



Using cellzilla for plant growth simulations at the cellular level

Bruce E. Shapiro^{1,2*}, Elliot M. Meyerowitz³ and Eric Mjolsness⁴

¹ Department of Mathematics, California State University, Northridge, CA, USA

² Biological Network Modeling Center, Caltech, Pasadena, CA, USA

³ Division of Biology, Howard Hughes Medical Institute, Caltech, Pasadena, CA, USA

⁴ Department of Computer Science, University of California, Irvine, CA, USA

Edited by:

Hartmut Stützel, Leibniz Universität Hannover, Germany

Reviewed by:

Lars H. Wegner, Karlsruhe Institute of Technology, Germany

Winfried Kurth, Georg-August

Universität Göttingen, Germany

*Correspondence:

Bruce E. Shapiro, Department of Mathematics, California State University, 18111 Nordhoff St., Northridge, CA 91330-8313, USA
e-mail: bruce.e.shapiro@csun.edu

Cellzilla is a two-dimensional tissue simulation platform for plant modeling utilizing Cellerator arrows. Cellerator describes biochemical interactions with a simplified arrow-based notation; all interactions are input as reactions and are automatically translated to the appropriate differential equations using a computer algebra system. Cells are represented by a polygonal mesh of well-mixed compartments. Cell constituents can interact intercellularly via Cellerator reactions utilizing diffusion, transport, and action at a distance, as well as amongst themselves within a cell. The mesh data structure consists of vertices, edges (vertex pairs), and cells (and optional intercellular wall compartments) as ordered collections of edges. Simulations may be either static, in which cell constituents change with time but cell size and shape remain fixed; or dynamic, where cells can also grow. Growth is controlled by Hookean springs associated with each mesh edge and an outward pointing pressure force. Spring rest length grows at a rate proportional to the extension beyond equilibrium. Cell division occurs when a specified constituent (or cell mass) passes a (random, normally distributed) threshold. The orientation of new cell walls is determined either by Errera's rule, or by a potential model that weighs contributions due to equalizing daughter areas, minimizing wall length, alignment perpendicular to cell extension, and alignment perpendicular to actual growth direction.

Keywords: mathematical model, computational model, software, meristem, cellerator, cellzilla, wuschel, clavata

INTRODUCTION

In recent years, there has been much interest in accurate multi-scale models of morphogenesis. Due to the various levels of complexity and the wide variety of tissue, there has necessarily been a trade-off between generality and specificity; an excellent review is given by Koumoutsakos et al. (2011). Of the most interest to plant biologists, perhaps, are platforms that describe the high-level structure of complete organisms, systems, or organs such as a meristem or sepal. Very few general purpose tools exist at all, and even fewer are specific to plants. L-System based tools, the best known being L-Studio (Karwowski and Prusinkiewicz, 2004), were among the first; they are ideally suited to branching structures because they are based on formal language theory (a grammar based on axioms, a short alphabet, strings derived from that alphabet based on specific production rules that are tuned to branching). L-system rules have been embedded in higher level languages such as C [specifically, *cpfg*, (Prusinkiewicz and Lindenmayer, 1990)] to allow the encoding of models and the description of geometrical and topological relationships. The Virtual Plant's OpenAlea Platform (Pradal et al., 2008) provides a general-purpose collection of simulation modules that use Python as the primary scripting language and allows L-systems to be incorporated (Boudon et al., 2012).

Outside of plant biology, much of the interest has focused on grid-based models such as the Cellular Potts Model (CPM)

(Graner and Glazier, 1992), which discretize to the desired sub-cellular level of detail as collections of entities driven by particle-particle interactions; and finite element models (FEM) which instead discretize to a sufficiently fine web-work of straight lines under the influence of mechanical forces. Because of the molecular level of detail of these methods they are often primarily stochastic in nature (Gillespie, 1976). There has been some effort to provide general purpose platforms; examples include MCell (Stiles and Bartol, 2001), designed originally to simulate the synaptic junction; Smoldyn (Andrews, 2012), primarily emphasizing nano-scale specificity; ChemCell (Plimpton and Slepoy, 2005), which models protein networks within cells; and CompuCell3D (Cickovski et al., 2007) which focuses on cellular function.

Of particular interest is the use of rule-based models, e.g., BioNetGen (Faeder et al., 2005, 2009) and NFSIM (Sneddon et al., 2011). In a rule-based model, molecules are considered structured objects and their interactions are described by rules for transforming these objects. Rule based models are interesting because every process that occurs within a biosystem, such as chemical and physical interactions and growth, development, and cell division and death, can be described by rules. Similarly, Dynamical Grammars (Mjolsness and Yosiphon, 2006; Yosiphon, 2009; Mjolsness, 2013) define models in terms of operator algebras of stochastic processes, with simulation algorithms

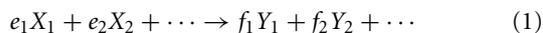
derived from the composition and expansion of time-evolution operators.

At an intermediate level of complexity there are tools that describe tissue at the multicellular level, treating each cell as a well-mixed compartment. Cells can be described either as simple point (or spherical) objects connected by breakable springs (Jönsson et al., 2004, 2005a,b; Mjølness, 2006), or at some level of geometric complexity with springs between polygonal vertices (Rudge and Haseloff, 2005; Sahlin and Johnsson, 2010), as is currently done by Cellzilla. The VirtualLeaf (Merks et al., 2011) provides an interesting hybrid that uses the CPM in combination with mechanical springs with a Markovian relaxation algorithm to describe tissue growth. Cellzilla has the distinction among these software implementations of extending the collection of Cellerator Arrows to include multicellular interactions (Shapiro et al., 2003).

MATERIALS AND METHODS

CELLERATOR REACTIONS

Interactions are described in terms of Cellerator arrows (Shapiro et al., 2003); the basic canonical form is $\{X \rightarrow Y, k\}$. When several species are interacting the arrow expression is written as



where the e_i and f_j are stoichiometries. Each reactant is converted to a single term in a differential equation according to mass action kinetics:

$$\frac{dU_j}{dt} = (f_j - e_j)kX_1^{e_1}X_2^{e_2}\dots \quad (2)$$

where e_j and f_j are the stoichiometries of species U_j in the reaction on the left and right hand side of the reaction, respectively, and k is the rate constant of the reaction. More complex expressions can be built from this canonical form to represent exact mass action descriptions of multi-species complex enzymatic reactions, as summarized in **Table 1**, and numerous regulatory approximations such as Michaelis-Menten, Hill functions, and MWC equations, as summarized in **Table 2**. Furthermore, a large number of exact enzymatic expansions have been implemented using the basic canonical form via the KMech toolbox, including BiBi, BiTer, BiUni, MulS, OrderedBiBi, OrderedBiUni, PingPong, PingPongTerTerF, PingPongTerTerR, RandomBiBi, TerBi, TerTer, UniBi, and UniUni reactions (Yang et al., 2005).

