Gillespie-Lindenmayer Systems
for Stochastic Simulation of Morphogenesis

Mikolaj Cieslak
msciesla@ucalgary.ca

Przemyslaw Prusinkiewicz
pwp@ucalgary.ca

Department of Computer Science, University of Calgary
2500 University Dr. N.W., Calgary, AB T2N 1N4 Canada

Abstract

Background. *Lindenmayer systems (L-systems) provide a useful framework for modelling the development of multicellular structures and organisms. The parametric extension of L-systems allows for incorporating molecular-level processes into the models. Until now, the dynamics of these processes has been expressed using differential equations, implying continuously-valued concentrations of the substances involved. This assumption is not satisfied, however, when the numbers of molecules are small. A further extension that accounts for the stochastic effects arising in this case is thus needed.*
Results. *We integrate L-systems and the Gillespie Stochastic Simulation Algorithm to simulate stochastic processes in fixed and developing linear structures. We illustrate the resulting formalism with stochastic implementations of diffusion-decay, reaction-diffusion, and auxin-transport-driven morphogenetic processes.*
Conclusions. *Our method and software can be used to simulate molecular and higher-level spatially explicit stochastic processes in static and developing structures, and study their behaviour in the presence of stochastic perturbations.*

Keywords: Auxin-based patterning; Gillespie's Stochastic Simulation Algorithm; L-system; leaf development; Lotka–Volterra equations

1 Introduction

L-systems are a powerful formalism for modelling the development of growing linear and branching structures, from basal filamentous organisms to trees and entire plant ecosystems (Prusinkiewicz and Lindenmayer, 1990; Lane and Prusinkiewicz, 2002). The introduction of parameters into L-systems (Lindenmayer, 1974) has made it possible to capture genetic regulatory mechanisms and physiological processes that affect development by using a continuous-level representation of the substances involved (Coen et al., 2004; Allen et al., 2005; Buck-Sorlin et al., 2005; Prusinkiewicz et al., 2009). However, in many biological processes the numbers of molecules are relatively low, and the assumption of continuity of concentration is not satisfied (McAdams and Arkin, 1997; Elowitz et al., 2002; Rao et al., 2002). In these cases individual molecules must be considered, and algorithms that simulate stochastic fluctuations in molecule numbers are needed.

Stochastic simulation of reaction kinetics was pioneered by Gillespie (1976, 1977). The essence of his method is a discrete-event simulation of reactions between individual molecules in a spatially homogeneous, well-mixed system. To simulate heterogeneous systems, Gillespie (1976) proposed an extension, in which molecules diffuse between subvolumes in compartmentalized space. Spicher et al. (2008) incorporated Gillespie's algorithm into P-systems (Păun and Rozenberg, 2002), where the modelled system, such as a cell, is structured into a hierarchy of compartments enclosed by membranes. Lloyd-Price et al. (2012) reported further advancements in this direction.

Here we extend Gillespie's algorithm to compartmentalized linear structures by combining it with

L-systems. This allows for a concise specification of processes involving numerous compartments and provides a convenient, well defined computational framework for the stochastic modelling of biochemical processes taking place in both static and developing structures. We illustrate the strength of Gillespie-Lindenmayer systems, or Gillespie L-systems for short, using examples of morphogenetic processes that include reaction-diffusion and auxin-driven patterning. In each case, we highlight the impact that the number of molecules has on the characteristics of the solution. We also show that—as expected—the stochastic solutions converge to their continuous counterparts as the number of molecules increases.

2 Gillespie L-systems

As the stepping stone for constructing Gillespie L-systems, let us first review its constituent components: Gillespie's Stochastic Simulation Algorithm and Lindenmayer systems.

2.1 Gillespie's Stochastic Simulation Algorithm

Consider N chemical substances $S_1, \ldots, S_N$ that interact via M reactions $R_1, \ldots, R_M$. The system state is described by the vector $\mathbf{X} = (X_1, \ldots, X_N)$, where each entry X_k is the number of molecules of substance S_k at time t. The probability that a particular reaction $R_j \in (R_1, \ldots, R_M)$ will occur in the infinitesimal time interval $[t, t + dt)$ is given by the product $\alpha_j(\mathbf{X})dt$, where the term $\alpha_j(\mathbf{X})$ is called a *propensity function*. Under the assumption of a uniform distribution of molecules in space, this function is the product of the number of distinct combinations of reacting molecules, $h_j(\mathbf{X})$, and the stochastic reaction parameter, c_j, which depends on the type of reaction and temperature, and is related to the reaction rate k_j in chemical kinetics (Gillespie, 1976, 1977, 2007) (Table 1).

The evolution of the system over time is simulated by iterating the following steps:

1. determine the delay τ (inter-reaction time) with which the next reaction will take place, and the index of the next reaction $j \in (1, \ldots, M)$,

2. modify the state $\mathbf{X}$, taking into account the reactants removed from the system and products added to the system by reaction R_j, and

3. advance simulation time t by τ.

To generate one random pair (τ, j), Gillespie proposed two methods: the *first-reaction* method and the *direct* method. In the first reaction method, a putative time $\tau_{j'}$, $j' \in (1, \ldots, M)$, is generated for each reaction, and the reaction with the smallest time, τ_j, is chosen. In the direct method values for τ and j are generated according to a joint probability function of τ and j. The direct method is more efficient than the first reaction method because fewer random numbers must be generated per iteration step. Consequently, we only consider the direct method.

Gillespie showed that the time between two reactions can be described by an exponential distribution. First, the combined propensity of all possible reactions is computed:

$$\alpha_0(\mathbf{X}) = \Sigma_{j'=1}^{M} \alpha_{j'}(\mathbf{X}). \tag{1}$$

The inter-reaction time τ is then calculated as an exponentially distributed random variable with a mean value of $1/\alpha_0(\mathbf{X})$. The index of the next reaction is described by a discrete probability distribution, where $\alpha_j(\mathbf{X})/\mathbf{ff_0}(\mathbf{X})$ is the probability that the next reaction is R_j. The inter-reaction time τ and the next reaction R_j are then generated according to these probability distributions using the inversion method (Gillespie, 1976). Specifically, given two independent random numbers r_1 and r_2, generated with uniform distribution in the interval $(0, 1]$, the inter-reaction time is obtained using the formula

$$\tau = \frac{1}{\alpha_0} \ln \frac{1}{r_1}, \tag{2}$$

and the reaction index j is determined by solving the equation

$$\sum_{j'=1}^{j-1} \alpha_{j'} < r_2 \alpha_0 \leq \sum_{j'=1}^{j} \alpha_{j'}. \tag{3}$$

To incorporate spatial inhomogeneities into simulations of chemical systems, Gillespie (1976) proposed an extension to the basic Stochastic Simulation Algorithm, in which the volume V is divided into n subvolumes (also referred to as compartments or components) V_i $(i = 1, 2, \ldots, n)$. Two basic ideas underly this extension:

- Propensities of reactions taking place in different subvolumes are calculated individually for each subvolume. Thus, instead of a reaction

Reaction name	Reaction	h	c
source	$\emptyset \xrightarrow{k} products$	1	$k\Omega$
unimolecular	$S_1 \xrightarrow{k} products$	X_1	k
bimolecular	$S_1 + S_2 \xrightarrow{k} products$	$X_1 X_2$	k/Ω
trimolecular	$S_1 + 2S_2 \xrightarrow{k} products$	$X_1 X_2 (X_2 - 1)/2$	$2k/\Omega^2$

Table 1: Terms involved in the calculation of reaction propensities for several reaction types. Column h: the number of distinct molecular combinations of reactants S_1 and S_2 as a function of the number of their molecules, X_1 and X_2. Column c: the stochastic reaction parameter as a function of the reaction rate k. Symbol $\emptyset$ denotes an external source of products. Parameter Ω is the volume in which the reactions take place.

propensities α_j, the algorithm calculates propensity α_{ij} of reaction j taking place in subvolume i.

- Transport of a molecule from subvolume i to its neighbor $\hat{\imath}$ is treated as a unimolecular reaction that removes a molecule from subvolume i and deposits it in subvolume $\hat{\imath}$.

