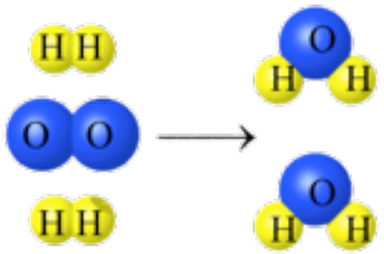
Stochastic Hybrid Analysis of Markov Population Models

Verena Wolf, Saarland University

Markov Chains with Population Structure

Queueing networks => many performance models of communication & computer networks

Models of chemical reaction networks



... (every Markov model with "counter variables", small jump distances, "density-dependent" transition rates)

Deterministic Approximation

popular tool: make state space continuous and approximate discrete jumps by continuous flow

=> mean-field approximation

=> fluid analysis

=> reaction rate equations

=> 1st order moment closure

Approximation of the (co-)variances

=> 2nd order moment closure

Deterministic Approximation

popular tool: make state space continuous and approximate discrete jumps by continuous flow

=> mean-field approximation

=> fluid analysis

=> reaction rate equations

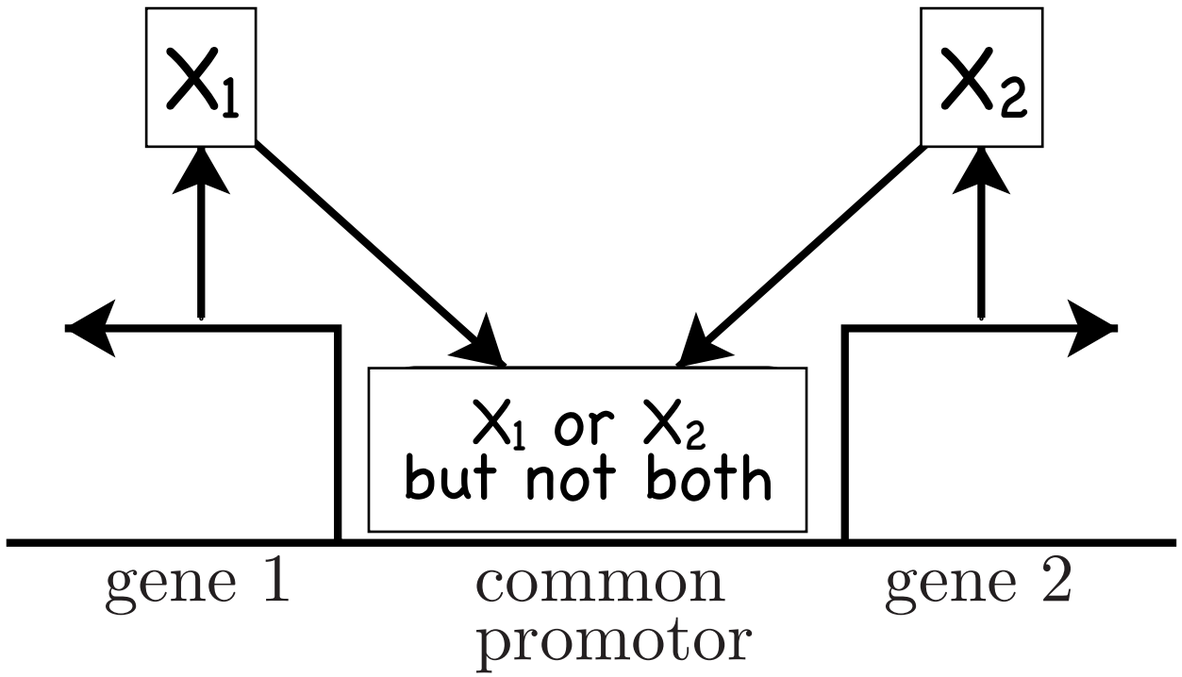
=> 1st order moment closure

Approximation of the (co-)variances

=> 2nd order moment closure

but: what if discreteness matters???

