



**HAL**  
open science

## A look at the visual modeling of plants using L-systems

Przemyslaw Prusinkiewicz

► **To cite this version:**

Przemyslaw Prusinkiewicz. A look at the visual modeling of plants using L-systems. *Agronomie*, 1999, 19 (3-4), pp.211-224. hal-00885925

**HAL Id: hal-00885925**

**<https://hal.science/hal-00885925>**

Submitted on 11 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A look at the visual modeling of plants using L-systems

Przemyslaw Prusinkiewicz

Department of Computer Science, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada

(Received 12 October 1998; accepted 10 February 1999)

**Abstract** – This paper presents a retrospective look at the history of plant modeling using L-systems, and a guide to the literature in the field. The topics include: distinctive features of the original L-system formalism, its subsequent extensions, visualization of L-system models, modeling of interactions between plants and their environment, the use of L-systems as a plant modeling language, and a survey of applications. (© Inra/Elsevier, Paris.)

**L-system / plant development / simulation model**

**Résumé** – Un aperçu de la modélisation visuelle des plantes à l'aide des L-systèmes. Cet article présente une rétrospective de la modélisation des plantes à l'aide des L-systèmes ainsi qu'un guide bibliographique pour ce domaine. Les sujets traités comprennent : les caractéristiques distinctives du formalisme du L-système original, ses développements ultérieurs, la modélisation des interactions entre les plantes et leur environnement, l'utilisation des L-systèmes comme langage pour modéliser les plantes et une vue d'ensemble des applications. (© Inra/Elsevier, Paris.)

**L-système / développement / plante / modèle de simulation**

## 1. Introduction

In 1968, Aristid Lindenmayer introduced a formalism for simulating the development of multicellular organisms, later named L-systems [57]. It was originally described in terms of linear or

branching chains of finite automata, but its subsequent reformulation in terms of rewriting systems [58] proved more elegant. The close relationship between L-systems, abstract automata and formal languages attracted the interest of computer scientists, who vigorously developed the mathematical theory of L-systems [40, 102, 104] (for an account

---

Communicated by Gérard Guyot (Avignon, France)

Correspondence and reprints  
E-mail: pwp@cpsc.ucalgary.ca

of the early history of L-systems, see also [65, 105]). This progress was followed by applications of L-systems to the modeling of plants, initiated by the development of the first simulation program based on L-systems called CELIA (an acronym for cellular linear iterative array simulator) by Baker and Herman [3, 4] and Herman and Liu [39].

In 1984, Smith [111] applied state-of-the-art computer graphics to visualize a class of abstract branching structures discovered by Hogeweg and Hesper [43]. The beauty of Smith's images and the life-like appearance of his developmental simulations inspired me to design and implement the simulation program called *pfg* (an acronym for plant and fractal generator, C code listing included in Prusinkiewicz and Hanan [89]). The first results obtained using *pfg* were focused on the visualization of fractals and abstract branching structures [82, 83]. Subsequently, *pfg* was applied to the realistic modeling of structures and processes found in real plants, in collaboration with Lindenmayer and Hanan [89, 95]. The results obtained by 1990 were collected in the book [94].

Many new results have been obtained since then. The purpose of the present paper is to survey current lines of research, and provide an updated guide to the literature on plant modeling using L-systems. For previous guides of a similar nature see [66] and [89, chapter 7]. General surveys of the modeling of plant architectures are presented in [16, 100]. For recent tutorial introductions to visual modeling using L-systems see [86, 87].

## 2. What are L-systems?

In order to retrace the advancement of modeling techniques based on L-systems, let us first identify the main features of the original formalism. From a mathematical point of view, L-systems are parallel rewriting systems, operating on strings of symbols that may represent individual cells [57] or larger components (modules) of a growing organism [24]. One of the simplest biologically relevant examples of L-systems is a model of the filamentous blue-green bacteria *Anabaena catenula* [60, 61, 63]. The

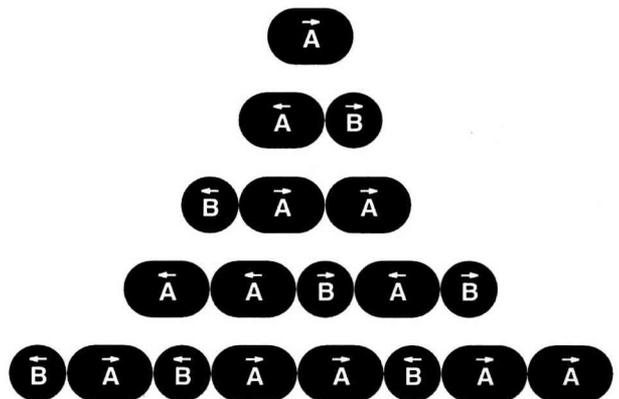
model describes the development of a so-called vegetative segment of *Anabaena* using rewriting rules (also called productions) operating on two types of cells, *A* and *B*. Each cell can have one of two different polarities indicated by superscript arrows:  $\vec{A}$ ,  $\overleftarrow{A}$ ,  $\vec{B}$  and  $\overleftarrow{B}$ . In given time intervals, cells *B* elongate and change their state to *A*, while cells *A* divide, producing a cell *A* and a cell *B*. This process is characterized by the following rules:

$$\vec{A} \rightarrow \vec{A} \vec{B} \quad \overleftarrow{A} \rightarrow \overleftarrow{B} \overleftarrow{A} \quad \vec{B} \rightarrow \vec{A} \quad \overleftarrow{B} \rightarrow \overleftarrow{A}$$

The L-system model integrates these local rules into a global description of a vegetative segment. The segment's development is simulated by an L-system derivation, in which productions are applied in parallel to all cells in the filament (*figure 1*). In spite of its simplicity, this model reveals several key features of the basic L-system formalism.

a) The model is inherently dynamic: it describes the development of a structure over time. In the words of d'Arcy Thompson, the form is viewed as "an event in space-time, and not merely a configuration in space" [115].

b) The model is discrete in three senses: "the state transformations are defined on discrete subunits (cells); each subunit may be present in one of a finite set of states; and the transformations are performed in discrete time steps" [64].



**Figure 1.** Visualization of the L-system model of *Anabaena catenula*. The model captures the arrangement of shorter and longer cells in a vegetative segment of the filament.

c) The cells are arranged in a linear filament. The original formalism also makes it possible to describe branching structures [57].

d) The model describes topology, in this case, the ordering of cells in the filament. There is no information regarding the geometry, that is the actual shape and size of the cells, and their position in space. In *figure 1*, it was arbitrarily decided that the cells would be represented as circles and rectangles with rounded corners, arranged along a straight line.

e) The topology of the organism changes as a result of cell division. No mechanism exists, however, to rearrange a set of existing cells. Consequently, L-systems are more suitable to model plants (in which cells are tightly cemented together) than animals (in which cells can move with respect to each other [112]).

f) The model represents the organism as a closed cybernetic system, which controls its development autonomously, without interacting with the environment.

g) The model describes global development of an organism in terms of local rules. In the *Anabaena* example, the state of each cell fully determines its fate in the next step. We say that these rules are context-free, and represent control of development by lineage [61]. The original L-system formalism also makes it possible to use context-sensitive rules, which capture interactions between adjacent elements of the developing structure.

