Appendix A

Software environment for plant modeling

This book is illustrated with images of plants which exist only as mathematical models visualized by means of computer graphics. The software environment used to construct and experiment with these models includes dozens of programs and hundreds of data files. This creates the nontrivial problem of organizing all components for easy definition, saving, retrieval and modification of the models. In order to solve it, the idea of simulation was extended beyond the level of individual plants to an entire laboratory in botany [98]. Thus, a user can create and conduct experiments in a *virtual laboratory* by applying intuitive concepts and techniques from the "real" world. As an operating system defines the way a user perceives a computing environment, the virtual laboratory determines a user's perception of the environment in which simulated experiments take place. In the future, a virtual laboratory may complement, extend, or even replace books as a means for gathering and presenting scientific information. Because of this potential, the laboratory in which the research reported in this book was produced is described here in more detail.

A.1 A virtual laboratory in botany

User's perspective

A virtual laboratory, like its "real" counterpart, is a playground for experimentation. It comes with a set of *objects* pertinent to its scientific domain (in this case, plant models), *tools* which operate on these objects, a *reference book* and a *notebook.* Once the concepts and tools are understood, the user can expand the laboratory by adding new objects, creating new experiments, and recording descriptions in the notebook. An experienced user can expand the laboratory further by creating and installing new tools.

Laboratory = microworld + hypertext

Technically, a virtual laboratory is a *microworld* which can be explored under the guidance of a *hypertext* system. The term "microworld" denotes an interactive environment for creating and conducting simulated experiments. The guidance could be provided in the form of a traditional book, but an electronic document is more suitable for integration with a microworld. In a sense, both components of the virtual laboratory are described by Nelson in *Dream Machines* [104]. The pioneering role of this book in introducing the concept of hypertext is known, but under the heading *The Mind's Eye* the notion of a microworld is also anticipated:

> Suppose that you have a computer.
> What sorts of things would you do with it?
> Things that are imaginative
>   and don't require too much else.
> I am hinting at something.
> You could have it make pictures and show you stuff
> and change what it shows depending on what you do.

Requirements

A virtual laboratory can be divided into two components: the application programs, data files and textual descriptions that comprise the experiments; and the system support that provides the framework on which these domain-dependent experiments are built. The following list specifies the features of this framework.

- **Consistent organization of the lab.** In the lab environment, experiments are run by applying tools (programs) to objects (data files). An object consists of files that are grouped together so that they can be retrieved easily. The format of the objects is sufficiently standardized to allow straightforward implementation of common operations such as object saving and deletion.

- **Inheritance of features.** It is often the case that several objects differ only in details. For example, two lilac inflorescences may differ only in the color of their petals. The mechanism of inheritance is employed to store such objects efficiently.

- **Version control.** Interaction with an object during experimentation may result in a temporary or permanent modification. In the latter case, the user is able to decide whether the newly created object replaces the old one or should be stored as another version of the original object.

- **Interactive manipulation of objects.** The laboratory provides a set of general-purpose tools for manipulating object parameters. For example, objects can be modified using control panels or by editing specific fields in a textual description of an experiment.

- **Flexibility in conducting experiments.** The user may apply tools to objects in a dynamic way while an experiment is being conducted. This can be contrasted to a static experiment designed when the object is initially incorporated into the system.
- **Guidance through the laboratory.** A hypertext system imposes a logical organization on the set of objects, provides a textual description of the experiments, and makes it possible to browse through the experiments in many ways. Specific experiments are invoked automatically when the corresponding text is selected, in order to facilitate demonstrations and assist a novice user.

Objects

So far, objects have been referred to in an intuitive way, relying on the analogy between a real and virtual laboratory. For example, if our interest is in the development of the gametophyte *Microsorium linguaeforme*, in a real laboratory we would experiment with a specimen of the plant, while in a virtual laboratory we explore the corresponding mathematical model. However, the analogy to real objects does not extend to the level of detailed object definition. Specific design decisions are needed for software development purposes. In the current design, a laboratory object is defined as a directory containing two types of files and a subdirectory.

- The *data files* comprise our knowledge of a particular model.
- A *specification file* defines the data files which make up the object and the tools which apply to them.
- A directory of *extensions* lists objects which inherit some features of the current object.

The object-oriented file structure which provides the basis for lab operation can be represented by a hierarchy of directories and files (Figure A.1).

Inheritance of features

The path of subdirectories leading to an object establishes the inheritance structure for the lab. Inheritance is based on the idea of specifying new objects in reference to objects which already exist [81]. The "old" object is called a *prototype* and the new one is its *extension.* The extension contains only those files which are different from the corresponding files in the prototype. Files that remain the same are *delegated* to the prototype by establishing links. In other words, the object directory will contain those files that are unique to the object, and links to files that are inherited from its prototype (Figure A.2). This approach saves space, facilitates creation of objects similar to the prototype, and allows a single change in the prototype to propagate through all descendents.

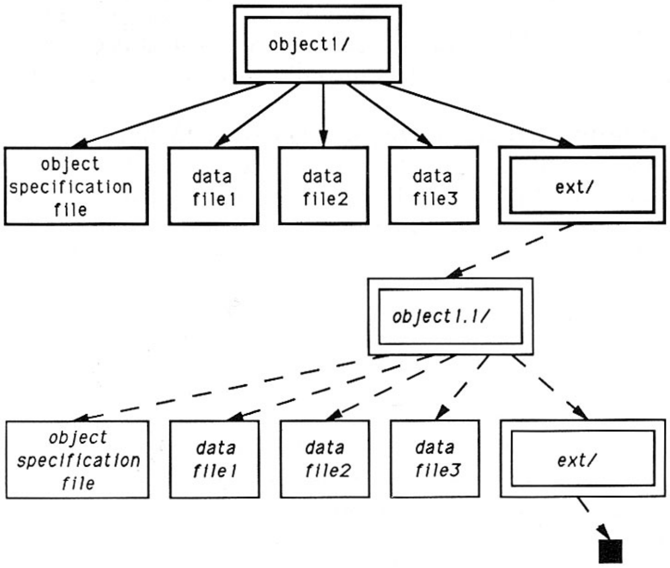

Figure A.1: The hierarchical structure of objects

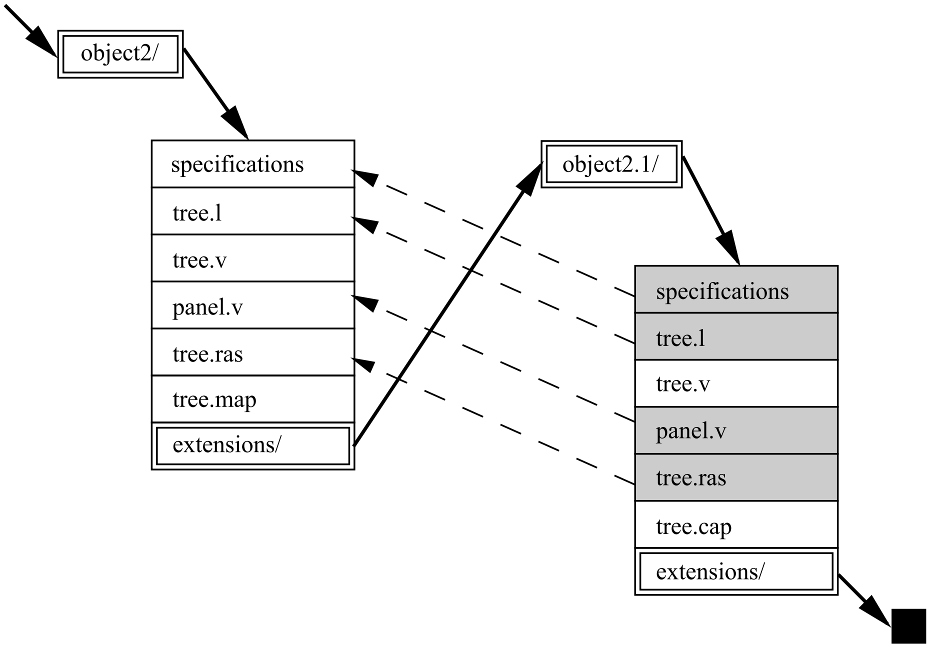

Figure A.2: A prototype and its extension. Shaded areas indicate links.

