# Chapter 8

# Fractal properties of plants

*Fractals vs. finite curves*

What is a fractal? In his 1982 book, Mandelbrot defines it as a set with Hausdorff-Besicovitch dimension $D_H$ strictly exceeding the topological dimension $D_T$ [95, page 15]. In this sense, none of the figures presented in this book are fractals, since they all consist of a finite number of primitives (lines or polygons), and $D_H = D_T$. However, the situation changes dramatically if the term "fractal" is used in a broader sense [95, page 39]:

> Strictly speaking, the triangle, the Star of David, and the finite Koch teragons are of dimension 1. However, both intuitively and from the pragmatic point of view of the simplicity and naturalness of the corrective terms required, it is reasonable to consider an advanced Koch teragon as being closer to a curve of dimension log 4/log 3 than to a curve of dimension 1.

Thus, a finite curve can be considered an approximate rendering of an infinite fractal as long as the interesting properties of both are closely related. In the case of plant models, this distinctive feature is self-similarity.

*Fractals vs. plants*

The use of approximate figures to illustrate abstract concepts has a long tradition in geometry. After all, even the primitives of Euclidean geometry — a point and a line — cannot be *drawn* exactly. An interesting question, however, concerns the relationship between fractals and real biological structures. The latter consist of a finite number of cells, thus are not fractals in the strict sense of the word. To consider real plants as approximations of "perfect" fractal structures would be acceptable only if we assumed Plato's view of the supremacy of ideas over their mundane realization. A viable approach is the opposite one, to consider fractals as abstract descriptions of the real structures. At first sight, this concept may seem strange. What can be gained by reducing an irregular contour of a compound leaf to an even more irregular fractal? Would it not be simpler to characterize the leaf using

*Complexity of fractals*

a smooth curve? The key to the answer lies in the meaning of the term "simple." A smooth curve may seem intuitively simpler than a fractal, but as a matter of fact, the reverse is often true [95, page 41]. According to Kolmogorov [80], the complexity of an object can be measured by the length of the shortest algorithm that generates it. In this sense, many fractals are particularly simple objects.

*Previous viewpoints*

The above discussion of the relationship between fractals and plants did not emerge in a vacuum. Mandelbrot [95] gives examples of the recursive branching structures of trees and flowers, analyzes their Hausdorff-Besicovitch dimension and writes inconclusively "trees may be called fractals in part." Smith [136] recognizes similarities between algorithms yielding Koch curves and branching plant-like structures, but does not qualify plant models as fractals. These structures are produced in a finite number of steps and consist of a finite number of line segments, while the "notion of fractal is defined only in the limit." Oppenheimer [105] uses the term "fractal" more freely, exchanging it with self-similarity, and comments: "The geometric notion of self-similarity became a paradigm for structure in the natural world. Nowhere is this principle more evident than in the world of botany." The approach presented in this chapter, which considers fractals as simplified abstract representations of real plant structures, seems to reconcile these previous opinions.

*Fractals in botany*

But why are we concerned with this problem at all? Does the notion of fractals provide any real assistance in the analysis and modeling of real botanical structures? On the conceptual level, the distinctive feature of the fractal approach to plant analysis is the emphasis on self-similarity. It offers a key to the understanding of complex-looking, compound structures, and suggests the recursive developmental mechanisms through which these structures could have been created. The reference to similarities in living structures plays a role analogous to the reference to symmetry in physics, where a strong link between conservation laws and the invariance under various symmetry operations can be observed. Weyl [159, page 145] advocates the search for symmetry as a cognitive tool:

> Whenever you have to deal with a structure-endowed entity $\Sigma$, try to determine its group of automorphisms, the group of those element-wise transformations which leave all structural relations undisturbed. You can expect to gain a deep insight into the constitution of $\Sigma$ in this way.

The relationship between symmetry and self-similarity is discussed in Section 8.1. Technically, the recognition of self-similar features of plant structures makes it possible to render them using algorithms developed for fractals as discussed in Section 8.2.

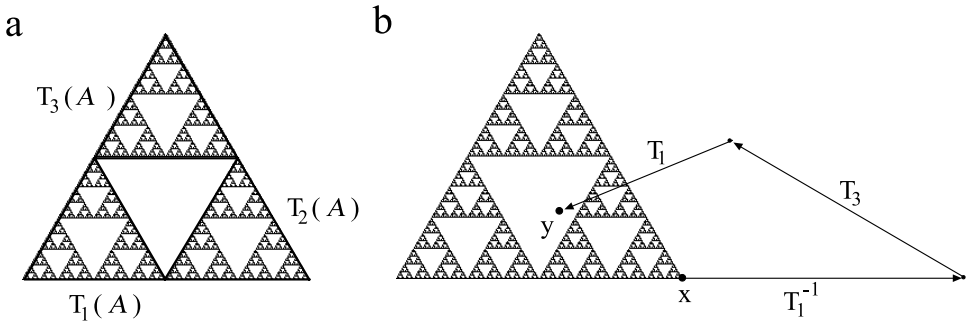



Figure 8.1: The Sierpiński gasket is closed with respect to transformations $T_1$, $T_2$ and $T_3$ (a), but it is not closed with respect to the set including the inverse transformations (b).