Although these basic features are sufficient to create many models of linear and branching structures, advanced applications require extensions and modifications of L-systems. In the following sections we discuss three of them: the inclusion of continuous attributes, graphical interpretation of the models, and incorporation of external (environmental) influences on the development.

### 3. Continuous extension of L-systems

An essential component of the mathematical theory of L-systems is their discrete character [64].

Nevertheless, in applications of L-systems to modeling and simulation, this can become a limiting factor. The postulate that each module may assume only a finite number of states was the first to be reconsidered. As early as 1972, the simulation program CELIA allowed for the association of numerical attributes and sets of attributes of different types with L-system symbols [4, 39]. The idea of “adding continuous components to L-systems” was subsequently discussed by Lindenmayer [59]. An analysis of the error related to the discretization of continuous variables, such as the concentrations of substances in the cells, was given by Baker and Herman [4] (see also [40]).

Impetus for further development of L-systems with parameters stemmed from the requirements of model visualization. Parameters were needed to specify the lengths of lines and the magnitudes of branching angles in the models (section 4). Formal definitions of L-systems with parameters were given by Chien and Jürgensen [8, 9] and Hanan and Prusinkiewicz [34, 90]. A simulation program *cpfg* [81] (a continuous-parameter extension of *pfg*) was implemented by Hanan and subsequently enhanced by James (1993, unpublished data) and Hammel and Měch [86]. The use of parametric L-systems is the key advance in the modeling techniques presented in the book [94] over its predecessor [89].

The needs of model visualization, in particular for the animation of developmental processes, motivated another departure from the discrete characteristics of L-systems: the introduction of continuous time. The first formalism for specifying continuous-time processes using L-systems, called timed L-systems, was presented by Prusinkiewicz and Lindenmayer [94]. It was limited to context-free models, thus did not capture possible interactions between coexisting modules during development. This limitation was overcome in the next formalism, called differential L-systems [88]. This is a combined discrete-continuous model of development, in which modules are created and cease to exist in discrete events captured by productions, but develop in a continuous fashion described by differential equations. Arguments to these equations may be provided by the neighboring modules, thus an exchange of information between modules

can be expressed. Sample animations of plant development simulated using this formalism have been presented by Prusinkiewicz and Hammel [85].

One can contemplate whether the remaining discrete aspect of L-systems – partitioning the modeled system into discrete units – should also be relinquished in some applications. The resulting notion, which could be termed partial differential L-systems (Mjolsness, pers. comm.) would treat a developing organism as a continuous, possibly growing medium with a linear or branching topology. This approach was proposed by de Koster and Lindenmayer [12] to model a growing filament. Continuous media were also considered by Hammel and Prusinkiewicz [30, 31] in an L-system restatement of reaction-diffusion models for pattern formation in sea shells. These models were originally formulated by Meinhardt and Klingler [72, 73] in terms of partial differential equations. A formal definition of partial differential L-systems remains to be stated.

#### 4. Graphical interpretation of L-systems

The first algorithms for visualizing branching structures generated by L-systems were proposed in 1974 by Frijters and Lindenmayer [24] and Hogeweg and Hesper [43]. The geometric aspects of the modeled structures were defined using a set of drawing rules external to the L-systems under consideration, acting globally on all components of the modeled structure. This global definition made some structures impossible to specify. For example, the rule stating that branches should be issued in alternating directions, first to the left, then to the right, did not allow for the modeling of structures with two consecutive branches oriented the same way. In 1979, Szilard and Quinton observed that L-systems could be applied to generate a variety of intricate geometric patterns if graphical interpretation was associated with specific symbols in the generated strings [113]. According to one technique, the L-system symbols represented lines (vectors) running in predefined directions: left,

right, up and down. Thus, the strings defined the images according to the chain coding mechanism [20]. In another approach, directions were specified relative to the previous lines. Pursuing this latter route, Prusinkiewicz proposed [82] to consider L-system symbols as commands controlling a LOGO-style turtle [1]: move forward, turn to the left and turn to the right. L-systems with turtle interpretation made it possible to generate many fractal curves. Moreover, saving and restoring the turtle's position on a pushdown stack allowed the creation of plant-like structures with branches.

Several extensions to turtle interpretation were introduced by Hanan, Hammel, Měch and Prusinkiewicz. They included an extension of turtle interpretation to three dimensions [83], the possibility of incorporating predefined surfaces to represent organs such as leaves and flower petals [33, 83], and the addition of numerical parameters needed to control quantitative attributes of model components [34, 90]. Developmental surfaces [34] made it possible to simulate changes of organ shape in animations of plant development [85, 88]. Other methods for specifying the shape of plant organs included planar surfaces bound by sequences of turtle steps [33, 95], and implicit contours built around branching skeletal structures [32]. Generalized cylinders with various cross sections were recently incorporated into the framework of L-systems to model smoothly curving branches [70].

A further formalization of turtle interpretation was proposed by Kurth [54]. In particular, his work improved and systematized the method for manipulating the turtle's attributes using parametric L-systems.

The use of turtle interpretation is convenient in a biological context, because it makes it easy to express branching angles. However, absolute directions also play a significant role in plant development. In the words of Dawkins [11]: “the world usually imposes a significant difference between up and down”. For example, branches show a tendency to grow upwards, and roots to grow downwards, most of the time. Under the general term of tropisms, these phenomena have been captured by biasing turtle orientation in a predefined direction

[94, 95]. Further research is needed, however, to fully integrate tropisms with turtle interpretation. For instance, simple reorientation of the turtle towards or away from a predefined direction does not capture the phenomena of plagiotropism and leaf reorientation mediated by gravity and light. Furthermore, tropisms should be incorporated into the general framework for modeling the interactions between plants and environment, discussed next. This would make it possible to simulate a plant's responses to locally defined vector fields, such as the local direction of incoming light.

