signaling and space

Jorg Stelling and Boris N Kholodenko. Signaling cascades as cellular devices for spatial computations. *J Math Biol*, 58(1-2):35–55, 2009 Jan.

CSB Mar 11

Vincent Danos

quick intro

Cell signaling is spatially heterogeneous even at steady state

signals sometimes generate spatially heterogeneous steady states, eg gradients of pho'ed proteins

done via microdomains on membranes or organelles, anchored Kinases, and anchored dual Kinases-Pho'ases

all this depends on shape, and size!

Turing patterns

Turing–pattern formation in a reaction–diffusion medium:

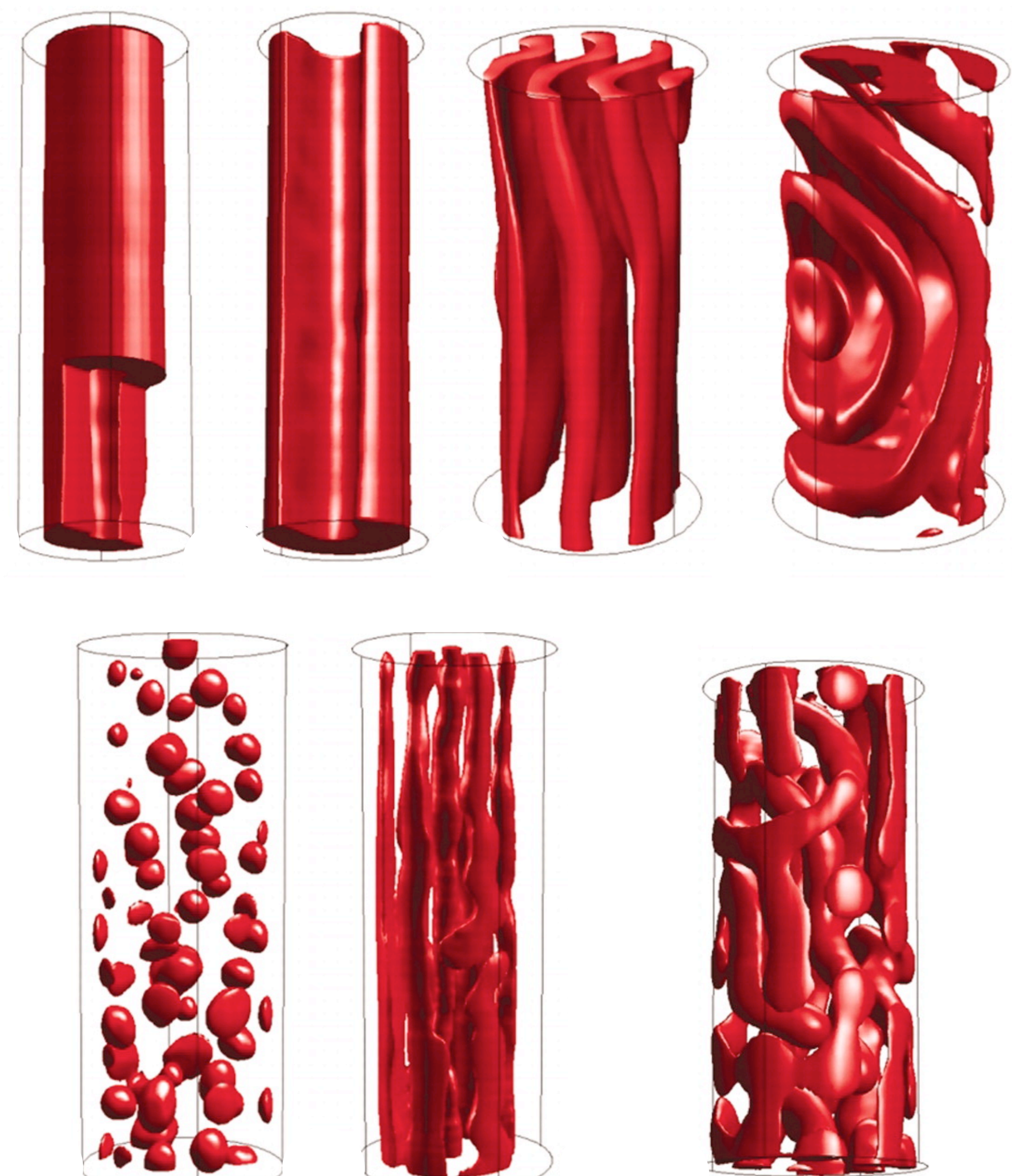
– long–range inhibition (fast diffusing inhibitor)

– short–range activation (slow diffusing activator)

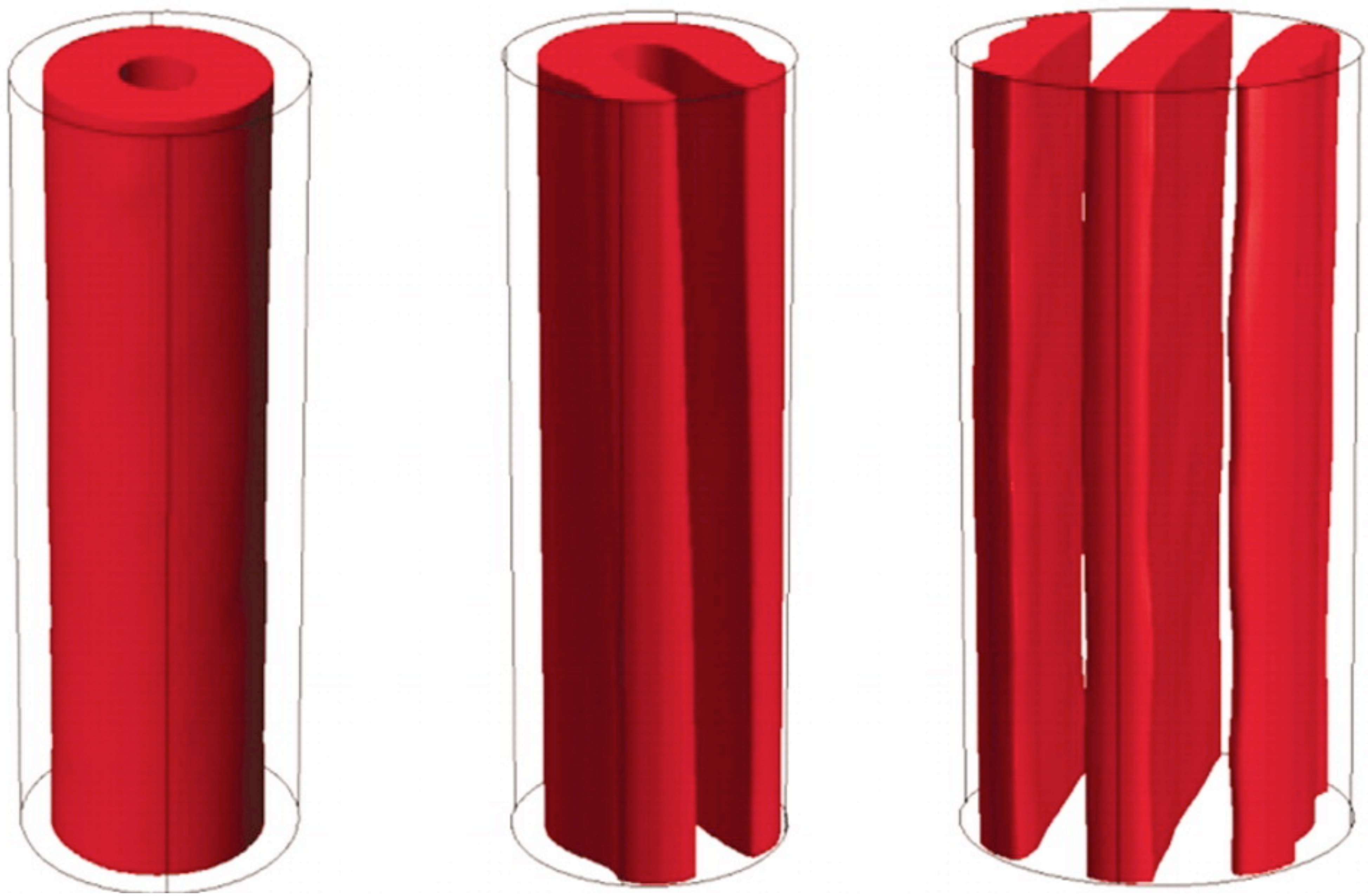
Numerical results are obtained from the model:

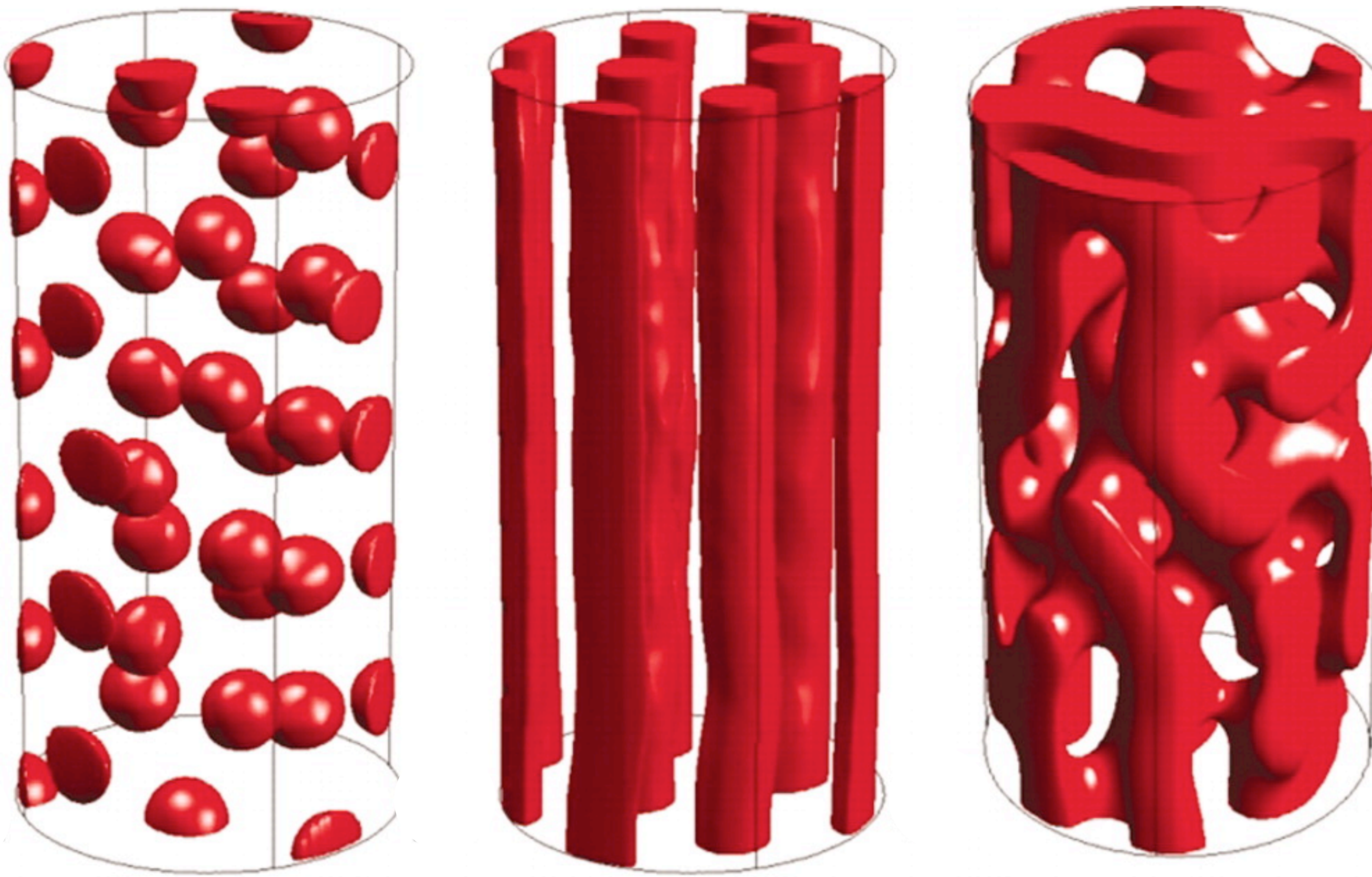
$$\begin{aligned} dx/dt &= (1/\epsilon)(fz(q-x)/(q+x) + x(1-mz)/(\epsilon_1+1-mz) - x^2) + \Delta x \\ dz/dt &= x(1-mz)/(\epsilon_1+1-mz) - z + d_z \Delta z \end{aligned}$$

