Implementation of spatial simulator and its SBML support

4th Sep. 2011
Tatsuhiro Matsui, Noriko Hiroi, Akira Funahashi
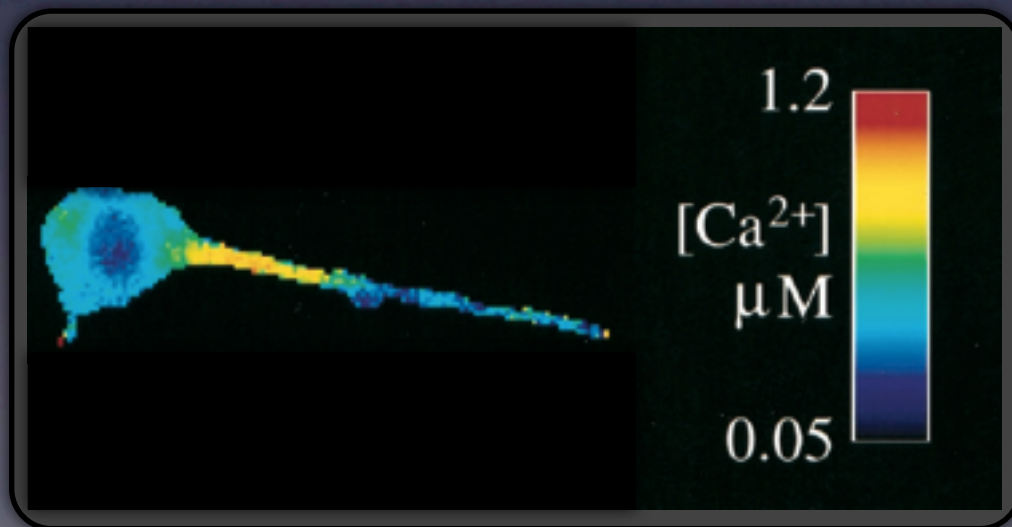**Keio University**

Keio University
1858
CALAMVS
GLADIO
FORTIOR

Motivation

- To understand the spatial dynamics and localization of molecules inside cell

- Improvement of experimental technology that enables to obtain quantitative data including spatial information

$[Ca^{2+}]$
μM

1.2

0.05

Charles C. Fink, Boris Slepchenko, Ion I. Moraru, James Watras, James C. Schaff, and Leslie M. Loew,
Biophysical Journal, 2000

Goal

- Implement a spatial model simulator which supports SBML Spatial extension[1]

[1] http://ntcnp.org/twiki/bin/view/VCell/SpatialSBML
    proposed in 2010.10.8

Current status

- Simulate an SBML model which has Advection–Diffusion equation

  – Define a reaction space by equation

    * (ex. $x^2 + y^2 \leq 1$)

  – Localization of molecules, parameters

Solving PDEs

- Discrete method of space: Finite Difference Method (FDM)

  – Diffusion equation: Forward–Time Central–Space method

  – Advection equation: CIP method [1]
    CiP: (Cubic Interpolated Pseudo–particle Method)

- Integration: 4th order Runge–Kutta

[1] H. Takewaki and T. Yabe, *J. Comput. Phys*, 1987

Example

- Diffusion–Reaction model
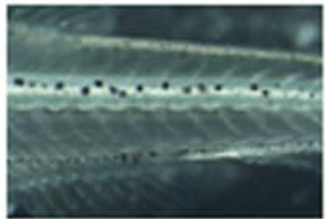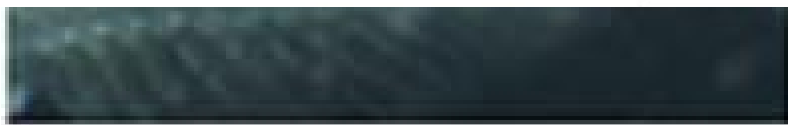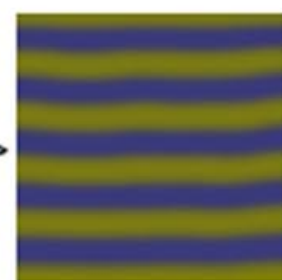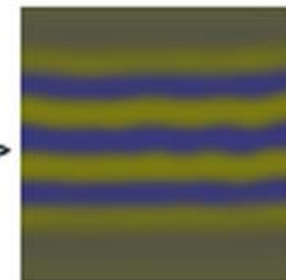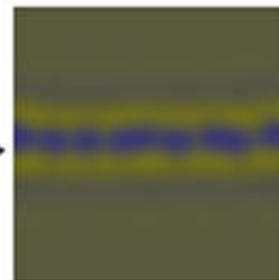
- Construction of a striped pattern (Zebrafish)