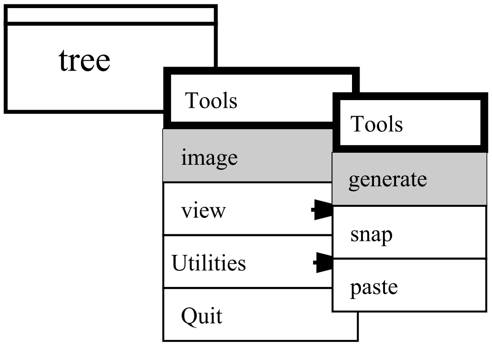

Figure A.3: An object icon with menus

Version control

To conduct an experiment, all files that make up the selected object are copied to a temporary location called the *lab table*. Consequently, manipulation of object parameters does not disturb the stored version. When the experiment is finished, the user may save the results by overwriting the original object or by creating an extension. In the latter case, the files on the lab table are compared with those in the prototype object; those files that differ from the prototype are saved, and links to the remaining files are established automatically.

Object manipulation

The ability to manipulate the parameters in an experiment easily is an essential feature of the virtual laboratory. As a rule, all parameters involved in an experiment are supplied to the tools through the object's data files. In order to modify a parameter, the user edits the appropriate file, which is subsequently re-read by the application. Though the editing of parameters can be accomplished using a text editor, in many cases parameter modification can be performed more conveniently using *virtual control panels* [114]. The current implementation of the laboratory provides the user with a general-purpose *control manager* which creates panels according to user-supplied configuration files.

Tool application

The user is able to apply a tool to an object as a whole, without detailed knowledge of the programs involved or the component files. This is achieved through the object's specification file which lists all files associated with an object and the tools that can be applied to them. This information is used to create a hierarchy of menus associated with an icon representing the object (Figure A.3). The end nodes in the hierarchy invoke tools that operate on the object. For example, selection of the item `image` followed by the item `generate` from the menus in the figure would invoke the plant modeling program *Pfg*.

Browsing

A user may browse through the objects in the lab by following either the hierarchical structure of objects or hypertext links. The *browser* is used to navigate through the hierarchy, moving down through successive extensions or up through previous levels. At any time, the user may request that an object be placed on the lab table. The hypertext document associated with the lab provides an alternative method of browsing and a means of relating objects independent of the hierarchy.

A.2 List of laboratory programs

The essential programs incorporated into the virtual laboratory in botany are listed below.

- *Plant and fractal generator (Pfg)*
  P. Prusinkiewicz and J. Hanan
  Given an L-system, a set of viewing parameters and optional files specifying predefined surfaces, *Pfg* generates the modeled structure by carrying out the derivation, then interpreting the resulting string using turtle geometry. Both non-parametric and parametric L-systems are supported. The model can be visualized directly on the screen of an IRIS workstation or output to a file. The first mode of operation is used to experiment with the model interactively and present developmental sequences. The output file can be either in Postscript format, particularly suitable for printing results such as fractal curves and inflorescence diagrams on a laser printer, or in the format required by the ray-tracer *Rayshade* for realistic rendering of the modeled structures.

- *Modeling program for phyllotactic patterns (Spiral)*
  D. R. Fowler
  Spiral is an interactive program for modeling organs with spiral phyllotactic patterns. The user can choose between planar and cylindrical patterns, and modify parameters which define model geometry (Chapter 4). This technique is faster than "growing" organs using parametric L-systems. Once an organ has been designed, it can be expressed using an L-system and incorporated into a plant structure.

- *Interactive surface editor (Ise)*
  J. Hanan
  Ise makes it possible to define and modify bicubic surfaces consisting of one or several arbitrarily connected patches. The output files produced by *Ise* are compatible with *Pfg* and *Spiral.*

- *Modeling program for cellular structures (Mapl)*
  F. D. Fracchia
  Mapl accepts the specification of a two-dimensional cell layer captured by a map L-system and generates the resulting developmental sequence using the dynamic method of map interpretation. Options include map generation on the surface of a sphere, and the simulation of development in three dimensions according to a given cellwork L-system. As in the case of *Pfg*, the models can be visualized directly on the screen or output to a file in either Postscript or *Rayshade* format.

- *Control panel manager (Panel)*
  L. Mercer and A. Snider
  This program creates control panels containing sliders and buttons, according to a configuration file provided by the user. Upon activation of a control by the mouse, *Panel* generates a message which indicates the corresponding control value. Application programs process this information and modify the appropriate parameters. For example, a panel can be used to control parameters used by *Pfg, Spiral,* or *Mapl.*

- *Ray tracer (Rayshade)*
  C. Kolb, Yale University
  Rayshade reads a scene description from a text file, and renders it using ray tracing. Scenes can be composed of primitives such as planes, triangles, polygons, spheres, cylinders, cones and height fields, grouped together to form objects. These objects can be instantiated in other object definitions to create a hierarchical description of a scene. Transformations including translation, rotation and scaling, and a variety of procedural textures can be applied to any object. Extended light sources, simulation of depth of field, and adaptive supersampling are supported. The program uses 3D grids to partition object space for fast intersection tests.

- *Previewer for the ray tracer (Preray)*
  A. Snider
  Preray is a previewer for *Rayshade* used to provide a fast wire frame rendering of a scene before committing time to ray tracing. A control panel associated with *Preray* makes it possible to set viewing parameters interactively.

- *Modeling program based on Euclidean constructions (L.E.G.O.)*
  N. Fuller
  L.E.G.O. makes it possible to model two- and three-dimensional objects using geometric constructions. In the scope of this book, *L.E.G.O.* was used to model man-made objects such as the *Zinnia* vase and the *Water-lilies* bridge.

- *Iterated function system generator (Ifsg)*
  D. Hepting
  A fractal defined by an iterated function system is described by a finite set of contractive affine transformations with an optional finite state control mechanism. *Ifsg* accepts input from a file specifying the transformations and rendering information. The program is capable of rendering by either attracting, distance-based or escape-time methods. The output can be displayed directly on an IRIS workstation or written to a file for further processing. In the scope of this book, *Ifsg* was used to obtain results which related plant models expressed using L-systems to fractals.

