that optimal sequences may be used to produce a wider variety of attractors, we now consider the IFS with three affine transformations w_a, w_b, w_c of ratios r_a, r_b, r_c , satisfying Assumption 6.1 below.

Assumption 6.1 For some positive integers $n \geq 2$, $r \geq 1$, we have

$$r_a^n = r_b = r_c \qquad r_b^r = \varepsilon. \tag{6.1}$$

Note Again, it is sufficient for (6.1) to hold for real numbers lower than the integers r, n.

6.1.1 The addresses and optimal sequences

We extend some of the results of Chapter 5 to model (6.1) before going on to illustrate how Algorithms 5.50, 5.51 and 5.52 may be easily modified to produce an optimal sequence for any IFS satisfying (6.1).

Definition 6.2 Since the transformations w_b, w_c are of the same ratio, the *weight*, wt(σ), of a sequence σ over {a, b, c} is

$$\mathrm{wt}(\sigma) = (\text{no of a in } \sigma) + n \times (\text{no of b in } \sigma) + n \times (\text{no of c in } \sigma).$$

Then, wt(a) = 1 and wt(b) = wt(c) = n.

To obtain, a complete list of the addresses for model (6.1), we begin with $\mathcal{A} = \mathcal{A}_a \cup \mathcal{A}_b \cup \mathcal{A}_c$ and recursively replace $\mathcal{A}_\sigma$ by $\mathcal{A}_{\sigma a} \cup \mathcal{A}_{\sigma b} \cup \mathcal{A}_{\sigma c}$ whenever $\rho(\sigma) > \varepsilon$. This may be represented by an address tree but equivalently, we have:-

Corollary 6.3 Since the transformations w_b, w_c each have weight n, the addresses corresponding to model (6.1) with $r \geq 1$, $n \geq 2$ are:-

(i) those corresponding to the nodes of $\Gamma_{r,n}$,

(ii) those obtained from (i) by replacing any number of the b's by c's.

In particular for $r = 2$, $n \geq 2$, the addresses are :-

a^{2n}

$a^x\gamma$ where $n \leq x \leq 2n - 1$ and $\gamma \in \{b, c\}$

$a^{x_1}\gamma a^{x_2}$ where $x_1 + x_2 = n$, $0 \leq x_1 \leq n - 1$, $1 \leq x_2 \leq n$ and $\gamma \in \{b, c\}$

$a^{x_1}\gamma_1 a^{x_2}\gamma_2$ where $x_1 + x_2 = n - 1$, $0 \leq x_1$, $x_2 \leq n - 1$ and $\gamma_i \in \{b, c\}$

while for $r = 3$, $n \geq 2$, the addresses are:-

a^{3n}

$a^x\gamma$ where $2n \leq x \leq 3n - 1$ and $\gamma \in \{b, c\}$

$a^{x_1}\gamma a^{x_2}$ where $x_1 + x_2 = 2n$, $0 \leq x_1 \leq 2n - 1$, $1 \leq x_2 \leq 2n$ and $\gamma \in \{b, c\}$

$a^{x_1}\gamma_1 a^{x_2}\gamma_2$ where $x_1 + x_2 = 2n - 1$, $0 \leq x_1$, $x_2 \leq 2n - 1$ and $\gamma_i \in \{b, c\}$

$a^{x_1}\gamma_1 a^{x_2}\gamma_2 a^{x_3}\gamma_3$ where $x_1 + x_2 + x_3 = n - 1$, $0 \leq x_1$, $x_2 \leq n - 1$ and $\gamma_i \in \{b, c\}$.

Example 6.4 With $r_a{}^2 = r_b = r_{c,}$ $r_b{}^2 = \varepsilon$, the addresses corresponding to (i) and (ii) of Corollary 6.3 are:-

(i)	a^4	a^3b	a^2b	aba	ba^2	ab^2	b^2	bab
(ii)	-	a^3c	a^2c	aca	ca^2	ac^2, abc, acb	c^2, cb, bc	cac, bac, cab

Corollary 6.5 For any IFS with $r \geq 1$, $n \geq 2$ satisfying Assumption 6.1, the total number of addresses is

$$\sum_{k=0}^{r} 2^k \binom{n(r - k + 1) + k - 1}{k}.$$

In particular, for $r = 2, 3$ respectively the number of addresses is $1 + 2n(n + 3)$ and $1 + 2n(2n^2 + 18n + 19)/3$ $(n \geq 2)$.

Proof This result is a simple extension of Corollary 5.27. Firstly, consider the addresses ending in a. There is a^{rn}. Also, for any $x_1, \ldots, x_k$ where $x_j \geq 0$ for $j = 1, \ldots, k - 1$, $x_k \geq 1$ and $\sum_{j=1}^{k} x_j = n(r - k + 1)$, there are 2^{k-1} different addresses of the form $a^{x_1}\gamma_1 a^{x_2}\gamma_2 \ldots a^{x_{k-1}}\gamma_{k-1}a^{x_k}$, where $\gamma_i \in \{b, c\}$. Then for $2 \leq k \leq r$, the number of addresses of this form is $2^{k-1}\Sigma_k`(n(r - k + 1))$, so that the total number of addresses ending with the symbol a is

$$1 + \sum_{k=1}^{r-1} 2^k \binom{n(r - k) + k - 1}{k}.$$

For any $x_1, \ldots, x_k$ where $x_j \geq 0$ for $j = 1, \ldots, k$ and $n(r - k) \leq \Sigma^k_{j=1} x_j \leq n(r - k) + (n - 1)$, there are 2^k different addresses of the form $a^{x_1}\gamma_1 a^{x_2}\gamma_2 \ldots a^{x_{k-1}}\gamma_{k-1}a^{x_k}\gamma_k$ where $\gamma_i \in \{b, c\}$. It is easily checked that the total number of addresses ending in either b or c is

$$2^r\binom{n + r - 1}{r} + \sum_{k=1}^{r-1} 2^k\left[\binom{n(r - k + 1) + k - 1}{k} - \binom{n(r - k) + k - 1}{k}\right].$$

It follows that the total number of addresses is as stated above.

Construction 6.6 Again, there are two methods of dovetailing, that is:-

(a) *Runs* - as for Construction 5.30(a),

(b) *Towers* - as for Construction 5.30(b) except that they may end in either b or c. Again there are two types of towers.

Type I $\quad$ a a ... a τ $\gamma = a^n\tau\,\gamma$, of height n where $\gamma \in \{b, c\}$

Type II $\quad$ a a ... a γ_1 τ $\gamma_2 = a^{h-1}\gamma_1\tau\,\gamma_2$, of height $h \leq n$ where $\gamma_i \in \{b, c\}$

Note As before, in any optimal sequence, the addresses of a tower are found within the tower and at no other point.

Theorem 6.7 Let S denote an optimal sequence for any IFS satisfying model (6.1). Then S is cyclic. Further, the reverse of a cyclic optimal sequence is itself an optimal sequence.

Proof Recall wt(b) = wt(c) = n. Then, the results are simple extensions of Theorem 5.28 and Theorem 5.32 respectively.

6.1.2 Implementation - Modifying algorithms

We wish to extend Algorithms 5.50, 5.51, 5.52 and Template 5.53 to model (6.1). We shall use the dovetailing graph $\Gamma_{r,n}$ as a guide but we must remember that for each symbol b in an address, there is an associated address with b replaced with c. Following Section 5.3.4, we will begin at the level 1 node, (nr). However, since the addresses $a^{nr-1}c$, ..., $a^{nr-n}c$ must also be included, we insert the symbol c between the nodes (nr) and (nr - 1.) to produce the dovetailed sequence $a^{nr}ca^{nr-1}b$. Again, we shall move along the k-runs, inserting each k-tuple in the first suitable position. But rather than simply inserting $a^{x_1}ba^{x_2}b$ for the pair $[x_1, x_2.]$, we include the corresponding cases for bc, cb, cc, by inserting the sequence

(i) $T = a^{x_1}ba^{x_2}ca^{x_1}ca^{x_2}ba^{x_1}ca^{x_2}ca^{x_1}ba^{x_2}b$ for the pair $[x_1, x_2.]$ and

(ii) $T = a^{x}ba^{x}ca^{x}ca^{x}b$ for the pair $[x, x.]$.

Similarly, for each k-tuple, we replace the 'all b' version of $[x_1, ..., x_k.]$ by a sequence T which dovetails the 2^k k-digit sequences over {b, c} with all k cyclic forms of $(x_1, ..., x_k.)$.

Below, we discuss in detail how these sequences T may be formed and how they may be used to extend Algorithms 5.50 and 5.51 to model (6.1). We then deal with the more general case $r \geq 2$, $n \geq 2$.

6.1.2.1 Model (6.1) - The case r = 2, n ≥ 2

We show that Algorithm 6.10 does produce a suitable optimal sequence and discuss why the corresponding sequences T are chosen.

Following Algorithm 6.10, the sequence begins $a^{2n}ca^{n-1}$, the first 2n+1 symbols of which provide the starting points for the addresses (in order) a^{2n}, $a^{2n-1}c$, ..., $a^{n}c$, $a^{n-1}ca$, ..., aca^{n-1}. Then, with j = 0, we require a sequence T for the pair [n - 1, 0.] which dovetails the corresponding bb, bc, cb, and cc towers. However, if the first non 'a' symbol of T were 'c', then the address $a^{2n-1}c$ would be repeated. Consequently, T must begin $a^{2n-1}b$. In fact, for the pair $[x_1, x_2.]$, the only suitable T is given in (i) above, while for [x, x.], the sequence T must be as in (ii) above. So (with j = 0), the sequence continues $a^{n}baca^{n-1}cba^{n-1}c^{2}a^{n-1}b^{2}$. Then, the addresses ca^{n}, $a^{2n-1}b$, ..., $a^{n}b$ are included, along with every tower of the form $a^{n-1}\gamma_1\gamma_2$ and $\gamma_1 a^{n-1}\gamma_2$, except for $ba^{n-1}b$ where $\gamma_1, \gamma_2 \in \{b, c\}$. But, with j = 1, the next part of the sequence begins $a^{n-1}ba...$, and thus the tower $ba^{n-1}b$ and the 2-run address $a^{n-1}ba$ are now included. The remainder of the corresponding T sequence includes every tower for the pair [n - 2, 1.] apart from $aba^{n-2}b$. However, this tower will be dovetailed into the next pair sequence T along with the 2-run address $a^{n-2}ba^2$, and so on until j = (n - 1) div 2. If n is even then the final pair is [n div 2, n div 2 -1.], while for n odd the final pair is [n div 2, n div 2.]. In either case, after the insertion of the corresponding T sequence, all addresses have now been included apart from the tower $a^{(n-1) \text{ div } 2}ba^{n-1-(n-1) \text{ div } 2}b$ and the 2-run addresses $a^{n \text{ div } 2}ba^{n-(n \text{ div } 2)}$, ..., aba^{n-1}, ba^n. But the addition of the final subsequence $a^{n \text{ div } 2}b$ ensures the inclusion of this tower and provides starting points for each of the remaining 2-run addresses.

6.1.2.2 Model (6.1) - The case r = 3, n ≥ 2

An algorithm to produce an optimal sequence for model (6.1) with r = 3, n ≥ 2, is similar to Algorithm 5.51 apart from a few modifications. In particular, each pair or triple insertion is replaced by the insertion of a suitable pair or triple sequence T. We show how to form these T sequences and conclude that Algorithm 6.11 produces a suitable optimal sequence.

All pairs are of the form [x_1, x_2.] with $x_1 \neq x_2$ and the corresponding sequences T will be the same as for r = 2 above. For each valid x_1, x_2, x_3, we replace the bbb version of the triple [x_1, x_2, x_3.] by a sequence T which dovetails all eight versions bbb, ..., ccc of all three of (x_1, x_2, x_3.), (x_2, x_3, x_1.), (x_3, x_1, x_2.) giving a total of twenty four. Let t be a sequence over {b, c} of period length eight containing every triple over {b, c} exactly once. This may be formed via a primitive polynomial of degree 3 over the finite field GF(2) or by combinatorial means [10]. Then the sequence T is formed from t in one of two ways:-

(a) *the triple [x_1, x_2, x_3.] where x_1, x_2, x_3 are not all equal*:- Marry the first twenty four symbols of t in order, with the sequence $x_1x_2x_3 \ldots x_1x_2x_3$ of length twenty four.

(b) *the triple [x, x, x.]*:- Marry the first eight symbols of t with the sequence xxxxxxxx.

However, so that these triple sequences T will dovetail properly with the pair sequences, we impose the condition that the triple sequence T ends with the 'all b's' tower $a^{x_1}ba^{x_2}b^{x_3}b$. Then t is cbcccbbb(repeated) or cccbcbbb(repeated).