A. Nakamatsu, G. Takahasi, A. Kanbe and S. Kondo, *PNAS*, 2009

Equations

$$\frac{\partial u}{\partial t} = F(u, v, w) - c_u u + D_u \nabla^2 u$$

$$F(u, v, w) = \begin{cases} 0 & (c_1 v + c_2 w + c_3 < 0) \\ c_1 v + c_2 w + c_3 & (0 \le c_1 v + c_2 w + c_3 \le U) \\ U & (U < c_1 v + c_2 w + c_3) \end{cases}$$

$$\frac{\partial v}{\partial t} = G(u, v, w) - c_v v + D_v \nabla^2 v$$

$$G(u, v, w) = \begin{cases} 0 & (c_4 u + c_5 w + c_6 < 0) \\ c_4 u + c_5 w + c_6 & (0 \le c_4 u + c_5 w + c_6 \le V) \\ V & (V < c_4 u + c_5 w + c_6) \end{cases}$$

$$\frac{\partial w}{\partial t} = H(u, v, w) - c_w w + D_w \nabla^2 w$$

$$H(u, v, w) = \begin{cases} 0 & (c_7 u + c_8 v + c_9 < 0) \\ c_7 u + c_8 v + c_9 & (0 \le c_7 u + c_8 v + c_9 \le W) \\ W & (W < c_7 u + c_8 v + c_9) \end{cases}$$

$c_1 = -0.04, c_2 = -0.055, c_3 = 0.37, c_4 = -0.05, c_5 = 0.0, c_6 = 0.25, c_7 = 0.016, c_8 = -0.03, c_9 = 0.24$

$c_u = 0.02, c_v = 0.025, c_w = 0.06, D_u = 0.02, D_v = 0.02, D_w = 0.2, U = 0.5, V = 0.5, W = 0.5$

Simulation conditions

Simulation space $0 \leq x \leq 100,\ 0 \leq y \leq 100$
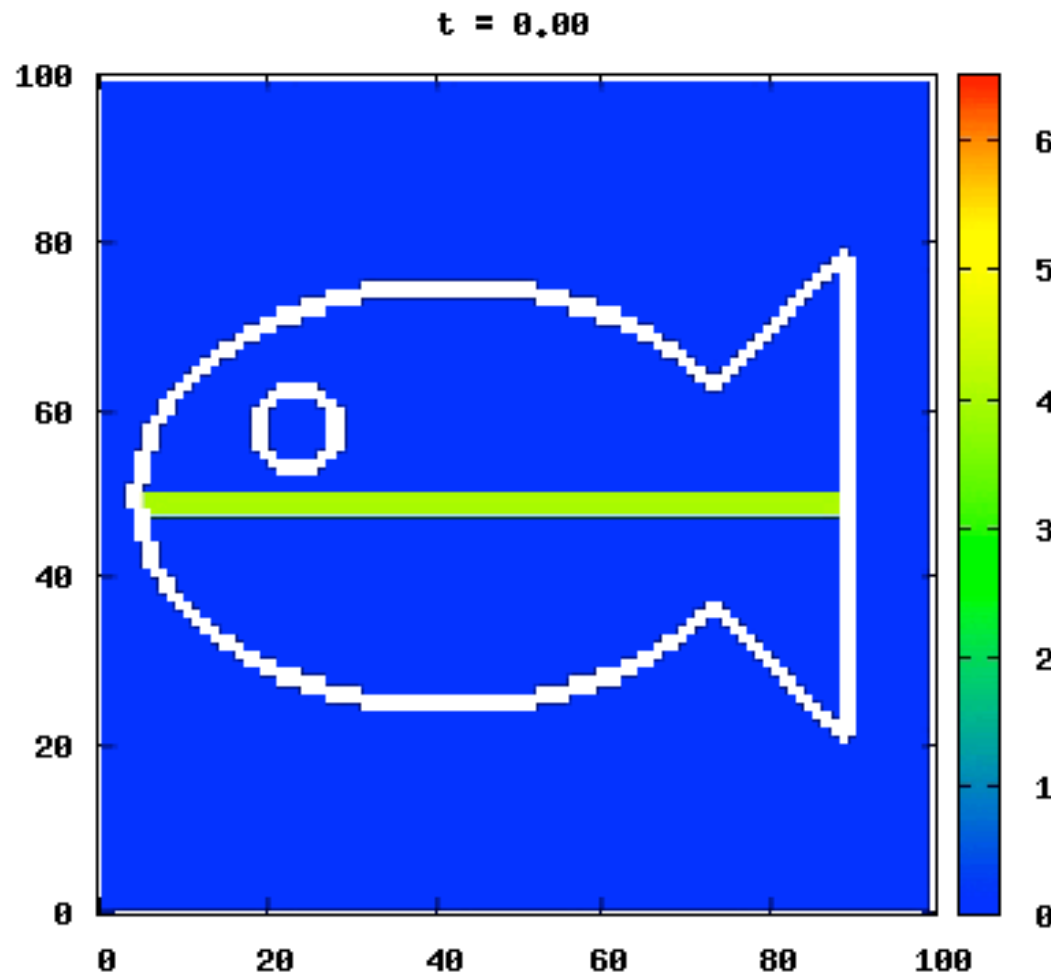
Eye (circle) $(x - 24)^2 + (y - 58)^2 < 25$

Body (ellipse +triangle)

