Developmental Simulations with Cellerator

Bruce E. Shapiro* and Eric D. Mjolsness

Machine Learning Systems Group, Jet Propulsion Laboratory, California Institute of Technology

*Correspondence: Mail Stop 126-347, 4800 Oak Grove Drive, Pasadena, CA 91109. email: bshapiro@jpl.nasa.gov. Machine Learning Web Page: http://www-aig.jpl.nasa.gov/public/mls/

What is Cellerator?

Cellerator is a *Mathematica* package designed to facilitate biological modeling via automated equation generation. Biochemical reactions are specified with an arrow-based notation, and are automatically translated into the appropriate ordinary differential equations. The implementation assumes that there is a one-to-one relationship between each class of interactions in a signal transduction network and the corresponding formal (i.e., mathematical) description of that interaction. Cellerator represents the nodes in signal transduction networks with variables (e.g., chemical concentrations) and links with arrows. A wide variety of biologically-based interactions can be described by Cellerator arrows. These interactions can be selected by clicking in a palette, and can also be entered manually.

The Cellerator arrow $S \overset{E}{\rightleftharpoons} P$, for example, describes a catalytic reaction in which species E facilitates the conversion of S into P. This is translated into lists of biochemical reactions as well as systems of ordinary differential equations. The user can edit either set of equations before proceeding. Output can be provided in a variety of forms: as *Mathematica* equations, SBML, C, FORTRAN, HTML, MATHML, or XML. The user also has the option of numerically integrating the system (using NDSolve) and plotting the concentration profiles, as illustrated in the figure below.

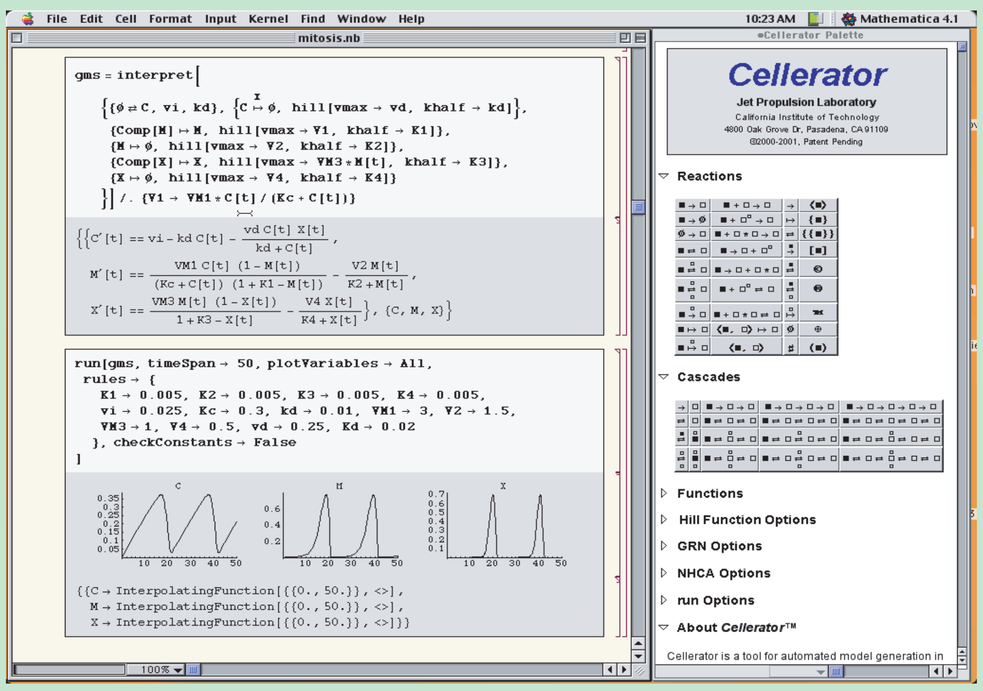
```
<sbml level="1" version="1">
 <model name="model1">
  <notes>
   Generated by Cellerator version 1.0803
   Mathematica Version used: 4.1 for Power Macintosh (November 2, 2000)
   This file was written: 15 October 2001 11:26:10
   reference: http://www.cds.caltech.edu/erato
  </notes>
  <listOfCompartments>
   <compartment name="cell1" />
  </listOfCompartments>
  <listOfSpecies>
   <specie name="C"
        compartment="cell1"
        initialAmount="0" />
   <specie name="M"
        compartment="cell1"
        initialAmount="0" />
   <specie name="X"
        compartment="cell1"
        initialAmount="0" />
  </listOfSpecies>
  <listOfRules>
   <parameterRule name="K1" formula="0.005" />
   <parameterRule name="K2" formula="0.005" />
   <parameterRule name="K3" formula="0.005" />
   <parameterRule name="K4" formula="0.005" />
   <parameterRule name="vi" formula="0.025" />
   <parameterRule name="Kc" formula="0.3" />
   <parameterRule name="kd" formula="0.01" />
   <parameterRule name="VM1" formula="3" />
   <parameterRule name="V2" formula="1.5" />
   <parameterRule name="VM3" formula="1" />
   <parameterRule name="V4" formula="0.5" />
   <parameterRule name="vd" formula="0.25" />
   <parameterRule name="Kd" formula="0.02" />
   <specieConcentrationRule specie="C"
                type="rate"
                formula="-(C*kd) + vi - (C*vd*X)/(C + kd)" />
   <specieConcentrationRule specie="M"
                type="rate"
                formula="-((M*V2)/(K2 + M)) +
                (C*(1 - M)*VM1)/((C + Kc)*(1 + K1 - M))" />
   <specieConcentrationRule specie="X"
                type="rate"
                formula="(M*VM3*(1 - X))/(1 + K3 - X) - (V4*X)/(K4 + X)" />
  </listOfRules>
 </model>
</sbml>
```