Following Algorithm 6.11, the corresponding sequence begins $a^{3n}ca^{3n-1}b$ so that the addresses a^{3n}, $a^{3n-1}c$, ..., $a^{2n}c$, $a^{2n-1}ca$, ..., ca^{2n}, $a^{3n-1}b$, ..., $a^{2n}b$ are included. Since, the limits for j and i_1 are the same as Algorithm 5.52, all the pairs and triples will be included provided the dovetailing between different T sequences is correct. Analogously to the case r = 2 above, the insertion of a sequence T for the pair [2n - 1 - j, j.] ensures the inclusion of every associated tower apart from $a^jba^{2n-j-1}b$. Further, when this is followed by a sequence T for the next pair, [2n - j - 2, j + 1.], the previously omitted tower will now be included. However, it is possible that the pair is followed by one or more triples before the insertion of the next pair. For example, let T_1 denote the dovetailed sequence for the pair $[2n - j - 1, j.]$ and suppose it is followed by T_2, the sequence corresponding to the triple $[n - j - i_1 - 1, i_1, j.]$. The last address completely contained in T_1 is $a^{n-j}ba^jb$ and the final $n + 1$ symbols provide the starting points for the 3-run addresses $a^{n-j-1}ba^jba$, ..., $a^{i_1+1}ba^jba^{n-j-i_1-1}$ and the towers $a^{i_1}ba^jba^{n-j-i_1-1}c$ and $a^jba^{n-j-i_1-1}ca^{i_1}c$, each of which extend into T_2. This dovetailing explains why T for the pair $[x_1, x_2.]$ must end with $a^{x_1}ba^{x_2}b$. The other towers for the triple $[n - j - i_1 - 1, i_1, j.]$ are found in T_2.The last address completely contained in T_2 is $ba^{i_1}ba^jb$ and the remaining $i + j + 2$ symbols are the starting points for addresses which continue into the next part of the sequence. The next part of the sequence will correspond either to the triple $[n - j - i_1, i_1 - 1, j.]$ or to the next pair $[2n - j -2, j + 1.]$ depending on the value of n, j, i_1.

Firstly suppose the sequence T_2 is followed by the sequence T_3 with associated triple $[n - j - i_1, i_1 - 1, j.]$. Then, the address $a^{i_1}ba^jba^{n-j-i_1}$ followed by the triples $a^{i_1-1}ba^jba^{n-j-i_1}c$ and $a^jba^{n-j-i_1}ca^{i_1-1}c$ begin in T_2 but end in T_3. The remaining towers for the triple $[n - j -i_1, i_1 - 1, j.]$ are found T_3.

Eventually, we reach a point where there are no more triples to be inserted for this value of j where upon j is increased by one and the symbol 'a' together with the pair sequence for $[2n - j - 2, j + 1.]$ follows the sequence for the triple $[n - j - i_1 - 1, i_1, j.]$. Then the 3-run addresses $a^{i_1}ba^jba^{n-j-i_1-1}$, ..., ba^jba^{n-j} along with the tower $a^jba^{2n-j-1}b$ are now included, and so on.

Hence the dovetailing between the subsequences works. Finally, with $j = (2n - 1)$ div 2, the addition of the corresponding sequence T means that every tower except $a^{(2n-1) \text{ div } 2}ba^{2n-1-(2n-1) \text{ div } 2}b$ has now been included. Consequently, the subsequence $a^{2n-1-(2n-1) \text{ div } 2}b$ is added to the end. This ensures the inclusion of the final tower and provides starting points for the remaining addresses, namely the 2-run addresses $a^{(2n-1) \text{ div } 2}ba^{2n-(2n-1) \text{ div } 2}$, ..., ba^{2n}.

6.1.2.3 Model (6.1) - The general case r ≥ 2, n ≥ 2

Template 5.53 may be extended to model (6.1). The initial *Power*('a', n) must be replaced by the command *SetChar*(rn, 'c') followed by *Power*('a', n-1). The limits for j, i_1, i_2, ... remain unchanged. However, each command relating to a k-tuple must be replaced by the corresponding k-tuple sequence T command. We impose the condition that for each k-tuple $[x_1, ..., x_k.]$, apart from the pair $[x, x.]$ (see Case(b)(i) below), the corresponding sequence T ends with the 'all b's' tower $a^{x_1}ba^{x_2}b ... a^{x_k}b$. Finally, so that the tower $a^jba^{rn-n-j-1}b$ and the 2-run addresses a^jba^{rn-n-j}, $\ldots$, ba^{rn-n}, with $j = ((r - 1)n - 1)$ div 2, are included, the command *SetChar*$((r - 1)n - 1 - ((r - 1)n - 1)$ div 2, 'b') is necessary for all r, n.

To form the k-tuple sequence T, we require a periodic sequence t of period length 2^k containing every k-tuple over {b, c} exactly once in each period - this may be formed via a primitive polynomial over the finite field GF(2) [8] or by combinatorial means [10]. Then t is used to form T in one of the following ways depending on the least integer $m \geq 1$ such that $[x_1, ..., x_k\bullet]$ equals $[x_1, \ldots, x_m, \ldots, x_1, \ldots, x_m\bullet]$ for some m and on whether k is a divisor of 2^k:-

Case (a) m = k

(i) *k is a divisor of 2^k*:- Marry the first 2^k symbols of the sequence t with the sequence $x_1\ldots x_k \ldots x_1\ldots x_k$ of length 2^k (i.e. $2^k/k$ repetitions of the sequence $x_1\ldots x_k$). Then marry symbols 2...(2^k + 1) of sequence t with a second copy of the sequence $x_1\ldots x_k \ldots x_1\ldots x_k$ of length 2^k, ..., finally marry symbols k...(2^k + k - 1) of sequence t with a kth copy of the sequence $x_1\ldots x_k \ldots x_1\ldots x_k$ of length 2^k (see Example 6.8 below).

(ii) *k is not a divisor of 2^k*:- Marry the first $k2^k$ symbols of t in order with 2^k repetitions of the sequence $x_1\ldots x_k$.

Case(b) $1 \leq m \leq k - 1$

(i) *k is a divisor of 2^k*:- Marry symbols i...(2^k + i - 1) of t with the sequence $x_1\ldots x_m \ldots x_1\ldots x_m$ of length 2^k for i = (k - m + 1), ..., k in that order.

Note For the pair [x, x•] marry the first 2^k symbols of t with the sequence x...x of length 2^k. Then the sequence T for this pair begins and ends with a^xb_1 so that the dovetailing is correct.

(ii) *k is not a divisor of 2^k*:- Marry the first $m2^k$ symbols of t in order with 2^k repetitions of the sequence $x_1\ldots x_m$.

Note Since T ends with $a^{x_1}ba^{x_2}b \ldots a^{x_k}b$, for case (a)(i) and (b)(i) t must begin with (k - 1) b's and end in b while for case(a)(ii) and (b)(ii) t must end with k b's

Due to the notation, the procedure to obtain the sequence T may appear to be quite complicated. We now give an example to illustrate how simple it is.

Example 6.8 k = 4. Here t = bbbccccbbcbccbcb (repeated) and there are three different cases for the corresponding k-tuples:-

(a) *m = 4* :- the k-tuple $(x_1, x_2, x_3, x_4\bullet)$ with x_i not all equal $(1 \leq i \leq 4)$.

The first 16 symbols of t are married with 4 copies of the sequence $x_1x_2x_3x_4$, then symbols 2 to 17 of t are married with another 4 copies of $x_1x_2x_3x_4$, symbols 3 to 18 of t are married to another 4 copies of $x_1x_2x_3x_4$ and finally symbols 4 to 18 of t are married to the last 4 copies of $x_1x_2x_3x_4$. Our grouping is then

$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$
b b b c	c c c b	b c b c	c b c b	b b c c	c c b b	c b c c	b c b b
$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$	$x_1x_2x_3x_4$
b c c c	c b b c	b c c b	c b b b	c c c c	b b c b	c c b c	b b b b

so that $T = a^{x_1}ba^{x_2}ba^{x_3}ba^{x_4}c \ldots a^{x_1}ba^{x_2}ba^{x_3}ba^{x_4}b$.

(b) $m = 2$:- the k-tuple $(x_1, x_2, x_1, x_2 .)$ with $x_1 \neq x_2$.

$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$	$x_1x_2x_1x_2$
b c c c	c b b c	b c c b	c b b b	c c c c	b b c b	c c b c	b b b b

so that $T = a^{x_1}ba^{x_2}ca^{x_1}ca^{x_2}c \ldots a^{x_1}ba^{x_2}ba^{x_1}ba^{x_2}b$.

(c) $m = 1$:- the k-tuple $(x, x, x, x.)$

x x x x	x x x x	x x x x	x x x x
c c c c	b b c b	c c b c	b b b b

so that $T = a^x ca^x ca^x ca^x c \ldots a^x ba^x ba^x ba^x b$.

Notation 6.9 In the following algorithms, various portions of the optimal sequence are generated by the routines listed below.

(i)*Pair Seq T*(x_1, x_2) represent the appropriate sequence T as described above for the pair $[x_1, x_2.]$,

(ii) *k-tuple Seq T*$(x_1, \ldots, x_k)$ represent the appropriate sequence T for the k-tuple $[x_1, \ldots, x_k.]$ $(k \geq 3)$,

(iii) *Insert k-tuples Seq T*(k, r) be the recursive subroutine $(k \geq 3)$:-

```
begin
    if $(n(r - k + 1) = kj + 1)$ and $((k = 3)$ or $((k > 3)$ and $(i_1 = \ldots = i_{k-3} = j)))$ then
        k-tuple Seq T$(j, \ldots, j)$                                {k-tuple $[j, \ldots, j.]$}
    for $i_{k-2}$:= up(k, r) downto down(k, r) do
        begin
            k-tuple Seq T$(n(r - k + 1) - j - 1 - \sum_{s=1}^{k-2} i_s, i_{k-2}, \ldots, i_1, j)$        {k-tuples}
            if $k < r$ then
                Insert k-tuple Seq T$(k + 1, r)$
        end
end
```

where *up*(k, r) and *down*(k, r) are suitably chosen to ensure each k-tuple is inserted exactly once.

In summary, the algorithms for an optimal sequence of model (6.1) are very similar to Algorithms 5.50, 5.51 and 5.52 except that each command relating to a k-tuple must be replaced by the corresponding k-tuple sequence T command. Further, we must insert $a^{nr}c$ at the start and add $a^{(r - 1)n - 1 - (((r - 1)n - 1) \text{ div } 2)}b$ to the end.

Algorithm 6.10 - produces an optimal sequence for r = 2, n ≥ 2 for any IFS satisfying (6.1).

```
begin
    Set Char(2n, 'c')
    Power('a', n - 1)
    for j := 0 to (n - 1) div 2 do
        begin
            Power('a', 1)
            Pair Seq T(n - 1 - j, j)            {Pairs}
        end
    SetChar(n div 2, 'b')
end
```

Algorithm 6.11 - produces an optimal sequence for r = 3, n ≥ 2 for any IFS satisfying (6.1).

```
begin
    Set Char(3n, 'c')
    Power('a', n - 1)
    for j := 0 to (2n - 1) div 2 do
        begin
            Power('a', 1)
            Pair Seq T(2n - 1 - j, j)     {Pairs}
            Insert k-tuple Seq T(3, 3)
        end
    SetChar(2n - 1 - ((2n - 1) div 2), 'b')
end
```

Algorithm 6.12 - produces an optimal sequence for r = 4, n ≥ 2 for any IFS satisfying (6.1).

```
begin
    Set Char(3n, 'c')
    Power('a', n - 1)
    for j := 0 to (3n - 1) div 2 do
        begin
            Power('a', 1)
            Pair Seq T(3n - 1 - j, j)     {Pairs}
            Insert k-tuple Seq T(3, 4)
        end
    SetChar(3n - 1 - ((3n - 1)/2), 'b')
end
```