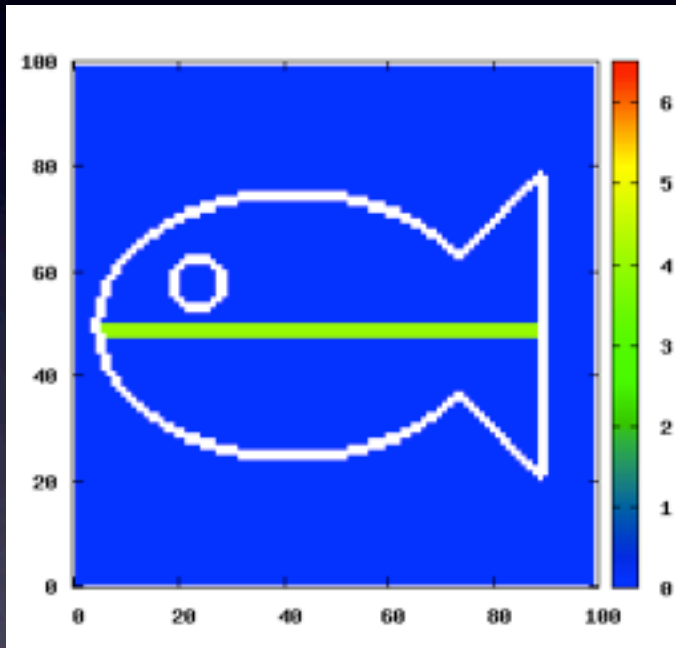$(0.27(x - 42))^2 + (0.4 * (y - 50))^2 < 100\ ||\ (y < x - 10\ \wedge\ y > -x + 110\ \wedge\ x < 90)$

- Reaction space: 100×100 mesh
- Time step: dt = 0.5
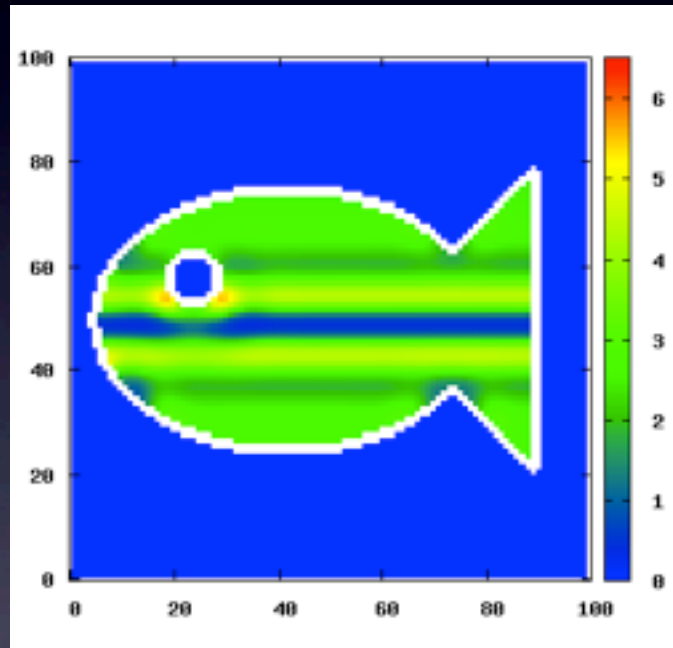- Simulation time: t = 0~1,500
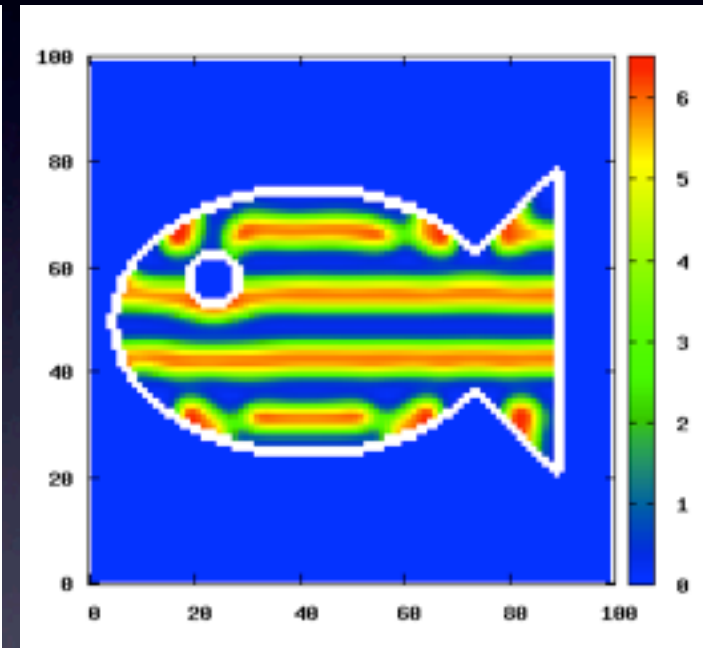
Simulation result

Simulation result

$t=0.00$ $t=600.00$ $t=1500.00$

Striped pattern is generated from one single line as an initial state

Summary

- Can simulate Advection-Diffusion reaction equation

  * Model is described with Spatial SBML ext.