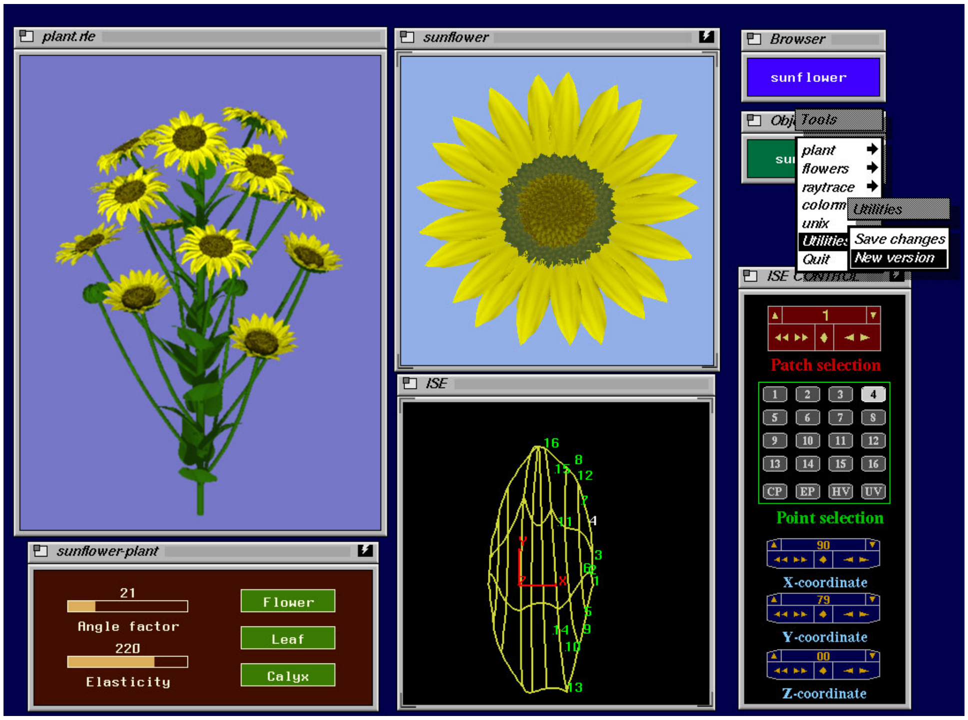

Figure A.4: A virtual laboratory screen

Figure A.4 presents a sample screen of a Silicon Graphics IRIS 4D/60 workstation running some of the above programs within the virtual laboratory framework. The icon in the top right corner represents the laboratory browser which was used to select a sunflower plant as the current object. The icon underneath and the associated menu were subsequently applied to select tools which operate on the object. The control panel in the bottom right corner of the screen is a part of the surface editor *Ise*. The manipulated petal is displayed as a wire frame in the window labeled *Ise*, and incorporated into a flower head by the modeling program *Spiral* which presents its output in the window *sunflower*. The flower heads are in turn incorporated into a complete plant model generated by *Pfg* and rendered using *Rayshade* in the window *plant.rle*. The panel below that window makes it possible to choose organs included in the model and change parameters related to the angles of the branching structure. The metaphor of a virtual laboratory provides a uniform interface to various operations on the selected plant, ranging from the modification of a petal to the rendering of the complete model.

Appendix B

About the figures

The following descriptions of the color images include details about the pictures not described in the main text. Unless otherwise stated, figures were created at the University of Regina.

Figure 1.19 [page 20] *Three-dimensional Hilbert curve*
F. D. Fracchia, P. Prusinkiewicz, N. Fuller (1989)

This image was rendered using ray-tracing without shadows.

Figure 1.25 [page 26] *Three-dimensional bush*
P. Prusinkiewicz (1986)

Simple branching structure, rendered using the firmware of a Silicon Graphics IRIS workstation. Total generating and rendering time on IRIS 4D/20: 4 seconds.

Figure 1.28 [page 29] *Flower field*
P. Prusinkiewicz (1986)

The field contains four rows of four plants. The scene was rendered with IRIS firmware, using depth-cueing to assign colors to petals.

Figure 1.35 [page 45] *Developmental stages of* Anabaena catenula
J. Hanan, P. Prusinkiewicz (1989)

Figure 2.1 [page 52] *Organic architecture*
Ned Greene, NYIT (1989)

An array of 300 x 300 x 300 voxel space automata was used to track a polygonal model of a house. Rendering was performed using a probabilistic radiosity method. See [54] for a full description.

Figure 2.3 [page 54] *Acer graphics*
Jules Bloomenthal, NYIT (1984)

A model of a maple tree. The basic branching structure was generated recursively. Limbs were modeled as generalized cylinders, obtained by moving discs of varying radii along spline curves. Real bark texture was digitized and used as a bump map. Leaf texture was obtained by digitizing a photograph of a real leaf and emphasizing the veins using a paint program. See [11] for details.

Figure 2.4 [page 54] *Forest scene*
Bill Reeves, Pixar (1984)

A scene from the film *The Adventures of André and Wally B*, modeled using particle systems. Shading and shadows were approximated using probabilistic techniques. Visible surfaces were determined using depth-sorting. See [119] for a full description.

Figure 2.5 [page 55] *Oil palm tree canopy*
CIRAD Modelisation Laboratory (1990)

A developmental model of oil palm trees, modeled using the method originated by de Reffye and described from the graphics perspective in [30].

Figure 2.10 [page 61] *Medicine lake*
F. K. Musgrave, C. E. Kolb, P. Prusinkiewicz,
B. B. Mandelbrot (1988)

A scene combining a fractal terrain model, a tree generated using L-systems, and a rainbow. The rainbow model was derived from a simulation of refraction with dispersion of light through an idealized raindrop. Procedural textures were applied to the mountains, the water surface and a vertical plane modeling the sky. See [101] for further details.

Figure 2.11 [page 62] *Surrealistic elevator*
A. Snider, P. Prusinkiewicz, N. Fuller (1989)

The elevator was modeled using L.E.G.O. The island is a superquadratic surface. Procedural textures were applied to create stars in the sky, craters on the moon, colored layers in the rock, waves in the lake and imperfections in the glass that covers the elevator.

Figure 3.2 [page 69] *Crocuses*
J. Hanan, D. R. Fowler (1990)

The petals were modeled as Bézier surfaces, with the shapes determined using lse.

Figure 3.4 [page 72] *Lily-of-the-valley*
P. Prusinkiewicz, J. Hanan (1987)

Figure 3.5 [page 74] *Development of* Capsella bursa-pastoris
P. Prusinkiewicz, A. Lindenmayer (1987)

Figure 3.6 [page 75] *Apple twig*
P. Prusinkiewicz, D. R. Fowler (1990)

This twig model was developed in one spring day, looking at a real twig nearby. This time is indicative for most inflorescence models shown.

Figure 3.11 [page 81] *A mint*
P. Prusinkiewicz (1988)

Figure 3.14 [page 84] *Development of* Lychnis coronaria
P. Prusinkiewicz, J. Hanan (1987)

Figure 3.17 [page 90] *Development of* Mycelis muralis
P. Prusinkiewicz, A. Lindenmayer (1987)

Figure 3.18 [page 91] *A three-dimensional rendering of the* Mycelis *models*
P. Prusinkiewicz, J. Hanan (1987)

All internodes in the model are assumed to have the same length. In reality, the internodes have different lengths, and the structure is less crowded.