Cellzilla is implemented in Mathematica and is invoked using the standard notebook interface. The details of numerical integration are normally hidden from the user, e.g., models (such as (30)–(32)) are submitted to the kernel by the Cellzilla `Grow` command and Cellerator `run` along with a list of simulation control parameters and initial conditions. These front-end commands directly invoke Mathematica's `NDSolve`. Normally, the default solver in `NDSolve` is chosen, unless the user selects otherwise; any control option for `NDSolve` may be passed along by `Grow` command. For ordinary differential equations, `NDSolve` switches between a non-stiff Adams method and a stiff Gear Backward Differentiation formula. However, users may optionally change the parameters or choose another solver.

TISSUE DESCRIPTION

Tissues are described by a polygonal lattice, with each lattice cell representing one biological cell. The `Tissue` data structure consists of: (1) a vertex list \mathbf{V} , where each vertex \mathbf{V}_i is an (x, y) pair; (2) an edge list \mathbf{E} , where each edge $\mathbf{E}_k = (i_k, j_k)$ is a pair of integers giving indices of vertices at the endpoints of the edge; and (3) a list of cells \mathbf{C} , where each cell $\mathbf{C}_k = \{k_1, k_2, k_3, \dots\}$ is an ordered list of edge indices. Externally the tissues may be saved (either read or written) as CSV files, either as lists of vertices, edges, and cells, or in a flattened versions with the edges omitted and the cells represented as ordered sequences of vertices. Internally the edges are always reconstructed since it is more efficient computationally to always have the edge information available, but it is user taste which I/O format is used. It was decided to use the CSV format for these files because of the wide availability of parsing tools and the ease of human readability should that be necessary.

Species in different cells are referenced by an index; e.g., the reaction $X[17] \rightarrow Y[17]$ takes place in cell 17. When a reaction network is expanded in every cell in the system it is not necessary to repeat this manually, as this is done automatically. Constituents in different cells can interact in the following ways: (a) diffusion; (b) action at a distance; (c) transport across the cell wall. Each of these may be specified in the model by one of the additional Cellzilla Arrow forms listed in **Table 3**. Such interactions are allowed to depend on a function $f(i, j, k)$ that depends on the properties of the constituents of the cells (i = present cell number, j = connecting cell numbers) and cell wall (k) between cells i and j . The basic single cell models may be input and output as SBML files using the Cellerator/MathSBML extensions for SBML (Hucka et al., 2003; Shapiro et al., 2004).

Table 1 | Examples of Cellerator mass action arrow form expansion, from the base form $A \rightarrow B$.

Arrow form*	Expansion
$e_1X_1 + e_2X_2 + \dots \rightleftharpoons f_1Y_1 + f_2Y_2 + \dots$	$e_1X_1 + e_2X_2 + \dots \rightarrow f_1Y_1 + f_2Y_2 + \dots$ $f_1Y_1 + f_2Y_2 + \dots \rightarrow e_1X_1 + e_2X_2 + \dots$
$X \xrightleftharpoons{\mathcal{E}} Y$	$X + \mathcal{E} \rightleftharpoons X_{\mathcal{E}}$ $X_{\mathcal{E}} \rightleftharpoons Y + \mathcal{E}$
$X \xrightleftharpoons[\mathcal{F}]{\mathcal{E}} Y$	$X \xrightleftharpoons{\mathcal{E}} Y$ and $Y \xrightleftharpoons{\mathcal{F}} X$
$X \xrightleftharpoons{\mathcal{E}} Y$	$X + \mathcal{E} \rightleftharpoons X_{\mathcal{E}}$ $X_{\mathcal{E}} \rightleftharpoons Y_{\mathcal{E}}$ $Y_{\mathcal{E}} \rightleftharpoons Y + \mathcal{E}$
$X \rightleftharpoons^{\mathcal{E}} Y \rightleftharpoons^{\mathcal{E}} Z \dots$	$X \xrightleftharpoons{\mathcal{E}} Y, Y \xrightleftharpoons{\mathcal{E}} Z, \dots$
$X \rightleftharpoons^{\mathcal{E}}_{\mathcal{F}} Y \rightleftharpoons^{\mathcal{E}} Z \dots$	$X \xrightleftharpoons[\mathcal{F}]{\mathcal{E}} Y, Y \xrightleftharpoons[\mathcal{F}]{\mathcal{E}} Z, \dots$

*The complete syntax of an arrow form is $\{arrowform, k_1, \dots\}$ where k_1, \dots is a sequence of numeric or symbolic rate constants. The subscripts e_i and f_j are stoichiometries.

Table 2 | Cellerator user defined, regulatory, and enzymatic arrow forms.

Arrow form ^{a,b}	Typical ODE Term ^c
$\{ \mathbf{e} \cdot \mathbf{X} \Rightarrow \mathbf{f} \cdot \mathbf{Y}, \mathbf{g}(\mathbf{X}) \}$	$[\mathbf{X}_i]' = -\mathbf{e}_i \mathbf{g}(\mathbf{X}), [\mathbf{Y}_i]' = \mathbf{f}_i \mathbf{g}(\mathbf{X})$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{Hill}[v, n, K, a, \mathbf{T}] \}$	$[\mathbf{Y}]' = \frac{v[\mathcal{E}](a + \mathbf{T} \cdot [\mathbf{X}])^n}{K^n + (a + \mathbf{T} \cdot [\mathbf{X}])^n}$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{GRN}[v, \mathbf{T}, n, h] \}$	$[\mathbf{Y}]' = \frac{v[\mathcal{E}]}{1 + \exp(-h - \mathbf{T} \cdot [\mathbf{X}]^n)}$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{SSystem}[\tau, C_+, C_-, \mathbf{n}^+, \mathbf{n}^-] \}$	$[\mathbf{Y}]' = \frac{[\mathcal{E}]}{\tau} \{ C_+ \prod_i X_i^{n_i^+} - C_- \prod_i X_i^{n_i^-} \}$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{NHCA}[v, \{\mathbf{T}^+, \mathbf{T}^-\}, \mathbf{n}, m, k] \}$	$[\mathbf{Y}]' = \frac{v[\mathcal{E}] \prod_i (1 + T_i^+ [X]_i^{n_i})^m}{k \prod_i (1 + T_i^- [X]_i^{n_i})^m + \prod_i (1 + T_i^+ [X]_i^{n_i})^m}$
$\{ \{\mathbf{X}, \mathbf{Y}\} \Rightarrow \text{rational}[\mathbf{a}, \mathbf{d}, \mathbf{m}, \mathbf{n}] \}$	$[\mathbf{Y}]'^d = \frac{a_0 + \text{rest}(\mathbf{a}) \cdot [\mathbf{X}]^{\mathbf{n}}}{d_0 + \text{rest}(\mathbf{d}) \cdot [\mathbf{X}]^{\mathbf{n}}}$
$\{ \mathbf{X} \mapsto \mathbf{Y}, \text{USER}[v, \mathbf{T}, \mathbf{n}, h, f] \}$	$[\mathbf{Y}]' = v f(h - \mathbf{T} \cdot [\mathbf{X}]^{\mathbf{n}})$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{MM}[K, v] \}$	$[\mathbf{Y}]' = -[\mathbf{X}]' = \frac{v[\mathcal{E}][\mathbf{X}]}{K + [\mathbf{X}]}$
$\{ \mathbf{X} \xrightarrow{\mathcal{E}} \mathbf{Y}, \text{MWC}[k, n, c, L, K] \}$	$[\mathbf{Y}]'^e = -[\mathbf{X}]' = k[\mathcal{E}] \frac{\alpha(1 + \alpha)^{n-1} + L\alpha c(1 + \alpha c)^{n-1}}{(1 + \alpha)^n + L(1 + \alpha c)^{n-1}}$