This SBML was generated by Cellerator for the minimal mitotic oscillator illustrated below.

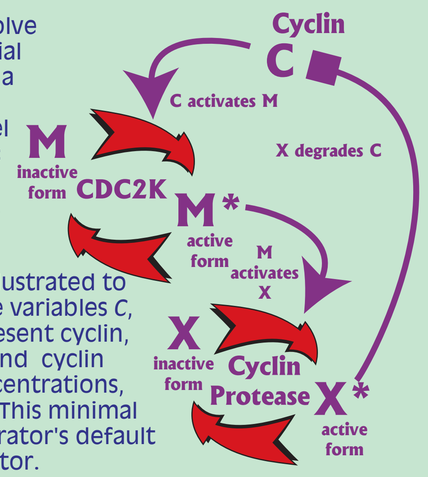
reference: http://www.cds.caltech.edu /erato

Cellerator Arrows

Syntax	Interpretation
$\{S \to P, k\}$	$[S]' = -[P]' = -k[S]$
$\{A+B \to C, k\}$	$[A]' = [B]' = -[C]' = -k[A][B]$
$\{A+nB \to C, k\}$	$[A]' = [B]' = -[C]' = -k[A][B]^n$
$\{A \rightleftharpoons B, k_f, k_r\}$	$[A]' = -[B]' = -k_f[A] + k_r[B]$
$\{A+B \rightleftharpoons C, k_f, k_r\}$	$[A]' = [B]' = -[C]' = -k_f[A][B] + k_r[C]$
$\{\Delta \to A, k\}$	$[A]' = k$
$\{A \to \Delta, k\}$	$[A]' = -k[A]$
$\{S \overset{E}{\rightleftharpoons} P, a, d, k\}$	$[S]' = -a[E][S] + d[S]$ $[P]' = k[SE]$ $[E]' = -[SE]' = -a[E][S] + (d+k)[SE]$
$\{S \overset{E}{\underset{F}{\rightleftharpoons}} P, a, d, k, a_1, d_1, k_1\}$	$[S]' = k_1[PF] - a[E][S] + d[ES]$ $[P]' = -a_1[F][P] + d_1[PF] + k[SE]$ $[E]' = -[SE]' = -a[E][S] + (d+k)[SE]$ $[F]' = -[PF]' = -a_1[F][P] + (d_1+k_1)[PF]$
$\{S \overset{E}{\Rightarrow} P, k\}$	$[S]' = -[P]' = -k[E][S]$
$\{S \overset{E}{\mapsto} P, k\}$	$[S]' = -[P]' = \dfrac{(k+\nu[E])[S]^n}{k^n + [S]^n}$



A screen view of Cellerator being used to generate and numerically solve the differential equations for a minimal cascade model of the mitotic oscillator is shown to left. The oscillator model being simulated is illustrated to the right. The variables *C*, *M*, and *X* represent cyclin, CDC2 kinase, and cyclin protease concentrations, respectively. This minimal model is Cellerator's default mitotic oscillator.



Model Reference : A. Goldbeter, A minimal cascade model for the mitotic oscillator, *PNAS* 88:9107-11,1991.