Figure 3.19 [page 92] *Lilac inflorescences*
P. Prusinkiewicz, J. Hanan, D. R. Fowler (1990)

Figure 3.21 [page 94] *The Garden of L*
P. Prusinkiewicz, F. D. Fracchia, J. Hanan,
D. R. Fowler (1988)

All plants were modeled with L-systems and rendered using the IRIS firmware. Images corresponding to different viewing planes (the background lilac twigs, the apple twig and the daisies) were defocused separately using low-pass filters to simulate the depth of field, then composited with a focused image of lilac inflorescences. The sky was generated using a fractal algorithm.

Figure 3.23 [page 96] *Wild carrot*
P. Prusinkiewicz (1988)

Figure 4.3 [page 102] *Close-up of a daisy capitulum*
D. R. Fowler (1988)

The petals and florets were modeled as Bézier surfaces.

Figure 4.4 [page 102] *Domestic sunflower head*
D. R. Fowler, P. Prusinkiewicz (1989)

Figure 4.5 [page 105] *Sunflower field*
D. R. Fowler, N. Fuller, J. Hanan, A. Snider
(1990)

This image contains approximately 400 plants, each with 15 flowers. A flower has 21 petals and 300 seeds, modeled using 600 triangles and 400 triangles respectively. Counting leaves and buds, the entire scene contains about 800,000,000 triangles. The image was ray-traced with adaptive supersampling on a grid of 1024 x 768 pixels using 45 hours of CPU time on a MIPS M-120 computer.

Figure 4.6 [page 106] *Zinnias*
D. R. Fowler, P. Prusinkiewicz, J. Hanan,
N. Fuller (1990)

The vase was modeled using L.E.G.O. and rendered with a procedural texture. The scene was illuminated by one extended light source.

Figure 4.7 [page 106] *Close-up of zinnias*
D. R. Fowler, P. Prusinkiewicz, A. Snider (1990)

This scene was rendered using distributed ray-tracing to simulate the depth field.

Figure 4.8 [page 108] *Water-lily*
D. R. Fowler, J. Hanan (1990)

Figure 4.9 [page 108] *Lily pond*
D. R. Fowler, J. Hanan, P. Prusinkiewicz, N. Fuller (1990)

The wavelets on the water surface were obtained using bump-mapping with a procedurally defined texture.

Figure 4.10 [page 109] *Roses*
D. R. Fowler, J. Hanan, P. Prusinkiewicz (1990)

Distributed ray-tracing with one extended light source was used to simulate depth of field and create fuzzy shadows.

Figure 4.11 [page 111] *Parastichies on a cylinder*
D. R. Fowler (1990)

Figure 4.15 [page 116] *Pineapples*
D. R. Fowler, A. Snider (1990)

The image incorporates a physically-based model of a tablecloth approximated as an array of masses connected by springs and placed in a gravitational field. The scene is illuminated by three extended light sources.

Figure 4.16 [page 117] *Spruce cones*
D. R. Fowler, J. Hanan (1990)

Figure 4.17 [page 117] Carex laevigata
J. Hanan, P. Prusinkiewicz (1989)

The entire plant, including the leaves, was modeled using parametric L-systems.

Figure 5.2 [page 121] *Maraldi figure*
Ned Greene, NYIT (1984)

The shapes of leaves, calyxes and petals were defined using a paint program, by interpreting gray levels as height. Painted textures were mapped onto the surfaces of leaves and calyxes. Smooth gradation of color across the petals was obtained by assigning colors to the vertices of the polygon meshes representing flowers, then interpolating colors across polygons using Gouraud shading. The vines were rendered with bump-mapping, using a digitized image of real bark.

Figure 5.3 [page 121] *The fern*
P. Prusinkiewicz (1986)

Figure 5.7 [page 125] *A rose in a vase*
D. R. Fowler, J. Hanan, P. Prusinkiewicz (1990)

Petals and thorns are Bézier surfaces incorporated into a rose model expressed using L-systems. The vase was modeled as a surface of revolution.

Figure 6.3 [page 141] *Development of* Anabaena catenula
P. Prusinkiewicz, F. D. Fracchia (1989)

Each developmental stage is plotted in one scan line.

Figure 7.13 [page 161] *Simulated development of* Microsorium linguaeforme
F. D. Fracchia, P. Prusinkiewicz,
M. J. M. de Boer (1989)

Cells are represented as polygons, rendered using the IRIS firmware. The development can be visualized directly on the screen of an IRIS 4D/20 workstation without resorting to single-frame animation techniques.

Figure 7.14 [page 161] *Microphotograph of* Microsorium linguaeforme
M. J. M. de Boer, University of Utrecht

Figure 7.16 [page 163] *Simulated development of* Dryopteris thelypteris
F. D. Fracchia, P. Prusinkiewicz,
M. J. M. de Boer (1989)

Figure 7.19 [page 169] *Developmental sequence of* Patella vulgata
F. D. Fracchia, A. Lindenmayer,
M. J. M. de Boer (1989)

Cells are represented as spheres. Intersections of spheres inside the modeled embryo are ignored, since they do not affect the ray-traced images.

Figure 7.20 [page 169] *An electron microscope image of* Patella vulgata
W. J. Dictus, University of Utrecht

Figure 8.4 [page 180] *Fern dune*
P. Prusinkiewicz, D. Hepting (1989)

The shape of the leaf has been captured using a controlled iterated function system. A continuous escape-time function defines point altitudes, resulting in a surrealistic incorporation of a leaf into the landscape.

Figure 8.9 [page 186] *Carrot leaf*
D. Hepting, P. Prusinkiewicz (1989)

The leaf shape has been modeled using a controlled iterated function system. The scene consists of a set of spheres, with the radius equal to the distance to the leaf. The image was rendered using ray-tracing.

Figure E.1 [page 191] *Water-lilies*
D. R. Fowler, J. Hanan, P. Prusinkiewicz,
N. Fuller (1990)

A scene inspired by *Water-lilies pool - Harmony in green* by Claude Monet (1899). All trees and water-lilies were modeled using L-systems. The willow twigs bend downwards due to a strong tropism effect, simulating gravity. The bridge was modeled using L.E.G.O. The sky is a sphere with a procedural texture. The entire scene was ray-traced, then the resulting image was represented as a set of small circles, with the colors close but not equal to the average of pixel colors underneath. This last operation was aimed at creating the appearance of an impressionistic painting.

Figure A.4 [page 200] *Virtual lab*
L. Mercer, D. R. Fowler (1990)

Turtle interpretation of symbols

Symbol	Interpretation	Page
F	Move forward and draw a line.	7, 46
f	Move forward without drawing a line.	7, 46
+	Turn left.	7, 19, 46
−	Turn right.	7, 19
∧	Pitch up.	19, 46
&	Pitch down.	19, 46
\	Roll left.	19, 46
/	Roll right.	19, 46
$\vert$	Turn around.	19, 46
$	Rotate the turtle to vertical.	57
[	Start a branch.	24
]	Complete a branch.	24
{	Start a polygon.	120, 127
G	Move forward and draw a line. Do not record a vertex.	122
.	Record a vertex in the current polygon.	122, 127
}	Complete a polygon.	120, 127
∼	Incorporate a predefined surface.	119
!	Decrement the diameter of segments.	26, 57
′	Increment the current color index.	26
%	Cut off the remainder of the branch.	73

Bibliography

[1] H. Abelson and A. A. diSessa. *Turtle geometry.* M.I.T. Press, Cambridge, 1982.