  * Results is exported as text (numbers) and PNG

Future works

- Support bio-images to define the reaction space

- Importing SBML

  - Level 3

  - Units

  - Membrane transportation (Reaction)

- Integration of PDE

  - More precise calculation (ex. FEM)

  - Calculation around boundary condition on Advection equation

An interactive equation-based biological model builder

4th Sep. 2011
Akito Tabira, Noriko Hiroi, Akira Funahashi
Keio University

Motivation

- Build an application which is easy to build mathematical model
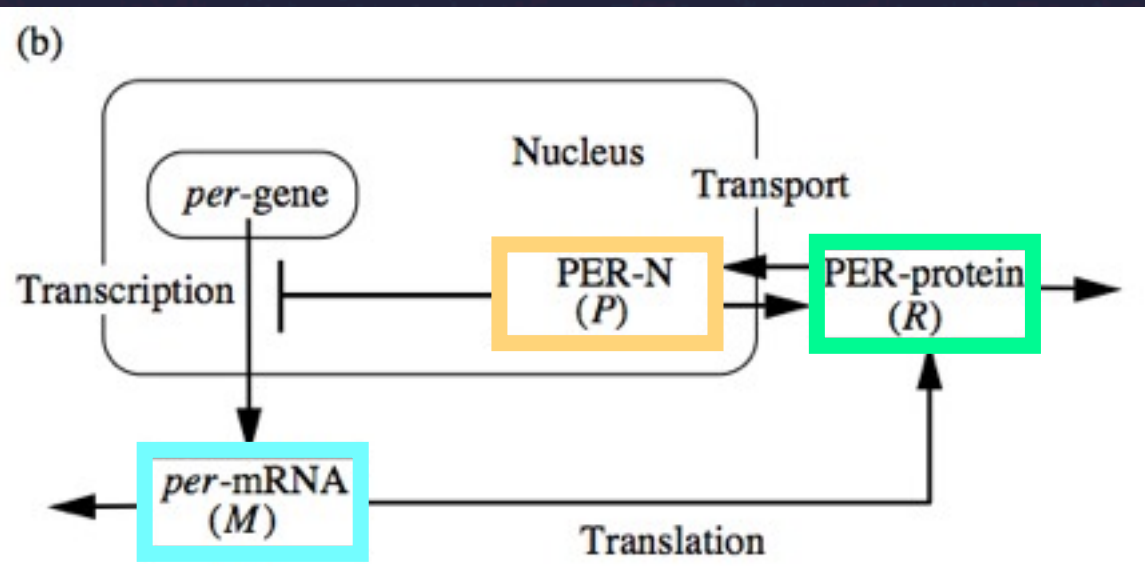

Kholodenko, Eur. J. Biochem. 267(6):1583–8 (2000)

Circadian clock model

- Protein (P) inhibits transcription of mRNA (M)
- M is translated to Protein (R)
- P / R will be transported to cytosol / nucleus

$$\frac{dM}{dt} = \frac{1}{1 + (P/h)^n} - aM,$$

$$\frac{dR}{dt} = sM - (d + u)R + vP,$$

$$\frac{dP}{dt} = uR - vP.$$

(b)

per-gene

Nucleus

Transport

Transcription

PER-N (P)

PER-protein (R)

per-mRNA (M)

