Chapter 5

Models of plant organs

Many concepts presented in the previous chapters were illustrated using realistic images, but the modeling techniques for leaves and petals have not been described yet. Three approaches are discussed below.

5.1 Predefined surfaces

Bicubic patches

The standard computer graphics method for defining arbitrary surfaces makes use of *bicubic patches* [9, 10, 40]. A patch is defined by three polynomials of third degree with respect to parameters s and t. The following equation defines the x coordinate of a point on the patch:

$$\begin{aligned} x(s,t) = \; & a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 \\ & + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 \\ & + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s \\ & + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44} \end{aligned}$$

Analogous equations define $y(s,t)$ and $z(s,t)$. All coefficients are determined by interactively designing the desired shape on the screen of a graphics workstation. Complex surfaces are composed of several patches.

Turtle interpretation

The surfaces are incorporated into a plant model in a manner similar to subfigures (Section 1.4.2). The L-system alphabet is extended to include symbols representing different surfaces. When the turtle encounters such a symbol preceded by a tilde ($\sim$), the corresponding surface is drawn.

The exact position and orientation of surface S representing an appendage is determined using a *contact point* P_S, the *heading vector* $\vec{H}_S$ and the *up vector* $\vec{V}_S$ as a reference (Figure 5.1). The surface is translated in such a way that its contact point matches the current position

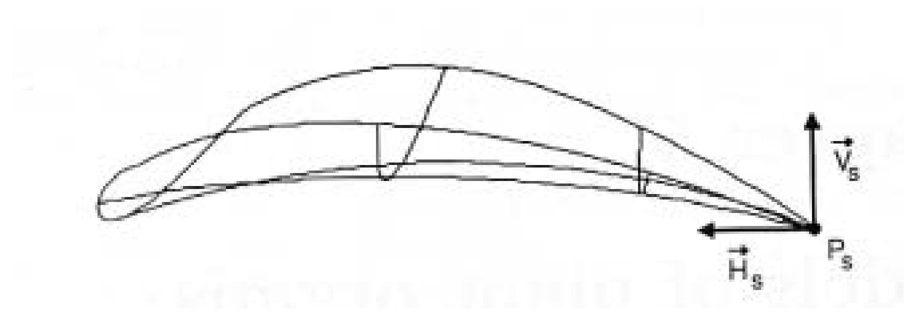


Figure 5.1: Specification of an appendage

of the turtle, and is rotated to align its heading and up vectors with the corresponding vectors of the turtle. If a surface represents an internal part of the structure such as an internode, a distinction between the entry and exit contact points is made.

Examples

The majority of organs presented in this book have been modeled this way. The petals of sunflowers, zinnias, water-lilies and roses shown in Chapter 4 provide good examples. Figure 5.2 illustrates an additional improvement in the appearance of organs, made possible by the application of textures to the surfaces of leaves, flowers and vine branches.

5.2 Developmental surface models

Predefined surfaces do not "grow." String symbols can be applied to control such features as the overall color and size of a surface, but the underlying shape remains the same. In order to simulate plant development fully, it is necessary to provide a mechanism for changing the shape as well as the size of surfaces in time. One approach is to trace surface boundaries using the turtle and fill the resulting polygons. A sample L-system is given below:

Contour tracing

$$\begin{array}{rl} \omega: & L \\ p_1: & L \to \{-FX+X-FX-|-FX+X+FX\} \\ p_2: & X \to FX \end{array}$$

Fern

Production p_1 defines leaf L as a closed planar polygon. The braces { and } indicate that this polygon should be filled. Production p_2 increases the lengths of its edges linearly. The model of a fern shown in Figure 5.3 incorporates leaves generated using this method, with the angle increment equal to 20°. Note the phase effect due to the "growth" of polygons in time. A similar approach was taken to generate the leaves, flowers and fruits of *Capsella bursa-pastoris* (Figure 3.5 on page 74).