[2] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, 1984.

[3] M. J. Apter. *Cybernetics and development.* Pergamon Press, Oxford, 1966. (International Series of Monographs in Pure and Applied Biology/Zoology Division Vol. 29).

[4] W. W. Armstrong. The dynamics of tree linkages with a fixed root link and limited range of rotation. *Actes du Colloque Internationale l'Imaginaire Numérique '86*, pages 16–21, 1986.

[5] J. W. Backus. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference. In *Proc. Intl. Conf. on Information Processing*, pages 125–132. UNESCO, 1959.

[6] B. I. Balinsky. *An introduction to embryology.* W. B. Saunders, Philadelphia, 1970.

[7] M. F. Barnsley. *Fractals everywhere.* Academic Press, San Diego, 1988.

[8] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constructive Approximation*, 5:3–31, 1989.

[9] B. A. Barsky. The Beta-spline: A local representation based on shape parameters and fundamental geometric measures. PhD thesis, Department of Computer Science, University of Utah, 1981.

[10] R. Bartels, J. Beatty, and B. Barsky, editors. *An introduction to splines for use in computer graphics and geometric modeling.* Morgan Kaufman, Los Altos, California, 1987.

[11] J. Bloomenthal. Modeling the mighty maple. Proceedings of SIGGRAPH '85 (San Francisco, California, July 22-26, 1985) in *Computer Graphics*, 19, 3 (July 1985), pages 305–311, ACM SIGGRAPH, New York, 1985.

[12] B. G. Briggs and L. A. S. Johnson. Evolution in the Myrtaceae – evidence from inflorescence structure, Appendix I: The relevance of Troll's system of inflorescence typology. In *Proceedings of the Linnean Society of New South Wales*, volume 102, pages 236–240, 1979.

[13] N. Chomsky. Three models for the description of language. *IRE Trans. on Information Theory*, 2(3):113–124, 1956.

[14] V. Claus, H. Ehrig, and G. Rozenberg, editors. *Graph grammars and their application to computer science; First International Workshop.* Lecture Notes in Computer Science 73. Springer-Verlag, Berlin, 1979.

[15] D. Cohen. Computer simulation of biological pattern generation processes. *Nature*, 216:246–248, 1967.

[16] E. Costes. Analyse architecturale et modelisation du litchi. PhD thesis, Université des Sciences et Techniques du Languedoc, 1988.

[17] H. S. M. Coxeter. *Introduction to geometry.* J. Wiley & Sons, New York, 1961.

[18] H. S. M. Coxeter. The role of intermediate convergents in Tait's explanation for phyllotaxis. *J. Algebra*, 20:167–175, 1972.

[19] K. Culik II and D. Wood. A mathematical investigation of propagating graph OL-systems. *Information and Control*, 43:50–82, 1979.

[20] P. Dabadie, P. de Reffye, and P. Dinouard. Modélisation de la croissance et de l'architecture d'un bambou. In *Deuxième congrès international du bambou*, 1988.

[21] C. Davis and D. E. Knuth. Number representations and dragon curves. *J. of Recreational Mathematics*, 3:66–81, 133–149, 1970.

[22] M. J. M. de Boer. Analysis and computer generation of division patterns in cell layers using developmental algorithms. PhD thesis, University of Utrecht, the Netherlands, 1989.

[23] M. J. M. de Boer and A. Lindenmayer. Map 0L-systems with edge label control: Comparison of marker and cyclic systems. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph-grammars and their application to computer science*, Lecture Notes in Computer Science 291, pages 378–392. Springer-Verlag, 1987.

[24] M. de Does and A. Lindenmayer. Algorithms for the generation and drawing of maps representing cell clones. In H. Ehrig, M. Nagl, and G. Rozenberg, editors, *Graph-grammars and their application to computer science*, Lecture Notes in Computer Science 153, pages 39–57. Springer-Verlag, 1983.

[25] C. G. de Koster and A. Lindenmayer. Discrete and continuous models for heterocyst differentiation in growing filaments of blue-green bacteria. In *Acta Biotheoretica*, volume 36, pages 249–273. Kluwer Academic, the Netherlands, 1987.

[26] P. de Reffye. Modéle mathématique aléatorie et simulation de la croissance et de l'architecture du caféier robusta. Première partie, *Café-Cacao-Thé*, 25(2):83–104, 1981. Deuxième partie, *Café-Cacao-Thé*, 25(4):219–230, 1981. Troisième partie, *Café-Cacao-Thé*, 26(2):77–96, 1982. Quatrième partie, *Café-Cacao-Thé*, 27(1):3–20, 1983.

[27] P. de Reffye. Travaux sur la caféier. CIRAD, Montpellier. Collection of earlier papers.

[28] P. de Reffye. Modélisation et simulation de la verse de caféier, à l'aide de la théorie de la résistance des matériaux. *Café-Cacao-Thé*, XX(4):251–272, 1976.

[29] P. de Reffye, M. Cognee, M. Jaeger, and B. Traore. Modélisation de la croissance et de l'architecture du cotonnier. Manuscript, 1988.

[30] P. de Reffye, C. Edilin, J. Françon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. Proceedings of SIGGRAPH '88 (Atlanta, Georgia, August 1-5, 1988), in *Computer Graphics* 22,4 (August 1988), pages 151–158, ACM SIGGRAPH, New York, 1988.

[31] F. M. Dekking. Recurrent sets. *Advances in Mathematics*, 44(1):78–104, 1982.

[32] F. M. Dekking. Recurrent sets: A fractal formalism. Report 82-32, Delft University of Technology, 1982.

[33] H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors. *Graph grammars and their application to computer science; Third International Workshop.* Lecture Notes in Computer Science 291. Springer-Verlag, Berlin, 1987.

[34] H. Ehrig, M. Nagl, and G. Rozenberg, editors. *Graph grammars and their application to computer science; Second International Workshop.* Lecture Notes in Computer Science 153. Springer-Verlag, Berlin, 1983.

[35] P. Eichhorst and W. J. Savitch. Growth functions of stochastic Lindenmayer systems. *Information and Control*, 45:217–228, 1980.

[36] R. O. Erickson. The geometry of phyllotaxis. In J. E. Dale and F. L. Milthrope, editors, *The growth and functioning of leaves*, pages 53–88. University Press, Cambridge, 1983.

[37] G. Eyrolles. Synthèse d'images figuratives d'arbres par des méthodes combinatoires. PhD thesis, Université de Bordeaux I, 1986.

[38] J. B. Fisher and H. Honda. Computer simulation of branching pattern and geometry in Terminalia (Combretaceae), a tropical tree. *Botanical Gazette*, 138(4):377–384, 1977.

[39] J. B. Fisher and H. Honda. Branch geometry and effective leaf area: A study of Terminalia–branching pattern, Parts I and II. *American Journal of Botany*, 66:633–655, 1979.

[40] J. D. Foley and A. Van Dam. *Fundamentals of interactive computer graphics*. Addison-Wesley, Reading, Massachusetts, 1982.

[41] L. Fox and D. F. Mayers. *Numerical solution of ordinary differential equations*. Chapman and Hall, London, 1987.