Translation

Circadian clock model

$$\frac{\mathrm{d}M}{\mathrm{d}t} = \frac{1}{1 + (P/h)^n} - aM,$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = sM - (d + u)R + vP,$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = uR - vP.$$
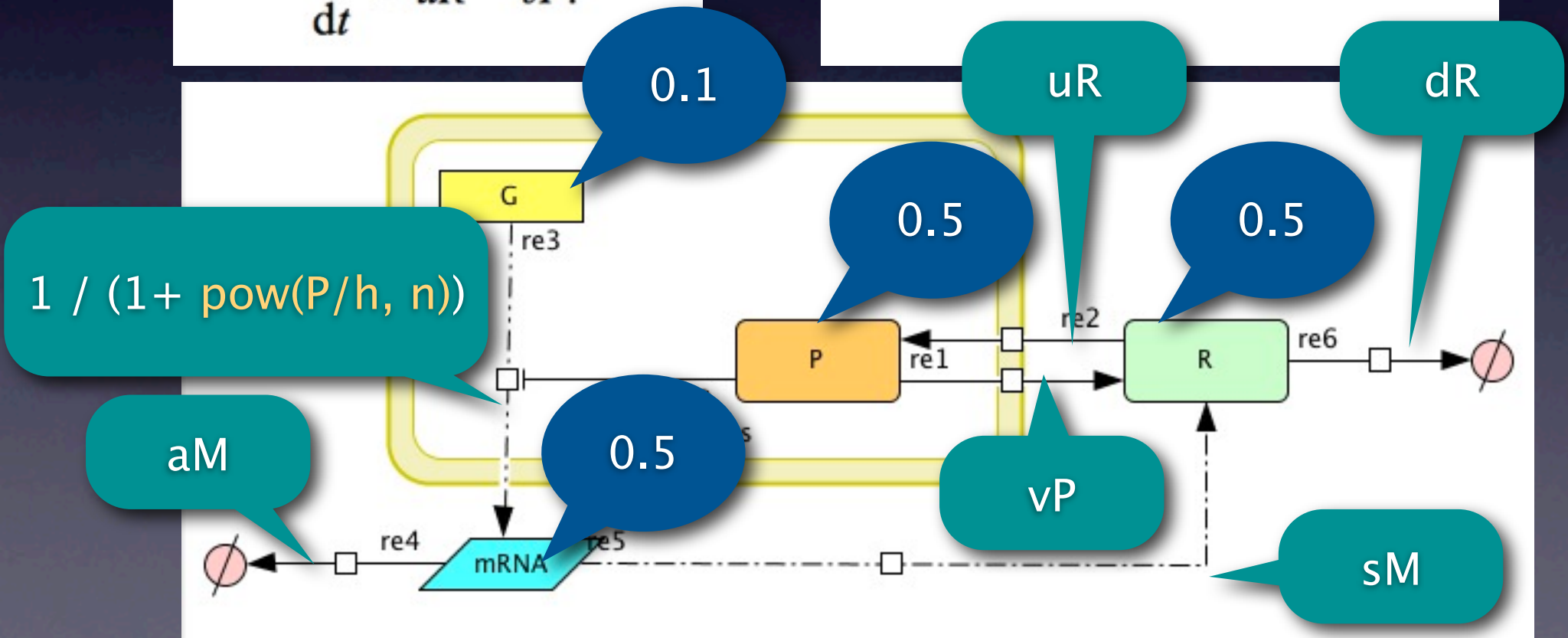
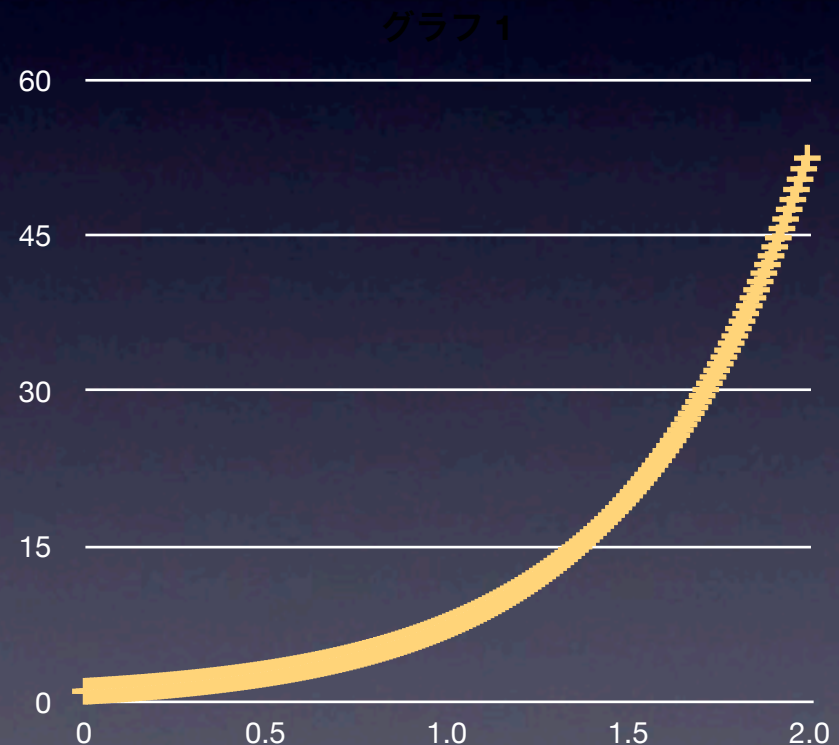$$a = s = d = v = 1.0$$
$$u = 0.1$$
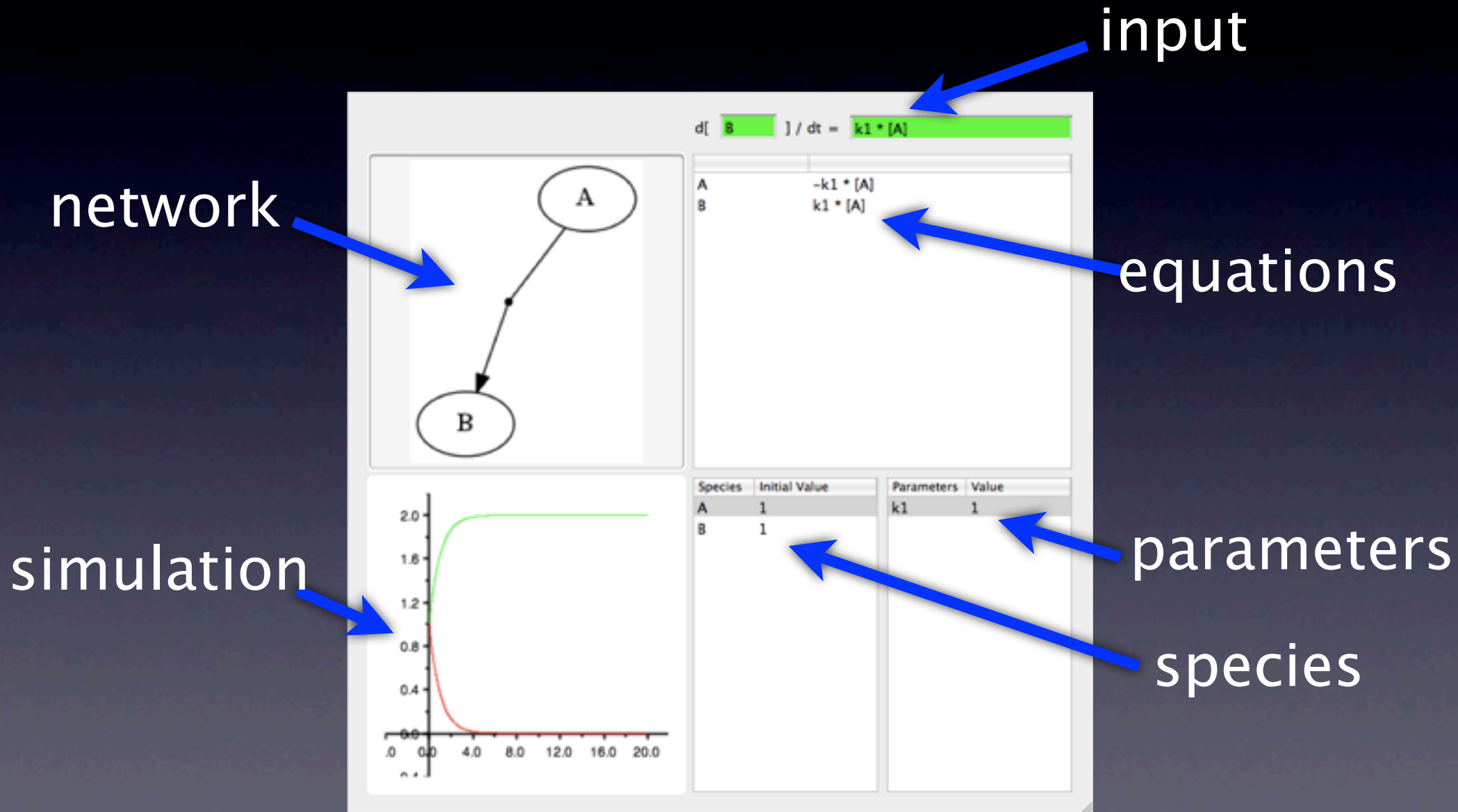$$h = 0.01$$
$$n = 40$$