## 8.1 Symmetry and self-similarity

The notion of symmetry is generally defined as the invariance of a configuration of elements under a group of automorphic transformations. Commonly considered transformations are congruences, which can be obtained by composing rotations, reflections and translations. Could we extend this list of transformations to similarities, and consider self-similarity as a special case of symmetry involving scaling operations?

On the surface, this seems possible. For example, Weyl [159, page 68] suggests: "In dealing with potentially infinite patterns like band ornaments or with infinite groups, the operation under which a pattern is invariant is not of necessity a congruence but could be a similarity." The spiral shapes of the shells *Turritella duplicata* and *Nautilus* are given as examples. However, all similarities involved have the same fixed point. The situation changes dramatically when similarities with different fixed points are considered. For example, the Sierpiński gasket is mapped onto itself by a set of three contractions $T_1$, $T_2$ and $T_3$ (Figure 8.1a). Each contraction takes the entire figure into one of its three main components. Thus, if $A$ is an arbitrary point of the gasket, and $T = T_{i_1} T_{i_2} \dots T_{i_n}$ is an arbitrary composition of transformations $T_1$, $T_2$ and $T_3$, the image $T(A)$ will belong to the set $A$. On the other hand, if the inverses of transformations $T_1$, $T_2$ and $T_3$ can also be included in the composition, one obtains points that do not belong to the set $A$ nor its infinite extension (Figure 8.1b). This indicates that the set of transformations that maps $A$ into itself forms a semigroup generated by $T_1$, $T_2$ and $T_3$, but does not form a group. Thus, self-similarity is a weaker property than symmetry, yet it still provides a valuable insight into the relationships between the elements of a structure.

Figure 8.2: The fern leaf from Barnsley's model [7]

## 8.2 Plant models and iterated function systems

Barnsley [7, pages 101–104] presents a model of a fern leaf (Figure 8.2), generated using an *iterated function system*, or IFS. This raises a question regarding the relationship between developmental plant models expressed using L-systems and plant-like structures captured by IFSes. This section briefly describes IFSes and introduces a method for constructing those which approximate structures generated by a certain type of parametric L-system. The restrictions of this method are analyzed, shedding light on the role of IFSes in the modeling of biological structures.

*IFS definition*

By definition [74], a planar iterated function system is a finite set of contractive affine mappings $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ which map the plane $\mathcal{R} \times \mathcal{R}$ into itself. The *set defined by* $\mathcal{T}$ is the smallest nonempty set $\mathcal{A}$, closed in the topological sense, such that the image $y$ of any point $x \in \mathcal{A}$ under any of the mappings $T_i \in \mathcal{T}$ also belongs to $\mathcal{A}$. It can be shown that such a set always exists and is unique [74] (see also [118] for an elementary presentation of the proof). Thus, starting from an arbitrary point $x \in \mathcal{A}$, one can approximate $\mathcal{A}$ as a set of images of $x$ under compositions of the transformations from $\mathcal{T}$. On

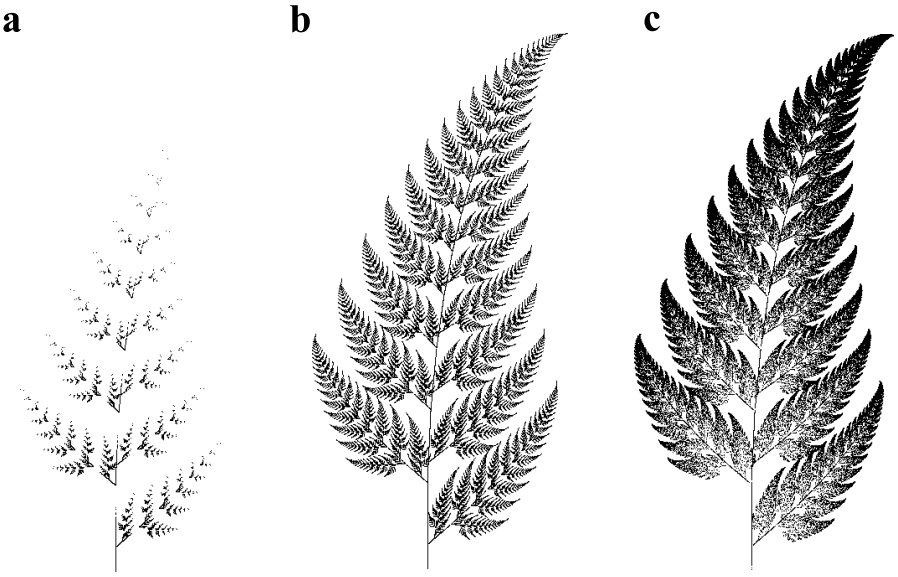



Figure 8.3: A comparison of three attracting methods for the rendering of a set defined by an IFS: (a) deterministic method using a balanced tree of depth $n = 9$ with the total number of points $N_1 = 349,525$, (b) deterministic method using a non-balanced tree with $N_2 = 198,541$ points, (c) stochastic method with $N_3 = N_2$ points