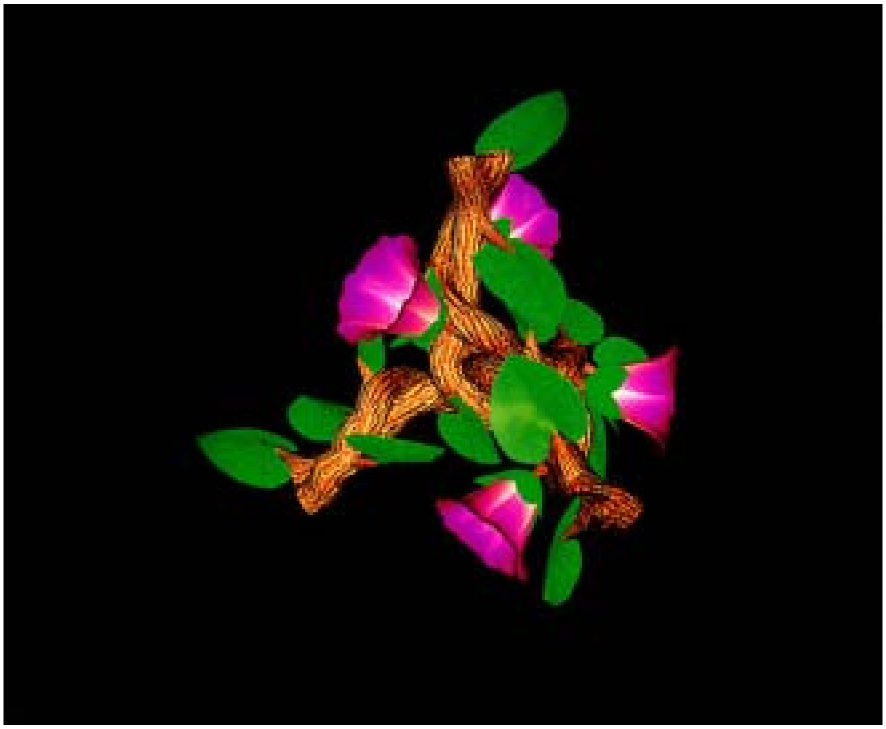


Figure 5.2: *Maraldi figure* by Greene [54]

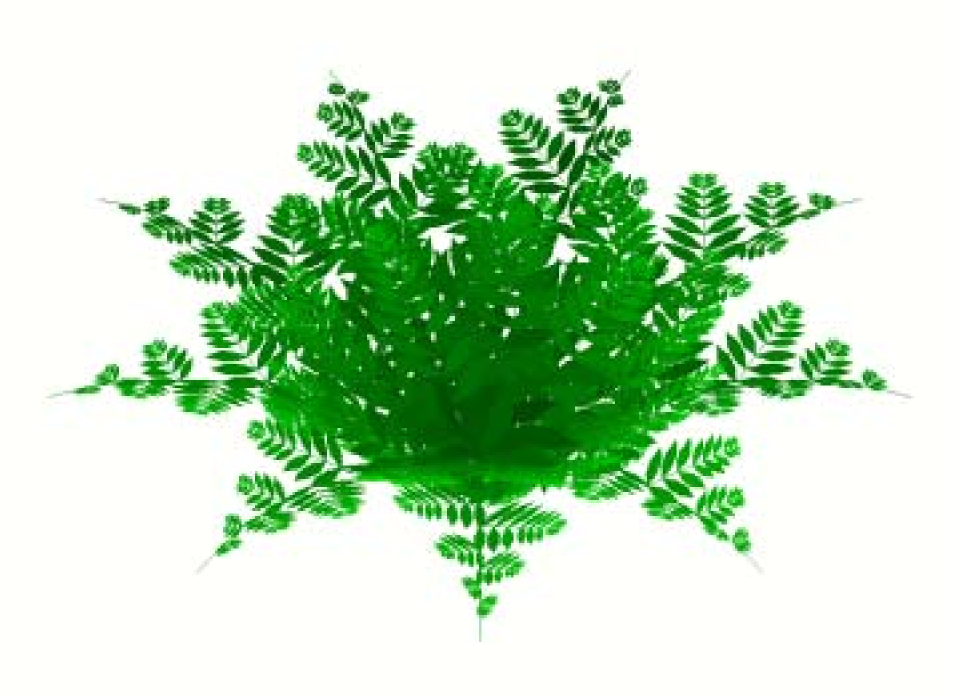
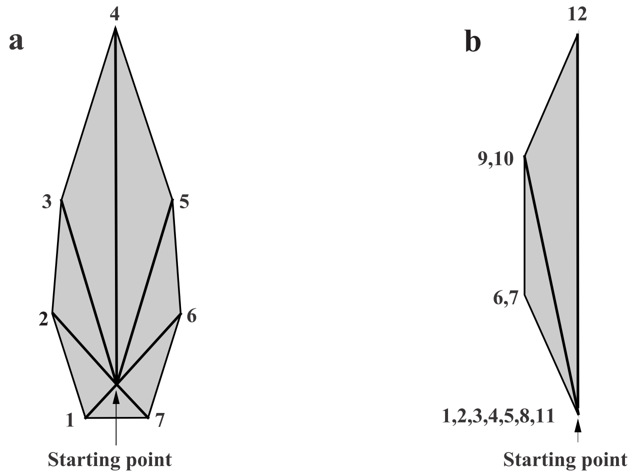