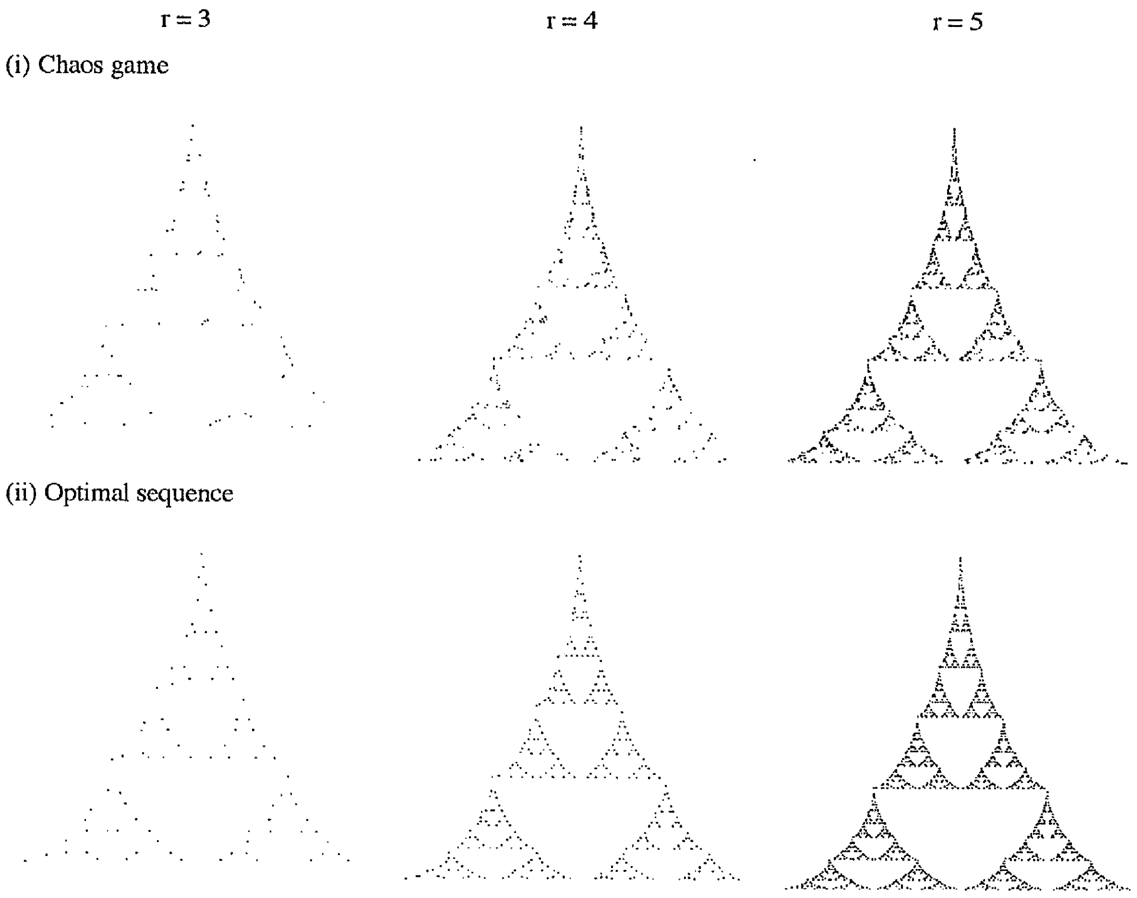
Note In a similar manner, the template algorithm 5.53, may be extended to model (6.1).

up(k, r) and down(k, r) are as given in Algorithms 5.50, 5.51 and 5.52.

6.1.3 The results

We now compare the optimal sequence method and the chaos game (with RandomX) for non-uniform IFS satisfying model (6.1).

Example 6.13 The *Eiffel Tower* IFS has three affine transformations w_a, w_b, w_c of respective ratios 0.7, 0.5, 0.5. Then this IFS fits model (6.1) with n = 2. The images produced by the chaos game and the optimal sequence method are given in Figure 6.1 for r = 3, 4, 5.



r = 3 r = 4 r = 5

(i) Chaos game

(ii) Optimal sequence

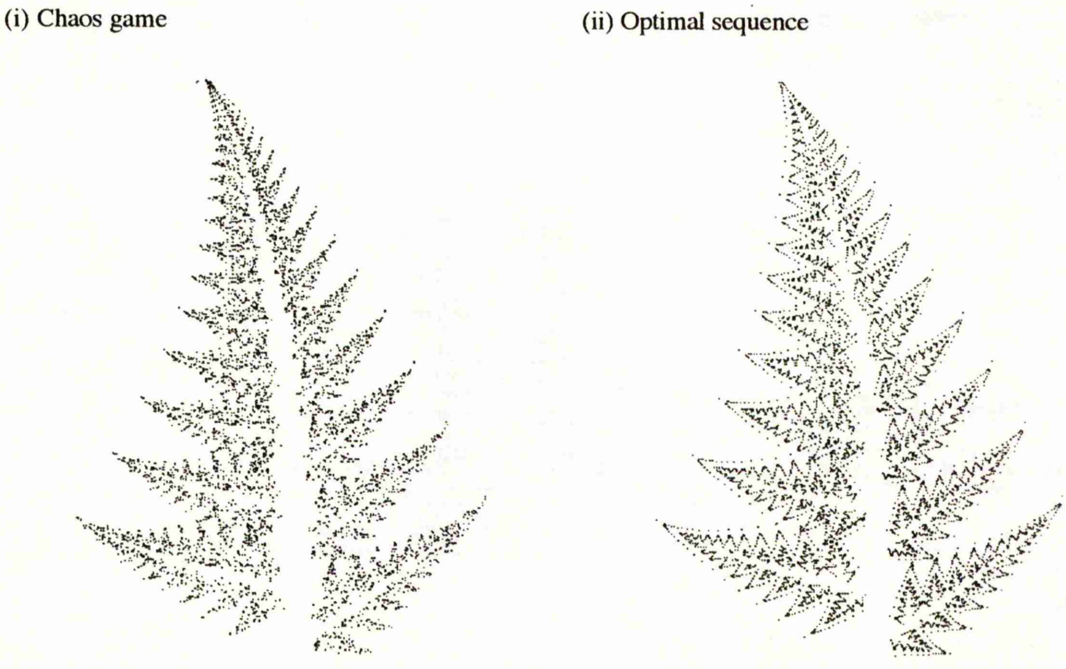
Figure 6.1 *Eiffel Tower* produced by (i) an optimal sequence and (ii) the chaos game for r = 3, 4, 5.

Although for r = 3, both methods plot 100% original points, the optimal sequence ensures that the points are well distributed in the attractor. In contrast, the positioning of the chaos game points is random. For example, the bottom right corner of this chaos game approximation contains very few points. As r is increased, the difference between these methods becomes more apparent. In particular, for r = 5, the optimal sequence and the chaos game plot 97.9% and 81.6% original points respectively. Moreover the optimal sequence method is faster and the image produced is sharper.

Attractor	map	a	b	c	d	e	f	probability p
The Eiffel	w_a	0.5	0	0	0.7	0.25	0.3	0.548
Tower	w_b	0.5	0	0	0.3	0	0	0.231
(Fig 6.1)	w_c	0.5	0	0	0.3	0.5	0	0.231

Table 6.1 The code and probabilities p for the *Eiffel Tower* [21].

Example 6.14 *The Fern without stem.* Here we have three affine transformations w_a, w_b, w_c with respective ratios 0.846, 0.387, 0.373. So we set $r_b = r_c = 0.387$ and choose n to be the least positive integer such that $(0.846)^n \leq 0.387$, namely $n = 6$. Setting $r = 4$, the result of an optimal sequence is compared with that of the chaos game (see Figure 6.2).


(i) Chaos game

(ii) Optimal sequence

Figure 6.2 *Fern (without stem)* produced by (i) the chaos game and (ii) an optimal sequence.

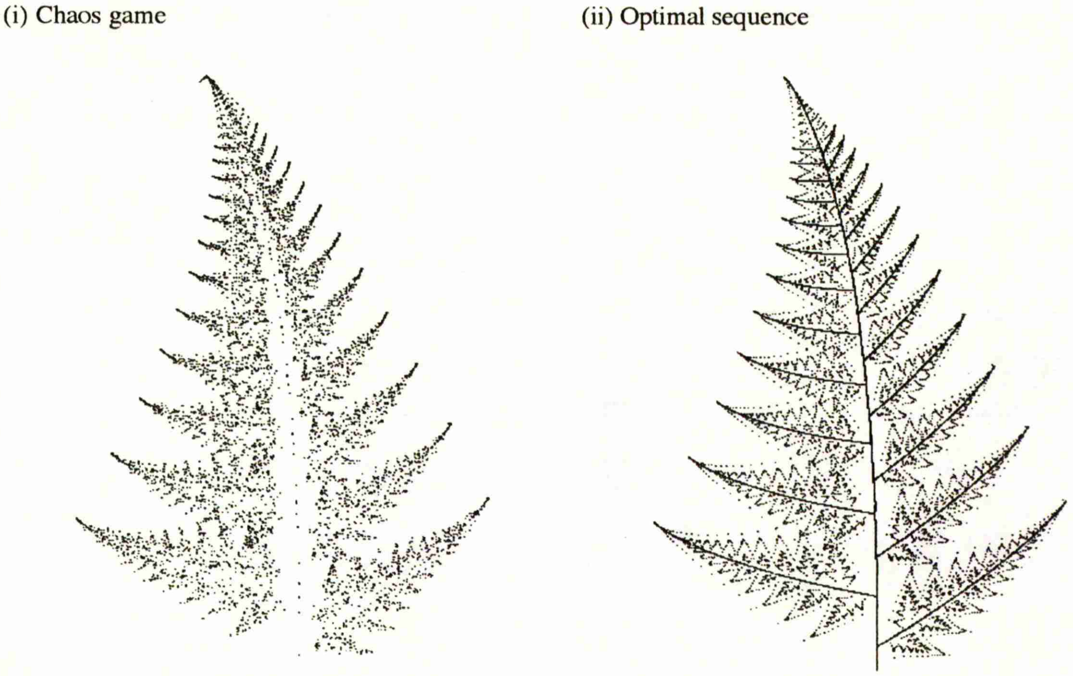
Note Although each image involves 5664 iterations of the RIA, the optimal sequence image shows finer detail.

The optimal sequence produced fern gives a clear impression of the final attractor while the chaos game version is very fuzzy and lacks detail. The optimal sequence version plots 96.2% original points compared with 89.7% for the chaos game. Moreover, the optimal sequence takes less time to produce the image (cf Section 7.1).

Attractor	map	a	b	c	d	e	f	probability p
The Fern	w_a	0.821	-0.028	0.03	0.845	0.088	0.176	0.753
(Fig. 6.2,	w_b	0.076	0.312	-0.257	0.204	0.494	0.133	0.112
6.3)	w_c	-0.024	-0.356	-0.323	0.074	0.470	0.260	0.135
The Stem	w_a	0.821	-0.028	0.03	0.845	0.088	0.176	0.99
(Fig. 6.3)	w_d	0	0	0	0.172	0.496	-0.091	0.01

Table 6.2 The code and probabilities p for the *fern* and the *stem.*

Example 6.15 *The Fern with stem and substems.* In [15], we added a stem to the fern by introducing a 4th affine transformation w_d such that the fern is the attractor of $\{w_a, w_b, w_c\}$ where r_a, r_b, r_c fit model (6.1) and the stem is the attractor of $\{w_a, w_d\}$ where r_a, r_d satisfy (5.1). We produced this fern with a stem, by running the optimal sequence method corresponding to (6.1) for w_a, w_b, w_c, resetting the initial point and running the optimal sequence corresponding to (5.1) for w_a, w_d. The image obtained was far superior to that of the chaos game. However, we wish to enhance this image further by adding substems to each of the branches of the fern. The fern may be represented by the collage $w_a(\mathcal{A})$, $w_b(\mathcal{A})$ and $w_c(\mathcal{A})$ with contributions w_a : the main body from the top down excluding the lower two leaves; w_b : the lower left leaf; w_c : the lower right leaf. Each time a point is plotted for the main stem, we wish to plot an equivalent point for some of the stems for the left and right branches. For example, suppose x lies on the stem, then the points $w_b(x)$, $w_c(x)$ will lie on the lower left and lower right substems respectively. Similarly, $w_aw_b(x)$, $w_aw_c(x)$ will be points of the second bottom left and right substems. We may extend this idea so that each of the first $(\alpha + 1)$ branches, starting at the bottom, will have substems simply by plotting $w_b(x)$, $w_aw_b(x)$, ..., $w_a^{\alpha}w_b(x)$ (where w_a is applied α times), $w_c(x)$, $w_aw_c(x)$, ..., $w_a^{\alpha}w_c(x)$ for each point x plotted on the main stem. We produced the images below by plotting substem points after every five points plotted on the main stem and with $\alpha = 11$. These parameters depend on the amount of detail required. We set $r = 4$ with $n = 10$ for the body of the fern (to enhance the detail) and $n = 11$ for the stem. Looking at the images produced, we see that the main stem in the chaos game version is represented only by a small number of dots while the substems are almost indistinguishable. In comparison the optimal sequence gives a good approximation of the attractor.


(i) Chaos game

(ii) Optimal sequence

Figure 6.3 *Fern with stem and substems* for (i) the chaos game and (ii) an optimal sequence.

Note To produce the body of the fern involves 25704 iterations of the RIA.