- x the activator and z the inhibitor
- $d_z = 10$ is the ratio of diffusion coefficients Dz/Dx
- t is time
- $q = 0.0002$, $m = 0.0007$, $\epsilon_1 = 0.02$, $\epsilon = 2.2$; $f = (A)1.1$, (B) 0.93, (C) to (F) 0.88.
- size of domains: diameter = 20 (A) to (C) and (F), 14 (D) and (E), height = 40.



in numero





simple diffusion-reaction

discrete

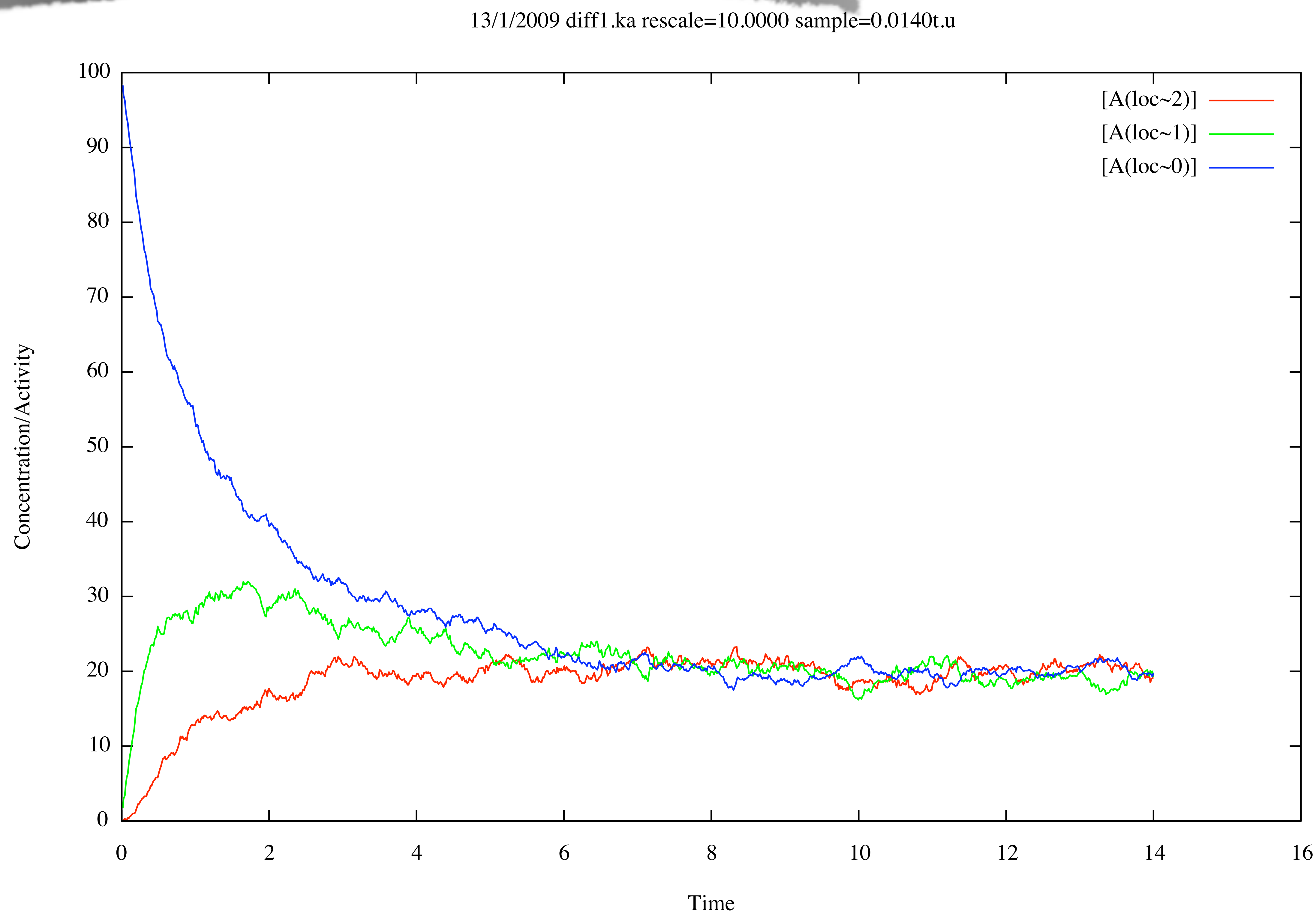
discrete model

```
A(loc~0) <-> A(loc~1) @ 1,1
A(loc~1) <-> A(loc~2) @ 1,1
A(loc~2) <-> A(loc~3) @ 1,1
A(loc~3) <-> A(loc~4) @ 1,1
A(loc~4) <-> A(loc~5) @ 1,1
A(loc~5) <-> A(loc~6) @ 1,1
A(loc~6) <-> A(loc~7) @ 1,1
A(loc~7) <-> A(loc~8) @ 1,1
A(loc~8) <-> A(loc~9) @ 1,1
```

```
%init: 100 * (A(a~blue,loc~9))
%init: 100 * (A(a~red,loc~0))
```

steady state

steady state is homogeneous



steady state - balance

$$
\begin{aligned}
&a_0 = a_1 \\
&2a_i = a_{i-1} + a_{i+1} \quad \text{if } 0 < i < m-1 \\
&a_{m-1} = a_{m-2}
\end{aligned}
$$