^aThe catalyst species \mathcal{E} is optional and may be omitted.

^bBoldface quantities may be either scalars or vectors. Vectors enclosed by curly brackets. The notation $\mathbf{v} = \mathbf{p}^q$ means a vector \mathbf{v} with components $v_i = p_i^{q_i}$.

^cIf the catalyst is omitted replace $[\mathcal{E}]$ with 1.

^dIn `rational`, `rest(x)` is the vector \mathbf{x} with its first element removed.

^eIn `MWC`, $\alpha = [X]/K$.

Table 3 | Additional Cellzilla arrow forms not recognized by Cellerator.

Arrow form*	Description
$\{ \mathbf{X} \rightarrow \mathbf{X}, \text{Diffusion}[P_I, P_0] \}$	Diffusion of \mathbf{X} through the tissue. P_I is the permeability of internal cell walls; the optional P_0 is the permeability of tissue boundary cell walls. Each may be specified as $\mathbf{f}[i, j, k]$ where i, j, k are cell and wall indices.
$\{ \mathbf{X} \rightarrow \mathbf{X}, \text{Transport}[\mathbf{f}_{out}, \mathbf{f}_{in}] \}$	Controlled transport of \mathbf{X} across the cell wall.
$\{ \mathbf{X} \mapsto \mathbf{Y}, \text{IGRN}[v, \mathbf{T}, \mathbf{n}, h] \}$	$[\mathbf{Y}[j]]' = \frac{v}{1 + \exp(-h - \mathbf{T} \cdot [\mathbf{X}[i]]^{\mathbf{n}})}$
$\{ \text{cell} \rightarrow \text{cell}, \text{Grow}[\dots] \}$	Specification of cell growth parameters: Pressure, growth rate, spring constant.
$\{ \text{cell} \rightarrow \text{cell} + \text{cell}, \text{model}[\dots] \}$	Specification of cell division model and parameters.

*Except for the `IGRN` the arrows here are longer versions of the right-pointing arrows used by the canonical Cellerator mass-action expansion.

Diffusion across cell boundaries is implemented according to Fick's law, so that the flux J through any membrane (e.g., in molecules/(cm²-s)) with diffusion constant D (e.g., in cm²/s) is

$$J = -D \frac{\partial X}{\partial x} \quad (3)$$

Defining the membrane permeability as $\beta = D/\delta$ (e.g., in cm/s), where δ the membrane (or wall) thickness gives

$$J = -\beta \delta \times \frac{\Delta X}{\delta} = -\beta \Delta X \quad (4)$$

where ΔX is the concentration difference across the membrane. Let cell i have area A_i and depth d (orthogonal to the simulation);

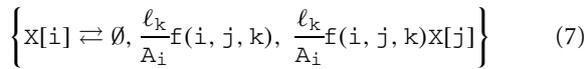
the the volume of cell j is $V_j = A_j d$, and the area of the cell wall between cell i and cell j is $\ell_k d$ where ℓ_k is the length of the wall between the cells. The flux across wall k into cell i is (for constant area):

$$J = \left(\frac{d[X_i]}{dt} \times A_i d \right) \times \left(\frac{1}{\ell_k d} \right) = \frac{A_i}{\ell_k} \frac{d[X_i]}{dt} \quad (5)$$

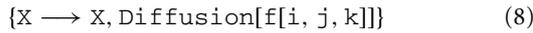
Therefore

$$\frac{d[X_i]}{dt} = \frac{\beta \ell_k}{A_i} ([X_j] - [X_i]) \quad (6)$$

Rather than implementing this equation directly, it is implemented as equivalent Cellerator reactions



where f is the input concentration-dependent permeability. Diffusion is specified to Cellzilla by incorporating arrows of the following form into the model:



where f is either a Mathematica pure function or a function that has been defined previously in the simulation.

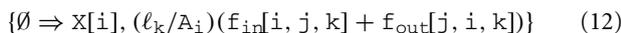
Action at a distance is technique for describing the way constituents in cell i can affect constituents in another cell j without specifying any of the intermediate reactions. We restrict action at a distance to adjacent cells with Cellerator GRN-like reactions written as $\{ X \mapsto Y, \text{IGRN}[\nu, \beta, n, h] \}$ which means that constituent X_i affects constituent Y_j in neighboring cell j according to (Mjolsness et al., 1991)

$$\frac{d[Y_j]}{dt} = \frac{\nu}{1 + e^{-h - \beta[X_i]}} \quad (9)$$

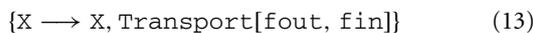
Facilitated membrane transport from cell j to cell i is described by the equations

$$\frac{d[X_i]}{dt} = \frac{\ell_k}{A_i} (f_{in}(i, j, k) - f_{out}(i, j, k) - f_{in}(j, i, k) + f_{out}(j, i, k)) \quad (10)$$

where $f_{out}(i, j, k)$ is the positive outward molecular flux from i through edge k , to j , and $f_{in}(i, j, k)$ is the positive inward molecular flux to cell i through edge k , originating from cell j . In most cases one will only want to specify one of f_{out} or f_{in} . These are implemented internally via the Cellerator reactions



Transport reactions of this sort are specified to Cellzilla by including arrows of the form



where the function f_{out} should be set to zero if only f_{in} is utilized.