6.2 A more general model

So far we have restricted ourselves to N = 2, 3. However, using the techniques of the previous section, we can produce optimal sequences for the more general IFS with N maps w_a, w_{b_1}, ..., $w_{b_{N-1}}$ of respective ratios r_a, r_{b_1}, ..., $r_{b_{N-1}}$ and satisfying

$$r_a^n = r_{b_i} \qquad r_{b_i}^r = \varepsilon \qquad \text{for } i = 1, \dots, N - 1. \qquad (6.2)$$

We assign the weights wt(a) = 1, wt(r_{b_i}) = n. An address tree may be used to list the addresses. However, explicitly the addresses for $r \geq 1$, $n \geq 2$ are

(i) those corresponding to the nodes of $\Gamma_{r,n}$ except that b_1 is used instead of b,

(ii) those obtained from (i) when every b_1 is alternatively replaced by b_2, ..., b_{N-1},

and consequently the total number of addresses is

$$\sum_{k=0}^{r} (N - 1)^k \binom{n(r - k + 1) + k - 1}{k} .$$

We may produce optimal sequences (which are cyclic and whose reverse is also optimal), by modifying Algorithms 5.50, 5.51, 5.52 in a similar way to that described in Section 6.1.2. The alterations to the algorithms and the formation of the k-tuple sequences

T depend on whether N is odd or even. We require a periodic sequence t of period length $(N - 1)^k$ containing every k-tuple over $\{b_1, ..., b_{N-1}\}$. When (N - 1) is a prime power, we may use finite fields to produce the sequence t. Otherwise alternative methods include graph theory (see Section 3.2) and *k-recurrent latin squares* (see Section 3.3.2). Analogously to Section 6.1.2.3, the way in which T is formed from t depends on the least integer $m \geq 1$ such that the k-tuple $[x_1, ..., x_{k\bullet}]$ equals $[x_1,...., x_m, ..., x_1, ..., x_{m\bullet}]$ and on whether k is a divisor of $(N - 1)^k$. In fact, the method to obtain a k-tuple sequence T is simply the method described in Section 6.1.2.3 where each 2^k is been replaced with $(N - 1)^k$. However, to ensure that the T sequences dovetail correctly, we specify that T ends with the 'all b_1's' tower $a^{x_1}b_1...a^{x_k}b_1$. Moreover, if N is even then T must begin $a^{x_1}b_{N-1}$. Finally, the exception noted in Section 6.1.2.3 for the pair $[x, x_\bullet]$ is only valid when N is odd. The template algorithms are given below.

Template Algorithm 6.16 - produces an optimal sequence for model (6.2) where N even, $r \geq 2$, $n \geq 2$

```
begin
  SetChar(rn, 'b_2')
  SetChar(rn - 1, 'b_3')
          .
          .
  SetChar(rn - 1, 'b_{N-2}')
  SetChar(rn - 1, 'b_1')
  Power('a', n - 1)
  for j := 0 to ((r - 1)n - 1) div 2 do
    begin
      Power('a', 1)
      Pair Seq T((r - 1)n - j -1, j)
      Insert k-tuple Seq T(3, r)
    end
  SetChar((r - 1)n - ((r - 1)n - 1) div 2), 'b_{N-1}')
end
```

Template Algorithm 6.17 - produces an optimal sequence for model (6.2) where N odd, $r \geq 2$, $n \geq 2$

```
begin
  SetChar(rn, 'b_2')
  SetChar(rn - 1, 'b_3')
          .
          .
  SetChar(rn - 1, 'b_{N-1}')
  Power('a', n - 1)
  for j := 0 to ((r - 1)n - 1) div 2 do
    begin
      Power('a', 1)
      Pair Seq T((r - 1)n - j -1, j)
      Insert k-tuple Seq T(3, r)
    end
  SetChar((r - 1)n - ((r - 1)n - 1) div 2),' b_1')
end
```

Note The values of *up*(k,r) and *down*(k,r) are as for the algorithms of Chapter 5.

Example 6.18 *The Snowflake.* We have $r_a^5 \le r_{b_1} = r_{b_2} = r_{b_3}$ (i.e. model (6.2) with n = 5). The images produced by the chaos game and the optimal sequence method for r = 2, 3, 4 are given in Figure 6.4. The optimal sequence produces a higher percentage of original points in each case. The optimal sequence images are superior since they display more structure and detail.

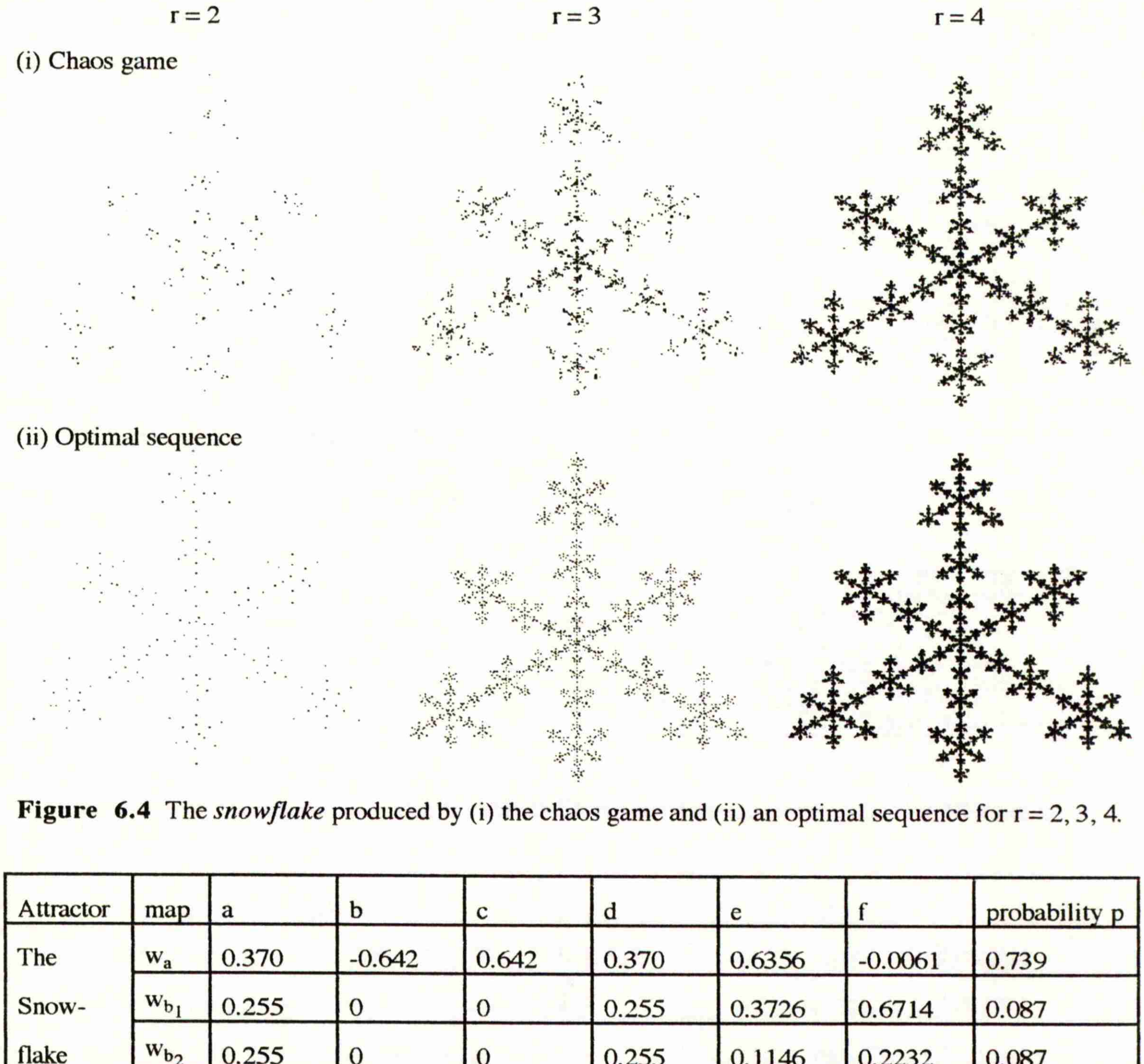


Figure 6.4 The *snowflake* produced by (i) the chaos game and (ii) an optimal sequence for r = 2, 3, 4.

Attractor	map	a	b	c	d	e	f	probability p
The Snowflake (Fig 6.4)	w_a	0.370	-0.642	0.642	0.370	0.6356	-0.0061	0.739
	w_{b_1}	0.255	0	0	0.255	0.3726	0.6714	0.087
	w_{b_2}	0.255	0	0	0.255	0.1146	0.2232	0.087
	w_{b_3}	0.255	0	0	0.255	0.6306	0.2232	0.087

Table 6.3 The code and probabilities p for the *snowflake* [28].

Example 6.19 The *Pine Tree.* Again, this IFS fits model (6.2) with n = 5. For, r = 3, the chaos game and the optimal sequence plot 96.4% and 97.4% original points respectively. However, the quality of the corresponding images differs greatly (see

Figure 6.5). Notice, the stem and main branches of the optimal sequence pine tree are clearly displayed whereas in contrast, the chaos game image is unstructured. For r = 4, the chaos game plots significantly more original points than the optimal sequence method (i.e. 81.4% compared to 70.0%), but the optimal sequence image has more structure and detail. For example, the sub-branches are clear in the optimal sequence approximation, while even the main branches are fuzzy in the chaos game image.

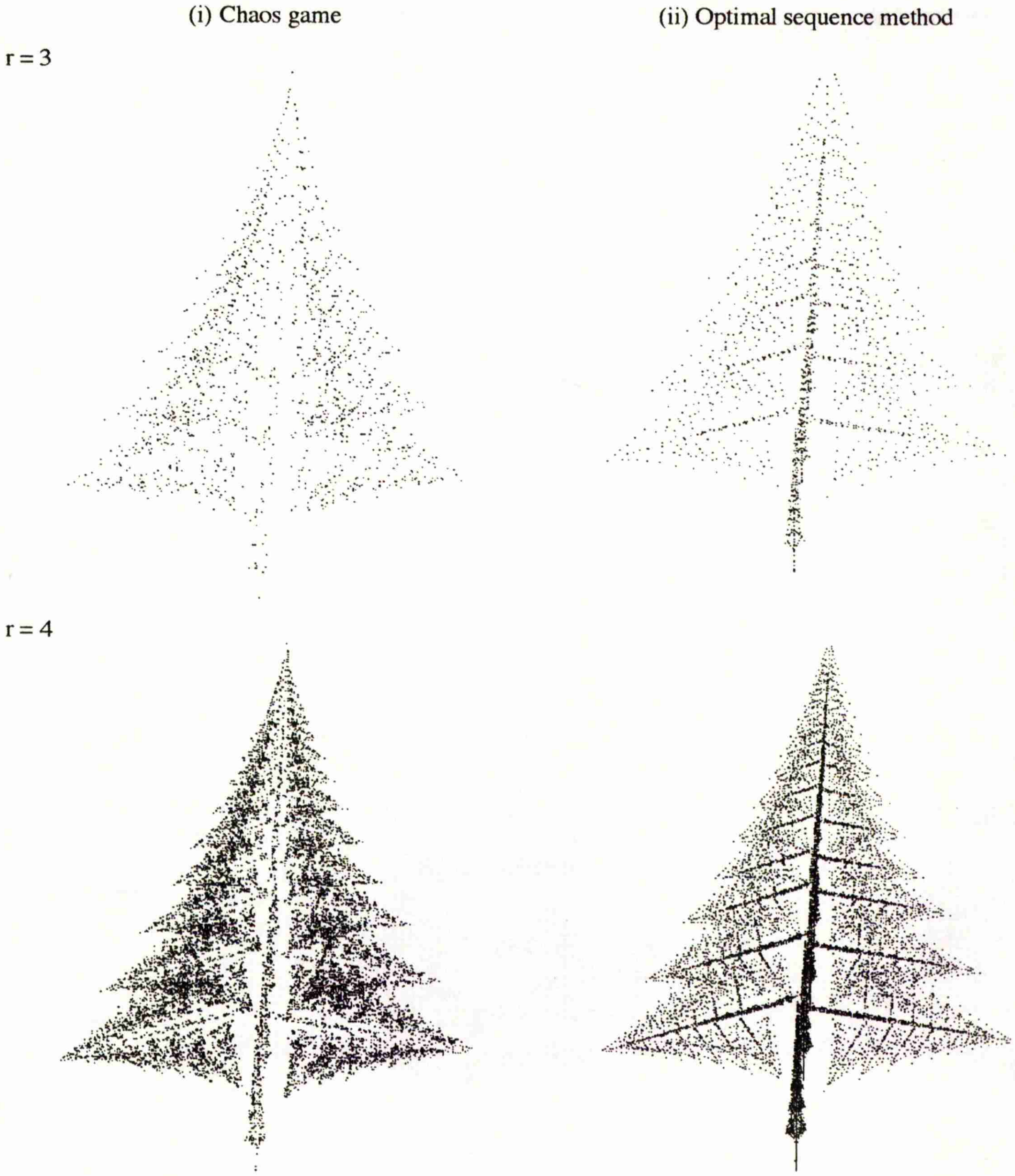
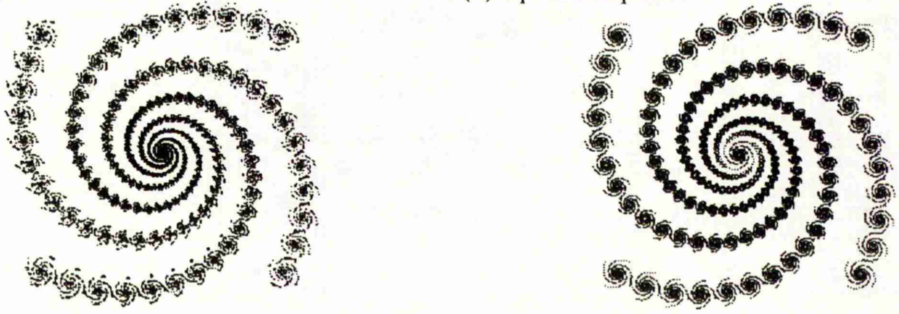


Figure 6.5 The *pine tree* produced by (i) the chaos game and (ii) an optimal sequence for r = 3, 4.