The inter-reaction time τ, the index j of the next reaction or transport event, and the index i of the subvolume where the next event occurs — or subvolume pair $(i, \hat{\imath})$ for transport events — are then generated using Equations 1–3, in which the summation extends to all pairs (i, j) of reaction events and triplets $(i, \hat{\imath}, j)$ of transport events. Formal details are presented by Stundzia and Lumsden (1996).

With the extension to subvolumes, Gillespie's algorithm can simulate the time evolution of a spatially explicit system in which the partition into subvolumes is fixed, while the molecules within each subvolume are distributed uniformly. To model the development of multicellular organisms, however, it is necessary to consider spatial structures in which the number of subvolumes and reactions associated with them may change over time (e.g., following cell division). We integrate Gillespie's algorithm with L-systems to provide a mechanism in which both fixed and developing structures can be simulated easily.

2.2 L-systems

Parallel rewriting systems, subsequently called L-systems, were introduced by Lindenmayer (1968) to specify, model, and reason about the development of multicellular structures (whole organisms or their parts) with filamentous or branching topology. A structure is represented by a *string* of *symbols* (letters) that correspond to its individual components, such as cells, higher level architectural units, or compartments resulting from a discretization of a continuous space. The evolution of the structure state over time is characterized by *rewriting rules*, also called *productions*, which specify how a *predecessor* symbol is replaced by zero, one, or more *successor* symbol(s) in the string. For example, the rule $A \rightarrow BC$ may be used to represent the division of cell A into two daughter cells, B and C. The rules are applied in parallel to the entire string, to capture the simultaneous progress of time in all parts of the organism (Prusinkiewicz and Lindenmayer, 1990).

Parametric L-systems associate numerical parameters with the symbols (Lindenmayer, 1974; Prusinkiewicz and Lindenmayer, 1990; Prusinkiewicz et al., 2018), for instance to quantify the chemical substances in each component of the structure. For example, the rule $C(z) \rightarrow C(z - \mu_z z \Delta t)$ describes the decay of substance Z in cell C, where z is the concentration of Z at time t, μ_z is the decay rate, and Δt is the time step.

A symbol with the associated parameters is referred to as a *module*. Communication between modules and the transport of substances within the structure can be modeled using *context-sensitive* productions, in which the production successor depends not only on the predecessor module, but also on its neighbors or *context*. Notationally, the context is separated from the predecessor by symbols $<$ and $>$. For example, production

$$W(J_L) < C(a) > W(J_R) \rightarrow C(a + (J_L - J_R)\Delta t) \quad (4)$$

specifies that concentration a of some substance in cell C changes according to the fluxes J_L and J_R through walls W on both sides of C. Context-sensitive productions facilitate modeling of multicellular structures, because they eliminate the need to index cells, and then reindex them as the structure develops and neighborhood relations change.

We will specify L-systems using the L+C modelling language implemented in the simulation program `lpfg`, which augments the expressive power of L-systems by combining them with C++ (Karwowski, 2002; Karwowski and Prusinkiewicz, 2003; Prusinkiewicz et al., 2007, 2018). L+C is relatively self-explanatory; for example, the context-sensitive production (4) is written in L+C as

```
W($J_L$) < C($a$) > W($J_R$):
  { produce C($a$+($J_L$-$J_R$)$\Delta t$); }
```

In a complete L+C program, symbols representing components of the structure are declared using the keyword `module`, with the parameter types listed in parentheses. The initial string is specified after the keyword `axiom`. Predefined keywords `Start` and `End` indicate optional blocks of C++ statements executed at the beginning or end of the simulation, for example to initialize variables used in the simulation or report statistics gathered during its execution. Likewise, `StartEach`, and `EndEach` indicate statements executed at the beginning or end of each simulation step. Following these blocks, the L-system productions are listed. A production may have more than one successor, each preceded by the `produce` keyword. In the original version of L+C the applicable successor is selected using a conditional statement; in Gillespie L-systems the selection mechanism is extended to include the stochastic mechanism described in the next section. The set of all productions can be partitioned into subsets called *groups* by the statement `group: id` (Prusinkiewicz et al., 2007), interspersed between productions. This statement assigns a numerical identifier `id` to all productions following it in the production list, until the next `group: id` statement occurs or the production list ends. The `group: id` statements are used in conjunction with the function `UseGroup(id)`, which is typically called within the `StartEach` block and specifies which group of productions should be used in the forthcoming simulation step. The notion of groups thus increases the flexibility of L-system programming by making it possible to employ different production groups in different simulation steps.

2.3 Integration of Gillespie's algorithm and L-systems

To combine Gillespie's algorithm and L-systems, we introduced the notion of a *Gillespie group* of productions, identified by the keyword `ggroup`. A Gillespie L-system may include both "ordinary" and Gillespie groups, but only one group is active in any simulation step. Productions in an ordinary group are applied to all modules in the string in parallel, consistent with the standard definition of L-systems. In contrast, a single module in the string and a single production or production successor are selected in a simulation step using a Gillespie group. This selection is effected using Gillespie's Stochastic Simulation Algorithm extended to subvolumes — identified with modules — with probabilities controlled by the expressions specified after the `propensity` keyword associated with each successor. As the selected production is applied, the current simulation time is advanced by the inter-reaction interval. The current time (integrated from the beginning of the simulation) can be accessed using the predefined L+C function `GillespieTime()`. Time management and event scheduling may require attention in simulations combining stochastic and deterministic productions; an example is discussed in Section 3.4 (Program 4b).

The following example illustrates the method. Consider substances A and B decaying in two types of modules, `C` and `D`, declared as follows:

```
module C(int,int); // A,B molecule count
module D(int);     // A molecule count
```

The decay events are defined by productions in the Gillespie group (although it is the only group, it must be selected explicitly using the `UseGroup(1)` statement):

```
Start: {UseGroup(1);}
ggroup 1:
C($A,B$): {
  propensity $\mu_a A$ produce C($A-1,B$);
  propensity $\mu_b B$ produce C($A,B-1$);
}
D($A$): {
  propensity $\mu_a A$ produce D($A-1$);
}
```

In general, the propensity of each reaction is the product of a stochastic reaction parameter and the number of combinations of the reacting molecules. As decay is a unimolecular reaction, the stochastic reaction parameter is equal to the reaction (decay) rate μ, and the numbers of combinations is equal to

	`C(4,2)`	`D(3)`	`D(4)`	`D(5)`	Σ
$\mathrm{A} \xrightarrow{0.1} \varnothing$	$4 \cdot 0.1$	$3 \cdot 0.1$	$4 \cdot 0.1$	$5 \cdot 0.1$	1.6
$\mathrm{B} \xrightarrow{0.2} \varnothing$	$2 \cdot 0.2$	—	—	—	0.4
Σ	0.8	0.3	0.4	0.5	2.0

Table 2: Example of propensity calculations for two decay reactions in a string of four modules.

the numbers of reactant molecules (Table 1). Suppose the rates are $\mu_a = 0.1$ and $\mu_b = 0.2$, and the current state of the system is described by the string of four modules

```
C(2,2) D(3) D(5) D(5).
```

A simulation step begins with the simulator (`lpfg`) calculating the propensity of each applicable reaction in each individual module, producing the results shown in Table 2. On this basis, the simulator stochastically selects a single reaction taking place in a single module using Equation 3. For instance, the probability of selecting reaction $\mathrm{A} \to \varnothing$ in module `C` is $\frac{0.4}{2.0} = 0.2$, the probability of selecting reaction $\mathrm{B} \to \varnothing$ in module `C` is also $\frac{0.4}{2.0} = 0.2$, and the probability of selecting reaction $\mathrm{A} \to \varnothing$ in the last module `D` is $\frac{0.5}{2.0} = 0.25$. Once the reaction and module have been selected, the simulator decrements the number of molecules of the reacting substance in the affected module by one, as specified by the production successor, and determines stochastically the inter-reaction time using Equation 2 with $a_0 = 2.0$. The next simulation step can then be performed, beginning with the recalculation of propensities. The entire simulation run is a sequence of such steps.