[42] F. D. Fracchia, P. Prusinkiewicz, and M. J. M. de Boer. Visualization of the development of multicellular structures. In *Proceedings of Graphics Interface '90*, pages 267–277, 1990.

[43] H. Freeman. On encoding arbitrary geometric configurations. *IRE Trans. Electronic. Computers*, 10:260–268, 1961.

[44] D. Frijters. Mechanisms of developmental integration of *Aster novae-angliae* L. and *Hieracium murorum* L. *Annals of Botany*, 42:561–575, 1978.

[45] D. Frijters. Principles of simulation of inflorescence development. *Annals of Botany*, 42:549–560, 1978.

[46] D. Frijters and A. Lindenmayer. A model for the growth and flowering of *Aster novae-angliae* on the basis of table (1,0)L-systems. In G. Rozenberg and A. Salomaa, editors, *L Systems*, Lecture Notes in Computer Science 15, pages 24–52. Springer-Verlag, Berlin, 1974.

[47] D. Frijters and A. Lindenmayer. Developmental descriptions of branching patterns with paracladial relationships. In A. Lindenmayer and G. Rozenberg, editors, *Automata, languages, development*, pages 57–73. North-Holland, Amsterdam, 1976.

[48] M. Gardner. Mathematical games: An array of problems that can be solved with elementary mathematical techniques. *Scientific American*, 216, 1967. 3:124–129 (March), 4:116–123 (April).

[49] M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223(4):120–123, October 1970.

[50] M. Gardner. Mathematical games: On cellular automata, self-reproduction, the Garden of Eden and the game "life". *Scientific American*, 224(2):112–117, February 1971.

[51] M. Gardner. Mathematical games – in which "monster" curves force redefinition of the word "curve". *Scientific American*, 235(6):124–134, December 1976.

[52] S. Ginsburg and H. G. Rice. Two families of languages related to ALGOL. *J. ACM*, 9(3):350–371, 1962.

[53] H. Gravelius. *Flusskunde.* Goschen, Berlin, 1914.

[54] N. Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. Proceedings of SIGGRAPH '89 (Boston, Mass., July 31-August 4, 1989), in *Computer Graphics* 23,4 (August 1989), pages 175–184, ACM SIGGRAPH, New York, 1989.

[55] B. E. S. Gunning. Microtubules and cytomorphogenesis in a developing organ: The root primordium of *Azolla pinnata*. In O. Kiermayer, editor, *Cytomorphogenesis in plants*, Cell Biology Monographs 8, pages 301–325. Springer-Verlag, Wien, 1981.

[56] A. Habel and H.-J. Kreowski. On context-free graph languages generated by edge replacement. In H. Ehrig, M. Nagl, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Second International Workshop*, Lecture Notes in Computer Science 153, pages 143–158. Springer-Verlag, Berlin, 1983.

[57] A. Habel and H.-J. Kreowski. May we introduce to you: Hyperedge replacement. In H. Ehrig, M. Nagl, G. Rozenberg, and A. Posenfeld, editors, *Graph grammars and their application to computer science; Third International Workshop*, Lecture Notes in Computer Science 291, pages 15–26. Springer-Verlag, Berlin, 1987.

[58] F. Hallé, R. A. A. Oldeman, and P. B. Tomlinson. *Tropical trees and forests: An architectural analysis.* Springer-Verlag, Berlin, 1978.

[59] P. H. Hellendoorn and A. Lindenmayer. Phyllotaxis in *Bryophyllum tubiflorum*: Morphogenetic studies and computer simulations. *Acta Biol. Neerl*, 23(4):473–492, 1974.

[60] D. Hepting, P. Prusinkiewicz, and D. Saupe. Rendering methods for iterated function systems. Manuscript, 1990.

[61] G. Herman, A. Lindenmayer, and G. Rozenberg. Description of developmental languages using recurrence systems. *Mathematical Systems Theory*, 8:316–341, 1975.

[62] G. T. Herman and G. Rozenberg. *Developmental systems and languages.* North-Holland, Amsterdam, 1975.

[63] D. Hilbert. Ueber stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalin.*, 38:459–460, 1891.

[64] P. Hogeweg and B. Hesper. A model study on biomorphological description. *Pattern Recognition*, 6:165–179, 1974.

[65] H. Honda. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology*, 31:331–338, 1971.

[66] H. Honda and J. B. Fisher. Tree branch angle: Maximizing effective leaf area. *Science*, 199:888–890, 1978.

[67] H. Honda and J. B. Fisher. Ratio of tree branch lengths: The equitable distribution of leaf clusters on branches. *Proceedings of the National Academy of Sciences USA*, 76(8):3875–3879, 1979.

[68] H. Honda, P. B. Tomlinson, and J. B. Fisher. Computer simulation of branch interaction and regulation by unequal flow rates in botanical trees. *American Journal of Botany*, 68:569–585, 1981.

[69] H. Honda, P. B. Tomlinson, and J. B. Fisher. Two geometrical models of branching of botanical trees. *Annals of Botany*, 49:1–11, 1982.

[70] R. E. Horton. Erosioned development of systems and their drainage basins, hydrophysical approach to quantitative morphology. *Bull. Geol. Soc. America*, 56:275–370, 1945.

[71] R. E. Horton. Hypsometric (area-altitude) analysis of erosional topology. *Bull. Geol. Soc. America*, 63:1117–1142, 1952.

[72] R. Hunt. *Plant growth analysis.* Studies in Biology 96. Edward Arnold, London, 1978.

[73] R. Hunt. *Plant growth curves – the functional approach to plant growth analysis*. Edward Arnold, London, 1982.

[74] J. E. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30(5):713–747, 1981.

[75] G. Van Iterson. *Mathematische und mikroskopish-anatomische Studien über Blattstellungen*. Gustav Fischer, Jena, 1907.

[76] M. Jaeger. Représentation et simulation de croissance des végétaux. PhD thesis, Université Louis Pasteur de Strasbourg, 1987.

[77] J. M. Janssen and A. Lindenmayer. Models for the control of branch positions and flowering sequences of capitula in *Mycelis muralis* (L.) Dumont (Compositae). *New Phytologist*, 105:191–220, 1987.

[78] R. V. Jean. Mathematical modelling in phyllotaxis: The state of the art. *Mathematical Biosciences*, 64:1–27, 1983.

[79] H. Jürgensen and A. Lindenmayer. Modelling development by OL-systems: Inference algorithms for developmental systems with cell lineages. *Bulletin of Mathematical Biology*, 49(1):93–123, 1987.

[80] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Int. J. Comp. Math*, 2:157–168, 1968.

[81] H. Lieberman. Using prototypical objects to implement shared behavior in object oriented systems. In *Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications*, pages 214–223, New York, 1986. Association for Computing Machinery.

[82] A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.

[83] A. Lindenmayer. Adding continuous components to L-systems. In G. Rozenberg and A. Salomaa, editors, *L Systems*, Lecture Notes in Computer Science 15, pages 53–68. Springer-Verlag, Berlin, 1974.

[84] A. Lindenmayer. Developmental algorithms: Lineage versus interactive control mechanisms. In S. Subtelny and P. B. Green, editors, *Developmental order: Its origin and regulation*, pages 219–245. Alan R. Liss, New York, 1982.