Attractor	map	a	b	c	d	e	f	probability p
The Pine	w_a	0.798	0	0	0.833	0.098	0.169	0.651
Tree	w_{b_1}	-0.132	-0.367	0.399	-0.141	0.5	0.181	0.162
(Fig 6.5)	w_{b_2}	0.100	0.404	0.408	-0.100	0.365	0.158	0.171
	w_{b_3}	0.049	0	0	0.333	0.399	0.010	0.016

Table 6.4 The code and probabilities p for the *Pine tree* [21].

Example 6.20 The *Spiral 2*. This IFS consists of one map of ratio 0.95 (w_a) and four, each of ratio 0.1 (w_{b_1}, w_{b_2}, w_{b_3}, w_{b_4}). For $r = 2$, $n = 45$, we compare the results of the chaos game and the optimal sequence method (see Figure 6.6). The optimal sequence method is faster, plots a higher percentage of original points (56.6% original points for the optimal sequence method compared with 46.3% for the chaos game), and the corresponding image displays more detail, particularly in the outer layers.



(i) Chaos game

(ii) Optimal sequence

Figure 6.6 The *spiral 2* produced by (i) the chaos game and (ii) an optimal sequence for $r = 2$.

Attractor	map	a	b	c	d	e	f	probability p
The	w_a	0.936	-0.165	0.165	0.936	0	0	0.9576
Spiral 2	w_{b_1}	0.100	0	0	0.100	0.750	-0.750	0.0106
(Fig 6.6)	w_{b_2}	0.100	0	0	0.100	-0.750	0.750	0.0106
	w_{b_3}	0.100	0	0	0.100	0.750	0.750	0.0106
	w_{b_4}	0.100	0	0	0.100	-0.750	-0.750	0.0106

Table 6.5 The code and probabilities p for the *spiral 2* [21].

We have illustrated how the algorithms of Chapter 5 can be easily adapted using M-sequences to other non-uniform IFS. Again, the optimal sequence results are superior to the chaos game. We also illustrated how optimal sequences may be spliced for more complicated images.

Chapter 7 Some further results and comparisons

7.1 Speed and accuracy

We previously observed the optimal sequence method offers the accuracy of the ACM combined with the speed of the chaos game and the resulting image is independent of the starting point x_0 (cf Section 3.4). Moreover, this optimal sequence image generally shows more structure than is guaranteed by the tolerance ε - the *free ride effect.* We will now study these observations in more detail.

7.1.1 Speed

Recall, both the chaos game and the optimal sequence method plot a sequence of points $\{x_n\}$ where $x_{n+1} = w(x_n)$. The difference between the two algorithms is the way in which the map w is determined at stage n. We have shown the optimal sequence method to be more accurate than the chaos game. In our examples, the optimal sequence also proved to be faster than the chaos game with a RandomX based RNG. Further, since this RNG is fairly simple to compute, it is likely that the optimal sequence method is usually faster than the chaos game with any reasonable RNG driver.

In contrast, both the optimal sequence method and the ACM guarantee the image to the same degree of accuracy. However, to produce the ACM approximation involves significantly more work. In Section 2.4, we stated a general recursive algorithm which for a given $\varepsilon > 0$, produces the ACM approximation of any IFS $\{w_{1\text{-}N}\}$. Below, we give explicit ACM algorithms for the uniform IFS $\{w_{1\text{-}N}\}$ and the non-uniform IFS $\{w_a, w_{b_1}, \ldots, w_{b_{N-1}}\}$ satisfying model (6.2) (see Algorithm 7.1 and 7.2 respectively). Then, for $\varepsilon > 0$, the ACM requires the computation of the code (a, b, c, d, e, f) for the composition of affine transformations corresponding to each node of the address tree. To compare the speed of these algorithms we must therefore find the ratio of addresses to nodes in the address tree. Table 7.1 summarises the calculations involved in each method.

Algorithm 7.1 The uniform IFS $\{w_{1\text{-}N}\}$. Let k be the smallest positive integer such that $\varepsilon \leq s^k \operatorname{diam}(\mathcal{A})$, $x_0 \in \mathcal{A}$ and let *subdivide* be a recursive algorithm which plots the point $w(x_0)$ if w is the composition of exactly k affine transformations from $w_1, \ldots, w_N$. Otherwise, it calls *subdivide*(ww_1), ..., *subdivide*(ww_N). Let Id denote the identity affine transformation with code (1, 0, 0, 1, 0, 0) and we take Id to be the composition of zero

affine transformations from w_1, ..., w_N. Then the ACM consists simply of calling *subdivide*(Id).

Algorithm 7.2 The non-uniform IFS $\{w_a, w_{b_1}, ..., w_{b_{N-1}}\}$ with integers $r \geq 1$, $n \geq 2$ satisfying model (6.2). For $x_0 \in \mathcal{A}$, let *subdivide* be a recursive algorithm which plots the point $w(x_0)$ if $wt(w) \geq nr$. Otherwise it calls *subdivide*(ww_a), subdivide(ww_{b_1}), ..., subdivide($ww_{b_{N-1}}$) where $wt(ww_i) = wt(w) + wt(w_i)$. Let Id be the identity affine transformation and we set wt(Id) = 0. Then the ACM consists of calling *subdivide*(Id).

Chaos Game	ACM	Optimal Sequence Method
One w(x) for each point	One w(x) for each point	One w(x) for each point
Determine each successive map by $z_{n+1} = (az_n + b) \bmod m$ and probabilities.	Calculate the composition of two affine transformations for each node of the address tree.	Determine each successive map as follows (i) *uniform IFS*:- from a kth-order linear recurrence relation (ii) *non-uniform IFS* :- from an optimal sequence produced by the algorithms of Chapters 5 and 6.

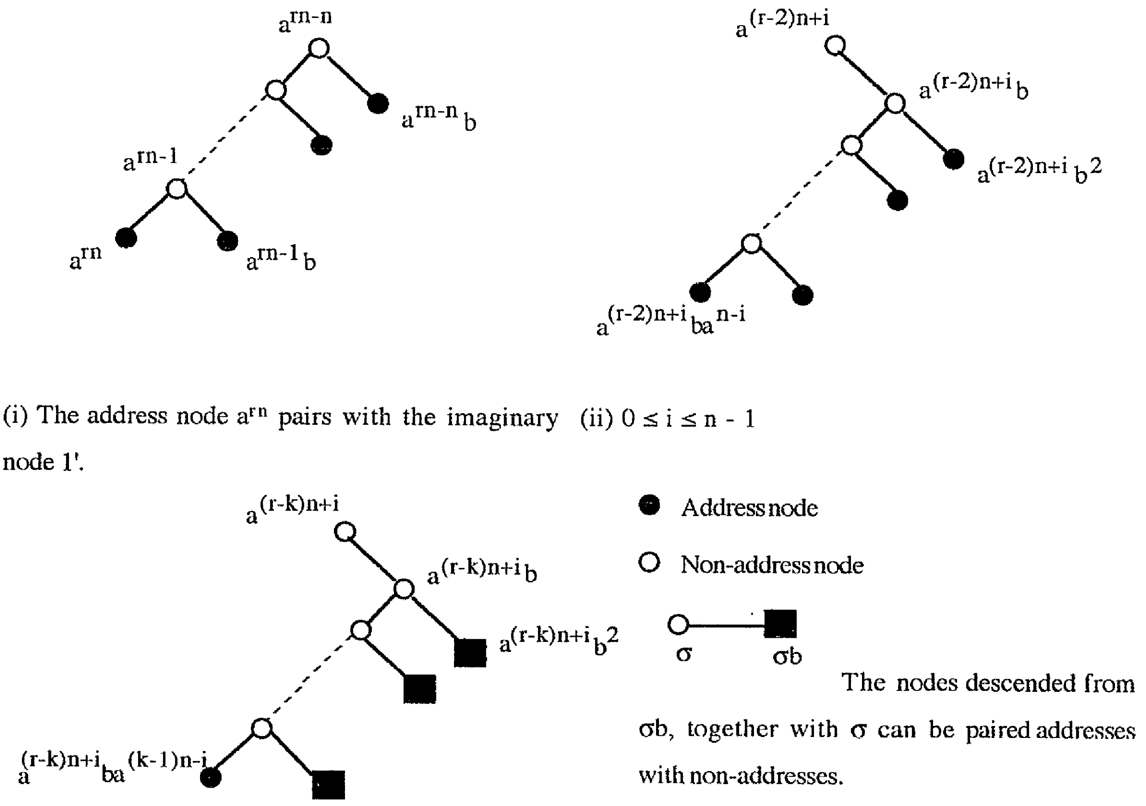
Table 7.1 Summary of the calculations of (i) the chaos game, (ii) the ACM and (iii) the optimal sequence method.

Lemma 7.3 In the address tree of the uniform IFS $\{w_{\sigma_1\text{-}\sigma_N}\}$, the addresses are approximately the fraction (N - 1)/N of the nodes.

Proof Recall the addresses are the N^k k-digit words over $\{\sigma_1, ..., \sigma_N\}$ and the address tree has k levels below the node 1. The ith level has N^i nodes corresponding to the N^i i-digit words over $\{\sigma_1, ..., \sigma_N\}$. Then, the N^k addresses are the N^k nodes of level k and the ratio of addresses to nodes is $N^k/(1 + N + N^2 + ... + N^{k-1}) \approx (N - 1)/N$.

Lemma 7.4 In the address tree for the non-uniform IFS $\{w_a, w_b\}$ with least positive integers $r \geq 1$, $n \geq 2$ satisfying model (5.1), counting one imaginary node 1' (besides 1 itself), one half of the nodes are addresses.

Proof The idea is to pair each address node with a non-address node. Figure 7.1 gives the basis for this.



(i) The address node a^{rn} pairs with the imaginary node 1'.

(ii) $0 \leq i \leq n - 1$

(iii) $3 \leq k \leq r$, $0 \leq i \leq n - 1$

The nodes descended from σb, together with σ can be paired addresses with non-addresses.

Case k = 3 Each such structure in (iii) is the same as (ii) where σ corresponds to $a^{(r-2)n+i}$.

Case 4 ≤ k ≤ r Each such structure in (iii) is recursively replaced by (iii) with k:=k-1 where σ corresponds to $a^{(r-k+1)n+i}$ until k = 3 when each structure is replaced as given for k = 3 above.

Figure 7.1 The pairing of address nodes with non-address node for $N = 2$, $r \geq 2$, $n \geq 2$.

Corollary 7.5 In the address tree of the non-uniform IFS $\{w_a, w_{b_1}, \ldots, w_{b_{N-1}}\}$ satisfying model (6.2), counting one imaginary node 1' (besides one itself), the addresses are the fraction (N - 1)/N of the nodes.

Proof Since the maps $w_{b_1}, \ldots, w_{b_{N-1}}$ are of equal ratio, the method of Figure 7.1 is easily adapted to associate (N - 1) addresses with every non-address node.