the other hand, if the starting point $x$ does not belong to $\mathcal{A}$, the consecutive images of $x$ gradually approach $\mathcal{A}$, since all mappings $T_i$ are contractions. For this reason, the set $\mathcal{A}$ is called the *attractor* of the IFS $\mathcal{T}$. The methods for rendering it are based on finding the images $T_{i_k}(T_{i_{k-1}}(\ldots(T_{i_1}(x))\ldots)) = xT_{i_1}\ldots T_{i_{k-1}}T_{i_k}$, and are termed *attracting methods*. According to the *deterministic approach* [123], a tree of transformations is constructed, with each node representing a point in $\mathcal{A}$. Various strategies, such as breadth-first or depth-first, can be devised to traverse this tree and produce different intermediate results [60]. If the transformations in $\mathcal{T}$ do not have the same scaling factors (Lipschitz constants), the use of a balanced tree yields a non-uniform distribution of points in $\mathcal{A}$. This effect can be eliminated by constructing a non-balanced tree, using a proper criterion for stopping the extension of a branch [60]. An alternative approach for approximating the set $\mathcal{A}$ is termed the *chaos game* [7] (see also [107, Chapter 5]). In this case, only one sequence of transformations is constructed, corresponding to a single path in the potentially infinite tree of transformations. The transformation applied in each derivation step is selected at random. In order to achieve a uniform distribution of points in the attractor, the probability of choosing transformation $T_i \in \mathcal{T}$ is set according to its Lipschitz constant. Figure 8.3 illustrates the difference between the stochastic and deterministic methods of rendering the attractor. The

*Rendering methods*

underlying IFS consists of four transformations, given below using homogeneous coordinates [40]:

$$T_1 = \begin{bmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.16 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0.20 & 0.23 & 0.00 \\ -0.26 & 0.22 & 0.00 \\ 0.00 & 1.60 & 1.00 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} -0.15 & 0.26 & 0.00 \\ 0.28 & 0.24 & 0.00 \\ 0.00 & 0.44 & 1.00 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 0.85 & -0.04 & 0.00 \\ 0.04 & 0.85 & 0.00 \\ 0.00 & 1.60 & 1.00 \end{bmatrix}$$

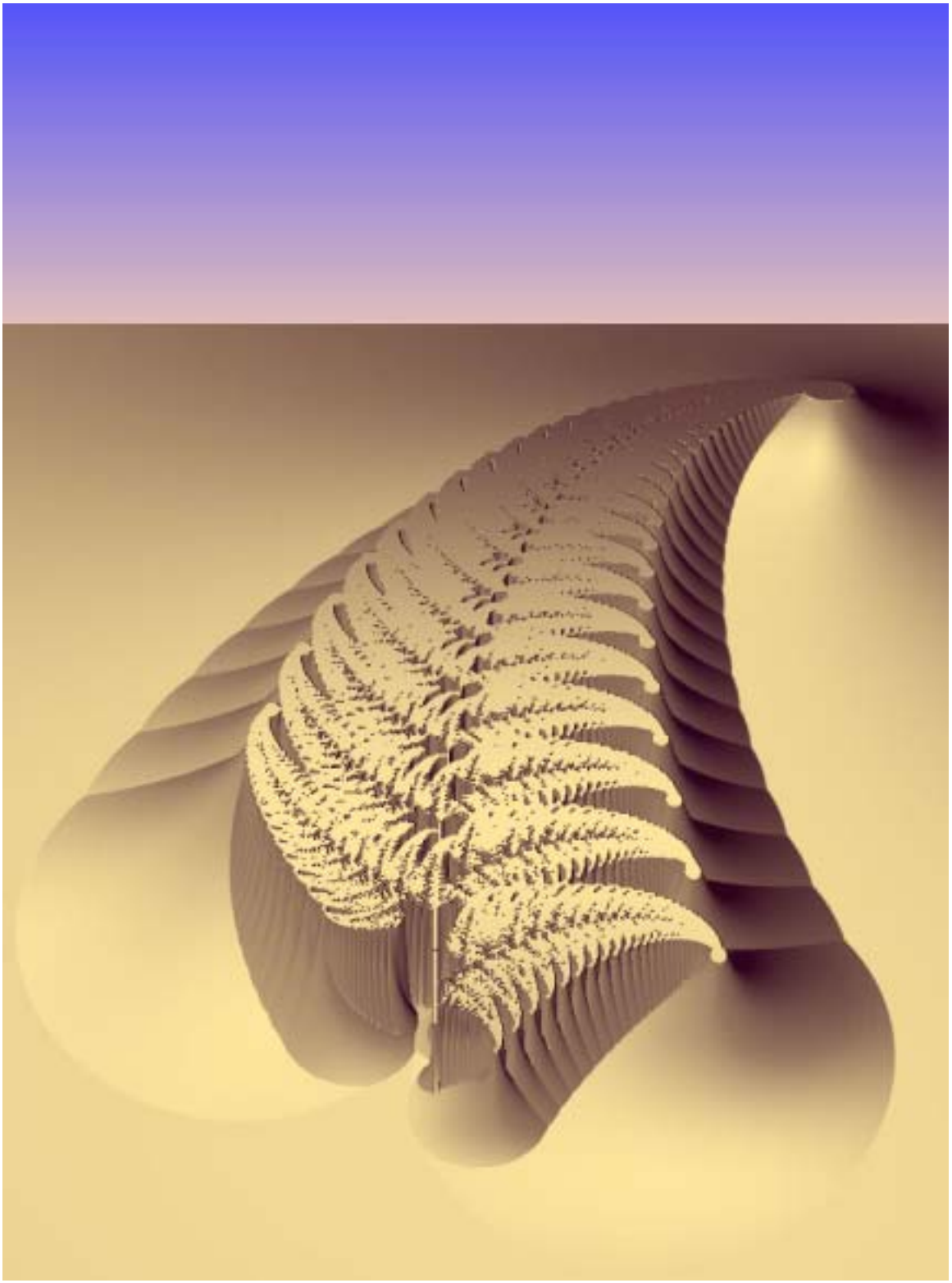
Other methods for the rendering of the set $\mathcal{A}$, defined by an interated function system $\mathcal{T}$, include the *repelling* or *escape-time* method and the *distance* method [60, 118]. Both methods assign values to points outside of $\mathcal{A}$. The first method determines how fast a point is repelled from $\mathcal{A}$ to infinity by the set of inverse transformations $T_i^{-1}$, where $T_i \in \mathcal{T}$. An example of the application of this method, with escape time values represented as a height field, is shown in Figure 8.4. The second method computes the Euclidean distance of a point from the attractor $\mathcal{A}$.