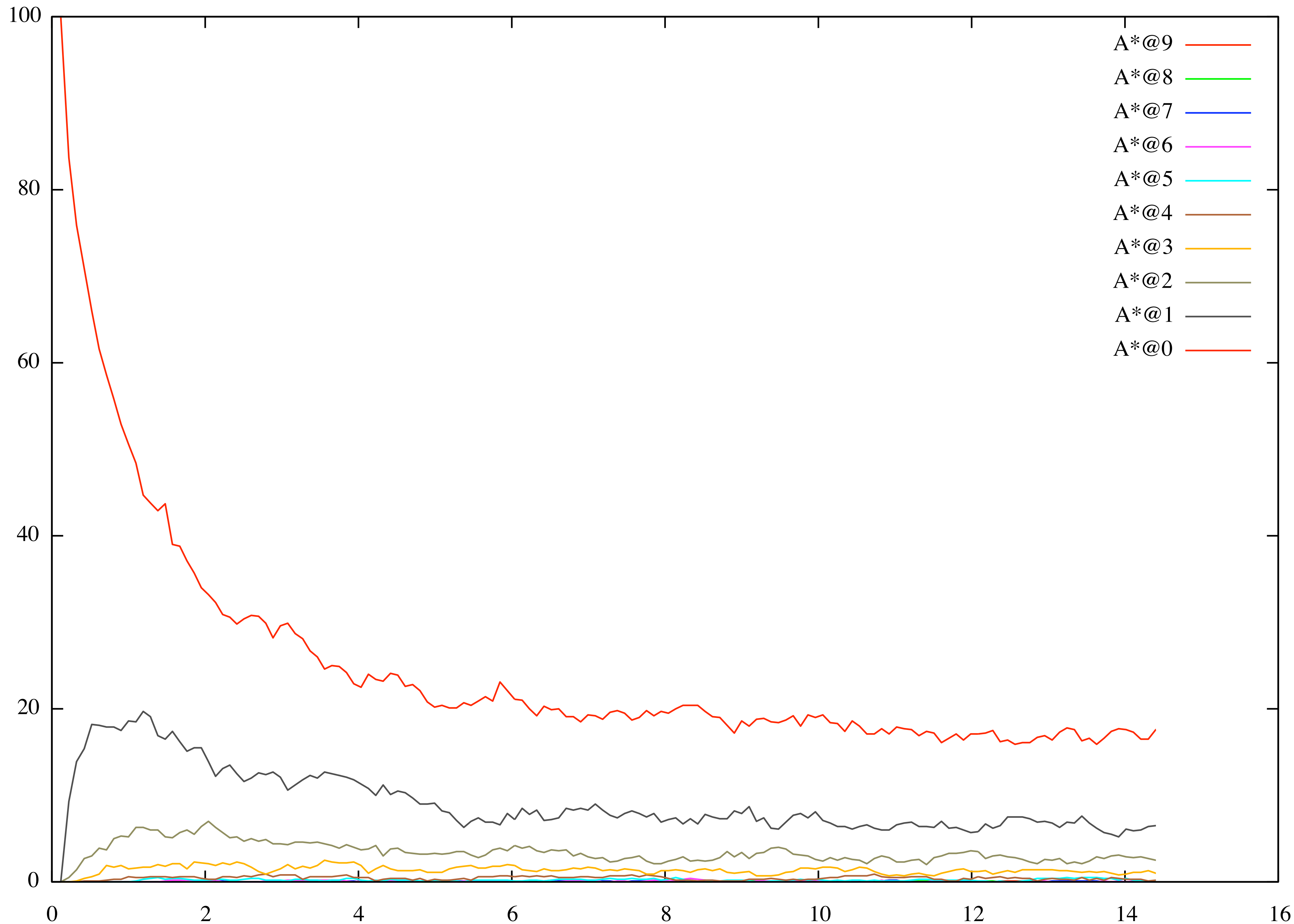
heat source at 0

kinase at 0, diffusible pho'ase

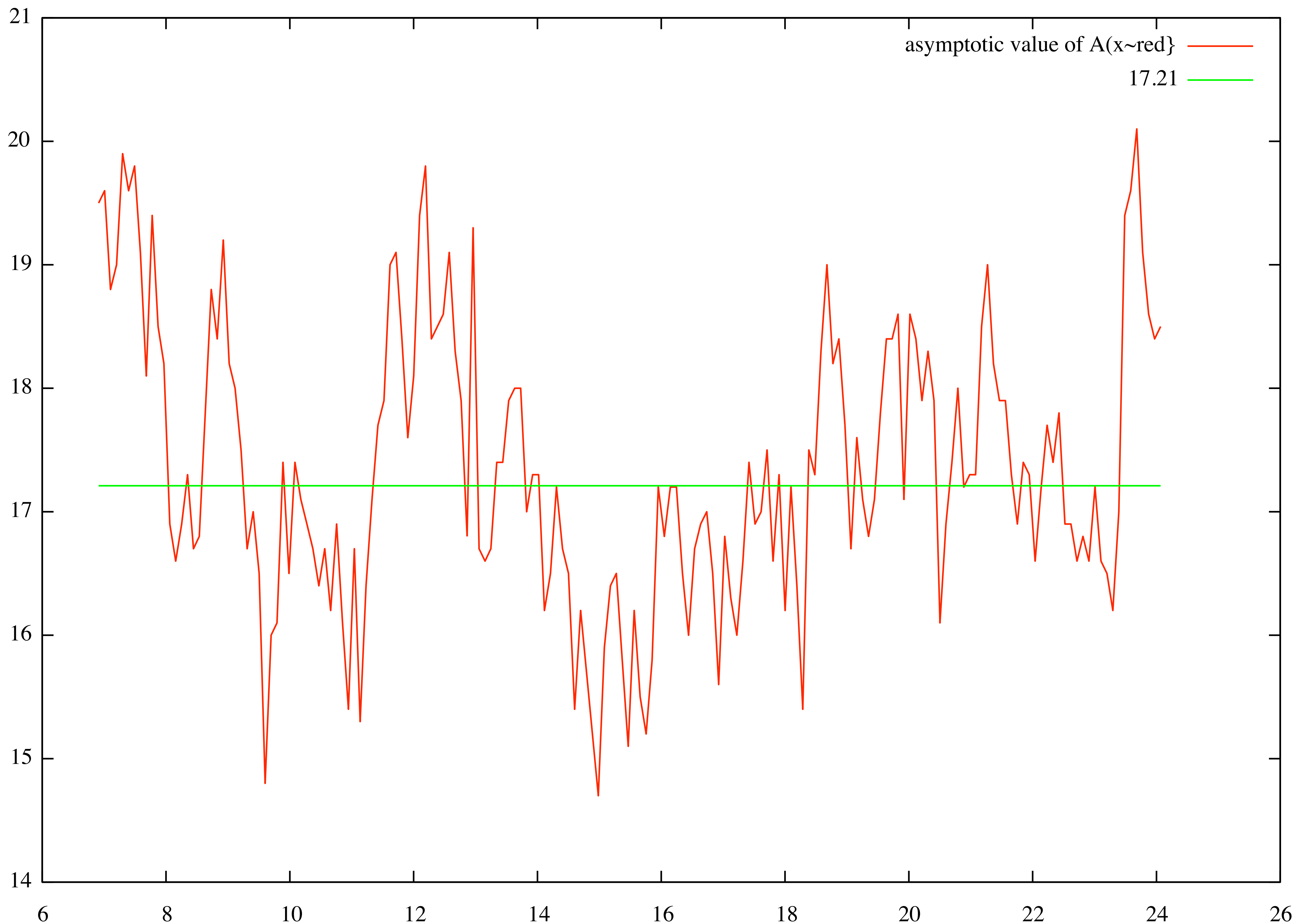
```
'red 0' A(loc~0,a~blue) -> A(loc~0,a~red) @ 10
'blue' A(a~red) -> A(a~blue) @ 1
```

```
# Simulation observables:
%obs: 'A*@0' A(a~red,loc~0)
%obs: A(a~red,loc~1)
%obs: A(a~red,loc~2)
%obs: A(a~red,loc~3)
%obs: A(a~red,loc~4)
%obs: A(a~red,loc~5)
%obs: A(a~red,loc~6)
%obs: A(a~red,loc~7)
%obs: A(a~red,loc~8)
%obs: A(a~red,loc~9)
```

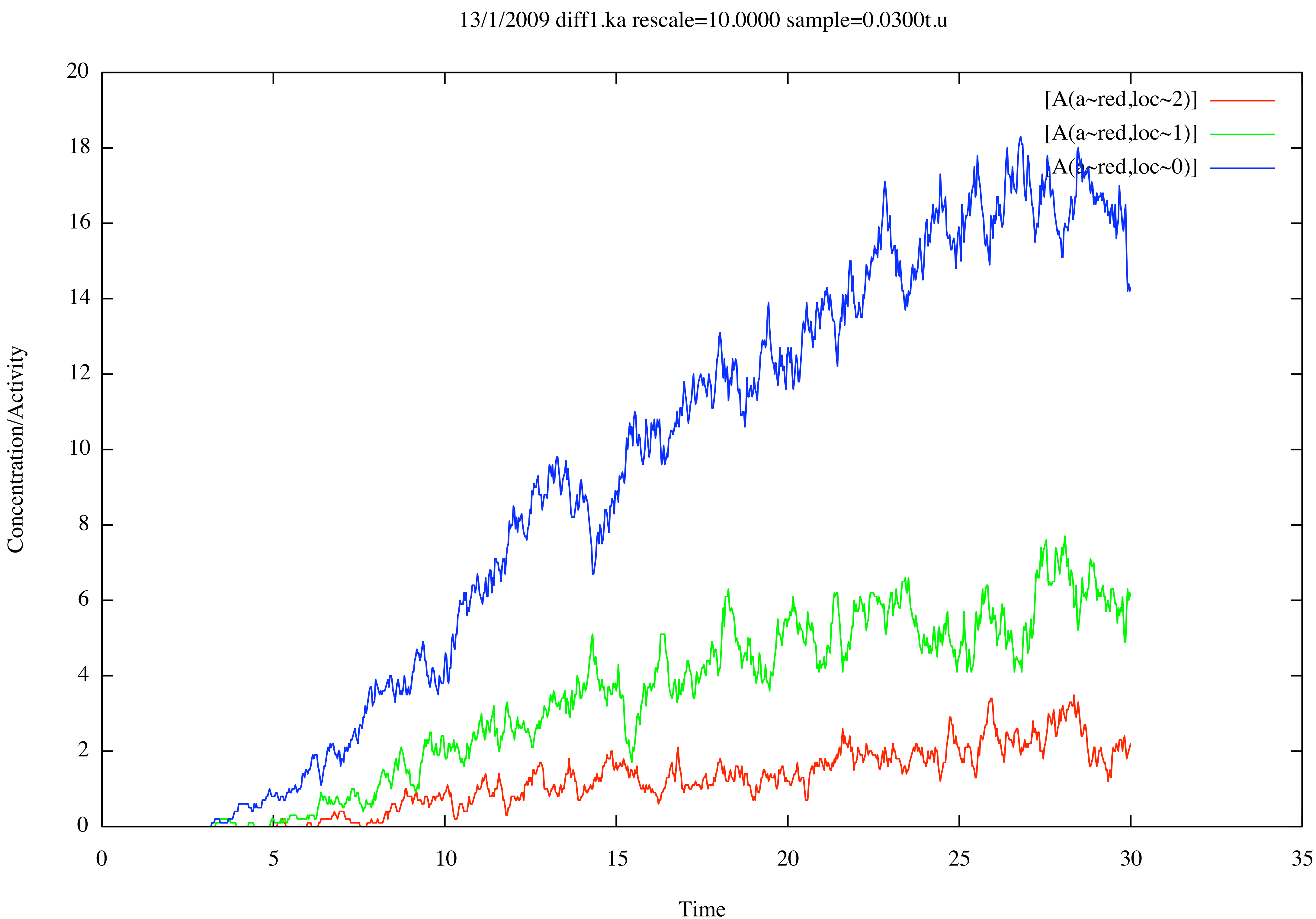
simulation



steady state A(a~red,loc~0)



guess initial condition



steady state

-mig -loc +mig +loc

$$
\begin{aligned}
&da_0^r + k'a_0^r = da_1^r + ka_0^b \\
&da_0^b + ka_0^b = da_1^b + k'a_0^r \\
&2da_i^r + k'a_i^r = da_{i-1}^r + da_{i+1}^r \quad \text{if } 0 < i < m-1 \\
&2da_i^b = da_{i-1}^b + da_{i+1}^b + k'a_i^r \quad \text{if } 0 < i < m-1 \\
&da_{m-1}^r + k'a_{m-1}^r = da_{m-2}^r \\
&da_{m-1}^b = da_{m-2}^b + k'a_{m-1}^r
\end{aligned}
$$

steady state

-mig -loc +mig +loc

$$
\begin{aligned}
&da_0^r + k'a_0^r = da_1^r + ka_0^b \\
&da_0^b + ka_0^b = da_1^b + k'a_0^r \\
&2da_i^r + k'a_i^r = da_{i-1}^r + da_{i+1}^r \quad \text{if } 0 < i < m-1 \\
&2da_i^b = da_{i-1}^b + da_{i+1}^b + k'a_i^r \quad \text{if } 0 < i < m-1 \\
&da_{m-1}^r + k'a_{m-1}^r = da_{m-2}^r \\
&da_{m-1}^b = da_{m-2}^b + k'a_{m-1}^r
\end{aligned}
$$

$$a_0^r = \frac{N}{m} \cdot \frac{1}{1 + \frac{k'}{2k}(1 + \sqrt{1 + \frac{4d}{k'}})}$$

$$a_{i+1}^r = \alpha_{i+1} a_i^r$$

$$\alpha_i = d/(2d + k' - \alpha_{i+1} d)$$

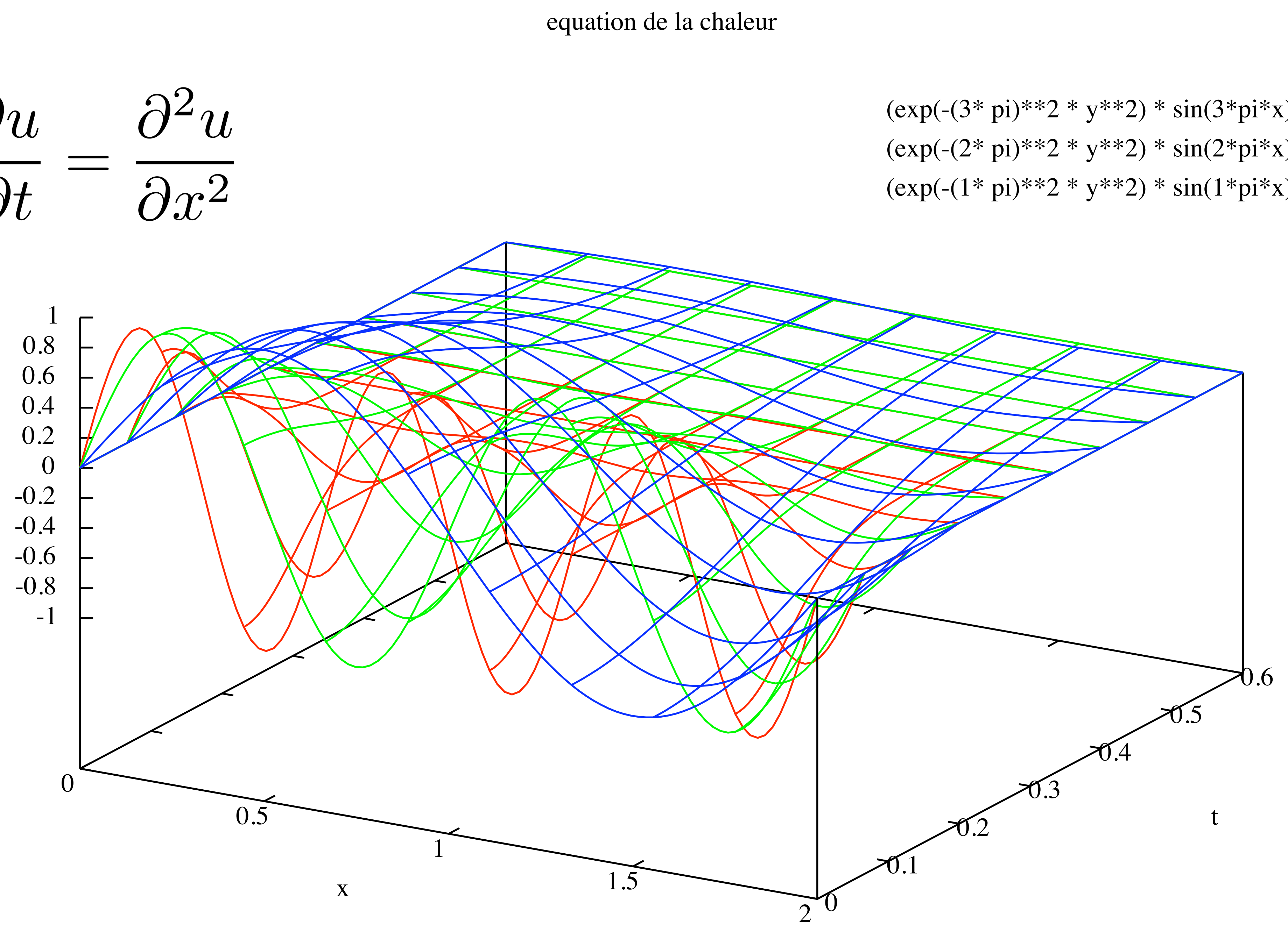
$$0 < \alpha := (1 + \frac{k'}{2d}) - (\frac{k'}{2d})^{1/2}(2 + \frac{k'}{2d})^{1/2} \leq 1$$