Figure 5.3: The fern



a {[++++G$\underset{1}{.}$][++GG$\underset{2}{.}$][+GGG$\underset{3}{.}$][GGGGG$\underset{4}{.}$][-GGG$\underset{5}{.}$][--GG$\underset{6}{.}$][----G$\underset{7}{.}$]}

b [+[+[+[+A{$\underset{1}{.}$]$\underset{2}{.}$C$\underset{3}{.}$}{$\underset{4}{.}$]$\underset{5}{.}$GC$\underset{6}{.}$}{$\underset{7}{.}$]$\underset{8}{.}$GGC$\underset{9}{.}$}{$\underset{10}{.}$]$\underset{11}{.}$GGGC$\underset{12}{.}$}

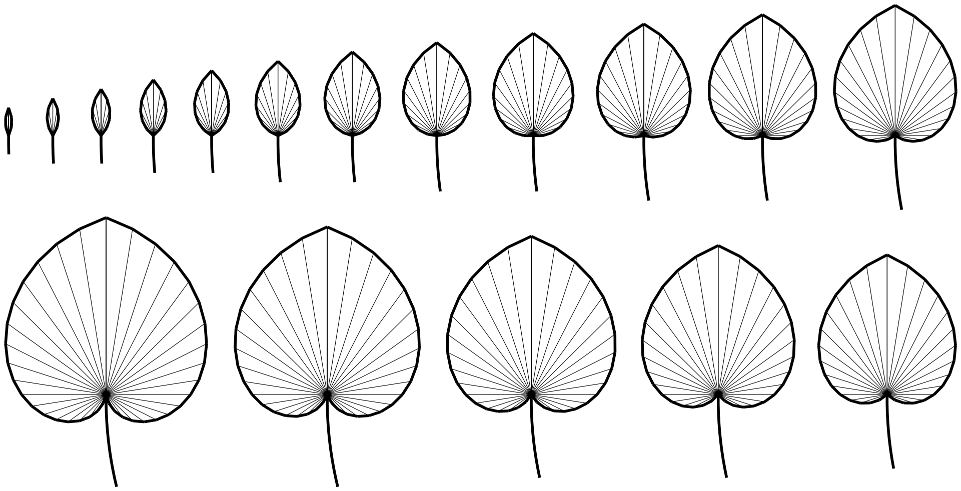
Figure 5.4: Surface specification using a tree structure as a framework

Framework approach

In practice, the tracing of polygon boundaries produces acceptable effects only in the case of small, flat surfaces. In other cases it is more convenient to use a tree structure as a *framework*. Polygon vertices are specified by a sequence of turtle positions marked by the dot symbol (.). An example is given in Figure 5.4a. The letter G has been used instead of F to indicate that the segments enclosed between the braces should not be interpreted as the edges of the constructed polygon. The numbers correspond to the order of vertex specification by the turtle.

Cordate leaf

Figure 5.5 shows the development of a *cordate leaf* modeled using this approach. The axiom contains symbols A and B, which initiate the left-hand and right-hand sides of the blade. Each of the productions p_1 and p_2 creates a sequence of axes starting at the leaf base and gradually diverging from the midrib. Production p_3 increases the lengths of the axes. The axes close to the midrib are the longest since they were created first. Thus, the shape of this leaf is yet another manifestation of the phase effect. The leaf blade is defined as a union of triangles rather than a single polygon. Such triangulation is advantageous if the blade bends, for example due to tropism (Chapter 2). Figure 5.4b provides an additional illustration of the model by magnifying the left side of the leaf after four derivation steps.