*IFS construction*

The problem of constructing an IFS that will approximate a branching structure modeled using an L-system can now be considered. This discussion focuses specifically on structures that develop in a biologically justifiable way, by subapical branching (Section 3.2). The compound leaf shown in Figure 5.11a on page 129 will be used as a working example. In this case, the apical delay $D$ is equal to zero, and the L-system can be represented in the simplified form:

$$\begin{array}{lllll} \omega: & A & & & \\ p_1: & A & : * & \rightarrow & F(1)[+A][-A]F(1)A \\ p_2: & F(a) & : * & \rightarrow & F(a * R) \end{array} \tag{8.1}$$

Figure 8.4: Fern dune →



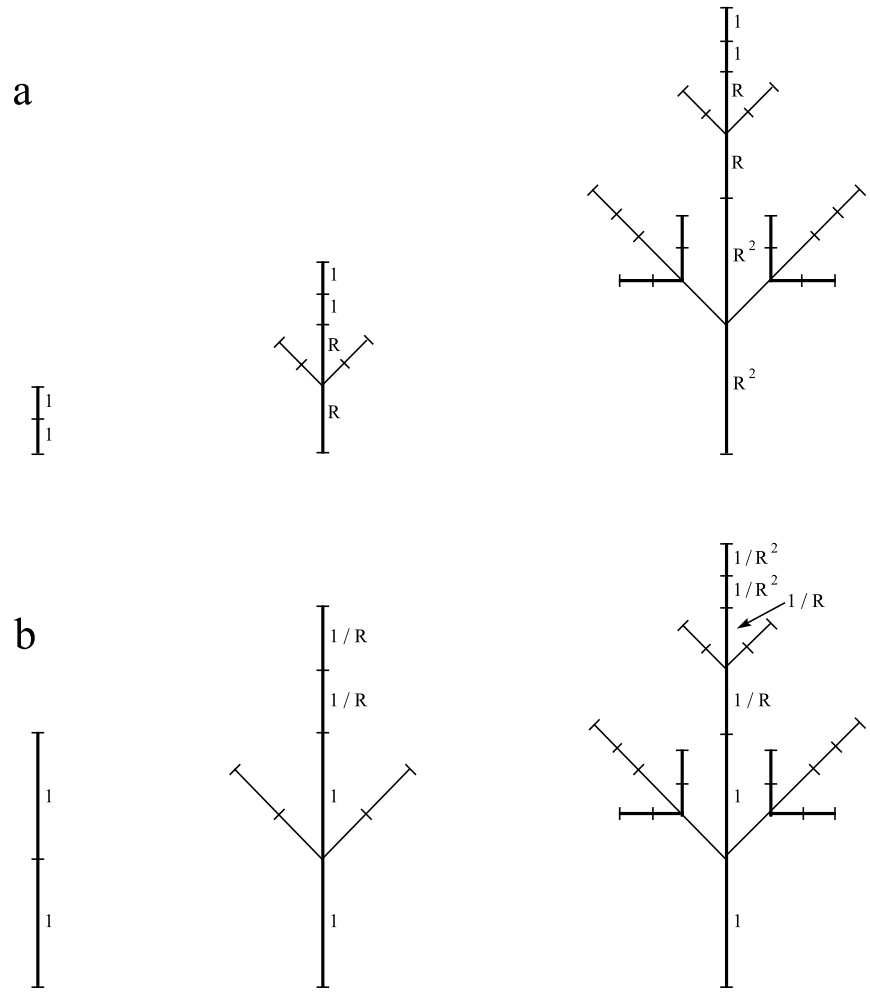



Figure 8.5: Initial sequences of structures generated by the L-systems specified in equations (8.1) and (8.2)