simple diffusion

continuous

pure diffusion PDE/heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$



$$u(x,t) = \sum_n B_n e^{-\pi^2 n^2 t} \sin n\pi x$$

reaction/diffusion PDE

$$
\begin{aligned}
\partial_t a^r &= d\partial_x^2 a^r - k' a^r \\
\partial_t a^b &= d\partial_x^2 a^b + k' a^r \\
\partial_t a^r(0,t) &= d\partial_x a^r(0,t) + k a^b(0,t) - k' a^r(0,t) \\
\partial_t a^b(0,t) &= d\partial_x a^b(0,t) + k' a^r(0,t) - k a^b(0,t) \\
\partial_t a^r(1,t) &= -d\partial_x a^r(1,t) - k' a^r(1,t) \\
\partial_t a^b(1,t) &= -d\partial_x a^b(1,t) + k' a^r(1,t)
\end{aligned}
$$

$$
\begin{aligned}
\partial_t a(x,t) &= d\partial_x^2 a(x,t) && \text{for } x \in (0,1) \\
\partial_t a(0,t) &= d(\partial_x a)(0,t) && \text{for } x = 0 \\
\partial_t a(1,t) &= -d(\partial_x a)(1,t) && \text{for } x = 1
\end{aligned}
$$

suppose red As are slower

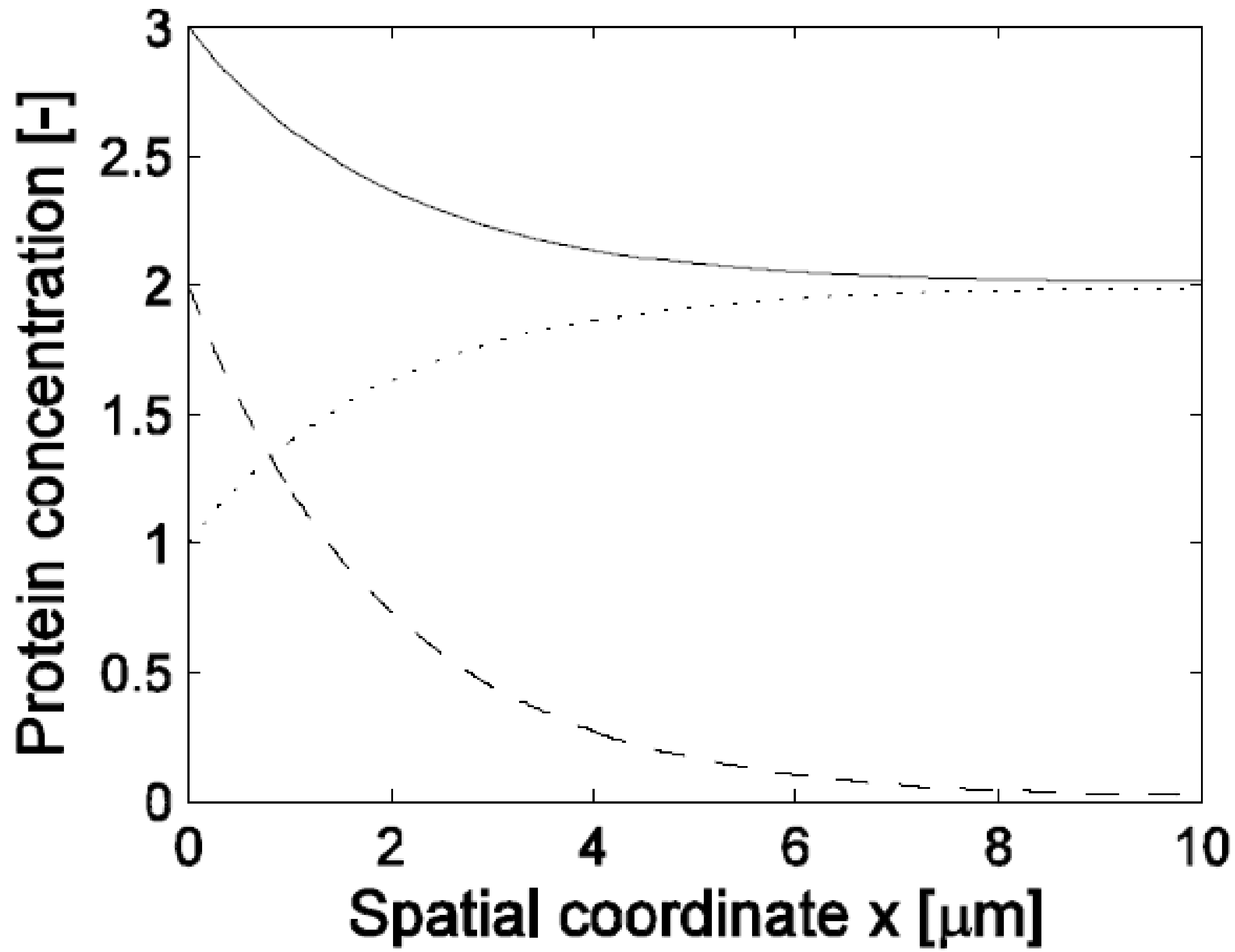
$$\partial_t a^r = d_r \Delta a^r - k' a^r$$
$$\partial_t a^b = d_b \Delta a^b + k' a^r$$

$$\partial_t a^r(0,t) = d_r \partial_x a^r(0,t) + k a^b(0,t) - k' a^r(0,t)$$
$$\partial_t a^b(0,t) = d_b \partial_x a^b(0,t) + k' a^r(0,t) - k a^b(0,t)$$

$$\partial_t a^r(1,t) = -d_r \partial_x a^r(1,t) - k' a^r(1,t)$$
$$\partial_t a^b(1,t) = -d_b \partial_x a^b(1,t) + k' a^r(1,t)$$

$$d_r \Delta a^r + d_b \Delta a^b = 0$$
$$d_r \partial_x a^r(0) + d_b \partial_x a^b(0) = 0$$
$$d_r \partial_x a^r + d_b \partial_x a^b = 0$$
$$d_b(a^b(x) - a^b(0)) = d_r(a^r(0) - a^r(x))$$

a = a(blue)+a(red) - material gradient



Grid step, continuous vs. discrete

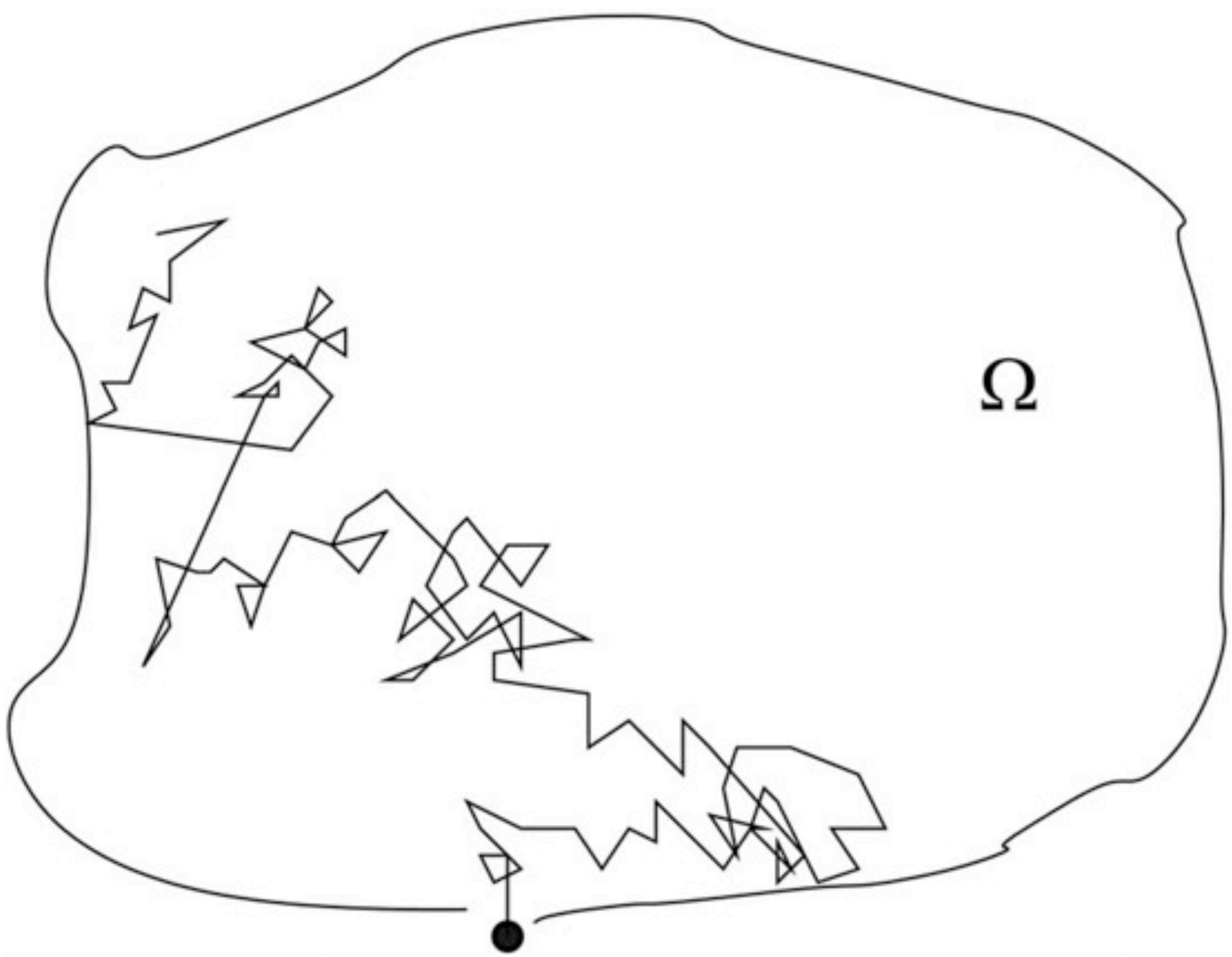
The relationship between the discrete grid diffusion rate δ_h, for steps of length h (size of a cell), and D the continuous diffusivity is:

$$D \quad = \quad \delta_h \cdot h^2$$

NB: h is sometimes called the length scale of the grid.
This equation is similar to the binary volume correction $k = \gamma \cdot AV$.
Dimensions check (do not confuse dimensions with units) $[L^2T^{-1}] = [T^{-1}][L^2]$.