Programming

```
int i;
double dxdt;
double t = 0.0;
double x = 1.0;
double dt = 0.01;

for (i = 0; i < 200; i++) {
  printf("%lf, %lf\n", t, x);
  dxdt = 2.0 * x;
  x = x + dxdt * dt;
  t = t + dt;
}
```
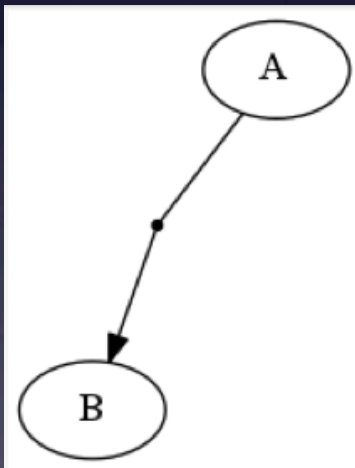
グラフ 1

Current status

- Generate an SBML model with ODE as an input

- Visualize a model as a network
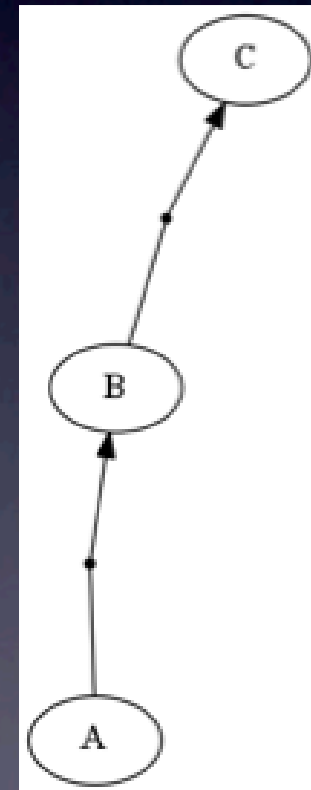
- Simulate a model