Example: Exclusive Switch



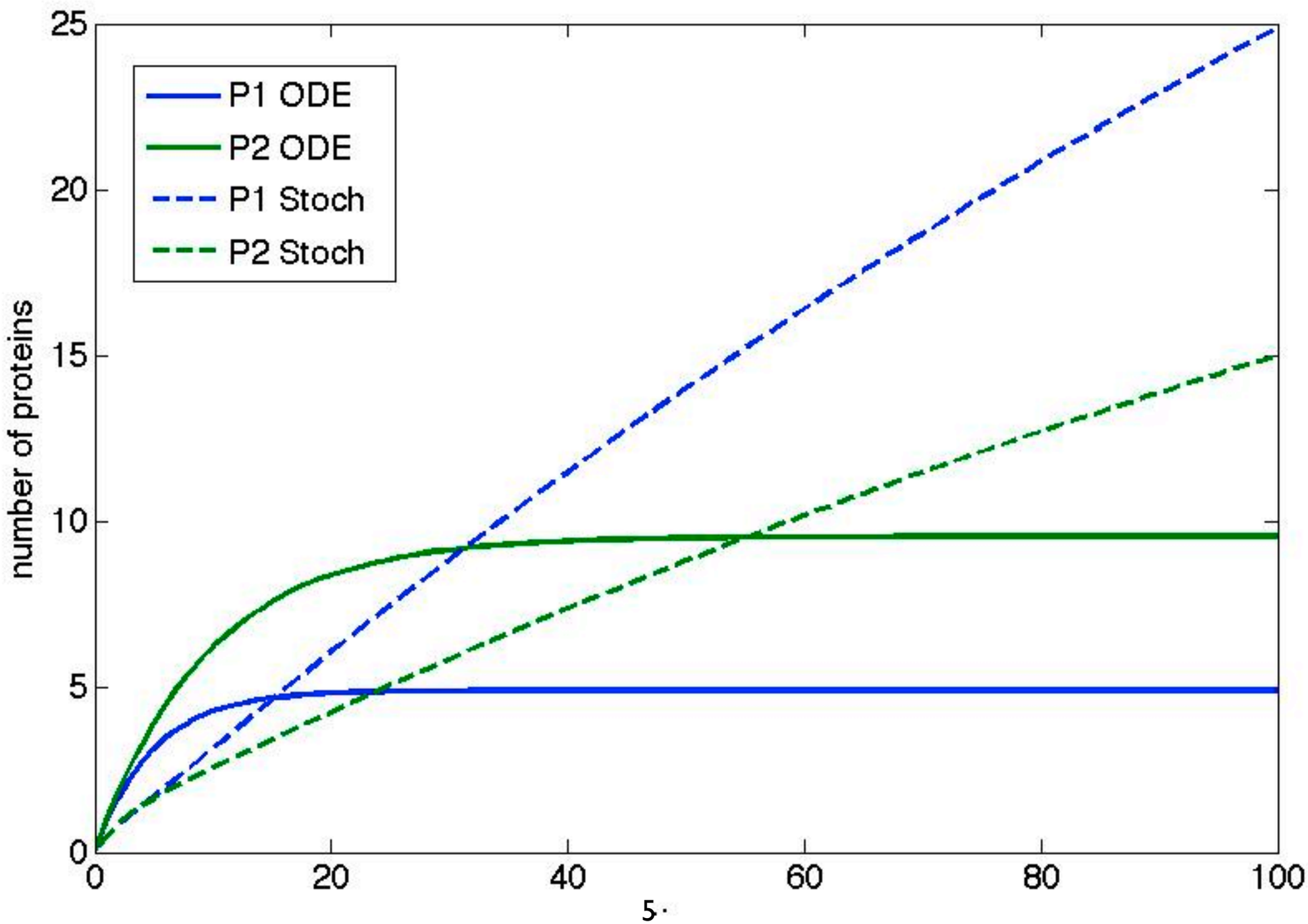
State variables:

promotor: free | X_1 bound | X_2 bound

populations of X_1 and X_2

Example: Exclusive Switch

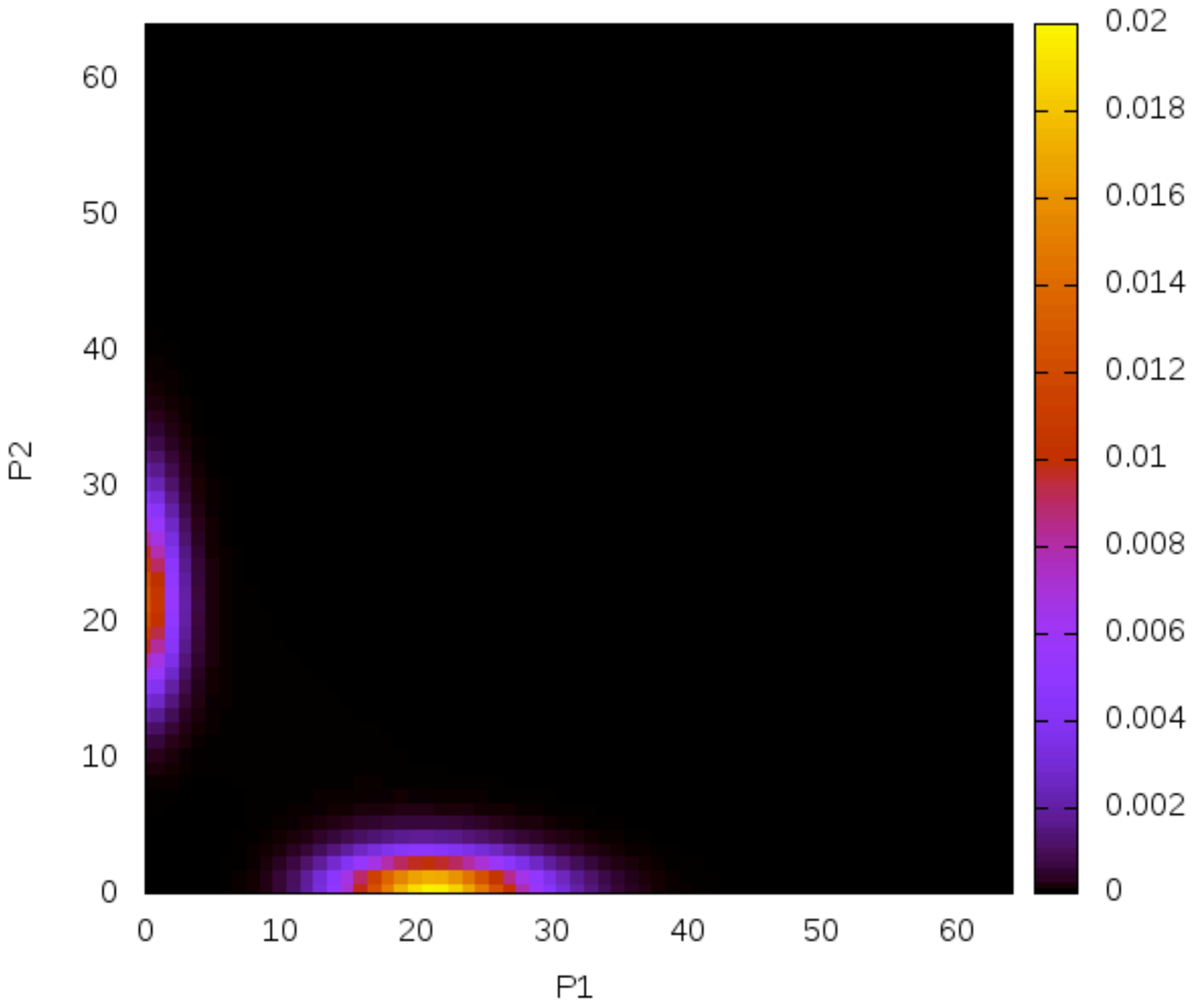
1 copy of each gene



Example: Exclusive Switch

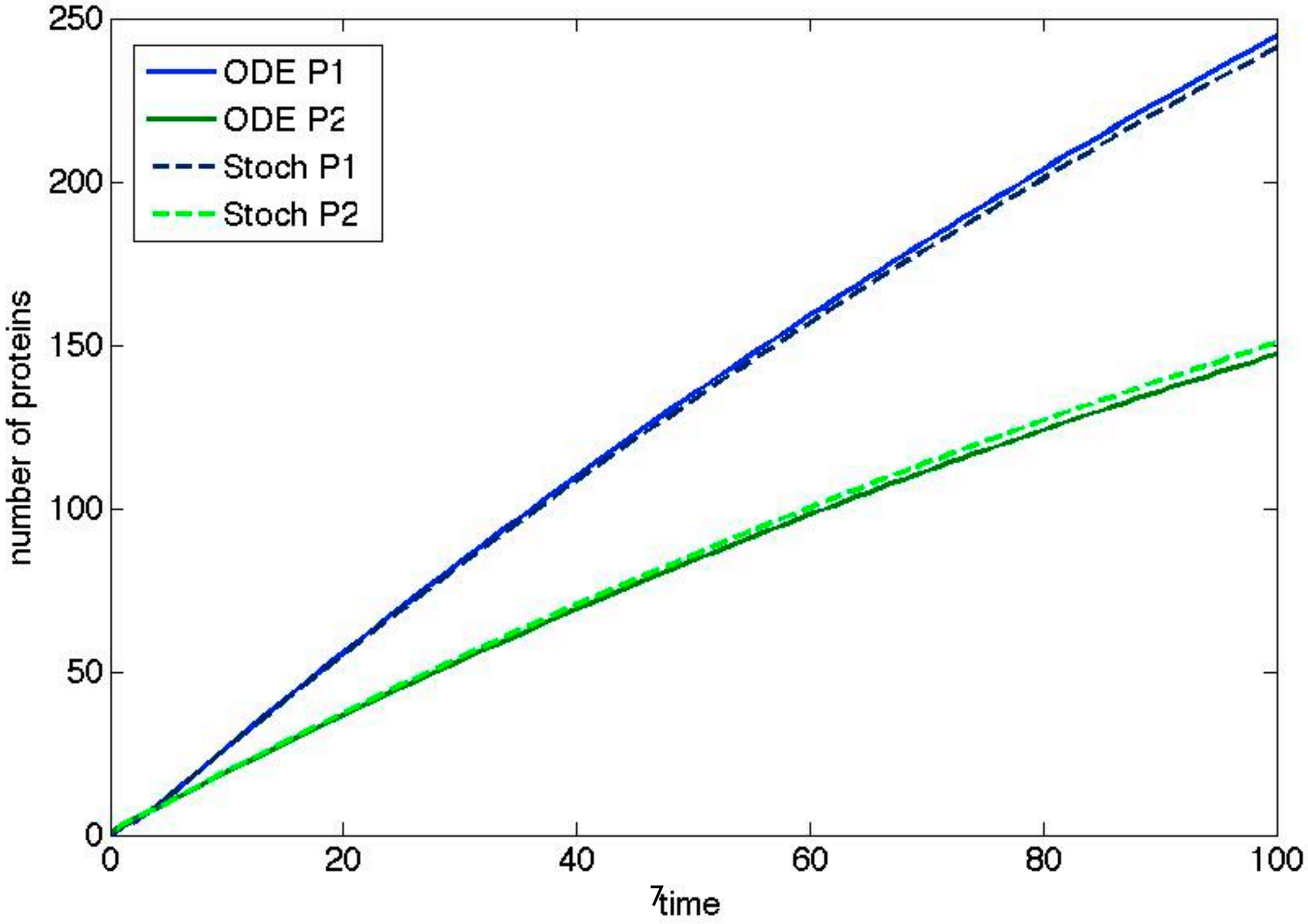
1 copy of each gene

probability distribution at time 50



Example: Exclusive Switch