Let β be the number of addresses for any IFS covered by Lemmas 7.3, 7.4 or Corollary 7.5 above. Suppose that the computation of the affine transformations v.w takes

α times as long to compute as it does to calculate $w(x_0)$. Then we make the approximation (time for ACM)/(time for optimal sequence) = $(\alpha\beta N/(N - 1) + \beta)/\beta = \alpha + 1 + \alpha/(N - 1)$, so we would expect the optimal sequence to take at most 1/2 the time of the ACM. These expectations are confirmed by the results of Table 7.2. The optimal sequence method takes less time than the chaos game based on RandomX. Further, since this RNG is fairly simple to compute, it is likely that optimal sequences are usually faster than any reasonable RNG.

Attractor	Chaos game	ACM	Optimal Sequence Method
Sierpinski gasket ($k = 7$)	42	91	33
Ear ($r = 3$)	29	64	23
Spiral ($r = 3$)	119	274	96
Eiffel Tower ($r = 5$)	30	51	24
Fern (no stem) ($r = 4$)	122	233	101
Snowflake ($r = 4$)	285	481	229

Table 7.2 Time taken in seconds to produce various attractors using (i) the chaos game, (ii) the ACM and (iii) the optimal sequence method.

7.1.2 The choice of starting point

In Section 3.4, we observed that the performance of the ACM depends on the choice of the point $x_0 \in \mathcal{A}$. In contrast, the optimal sequence image is independent of the starting point x_0. For let x_0, x_0' be two different starting points which lead to the sequences $\{x_n\}$ and $\{x_n'\}$ for an optimal sequence. Then with ratio s, we have $|x_n - x_n'| \leq s^n |x_0 - x_0'|$ so that after a few iterations the sequences merge. Similarly, the performance of the chaos game is not affected by the choice of x_0.

Figure 7.2 gives the dragon approximations produced by the ACM and the optimal sequence method for each of two different starting points x_0, x_0' .

Notice for some values of k, the ACM image with x_0 has white horizontal and vertical lines which are not part of the true structure of the attractor while with x_0', the ACM image has very black patches which again could be wrongly interpreted as a feature of the attractor. However, when k is taken sufficiently large these faults disappear and the ACM produces a good approximation of the dragon. In comparison, the optimal sequence with starting points x_0, x_0', plots well distributed points for all k. Thus, with a smaller k, that is with fewer points, we may be confident that the optimal sequence image is correct. The same cannot always be said about the ACM image.

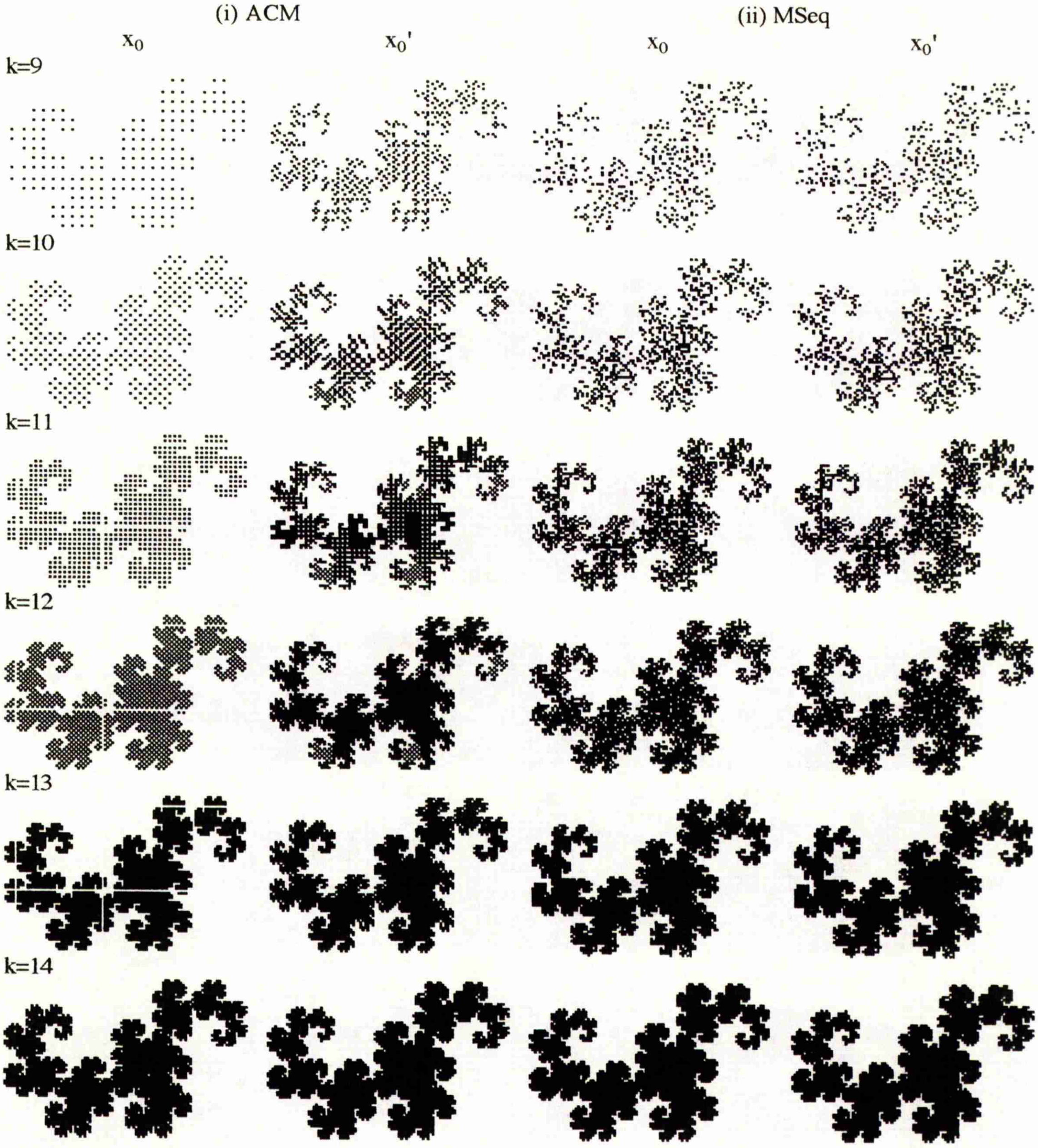


Figure 7.2 The *dragon* with starting points x_0, x_0' produced by (i) the ACM and (ii) the optimal sequence method.

Figure 7.3 shows some other rather odd approximations of (i) the dragon and (ii) the Peano curve that the ACM produces with various different choices of x_0. The corresponding optimal sequence images are not given since we have now concluded that this method is unaffected by the choice of x_0.

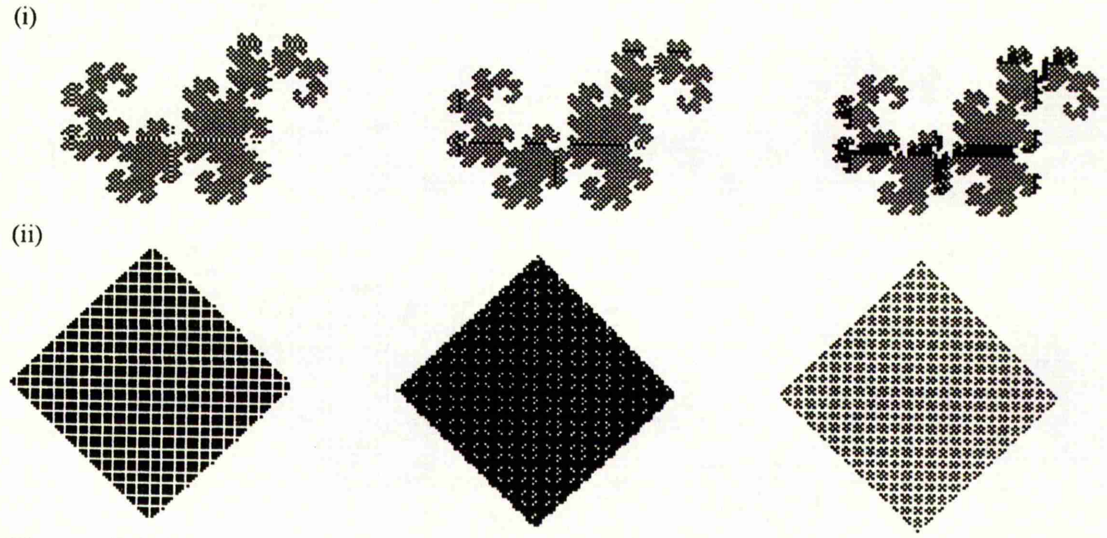


Figure 7.3 (i) The *dragon* produced by the ACM with various different points x_0 (k = 12) and (ii) The *Peano curve* produced by the ACM with various different points x_0 (k = 4).

The images of Figure 7.2 also illustrate the extra accuracy obtained by using an optimal sequence. This property which we describe as the *free ride effect* is explained in Section 7.1.3 below.

7.1.3 Accuracy - The free ride effect

In theory, both the ACM and optimal sequences offer the same degree of accuracy for a given number of iterations - since both methods plot a point in each partition of the attractor and the partitions are of diameter less than ε (Recall, these partitions correspond to the addresses. However, in practice the optimal sequence method provides an image of greater accuracy and, as observed above, this image is independent of the choice of starting point x_0 [14, 26]. We refer to this extra accuracy as the *free ride effect* and below we explain its presence using the fern as our example. But the idea may be simply extended to any attractor. Let $\mathcal{A}$ denote the fern attractor. Then recall $w_b(\mathcal{A})$, $w_c(\mathcal{A})$ are the lower left and lower right branches respectively (see Figure 7.4). Further $w_a w_b(\mathcal{A})$ is the second bottom left branch, $w_a w_a w_b(\mathcal{A})$ the third bottom left branch and so on. Let σ denote some (possibly long) part of an optimal sequence. Then σ is some feature of the fern, for example the tip. So that $b\sigma$ is the tip of the lower left branch, $ab\sigma$ the tip of the next left branch, $aab\sigma$ the next and so on. Hence ...a...$ab\sigma$ gives a sequence of correctly positioned features, in this case the tip of branches. In other words, optimal sequences preserve features and so the image produced generally shows more structure than is guaranteed by the tolerance ε. Thus, the extra detail in the image is due to a remarkable property of optimal sequences which plot points in or very close to the best position within each partition of diameter ε (see Figure 7.5).

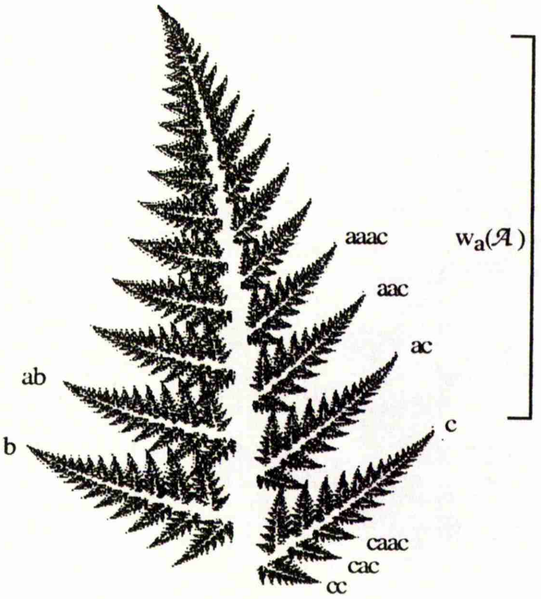


Figure 7.4 *Fern (without stem)* with some labelled branches.

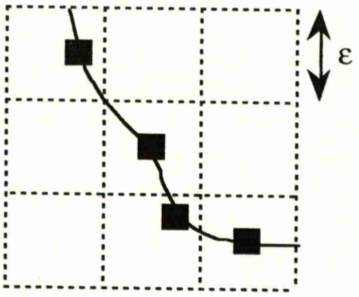
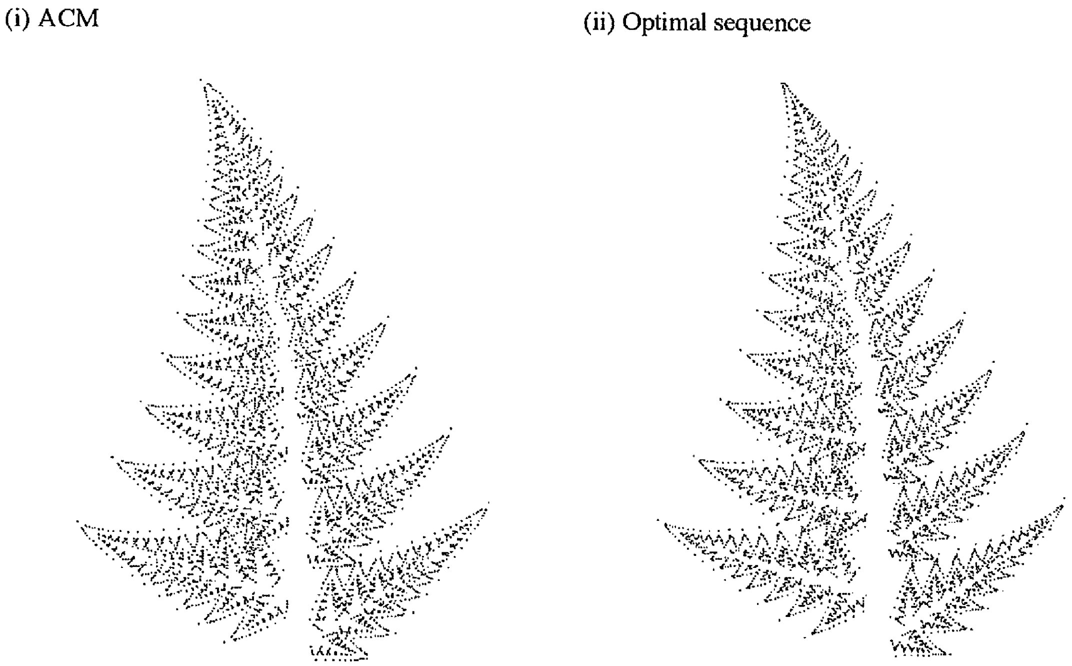


Figure 7.5 Illustrating the pixels lit by an optimal sequence.