- Export an SBML model

Interface
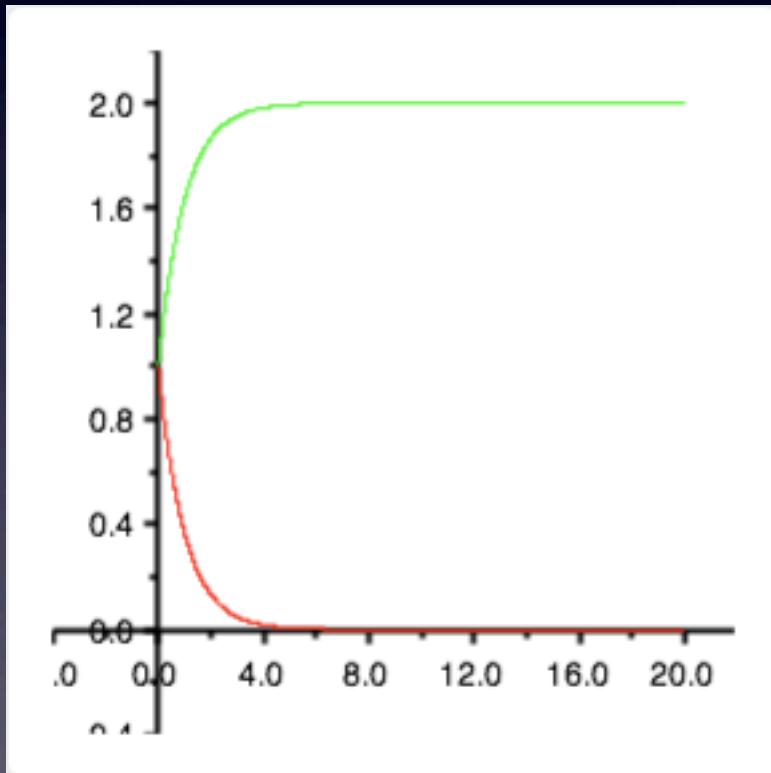
Generate a model from ODEs

d[B] / dt = k1 * [A]

d[C] / dt = k2 * [B]