narrow escape in 3d

Schuss Z et al. PNAS 2007;104:16098-16103



$$T = \frac{V}{4D}\epsilon^{-1}$$

$$\frac{\gamma_{2\to1}}{\gamma_{1\to2}} = \frac{V_1}{V_2} = \frac{n_1}{n_2}$$

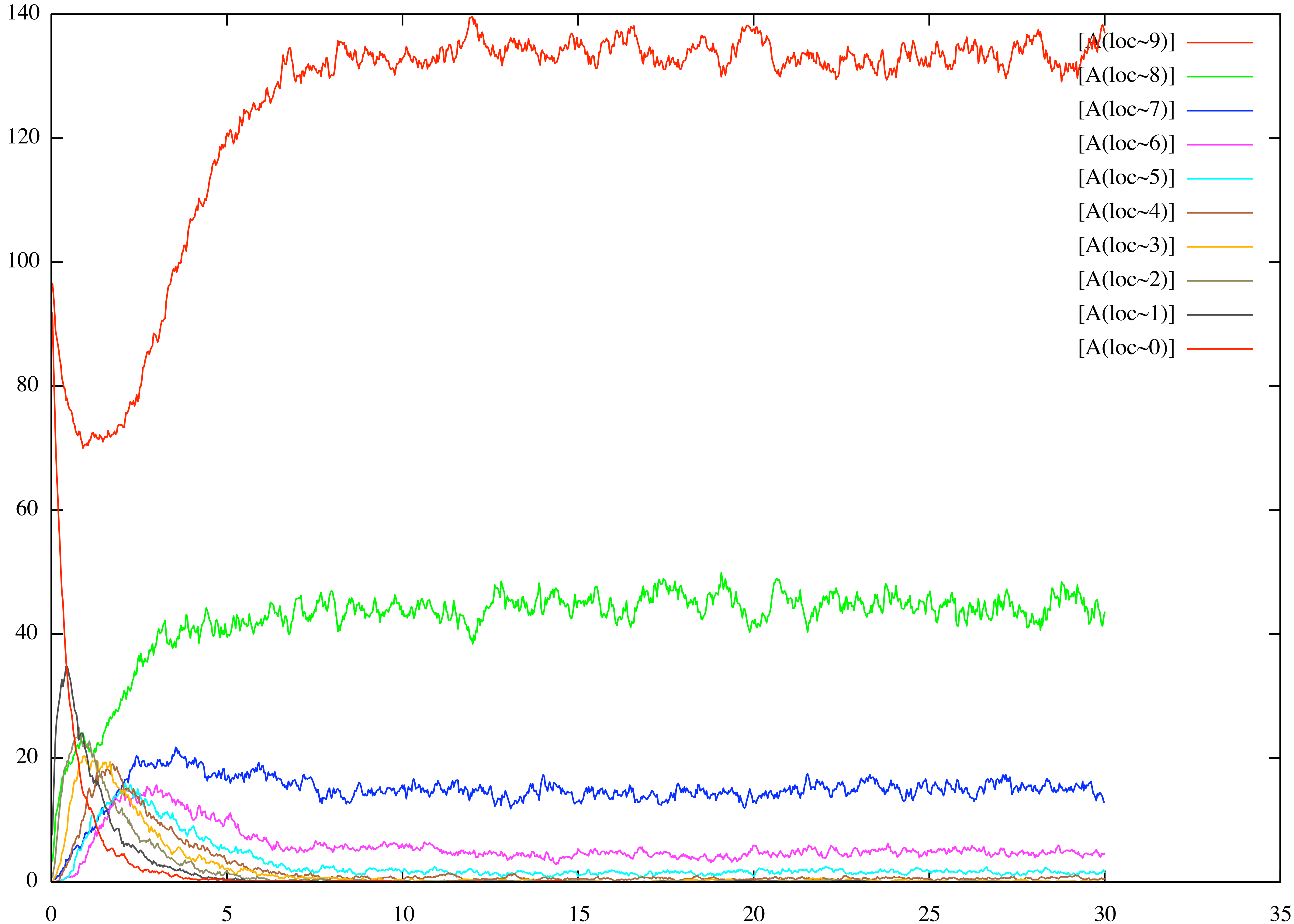
at equilibrium, as $\gamma_{1\to2}n_1 = \gamma_{2\to1}n_2$

so concentrations are equal

anistropic diffusion: d_r=3*d_b

```
A(loc~0) <-> A(loc~1) @ 3,1
A(loc~1) <-> A(loc~2) @ 3,1
A(loc~2) <-> A(loc~3) @ 3,1
A(loc~3) <-> A(loc~4) @ 3,1
A(loc~4) <-> A(loc~5) @ 3,1
A(loc~5) <-> A(loc~6) @ 3,1
A(loc~6) <-> A(loc~7) @ 3,1
A(loc~7) <-> A(loc~8) @ 3,1
A(loc~8) <-> A(loc~9) @ 3,1
```

material gradient II



cascades

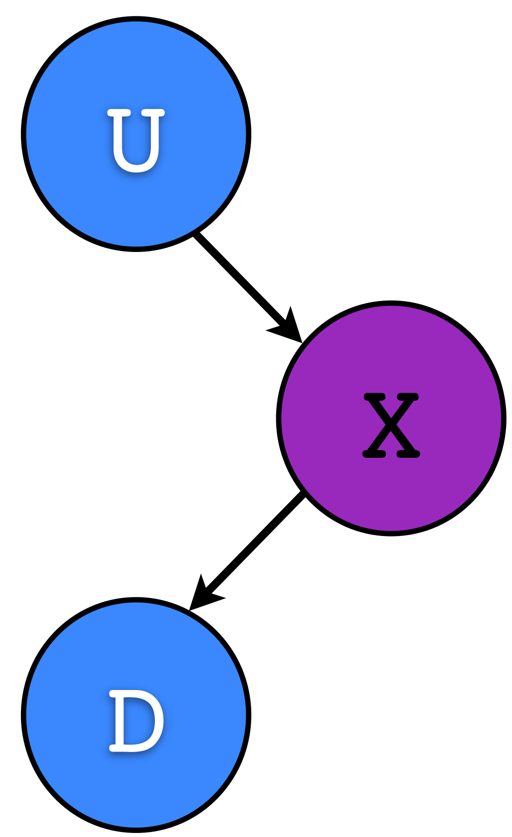
the art of cascading

we can compose cascades of reversible modifications (in a purely forward fashion here)