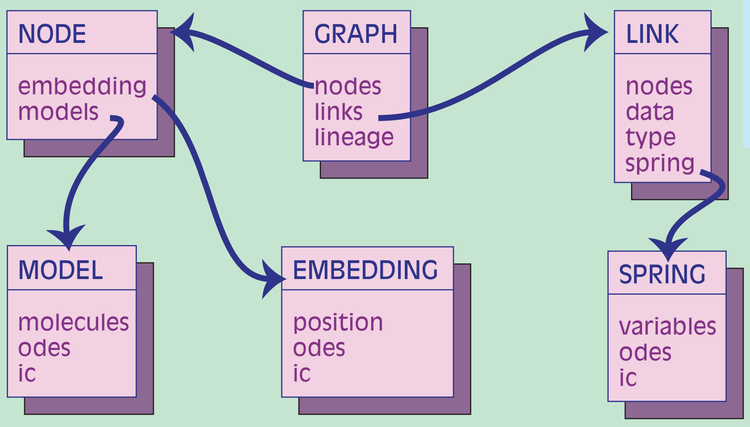
Transcriptional Regulation

Syntax	Interpretation
$\{A_1 \mapsto B, hill[options]\},\ldots,$ $\{A_p \mapsto B, hill[options]\}$ (*options* specifies r_0, r_1, ν_i, n)	$[B]' = r_0 + \dfrac{\left(r_1 + \sum_{i=1,..,p} \nu_i[A_i]\right)^n}{K^n + \left(r_1 + \sum_{i=1,..,p} \nu_i[A_i]\right)^n}$
$\{A_1 \mapsto B, grn[options]\},\ldots,$ $\{A_p \mapsto B, grn[options]\}$ (*options* specifies R, T_i, h_i, n_i)	$[B]' = \dfrac{R}{1 + exp\left(-\sum_{i=1,..,p} T_i[A_i]^{n_i} + h_i\right)}$
$\{A_1 \mapsto B, nhca[options]\},\ldots,$ $\{A_p \mapsto B, nhca[options]\}$ (*options* specifies T_i, n_i, m, k)	$[B]' = \dfrac{\prod_{i=1,..,p}\left(1 + T_i^+ A_i^{n_i}\right)^m}{k\prod_{i=1,..,p}\left(1 + T_i^- A_i^{n_i}\right)^m + \prod_{i=1,..,p}\left(1 + T_i^+ A_i^{n_i}\right)^m}$

Graph Representation of Multicellular Structures

A growing organism (or more likely, selected tissue within a growing organism) is represented by a graph data structure. In our usage, a graph is composed of three parts: a list of *nodes,* a list of *links,* and a *lineage tree*.

Nodes represent cells; links represent cell-cell interactions; and the lineage tree records the cell's family tree. The overall object hierarchy is illustrated below.



Domains and Fields

Domains and *Fields* provide Cellerator with an object-oriented technique to implement organisms as graph.

Domains represent objects. For example, an *intDomain* is a set of integers. Cellerator defines domains for each of the boxes in the object hierarchy illustrated on the left, *e.g.*, there is a *graphDomain (*an example is shown below to the right*)* a *modelDomain*, *etc.*

Fields represents functions that map domains to real numbers. *Fields of domains* are functions that map domain elements to domains, *e.g.*, the *neighbor field of domains* maps a cell to all of its neighbor cells. This could be used, for example, to calculate the total electrostatic potential acting on a cell due to its near neigbors: it is the sum of $V(x,y)$ over the neighbor field of x, where $V(x,y)$ is the potential between cells x and y.

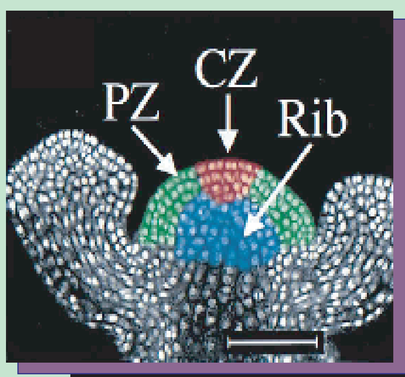
Cell Birth: Variable Structure Systems

Cell birth (death) changes the number of variables -- and hence the number of ODEs -- in the system. Changing spatial and geometric relationships cause individual equations to change their form. We assume that birth (death) is triggered when the concentration of some (or several) molecular species pass a threshold, such as in the minimal mitototic oscillator. When this happens, numerical integration is stopped and the system is adjusted accordingly.