This L-system operates by creating segments of constant size, then increasing their length by constant factor $R$ in each derivation step (Figure 8.5a). As discussed in Section 1.10.3, a structure with the same proportions can be obtained by successively appending segments of decreasing length (Figure 8.5b):

$$\begin{array}{lll} \omega & : & A(1) \\ p_1 & : & A(s) \quad : \quad * \quad \rightarrow \quad F(s)[+A(s/R)][-A(s/R)]F(s)A(s/R) \end{array} \tag{8.2}$$

Let $A_n(s)$ denote the structure generated by module $A(s)$ in $n \geq 1$ derivation steps. According to production $p_1$, the following equality holds:

$$A_n(s) = F(s)[+A_{n-1}(s/R)][-A_{n-1}(s/R)]F(s)A_{n-1}(s/R) \tag{8.3}$$

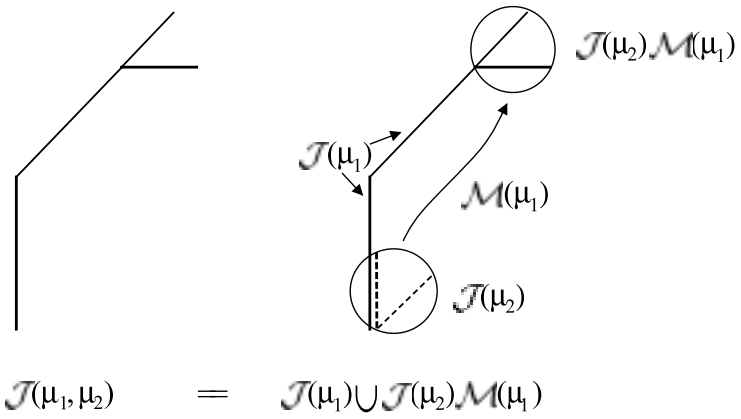



Figure 8.6: Illustration of equation (8.4), with $\mu_1 = F(1) - (45)$ and $\mu_2 = [-(45)F(0.5)]F(0.5)$

*Properties of turtle interpretation*

It is important to clearly distinguish between a parametric word $\mu$ (a string of modules) and its turtle interpretation $\mathcal{J}(\mu)$ (a set of points in the plane). The symbol $\mathcal{M}(\mu)$ will be used to denote the *transformation induced by* $\mu$. This transformation moves the turtle from its initial position and orientation to those resulting from the interpretation of word $\mu$. According to the definition of turtle interpretation (Chapter 1), if a word $\mu$ is decomposed into subwords $\mu_1$ and $\mu_2$ such that $\mu_2$ does not contain unbalanced right brackets, then

$$\mathcal{J}(\mu) = \mathcal{J}(\mu_1\mu_2) = \mathcal{J}(\mu_1) \cup \mathcal{J}(\mu_2)\mathcal{M}(\mu_1). \tag{8.4}$$

See Figure 8.6 for an illustration. By applying equation (8.4) to (8.3), we obtain

$$\begin{aligned} \mathcal{J}(A_n(s)) \;=\; & \mathcal{J}(F(s)) \cup \\ & \mathcal{J}(A_{n-1}(s/R))\mathcal{M}(F(s)+) \cup \\ & \mathcal{J}(A_{n-1}(s/R))\mathcal{M}(F(s)-) \cup \\ & \mathcal{J}(F(s))\mathcal{M}(F(s)) \cup \\ & \mathcal{J}(A_{n-1}(s/R))\mathcal{M}(F(s)F(s)), \end{aligned} \tag{8.5}$$

*Passage to infinity*

which is true for any $n \geq 1$. Now let $\mathcal{A}(s)$ be the limit of the sequence of sets $\mathcal{J}(A_n(s))$ from equation 8.6,

$$\mathcal{A}(s) = \lim_{n\to\infty} \mathcal{J}(A_n(s)).$$

At the limit we obtain:

$$\mathcal{A}(s) = \mathcal{J}(F(2s)) \cup \mathcal{A}(s/R)(T_1' \cup T_2' \cup T_3'), \tag{8.6}$$

where $T_1' = \mathcal{T}(F(s)+)$, $T_2' = \mathcal{T}(F(s)-)$, and $T_3' = \mathcal{T}(F(2s))$. Let $S(s/R)$ be the operation of scaling by $s/R$, then

$$\mathcal{A}(s/R) = \mathcal{A}(s)S(s/R).$$

By noting $T_i = T_i'S(s/R)$ for $i = 1, 2, 3$, equation (8.6) can be transformed to

$$\mathcal{A}(s) = \mathcal{J}(F(2s)) \cup \mathcal{A}(s)(T_1 \cup T_2 \cup T_3). \tag{8.7}$$

The solution of this equation with respect to $\mathcal{A}(s)$ is

$$\mathcal{A}(s) = \mathcal{J}(F(2s))(T_1 \cup T_2 \cup T_3)^*, \tag{8.8}$$

where $(T_1 \cup T_2 \cup T_3)^*$ stands for the iteration of the union of transformations $T_1$, $T_2$ and $T_3$. Equation (8.8) suggests the following method for constructing the set $(\mathcal{A}(s))$:

- create segment $\mathcal{J}(F(2s))$
- create images of $\mathcal{J}(F(2s))$ using transformations $T_1, T_2, T_3$ and their compositions

Equation (8.7) and the method of constructing the set $\mathcal{A}(s)$ based on equation (8.8) are closely related to the definition of iterated function systems stated at the beginning of this chapter. However, instead of starting from an arbitrary point $x \in \mathcal{A}(s)$, the iteration begins with the set $\mathcal{J}(F(2s))$. Although this is simply a straight line segment, a question arises as to how its generation can be incorporated into an IFS. Two approaches can be distinguished.