and choose

the wiring between layers

the anchoring



A simple cascade: U -> X -> D

- `U(s), X(s~u) -> U(s!1), X(s~u!1)`

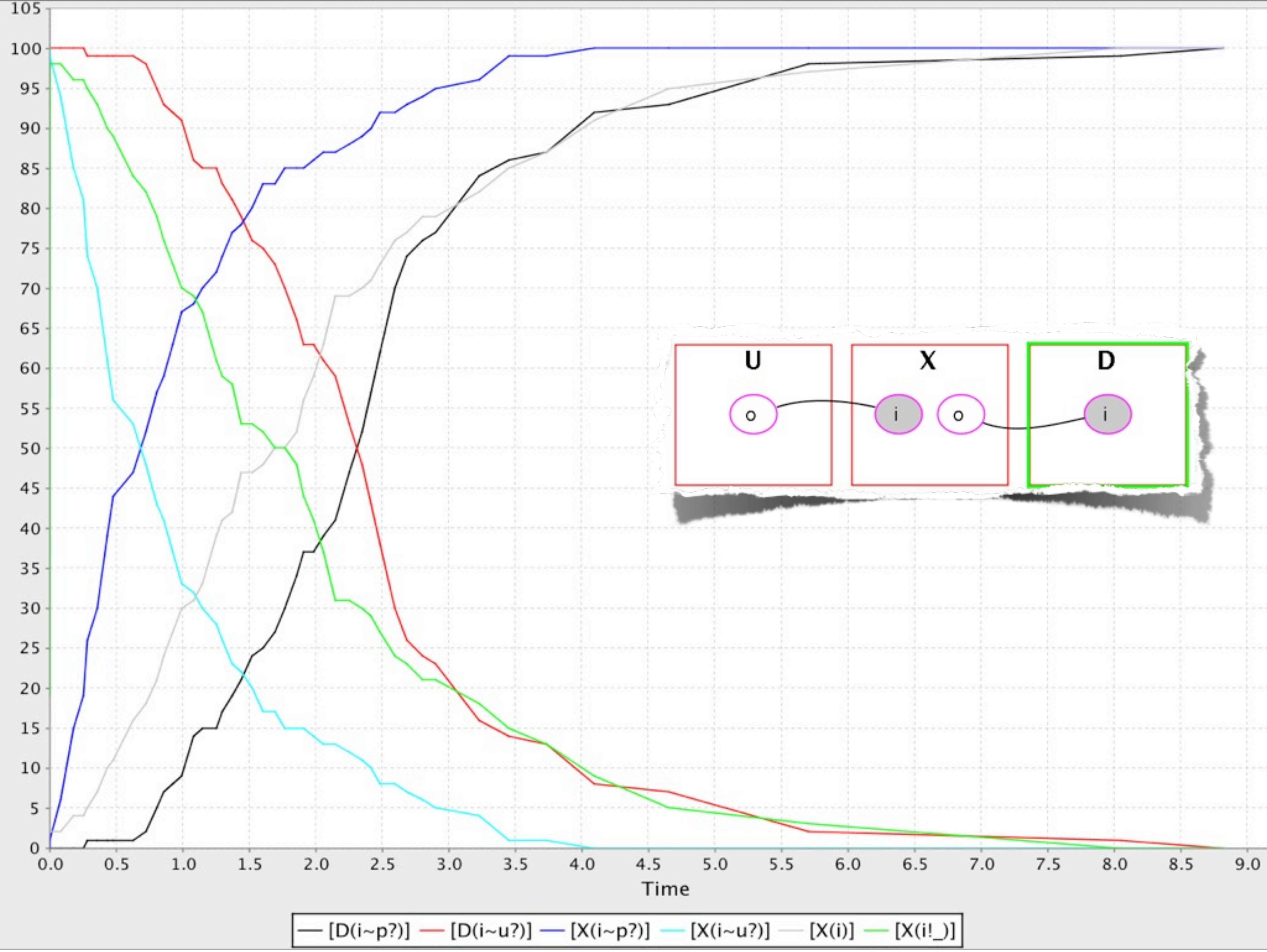
  `U(s!1), X(s!1) -> U(s), X(s)`

  `U(s!1), X(s~u!1) -> U(s!1), X(s~p!1)`

- `X(s~p?,d), D(s~u) -> X(s~p?,d!1), D(s~u!1)`

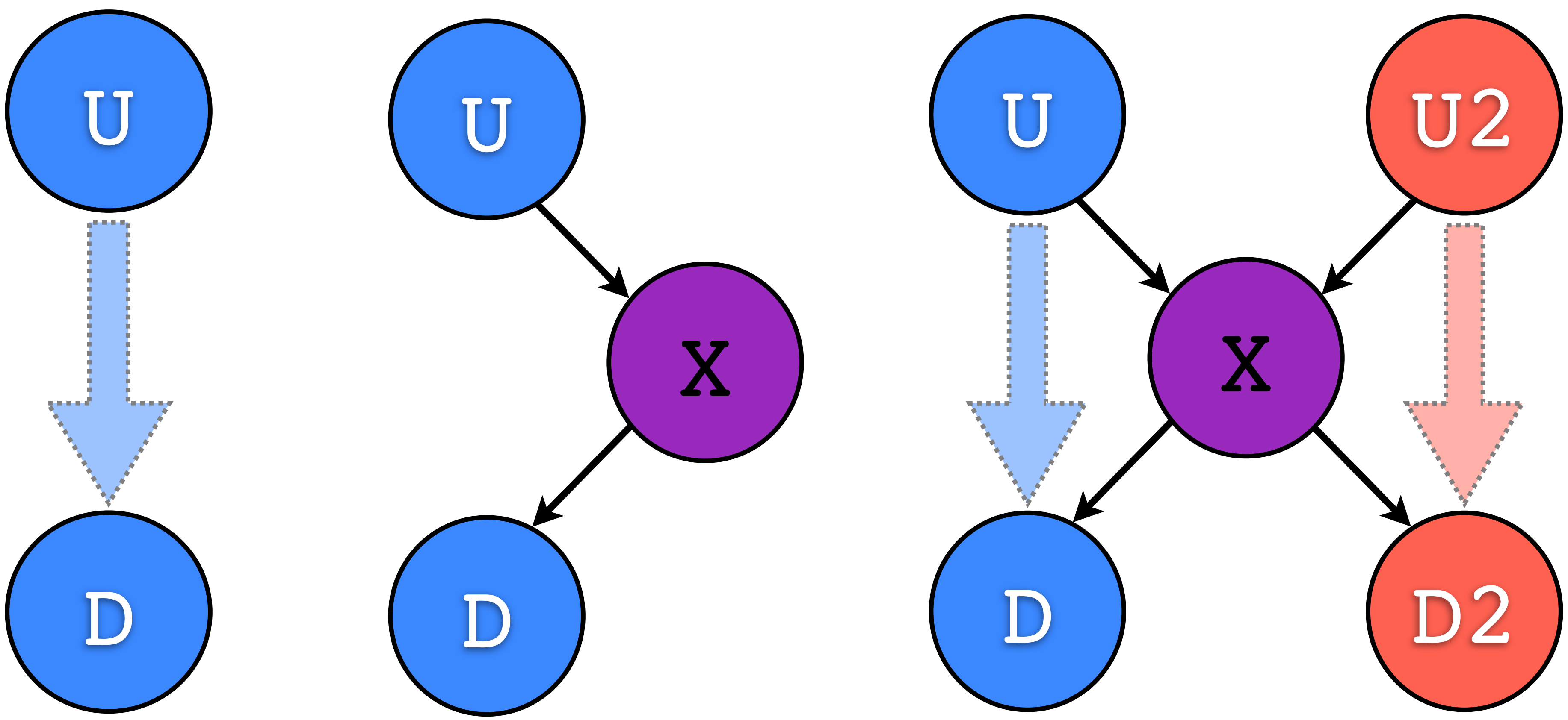
  `X(d!1), D(s!1) -> X(d), D(s)`

  `X(d!1), D(s~u!1) -> X(d!1), D(s~p!1)`

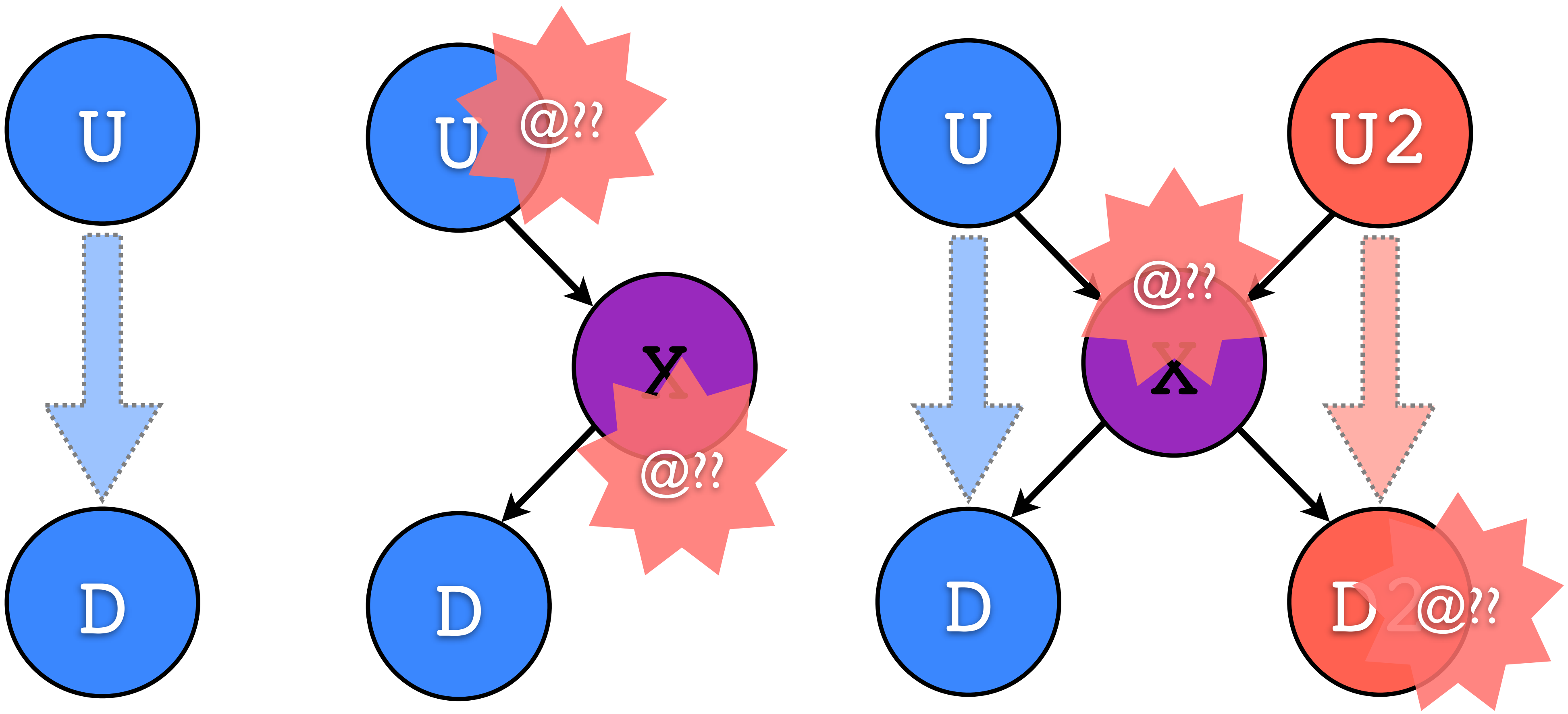

U
X
D
o
i
o
i
Time
[D(i~p?)]
[D(i~u?)]
[X(i~p?)]
[X(i~u?)]
[X(i)]
[X(i!_)]