2.4 Petri nets

As the reactions become more complicated, it is convenient to represent them graphically using *stochastic Petri nets* (Goss and Peccoud, 1998). We use them in this paper to visually complement textual specifications of reaction systems. A Petri net is a directed graph with two types of nodes: *places*, drawn as circles, and *transitions*, drawn as rectangles (Figure 1). In the context of chemical processes, places represent substances, and transitions represent reactions or movement of molecules between components. The nodes are connected by directed arcs (arrows), such that the arcs pointing from places to transitions denote the reactants, and arcs pointing from transitions to places denote the products. Arcs connecting places to places or transitions to transitions are not allowed. In a stochastic Petri net transitions are executed or *fired* by a stochastic process: in the scope of this paper, using Gillespie's Stochastic Simulation Algorithm.

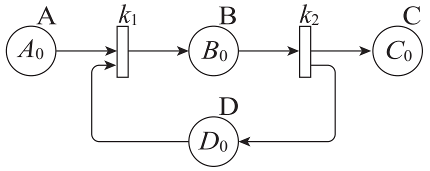

Figure 1: Petri net example of the chemical system $\mathrm{A} + \mathrm{D} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C} + \mathrm{D}$. The labels A_0, B_0, C_0, and D_0 indicate the initial ($t = 0$) numbers of molecules of each substance.

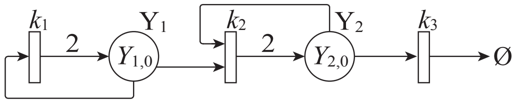

Figure 2: Stochastic Petri net for Lotka's chemical system.

3 Examples

We now illustrate Gillespie L-systems with a sequence of examples of increased complexity.

3.1 The Lotka-Volterra process

The first example revisits the Stochastic Simulation Algorithm implementation of Lotka's (1920) chemical process with an oscillatory behaviour, originally presented by Gillespie (1977). The purpose of this example is to show the basic structure of an L+C program using a Gillespie group of productions. The reactions are:

$$R_1 : \mathrm{Y}_1 \xrightarrow{k_1} 2\mathrm{Y}_1$$

$$R_2 : \mathrm{Y}_1 + \mathrm{Y}_2 \xrightarrow{k_2} 2\mathrm{Y}_2$$

$$R_3 : \mathrm{Y}_2 \xrightarrow{k_3} \varnothing,$$

where Y_1 and Y_2 are two chemical substances, and k_1, k_2 and k_3 are the reaction rates. The stochastic Petri net describing this system is shown in Figure 2, and the corresponding L+C implementation is given in Program 1.

Program 1 L+C implementation of Lotka's chemical system.

```
/* declaration of constants */
const float k1 = 1.0;
const float k2 = 0.01;
const float k3 = 1.0;

/* variable initialization */
float t; // time of last reaction
Start: {t = 0; UseGroup(1);}
EndEach: {t = GillespieTime();}

/* define module with two substances */
module C(int,int);

axiom: C(100,100); // initial conditions

/* computation of reaction propensities */
ggroup: 1;
C(Y1,Y2): {
  propensity k1Y1 produce C(Y1+1,Y2);
  propensity k2Y1Y2 produce C(Y1−1,Y2+1);
  propensity k3Y2 produce C(Y1,Y2−1);
}
```

The program operates on a single module C representing the entire reaction volume, with two parameters specifying the current number of molecules Y_1 and Y_2. We assume that the volume Ω in which the reactions take place is equal to 1, and thus the stochastic reaction parameters c_j for all reactions, including the bimolecular reaction R_2, are equal to their reaction rates k_j (Table 1). In each simulation step, the simulator computes the propensities of the applicable productions (in this example, all three of them are always applicable), selects one using Gillespie's direct method, applies it to module C, and advances the current simulation time by the inter-reaction interval. A sample run of Program 1 is shown in Figure 3. Consistent with the theory (Lotka, 1920), the number of molecules exhibits an oscillatory behaviour. The jagged character of the plots, clearly seen in Figure 3B, reflects the inherently discrete and stochastic nature of the Lotka system, as the molecules only occur in integer numbers and enter into reactions randomly.

Independently of Lotka's chemical system, Volterra (1928) proposed an ecological model of fish catches in the Adriatic Sea that exhibits the same oscillatory behaviour. In this case, the reactions are interpreted as follows:

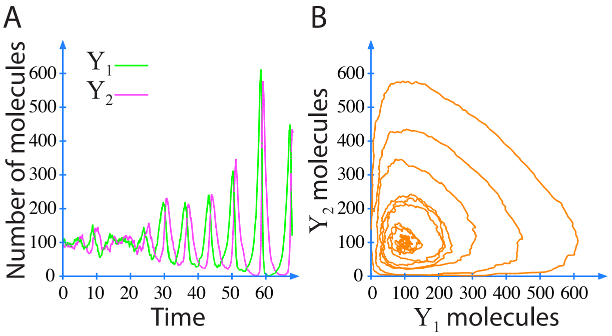

Figure 3: Stochastic simulation of Lotka's chemical system. (A) A plot showing the number of molecules of Y_1 (green line) and Y_2 (magenta line) over time. (B) A phase plot of the number of molecules.

R_1**:** a prey species Y_1 reproduces while feeding on some food source that does not deplete over time,

R_2**:** a predator species Y_2 reproduces while feeding on the prey species, and

R_3**:** the predator species dies by natural causes.

This reinterpretation shows that applications of Gillespie L-systems are not limited to molecular-level simulations.

3.2 Diffusion and decay

Let us now apply a Gillespie L-system to model a simple spatially explicit process, in which substance A diffuses and decays in a one-dimensional medium. The standard description of this process has the form of the partial differential equation

$$\frac{\partial a}{\partial t} = -\mu_a a + D_a \frac{\partial^2 a}{\partial x^2}, \qquad (5)$$

where $a = a(x,t)$ is the concentration of A at point x and time t, μ_a is its decay rate, and D_a is the diffusion rate (Edelstein-Keshet, 1988). This model can be spatially discretized into a linear structure

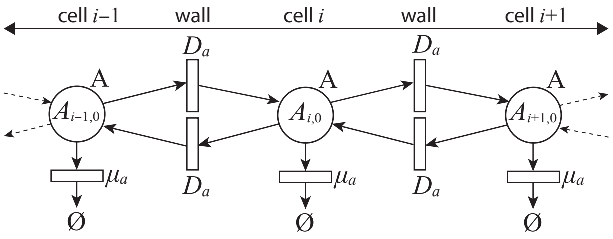

Figure 4: Stochastic Petri net for diffusion and decay.

(a one-dimensional cell complex) with two types of components: cells and cell walls (Prusinkiewicz and Lane, 2013). Following the law of mass conservation, the rate of change in concentration a_i of substance A in cell i is then equal to:

$$\frac{da_i}{dt} = J_{(i-1)\to i} - J_{i\to i+1} - \mu_a a_i, \tag{6}$$

where $J_{(i-1)\to i}$, $i = 2, 3, \ldots, n-1$, is the flux of A through the wall between cells $i-1$ and i. According to Fick's law, this flux is proportional to the concentration difference $a_{i-1} - a_i$:

$$J_{(i-1)\to i} = D_a(a_{i-1} - a_i). \tag{7}$$

Program 2a Gillespie L-system implementation of the diffusion-decay process using explicit representation of walls.

```
const float $D_a = 2.5$; // diffusion rate
const float $\mu_a = 0.01$; // decay rate
const int $X = 100$; // a boundary condition

module C(int); // a cell
module W(int); // a wall

/* control of production application */
int $n = 1$;
StartEach:{UseGroup($n$);}
EndEach:{$n = (n == 1)$?2 : 1;}

/* create cells separated by walls */
axiom: C(X)W(0)C(0)W(0)...C(0)W(0)C(X)

/* decay and diffusion events */
ggroup 1:
W($dir_l$) < C($A$) > W($dir_r$): {
  propensity $\mu_a A$ produce C($A-1$);
}
C($A_l$) < W($dir$) > C($A_r$): {
  propensity $D_a A_l$ produce W(1);
  propensity $D_a A_r$ produce W(-1);
}

/* transport a molecule if necessary */
group 2:
W($dir_l$) < C($A$) > W($dir_r$): {
  produce C($A + dir_l - dir_r$);
}
W($dir$): {
  produce W(0);
}
```