*Controlled IFS*

The first approach is related to the notions of hierarchical iterated function systems discussed by Reuter [123] and recurrent IFSes introduced recently by Barnsley [8]. The line segment $\mathcal{J}(F(2s))$ is generated using an IFS, for example consisting of two scaling transformations $Q_1$ and $Q_2$ which map it onto its upper and lower half (Figure 8.7a). Subsequently, transformations $T_1$, $T_2$ and $T_3$ are applied to create other points of the set $\mathcal{A}(s)$. The order of transformation application is important. Transformations $Q_1$ and $Q_2$ are used solely for the purpose of initial segment creation. They must not be applied after $T_1$, $T_2$ or $T_3$, since in this case they would affect the branching structure under consideration. The admissible sequences of transformations can be defined using a directed *control graph* (Figure 8.7b), and correspond to the infinite set of paths starting at node $a$.[1] The term *controlled iterated function system* (CIFS) denotes an IFS with restrictions on the transformation sequences imposed by a control graph. Thus, noting the angle increment associated with symbols + and − by $\delta$, the fractal

[1] Formally, the sequence of admissible transformations is the regular language accepted by the finite (Rabin-Scott) automaton represented by the graph in Figure 8.7b.

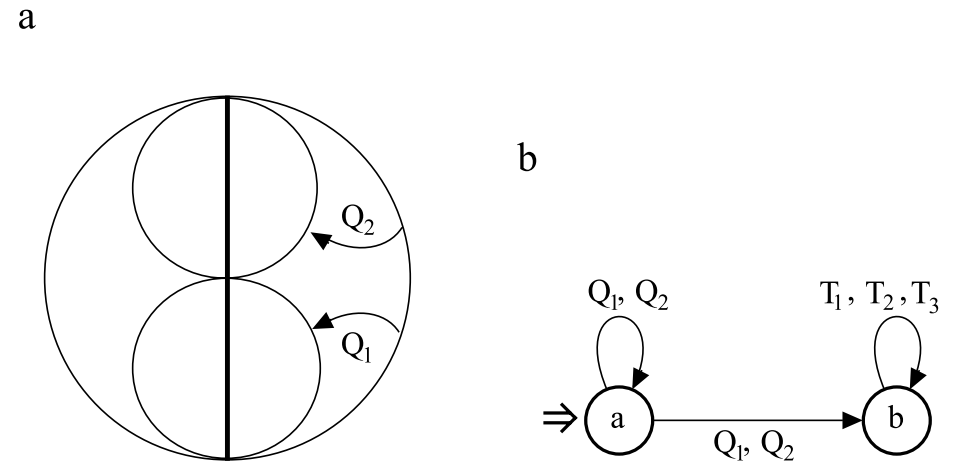



Figure 8.7: Construction of the set $\mathcal{A}(S)$: (a) definition of an IFS $\{Q_1, Q_2\}$ that generates the initial line segment, (b) the control graph specifying the admissible sequences of transformation application

*Resulting CIFS*

approximation of the leaf in Figure 5.11a is given by the CIFS with the control graph in Figure 8.7b and the transformations specified below:

$$Q_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & s & 1 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 1/R\cos\delta & 1/R\sin\delta & 0 \\ -1/R\sin\delta & 1/R\cos\delta & 0 \\ 0 & s & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1/R\cos\delta & -1/R\sin\delta & 0 \\ 1/R\sin\delta & 1/R\cos\delta & 0 \\ 0 & s & 1 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 1/R & 0 & 0 \\ 0 & 1/R & 0 \\ 0 & 2s & 1 \end{bmatrix}$$

*Noninvertible transformations*

The second approach to the generation of the line segment $\mathcal{J}(F(2s))$ is consistent with the method applied by Barnsley to specify the fern leaf in Figure 8.2. The idea is to map the entire branching structure $\mathcal{A}(s)$ onto the line $\mathcal{J}(F(2s))$. This can be achieved using a noninvertible transformation $Q$ which collapses all branches into a vertical line. The scaling factor along the $y$ axis is the ratio of the desired segment length $2s$, and the limit height of the entire structure $\mathcal{A}(s)$,