[85] A. Lindenmayer. Models for plant tissue development with cell division orientation regulated by preprophase bands of microtubules. *Differentiation*, 26:1–10, 1984.

[86] A. Lindenmayer. Positional and temporal control mechanisms in inflorescence development. In P. W. Barlow and D. J. Carr, editors, *Positional controls in plant development.* University Press, Cambridge, 1984.

[87] A. Lindenmayer. An introduction to parallel map generating systems. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Third International Workshop*, Lecture Notes in Computer Science 291, pages 27–40. Springer-Verlag, Berlin, 1987.

[88] A. Lindenmayer. Models for multicellular development: Characterization, inference and complexity of L-systems. In A. Kelmenová and J. Kelmen, editors, *Trends, techniques and problems in theoretical computer science*, Lecture Notes in Computer Science 281, pages 138–168. Springer-Verlag, Berlin, 1987.

[89] A. Lindenmayer and P. Prusinkiewicz. Developmental models of multicellular organisms: A computer graphics perspective. In C. Langton, editor, *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems held September, 1987, in Los Alamos, New Mexico*, pages 221–249. Addison-Wesley, Redwood City, 1989.

[90] A. Lindenmayer and G. Rozenberg, editors. *Automata, languages, development.* North-Holland, Amsterdam, 1976.

[91] A. Lindenmayer and G. Rozenberg. Parallel generation of maps: Developmental systems for cell layers. In V. Claus, H. Ehrig, and G. Rozenberg, editors, *Graph grammars and their application to computer science; First International Workshop*, Lecture Notes in Computer Science 73, pages 301–316. Springer-Verlag, Berlin, 1979.

[92] J. Lück, A. Lindenmayer, and H. B. Lück. Models for cell tetrads and clones in meristematic cell layers. *Botanical Gazette*, 149:1127–141, 1988.

[93] J. Lück and H. B. Lück. Generation of 3-dimensional plant bodies by double wall map and stereomap systems. In H. Ehrig, M. Nagl, and G. Rozenberg, editors, *Graph Grammars and Their Application to Computer Science; Second International Workshop*, Lecture Notes in Computer Science 153, pages 219–231. Springer-Verlag, Berlin, 1983.

[94] N. Macdonald. *Trees and networks in biological models.* J. Wiley & Sons, New York, 1983.

[95] B. B. Mandelbrot. *The fractal geometry of nature.* W. H. Freeman, San Francisco, 1982.

[96] D. M. McKenna. SquaRecurves, E-tours, eddies and frenzies: Basic families of Peano curves on the square grid. In *Proceedings of the Eugene Strens Memorial Conference on Recreational Mathematics and its History*, 1989. To appear.

[97] H. Meinhardt. *Models of biological pattern formation.* Academic Press, New York, 1982.

[98] L. Mercer, P. Prusinkiewicz, and J. Hanan. The concept and design of a virtual laboratory. In *Proceedings of Graphics Interface '90*, pages 149–155. CIPS, 1990.

[99] G. J. Mitchison and Michael Wilcox. Rules governing cell division in *Anabaena. Nature*, 239:110–111, 1972.

[100] D. Müller-Doblies and U. Müller-Doblies. Cautious improvement of a descriptive terminology of inflorescences. *Monocot Newsletter 4*, 1987.

[101] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. Proceedings of SIGGRAPH '89 (Boston, Mass., July 31-August 4, 1989), in *Computer Graphics* 23,4 (August 1989), pages 41–50, ACM SIGGRAPH, New York, 1989.

[102] A. Nakamura, A. Lindenmayer, and K. Aizawa. Some systems for map generation. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 323–332. Springer-Verlag, Berlin, 1986.

[103] P. Naur et al. Report on the algorithmic language ALGOL 60. *Communications of the ACM*, 3(5):299–314, 1960. Revised in Comm. ACM 6(1):1-17.

[104] T. Nelson. Computer lib and dream machines. Self-published, 1980.

[105] P. Oppenheimer. Real time design and animation of fractal plants and trees. *Computer Graphics*, 20(4):55–64, 1986.

[106] G. Peano. Sur une courbe, qui remplit tout une aire plaine. *Math. Annln.*, 36:157–160, 1890. Translated in G. Peano, *Selected works of Giuseppe Peano*, H. C. Kennedy, editor, pages 143–149, University of Toronto Press, Toronto, 1973.

[107] H. Peitgen and D. Saupe, editors. *The science of fractal images.* Springer-Verlag, New York, 1988.

[108] F. P. Preparata and R. T. Yeh. *Introduction to Discrete Structures.* Addison-Wesley, Reading, Massachusetts, 1973.

[109] P. Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 — Vision Interface '86*, pages 247–253. CIPS, 1986.

[110] P. Prusinkiewicz. Score generation with L-systems. In *Proceedings of the International Computer Music Conference '86*, pages 455–457, 1986.

[111] P. Prusinkiewicz. Applications of L-systems to computer imagery. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Third International Workshop*, pages 534–548. Springer-Verlag, Berlin, 1987. Lecture Notes in Computer Science 291.

[112] P. Prusinkiewicz and J. Hanan. *Lindenmayer systems, fractals, and plants*, volume 79 of *Lecture Notes in Biomathematics.* Springer-Verlag, Berlin, 1989.

[113] P. Prusinkiewicz and J. Hanan. Visualization of botanical structures and processes using parametric L-systems. In D. Thalmann, editor, *Scientific Visualization and Graphics Simulation*, pages 183–201. J. Wiley & Sons, 1990.

[114] P. Prusinkiewicz and K. Krithivasan. Algorithmic generation of South Indian folk art patterns. In *Proceedings of the International Conference on Computer Graphics ICONCG '88*, Singapore, 1988.

[115] P. Prusinkiewicz, K. Krithivasan, and M. G. Vijayanarayana. Application of L-systems to algorithmic generation of South Indian folk art patterns and karnatic music. In R. Narasimhan, editor, *A perspective in theoretical computer science — commemorative volume for Gift Siromoney*, pages 229–247. World Scientific, Singapore, 1989. Series in Computer Science Vol. 16.

[116] P. Prusinkiewicz, A. Lindenmayer, and F. D. Fracchia. Synthesis of space-filling curves on the square grid. To appear in *Proceedings of FRACTAL '90*, the 1st IFIP conference on fractals, Lisbon, Portugal, June 6-8, 1990.

[117] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan. Developmental models of herbaceous plants for computer imagery purposes. Proceedings of SIGGRAPH '88 (Atlanta, Georgia, August 1-5, 1988), in *Computer Graphics* 22,4 (August 1988), pages 141–150, ACM SIGGRAPH, New York, 1988.

[118] P. Prusinkiewicz and G. Sandness. Koch curves as attractors and repellers. *IEEE Computer Graphics and Applications*, 8(6):26–40, 1988.

[119] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. Proceedings of SIGGRAPH '85 (San Francisco, California, July 22-26, 1985) in *Computer Graphics*, 19, 3 (July 1985), pages 313–322, ACM SIGGRAPH, New York, 1985.

[120] W. R. Remphrey, B. R. Neal, and T. A. Steeves. The morphology and growth of *Arctostaphylos uva-ursi* (bearberry), parts i and ii. *Canadian Journal of Botany*, 61(9):2430–2458, 1983.