A Petri net corresponding to Equations 6 and 7 is shown in Figure 4. It leads to the Gillespie L-system implementation in Program 2a. The program begins with the definition of parameters: diffusion rate D_a, decay rate μ_a, and the number X of molecules A in the boundary cells. Declarations of two module types — cell C and wall W — follow. The parameter of cell C is a non-negative integer denoting the number of molecules of substance A in this module. The parameter of wall W is the integer -1, $+1$ or 0, indicating whether a molecule will be transported through W to the left, to the right, or not at all. Program execution is controlled by the StartEach and EndEach statement blocks, which alternate between ggroup 1 and group 2 in consecutive simulation steps. The Gillespie group, ggroup 1, has two rules. The first rule, with the predecessor W(dir_l) $<$ C(A) $>$ W(dir_r), describes the decay of a single molecule of A. The context is included to maintain boundary conditions: the first and the last cell lack an incident wall (see the axiom), which fixes the number of molecules of A in them. The second rule, with the predecessor C(A_l) $<$ W(dir) $>$ C(A_r), specifies

Program 2b Alternative Gillespie L-system implementation of the diffusion-decay process using productions operating on cell pairs.

```
const float $D_a = 2.5$; // diffusion rate
const float $\mu_a = 0.01$; // decay rate
const int $X = 100$; // a boundary condition

module B(int); // a boundary cell
module C(int); // an interior cell

/* use Gillespie group */
Start:{UseGroup(1);}

/* create a string of cells */
axiom: B(X)C(0)C(0)...C(0)B(X)

/* decay and diffusion events */
ggroup 1:
C($A$): {
  propensity $\mu_a A$ produce C($A-1$);
}
C($A_l$) C($A_r$): {
  propensity $D_a A_l$ produce C($A_l-1$) C($A_r+1$);
  propensity $D_a A_r$ produce C($A_l+1$) C($A_r-1$);
}

/* enforce boundary conditions */
B($A_l$) < C($A_r$): {
  propensity $D_a A_l$ produce C($A_r+1$);
  propensity $D_a A_r$ produce C($A_r-1$);
}
C($A_l$) > B($A_r$): {
  propensity $D_a A_l$ produce C($A_l-1$);
  propensity $D_a A_r$ produce C($A_l+1$);
}
```

diffusion of a molecule to the left or to the right using alternative propensity... produce statements. The actual transport is effected by the first production in the standard L+C group 2, which operates in parallel on all modules C in the string except for the boundary cells. If the transport direction is left to right (dir_l or dir_r is 1), the cell to the left of the wall will lose a molecule and the cell to the right will gain one. Conversely, if the direction is right to left (dir_l or dir_r is -1), the cell to the left of the wall will gain a molecule and the cell to the right will lose it. The last production resets all diffusion events to 0, in preparation for the next iteration of the simulation.

Results of sample simulation runs are shown in Figures 5A–C. The initial concentration of molecules A in the interior cells was assumed to be 0. We varied the number X of molecules of A in the boundary cells between different simulation runs to show the effects of increasing this number on the results. In each simulation run, the cell volume Ω was set to be numerically equal to X, so that concentration $\frac{X}{\Omega}$ was equal to 1. This normalization facilitated comparisons of simulations with different molecule numbers. As expected (Wang et al., 2007; Vigelius and Meyer, 2012; Gillespie, 2007), the stochastic solution became less noisy as the number of molecules increased. For a comparison, Figure 5D shows a solution to a deterministic diffusion-decay system with continuous representation of concentrations (Equations 6 and 7). The apparent convergence to the deterministic solution is consistent with the convergence of the molecular motion produced by voxel-hopping to the standard diffusion equation (Gillespie et al., 2014). The stochastic simulation has the advantage of better representing the diffusion-decay process when the number of molecules is small.

Incidentally, the same diffusion-decay process can also be simulated using the Gillespie L-system given by Program 2b. In this case, diffusion is effected by a single production (with two alternative produce statements) operating on a *pair* of modules. The boundary conditions, maintaining constant number of molecules in the first and last cell, are enforced by representing these cells using distinct modules B. The implementation of diffusion in a single simulation step makes Program 2b somewhat faster than Program 2a (approximately 20% in our implementation). On the other hand, Program 2a emphasizes the local character of diffusive transport, and is consistent with the standard definition of L-systems, according to which each production has a single predecessor (this assumption is key to the parallel operation of standard L-systems). The choice between Programs 2a and 2b is thus largely a matter of programming style.

3.3 Reaction-diffusion

In this example, we construct a Gillespie L-system to simulate a stochastic reaction-diffusion patterning process (Turing, 1952; Gierer and Meinhardt, 1972; Meinhardt, 1982).We focus on pigmentation patterning in sea shells, for which models expressed using partial differential equations are well understood (Meinhardt and Klingler, 1987, 1988; Fowler et al., 1992; Meinhardt, 2009). The *activator-substrate* variant of these models is described by the following equations (Meinhardt, 2009):

$$\frac{\partial a}{\partial t} = \rho s \left(a^2 + \rho_0\right) - \mu_a a + D_a \frac{\partial^2 a}{\partial x^2}, \tag{8}$$

$$\frac{\partial s}{\partial t} = \sigma - \rho s \left(a^2 + \rho_0\right) - \mu_s s + D_s \frac{\partial^2 s}{\partial x^2}. \tag{9}$$

Each of these equations extends the diffusion-decay system, discussed previously, with terms representing reactions between activator A with concentration a and substrate S with concentration s. At the molecular level, these reactions have the form:

$$R_1 : \emptyset \xrightarrow{\sigma} \mathrm{S} \tag{10}$$

$$R_2 : \mathrm{S} \xrightarrow{\rho\rho_0} \mathrm{A} \tag{11}$$

$$R_3 : \mathrm{S} + 2\mathrm{A} \xrightarrow{\rho} 3\mathrm{A}. \tag{12}$$

Following Table 1, their propensities are $\alpha_1 = \sigma\Omega$, $\alpha_2 = \rho\rho_0 S$ and $\alpha_3 = \frac{\rho}{\Omega^2} SA(A-1)$, respectively. The resulting stochastic Petri net combines these reactions with two diffusion-decay models: one for activator A and another for substrate S (Figure 6). Correspondingly, Program 3, specifying the activator-substrate process in L+C, has production groups similar to Program 2.

Figures 7A-C show the results of three runs of the simulation of the pigmentation pattern found in the sea shell *Amoria undulata* (Meinhardt, 2009; Fowler et al., 1992). The images represent consecutive states of the simulation in a row of $n = 100$ modules, obtained for three different values of (mathematical) cell volume Ω: 10, 100 and 1000. As in the diffusion-decay model, the volume Ω was numerically equal to the initial number of molecules

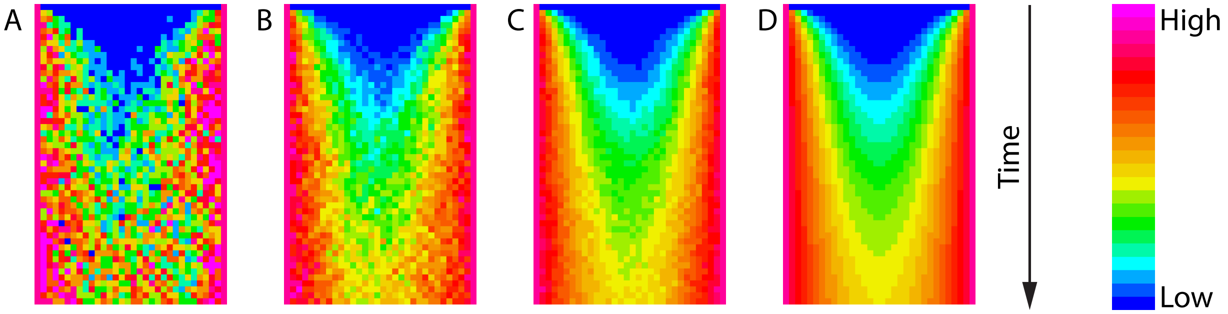

Figure 5: Visual representation of the stochastic and deterministic solutions to the diffusion-decay process. The change in concentration over time is shown from top to bottom in unit time steps from 0 to 50 units. High to low concentrations correspond to the colours given in the bar on the right. (A–C) Stochastic solutions produced by Program 2 with $X = 10$ (A), 100 (B) and 1000 (C) molecules in the boundary cells. In each case, the filament is visualized as a row of $n = 32$ cells, with volume Ω equal numerically to X. (D) Deterministic solution with continuous concentration obtained by solving Equations 6 and 7 numerically.

