Simulation Modeling of Growing Tissues

Colin Smith and Przemyslaw Prusinkiewicz
University of Calgary

Introduction

Growing biological systems can be described as *dynamical systems with a dynamical structure*. In such systems not only the values of variables characterizing system components, but also the number of components and the connections between them, may change over time. For example, in a developing plant, the numbers of branches, leaves and flowers change as the plant grows, and the topology and form of the plant are gradually modified. A simulation model of a plant must therefore be capable of dealing with the changing numbers of components, and their varying configurations. This problem was explicitly addressed by Lindenmayer, who introduced L−systems as a formalism for modeling structures with a dynamically changing topology. This formalism, extended with a geometric interpretation, has subsequently been applied to model a wide variety of plants.

...

The modeling of developing surface structures has recently been addressed from new viewpoints by Giavitto and Michel (the MGS system), and Smith et al. (the vv system). Here we outline this latter approach and its potential applications to the modeling of biological structures.

Reference

Colin Smith and Przemyslaw Prusinkiewicz. Simulation Modeling of Growing Tissues. In *Proceedings of the 4th International Workshop on Functional−Structural Plant Models*, pp. 365−370.

Simulation Modeling of Growing Tissues

Colin Smith and Przemyslaw Prusinkiewicz
University of Calgary

1 Introduction

Growing biological systems can be described as *dynamical systems with a dynamical structure* [8]. In such systems not only the values of variables characterizing system components, but also the number of components and the connections between them, may change over time. For example, in a developing plant, the numbers of branches, leaves and flowers change as the plant grows, and the topology and form of the plant are gradually modified. A simulation model of a plant must therefore be capable of dealing with the changing numbers of components, and their varying configurations. This problem was explicitly addressed by Lindenmayer, who introduced *L-systems* as a formalism for modeling structures with a dynamically changing topology [13]. This formalism, extended with a geometric interpretation, has subsequently been applied to model a wide variety of plants [16].

Unfortunately, the original formalism of L-systems only captures linear and branching structures. To overcome this limitation, Lindenmayer and Rozenberg introduced *map L-systems*, which operate on a class of graphs with cycles called *maps* [14]. This seminal work led to several variants and extensions of map L-systems, which were reviewed in [17, Chapter 7]. In order to visualize the modeled structures, Siero *et al.* introduced a *geometric interpretation* of map L-systems [18]. An alternative *physically-based interpretation* was proposed by Fracchia et al. [7] (see also [17]). In the latter case, cell walls were simulated as springs, and pressure within cells was simulated using additional forces that acted in directions perpendicular to these spring. The geometry of the resulting structure was determined as the static equilibrium state. Developing fern gametophytes of *Microsorium linguaeforme* and *Dryopteris thelypteris*, as well as several other developmental processes and structures, were successfully modeled using this approach [3, 7, 17].

In spite of these encouraging results, the modeling experience revealed several shortcomings of map L-systems. The specification of map L-system models according to experimental data was difficult, control over the generated topologies was insufficient, and no mechanism for interactions between cells, other than mechanical interactions, was supported. Although these limitations were partially addressed in subsequent works [4, 6], the resulting formalisms still did not offer sufficient control over the simulated topologies and forms.

The modeling of developing surface structures has recently been addressed from new viewpoints by Giavitto and Michel (the MGS system [8]), and Smith *et al.* (the vv system [19]). Here we outline this latter approach and its potential applications to the modeling of biological structures.

2 Vertex-vertex systems

We developed the *vertex-vertex* (vv) formalism [19] to specify and implement algorithms that operate on dynamical systems with dynamical structure in the topological space of polygon meshes. VV is inspired by the philosophy of L-systems, in that it uses lineage information and context to identify elements of the structure. This reliance on local information makes it possible to avoid global identifiers, such as indices, which we consider undesirable both from the conceptual point of view (in reality, elements of a structure "know" only their local neighborhoods) and from the implementation point of view (indices are not a convenient construct for identifying elements of dynamically changing, irregular structures). VV also shares other characteristics with L-systems, such as the ease with which structural components can be inserted and deleted, and the ease of locally transporting information within the structure. In contrast to the declarative nature of L-system programs, however, vv programs have an imperative character.