In Figure 7.5, the background grid represents partitions of the attractor, $\mathcal{A}$, each of diameter ε. The black line shows the shape of part of $\mathcal{A}$. The pixels lit (black squares) by an optimal sequence within each partition are very close to this line. Thus there is more structure in the image produced.

Note Figure 7.5 is idealised in that subsets are not in general even rectangular.

Example 7.6 To illustrate the free ride effect, we set r = 4, n = 6 and produced *the fern (without stem)* using (i) the ACM and (ii) an optimal sequence. The results are given in Figure 7.6. Both involve the same number of iterations and plot one point in each partition. However, the image produced by an optimal sequence gives more detail - the structure of the subbranches, sub-subbranches etc is clearer in (ii). For example, comparing the lower left branch of each image, we see that the tip is more defined as are the tips of the subbranches. Moreover, the optimal sequence plots a higher percentage of original points and is significantly faster than the ACM.



(i) ACM

(ii) Optimal sequence

Figure 7.6 *Fern (without stem)* produced by (i) the ACM and (ii) an optimal sequence.

Example 7.7 *The tree* IFS has five affine transformations which we shall label w_a, w_{b_1}, w_{b_2}, w_{b_3}, w_{b_4} with respective ratios 0.664, 0.656, 0.525, 0.603, 0.479. Setting $\varepsilon = r_{b_i}$ $(i = 1, 2, 3, 4)$, we have n equal to two so that $r_a^2 \leq \varepsilon$. The images produced and the percentage of original points plotted by the chaos game, the ACM and the optimal sequence method for r = 4, 5 are given in Figure 7.7 below.

Attractor	map	a	b	c	d	e	f	probability p
The Tree	w_a	0.343	0.376	-0.203	0.546	-0.022	0.330	0.262
(Fig 7.7)	w_{b_1}	0.459	-0.226	0.073	0.602	-0.002	0.319	0.292
	w_{b_2}	0.136	0.503	-0.313	0.138	-0.020	0.217	0.176
	w_{b_3}	0.253	-0.490	0.308	0.350	-0.007	0.198	0.239
	w_{b_4}	0.066	0	0	0.479	-0.015	-0.024	0.031

Table 7.3 The code and probabilities p for the *tree* [21].

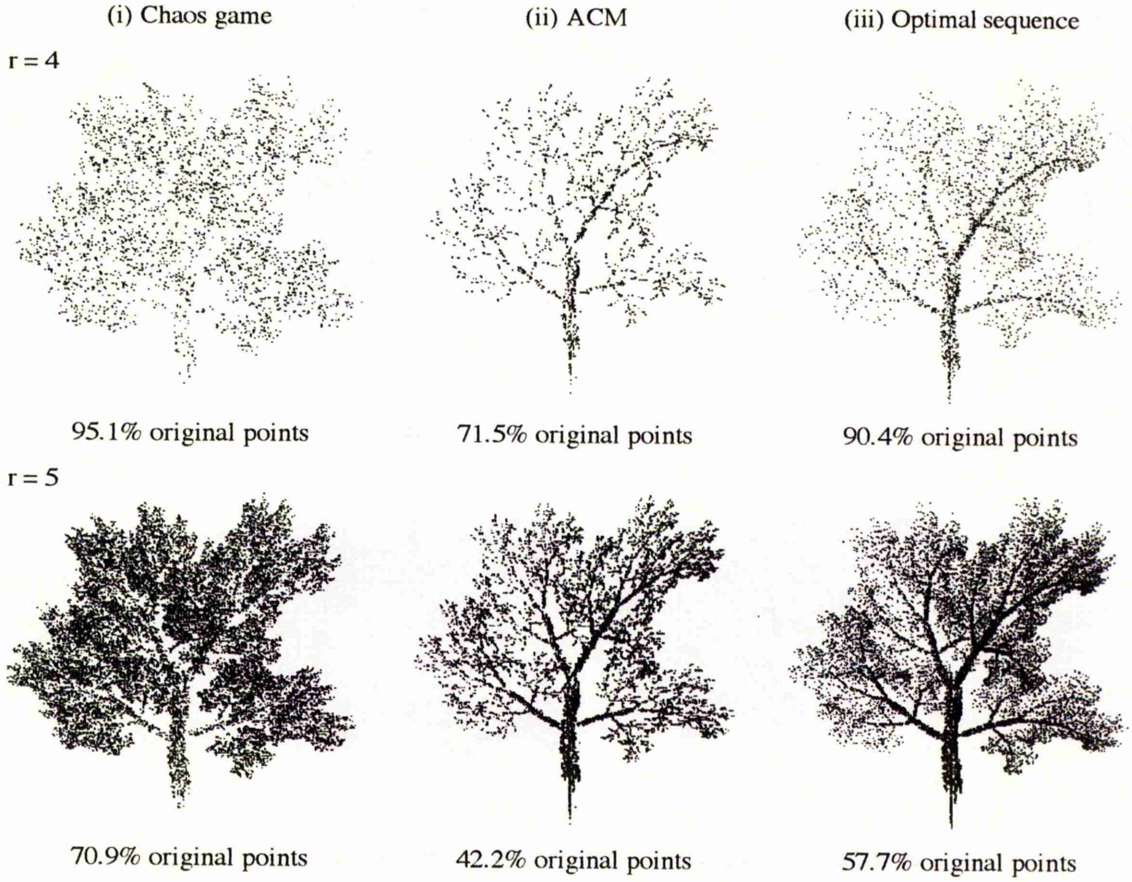


Figure 7.7 *Tree* produced by (i) the Chaos game, (ii) ACM and (iii) Optimal sequence.

Comparing the percentage of original points plotted by each method, one would expect the chaos game image to be superior. However, this is not the case. Although both the optimal sequence and the ACM produce fewer original points than the chaos game, the positioning of these points means the structure of the tree is clearer. In particular, the branches are well defined in the optimal sequence and ACM images whereas the chaos game image is very poor. The optimal sequence produces a higher percentage of original points and is faster than the ACM. Further, due to the free ride effect, the optimal sequence image shows more detail than the ACM. For example, the sub-branches are clearer in the optimal sequence image. Notice, although for r = 5, the ACM image may be more aesthetically pleasing, the optimal sequence image gives a truer approximation of the final attractor.

Note Since the stem is formed of transformed copies of the whole tree, the spike at the bottom is unavoidable.

7.2 The chaos game - Determining the probabilities

Recall, the chaos game requires probabilities p_i which determine how often each map is applied. We seek optimal values for these probabilities so that the attractor is produced as quickly as possible, with a minimal number of repetitions. However, there is no method at present which guarantees these values. In all previous examples, we set $p_i = |\det A_i|/\Sigma\ |\det A_i|$ where $w_i = A_i\mathbf{x} + \mathbf{t}_i$, or $p_i = \delta$ for some small positive number if $\det A_i = 0$.

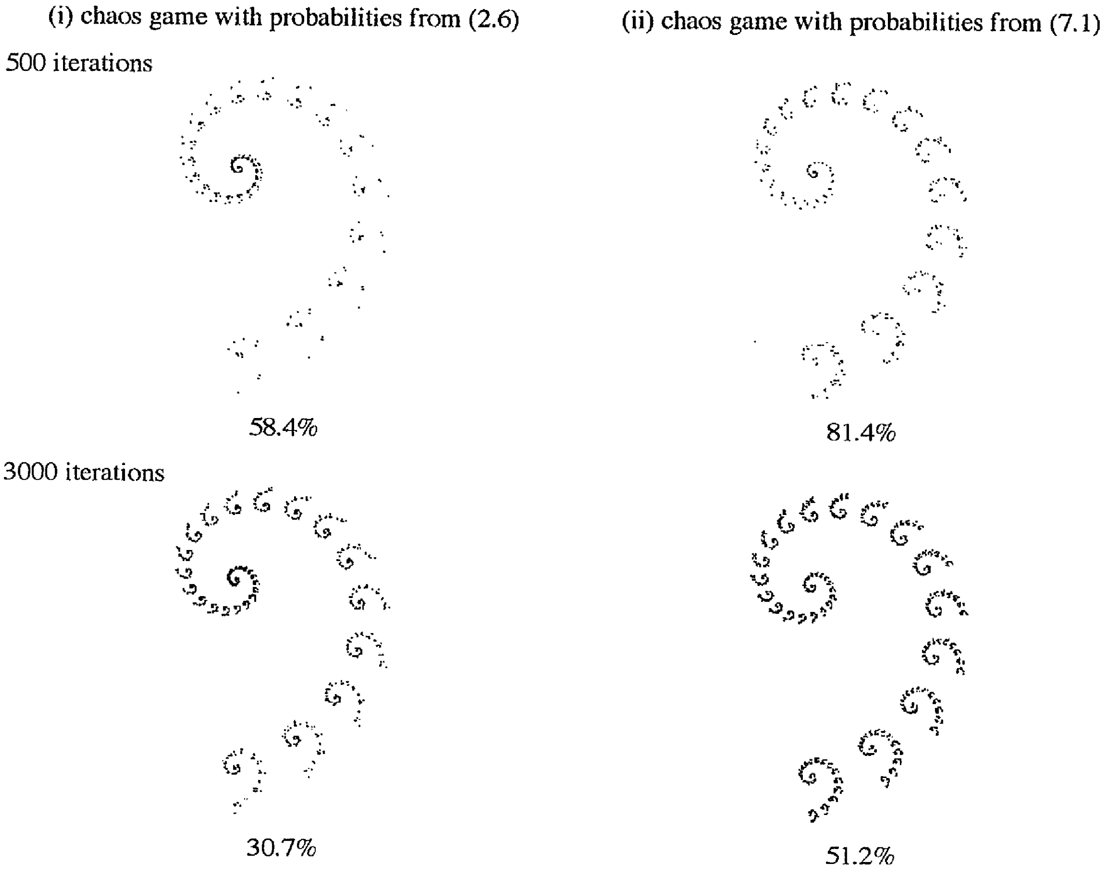
Addresses may provide an alternative way to determine these probabilities [28]. For a given $\varepsilon > 0$, both the ACM and the optimal sequence method divide the attractor into subsets of diameter not exceeding ε and plot one point in each. These subsets are $w_{\sigma_1}\dots w_{\sigma_k}(\mathcal{A})$ for each address $\sigma_1\dots\sigma_k$ but the actual point plotted in each subset is not usually the same for both methods. Recall $w_{\sigma_1}\dots w_{\sigma_k}(\mathcal{A})$ is contained within the subset $w_{\sigma_1}(\mathcal{A})$. Then, each address $\sigma_1\dots\sigma_k$ contributes one point to the subset $w_{\sigma_1}(\mathcal{A})$ and consequently we shall set the probabilities p_i as

$$p_i = \frac{\text{number of addresses beginning i}}{\text{total number of addresses}} \tag{7.1}$$

Although, the optimal sequence method and the ACM will both result in the same set of probabilities p_i (since they are based on the same address tree), the work involved differs significantly. To compute (7.1) using the ACM, we must calculate the code of each node of the address tree and keep a count of the number of addresses beginning with each symbol. In contrast, an optimal sequence S may be found quickly using the algorithms of Chapters 5 and 6. Then, to determine (7.1), since each symbol of S is the start of a unique address, we simply count the number of times each symbol i is present and divide by the length of S.

In the examples below, we compare the chaos game using the probabilities determined by formulae (2.6) and (7.1).