A and S in each run. The solutions used the following parameter values: $\rho = 0.1$, $\rho_0 = 0.005$, $\mu_a = 0.08$, $D_a = 0.004$, $\mu_s = 0$, and $D_s = 0$. The σ parameter was modulated for each cell according to a sine function in order to generate lines of undulating shape (Fowler et al., 1992). Specifically, $\sigma = \sigma_{min} + (sin(2\pi \cdot 3i/n) + 1)(\sigma_{max} - \sigma_{min})/2$, with $\sigma_{min} = 0.02$ and $\sigma_{max} = 0.032$, for each cell $i = 1, \ldots, n$. For a comparison, Figure 7D shows a numerical solution of Equations 8 and 9 assuming a continuous representation of concentrations. In nature, these temporal progressions take place on the growing shell margin, leaving a pigmentation pattern “frozen" on the shell surface. The patterns found in *Amoria undulata* seashells, e.g. (Fowler et al., 1992, Fig. 12) exhibit irregularities that are consistent with the stochastic simulation in Figure 7C. The stochastic model thus gives a satisfactory explanation for irregularities that are observed in the natural patterns, but are not captured by the deterministic model.

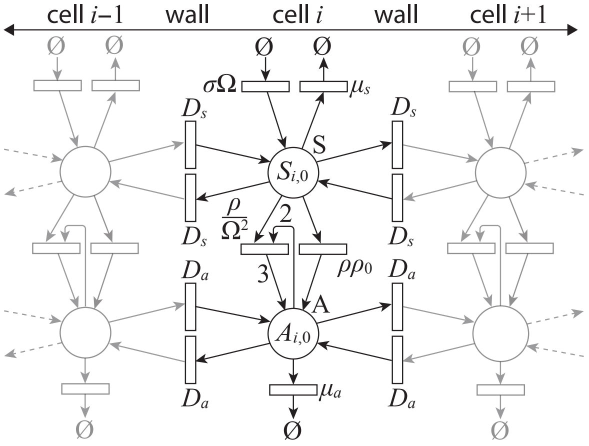

Figure 6: Stochastic Petri net for the activator-substrate process.

Program 3 Gillespie L-system productions implementing a stochastic model of the activator-substrate process.

```
int n = 1;
StartEach:{UseGroup(n);}
EndEach:{n = (n == 1)?2 : 1;}

ggroup 1:
W(dir_Al,dir_Sl) < C(A,S) > W(dir_Ar,dir_Sr): {
  propensity σΩ produce C(A,S+1);
  propensity μ_sS produce C(A,S−1);
  propensity ρ/Ω² SA(A−1) produce C(A+1,S−1);
  propensity ρρ_0S produce C(A+1,S−1);
  propensity μ_aA produce C(A−1,S);
}
C(A_l,S_l) < W(dir_A,dir_S) > C(A_r,S_r): {
  propensity D_aA_l produce W(1,0);
  propensity D_aA_r produce W(−1,0);
  propensity D_sS_l produce W(0,1);
  propensity D_sS_r produce W(0,−1);
}

group 2:
W(dir_Al,dir_Sl) < C(A,S) > W(dir_Ar,dir_Sr): {
  produce C(A+dir_Al−dir_Ar,S+dir_Sl−dir_Sr);
}
W(dir): {
  produce W(0);
}
```

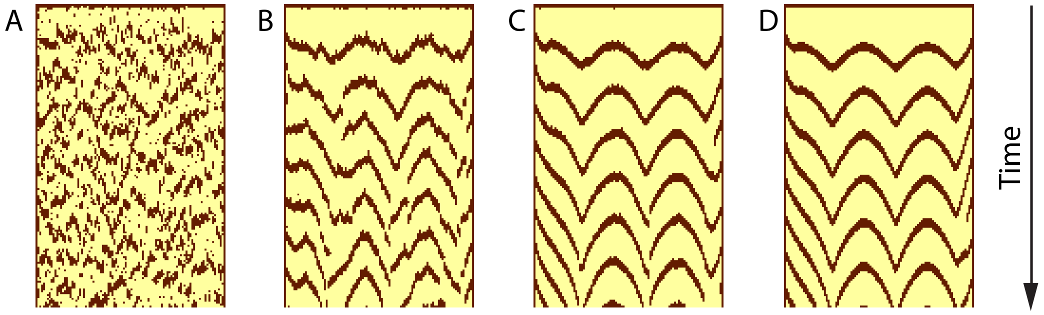

Figure 7: Simulations of the *Amoria undulata* seashell pattern formation showing the stochastic solutions with (A) $\Omega = 10$, (B) $\Omega = 100$ and (C) $\Omega = 1000$, and (D) the deterministic solution to the activator-substrate process. Each pattern has 100 columns and 160 rows.

3.4 Auxin-driven morphogenesis

In this last example, we consider the regulation of leaf shape by a key component of plant morphogenesis, the hormone auxin (Sachs, 1991; Zažímalová et al., 2014). The export of auxin from a cell relies on the activity of carriers in the cell membrane. Among them, the PIN1 protein appears to play the most prominent morphogenetic role. The allocation of PIN1 to different regions of the membrane is regulated by auxin itself (Paciorek et al., 2005), creating a feedback loop that plays an essential role in many aspects of plant morphogenesis. Molecular-level details of this process are the subject of ongoing research (Abley et al., 2013; Cieslak et al., 2015), but within the leaf margin the end result is a preferential allocation of PIN1 to regions of the cell membrane abutting neighbouring cells with a high concentration of auxin (Scarpella et al., 2006; Hay et al., 2006). As a result of this “up-the-gradient" allocation (Smith et al., 2006; Jönsson et al., 2006), a pattern of auxin concentration maxima and minima emerges. The maxima promote an outgrowth of future serrations, lobes or entire leaflets, thus shaping the developing leaf (Hay et al., 2006; Bilsborough et al., 2011; Bar and Ori, 2014; Runions et al., 2017; Conklin et al., 2019).

The first model of the above process was formulated in terms of differential equations (Bilsborough et al., 2011). Here we construct a stochastic model paralleling the version described by Prusinkiewicz and Lane (2013). The leaf margin is represented as sequence of cells that grow and divide upon reaching a threshold length. The volume of each cell is computed dynamically as the product of its cross-sectional area S, assumed to be constant, and length x, affected by growth and divisions. Within each cell, the model accounts separately for the number of PINs in the cytoplasm (PIN) and in the membrane regions abutting the left and right neighboring cells (PIN_l and PIN_r, respectively). We distinguish between molecule count, denoted without brackets, for instance A for auxin molecules A, volumetric concentration $[A] = \frac{A}{xS}$, and — in the case of molecules allocated to membrane regions — area concentrations, for instance $[\![PIN_l]\!] = \frac{PIN_l}{S}$.