GROWTH MODEL

Cellzilla implements two types of time-dependent simulations: static, and growing. In static simulations the shape of the tissue and its component parts do not change but its constituents are allowed to vary as described previously. In a growing tissue, the shape of the cells are also allowed to evolve with time. Cell growth

is described by associating a Hooke's law spring potential of the form

$$V_{ij} = \frac{1}{2} \sum k_{ij} (\delta_{ij} - \ell_{ij})^2 \quad (14)$$

with the edge connecting vertices \mathbf{x}_i and \mathbf{x}_j . Here δ_{ij} is an equilibrium length assigned to the edge, ℓ_{ij} is the actual length, and k_{ij} is a constant. When the wall is under compression (so that $\delta_{ij} > \ell_{ij}$) there will be a force, acting along the length of the edge, pushing the two vertices apart; when the wall is extended ($\delta_{ij} < \ell_{ij}$), the force will tend to pull the vertices toward one-another. The magnitude of this force is equal to the negative gradient of V_{ij} . In addition, a pressure P_a associated with each cell a is, is applied outward at each vertex. The net force on each vertex is proportional to each wall incident on that vertex and, and is split evenly between the vertices. Then equation of motion for vertex \mathbf{x}_i is then

$$\frac{d\mathbf{x}_i}{dt} = - \sum_j k_{ij} \hat{\mathbf{x}}_{ij} (\ell_{ij} - \delta_{ij}) + \frac{1}{2} \sum_{j,a} P_a \mathbf{n}_{ij,a} \ell_{ij} \quad (15)$$

where $\hat{\mathbf{x}}_{ij}$ is a unit vector pointing from \mathbf{x}_i to \mathbf{x}_j . The sum in the first term is over all the neighbors j of vertex i . In the second term, $\mathbf{n}_{ij,a}$ is an outward pointing unit-normal vector from cell a , normal to edge ℓ_{ij} , and the sum is over all neighbors j of i and over all cells a in incident at vertex i . The pressure force will cause the springs to extend, simulating cell growth. The resting length is allowed to increase linearly at a rate proportional to the extension beyond resting length,

$$\frac{d\delta_{ij}}{dt} = \mu_{ij} \Theta(\ell_{ij} - \delta_{ij}) \quad (16)$$

where

$$\Theta(x) = \frac{1}{2}(x + |x|) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

If the spring is not extended, then growth does not occur. Equations (15) and (16) capture the phenomenological behavior observed in nature by both plant and animal cells. Pressure drives cell expansion; as pressure increases, cells expand more quickly. The growth rate is limited by the spring force. The dynamics of (15) can be solved exactly for a square cell to give

$$\frac{1}{L} \frac{dL}{dt} = k \left(\frac{P}{k} - \frac{2\Delta}{L} \right) \quad (18)$$

where L is the cell perimeter and $\Delta = \sum_i \delta_i$ is the sum of the resting lengths. Consequently, when $\mu_{ij} = 0$ in (16), the sigmoidal growth pattern of plant cells is observed [e.g., when extensibility decreases linearly over time and osmotic pressure is held constant, as in Figure 1 of Lockhart (1965)]. Comparing (18) to the Lockhart equation $\ell'/\ell = \Phi(P - P_E)$ we see that the k and δ are parameters that can tuned to fit effective extensibility Φ and yield pressure P_E ; for constant P and k sigmoidal behavior can be tuned for $P < 2k$ in square cells. Additionally, allowing $\mu > 0$, along

with the spring force produces a more general growth model that is more generally applicable, not just in plant tissue, as the springs can be cut beyond a specified threshold, thereby removing cell-cell interactions (Shapiro and Mjolsness, 2001).

Let r be the index of the edge connecting vertices i and j , so that δ_r and ℓ_r are short notations δ_{ij} and ℓ_{ij} . The spring dynamics of (16) are implemented internally as

$$\left\{ \emptyset \rightarrow \mathbf{x}[i, p], \sum_{\text{Neighbors}(i)} k[r] (\mathbf{x}[j, p] - \mathbf{x}[i, p]) (1 - \delta_r/\ell_r) \right\} \quad (19)$$

where $\mathbf{x}[i, p]$ is the p th Cartesian component ($p = 1, 2$ for x or y) of vertex \mathbf{x}_i , and $k[r]$ is a spring constant whose value may depend on the properties (e.g., constituents of) the cells abutting edge r . Similarly, the growth of edge δ_r described by (16) is implemented internally as the Cellerator reactions

$$\{\emptyset \rightarrow \delta_r, \mu[p, q, r] (\ell_r - \delta_r)\} \quad (20)$$

where p and q are the indices of the cells that about edge r . The pressure force in (15) on vertex x_i due to cell a on the edge connecting vertices x_i and x_j is implemented internally as

$$\left\{ \emptyset \rightarrow \mathbf{x}[i, k], \frac{1}{2} P[a] n[a, i, j, k] \right\} \quad (21)$$

where $k = 1, 2$ (to indicate x or y Cartesian component); $P[a]$ is the pressure in cell a ; and $n[a, i, j, k]$ is the k^{th} component of a unit normal vector to edge ℓ_{ij} pointing outwards from cell a .

Cell growth is specified in a Cellzilla model by a reaction of the form



where the arguments to `Grow` specify growth parameters such as the dependence of k , P , and μ on cell constituents (see **Table 3**). Chemical concentrations change during growth in each cell occur even in the absence of reactions. If there are n molecules of X in volume V then $[X]' = (n/V)' = (Vn' - nV')/V^2 = n'/V - [X]V'/V$. The second term gives the correction in $[X]'$ due to volume changes.

CELL DIVISION

Division occurs when a cell's area passes a threshold. Upon birth, each cell is assigned a threshold that is distributed normally (with the mean μ and standard deviation σ as optional control parameters). Chemical concentrations are distributed equally between the child cells (so that the chemical amounts are proportional to cell area). A single (linear) near cell wall is placed according to one of two user-selectable model: the standard modern interpretation of (Errera, 1888) and a potential model. In the modern interpretation of Errera's rule, the shortest wall that divides the two cells in half (by area) is chosen. [Technically, this is not Errera's rule, which only defines the shape of the cell wall, once the endpoints are already know; however, the area-equalization constraint is typically added to provide this boundary condition. (Smith, 2001; Besson and Dumais, 2011; Prusinkiewicz and

Runions, 2012)] In the potential model (Shapiro et al., 2010) a function

$$V(\theta_1, \theta_2) = \sum_i w_i V_i(\theta_1, \theta_2) \quad (23)$$

is minimized over the central angles θ_1 and θ_2 . These give the central angles of the end points of the new wall measured from the cell centroid. Here \mathbf{w} is a weight vector, and V_i represents each contributor to cell division, where $i \in \{A, L, e, g\}$, as described in the below.