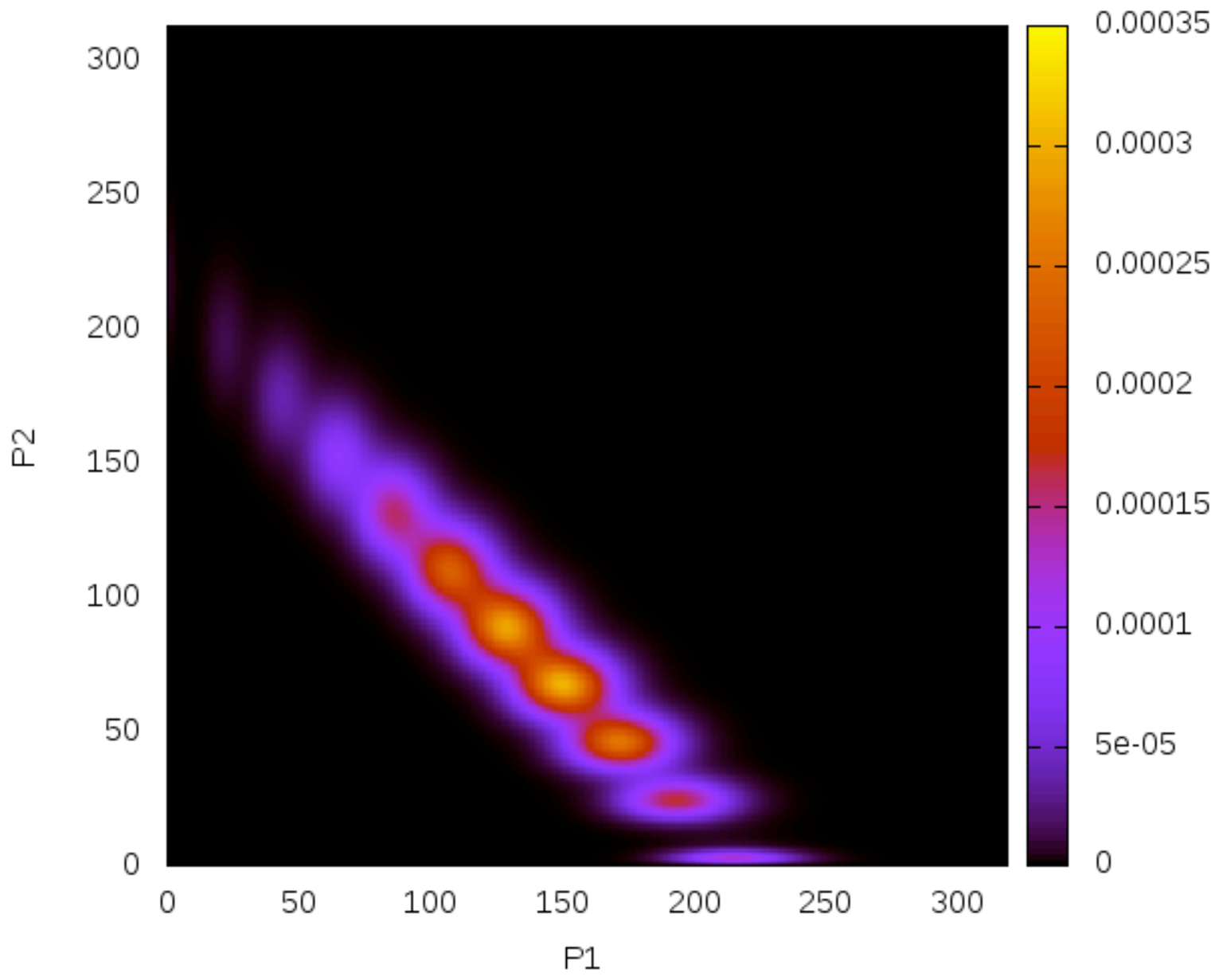
10 copies of each gene



Example: Exclusive Switch

10 copies of each gene

probability distribution at time 50



Stochastic hybrid approach

- keep small populations discrete stochastic
- make large populations continuous (with stochastic or deterministic dynamics)

Example: Exclusive Switch