[121] W. R. Remphrey and G. R. Powell. Crown architecture of *Larix laricina* saplings: Quantitative analysis and modelling of (nonsylleptic) order 1 branching in relation to development of the main stem. *Canadian Journal of Botany*, 62(9):1904–1915, 1984.

[122] W. R. Remphrey and G. R. Powell. Crown architecture of *Larix laricina* saplings: Sylleptic branching on the main stem. *Canadian Journal of Botany*, 63(7):1296–1302, 1985.

[123] L. H. Reuter. Rendering and magnification of fractals using interated function systems. PhD thesis, Georgia Institute of Technology, 1987.

[124] J. N. Ridley. Computer simulation of contact pressure in capitula. *Journal of Theoretical Biology*, 95:1–11, 1982.

[125] J. N. Ridley. Packing efficiency in sunflower heads. *Mathematical Biosciences*, 58:129–139, 1982.

[126] D. F. Robinson. A notation for the growth of inflorescences. *New Phytologist*, 103:587–596, 1986.

[127] G. Rozenberg and A. Salomaa. *The mathematical theory of L-systems.* Academic Press, New York, 1980.

[128] A. Salomaa. *Formal languages.* Academic Press, New York, 1973.

[129] F. W. Sears, M. W. Zemansky, and H. D. Young. *College physics.* Addison-Wesley Publ. Co., Reading, 6th edition, 1985.

[130] M. Shebell. Modeling branching plants using attribute L-systems. Master's thesis, Worcester Polytechnic Institute, 1986.

[131] P. L. J. Siero, G. Rozenberg, and A. Lindenmayer. Cell division patterns: Syntactical description and implementation. *Computer Graphics and Image Processing*, 18:329–346, 1982.

[132] W. Sierpiński. Sur une courbe dont tout point est un point de ramification. *Comptes Rendus hebdomadaires des séances de l'Académie des Sciences*, 160:302–305, 1915. Reprinted in W. Sierpiński, *Oeuvres choisies*, S. Hartman et al., editors, pages 99–106, PWN – Éditions Scientifiques de Pologne, Warsaw, 1975.

[133] G. Siromoney and R. Siromoney. Rosenfeld's cycle grammars and kolam. In H. Ehrig, M. Nagl, A. Rosenfeld, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Third International Workshop*, Lecture Notes in Computer Science 291, pages 564–579. Springer-Verlag, Berlin, 1987.

[134] G. Siromoney, R. Siromoney, and T. Robinsin. Kambi kolam and cycle grammars. In R. Narasimhan, editor, *A perspective in theoretical computer science — commemorative volume for Gift Siromoney*, Series in Computer Science Vol. 16, pages 267–300. World Scientific, Singapore, 1989.

[135] R. Siromoney and K. G. Subramanian. Space-filling curves and infinite graphs. In H. Ehrig, M. Nagl, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Second International Workshop*, Lecture Notes in Computer Science 153, pages 380–391. Springer-Verlag, Berlin, 1983.

[136] A. R. Smith. Plants, fractals, and formal languages. Proceedings of SIGGRAPH '84 (Minneapolis, Minnesota, July 22-27, 1984) in *Computer Graphics*, 18, 3 (July 1984), pages 1–10, ACM SIGGRAPH, New York, 1984.

[137] A. R. Smith. About the cover: Reconfigurable machines. *Computer*, 11(7):3–4, 1978.

[138] P. S. Stevens. *Patterns in nature.* Little, Brown and Co., Boston, 1974.

[139] R. J. Stevens, A. F. Lehar, and F. H. Perston. Manipulation and presentation of multidimensional image data using the Peano scan. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-5(5):520–526, 1983.

[140] A. L. Szilard. Growth functions of Lindenmayer systems. Technical Report 4, Computer Science Department, University of Western Ontario, 1971.

[141] A. L. Szilard and R. E. Quinton. An interpretation for DOL systems by computer graphics. *The Science Terrapin*, 4:8–13, 1979.

[142] R. Thom. *Structural stability and morphogenesis. An outline of a general theory of models.* Benjamin/Cummings, Reading, Massachusetts, 1975.

[143] d'Arcy Thompson. *On growth and form.* University Press, Cambridge, 1952.

[144] W. Troll. *Die Infloreszenzen*, volume I. Gustav Fischer Verlag, Stuttgart, 1964.

[145] W. Troll. *Die Infloreszenzen*, volume II. Gustav Fischer Verlag, Jena, 1969.

[146] A. Turing. On computable numbers with an application to the Entscheidungsproblem, 1936. *Proc. Lond. Math. Soc.* (ser. 2), 42:230–265, 1936–37, and 43:544–546, 1937.

[147] A. Turing. The chemical basis of morphogenesis. *Philosophical Trans. Roy. Soc. B*, 237(32):5–72, 1952.

[148] W. T. Tutte. *Graph theory.* Addison-Wesley, Reading, Massachusetts, 1982.

[149] S. Ulam. Patterns of growth of figures: Mathematical aspects. In G. Kepes, editor, *Module, Proportion, Symmetry, Rhythm*, pages 64–74. Braziller, New York, 1966.

[150] J. A. M. van den Biggelaar. Development of dorsoventral polarity and mesentoblast determination in *Patella vulgata. Journal of Morphology*, 154:157–186, 1977.

[151] A. H. Veen and A. Lindenmayer. Diffusion mechanism for phyllotaxis: Theoretical physico-chemical and computer study. *Plant Physiology*, 60:127–139, 1977.

[152] X. G. Viennot, G. Eyrolles, N. Janey, and D. Arquès. Combinatorial analysis of ramified patterns and computer imagery of trees. Proceedings of SIGGRAPH '89 (Boston, Mass., July 31-August 4, 1989), in *Computer Graphics* 23,4 (August 1989), pages 31–40, ACM SIGGRAPH, New York, 1989.

[153] P. M. B. Vitányi. Development, growth and time. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 431–444. Springer-Verlag, Berlin, 1986.

[154] H. Vogel. A better way to construct the sunflower head. *Mathematical Biosciences*, 44:179–189, 1979.

[155] H. von Koch. Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. *Acta mathematica*, 30:145–174, 1905.

[156] J. von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, Urbana, 1966. Edited by A. W. Burks.

[157] F. Weberling. Typology of inflorescences. *J. Linn. Soc. (Bot.)*, 59(378):215–222, 1965.

[158] F. Weberling. *Morphologie der Blüten und der Blütenstände*. Verlag Eugen Ulmer, Stuttgart, 1981.

[159] H. Weyl. *Symmetry*. Princeton University Press, Princeton, New Jersey, 1982.

[160] S. Wolfram. Computer software in science and mathematics. *Scientific American*, 251(3):188–203, 1984.

[161] S. Wolfram. Some recent results and questions about cellular automata. In J. Demongeot, E. Goles, and M. Tchuente, editors, *Dynamical systems and cellular automata*, pages 153–167. Academic Press, London, 1985.

[162] T. Yokomori. Stochastic characterizations of EOL languages. *Information and Control*, 45:26–33, 1980.

[163] D. A. Young. On the diffusion theory of phyllotaxis. *Journal of Theoretical Biology*, 71:421–423, 1978.

[164] M. H. Zimmerman and C. L. Brown. *Trees — structure and function*. Springer-Verlag, Berlin, 1971.

Index