diffusive cascades

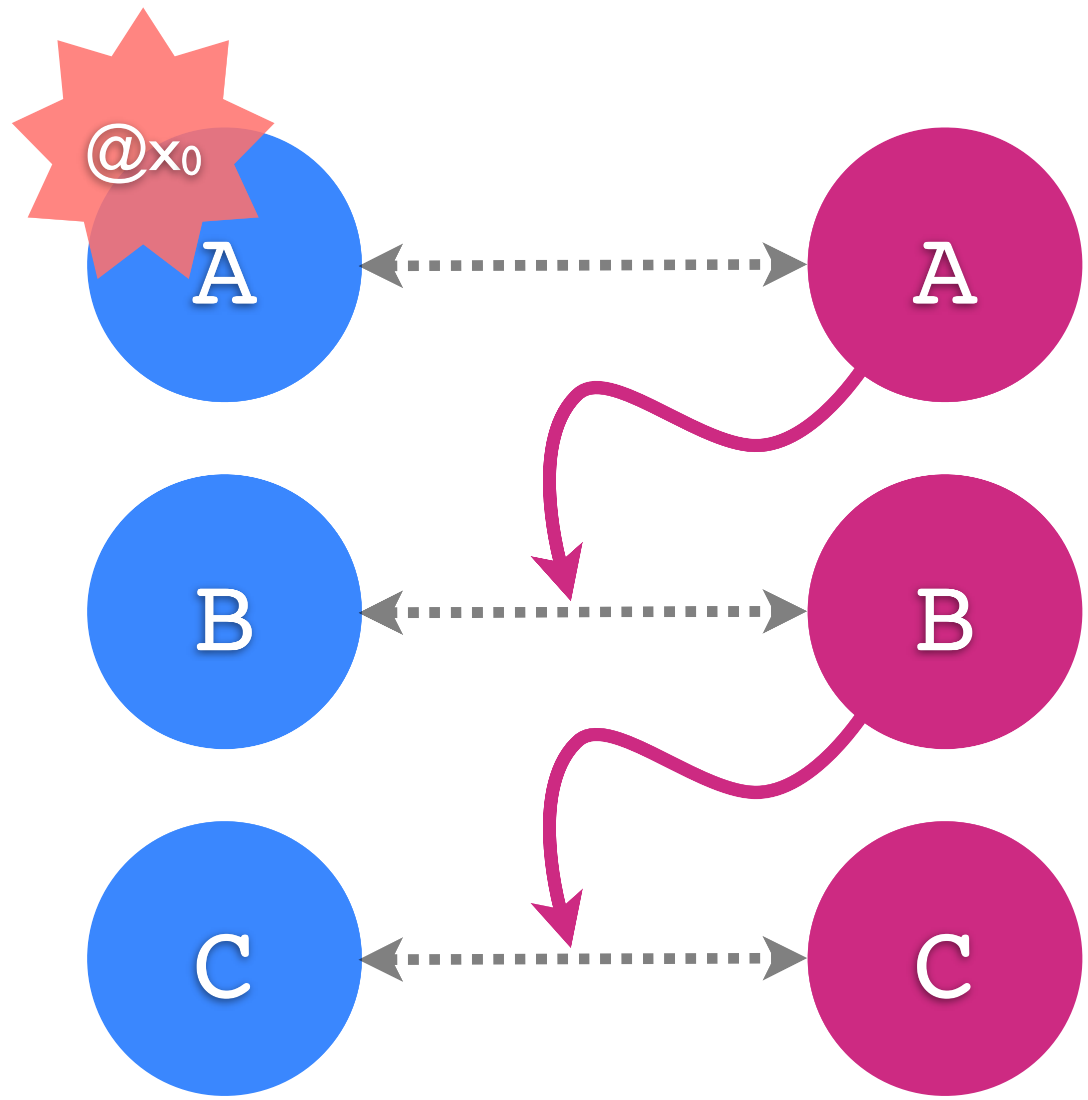
anchoring cascades



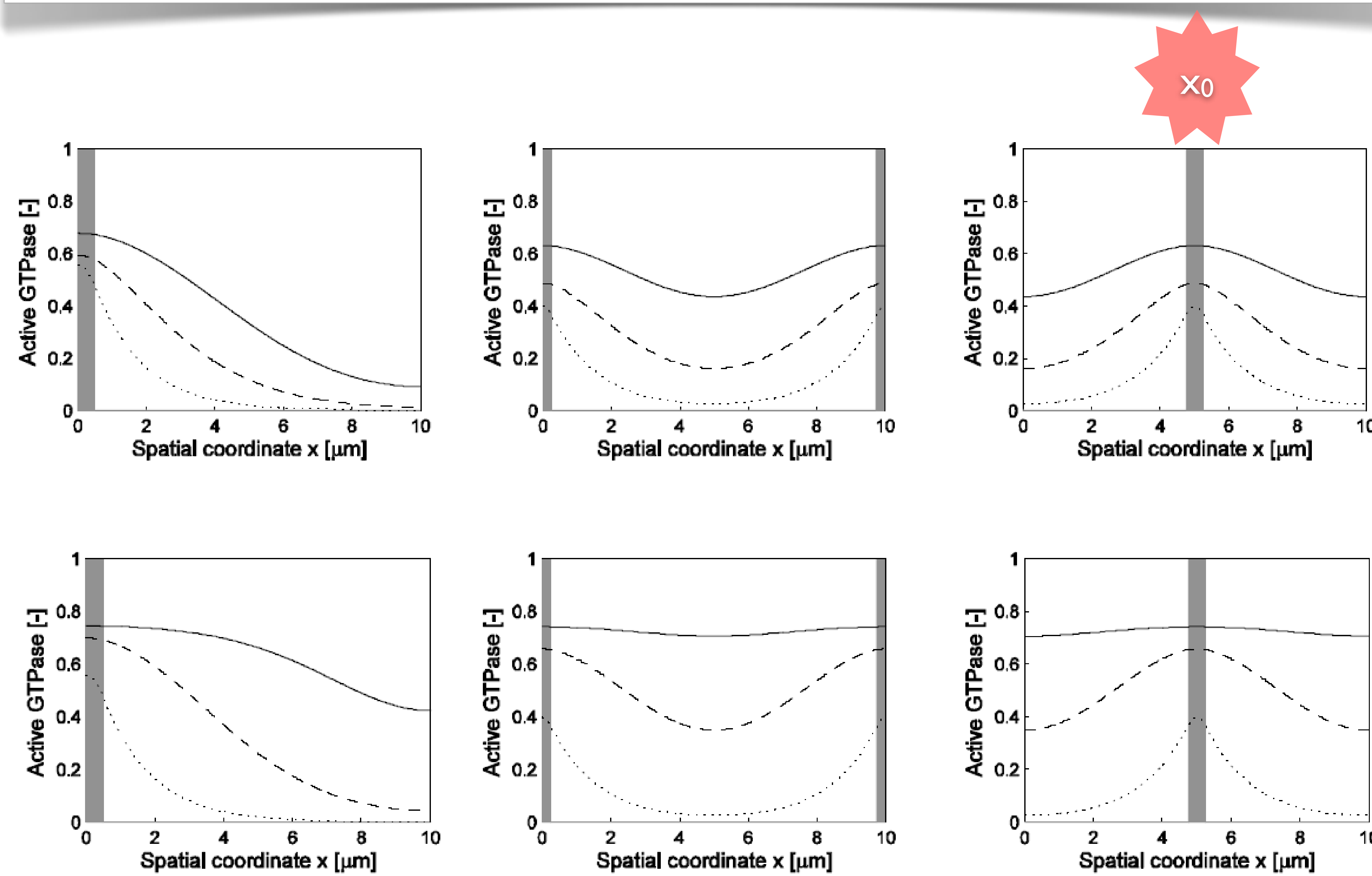
anchoring cascades



first wiring



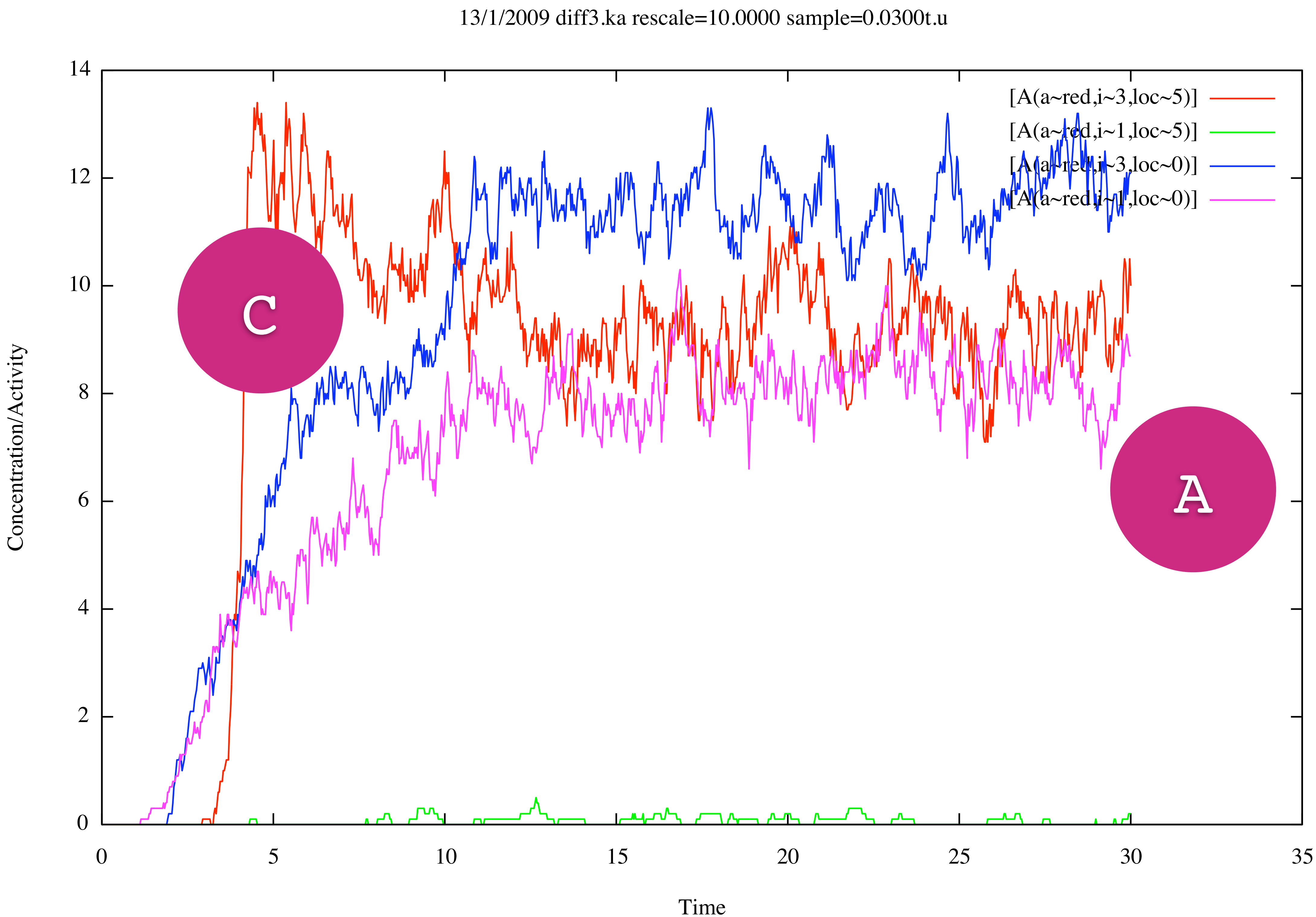
1st wiring: homogeneization



linear regulation (top), MM (bottom)

```
# we rely on id to distinguish agents
A(loc~0) <-> A(loc~1) @ 1,1
A(loc~1) <-> A(loc~2) @ 1,1
A(loc~2) <-> A(loc~3) @ 1,1
A(loc~3) <-> A(loc~4) @ 1,1
A(loc~4) <-> A(loc~5) @ 1,1
A(loc~5) <-> A(loc~6) @ 1,1
A(loc~6) <-> A(loc~7) @ 1,1
A(loc~7) <-> A(loc~8) @ 1,1
A(loc~8) <-> A(loc~9) @ 1,1

# uniform cooling reaction
A(a~red) -> A(a~blue) @ 1

# activation source located at loc=0..9
A(i~1,loc~0,a~blue) -> A(i~1,loc~0,a~red) @ 10

# the remainder of the cascade players are diffusible
A(i~1,loc~0,a~red), A(i~2,loc~0,a~blue) -> A(i~1,loc~0,a~red), A(i~2,loc~0,a~red) @ 10
A(i~1,loc~1,a~red), A(i~2,loc~1,a~blue) -> A(i~1,loc~1,a~red), A(i~2,loc~1,a~red) @ 10
A(i~1,loc~2,a~red), A(i~2,loc~2,a~blue) -> A(i~1,loc~2,a~red), A(i~2,loc~2,a~red) @ 10
A(i~1,loc~3,a~red), A(i~2,loc~3,a~blue) -> A(i~1,loc~3,a~red), A(i~2,loc~3,a~red) @ 10
A(i~1,loc~4,a~red), A(i~2,loc~4,a~blue) -> A(i~1,loc~4,a~red), A(i~2,loc~4,a~red) @ 10
A(i~1,loc~5,a~red), A(i~2,loc~5,a~blue) -> A(i~1,loc~5,a~red), A(i~2,loc~5,a~red) @ 10
A(i~1,loc~6,a~red), A(i~2,loc~6,a~blue) -> A(i~1,loc~6,a~red), A(i~2,loc~6,a~red) @ 10
A(i~1,loc~7,a~red), A(i~2,loc~7,a~blue) -> A(i~1,loc~7,a~red), A(i~2,loc~7,a~red) @ 10
A(i~1,loc~8,a~red), A(i~2,loc~8,a~blue) -> A(i~1,loc~8,a~red), A(i~2,loc~8,a~red) @ 10
A(i~1,loc~9,a~red), A(i~2,loc~9,a~blue) -> A(i~1,loc~0,a~red), A(i~2,loc~0,a~red) @ 10

A(i~2,loc~0,a~red), A(i~3,loc~0,a~blue) -> A(i~2,loc~0,a~red), A(i~3,loc~0,a~red) @ 10
A(i~2,loc~1,a~red), A(i~3,loc~1,a~blue) -> A(i~2,loc~1,a~red), A(i~3,loc~1,a~red) @ 10
A(i~2,loc~2,a~red), A(i~3,loc~2,a~blue) -> A(i~2,loc~2,a~red), A(i~3,loc~2,a~red) @ 10
A(i~2,loc~3,a~red), A(i~3,loc~3,a~blue) -> A(i~2,loc~3,a~red), A(i~3,loc~3,a~red) @ 10
A(i~2,loc~4,a~red), A(i~3,loc~4,a~blue) -> A(i~2,loc~4,a~red), A(i~3,loc~4,a~red) @ 10
A(i~2,loc~5,a~red), A(i~3,loc~5,a~blue) -> A(i~2,loc~5,a~red), A(i~3,loc~5,a~red) @ 10
A(i~2,loc~6,a~red), A(i~3,loc~6,a~blue) -> A(i~2,loc~6,a~red), A(i~3,loc~6,a~red) @ 10
A(i~2,loc~7,a~red), A(i~3,loc~7,a~blue) -> A(i~2,loc~7,a~red), A(i~3,loc~7,a~red) @ 10
A(i~2,loc~8,a~red), A(i~3,loc~8,a~blue) -> A(i~2,loc~8,a~red), A(i~3,loc~8,a~red) @ 10
A(i~2,loc~9,a~red), A(i~3,loc~9,a~blue) -> A(i~2,loc~0,a~red), A(i~3,loc~0,a~red) @ 10


%init: 100 * (A(i~1,a~blue,loc~5))
%init: 100 * (A(i~2,a~blue,loc~5))
%init: 100 * (A(i~3,a~blue,loc~5))

%obs: A(i~1,a~red,loc~0)
%obs: A(i~2,a~red,loc~0)
%obs: A(i~3,a~red,loc~0)
%obs: A(i~1,a~red,loc~5)
%obs: A(i~2,a~red,loc~5)
%obs: A(i~3,a~red,loc~5)
```