discrete state (MODE) changes of the promotor

Example: Exclusive Switch

ODE

$$d/dt\ X_1 = k_1 - d_1 X_1 + u_1$$
$$d/dt\ X_2 = k_2 - d_2 X_2 + u_2$$

discrete state (MODE) changes of the promotor

Example: Exclusive Switch

ODE

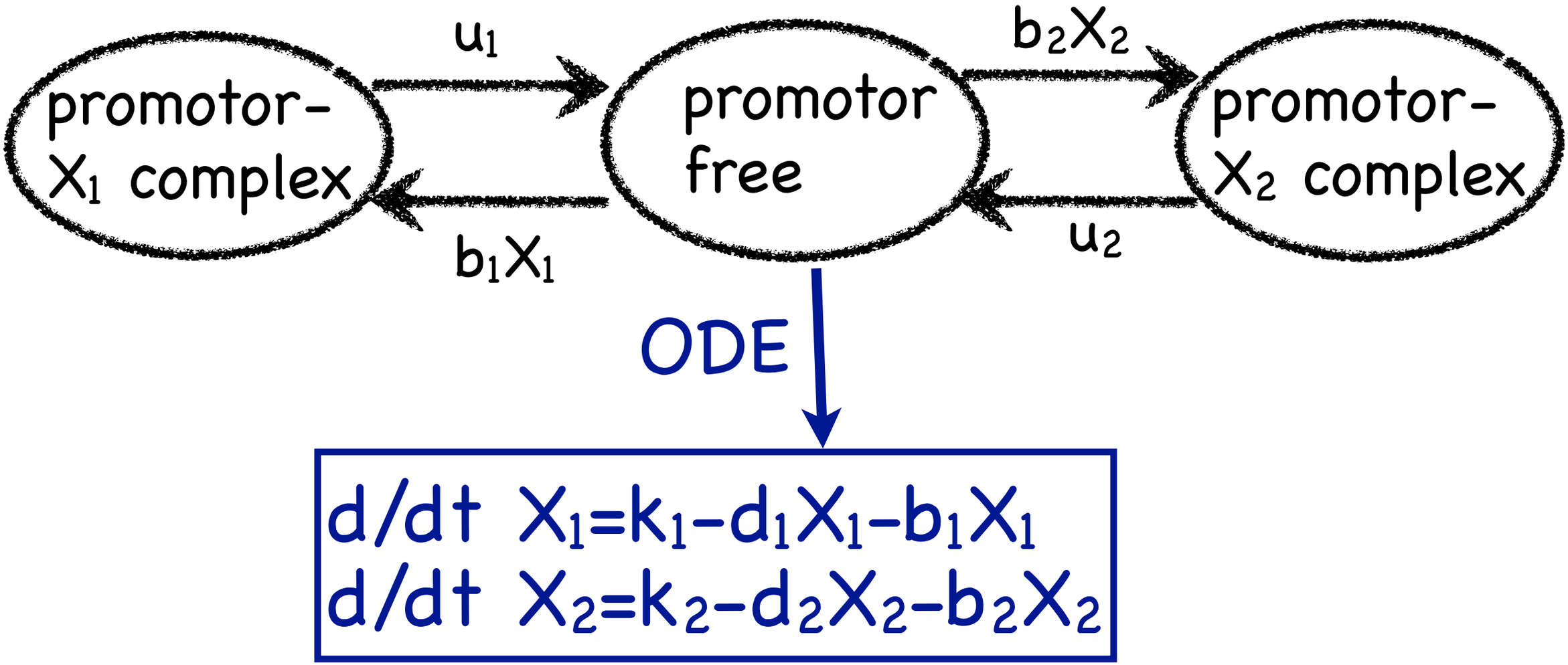
$$d/dt\ X_1 = k_1 - d_1X_1 + u_1$$
$$d/dt\ X_2 = k_2 - d_2X_2 + u_2$$