The area potential V_A is minimized when the cell divides in half; if the daughter cells have areas A_1 and A_2 , respectively, then we define

$$V_A = \left(\frac{A_1 - A_2}{A} \right)^2 \quad (24)$$

The function is squared to improve computational stability near the minimum (which would otherwise have a non-differentiable corner there). The disadvantage of this potential is that it does not have a unique global minimum, i.e., any line of cell division will that divides the area in half will give a value of zero. Thus, the area potential must be tempered by either an additional potential function (such as the perpendicularity and/or length potential) or an additional heuristic to select the desired minimum value that does not require a unique minimum (e.g., randomly select division direction from amongst all equivalent minima).

The length potential V_L will be minimized when the new cell wall is most closely aligned with the shortest possible diameter d_{min} that this, the shortest line segment dividing the cell passing through the cell center. If d is the length of the new cell wall, then V_L is given by

$$V_L = \frac{(d - d_{min})^2 + \epsilon_L \Delta^2}{(d + d_{min})^2} \quad (25)$$

where Δ is the shortest distance between the new wall and the cell center, and ϵ_L is a tunable parameter.

For V_e and V_g we define a perpendicularity potential $V_{\perp}(\mathbf{v})$ that is minimized when the new wall is perpendicular to particular unit vector \mathbf{v} . Let \mathbf{W} be a unit vector parallel to the new wall. Then

$$V_{\perp}(\mathbf{v}) = \mathbf{v} \cdot \mathbf{W} + \frac{\epsilon_{\perp} \Delta}{d} \quad (26)$$

where ϵ_{\perp} is a parameter. Letting \mathbf{e} be the direction of maximal cell extension and \mathbf{g} be the direction of maximal cell growth, we then define

$$V_e = V_{\perp}(\mathbf{e}) \quad (27)$$

$$V_g = V_{\perp}(\mathbf{g}) \quad (28)$$

so that V_e and V_g are minimized when \mathbf{W} is most nearly perpendicular to the directions of maximal extension and maximal cell growth, respectively. The direction of maximal extension is taken

as the unit eigenvector corresponding to the larger eigenvalue of the covariance matrix $\mathbf{M} = (\mathbf{X}^T \mathbf{X}) / (n - 1)$ where n is the number of cell vertices; and $\mathbf{X} = [\mathbf{x} - \mathbf{x}_c | \mathbf{y} - \mathbf{y}_c]$, where \mathbf{x} and \mathbf{y} are column vectors of cell vertex coordinates $\{(x_1, y_1), \dots, (x_n, y_n)\}$ and \mathbf{x}_c and \mathbf{y}_c is their mean. The direction of instantaneous maximal growth is found in the same manner, using the velocities of the vertices. The covariance matrix \mathbf{M} is calculated using the Mathematica function `Covariance`.

Cell division is specified in a Cellzilla model with the arrow

$$\{\text{cell} \rightarrow \text{cell} + \text{cell}, \text{model}[\dots]\} \quad (29)$$

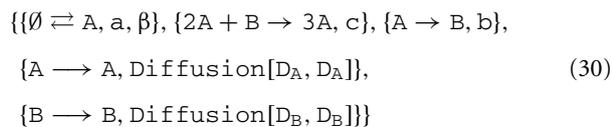
where `model` is either `ErreraModel` or `Potential`, and the arguments specify the threshold variable, mean, standard deviation, and weight vector (for the potential model). A more generalized version of this notation has been introduced by Yosiphon (2009).

RESULTS

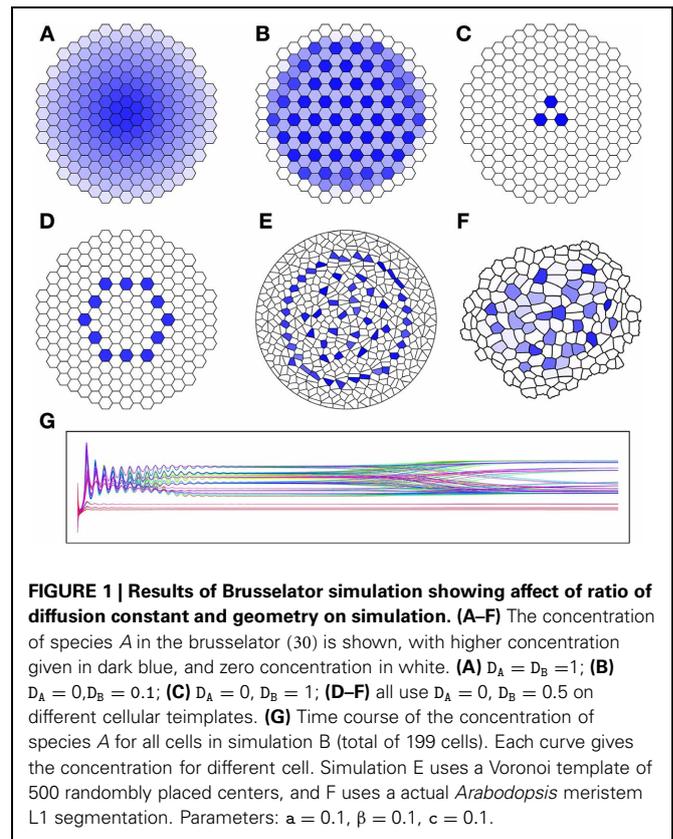
TEMPLATES AND THE BRUSSELATOR

Cellzilla has variety of shapes that can be used for basic simulations such as rectangular and hexagonal arrays, as well as circular, semicircular and parabolic templates that can be populated with randomly placed Voronoi centers. Alternatively the user can supply a template of his or her own consisting either of cell centers (in which case the walls will be interpolated with a Voronoi algorithm) or cell walls as described previously. Here we present the use of several of these templates (hexagonal, Voronoi, and user-supplied) to implement a common reaction-diffusion system.

The Brusselator (Prigogine and Lefever, 1968) is frequently cited in mathematical modeling because it consists of a system of chemical reactions that in the appropriate parameter regime will maintain sustained oscillations. When combined with diffusion such a system can also be used to establish a wide variety of interesting patterns such as stripes, spirals, and central maxima. The establishment of these different patterns depends on the choice of system geometry, boundary conditions, and parameter values. A diffusible Brusselator is easily implemented in Cellzilla with

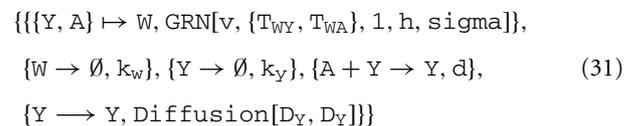


where a, β, b, c, D_A and D_B are tunable parameters. As illustrated in **Figure 1**, the ratio of the diffusion constants will change the number of maxima achieved. In the second row of the figure we see that the geometry is also significant. While the qualitative features of each of the results D-F are identical, their symmetry becomes more and more broken as the symmetry of the template becomes lost. With the exception of the diffusion constants, all other parameters were identical through each of these simulations. We are particularly interested in the parameter set shown in **Figure 1A**, because it can be used as described in the following section to establish an organizing center for simulation of the WUS/CLV network.