$$
\begin{array}{lcl}
\omega & : & \texttt{[A][B]} \\
p_1 & : & \texttt{A} \rightarrow \texttt{[+A\{.].C.\}} \\
p_2 & : & \texttt{B} \rightarrow \texttt{[-B\{.].C.\}} \\
p_3 & : & \texttt{C} \rightarrow \texttt{GC}
\end{array}
$$

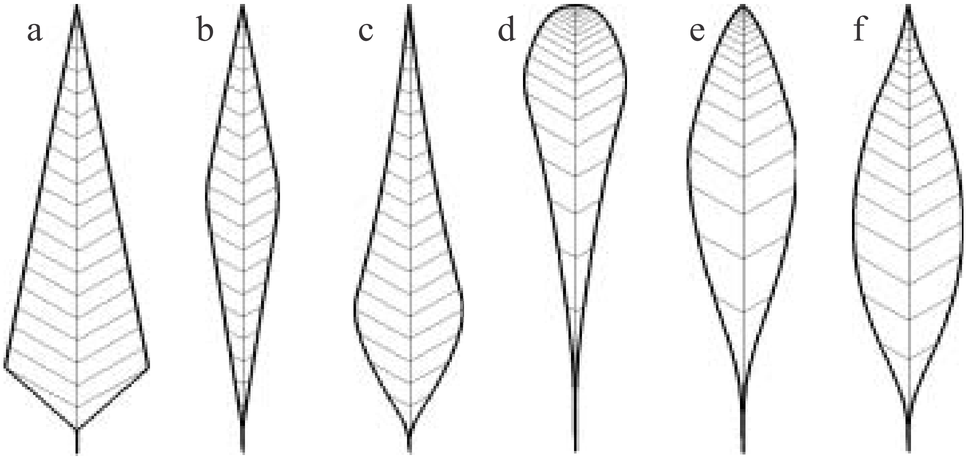
Figure 5.5: Developmental sequence of a cordate leaf generated using an L-system

Simple leaves

The described method makes it possible to define a variety of leaves. Their shapes depend strongly on the growth rates of segments. For example, a family of simple leaves and the corresponding parametric L-system are shown in Figure 5.6.

According to production p_1, in each derivation step apex $A(t)$ extends the main leaf axis by segment $G(LA, RA)$ and creates a pair of lateral apices $B(t)$. New lateral segments are added by production p_2. Parameter t, assigned to apices B by production p_1, plays the role of "growth potential" of the branches. It is decremented in each derivation step by a constant PD, and stops production of new lateral segments upon reaching 0. Segment elongation is captured by production p_3.

For the purpose of analysis, it is convenient to divide a leaf blade into two areas. In the basal area, the number of lateral segments is determined by the initial value of growth potential t and constant PD. Since the initial value of t assigned to apices B increases towards the leaf apex, the consecutive branches contain more and more segments. On the other hand, branches in the apical area exist for too short a time to reach their limit length. Thus, while traversing the leaf from the base towards the apex, the actual number of segments in a branch first increases, then decreases. As a result of these opposite tendencies, the leaf reaches its maximum width near the central part of the blade.



```
n=20, δ=60°

#define LA 5      /* initial length - main segment */
#define RA 1      /* growth rate - main segment */
#define LB 1      /* initial length - lateral segment */
#define RB 1      /* growth rate - lateral segment */
#define PD 1      /* growth potential decrement */

ω  :  {.A(0)}
p1 :  A(t)    :  *    → G(LA,RA)[-B(t).][A(t+1)][+B(t).]
p2 :  B(t)    :  t>0  → G(LB,RB)B(t-PD)
p3 :  G(s,r)  :  *    → G(s*r,r)
```

Figure 5.6: A family of simple leaves generated using a parametric L-system

Figure	LA	RA	LB	RB	PD
a	5	1.0	1.0	1.00	0.00
b	5	1.0	1.0	1.00	1.00
c	5	1.0	0.6	1.06	0.25
d	5	1.2	10.0	1.00	0.50
e	5	1.2	4.0	1.10	0.25
f	5	1.1	1.0	1.20	1.00