include updates of discrete jumps in ODE if populations are large => continuous trajectories

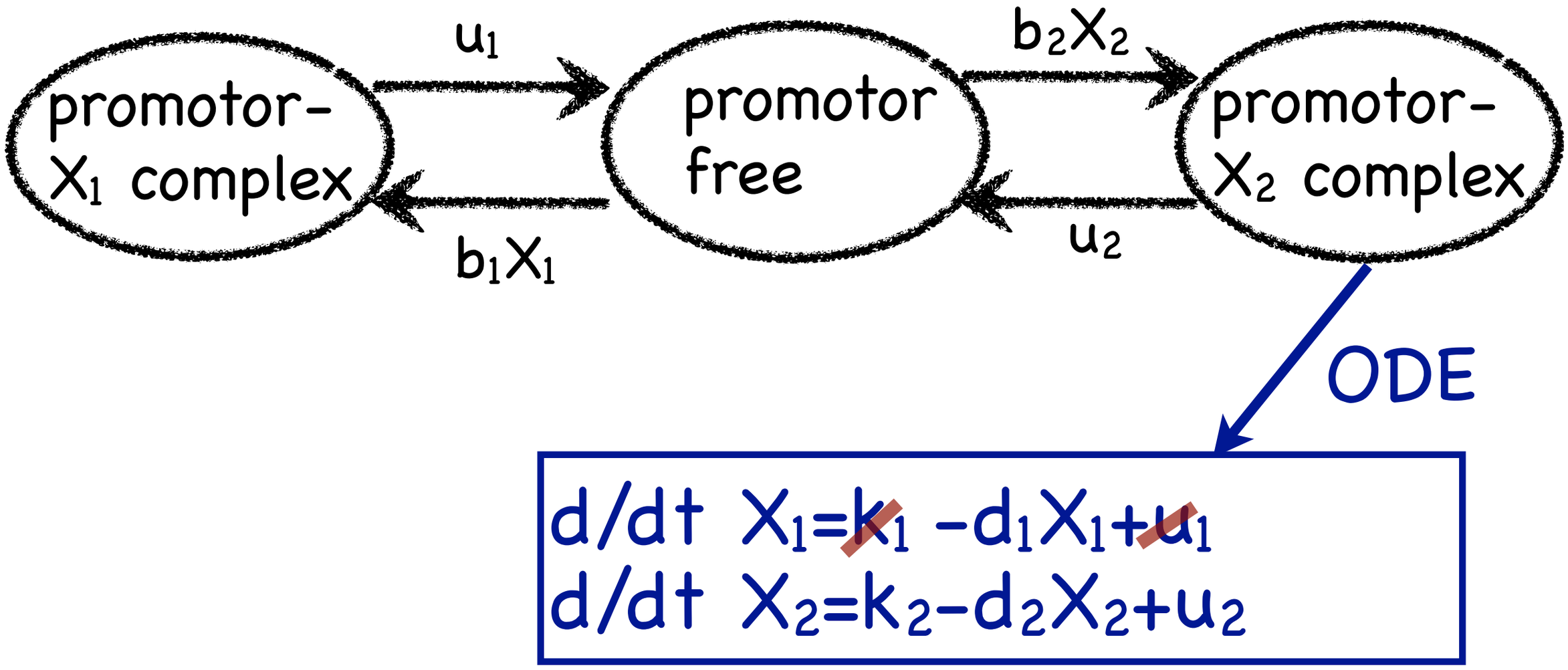
discrete state (MODE) changes of the promotor