ESTABLISHMENT OF STEM CELL NICHE

In Jönsson et al. (2005a,b) we presented a predictive model of feedback interaction between the Wuschel (WUS) and Clavata3 (CLV) signals in the shoot apical meristem. This simplified model was able both to organize the WUS expression domain and to predict the reorganization due to the removal of the CLV signal from the WUS domain as seen in experiments when cells are ablated. This model uses a reaction-diffusion mechanism to induce WUS; the pattern is induced by a Brusselator. The original model relies on a diffusible parameter Y that is produced only in the L1 layer of a slice. We present an implementation in which our slice has the L1 layer omitted, and replace this with a boundary condition in which Y is held fixed, and allowed to diffuse inward. Assuming the Brusselator is implemented by (30), the Cellzilla network for the WUS activator is given by

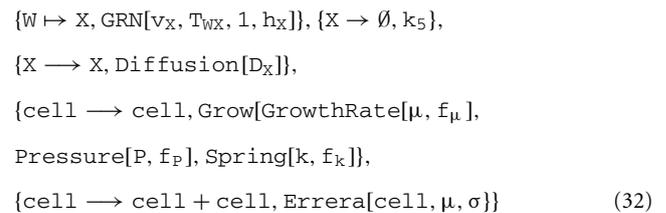
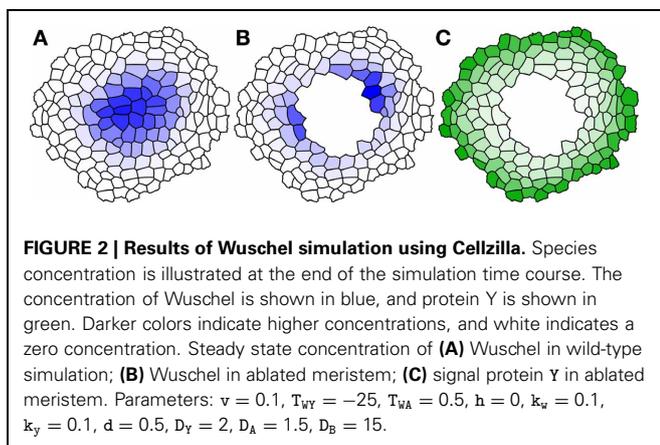
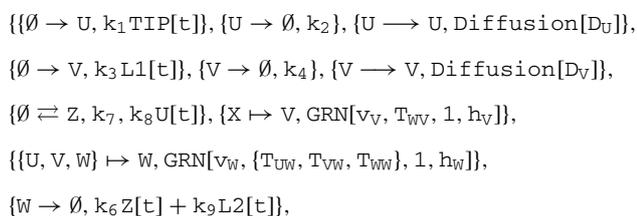


where $v, T_{WY}, T_{WA}, h, k_w, k_y$ and D_Y are tunable parameters, and the control word `sigma` tells Cellerator to replace the usual logistic control function $f(x) = 1/(1 + e^{-x})$ with $f(x) = (1 + 1/\sqrt{1 + x^2})/2$ (in fact, any monotonic increasing saturating function would work). The results illustrated in **Figure 2** show that both the original central maximum (in wild type) and dual, smaller maxima result (in the ablation experiment) as modeled

previously. In addition, we show the steady state distribution of the constituent Y in **Figure 2C**, illustrating how it forms a ring in the outer cells abutting L1 and decreasing inward, as desired.

GROWTH INDUCED BY ORGANIZING CENTER

The Brusselator is a useful mathematical/computational artifice for establishing patterns that can be otherwise studied but its biological meaning becomes lost if the variables in the equations do not have biological analogues that are present in the actual tissue. It is more meaningful if the organizing tissue can be established based on specific networks whose constituents have been observed and whose interactions are believed to be present, although this may at times be more computationally intensive. For example, Nikolaev (Nikolaev et al., 2007, 2013) has shown that a combination of reaction-diffusion and feedback in the WUS/CLV network is sufficient to establish a stem cell niche. In a one dimensional dynamic model including cell division, Chickarmane et al. (2012) has shown that negative feedback between WUS and cytokinin synthesis may be sufficient for maintenance of this niche as the tissue grows. Here we present for illustrative purposes of Cellzilla capability a simplified version of a Chickarmane-inspired model, in which two diffusible species are used to establish the pattern: U (e.g., that may be part of the cytokinin network), which is produced only in the cells at the tip of the meristem; and a second species V that is produced in the L1 layer (e.g., that may produced as part of the CLV network). The species that represents the organizing center is called W ; U and V then activate and repress W , respectively, while W is self-activating, perhaps through an intermediate. Positive feedback of W onto V is provided by a third diffusible species X ; and the epidermis is impermeable to U , V and W . Finally, the constitutive degradation of W is slightly enhanced in the L2 layer, but occurs everywhere. The network is

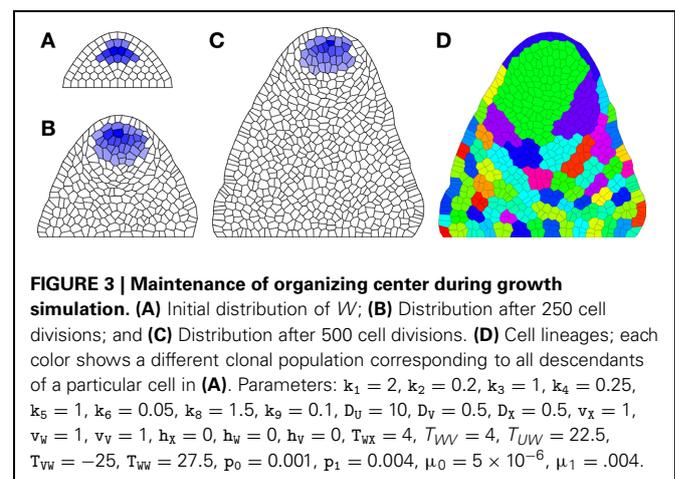


where $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, h_V, h_W, v_V, v_W, T_{WX}, T_{UW}, T_{VW}, T_{WW}, D_U, D_V$, and D_X are tunable parameters; $L1[t]$, $L2[t]$, and $\text{Tip}[t]$ are built in indicator functions for these cell locations; and the functions f_P and f_μ describe pressure and growth feedback; e.g., $P[i] = p_0 + p_1 W[i]$ and $\mu[j] = \mu_0 + \mu_1 (W[i] + W[j])$ (where p_0, p_1 , and μ_0 are tunable control parameters). The user would type these function definitions either in the lines before the model or in line as lambda functions in place of the function references in the model. Simulation results are shown in **Figure 3**. As seen in **Figures 3A–C**, the stem cell niche is maintained through at least 500 cell divisions.