Table 5.1: Values of constants used to generate simple leaves

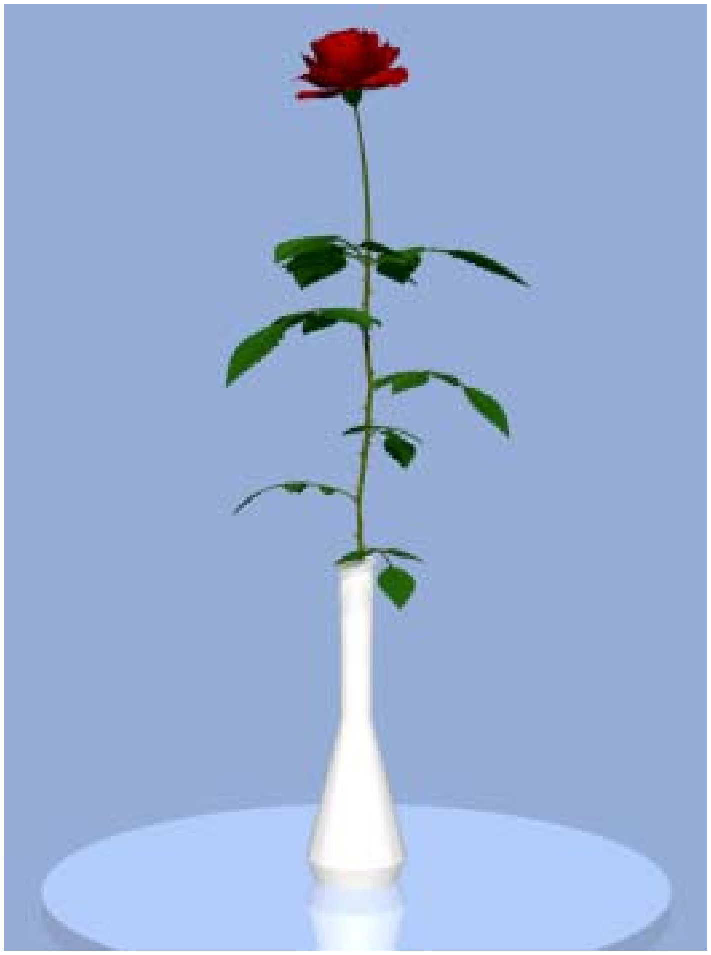
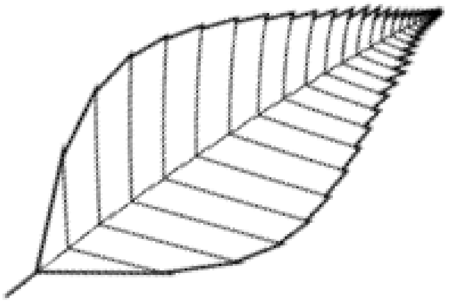


Figure 5.7: A rose in a vase

Shape control

Table 5.1 lists the values of constants corresponding to particular shapes. For a given derivation length, the exact position of the branch with the largest number of segments is determined by PD. If PD is equal to 0, all lateral branches have an unlimited growth potential, and the basal part of the leaf does not exist (Figure 5.6a). If PD equals 1, the basal and apical parts contain equal numbers of lateral branches (Figures 5.6 b and f). Finer details of the leaf shape are determined by the growth rates. If the main axis segments and the lateral segments have the same growth rates ($RA = RB$), the edges of the apical part of the leaf are straight (Figures 5.6 a and b). If RA is less than RB, the segments along the main axis elongate at a slower rate than the lateral segments, resulting in a concave shape of the apical part (Figures 5.6 c and f). In the opposite case, with RA greater than RB, the apical part is convex (Figures 5.6 d and e). The curvature of the basal edges can be analyzed in a similar way.



```
n=25, δ=60°

#define LA 5      /* initial length - main segment */
#define RA 1.15   /* growth rate - main segment */
#define LB 1.3    /* initial length - lateral segment */
#define RB 1.25   /* growth rate - lateral segment */
#define LC 3      /* initial length - marginal notch */
#define RC 1.19   /* growth rate - marginal notch */

ω  :  [{A(0,0).}][{A(0,1).}]
p1 :  A(t,d)   :  d=0  →  .G(LA,RA).[+B(t)G(LC,RC,t).}]
                                [+B(t){.]A(t+1,d)
p2 :  A(t,d)   :  d=1  →  .G(LA,RA).[-B(t)G(LC,RC,t).}]
                                [-B(t){.]A(t+1,d)
p3 :  B(t)     :  t>0  →  G(LB,RB)B(t-1)
p4 :  G(s,r)   :  *    →  G(s*r,r)
p5 :  G(s,r,t) :  t>1  →  G(s*r,r,t-1)
```