Example: Exclusive Switch

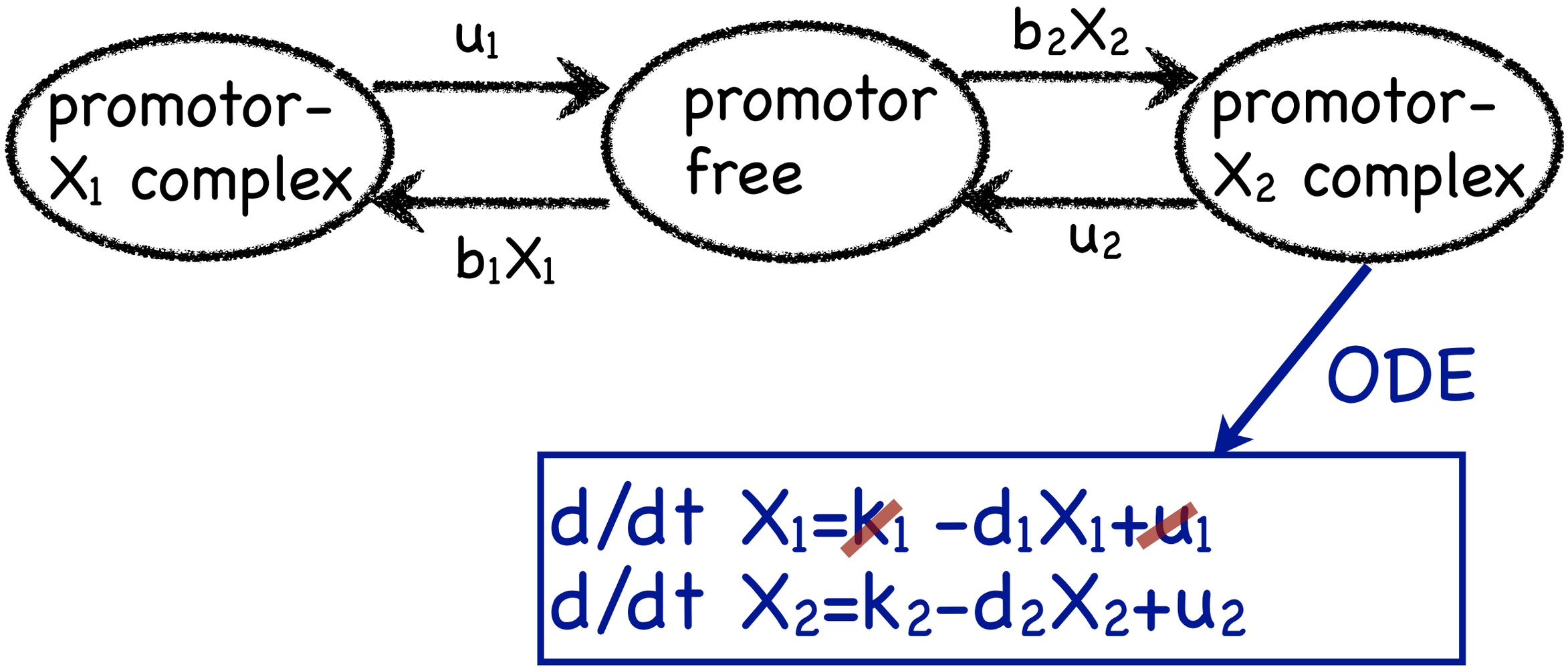
Example: Exclusive Switch



Example: Exclusive Switch



Example: Exclusive Switch



one may add ODEs for the (co-)variances ...

Outlook

- From multistep to hybrid simulation
- Transient numerical solution
- Steady-state solutions and stability analysis

From Multistep to Hybrid Simulation

Multistep Simulation

Several techniques for multistep simulation have been developed in the area of chemical kinetics

- τ-leaping (Gillespie 2001, ...)
- Approximate Simulation (Haseltine and Rawlings 2002)
- Hybrid Stochastic Simulation (Salis and Kaznessis 2005)
- ...

Multistep Simulation

Several techniques for multistep simulation have been developed in the area of chemical kinetics

- τ-leaping (Gillespie 2001, ...)
- Approximate Simulation (Haseltine and Rawlings 2002)
- Hybrid Stochastic Simulation (Salis and Kaznessis 2005)
- ...