The main data structure supported by vv (a counterpart to L-system strings) is a *graph rotation system* [5, 21]. A graph rotation system associates each vertex of a polygon mesh with an oriented circular list of

Name	Math. notation	vv statement	Description	Note	Fig
Query operations					
membership	$v \in x^\star$	is x in v	true iff vertex x is in the neighborhood of v		
order	$x < v$	$x < v$	true iff vertex x precedes vertex v in the universe U		
valence	$\lvert v^\star \rvert$	valence v	returns the number of neighbors of vertex v		
Selection operations					
any	let $x \in v^\star$	any in v	returns a neighbor of v	1	
next	$v^\star \uparrow x$	nextto x in v	returns vertex that follows x in the neighborhood of v	2	b
previous	$v^\star \downarrow x$	prevto x in v	returns vertex that precedes x in the neighborhood of v	2	c
Editing operations					
create	let $v \in U$	vertex v	create a vertex		
set neighborhood	$v^\star = \{a, b, c\}$	make $\{a,\ b,\ c\}$ nb_of v	set the neighborhood of v to the given circular list	3	a
erase	$v^\star = v^\star - x$	erase x from v	remove x from the neighborhood of v if $v \in x^\star$	4	d
replace	$v^\star = v^\star - a + x$	replace a with x in v	substitute x for a in the neighborhood of v	5	e
splice after	$v^\star + x \succ a$	splice x after a in v	insert x after a in the neighborhood of v	5	f
splice before	$v^\star + x \prec a$	splice x before a in v	insert x before a in the neighborhood of v	5	g

1) Returns the null vertex if $v^\star$ is empty. 2) Returns the null vertex if $x \notin v^\star$.
3) Not defined (error reported) if v appears in the list, or the same vertex is listed twice.
4) No effect if $v \notin x^\star$. 5) No effect if $v \notin a^\star$; not defined (error reported) if $x = v$ or $v \in x^\star$.

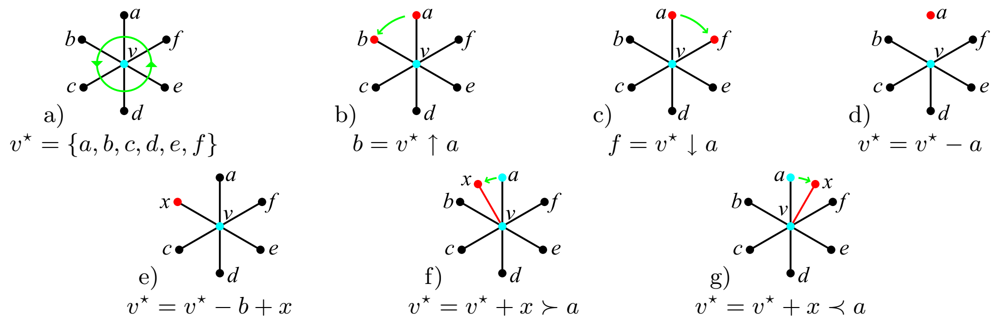
a) $v^\star = \{a, b, c, d, e, f\}$
b) $b = v^\star \uparrow a$
c) $f = v^\star \downarrow a$
d) $v^\star = v^\star - a$
e) $v^\star = v^\star - b + x$
f) $v^\star = v^\star + x \succ a$
g) $v^\star = v^\star + x \prec a$
a b c d e f v x

Table 1: Top: definition of the topological operations of the vertex-vertex algebra. Bottom: graphical interpretation of the selection and editing operations. a) Setting the initial neighborhood of vertex v. b-g) The results of various operations applied to v.

its adjacent vertices. This structure is manipulated using a set of operations called the *vv algebra* (Tab. 1). The *vv programming language* is an extension of C++ with statements of this algebra. VV programs are executed, and the results are visualized, in the *vv programming environment* [19].

3 Physically-based modeling of growing tissues

As with L-systems, vv captures the topology of the modeled structures. This topology must be complemented with geometric attributes to represent growing forms. In the previous work [19], these attributes were defined in purely geometric terms, using affine geometry operations [9]. Here we return to the concept of defining geometry through the intermediacy of physics. We consider the physically-based approach well suited to biological modeling, because it captures the emergence of form as a result of tensions (stresses) introduced into tissues by growth [1, 15].