## 5. Incorporation of environmental factors

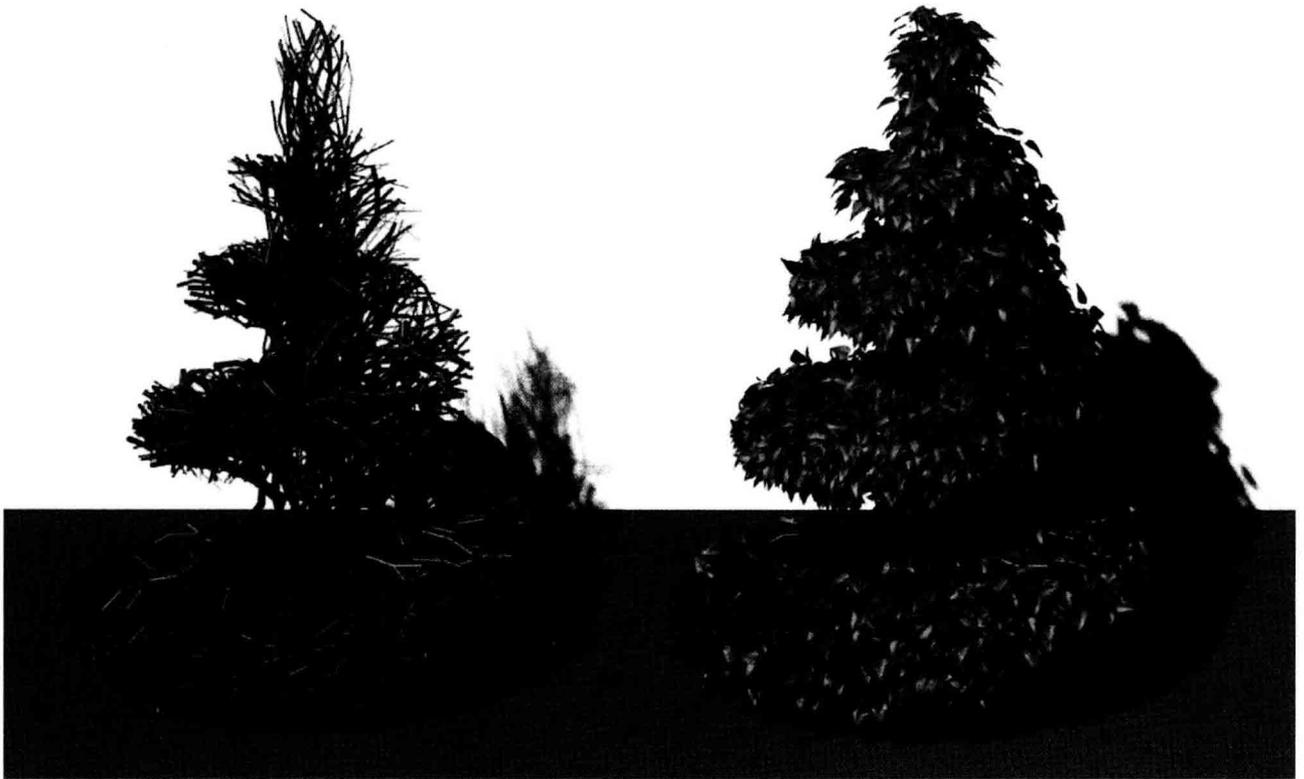
Plants modeled using the original formalism of L-systems were treated as closed cybernetic systems, developing without interaction with the environment. In reality, however, interaction with the environment plays a major role in the development of plants and plant communities, and cannot be neglected in practical models with predictive value. In the first step towards the inclusion of environmental factors, Rozenberg defined table L-systems, which allow changes to the production set from one derivation step to another [101] (see also [40, 104]). Table L-systems were applied, for example, to capture the switch from the production of leaves to the production of flowers by the apex of a flowering plant owing to a change in day length [21, 23, 24].

Table L-systems can only capture the impact of global environmental characteristics on plant development. Many phenomena depend, however, on local aspects of the environment. Prusinkiewicz et al. [92] introduced environmentally-sensitive L-systems to capture situations where the environment affects the plant, but the reciprocal influence of the plant on the environment can be ignored. This formalism was illustrated using examples of plant responses to pruning (*figure 2*). A related approach was applied by Fournier to model the effect of the local temperature of organs on the development of maize [17].

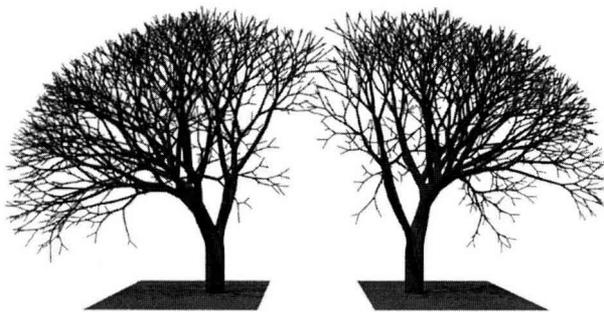
Plants may also interact with the environment in a feedback loop that includes information flow to and from the environment. Examples include competition for space between individual plants (ramets) in a clonal plant, competition for light between branches of a tree (where the upper branches change the amount of light available to the lower branches), and competition between roots for water in the soil. To express such phenomena, Měch and Prusinkiewicz introduced the formalism of open L-systems [70, 71] (*figure 3*). It extends the L-system alphabet with communication symbols, which can exchange parameter values with the environment. Thus, a model of a developmental process consists of two components: a plant model expressed using an L-system, and a program simulating the relevant aspects of the environment. A similar approach was devised by Fournier and Andrieu [18], who improved the model of maize [17] by including the effect of shading on the temperature of the apex.

A different organization of the modeling software was proposed by Kurth [53, 54], who incorporated predefined functions that return environmental information directly into the simulation program. This approach requires the simulator to be recompiled each time a new environmental function is added. A technique for incorporating environmental factors into L-systems was also presented by Vaario et al. [118]. An interesting aspect of this latter work is the merging of two fundamental models of morphogenesis: L-systems and diffusion-limited aggregation models [119]. All of these extensions require the location of different modules in 3D space to be known; thus, they have been formulated for L-systems with turtle interpretation, rather than L-systems in general.

A large amount of fundamental work on modeling developmental processes in an environmental context has been carried out outside the L-system framework, especially by Blaise [6] and Kaandorp [51]. Given the environmentally sensitive and open L-system extensions, a link between their work and L-systems can now be established (see section 9).



**Figure 2.** An L-system model of plants affected by pruning. From [92].



**Figure 3.** A model of deciduous trees competing for light. The trees have been moved apart after they grew to reveal adaptation of crown geometry to the presence of the neighbor tree. Leaves are not shown to expose the branching structures. From [74].

## 6. L-systems as programming languages

As Chomsky grammars are the foundation for many common programming languages, the formalism of L-systems is the basis on which programming languages for the modeling and simulation of plants have been and are being built. These languages offer the users of simulation programs the capability of expressing models easily, without the burden of constructing them ‘from scratch’ in a general-purpose language, and without the limitations of predefined, ‘hard-coded’ models, where only numerical parameter values can easily be changed. This advantage is essential in computer-assisted biological research, where a convenient mechanism for specifying, modifying and experimenting with all aspects of the models is particularly important.