Figure 5.8: A rose leaf

Rose leaf

Figure 5.7 shows a long-stemmed rose with the leaves modeled according to Figure 5.8. The L-system combines the concepts explored in Figures 5.5 and 5.6. The axiom contains modules $A(0,0)$ and $A(0,1)$, which initiate the left-hand and right-hand side of the leaf. The development of the left side will be examined in detail. According to production p_1, in each derivation step apex $A(t,0)$ extends the midrib by internodes $G(LA,RA)$ and creates two colinear apices $B(t)$ pointing to the left. Further extension of the lateral axes is specified by production p_3. The leaf blade is constructed as a sequence of trapezoids, with two vertices lying on the midrib and the other two vertices placed at the endpoints of a pair of lateral axes formed in consecutive derivation steps. The module $G(LC,RC,t)$ introduces an offset responsible for the formation of notches at the leaf margin. Production p_4 describes the elongation of internodes responsible for overall leaf shape, while production p_5 controls the size of the notches. The development of the right side of the blade proceeds in a similar manner, with production p_2

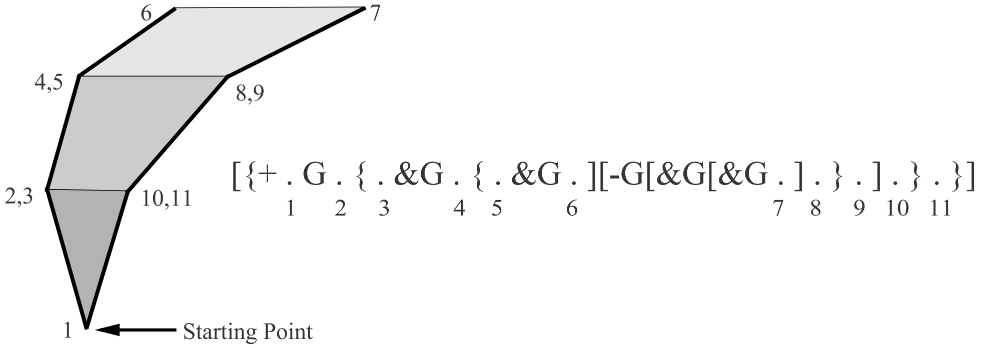


Figure 5.9: Surface specification using stacked polygons

recreating the midrib and creating lateral apices pointing to the right. The bending of the midrib to the right is a result of tropism.

Nested polygons

In the examples discussed so far, the turtle specifies the vertices of one polygon, then moves on to the next. Further flexibility in surface definition can be achieved by interleaving vertex specifications for different polygons. The turtle interpretation of the braces is redefined in the following way. A string containing nested braces is evaluated using two data structures, an *array* of vertices representing the current polygon and a polygon *stack*. At the beginning of string interpretation, both structures are empty. The symbols {, } and . are then interpreted as follows:

{ Start a new polygon by pushing the current polygon on the polygon stack and creating an empty current polygon.

. Append the new vertex to the current polygon.

} Draw the current polygon using the specified vertices, then pop a polygon from the stack and make it the current polygon.

An example of string interpretation involving nested braces is given in Figure 5.9.

Lily-of-the-valley

The above technique was applied to construct the flowers of the lily-of-the-valley shown in Figure 3.4 (page 72), and magnified in Figure 5.10. A flower is represented by a polygon mesh consisting of five sequences of trapezoids spread between pairs of curved lines that emanate radially from the flower base. A single sequence is generated by the following L-system:

$$\begin{array}{lllll}
\omega : & [X(36)A]/(72)[X(36)B] & & & \\
p_1 : & A \quad : & * & \rightarrow & [\&GA\{.]. \\
p_2 : & B \quad : & * & \rightarrow & B\&.G.\} \\
p_3 : & X(a) : & * & \rightarrow & X(a+4.5)
\end{array}$$

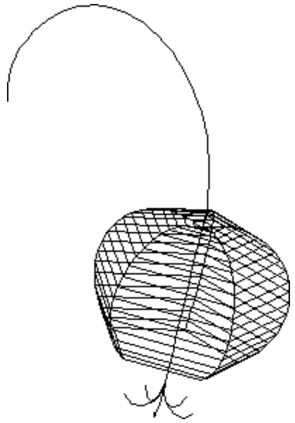


Figure 5.10: Structure of a lily-of-the-valley flower

Productions p_1 and p_2 create two adjacent framework lines and mark polygon vertices consistently with Figure 5.9. Production p_3 controls the angle at which the framework lines leave the stalk at the flower base.