PIN concentration on the membrane is the result of two processes: *exocytosis*, or the allocation of PIN from the cytoplasm to the membrane, and *endocytosis*, or the return to the cytoplasm. Consistent with the up-the-gradient polarization model, we assume that the propensity of exocytosis PIN → PIN$_l$, which allocates a PIN molecule in cell i to the membrane region abutting its left neighbor $i-1$, is proportional to the region area S, the PIN concentration in the cytoplasm, $[PIN]$, and the auxin concentration in the neighboring cell, $[A_l]$:

$$\alpha_{1l} = \sigma_p[A_l][PIN]S = \frac{\sigma_p}{x_l x S} A_l PIN. \tag{13}$$

We further assume that endocytosis PIN$_l$ → PIN, which deallocates a PIN molecule from the membrane to the cytoplasm, has a propensity characteristic of a decay process:

$$\alpha_{2l} = \mu_p PIN_l. \tag{14}$$

Analogous equations apply to the membrane segment abutting cell $i+1$:

$$\alpha_{1r} = \sigma_p[A_r][PIN]S = \frac{\sigma_p}{x_r x S} A_r PIN, \tag{15}$$

$$\alpha_{2r} = \mu_p PIN_r. \tag{16}$$

The number of auxin molecules in the cell changes as the result of their production, turnover, and transport to and from the neighboring cells. We assume that auxin is produced at a constant rate throughout the cell volume, which yields the propensity of event $\varnothing \to$ A equal to

$$\alpha_3 = \sigma_a x S. \tag{17}$$

Auxin turnover A $\to \varnothing$ is treated as a random decay of auxin molecules, which yields propensity

$$\alpha_4 = \mu_a A. \tag{18}$$

Consistent with Fick's law, the propensity of exporting auxin diffusively to a neighboring cell is proportional to auxin concentration $[A]$ and interface area S:

$$\alpha_5 = D[A]S = \frac{D}{x}A. \tag{19}$$

A related formula describes the propensity of auxin transport facilitated by PIN, except that, in this case, the transport rate is modulated by the area density of PIN in the respective membrane:

$$\alpha_{6l} = T[\![PIN_l]\!][A]S = \frac{T}{xS}PIN_l A. \tag{20}$$

Analogously, propensity of transport to the right neighbor is

$$\alpha_{6r} = T[\![PIN_r]\!][A]S = \frac{T}{xS}PIN_r A. \tag{21}$$

The Petri net summarizing these processes is shown in Figure 8, and the essential part of the resulting Gillespie L-system is given in Program 4a. The first rule in Gillespie ggroup 1 captures the exocytosis and endocytosis of PINs, and the production and turnover of auxin. The application of this rule begins with a calculation of the number of PIN molecules in the cytoplasm using the equation

$$PIN = \lfloor xS[PIN_0] \rfloor - PIN_l - PIN_r, \tag{22}$$

where $[PIN_0]$ is the volumetric concentration of PIN in the cell, assumed to be constant. The second Gillespie rule stochastically selects a transport event. This rule is centered on a wall rather than a cell, and requires slightly more complicated indexing than that in Equations 19–21. The first subscript points to the left or right incident cell, and the second one, if present, to the left or right membrane region within that cell. This rule works in concert with the productions in group 2 to effect molecule transport between cells, in a manner similar to the diffusion-decay and reaction-diffusion Gillespie L-systems.

The final element of the model is the development of the leaf margin, giving the simulated leaf its shape. Beginning with an initial shape resembling a leaf primordium, the cells elongate at a constant rate. Moreover, they are displaced in the normal direction at a rate proportional to the concentration of auxin, creating leaf lobes as described by Bilsborough et al. (2011) and Prusinkiewicz and Lane (2013). The rules implementing this growth are not shown in Program 4, as they are not specific to Gillespie L-systems. However, cell division is described by the rule in group 3. Upon reaching threshold length x_{max} the cell divides symmetrically. Auxin molecules are apportioned according to the size of child cells: due to the symmetry, each daughter cell inherits $\frac{1}{2}$ of all molecules (only slightly more complex, but more consistent with the spirit of stochastic simulation would be to divide molecules A between left and right cell using a binomial distribution). Auxin concentration in the daughter cells is thus the same as it was in the mother cell. Likewise, Equation 22 apportions PIN molecules to the daughter cells proportionally to their volume. The numbers of PIN molecules allocated to the left and right membrane regions of each daughter cell is assumed to be the same as it was in the mother cell, so that the cell division does not disrupt the auxin flow. This

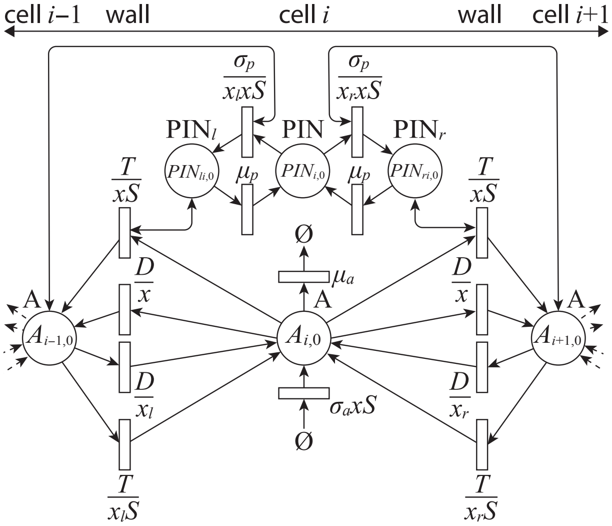

Figure 8: A stochastic Petri net model of the auxin-driven patterning of a leaf margin.

Program 4a Gillespie L-system productions implementing a stochastic model of auxin-driven leaf margin development.

```
ggroup 1:
C($A_l, PIN_{ll}, PIN_{lr}, x_l$) W($dir_l$)
   < C($A, PIN_l, PIN_r, x$) >
      W($dir_r$) C($A_r, PIN_{rl}, PIN_{rr}, x_r$): {
 int $PIN$ = $\lfloor xS[PIN_0] \rfloor - PIN_l - PIN_r$;
 propensity $\frac{\sigma_p}{x_l x S} A_l PIN$ // exocytosis: left
   produce C($A, PIN_l + 1, PIN_r, x$);
 propensity $\mu_p PIN_l$       // endocytosis: left
   produce C($A, PIN_l - 1, PIN_r, x$);
 propensity $\frac{\sigma_p}{x_r x S} A_r PIN$ // exocytosis: right
   produce C($A, PIN_l, PIN_r + 1, x$);
 propensity $\mu_p PIN_r$       // endocytosis: right
   produce C($A, PIN_l, PIN_r - 1, x$);
 propensity $\sigma_a x S$        // auxin production
   produce C($A + 1, PIN_l, PIN_r, x$);
 propensity $\mu_a A$          // auxin turnover
   produce C($A - 1, PIN_l, PIN_r, x$);
}
C($A_l, PIN_{ll}, PIN_{lr}, x_l$)
   < W($dir$) >
      C($A_r, PIN_{rl}, PIN_{rr}, x_r$): {
 propensity $\frac{T}{x_l S} PIN_{lr} A_l + \frac{D_a}{x_l} A_l$
   produce W(1);  // auxin transport: right
 propensity $\frac{T}{x_r S} PIN_{rl} A_r + \frac{D_a}{x_r} A_r$
   produce W($-1$); // auxin transport: left
}

group 2:
W($dir_l$) < C($A, PIN_l, PIN_r, x$) > W($dir_r$): {
   produce C($A + dir_l - dir_r, PIN_l, PIN_r, x$);
}
W($dir$): { produce W(0); }

group 3:
C($A, PIN_l, PIN_r, x$): {
  /* grow the cell margin */
  /*   details not shown   */
  if ($x \geq x_{max}$) {
    float $x_l = 0.5x$;
    int $A_l = \lfloor \frac{x_l}{x} A \rfloor$;
    produce C($A_l, PIN_l, PIN_r, x_l$)
            W(0)
            C($A - A_l, PIN_l, PIN_r, x - x_l$);
  }
  produce C($A, PIN_l, PIN_r, x$);
}
```

implies preserving PINs in the existing membrane regions, and allocating PINs to the emerging wall between the daughter cells.

Growth and cell divisions could be simulated using stochastic productions as well, but the model achieves better performance by employing standard, deterministic L-system productions for this purpose.