To outline the gradual evolution of L-systems from theoretical concept to programming languages, let us consider the simulation programs *pfg* and *cpfg* (sections 1 and 3). The first version of these programs adopted a straightforward syntax, mimicking the mathematical notation for production specification [82, 89]. Modeling experience indicated that further constructs were needed to increase the expressive power and the flexibility of the language. The introduction of numerically valued parameters (section 3) was the first major improvement. This concept was extended with local variables (limited to individual productions) and global variables (shared between productions) [34, 91]. Further extensions included the incorporation of standard programming constructs such as predefined mathematical functions, arrays and flow control statements [70]. Sub-L-systems [34], based on the idea of subroutines, made it possible to partition complex models into a hierarchy of components, which can be defined independently.

Several extensions have been inspired, or can be related, to the notions of L-system theory. Programming constructs for stochastic L-systems [83, 94] closely follow their formal definitions [15, 76, 120]. The 'cut symbol', introduced to simulate the shedding of organs such as leaves, petals or entire branches [34, 94], has its counterpart in L-systems with fragmentation [103, 106]. The idea of fragmentation is also related to the modeling of collections of objects, such as clonal plants that become separated during vegetative propagation [87]. L-systems with homomorphisms [70] (termed two-phase growth grammars by Kurth [54]) make use of the homomorphic transformations of generated strings [75, 104] to separate the logic of the models from the details of their graphical interpretation. This separation makes complex models clearer and better structured.

Extensions analogous to those outlined above using the example of *pfg* and *cpfg* can also be found in other implementations of languages based on L-systems. A syntactic variant of the parameter-passing mechanism for parametric context-free L-systems was proposed by Borovikov [7] and implemented in the commercial program World Builder [2]. It makes it possible to handle modules with

large numbers of parameters in a concise manner, because parameters that are not affected by a particular production do not have to be explicitly listed. Constructs borrowed from standard programming languages have been incorporated in the modeling systems ELSYS by Goel and Rozehnal [28], and GROGRA by Kurth [54]. An idea similar to fragmentation has been applied to the animation of multiple interacting objects by Noser and Thalmann [77, 78]. One of their examples, particularly interesting from the biological perspective, is the simulation of butterflies flying in a flower field.

Unfortunately, in spite of the practical importance of programming languages based on L-systems, they have not yet been systematically studied from the perspective of programming language theory. A notable exception is the work by Oritz et al. [79] and Pinter and Pinter [80], where L-systems are considered in the context of programming massively parallel computers such as the Connection Machine.

Plant modeling can also be regarded in the general framework of simulation theory. This point of view was first adopted by Hogeweg, who considered L-system derivations as discrete-event simulations, and used SIMULA to implement the models [41, 42]. Pursuing a similar approach, Hammel [30] applied a combined discrete-continuous simulation extension of SIMULA called DISCO [38] to implement differential L-systems (cf. section 3). Both implementations made it possible to relinquish the assumption of a strictly synchronous operation of L-systems, which may be unrealistic from a biological perspective. In addition, they took advantage of the object-oriented programming environment provided by SIMULA to conveniently express the models. Object-oriented extensions of L-systems have also been proposed using the framework of C++ by Borovikov [2, 7] and Guzy (1995, unpublished). The development of programming languages and environments based on L-systems is an active research domain. As the understanding of modeling using L-systems grows, we may expect that new, more systematically designed languages will emerge. They will take full advantage of L-system theory, and combine useful con-

structs found at present in separate languages and programs.

## 7. Applications to plant modeling

In general, plant models can be divided into mechanistic (causal) and empirical (descriptive). The purpose of mechanistic models is to gain an understanding of plant development in terms of the interactions between the component modules and processes. Thus, “mechanistic modeling follows the traditional reductionist method that has been so very successful in the physical sciences” [116]. In contrast, empirical models reproduce the morphology of the described plants without re-enacting the control mechanisms. We will discuss the mechanistic models first.

L-systems were introduced as a formalism for modeling and simulating the development of simple multicellular organisms, such as filamentous bacteria and algae [57]. In this spirit, Tunbridge and Jones recently applied a context-sensitive parametric L-system to model the development of the fungus *Aspergillus nidulans* [117]. In 1974, Frijters and Lindenmayer proposed L-systems for modeling the structures found in higher plants, in particular compound inflorescences [21–24]. From a biological perspective, the main purpose of their studies was to present plausible explanations of flowering sequences and differences in relative branch sizes in the studied plants. Frijters and Lindenmayer observed [25] that simple control of development by lineage, expressed by non-parametric context-free L-systems, could not capture basipetal flowering sequences (with the flowering zone progressing from the top of the plant downwards) and acrotonic patterns of branch development (with the largest branches situated near the top of the plant). A formal analysis supporting this observation was presented later by Lück et al. [67] and Prusinkiewicz and Kari [93]. Overcoming this limitation, Janssen and Lindenmayer [50] and Lindenmayer showed [62] that acrotonic flowering patterns and basipetal flowering sequences could be reproduced assuming control of development by

hormones that flow through the developing structure and trigger developmental events. This work formed the basis for subsequent realistic modeling and visualization of herbaceous plants [89, 94, 95].

The ease of describing interactive control mechanisms using context-sensitive L-systems is one of the most appealing features of the L-system formalism. In addition to the work cited above, examples include models of trees affected by pruning, in which signals initiate the development of dormant lateral buds after the apices of the main branches have been removed (section 5). In a less typical application, inspired by Room (pers. comm.), a signal represents an insect that feeds on a plant [86, 87]. The incorporation of parameters makes it possible to quantify concentrations of substances flowing in a growing structure, such as water, minerals or products of photosynthesis. A simple example of a plant model including the flow and partitioning of resources between the shoot and the root is presented by Prusinkiewicz et al. [86, 87]. The flow of resources (products of photosynthesis) is also a part of the models of trees competing for light proposed by Takenaka [114] and reproduced using L-systems by Měch [70] and Měch and Prusinkiewicz [71]. These capabilities, combined with the mechanisms for communicating with the environment (section 5), make it possible to apply L-systems to express functional–structural models [110], which incorporate physiological aspects into three-dimensional plant models.

L-systems have also been used to construct descriptive developmental models. A significant body of work devoted to modeling algae has been carried out by Corbit and Garbary [10, 26], and Morelli et al. [74], Schneider and Walde [107] and Schneider et al. [108]. The latter group suggested an interesting if hypothetical link between branching architectures, their L-system models and the genetic makeup of the studied species. It is exemplified by the following statement: “If the character strings necessary to code our graphical images corresponded in some way to the information represented in the genetic code of *Dipterosiphonia* species, then only minor genetic changes would be necessary to account for speciation where branch-

ing pattern was the main defining species characteristic" [108].