$$h = \frac{2s}{1 - 1/R}.$$

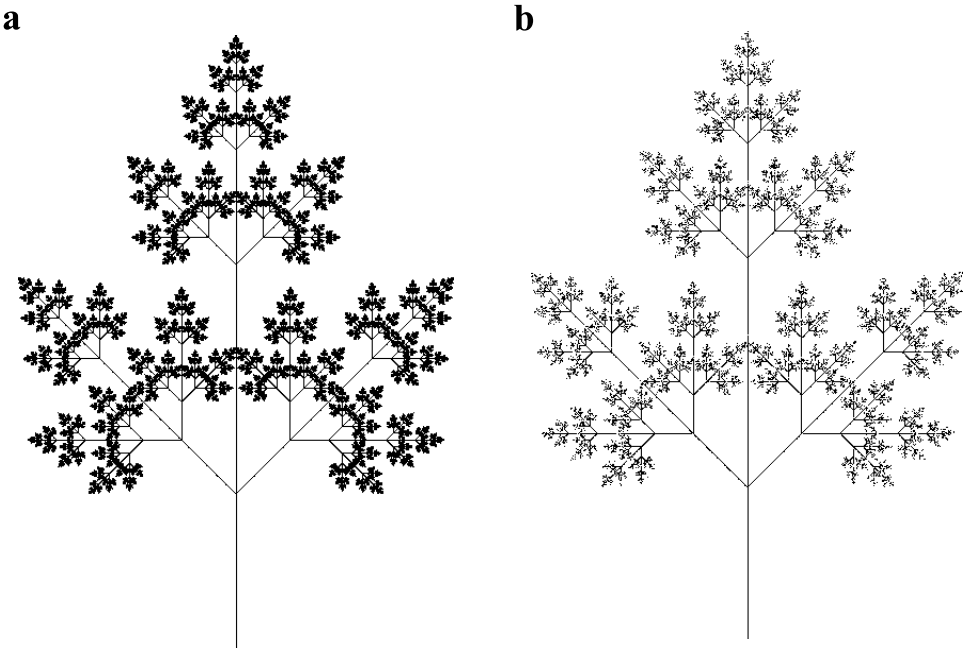



Figure 8.8: Two renderings of the compound leaf from Figure 5.11a, generated using iterated function systems

This last value is calculated as the limit of the geometric series with the first term equal to $2s$ and the ratio equal to $1/R$. Thus, the compound leaf of Figure 5.11a is defined by an IFS consisting of transformation

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1-1/R & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

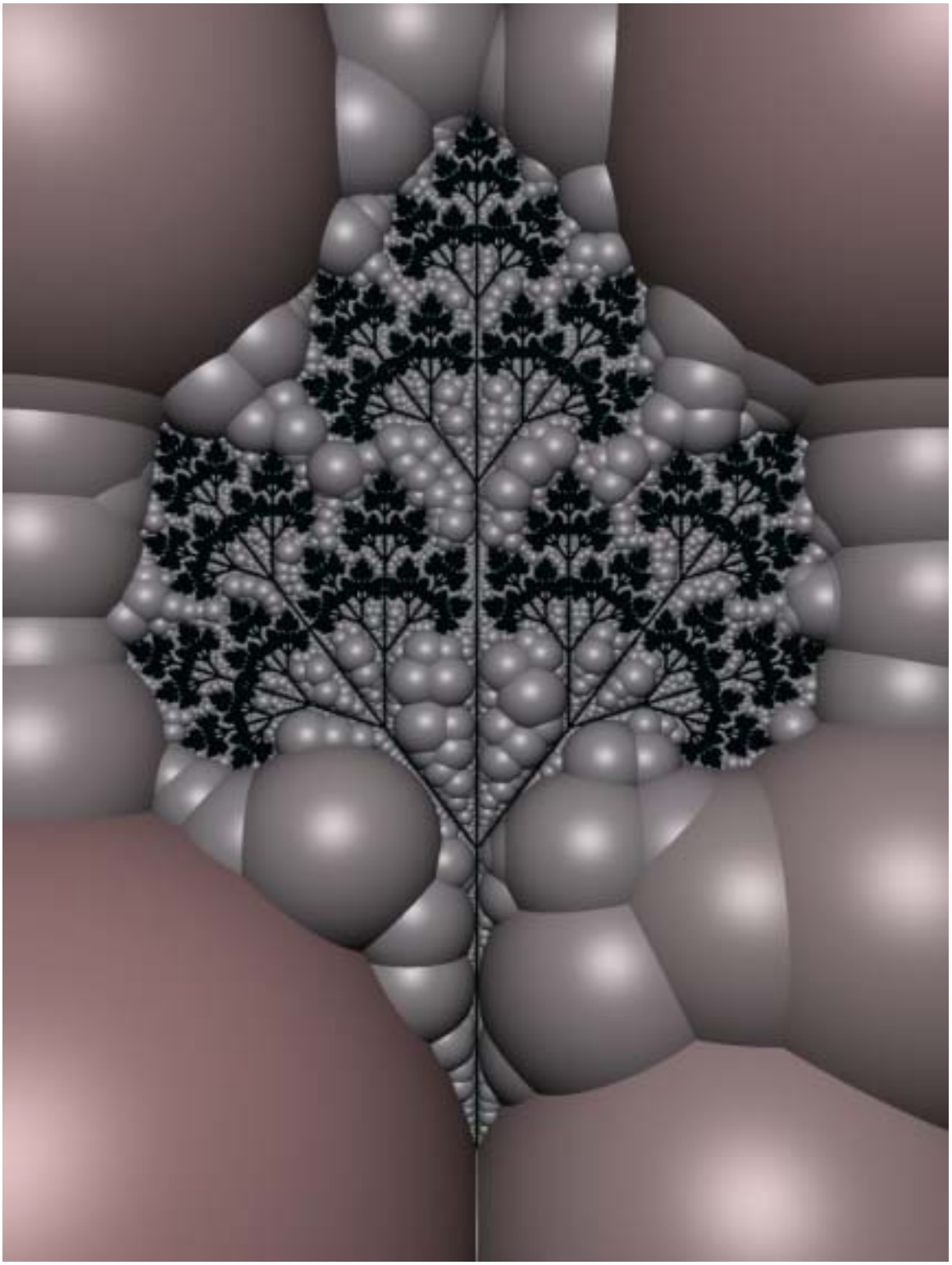
and transformations $T_1$, $T_2$ and $T_3$ specified as in the case of the controlled IFS.