Simulation
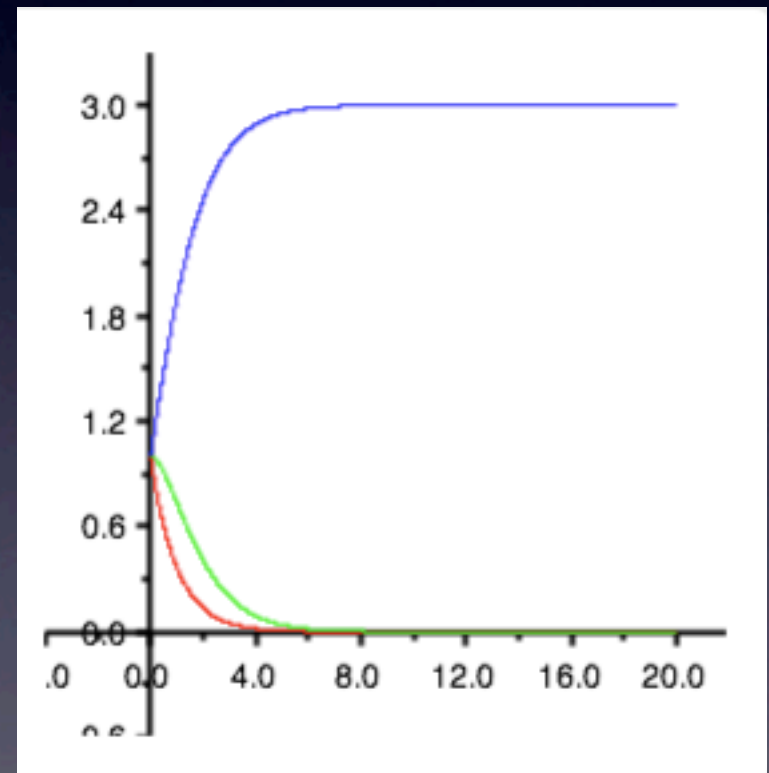
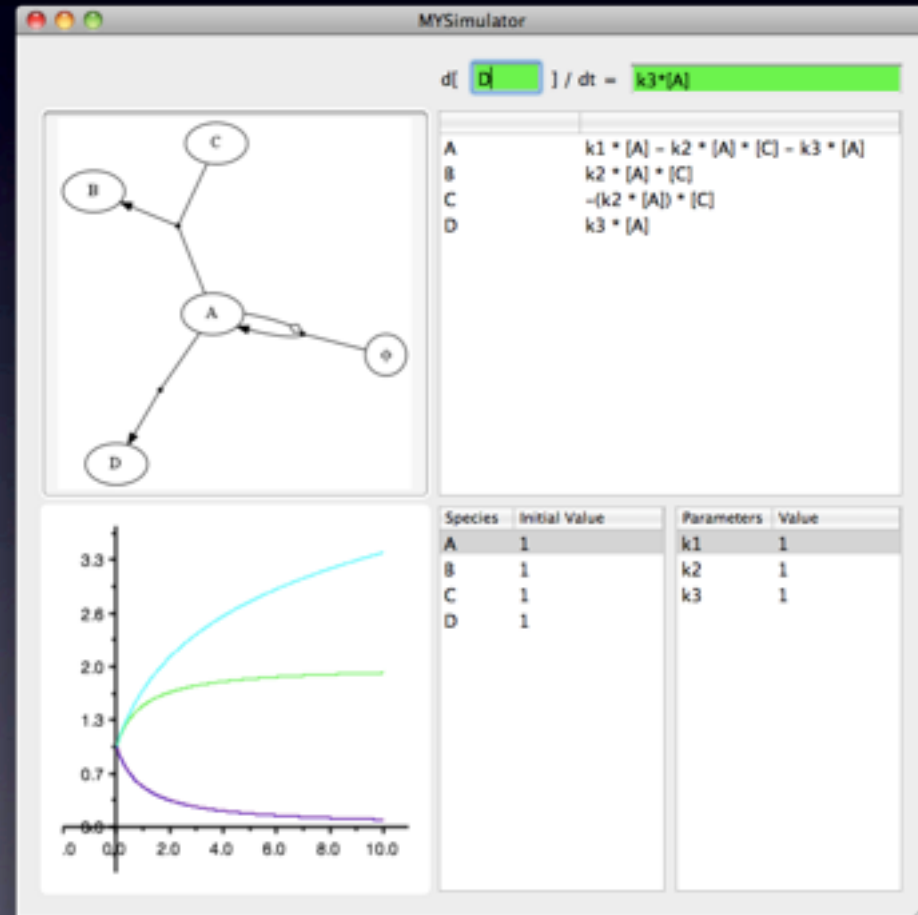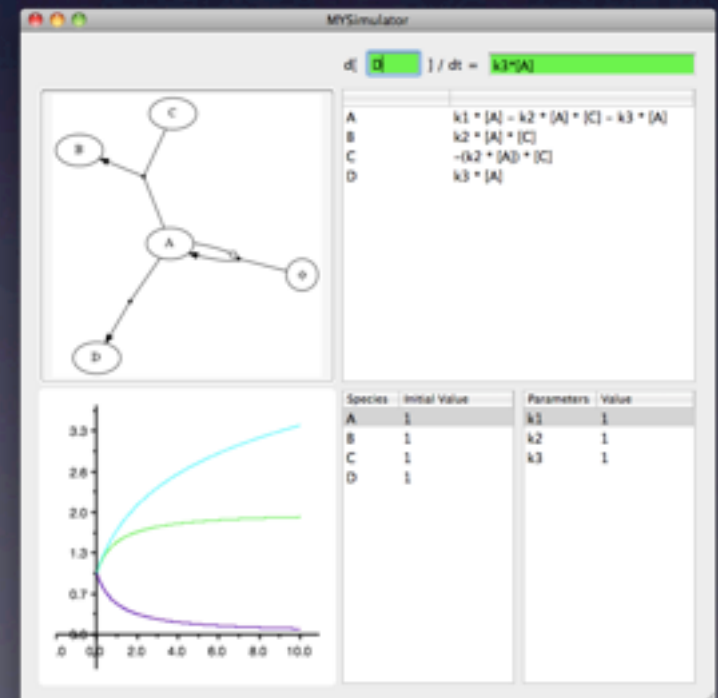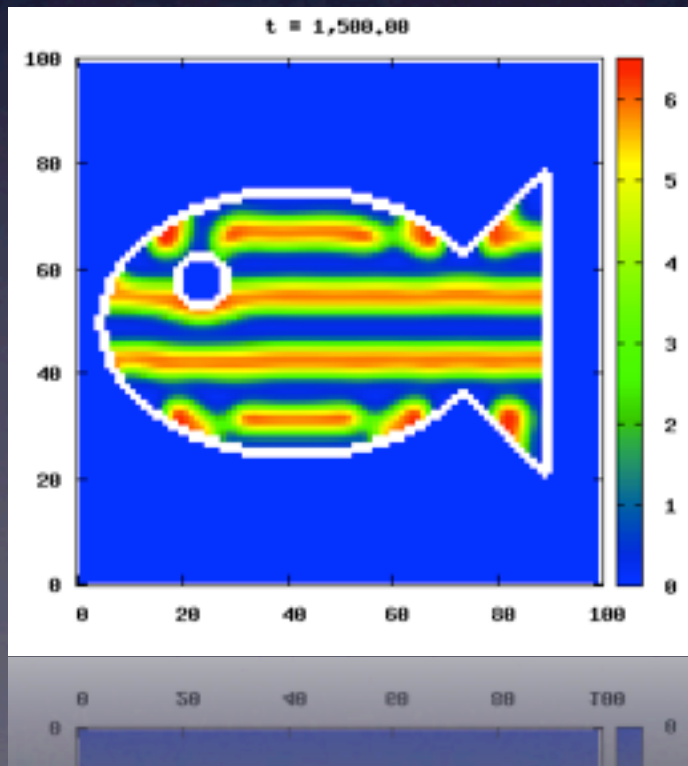# Future works

- Import SBML
- Michaelis-Menten, Hill, etc.
- Analysis
  - nullcline, isocline
- Visualization (auto-layout)

Acknowledgement

- James Schaff University of Connecticut Health Center, Farmington (CT), USA

- KAKENHI (21700328)

CellDesigner4.2

- Reduced Notation support
- Simulation
  - COPASI library is now included in CellDesigner
  - Parameter Polymorphism support
  - Export SED-ML
- Database Connection
  - Connect to UniProt
  - Connect to databased using MIRIAM annotation
  - Import models from JWS Online
- Plug-in API (run simulation from plugin, etc.)
- SBGN View enhancement