In contrast to the relatively simple models of algae, empirical models of higher plants rely on large amounts of quantitative data and statistical analysis of plant morphology. Examples include models of the Japanese cypress scale leaves [76], green ash shoots [96], young green ash trees [97], Norway spruce trees [56], cotton plants [98], bean plants [14, 35], maize shoots [17, 18] and maize root systems [109]. A novel use of L-systems has been proposed by Battjes and Bachmann (1996, unpublished), who related parameter values in the L-system models to genetic variation between modeled plants (four species of *Microseris*, a herbaceous plant in the aster family).

The large amount of observational data needed to construct models raises a number of practical problems. What features of plant morphology should be measured? What devices should be used to perform the measurements? How should the acquired data be represented in a database for easy access, processing and incorporation into the final models? An overview of these problems in the context of L-system modeling has been presented by Remphrey and Prusinkiewicz [97]; a detailed example of empirical model construction has been given by Prusinkiewicz et al. [96]. The process of data acquisition has been presented in detail by Hanan and Room [37], and emphasized in the article on the applications of L-systems to plant modeling Room et al. [99]. The underlying digitizing software is available through the Internet [36]. Theoretical aspects of model construction according to quantitative data have been described by Godin et al. [27]. Although this paper is not presented in the framework of L-systems, the results can be easily adapted.

## 8. L-systems and evolution

L-system models can be subject to a cyclic process of artificial evolution, in which changes to the rules are introduced, the resulting models are evaluated, and the L-systems producing the best

models (for a given criterion) are selected for the next iteration of changes and evaluation. An early pursuit of this concept was presented by MacKenzie [68] and MacKenzie and Prusinkiewicz [69]. The key idea was to apply genetic algorithms (for example, see [29]) to introduce variations in the class of L-systems being explored. Experiments included several selection mechanisms, such as the fractal dimension and the amount of light captured by the resulting structures. Recently, the concept of evolving L-system models has been extensively studied by Jacob [44–49]. Evolving models of abstract structures, motivated by plants, were also proposed by Kim [52]. All of these models operate at a highly abstract level, and the 'genetic operations' used to modify L-systems have so far been removed from the actual plant genetics.

## 9. Concluding remarks

Advancements in the modeling of plants using L-systems have been largely motivated by the desire to expand the range of phenomena that can be formally described, simulated and studied. Now that this range is quite extensive, questions regarding the relationship between L-systems and other models of plant architecture emerge. It appears that the use of a special-purpose modeling language is the most distinctive feature of the L-system-based approach. It makes it easy to specify models as an input to general-purpose simulation programs or as a part of a model description in publications. In contrast, models expressed in general-purpose programming languages may require multi-page program listings. The essence of the models, however, is often similar in spite of different software implementations. Indeed, the modeling power of L-systems has been repetitively evaluated by re-implementing various models constructed originally within other frameworks [70, 71, 87, 94]. In this context, Françon [19] and Kurth [53, 55] observed an interesting convergence between models expressed using L-systems and the large body of models developed at the Atelier de modélisation de l'architecture des plantes AMAP, CIRAD,

Montpellier, France (for a recent description of the work at AMAP see [5, 13]). A further comparison of different approaches to the modeling of plant architecture would be an interesting research project in itself.

**Acknowledgements:** The kind invitation from Bruno Andrieu to the Architectural Modeling seminar in Paris prompted me to update an earlier version of this survey [84]. Many of the papers included in this review were sent to me directly by the authors, or brought to my attention by my colleagues. I would like to express my gratitude for this input, without which important references may have been missed. I would also like to thank Jim Hanan, Hugh McEvoy, and Lynn Mercer for helpful comments on the manuscript. This work has been supported in part by research and equipment grants from the Natural Sciences and Engineering Research Council of Canada, and by the Killam Resident Fellowship held at the University of Calgary in the Fall of 1996. This support is gratefully acknowledged.

## References

- [1] Abelson H., di Sessa A.A., Turtle geometry, M.I.T. Press, Cambridge, 1982.
- [2] AnimaTek International, Inc., AnimatTek's World Builder 2.0 Yosemite Edition, 1997.
- [3] Baker R., Herman G.T., CELIA – a cellular linear iterative array simulator, Proceedings of the Fourth Conference on Applications of Simulation (9–11 December 1970), 1970, pp. 64–73.
- [4] Baker R., Herman G.T., Simulation of organisms using a developmental model, parts I and II, Int. J. Bio-Med. Comput. 3 (1972) 201–215 and 251–267.
- [5] Barthélémy D., Blaise F., Fourcaud T., Nicolini E., Modélisation et simulation de l'architecture des arbres: Bilan et perspectives, Rev. For. Fr. XLVII (1995) 71–96.
- [6] Blaise F., Simulation du parallélisme dans la croissance des plantes et applications, Ph.D. thesis, Université Louis Pasteur, Strasbourg, 1991.
- [7] Borovikov A., L-systems with inheritance: an object-oriented extension of L-systems, ACM SIG-PLAN Notices 30 (1995) 43–60.
- [8] Chien T.W., Graphical interpretation of the biological development modelled by VDOL-systems, Master's thesis, University of Western Ontario, 1989.
- [9] Chien T.W., Jürgensen H., Parameterized L systems for modelling: Potential and limitations, in: Rozenberg G., Salomaa A. (Eds.), Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology, Springer-Verlag, Berlin, 1992, pp. 213–229.
- [10] Corbit J.D., Garbary D.J., Computer simulation of the morphology and development of several species of seaweed using Lindenmayer systems, Comput. Graphics 17 (1993) 85–88.
- [11] Dawkins R., Climbing Mount Improbable, W. Norton and Co., New York, 1996.
- [12] de Koster C.G., Lindenmayer A., Discrete and continuous models for heterocyst differentiation in growing filaments of blue-green bacteria, Acta Biotheor. 36 (1987) 249–273.
- [13] de Reffye P., Houllier F., Blaise F., Barthélémy D., Dauzat J., Auclair D., A model simulating above- and below-ground tree architecture with agroforestry applications, Agrofor. Systems 30 (1995) 175–197.
- [14] Diaz-Ambrona C.H., Tarquis A.M., Minguez M.I., Faba bean canopy modelling with a parametric open L-system: a comparison with the Monsi and Saeki model, Field Crops Res. 3680 (1998) 1–13.
- [15] Eichhorst P., Savitch W.J., Growth functions of stochastic Lindenmayer systems, Inf. Control 45 (1980) 217–228.
- [16] Fisher J.B., How predictive are computer simulations of tree architecture, Int. J. Plant Sci. 153 (suppl.) (1992) 137–146.
- [17] Fournier C., Introduction des réponses écophysiologiques à la température dans un modèle de plante à la base de L-Systèmes, Master's thesis, Institut National Agronomique Paris-Grignon, 1995.
- [18] Fournier C., Andrieu B., A 3D architectural and process-based model of maize development, Ann. Bot. 81 (1998) 233–250.
- [19] Françon J., Sur la modélisation de l'architecture et du développement des végétaux, in: Edelin C. (Ed.), L'Arbre. Biologie et Développement, Naturalia Monspeliensia, 1991, No hors série,
- [20] Freeman H., On encoding arbitrary geometric configurations, IRE Trans. Electronic. Comput. 10 (1961) 260–268.
- [21] Frijters D., An automata-theoretical model of the vegetative and flowering development of *Hieracium murorum* L., Biol. Cybern. 24 (1976) 1–13.