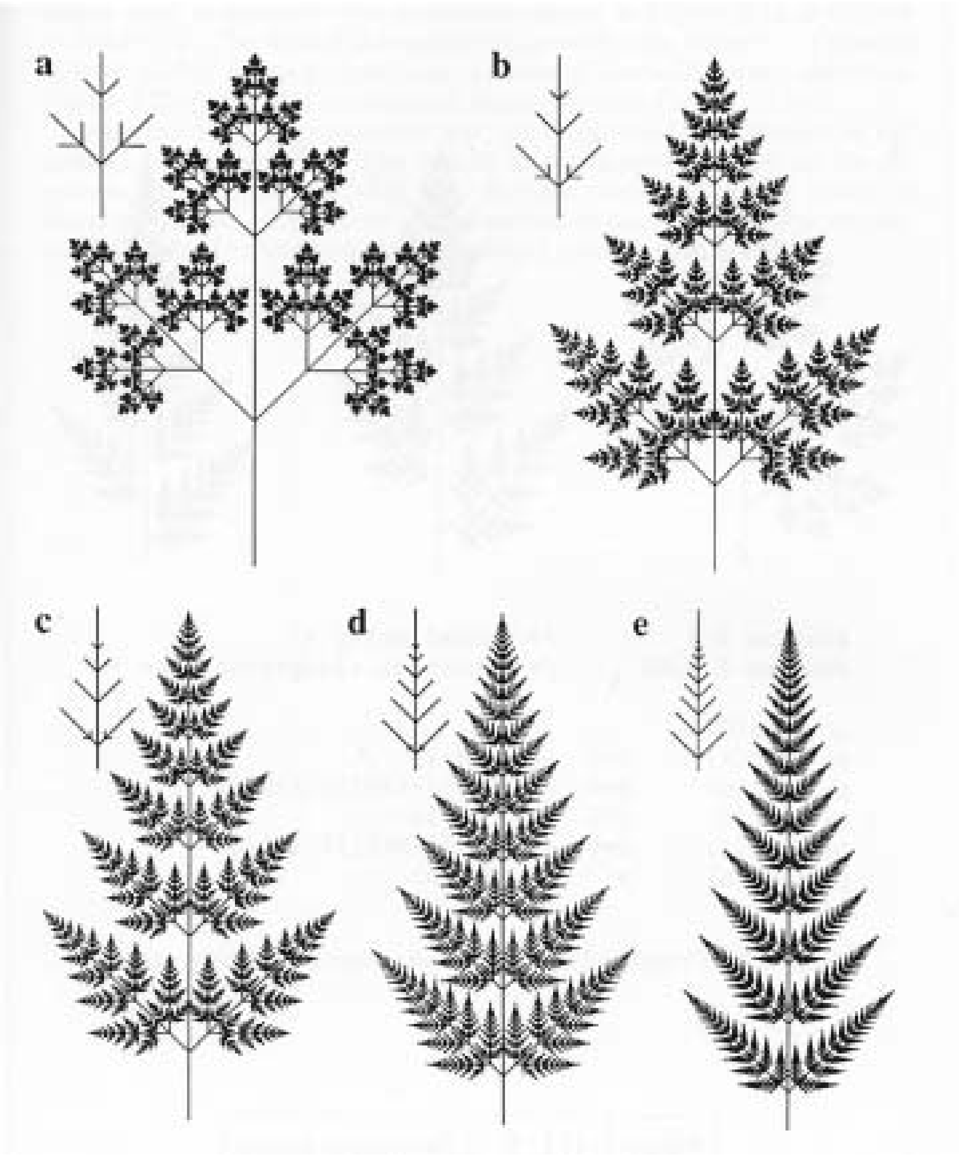
5.3 Models of compound leaves

Symmetric branching

So far, the discussion of organ models has focused on the definition of surfaces. However, in the case of highly self-similar structures, the individual surfaces become inconspicuous, and the expression of the geometric relationships between younger and older parts of the structure becomes the key issue. For example, Figure 5.11 shows compound leaves often found in the family Umbelliferae. According to production p_2, the apex $A(0)$ creates two segments $F(1)$ and a pair of lateral apices $A(D)$ in each derivation step. Production p_1 delays the development of the daughter branches by D steps with respect to the mother branch. This pattern is repeated recursively in branches of higher order. Production p_3 gradually elongates the internodes, and in this way establishes proportions between parts of a leaf. The values of the con-

Figure	D	R	Derivation length
a	0	2.00	10
b	1	1.50	16
c	2	1.36	21
d	4	1.23	30
e	7	1.17	45

Table 5.2: Values of constants used to generate compound leaves



```
#define D 1          /* apical delay */
#define R 1.5        /* internode elongation rate */
```

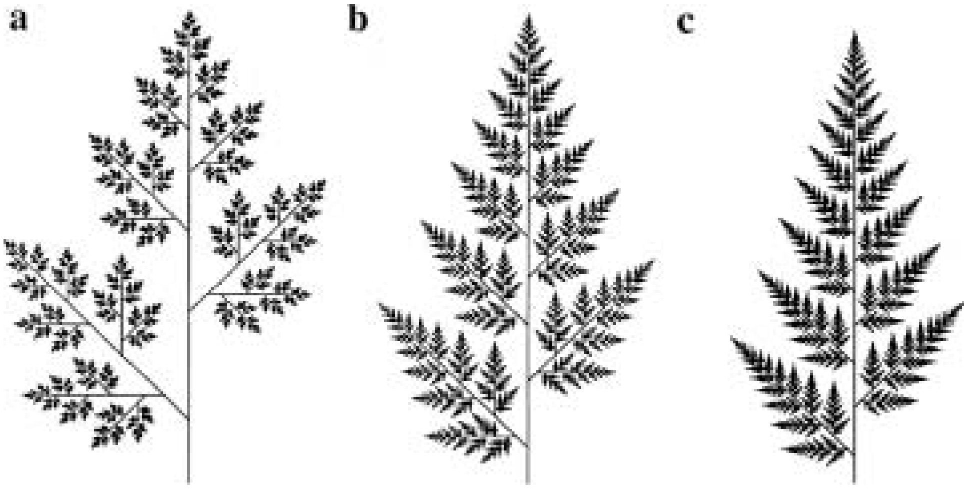
ω : `A(0)`

p_1 : `A(d) : d > 0` $\rightarrow$ `A(d-1)`