DISCUSSION

We have illustrated that meaningful quantitative results for plant morphodynamics can be obtained using a simple polygonal tissue model coupled with a spring growth equations. In particular, our implementation utilizes and extends an existing arrow-based computational framework that is easy and intuitive to use. This framework is built within a standardized computer algebra system (Mathematica) that is widely available and provides access to a wide selection of analytical tools. In addition we believe that our framework is generalizable and extensible to the wider world of rule-based systems. While more detailed particle-based or molecular dynamics frameworks will certainly produce more accurate frameworks their ultimate extensibility is limited by CPU availability and time constraints. Our implementation provides a useful platform for rapid model development and testing that can easily be transformed to one of the more detailed frameworks, if desired, once suitable results are obtained.

Cellzilla can be used for 2D simulations of plant tissues at the multicellular level. Because models can be rapidly built and



tested it allows one to quickly test developmental models both in steady-state and during growth. However, several improvements are planned for future versions. In the current version, only simple transport and diffusion are considered; however, in many cell types, not just plant cells, osmotic and electrical gradients are significant. We plan to implement rules to incorporate both features in future releases. Furthermore, with the addition of osmotic models and pressure gradients we plan to add additional plant-specific growth models (Lockhart, 1965; Ortega, 1985; Geitmann and Ortega, 2009). These can be optionally used in place of the phenomenologically-based spring-based growth model. Additions to the cell division model will be included. For example, the constraints used in the Errera implementation of linear walls and equal areas can be relaxed to quadratic and circular arcs, and the areas can be randomized in a cloud about equality.

A three-dimensional model is also planned, although we expect that this will have significantly greater computational demands. To avoid mathematically unstable solutions in three dimensions the spring model will need to be modified or replaced,

most likely with an elastic dynamics model incorporating pressure and stress tensors. An alternative is the use of triangular springs (Delingette, 2008).

DATA SHARING

All of the software described here is open source Mathematica (GPL license) and freely downloadable from launchpad at <https://launchpad.net/cellerator>. The software is fully documented and all examples are available at the project website <https://www.cellzilla.info>.

ACKNOWLEDGMENTS

We would like to thank Sergey Nikolaev for valuable suggestions. This work was supported by grants from the Beckman Institute, the Division of Biology, and the Provost's Office at Caltech; a gift from Peter Cross to Caltech; by the Howard Hughes Medical Institute and the Gordon and Betty Moore Foundation (through Grant GBMF3406) to Elliot M. Meyerowitz; and by NIH Grant R01 GM086883 to Eric D. Mjolsness at UCI.

REFERENCES

- Andrews, S. (2012). "Spatial and stochastic cellular modeling with the smoldyn simulator," in *Bacterial Molecular Networks: Methods and Protocols: Methods in Molecular Biology*, eds J. van Helden, A. Toussaint, and D. Thieffry (New York, NY: Springer), 519–540.
- Besson, S., and Dumais, J. (2011). Universal rule for the symmetric division of plant cells. *Proc. Natl. Acad. Sci. U.S.A.* 108, 6294–6299. doi: 10.1073/pnas.1011866108
- Boudin, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., and Godin, C. (2012). L-Py: an L-system simulation framework for modeling plant architecture development based on a dynamic language. *Front. Plant Sci.* 3:76. doi: 10.3389/fpls.2012.00076
- Chickarmane, V. S., Gordon, V., Tarr, S. P., Heisler, M. G., and Meyerowitz, E. M. (2012). Cytokinin signaling as a positional cue for patterning the apicalbasal axis of the growing arabidopsis shoot meristem. *Proc. Natl. Acad. Sci. U.S.A.* 109, 4002–4007. doi: 10.1073/pnas.1200636109
- Cickovski, T., Aras, K., Swat, M., Merks, R., Glimm, T., George, H., et al. (2007). From genes to organisms via the cell: a problem-solving environment for multicellular development. *Comput. Sci. Eng.* 9, 50–60. doi: 10.1109/MCSE.2007.74
- Delingette, H. (2008). Triangular springs for modeling nonlinear membranes. *IEEE Trans. Vis. Comput. Graph.* 14, 239–241. doi: 10.1109/TVCG.2007.70431
- Errera, L. (1888). Ber zellformen und seifenblasen *Bot. Centralbl.* 34, 395–398. doi: 10.1002/cplx.20074
- Faeder, J. R., Blinov, M. L., Goldstein, B., and Hlavacek, W. S. (2005). Rule-based modeling of biochemical networks. *Complexity* 10, 22–39. doi: 10.1002/cplx.20074
- Faeder, J. R., Blinov, M. L., and Hlavacek W. S. (2009). "Rule based modeling of biochemical systems with BioNetGen," in *Systems Biology*, ed I. V. Maly (New York, NY: Springer), 113–167.
- Geitmann, A., and Ortega, J. K. E. (2009). Mechanics and modeling of plant cell growth. *Trends Plant Sci.* 14, 1360–1385. doi: 10.1016/j.tplants.2009.07.006
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 22, 403–434. doi: 10.1016/0021-9991(76)90041-3
- Graner F., and Glazier J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended potts model. *Phys. Rev. Lett.* 69, 2013–2016. doi: 10.1103/PhysRevLett.69.2013
- Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 513–523. doi: 10.1093/bioinformatics/btg015
- Jönsson, H., Heisler, M., Reddy, G. V., Agrawal, V., Gor, V., Shapiro, B. E., et al. (2005a). Modeling the organization of the wuschel expression domain in the shoot apical meristem. *Bioinformatics* 21, i232–i240. doi: 10.1093/bioinformatics/bti1036
- Jönsson, H., Heisler, M., Shapiro, B. E., Meyerowitz, E. M., and Mjolsness, E. (2005b). An auxin-drive polarized transport model for phyllotaxis. *Proc. Natl. Acad. Sci. U.S.A.* 103, 1633–1638. doi: 10.1073/pnas.0509839103
- Jönsson, H., Shapiro, B. E., Meyerowitz, E. M., and Mjolsness, E. (2004). "Modeling plant development with gene regulation networks including signaling and cell division," in *Bioinformatics of Genome Regulation and Structure*, eds R. Hofstaedt and N. Kolchanov (New York, NY: Kluwer Publications), 311–318.
- Karwowski, R., and Prusinkiewicz, P. (2004). "The L-System-based plant-modeling environment L-Studio 4.0," in *Presented at 4th International Workshop on Functional and Structural Plant Models* (Montpellier).
- Koumoutsakos, P., Bayati, B., Milde, F., and Tauriello, G. (2011). Particle simulations of morphogenesis. *Math. Models Methods Appl. Sci.* 21(Suppl.), 995–1006. doi: 10.1142/S021820251100543X
- Lockhart, J. (1965). An analysis of irreversible plant cell elongation. *J. Theor. Biol.* 8, 264–275. doi: 10.1016/0022-5193(65)90077-9
- Merks, R. M. H., Guravage, M., Inze, D., and Beemster, G. T. S. (2011). Virtualleaf: an open-source framework for cell-based modeling of plant tissue growth and development. *Plant Physiol.* 155, 656–666. doi: 10.1104/pp.110.167619
- Mjolsness, E. (2006). The growth and development of some recent plant models: a viewpoint. *J. Plant Growth Regul.* 25, 270–277. doi: 10.1007/s00344-006-0069-7
- Mjolsness, E. D. (2013). Time-ordered product expansions for computational stochastic systems biology. *Phys. Biol.* 10:035009. doi: 10.1088/1478-3975/10/3/035009
- Mjolsness, E. D., Sharp, D. H., and Reinist, J. (1991). A connectionist model of development. *J. Theor. Biol.* 152, 429–453. doi: 10.1016/S0022-5193(05)80391-1
- Mjolsness, E. D., and Yosiphon, G. (2006). Stochastic process semantics for dynamical grammars. *Ann. Math. Artif. Intell.* 47, 329–395. doi: 10.1007/s10472-006-9034-1
- Nikolaev, S. V., Penenko, A. V., Lavreha, V. V., Mjolsness, E. D., and Kolchanov, N. A. (2007). A model study of the role of proteins CLV1, CLV2, CLV3, and WUS in regulation of the structure of the shoot apical meristem. *Russ. J. Dev. Biol.* 38, 383–388. doi: 10.1134/S1062360407060069
- Nikolaev, S. V., Zubairova, U. S., Penenko, A. V., Mjolsness, E. D., Shapiro, B. E., and Kolchanov, N. A. (2013). Model of structure regulation of stem cell niche in shoot apical meristem of *Arabidopsis thaliana* (in Russian). *Doklady Akademii Nauk* 453, 336–338.
- Ortega, J. K. E. (1985). Augmented growth equation for cell wall