- [22] Frijters D., Mechanisms of developmental integration of *Aster novae-angliae* L. and *Hieracium muro-rum* L., *Ann. Bot.* 42 (1978) 561–575.
- [23] Frijters D., Principles of simulation of inflorescence development, *Ann. Bot.* 42 (1978) 549–560.
- [24] Frijters D., Lindenmayer A., A model for the growth and flowering of *Aster novae-angliae* on the basis of table (1,0) L-systems, in Rozenberg G., Salomaa A. (Eds.), *L Systems, Lecture Notes in Computer Science 15*, Springer-Verlag, Berlin, 1974, pp. 24–52.
- [25] Frijters D., Lindenmayer A., Developmental descriptions of branching patterns with paracladial relationships, in: Lindenmayer A., Rozenberg G. (Eds.), *Automata, Languages, Development*, North-Holland, Amsterdam, 1976, pp. 57–73.
- [26] Garbary D.J., Corbit J.D., Lindenmayer systems as models of red algal morphology and development, *Progr. Phycol. Res.* 8 (1992) 143–177.
- [27] Godin C., Guédon Y., Costes E., Caraglio Y., Measuring and analysing plants with the AMAPmod software, in: Michalewicz M.T. (Ed.), *Plants to Ecosystems: Advances in Computational Life Sciences I*, CSIRO Publishing, Melbourne, 1997, pp. 53–84.
- [28] Goel N., Rozenhal I., A high-level language for L-systems and its applications, in: Rozenberg G., Salomaa A. (Eds.), *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Springer-Verlag, Berlin, 1992, pp. 231–251.
- [29] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [30] Hammel M., *Differential L-systems and their application to the simulation and visualization of plant development*, Ph.D. thesis, University of Calgary, 1996.
- [31] Hammel M., Prusinkiewicz P., Visualization of developmental processes by extrusion in space-time, *Proceedings of Graphics Interface '96*, 1996, pp. 246–258.
- [32] Hammel M., Prusinkiewicz P., Wyvill B., Modelling compound leaves using implicit contours, in: Kunii T.L. (Ed.), *Visual Computing – Integrating Computer Graphics with Computer Vision*, Springer Verlag, Tokyo, 1992, pp. 199–212.
- [33] Hanan J.S., *PLANTWORKS: A software system for realistic plant modelling*, Master's thesis, University of Regina, November 1988.
- [34] Hanan J.S., *Parametric L-systems and their application to the modelling and visualization of plants*, Ph.D. thesis, University of Regina, 1992.
- [35] Hanan J.S., Virtual plants – Integrating architectural and physiological models, *Environ. Model. Software* 12 (1997) 35–42.
- [36] Hanan J.S., Room P.M., *Virtual plants: A hyper-text document and digitizing software distribution*, Cooperative Research Centre for Tropical Pest Management, Brisbane, Australia, 1996; available at: <http://www.ctpm.uq.edu.au/Programs/IPI/ipivp.html>.
- [37] Hanan J.S., Room P.M., Practical aspects of virtual plant research, in: Michalewicz M.T. (Ed.), *Plants to Ecosystems: Advances in Computational Life Sciences I*, CSIRO Publishing, Melbourne, 1997, pp. 28–44.
- [38] Helsingaun K., DISCO – A SIMULA-based language for combined continuous and discrete simulation, *Simulation* 35 (1980) 1–12.
- [39] Herman G.T., Liu W.H., The daughter of CELIA, the French flag, and the firing squad, *Simulation* 21 (1973) 33–41.
- [40] Herman G.T., Rozenberg G., *Developmental systems and languages*, North-Holland, Amsterdam, 1975.
- [41] Hogeweg P., Simulating the growth of cellular forms, *Simulation* September (1978) 90–96.
- [42] Hogeweg P., Locally synchronized developmental systems: Conceptual advantages of discrete event formalism, *Int. J. Gen. Syst.* 6 (1980) 57–73.
- [43] Hogeweg P., Hesper B., A model study on biomorphological description, *Pattern Recognit.* 6 (1974) 165–179.
- [44] Jacob C., Genetic L-system programming, *Parallel Problem Solving from Nature PPSN III*, Lecture Notes in Computer Science 866, Springer-Verlag, Berlin, 1994, pp. 334–343.
- [45] Jacob C., Genetic L-system programming: Breeding and evolving artificial flowers with Mathematica, *Proceeding of the First International Mathematica Symposium '95*, Southampton, UK, 1995, pp. 215–222.
- [46] Jacob C., *MathEvolvica: Simulierte Evolution von Entwicklungsprogrammen der Natur*, Ph.D. thesis, University of Erlangen, 1995.
- [47] Jacob C., Evolution programs evolved, *Parallel Problem Solving from Nature PPSN-IV*, Lecture Notes in Computer Science 1141, Springer-Verlag, Berlin, 1996, pp. 42–51.