p_2 : `A(d) : d = 0` $\rightarrow$ `F(1)[+A(D)][-A(D)]F(1)A(0)`

p_3 : `F(a) : *` $\rightarrow$ `F(a*R)`

Figure 5.11: Compound leaves



```
#define D 1          /* apical delay */
#define R 1.36       /* internode elongation rate */

$\omega$  :  A(0)
$p_1$ :  A(d)  :  d>0  →  A(d-1)
$p_2$ :  A(d)  :  d=0  →  F(1)[+A(D)]F(1)B(0)
$p_3$ :  B(d)  :  d>0  →  B(d-1)
$p_4$ :  B(d)  :  d=0  →  F(1)[-B(D)]F(1)A(0)
$p_5$ :  F(a)  :  *    →  F(a*R)
```

Figure 5.12: Compound leaves with alternating branching patterns

Figure	D	R	Derivation length
a	1	1.36	20
b	4	1.18	34
c	7	1.13	46

Table 5.3: Values of the constants used to generate compound leaves with alternating branches

stants used to generate the structures shown in Figure 5.11 are listed in Table 5.2. The model is sensitive to growth rate values — a change of 0.01 visibly alters proportions. Leaves of the wild carrot, shown in Figure 3.23 (page 96), correspond closely to case Figure 5.11b.

Alternating branching

Another type of compound leaf, with alternating branches, is examined in Figure 5.12. The values of the constants used in the L-system are specified in Table 5.3. Further examples can be found in the next chapter, devoted to the animation of developmental processes, and Chapter 8, which ties developmental models with fractals.