In the examples discussed below, we treat the edges of a graph as springs that act on masses located at the nodes (as was the case for map L-systems). At each time step, the force acting on each vertex is the sum of the forces exerted by the springs connecting it to its neighbours. Other sources of forces may also be considered; for example, to represent the pressure of internal tissues on the external layer of a volumetric structure. The resulting velocities and accelerations are updated by forward Euler integration until an equilibrium is met. Algorithm 1 shows the simple vv implementation of this relaxation process. Growth and development are simulated by changing parameters (eg. the rest length) and configuration of selected springs. These changes can be induced and controlled by external factors, such as the orientation of springs and polygons with respect to gravity, and internal factors, such as the concentration of morphogens that diffiuse and react with each other in the simulated tissue.

```
forall v in V {
  v$acceleration.set(0.0, 0.0, 0.0);
  forall x in v {
    float dist = v$pos.distance(x$pos);
    float k = (rest_length - dist) / dist;
    v$acceleration += k * (v$pos - x$pos);
  }
  v$velocity += (v$acceleration - v$velocity * drag) * dt;
  v$pos += v$velocity * dt;
}
```

Algorithm 1: Sample vv code for calculating the force at each vertex in a set V when all vertices are interpreted as masses.

4 Modeling the Korn-Spalding cell division pattern

In [12], Korn and Spalding proposed a simple algorithm to mimic cell division patterns in plant epidermal tissues. This pattern was previously used as a test case for simulating developing tissues (e.g. [4]), and we include it here for the same purpose.

The Korn-Spalding pattern begins with a single hexagonal cell (Figure 1a). Its division is simulated by inserting an edge (cell wall) between two opposite sides of the cell, at one third the distance along each edge (Figure 1b). The daughter cells then return to the regular hexagonal shape (Figure 1c). In the subsequent iterations the process of cell division is repeated, with the dividing walls rotated each time approximately 120° with respect to their previous orientation.

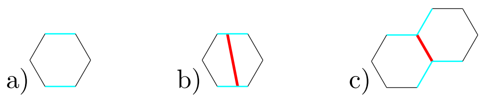

Figure 1: a) The initial cell with the top and bottom edge selected b) The insertion of an edge to divide the cell into two c) The cells after the physical simulation

To model this process in vv, we assume that all springs of the developing mass-spring system have the same rest length. We also assume that each cell exerts some pressure on its walls. Consequently, after division, the cells assume approximately regular hexagonal shapes. The first steps in the simulation of a growing tissue according to the Korn-Spaliding algorithm are shown in Figure 2.

5 Modeling a root

As a sample application of vv to the modeling of three-dimensional structures, we created a simple root model. The root is represented as a tubular polygonized surface, growing near the tip. Unlike the Korn-Spalding

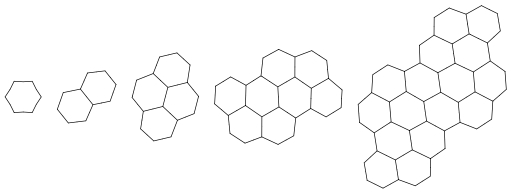

Figure 2: Development of the Korn-Spalding pattern

pattern, where polygons represented individual cells, here we treat the polygons as a formal discretization of a conceptually continuous surface.

In order to extend the root, new vertices are inserted into the mesh uniformly around the apical vertex (Figure. 3a and b). The arrangement of the edges around the apical vertex is then modified in a manner similar to Kobelt's $\sqrt{3}$ surface subdivision algorithm [11], so that the valence of the apical vertex returns to 6 (Figure 3c). Algorithm 2 specifies these operations in vv. The newly created edges have the same rest length as the older edges; thus, they expand, making the structure grow at the apex. An aditional force normal to the polygons assures that the apical vertex grows outwards.