- [48] Jacob C., Evolving evolution programs: Genetic programming and L-systems, Proceedings of the First Annual Conference on Genetic Programming, M.I.T. Press, Cambridge, 1996, pp. 107–115.
- [49] Jacob C., *Principia Evolvica*, Simulierte Evolution mit Mathematica, dpunkt-Verlag, Heidelberg, 1997.
- [50] Janssen J.M., Lindenmayer A., Models for the control of branch positions and flowering sequences of capitula in *Mycelis muralis* (L) Dumont (Compositae), *New Phytol.* 105 (1987) 191–220.
- [51] Kaandorp J., *Fractal Modelling: Growth and Form in Biology*, Springer-Verlag, Berlin, 1994.
- [52] Kim J., Untersuchungen zur Evolution von morphologischer und taxonomischer Diversität und Komplexität anhand von Computermodellen. Ph.D. thesis, University of Cologne, 1996.
- [53] Kurth W., Stochastic sensitive growth grammars: A basis for morphological models of tree growth, *Nat. Monspel.*, in press.
- [54] Kurth W., Growth grammar interpreter GROGRA 2.4: A software tool for the 3-dimensional interpretation of stochastic, sensitive growth grammars in the context of plant modeling, Introduction and reference manual, Forschungszentrum Waldökosysteme der Universität Göttingen, Göttingen, 1994.
- [55] Kurth W., Morphological models of plant growth: Possibilities and ecological relevance, *Ecol. Model.* 75/76 (1994) 299–308.
- [56] Kurth W., Lanwert D., Biometrische Grundlagen für ein dynamisches Architekturmodell der Fichte (*Picea abies* (L.) Karst.), *Allg. Forst und Jagdztg.* 166 (1995) 177–184.
- [57] Lindenmayer A., Mathematical models for cellular interaction in development, Parts I and II, *J. Theor. Biol.* 18 (1968) 280–315.
- [58] Lindenmayer A., Developmental systems without cellular interaction, their languages and grammars, *J. Theor. Biol.* 30 (1971) 455–484.
- [59] Lindenmayer A., Adding continuous components to L-systems, in: Rozenberg G., Salomaa A. (Eds.), *L Systems*, Lecture Notes in Computer Science 15, Springer-Verlag, Berlin, 1974, pp. 53–68.
- [60] Lindenmayer A., Algorithms for plant morphogenesis, in: Sattler R. (Ed.), *Theoretical Plant Morphology*, Leiden University Press, The Hague, 1978, pp. 37–81.
- [61] Lindenmayer A., Developmental algorithms: Lineage versus interactive control mechanisms, in: Subtelny S., Green P.B. (Eds.), *Developmental Order: Its Origin and Regulation*, Alan R. Liss, New York, 1982, pp. 219–245.
- [62] Lindenmayer A., Positional and temporal control mechanisms in inflorescence development, in: Barlow P.W., Carr D.J. (Eds.), *Positional Controls in Plant Development*, University Press, Cambridge, 1984.
- [63] Lindenmayer A., Models for multicellular development: Characterization, inference and complexity of L-systems, in: Kelemenová A., Kelemen J. (Eds.), *Trends, Techniques and Problems in Theoretical Computer Science*, Lecture Notes in Computer Science 281, Springer-Verlag, Berlin, 1987, pp. 138–168.
- [64] Lindenmayer A., Jürgensen H., Grammars of development: Discrete-state models for growth, differentiation and gene expression in modular organisms, in: Rozenberg G., Salomaa A. (Eds.), *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Springer-Verlag, Berlin, 1992, pp. 3–21.
- [65] Lindenmayer A., Kelemen J., Kelemenová A., Answers by A. Lindenmayer to questions of J. Kelemen and A. Kelemenová, *Bull. Eur. Assoc. Theor. Comput. Sci.* 23 (1984) 185–198.
- [66] Lindenmayer A., Prusinkiewicz P., An annotated bibliography of plant modeling and growth simulation, in: Langton C.G. (Ed.), *Artificial Life*, Addison-Wesley, Redwood City, 1988, pp. 625–643.
- [67] Lück J., Lück H.B., Bakkali M., A comprehensive model for acrotonic, mesotonic, and basitonic branching in plants, *Acta Biotheor.* 38 (1990) 257–288.
- [68] MacKenzie C., Artificial evolution of generative models in computer graphics, Master's thesis, University of Calgary, 1993.
- [69] MacKenzie C., Prusinkiewicz P., Artificial evolution of plant forms, Proceedings of the Fifth Annual Western Computer Graphics Symposium, Vernon, BC, 28–30 March, 1993, 9 pp.
- [70] Měch R., Modeling and simulation of the interactions of plants with the environment using L-systems and their extensions, Ph.D. thesis, University of Calgary, 1997.
- [71] Měch R., Prusinkiewicz P., Visual models of plants interacting with their environment, Proceedings of SIGGRAPH '96 (New Orleans, Louisiana, 4–9 August 1996), ACM SIGGRAPH, New York, 1996, pp. 397–410.
- [72] Meinhardt H., *The algorithmic beauty of sea shells*, Springer-Verlag, Berlin, 1995.

- [73] Meinhardt H., Klinger M., A model for pattern formation on the shells of molluscs, *J. Theor. Biol.* 126 (1987) 63–89.
- [74] Morelli R.A., Walde R.E., Akstin E., Schneider C.W., L-system representation of speciation in the red algal genus *Dipterosiphonia* (Ceramiales, Rhodomelaceae), *J. Theor. Biol.* 149 (1991) 453–465.
- [75] Nielsen M., Rozenberg G., Salomaa A., Skyum S., Nonterminals, homomorphisms and codings in different variations of OL-systems. Part I: Deterministic systems, *Acta Inform.* 4 (1974) 87–106.
- [76] Nishida T., KOL-systems simulating almost but not exactly the same development – the case of Japanese cypress, *Mem. Fac. Sci., Kyoto Univ., Ser. Bio.* 8 (1980) 97–122.
- [77] Noser H., Thalmann D., Simulating life of virtual plants, fishes and butterflies, in: Magnenat Thalmann N., Thalmann D. (Eds.), *Artificial Life and Virtual Reality*, J. Wiley & Sons, Chichester, 1994.
- [78] Noser H., Thalmann D., The animation of autonomous actors based on production rules, *Proceedings of Computer Animation '96*, IEEE Computer Society Press, Los Alamitos, 1996, pp. 47–57.
- [79] Ortiz L.F., Pinter R.Y., Pinter S.S., An array language for data parallelism: Definition, compilation, and applications, *J. Supercomput.* 5 (1991) 7–29.
- [80] Pinter R.Y., Pinter S.S., Efficient breadth-first expansion on the Connection Machine, or: Parallel processing of L-systems, Technical Report YALEU/DCS/TR-719, Department of Computer Science, Yale University, 1989.
- [81] Prusinkiewicz P., Virtual plant laboratory, A hypertext document and software distribution, Department of Computer Science, University of Calgary, 1996; available at: <http://www.cpsc.ucalgary.ca/projects/bmv/vlab/index.html>.
- [82] Prusinkiewicz P., Graphical applications of L-systems, *Proceedings of Graphics Interface '86 – Vision Interface '86*, 1986, pp. 247–253.
- [83] Prusinkiewicz P., Applications of L-systems to computer imagery, in: Ehrig H., Nagl M., Rosenfeld A., Rozenberg G. (Eds.), *Graph Grammars and their Application to Computer Science*, Third International Workshop, Lecture Notes in Computer Science 291, Springer-Verlag, Berlin, 1987, pp. 534–548.
- [84] Prusinkiewicz P., A look at the visual modeling of plants using L-systems, in: Hofestädt R., Lengauer T., Löffler M., Schomburg D. (Eds.), *Bioinformatics*, Lecture Notes in Computer Science 1278, Springer-Verlag, Berlin, 1997, pp. 11–29.
- [85] Prusinkiewicz P., Hammel M., Visual models of morphogenesis: A guided tour. Hypertext document, Department of Computer Science, University of Calgary, 1994. Available at: <http://www.cpsc.ucalgary.ca/projects/bmv/vmm/intro.html>.
- [86] Prusinkiewicz P., Hammel M., Hanan J., Měch R., L-systems: from the theory to visual models of plants, in: Michalewicz M.T. (Ed.), *Plants to Ecosystems: Advances in Computational Life Sciences I*, CSIRO Publishing, Melbourne, 1997, pp. 1–27.
- [87] Prusinkiewicz P., Hammel M., Hanan J., Měch R., Visual models of plant development, in: Rozenberg G., Salomaa A. (Eds.), *Handbook of Formal Languages*, Vol. III: Beyond Words, Springer, Berlin, 1997, pp. 535–597.
- [88] Prusinkiewicz P., Hammel M., Mjolsness E., Animation of plant development, *Proceedings of SIGGRAPH 93* (Anaheim, California, 1–6 August 1993), ACM SIGGRAPH, New York, 1993, pp. 351–360.
- [89] Prusinkiewicz P., Hanan J., Lindenmayer Systems, Fractals, and Plants, *Lecture Notes in Biomathematics* 79, Springer-Verlag, Berlin, 1989.
- [90] Prusinkiewicz P., Hanan J., Visualization of botanical structures and processes using parametric L-systems, in: Thalmann D., (Ed.), *Scientific Visualization and Graphics Simulation*, J. Wiley & Sons, Chichester, 1990, pp. 183–201.
- [91] Prusinkiewicz P., Hanan J., L-systems: From formalism to programming languages, in: Rozenberg G., Salomaa A. (Eds.), *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Springer-Verlag, Berlin, 1992, pp. 193–211.
- [92] Prusinkiewicz P., James M., Měch R., Synthetic topiary, *Proceedings of SIGGRAPH '94* (Orlando, Florida, 24–29 July 1994), ACM SIGGRAPH, New York, 1994, pp. 351–358.
- [93] Prusinkiewicz P., Kari L., Subapical bracketed L-systems, in: Cuny J., Ehrig H., Engels G., Rozenberg G. (Eds.), *Graph Grammars and their Application to Computer Science*, Fifth International Workshop, Lecture Notes in Computer Science 1073, Springer-Verlag, Berlin, 1996, pp. 550–564.
- [94] Prusinkiewicz P., Lindenmayer A., with Hanan J.S., Fracchia F.D., Fowler D.R., de Boer M.J.M., Mercer L., *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990.