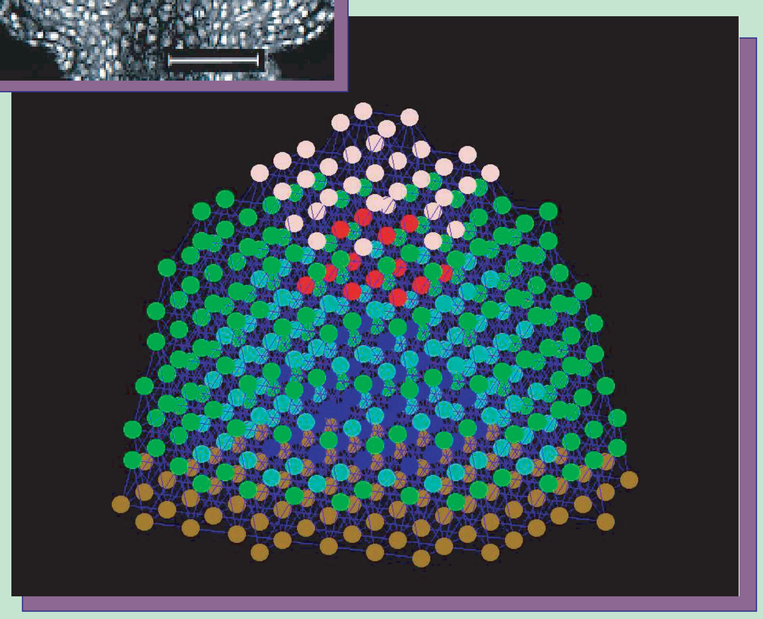
Since not all differential equation solvers allow stopping conditions we include a pair of flag variables $\{y, z\}$ for each concentration x that may cross a threshold T:

$$y' = \Theta(x - T),\ y(0) = 0$$
$$z' = \Theta(y),\ z(0) = 0$$

where Θ is the unit step function. Then y increases linearly whenever x is above (below) threshold. Even if x crosses back below (above) threshold, y remains positive. Since z increases linearly from the time that x first passes threshold, it measures total elapsed time since first threshold passage. The largest z tells us which cell passed threshold first.



Left: A laser scanning confocal microscope optical section of the shoot apical meristem (SAM) and adjacent floral meristems of wild-type Arabidopsis, stained with propidium iodide to show nuclei, colored to show typical SAM zonation. CZ, central zone; PZ, peripheral zone; Rib, rib meristem. The scale bar is 50 microns. Image provided courtesy of Elliot Meyerowitz, Caltech Division of Biology.



Left: Initial Conditions for a 424 cell Cellerator SAM simulation, with the colors indicating different cell types. This image was generated by Cellerator.

Right: A typical *graphDomain* for a single cell organism and the minimal developmental system.

```
graphDomain[nodes → {nodeDomain[
    embedding → embeddingDomain[position → {x[1], y[1], z[1]},
      odes → {x[1]'[t] == 0, y[1]'[t] == 0, z[1]'[t] == 0},
      ic → {x[1][0] == 0.668132, y[1][0] == 0.700718,
        z[1][0] == 0.342056}, time → 0],
    models → {modelDomain[molecules → {C[1], M[1], X[1]},
      odes → {C[1]'[t] == 0.025 - 0.01 C[1][t] -
          0.25 C[1][t] X[1][t] / (0.01 + C[1][t]), M[1]'[t] ==
          3 C[1][t] (1 - M[1][t]) / ((0.3 + C[1][t]) (1.005 - M[1][t])) - 1.5 M[1][t] / (0.005 + M[1][t]),
        X[1]'[t] == M[1][t] (1 - X[1][t]) / (1.005 - X[1][t]) - 0.5 X[1][t] / (0.005 + X[1][t])},
      ic → {C[1][0] == 0.228411, M[1][0] == 0, X[1][0] == 0},
      time → 0], modelDomain[molecules → {splv[1], tspl[1]},
      odes → {splv[1]'[t] == UnitStep[-0.65 + M[1][t]],
        tspl[1]'[t] == UnitStep[-1. × 10^-8 + splv[1][t]]},
      ic → {splv[1][0] == 0, tspl[1][0] == 0}, time → 0],
    modelDomain[molecules → {mass[1]},
      odes → {mass[1]'[t] == mu mass[1][t]},
      ic → {mass[1][0] == 1}, time → 0]},
    nodeData → {birth → 0}, nodeType → Cell]},
 links → {}, lineage →
  tree[
   1]]
```

"Spring" objects associated with each link are used to optimize the position of the nodes after each growth step. The spring "length" gives the desired separation between cell centers. After each step the potential is minimized (e.g. by gradient descent or simulated annealing) to determine the cellular position. This potential gets "turned off" when the actual cell to cell separation exceeds a maximum desired interaction distance.