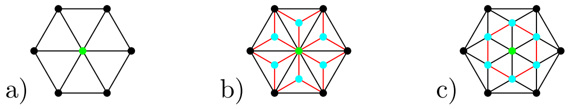

Figure 3: Simulating root tip growth. a) The initial configuration; the apical vertex is in the center. b) The configuration after the insertion of new vertices. c) The edges are adjusted such that the valence of the apical vertex returns to six. In the next iteration, steps a-c will be re-applied to the new hexagon surrounding the apical vertex.

```
 1  forall x in ‘apex {
 2    vertex y = nextto x in apex;
 3    vertex n;
 4    n$pos = 0.33 * (apex$pos + x$pos + y$pos);
 5    make {apex, x, y} nb_of n;
 6    splice n after x in apex;
 7    splice n after y in x;
 8    splice n after apex in y;
 9  }
10  forall x in ‘apex {
11    vertex a = nextto apex in x;
12    vertex b = prevto apex in x;
13    erase x from apex;
14    erase apex from x;
15    splice a after apex in b;
16    splice b after x in a;
17  }
```

Algorithm 2: Sample vv code for the insertion of vertices around a particular vertex *apex*. Lines 1–9 insert new vertices and edges as in Figure 3b. Lines 10–17 change the topology around the apex as in Figure 3c.

A sample structure generated by this algorithm is shown in Figure 5a. A simple model of root gravitropism is obtained by modulating the rest length of the springs as a function of the orientation of their incident polygons with respect to gravity, as shown in Figure 5b.

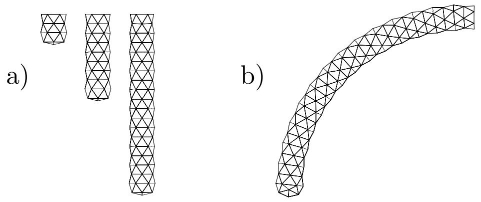

Figure 4: a) A growing root tip model, where all springs have the same rest length b) A simple model of root response to gravity. The rest length of springs at the top is larger than the length of the springs at the bottom.

6 Modeling a shoot apical meristem with primordia

Our last example is a generic model of a shoot apex. The growth of the receptacle is modeled in a manner similar to the growth of the root, described in the previous section (Figures 5a and b). Selected areas of the apex then undergo accelerated growth, forming new primordia. In the models shown, these areas are determined by Thornley's model of phyllotaxis [20], in which a morphogen, produced by the existing primordia, diffuses on the surface of the receptacle and assures proper positioning of the new primordia. Once the location of a new primordium has been selected, the corresponding region of the receptacle is subdivided (Figure 5c). Since the lengths of the resulting springs is smaller than the rest lengths, a tangential stress is created, causing the subdivided regions to buckle. An additional force, normal to the modeled surface, assures that the primordia grow outwards. (Figure 5d). As the existing primordia continue to grow and new primordia are created, the main geometric features of a growing apex emerge (Figure 6).

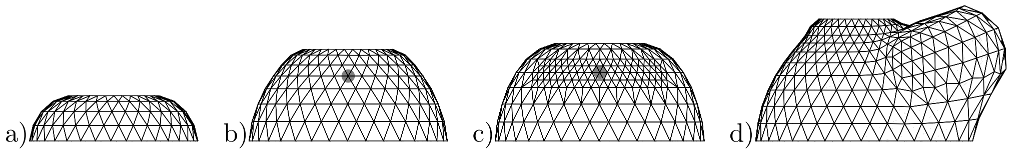

Figure 5: A model of a shoot apical meristem. a) The apex prior to the introduction of a primordium. b) As the apex continues to grow, the phyllotaxis model selects the vertex that will become the center of a primordium. c) The region around the selected vertex is adaptively subdivided. d) The subdivided region grows outwards. The apex has been rotated to best show the resulting shape.