Example 7.8 *The Ear.* Using (2.6), the probabilities p_a and p_b are found to be 0.953 and 0.047 respectively. Recall $r_a^{16} \leq r_b$ and with $r = 4$, Algorithm 5.52 will give an associated optimal sequence. Subsequently, the probabilities p_a, p_b using (7.1) are 0.879 and 0.121 respectively. The corresponding images produced by the chaos game after 500 and 3000 iterations for each case are given in Figure 7.8. Notice with the (7.1) probabilities, the points do not tend to cluster together as much and so a good approximation will be achieved after fewer iterations. For example, after 500 iterations, the image corresponding to (7.1) has smaller outlined copies of itself. On the other hand, although probabilities (2.6) result in an image which is clearer at the centre, this is at the expense of the remainder of the image. After 3000 iterations, the image for (7.1) has one more level of detail than that for (2.6). The percentage original points in each case is given below the image.



(i) chaos game with probabilities from (2.6)

(ii) chaos game with probabilities from (7.1)

500 iterations

58.4%

81.4%

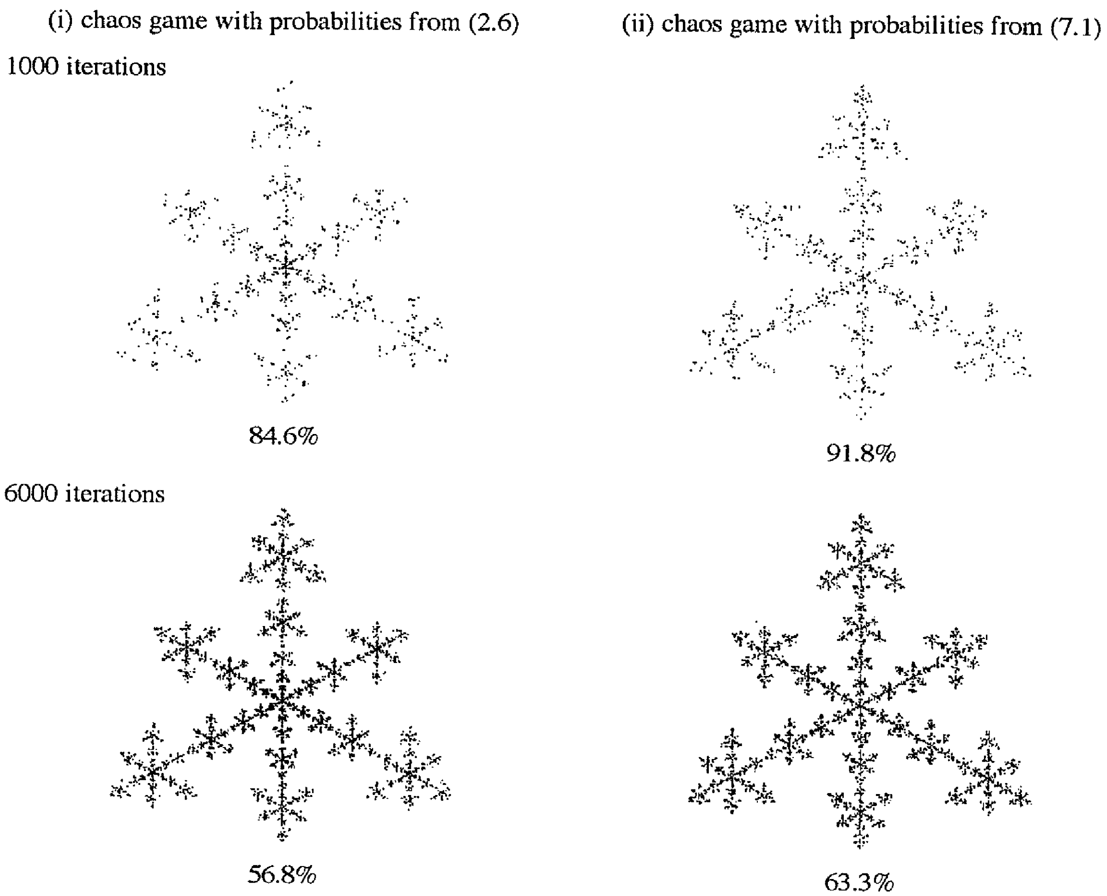
3000 iterations

30.7%

51.2%

Figure 7.8 The *Ear* produced by 500 and 3000 iterations of the chaos game with the probabilities given by (i) (2.6) and (ii) (7.1).

Example 7.9 *The Snowflake.* Using (2.6) $p_a = 0.739$ and $p_{b_1} = p_{b_2} = p_{b_3} = 0.087$ while (7.1) gives $p_a = 0.664$ and $p_{b_1} = p_{b_2} = p_{b_3} = 0.112$. For both cases, the results of the chaos game after 1000 and 6000 iterations are given in Figure 7.9. Again (7.1) yields a more favourable set of probabilities. With probabilities (2.6), the plotted points are concentrated around the centre and this suggests that these values are unbalanced.



(i) chaos game with probabilities from (2.6)

(ii) chaos game with probabilities from (7.1)

1000 iterations

84.6%

91.8%

6000 iterations

56.8%

63.3%

Figure 7.9 The *Snowflake* produced by 1000 and 6000 iterations of the chaos game with the probabilities given by (i) (2.6) and (ii) (7.1).

The two examples above illustrate that (7.1) produces a fairly efficient set of probabilities p_i. Further, unlike formula (2.6), it will never give $p_i = 0$.

Recall, the ratio of an affine transformation depends on the metric which is used. Throughout, we worked over the Euclidean metric. However, over the maximum metric (see below Definition 2.13), the ratio of the map w_σ with code (a, b, c, d, e, f) is $\rho_\infty(\sigma) = \max\{|a|+|b|, |c|+|d|\}$. Then, with respect to the maximum metric the map w_a for each of the Ear, the Spiral, the Snowflake and the Spiral 2 is not contractive. The ratios of the Eiffel Tower and the various uniform IFS of Chapter 3 are the same over the Euclidean and maximum metrics. Although, the actual ratios of each map of the fern and the tree varies slightly over the maximum metric (rather than the Euclidean metric), these ratios still fit model (6.1) with n = 6 and (6.2) with n = 2 respectively. Thus the optimal sequences used will be the same over both metrics. Finally over the maximum metric, the Pine tree satisfies model (6.2) with n = 4 (Recall we took n = 5 over the Euclidean metric.). It may be of interest to study this further and to consider the ratio over other metrics. However, due to lack of time, this will not be pursued at present.

Chapter 8 Discussion

In this thesis, we studied combinatorial ways of producing faster fractal images. We introduced the optimal sequence method.

We considered the uniform IFS $\{w_{1\text{-}N}\}$. We defined optimal sequences and studied ways of producing them. We illustrated that M-sequences defined over the finite field F_N are optimal sequences. We extended the use of the term M-sequence to mean any kth-order linear recurring sequence of period $N^k - 1$, over any structure with N elements. We introduced far rings and k-recurrent latin squares and illustrated the existence of M-sequences over far rings.

We then considered a simple model non-uniform IFS with just two maps, $\{w_a, w_b\}$. We studied the addresses and the formation of optimal sequences. We gave a template algorithm which we will be able to apply to all IFS $\{w_a, w_b\}$ as soon as we can determine *up*(k, r) and *down*(k, r) for all $k \geq 2$, $r \geq 2$. We adapted the algorithms to a more general non-uniform IFS using M-sequences.

For both uniform and non-uniform IFS, we proved that the optimal sequences are cyclic and that they remain optimal when reversed. We produced a wide variety of images including a Sierpinski gasket, an Eiffel tower, a fern and a tree.

On comparing the results of the chaos game and the optimal sequence method, we found the optimal sequence image gave a clearer impression of the attractor and had fewer repeated points. Moreover, the optimal sequence method was faster than the chaos game.

We also considered the adaptive cut method (ACM). The optimal sequence method and the ACM both partition the attractor in the same way and plot at least one point in each subset. Thus, for essentially the same number of iterations, the ACM and the optimal sequence method guarantee an image to the same degree of accuracy. However, we illustrated that in practice, the optimal sequence image generally is a truer approximation with more structure and detail than the corresponding ACM image. This was explained in Section 7.1.3. The optimal sequence image was significantly faster than the ACM. We discovered that the ACM image can depend on the choice of the point x_0 whereas neither the optimal sequence method nor the chaos game are affected by this.

We studied far rings and k-recurrent latin squares. We made various conjectures. In particular, for every positive integer k and every positive integer $N \neq p^r$, we hope to prove that there is an (N x N) k-recurrent latin square.

It is hoped that a general algorithm for generating optimal sequences corresponding to a wider class of attractors may be found. The possibilities of splicing sequences together (see Example 6.15) should also give more scope.

References

[1] Barnsley, M. F. and Demko, S., Iterated function systems and the global construction of fractals, Proc. of the Royal Soc of London, A 399, 1985, pp 243 - 275.

[2] Barnsley, M. F. and Hurd, L. P., Fractal image compression, A. K. Peters Ltd., Wellesley, Massachusetts, 1993.

[3] Barnsley, M. F. and Sloan, A. D., A better way to compress images, Byte Magazine, January 1988, pp 215 - 223.

[4] Barnsley, M. F., Fractals everywhere, Academic Press, 1988.

[5] Brammer, R.F., Unified image computing based on fractals and chaos model techniques, Optimal Engineering, Vol. 28, No. 7, July 1989, pp 726 - 734.

[6] Busacker, R.G. and Saaty, T.L., Finite graphs and networks - An introduction with applications, McGraw-Hill, 1965.

[7] Carrè, B. Graphs and Networks, Oxford University Press, 1979.

[8] Denes, J. and Keedwell, A.D., Latin squares and their applications, English Universities Press Limited, 1974.

[9] Dubac, S. and Elqortobi, A., Approximations of fractal sets, Journal of Computational and Applied Mathematics 29, 1990, pp 79-89.

[10] Fredricksen, H. and Maiorana, J., Necklaces of beads in k colours and k-ary de Bruijn sequences, Discrete Mathematics 23, 1978, pp 207 - 210.

[11] Goodman, G. S., A probabilist looks at the chaos game, FRACTAL 90 - Proc, of the 1st IFIP Conference on Fractals, Lisbon, June 6-8, 1990 (Peitgen, H.-O., Henriques, J. M., Penedo, L. F. eds.), Elsevier, Amsterdam, 1991, pp 159 - 168.

[12] Greenberger, M., CACM 8, 1965, pp 177 - 179.

[13] Hepting, D., Prusinkiewicz, P. and Saupe, D., Rendering methods for iterated function systems, FRACTAL 90 - Proc, of the 1st IFIP Conference on Fractals, Lisbon, June 6-8, 1990 (Peitgen, H.-O., Henriques, J. M., Penedo, L. F. eds.), Elsevier, Amsterdam, 1991, pp 183 - 224.

[14] Hoggar, S. G. and McFarlane, I., Faster fractal pictures by finite fields and far rings, 1993.

[15] Hoggar, S. G. and McFarlane, I., Optimal sequences for iterated function systems, 1993.

[16] Hoggar, S. G., Mathematical foundations of computer graphics, Cambridge University Press, 1992.

[17] Hutchinson J. E., Fractals and self similarity, Indiana University Mathematics Journal, Vol. 30, No. 5, 1981, pp 713 - 747.

[18] Jones, H., Durer, gaskets and Barnsley's chaos game, Computer Graphics Forum 9, 1990, pp 327 - 332.

[19] Knuth, D.E., The art of computer programming, Volume 1, Fundamental algorithms, Second Edition, Addison-Wesley, 1981.

[20] Knuth, D.E., The art of computer programming, Volume 2, Seminumerical algorithms, Second Edition, Addison-Wesley, 1981.

[21] Lee, K. D. and Cohen Y., Fractal attraction: A fractal design system for the Macintosh, Academic Press, 1992.

[22] Lehmer, D. H., Proc. 2nd Symp. on Large-scale Digital Calculating Machinery, Cambridge: Harvard University. Press, 1951 pp 141 - 146.

[23] Lidl, R. and Nierderreiter, H., Introduction to finite fields and their applications, Cambridge University Press, 1986.

[24] Mandelbrot, B. B., The fractal geometry of nature, Freeman, San Francisco, CA, 1982.

[25] McEliece, R. J., Finite fields for computer scientists and engineers, Kluwer, 1987.

[26] McFarlane, I. and Hoggar, S. G., Optimal drivers for the 'random' iteration algorithm, 1992.

[27] McFarlane, I. and S. G., Hoggar, Combinatorics for faster fractal pictures, 1993.

[28] Peitgen, H.-O., Jurgens, H. and Saupe, D., Fractals for the classroom, Springer-Verlag, 1988 (essentially incorporated in Chaos and fractals by the same authors and publishers).