Program 4b Control of the production application in Program 4a.

```
float $t, gt, \Delta gt$;
int $n$;
Start: {$t = 0$; $gt = \Delta gt = 0.5$; $n = 1$;}
StartEach: {UseGroup($n$);}
EndEach: {
  switch ($n$) {
    case 1:
      $n = 2$;
      break;
    case 2:
      $t$ = GillespieTime();
      $n = (t < gt)$ ? 1 : 3;
      break;
    case 3:
      $gt = gt + \Delta gt$;
      $n = (t < gt)$ ? 1 : 3;
      break;
  }
}
```

The resulting combination of stochastic and deterministic productions requires a careful scheduling of events (Lu et al., 2004). This is achieved by preceding the productions in Program 4a with a code controlling their execution, listed in Program 4b. Molecular processes are simulated first, by alternating between productions in ggroup 1 and group 2 as in Programs 2a and 3. This phase continues until the current simulation time t, returned by the GillespieTime function, reaches time gt of the next growth-and-cell-division event. Productions in group 3 are then executed, and time gt is incremented by the predefined interval Δgt. At this point, the simulation of molecular processes captured by productions in ggroup 1 usually resumes. It is possible, however, that simulation time t is still greater than growth-and-division time gt (this is particularly likely when the number of molecules in the model is small, implying large inter-reaction times and steps in the values of t). In this case, time gt is incremented again, and growth and cell division are simulated, until the condition $t < gt$ becomes true.

Figure 9 shows sample simulation results, obtained using parameter values: $T = 0.2$, $D = 35$, $\sigma_a = 4$, $\mu_a = 0.25$, $[PIN_0] = 100$, $\sigma_p = 40$, and $\mu_p = 5$. All cells had cross section S = 1 and were dividing upon reaching the threshold length $x_{max} = 22.5$. Their initial lengths $x_{i,0}$ were close to x_{max}. The boundary cells were set to maintain zero auxin concentration, and the remaining 6 cells of the

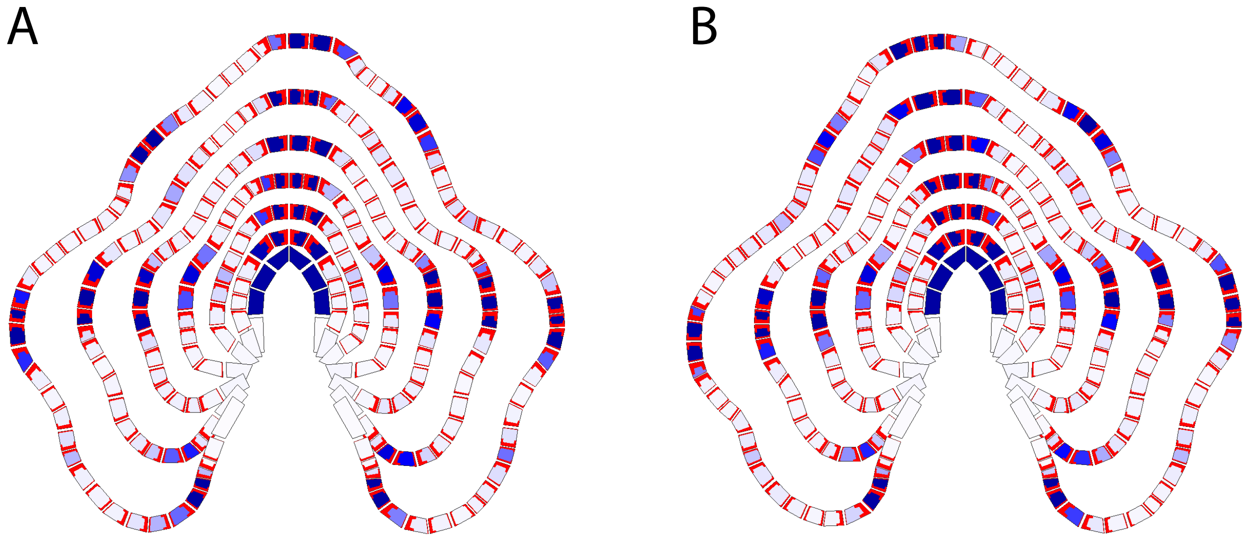

Figure 9: Visualization of the ivy leaf model showing growth of the margin over time. Figures A and B are examples of forms resulting from different runs of the same stochastic model. Each cell is represented as a trapezoid, coloured according to its auxin concentration: white to dark blue corresponds to low to high concentration. PIN concentrations are visualized as red lunes on cell edges; wider lunes represent larger concentrations.

leaf primordium were initialized with $100x_iS$ auxin molecules each. Initially no PIN molecules were allocated to the cell membranes. Although these values — and details of the underlying equations — have been chosen arbitrarily, the model does illustrate the general applicability of Gillespie L-systems to the simulation of auxin-driven patterning processes, and their ability to capture random variations in this context.

4 Conclusions

We have proposed an integration of Gillespie's Stochastic Simulation Algorithm and L-systems as a method for simulating stochastic processes in structures with a constant or variable number of modules representing cells or higher-level compartments. While "ordinary" L-system productions are applied to all modules in parallel, Gillespie-style productions are selected according to a set of propensity functions and applied to one module per simulation step. We have illustrated the operation of Gillespie L-systems with examples progressing from a single-compartment Lotka-Volterra model to diffusion-decay, reaction-diffusion and auxin-driven morphogenetic processes. For simplicity, we have only considered linear structures (files of cells), although the formalism inherits from L-systems the capability of simulating branching structures as well. The combination of Gillespie's algorithm and L-systems makes it possible to account for the noise occurring in systems in which the number of molecules is small, and captures the variation in patterns and forms stemming from this noise. Prospective improvements and extensions include acceleration of simulations. One possibility is to limit the explicit computation of propensities to those affected by the previous simulation step, while propagating the remaining propensities intact (Gibson and Bruck, 2000). The challenge is to automatically construct the dependency graph that would identify the propensities in need of updating, given an arbitrary Gillespie L-system. Other paths to acceleration are offered by improvements to the subvolume method (Elf and Ehrenberg, 2004) and fast approximations of the Stochastic Simulation Algorithm (Gillespie, 2007; Marquez-Lago and Burrage, 2007; Lampoudi et al., 2009). Of interest is also an extension of Gillespie L-systems to two- and three-dimensional cell complexes (Desbrun et al., 2008; Lane, 2015), which would allow for a stochastic simulation of processes taking place in growing tissues.

Acknowledgements

We thank Radoslaw Karwowski for incorporating Gillespie groups into the *lpfg* simulator, Christiane Lemieux for advice in their development, Lynn Mercer for helpful comments on the manuscript, and the anonymous reviewer for the insightful review, highlighting points needing clarification or improvement. The support of this work by Discovery Grants 2014-05325 and 2019-06279 from the Natural Sciences and Engineering Research Council of Canada (PP), and the Plant Phenotyping and Imaging Research Centre / Canada First Research Excellence Fund (PP and MC) is gratefully acknowledged.

References

Abley, K, Barbier de Reuille, P, Strutt, D, Bangham, A, Prusinkiewicz, P, Marée, AF, Grieneisen, VA and Coen, E. 2013. An intracellular partitioning-based framework for tissue cell polarity in plants and animals. *Development* **140**:2061–2074.

Allen, MT, Prusinkiewicz, P and DeJong, TM. 2005. Using L-systems for modeling source-sink interactions, architecture and physiology of growing trees: the L-PEACH model. *New Phytologist* **166**:869–880.

Bar, M and Ori, N. 2014. Leaf development and morphogenesis. *Development* **141**:4219–4230.

Bilsborough, GD, Runions, A, Barkoulas, M, Jenkins, HW, Hasson, A, Galinha, C, Laufs, P, Hay, A, Prusinkiewicz, P and Tsiantis, M. 2011. Model for the regulation of *Arabidopsis thaliana* leaf margin development. *Proceedings of the National Academy of Sciences* **108**:3424–3429.

Buck-Sorlin, GH, Kniemeyer, O and Kurth, W. 2005. Barley morphology, genetics and hormonal regulation of internode elongation modelled by a relational growth grammar. *New Phytologist* **166**:859–867.

Cieslak, M, Runions, A and Prusinkiewicz, P. 2015. Auxin-driven patterning with unidirectional fluxes. *Journal of Experimental Botany* **66**:5083–5102.

Coen, E, Rolland-Lagan, AG, Matthews, M, Bangham, JA and Prusinkiewicz, P. 2004. The genetics of geometry. *Proceedings of the National Academy of Sciences* **101**:4728–4735.