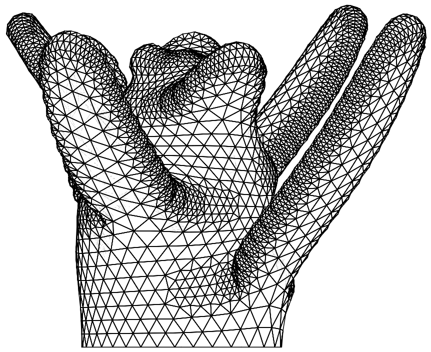

Figure 6: An apex with many primordia

7 Conclusions

We have outlined a method for modeling growing plant tissues and organs. The modeled objects (thin cellular layers or surfaces of volumetric structures) are represented as polygon meshes. These meshes are manipulated using a set of locally defined operators, called the vertex-vertex algebra, and acquire their form through simulation of stresses that result from local growth. The models can couple morphogenetic mechanisms of diffusive and reaction-diffusion type with physically-based simulations of tissue growth (for examples of related work see [2, 10]). In this abstract, we included models of an epidermal cell division pattern, root gravitropism, and shoot apex development. The presented methodology may serve as a computational framework for understaning complex developmental processes in nature, and may possibly be used to link patterns of gene expression to the development of plant form.

Acknowledgements

The support of this research by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

[1] E. Coen, A.-G. Rolland-Lagan, M. Matthews, A. Bangham, and P. Prusinkiewicz. The genetics of geometry. *PNAS*, 101(14):4728–4735, 2004.

[2] F. Cummings. The interaction of surface geometry with morphogens. *J. of Theor. Biol.*, 212:303–313, 2001.

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

[4] M. J. M. de Boer, F. D. Fracchia, and P. Prusinkiewicz. A model for cellular development in morphogenetic fields. In G. Rozenberg and A. Salomaa, editors, *Lindenmayer systems: Impacts on theoretical computer science, computer graphics, and developmental biology*, pages 351–370. Springer-Verlag, Berlin, 1992.

[5] J. Edmonds. A combinatorial representation of polyhedral surfaces (abstract). *Notices of the American Mathematical Society*, 7:646, 1960.

[6] F. D. Fracchia. Integrating lineage and interaction for the visualization of cellular structures. In J. Cuny, H. Ehrig, G. Engels, and G. Rozenberg, editors, *Graph grammars and their application to computer science; Fifth International Workshop*, Lecture Notes in Computer Science 1073, pages 521–535. Springer-Verlag, Berlin, 1996.

[7] F. D. Fracchia, P. Prusinkiewicz, and M. J. M. de Boer. Animation of the development of multicellular structures. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation '90*, pages 3–18, Tokyo, 1990. Springer-Verlag.

[8] J.-L. Giavitto and O. Michel. MGS: A programming language for the transformation of topological collections. Research Report 61-2001, CNRS - Université d'Evry Val d'Esonne, Evry, France, 2001.

[9] R. Goldman. On the algebraic and geometric foundations of computer graphics. *ACM Transactions on Graphics*, 21(1):52–86, Janaury 2002.

[10] L. Harrison, S. Whner, and D. Holloway. Complex morphogenesis of surfaces: theory and experiment on coupling of reaction-diffusion patterning to growth. *The Royal Society of Chemistry*, 120:277–294, 2002.

[11] L. Kobbelt. $\sqrt{3}$-subdivision. In *SIGGRAPH 2000*. ACM, 2000.

[12] R. Korn and R. Spalding. The geometry of plant epidermal cells. *New Phytologist*, 72(6):1357–1365, 1973.

[13] A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18(3):280–315, March 1968.

[14] 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.

[15] M. Marder, E. Sharon, S. Smith, and B. Roman. Theory of edges of leaves. *Europhysics Letters*, 62(4):498–504, 2003.

[16] P. Prusinkiewicz. A look at the visual modeling of plants using L-systems. *Agronomie*, 19:211–224, 1999.

[17] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[18] 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.

[19] C. Smith and P. Prusinkiewicz. Local specification of surface subdivision algorithms. In J. Pfaltz, M. Nagl, and B. Böhlen, editors, *Applications of Graph Transformations with Industrial Relevance (Second International Workshop, AGTIVE 2003)*, volume 3062 of *Lecture Notes in Computer Science*, pages 313–327, 2004.

[20] J. Thornley. Phyllotaxis. I. A Mechanistic Model. *Annals of Botany*, 39:491–507, 1975.

[21] A. White. *Graphs, groups and surfaces.* North-Holland, Amsterdam, 1973.