- [95] Prusinkiewicz P., Lindenmayer A., Hanan J., Developmental models of herbaceous plants for computer imagery purposes, Proceedings of SIGGRAPH '88 (Atlanta, Georgia, 1–5 August 1988), ACM SIGGRAPH, New York, 1988, in: Computer Graphics 22 (1988) 141–150.
- [96] Prusinkiewicz P., Remphrey W., Davidson C., Hammel M., Modeling the architecture of expanding *Fraxinus pennsylvanica* shoots using L-systems, Can. J. Bot. 72 (1994) 701–714.
- [97] Remphrey W., Prusinkiewicz P., Quantification and modelling of tree architecture, in: Michalewicz M.T. (Ed.), Plants to Ecosystems: Advances in Computational Life Sciences I, CSIRO Publishing, Melbourne, 1997, pp. 45–52.
- [98] Room P.M., Hanan J.S., Virtual cotton: taking plant architecture into account, The Australian Cottongrower 18 (1997) 32–36.
- [99] Room P.M., Hanan J.S., Prusinkiewicz P., Virtual plants: new perspectives for ecologists, pathologists, and agricultural scientists, Trends Plant Sci. 1 (1996) 33–38.
- [100] Room P.M., Maillette L., Hanan J., Module and metamer dynamics and virtual plants, Adv. Ecol. Res. 25 (1994) 105–157.
- [101] Rozenberg G., TOL systems and languages, Inf. Control 23 (1973) 357–381.
- [102] Rozenberg G., Penttonen M., Salomaa A., Bibliography of L systems, Theor. Comput. Sci. 5 (1977) 339–354.
- [103] Rozenberg G., Ruohonen K., Salomaa A., Developmental systems with fragmentation, Int. J. Comput. Math. 5 (1976) 177–191.
- [104] Rozenberg G., Salomaa A., The Mathematical Theory of L Systems, Academic Press, New York, 1980.
- [105] Rozenberg G., Salomaa A., When L was young, in: Rozenberg G., Salomaa A. (Eds.), The Book of L, Springer-Verlag, Berlin, 1986, pp. 383–392.
- [106] Ruohonen K., Developmental systems with interaction and fragmentation, Inf. Control 28 (1975) 91–112.
- [107] Schneider W., Walde R.E., L-system computer simulations of branching divergence in some dorsiventral members of the tribe Polysiphoniaeae (Rhodomelaceae, Rhodophyta), Phycologia 31 (1992) 6, 581–590.
- [108] Schneider W., Walde R.E., Morelli R.A., L-system computer models generating distichous from spiral organization in the Dasyaceae (Ceramiaceae, Rhodophyta), Eur. J. Phycol. 29 (1994) 165–170.
- [109] Shibusawa, Modelling the branching growth fractal pattern of the maize root system, Plant and Soil 165 (1994) 339–347.
- [110] Sievänen R., Mäkelä A., Nikinmaa E., Korpilahti E., Preface, Silva Fennica 31 (1997) 3, 237–238, special issue on functional-structural tree models.
- [111] Smith A.R., Plants, fractals, and formal languages, Proceedings of SIGGRAPH '84 (Minneapolis, Minnesota, 22–27 July 1984), ACM SIGGRAPH, New York, 1984, in Computer Graphics 18 (1984) 1–10.
- [112] Steeves T.A., Sussex I.M., Patterns in Plant Development, Cambridge University Press, Cambridge, 1989.
- [113] Szilard A.L., Quinon R.E., An interpretation for DOL systems by computer graphics, The Science Terrapin 4 (1979) 8–13.
- [114] Takenaka A., A simulation model of tree architecture development based on growth response to local light environment, J. Plant Res. 107 (1994) 321–330.
- [115] d'Arcy Thompson, On Growth and Form, University Press, Cambridge, 1952.
- [116] Thornley J.H.M., Johnson I.R., Plant and Crop Modeling: A Mathematical Approach to Plant and Crop Physiology, Oxford University Press, New York, 1990.
- [117] Tunbridge A., Jones H., An L-systems approach to the modelling of fungal growth, J. Visualization Comput. Animat. 6 (1995) 91–107.
- [118] Vaario J., Ogata N., Shimohara K., Synthesis of environment directed and genetic growth. ALIFE V oral presentations (Nara, Japan, 16–18 May 1996), pp. 207–214.
- [119] Witten T.A., Sander L.M., Diffusion-limited aggregation, a kinetic critical phenomenon, Phys. Rev. Lett. 47 (1981) 1400–1403.
- [120] Yokomori T., Stochastic characterizations of EOL languages, Information and Control 45 (1980) 26–33.