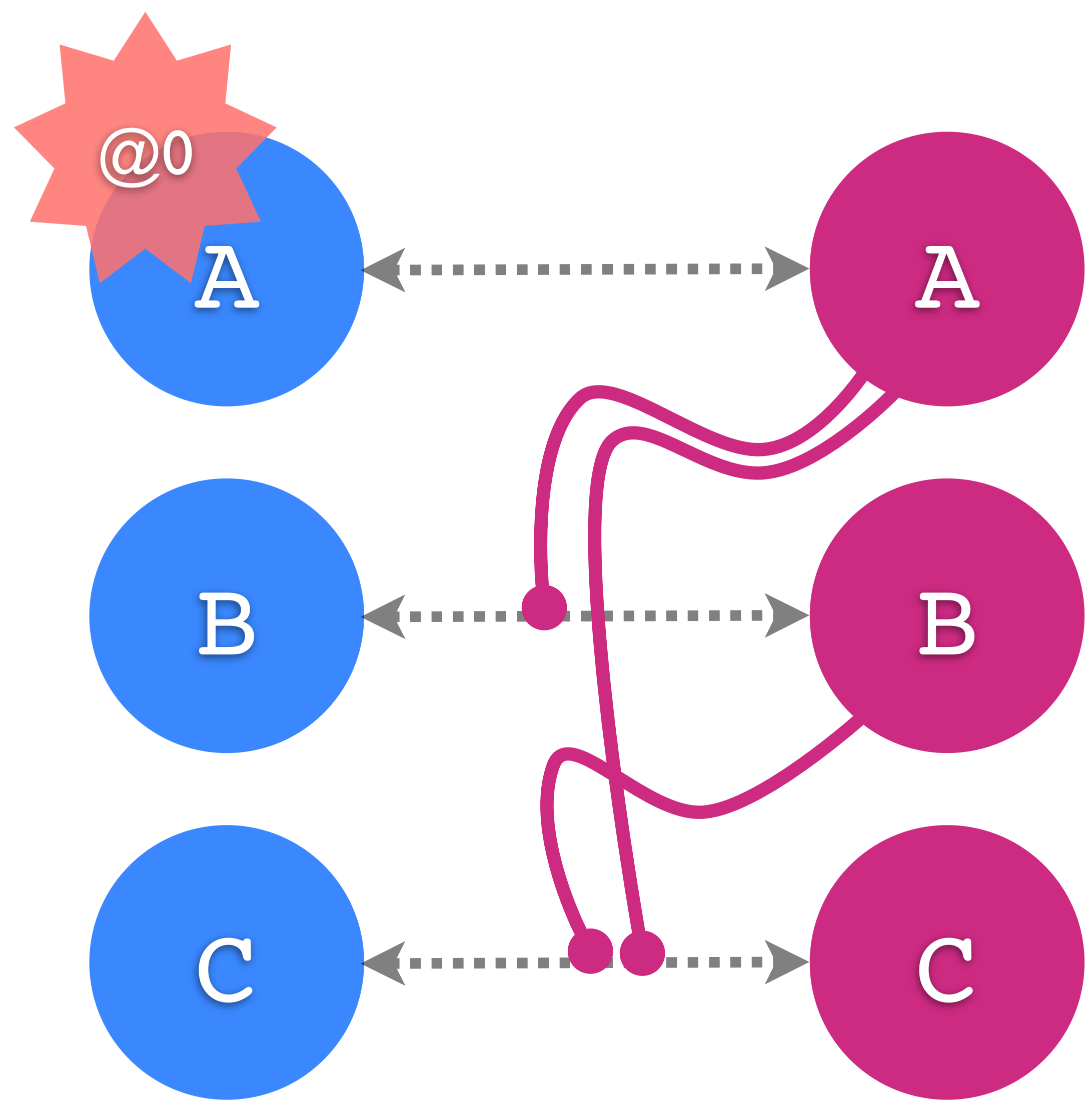
stochastic simulation



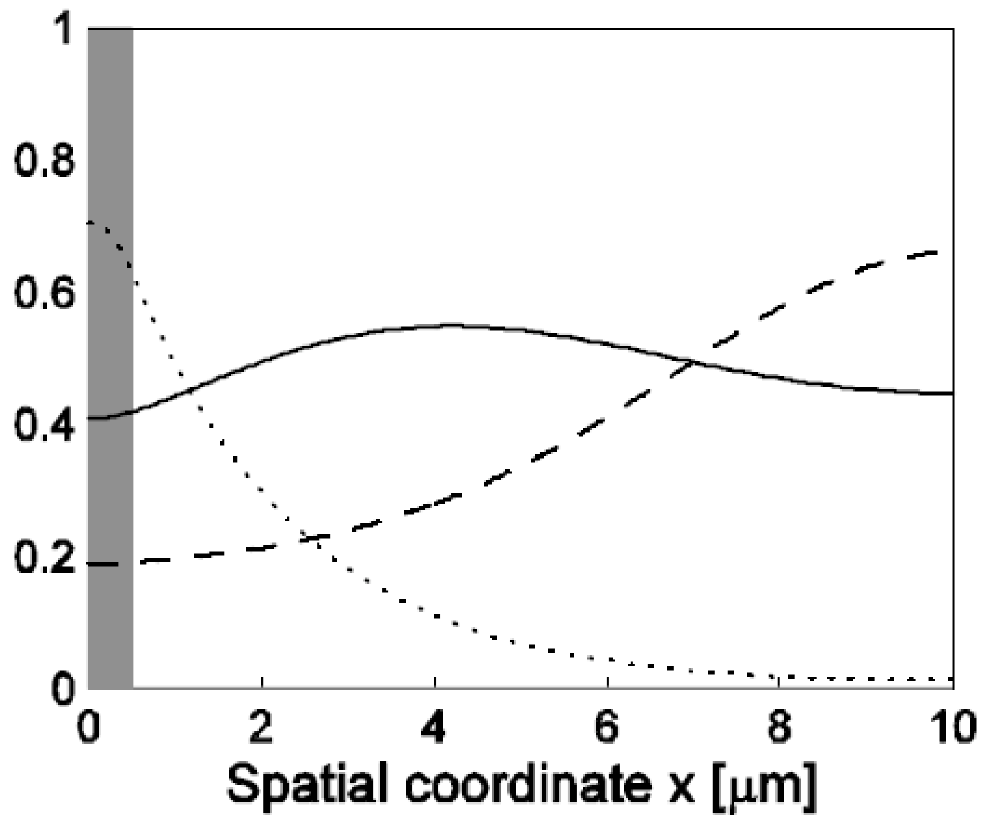
Spatial Kappa simulator v1.0.0

https://github.com/donal-s/SpatialKappa

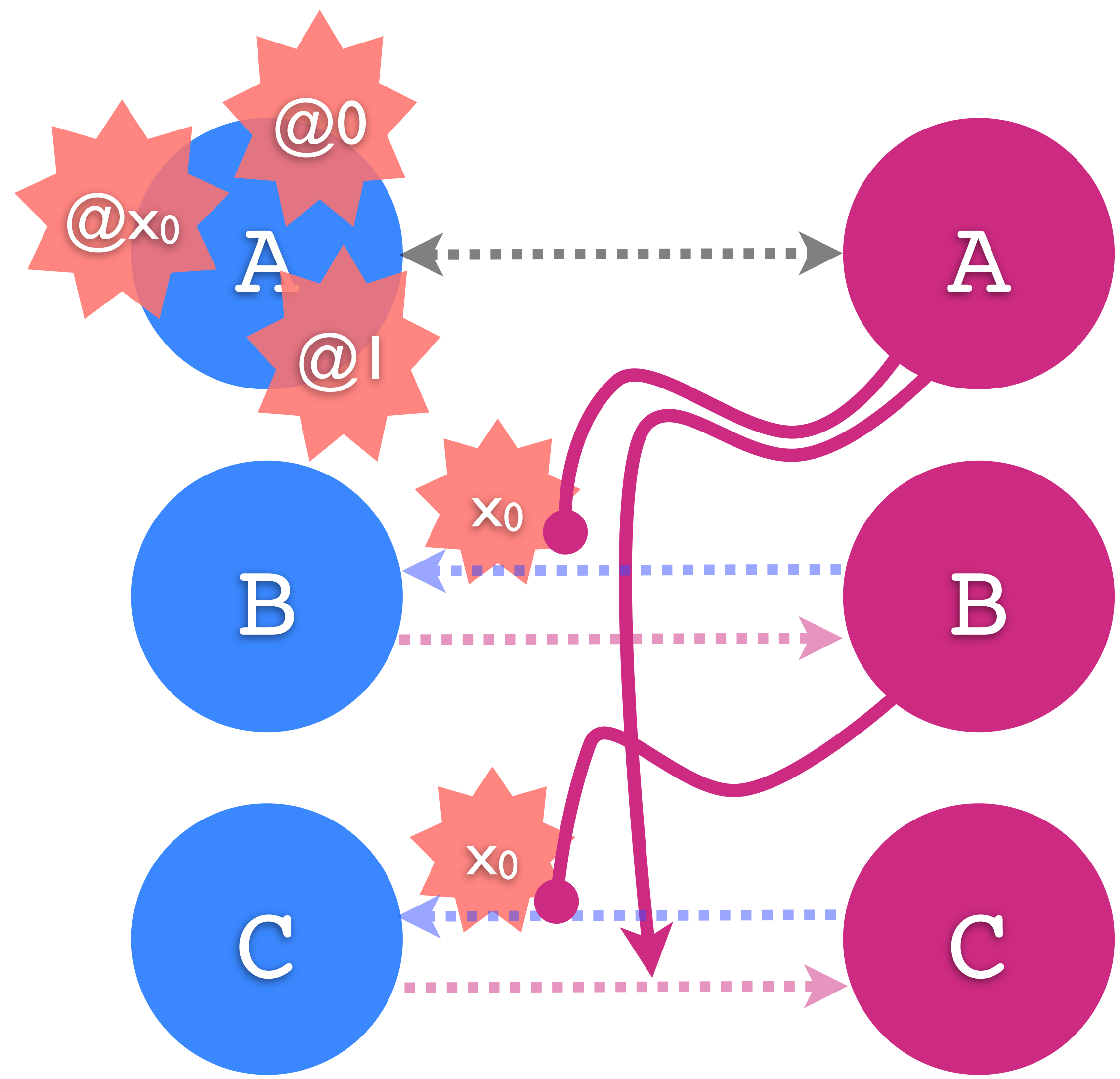
2nd wiring



2nd wiring: non monotic gradients



3rd wiring



3rd wiring: distance sensitive gradients