- expansion. *Plant Physiol.* 79, 318–320. doi: 10.1104/pp.79.1.318
- Plimpton, S. J., and Slepoy, A. (2005). Microbial cell modeling via reacting diffusing particles. *J. Phys.* 16, 305–309. doi: 10.1088/1742-6596/16/1/042
- Pradal, C., Dufour-Kowalski, S., Boudon, F., Fournier, C., and Godin, C. (2008). Openalea: a visual programming and component-based software platform for plant modeling. *Funct. Plant Biol.* 35, 751–760. doi: 10.1071/FP08084
- Prigogine, L., and Lefever, R. (1968). Symmetry breaking instabilities in dissipative systems. II. *J. Chem. Phys.* 48, 1695–1700. doi: 10.1063/1.1668896
- Prusinkiewicz, P., and Lindenmayer, A. (1990). *Algorithmic Beauty of Plants*. New York, NY: Springer Verlag.
- Prusinkiewicz, P., and Runions, A. (2012). Computational models of plant development and form. *New Phytol.* 193, 549–569. doi: 10.1111/j.1469-8137.2011.04009.x
- Rudge, T., and Haseloff, J. (2005). A computational model of morphogenesis in plants. *Adv. Artif. Life* 3630, 78–87.
- Sahlin, P., and Jönsson, H. (2010). A modeling study of how cell division affects properties of epithelial tissues under isotropic growth. *PLoS ONE* 5:e11750. doi: 10.1371/journal.pone.0011750
- Shapiro, B. E., Heisler, M., Tobin, C., Cunha, A., Davis, A., Mjolsness, E. D., et al. (2010). “Using geometric markers to predict the cell division plane in meristem cells,” in *Proceedings of the 6th International Workshop on Functional-Structural Plant Models*, eds T. DeJong and D. Da Silva (Davis, CA: University of California), 144–146.
- Shapiro, B. E., Hucka, M., Finney, A., Doyle, J. (2004). MathSBML: a package for manipulating SBML based biological models. *Bioinformatics* 20, 2829–2831. doi: 10.1093/bioinformatics/bth271
- Shapiro, B. E., Levchenko, A., Meyerowitz, E. M., Wold, B. J., and Mjolsness, E. D. (2003). Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. *Bioinformatics* 19, 677–678. doi: 10.1093/bioinformatics/btg042
- Shapiro, B. E., and Mjolsness, E. D. (2001). “Developmental simulations with cellerator,” in *International Conference on Systems Biology* (Pasadena, LA).
- Smith, L. G. (2001). Plant Cell Division: Building Walls in the Right Places. *Nat. Rev.* 2, 33–39. doi: 10.1038/35048050
- Sneddon, M. W., Faeder, J. R., Emonet, T. (2011). Efficient modeling, simulation and coarse graining of biological complexity with nfsim. *Nat. Methods* 8 177–183. doi: 10.1038/nmeth.1546
- Stiles, J. R., and Bartol, T. M. (2001). “Monte carlo methods for simulating realistic synaptic microphysiology using MCell,” in *Computational Neuroscience: Realistic Modeling for experimentalists*, ed E. De Schutter (Boca Raton, FL: CRC Press), 87–127.
- Yang, C.-R., Shapiro, B. E., Mjolsness, E. D., and Hatfield, G. W. (2005). An enzyme mechanism language for the mathematical modeling of metabolic pathways. *Bioinformatics* 21, 774–780. doi: 10.1093/bioinformatics/bti068
- Yosiphon, G. (2009). *Stochastic Parameterized Grammars: Formalization, Inference, and Modeling Applications*. PhD Thesis, UC Irvine.
- Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 May 2013; accepted: 26 September 2013; published online: 16 October 2013.

Citation: Shapiro BE, Meyerowitz EM and Mjolsness E (2013) Using cellzilla for plant growth simulations at the cellular level. *Front. Plant Sci.* 4:408. doi: 10.3389/fpls.2013.00408

This article was submitted to *Plant Biophysics and Modeling*, a section of the journal *Frontiers in Plant Science*.

Copyright © 2013 Shapiro, Meyerowitz and Mjolsness. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.