*Rendering examples*

Two fractal-based renderings of the set $\mathcal{A}(s)$ are shown in Figure 8.8. Figure 8.8a was obtained using the controlled IFS and a deterministic algorithm to traverse the tree of admissible transformations. Figure 8.8b was obtained using the "ordinary" IFS and the random selection of transformations. Figure 8.9 shows another fractal-based rendering of the same structure. The spheres have radii equal to the distance from the sphere center to the leaf, within a specified $\epsilon$.

---

Figure 8.9: Carrot leaf 



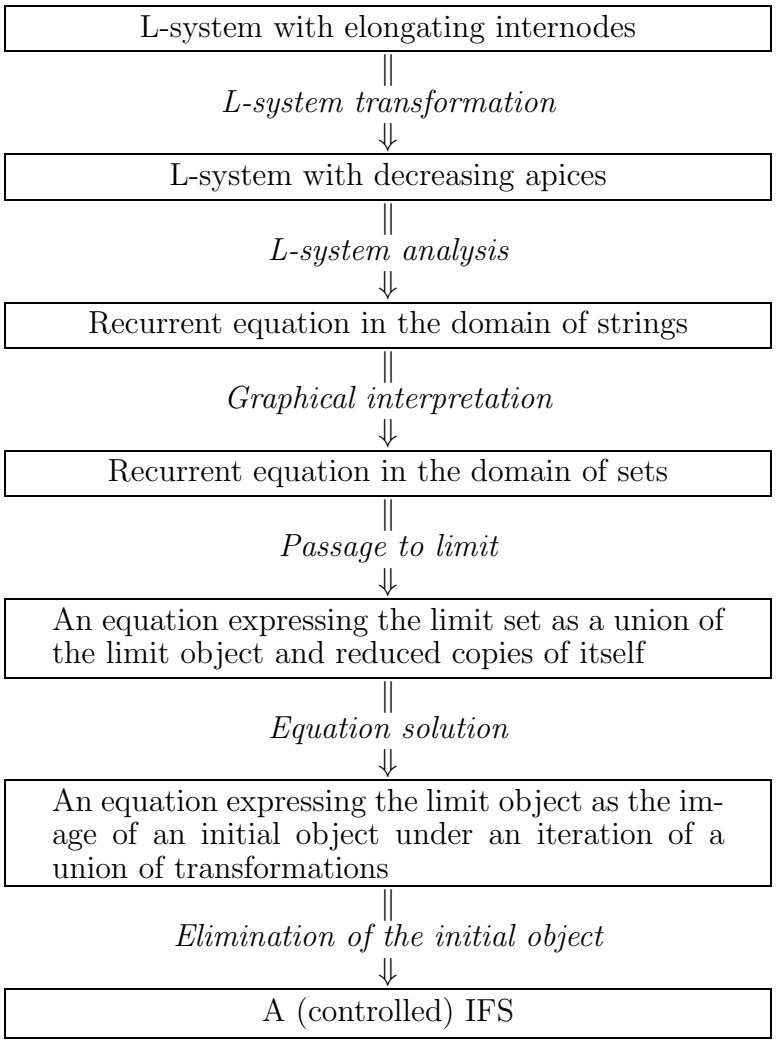



Figure 8.10: Steps in the construction of an IFS given an L-system capturing a developmental model

*Conclusions*

It is instructive to retrace the logical construction that started with an L-system, and ended with an iterated function system which can generate fractal approximations of the same object (Figure 8.10). An analysis of the operations performed in the subsequent steps of this construction reveals its limitations, and clarifies the relationship between strictly self-similar structures and real plants. The critical step is the transformation of the L-system with elongating internodes to the L-system with decreasing apices. It can be performed as indicated in the example if the plant maintains constant branching angles as well as fixed proportions between the mother and daughter segments, independent of branch order. This, in turn, can be achieved if all segments in the modeled plant elongate exponentially over time. These are strong assumptions, and may be satisfied to different degrees in real plants. Strict self-similarity is an abstraction that captures the essential properties of many plant structures and represents a useful point of reference when describing them in detail.

# Epilogue

This quiet place, reminiscient of Claude Monet's 1899 painting *Water-lilies pool — Harmony in green*, does not really exist. The scene was modeled using L-systems that captured the development of trees and water plants, and illuminated by simulated sunlight. It is difficult not to appreciate how far the theory of L-systems and the entire field of computer graphics have developed since their beginnings in the 1960's, making such images possible. Yet the results contained in this book are not conclusive and constitute only an introduction to the research on plant modeling for biological and graphics purposes. The algorithmic beauty of plants is open to further exploration.

---



Figure E.1: Water-lilies