Conklin, PA, Strable, J, Li, S and Scanlon, MJ. 2019. On the mechanisms of development in monocot and eudicot leaves. *New Phytologist* **221**:706–724.

Desbrun, M, Kanso, E and Tong, Y. 2008. Discrete differential forms for computational modeling. In: Bobenko, A, Schröder, P, Sullivan, J and Ziegler, G, eds. *Discrete Differential Geometry*. Basel: Birkhauser Verlag, 287–324.

Edelstein-Keshet, L. 1988. *Mathematical Models in Biology*. New York: Random House.

Elf, J and Ehrenberg, M. 2004. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Systems Biology* **1**:230–236.

Elowitz, MB, Levine, AJ, Siggia, ED and Swain, PS. 2002. Stochastic gene expression in a single cell. *Science* **297**:1183–1186.

Fowler, DR, Meinhardt, H and Prusinkiewicz, P. 1992. Modeling seashells. *ACM SIGGRAPH Computer Graphics* **26**:379–387.

Gibson, MA and Bruck, J. 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A* **104**:1876–1889.

Gierer, A and Meinhardt, H. 1972. A theory of biological pattern formation. *Kybernetik* **12**:30–39.

Gillespie, DT. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* **22**:403–434.

Gillespie, DT. 1977. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81**:2340–2361.

Gillespie, DT. 2007. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry* **58**:35–55.

Gillespie, DT, Petzold, LR and Seitaridou, E. 2014. Validity conditions for stochastic chemical kinetics in diffusion-limited systems. *The Journal of Chemical Physics* **140**:054111.

Goss, PJE and Peccoud, J. 1998. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proceedings of the National Academy of Sciences* **95**:6750–6755.

Hay, A, Barkoulas, M and Tsiantis, M. 2006. ASYMMETRIC LEAVES1 and auxin activities converge to repress BREVIPEDICELLUS expression and promote leaf development in arabidopsis. *Development* **133**:3955–3961.

Jönsson, H, Heisler, MG, Shapiro, BE, Meyerowitz, EM and Mjolsness, E. 2006. An auxin-driven polarized transport model for phyllotaxis. *Proceedings of the National Academy of Sciences* **103**:1633–1638.

Karwowski, R. 2002. *Improving the Process of Plant Modelling: The L+C Modeling Language*. PhD thesis, Universtiy of Calgary, Calgary, Canada.

Karwowski, R and Prusinkiewicz, P. 2003. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science* **86**:1–19.

Lampoudi, S, Gillespie, DT and Petzold, LR. 2009. The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *The Journal of Chemical Physics* **130**:094104.

Lane, B. 2015. *Cell complexes: The structure of space and the mathematics of modularity*. PhD thesis, University of Calgary, Calgary, Canada.

Lane, B and Prusinkiewicz, P. 2002. Generating spatial distributions for multilevel models of plant communities. In: *Proceedings of Graphics Interface 2002*. Waterloo: Canadian Human-Computer Communications Society, 69–80.

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

Lindenmayer, A. 1974. Adding continuous components to L-systems. In: Rozenberg, G and Salomaa, A, eds. *L-systems*. Berlin: Springer, **15**:53–68.

Lloyd-Price, J, Gupta, A and Ribeiro, AS. 2012. SGNS2: A compartmentalized stochastic chemical kinetics simulator for dynamic cell populations. *Bioinformatics* **28**:3004–3005.

Lotka, AJ. 1920. Undamped oscillations derived from the law of mass action. *Journal of the American Chemical Society* **42**:1595–1599.

Lu, T, Volfson, D, Tsimring, L and Hasty, J. 2004. Cellular growth and division in the gillespie algorithm. *IEE Proceedings - Systems Biology* **1**:121–128.

Marquez-Lago, TT and Burrage, K. 2007. Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *The Journal of Chemical Physics* **127**:104101.

McAdams, HH and Arkin, A. 1997. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences* **94**:814–819.

Meinhardt, H. 1982. *Models of Biological Pattern Formation*. London: Academic Press.

Meinhardt, H. 2009. *The algorithmic beauty of sea shells*, 4th edn. Berlin: Springer.

Meinhardt, H and Klingler, M. 1987. A model for pattern formation on the shells of molluscs. *Journal of Theoretical Biology* **126**:63–89.

Meinhardt, H and Klingler, M. 1988. Pattern formation by coupled oscillations: The pigmentation patterns on the shells of molluscs. In: Markus, M, MÃijller, S and Nicolis, G, eds. *From Chemical to Biological Organization*. Berlin: Springer, **39**:193–202.

Paciorek, T, Zažímalová, E, Ruthardt, N, Petrášek, J, Stierhof, YD, Kleine-Vehn, J, Morris, DA, Emans, N, Jürgens, G, Geldner, N et al. 2005. Auxin inhibits endocytosis and promotes its own efflux from cells. *Nature* **435**:1251–1256.

Prusinkiewicz, P, Cieslak, M, Ferraro, P and Hanan, J. 2018. Modeling plant development with L-systems. In: Morris, RJ, ed. *Mathematical Modelling in Plant Biology*. Cham: Springer, 139–169.

Prusinkiewicz, P, Crawford, S, Smith, RS, Ljung, K, Bennett, T, Ongaro, V and Leyser, O. 2009. Control of bud activation by an auxin transport switch. *Proceedings of the National Academy of Sciences* **106**:17431–17436.

Prusinkiewicz, P, Karwowski, R and Lane, B. 2007. The L+C plant-modelling language. In: Vos, J, Marcelis, L, De Visser, P, Struik, P and Evers, J, eds. *Functional–structural plant modelling in crop production*. Dordrecht: Springer, 27–42.

Prusinkiewicz, P and Lane, B. 2013. Modeling morphogenesis in multicellular structures with cell complexes and L-systems. In: Capasso, V, Gromov, M, Harel-Bellan, A, Morozova, N and Pritchard, LL, eds. *Pattern Formation in Morphogenesis*. Berlin: Springer, **15**:137–151.

Prusinkiewicz, P and Lindenmayer, A. 1990. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag. With Hanan J, Fracchia FD, Fowler D, de Boer MJM, and Mercer L.

Păun, G and Rozenberg, G. 2002. A guide to membrane computing. *Theoretical Computer Science* **287**:73–100.

Rao, CV, Wolf, DM and Arkin, AP. 2002. Control, exploitation and tolerance of intracellular noise. *Nature* **420**:231–237.

Runions, A, Tsiantis, M and Prusinkiewicz, P. 2017. A common developmental program can produce diverse leaf shapes. *New Phytologist* **216**:401–418.

Sachs, T. 1991. *Pattern Formation in Plant Tissues*. Cambridge: Cambridge University Press.

Scarpella, E, Marcos, D, Friml, J and Berleth, T. 2006. Control of leaf vascular patterning by polar auxin transport. *Genes & Development* **20**:1015–1027.

Smith, RS, Guyomarc'h, S, Mandel, T, Reinhardt, D, Kuhlemeier, C and Prusinkiewicz, P. 2006. A plausible model of phyllotaxis. *Proceedings of the National Academy of Sciences* **103**:1301–1306.

Spicher, A, Michel, O, Cieslak, M, Giavitto, JL and Prusinkiewicz, P. 2008. Stochastic P systems and the simulation of biochemical processes with dynamic compartments. *Biosystems* **91**:458–472.

Stundzia, AB and Lumsden, CJ. 1996. Stochastic simulation of coupled reaction-diffusion processes. *Journal of Computational Physics* **127**:196–207.

Turing, A. 1952. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B* **237**:37–52.

Vigelius, M and Meyer, B. 2012. Stochastic simulations of pattern formation in excitable media. *PLoS ONE* **7**:e42508.

Volterra, V. 1928. Variations and Fluctuations of the Number of Individuals in Animal Species living together. *ICES Journal of Marine Science* **3**:3–51.

Wang, H, Fu, Z, Xu, X and Ouyang, Q. 2007. Pattern formation induced by internal microscopic fluctuations. *The Journal of Physical Chemistry A* **111**:1265–1270.

Zažímalová, E, Petrášek, J and Benkova, E. 2014. *Auxin and its role in plant development*. Heidelberg: Springer-Verlag.