For Monte-Carlo simulation discreteness is not a problem, but stiffness is!

Multiscale Problem

For direct numerical simulations (= approximations of the probability distributions):

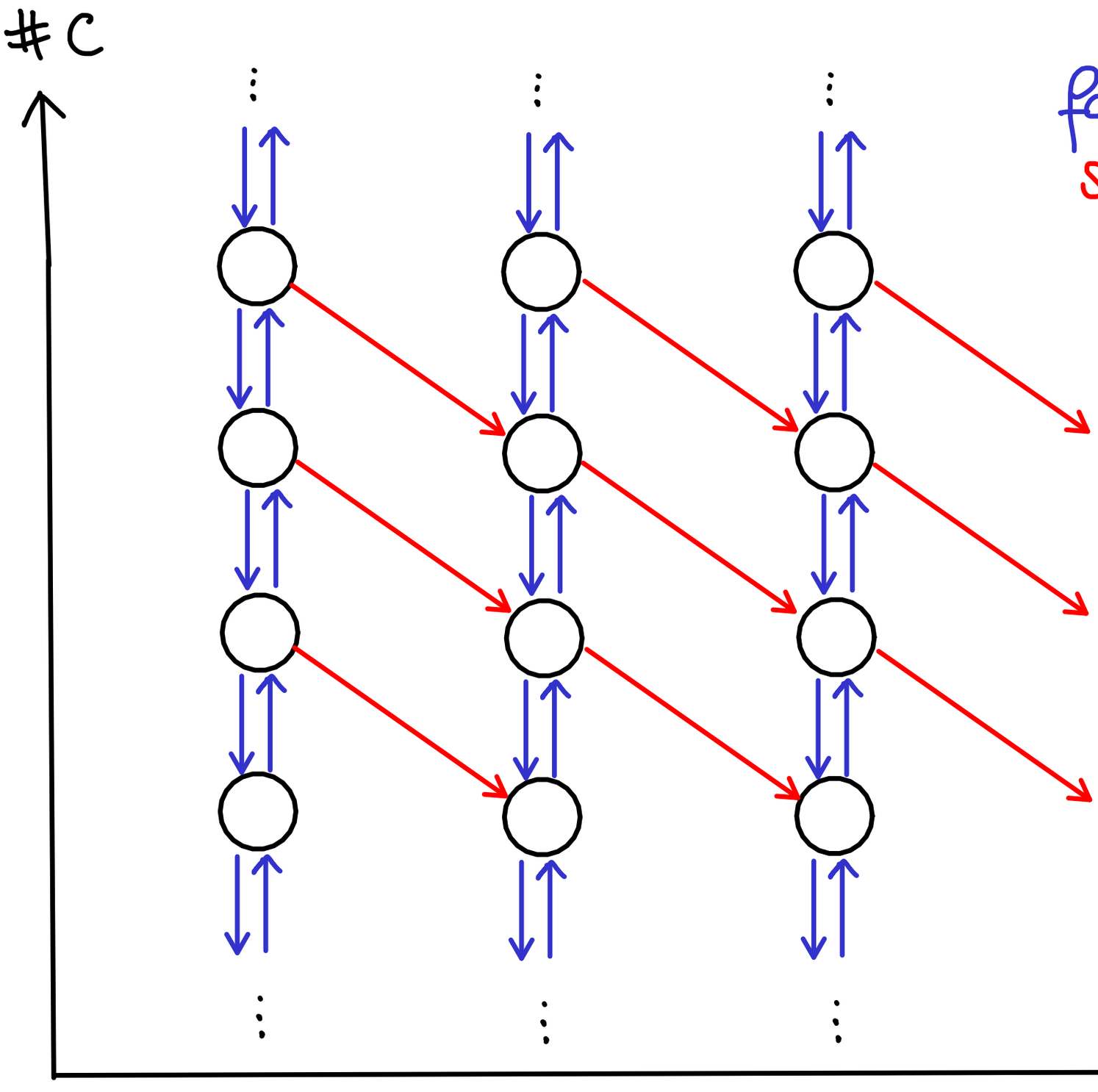
=> one may use a stochastic hybrid approach because

(1) populations are large, keeping variables discrete is expensive (state space explosion)

(2) model is stiff and simulation is very slow (step-size of numerical integration is too small)

often we have both!

Stiffness in Enzyme Kinetics



fast
slow

$$E + S \xrightarrow{c_1} C$$

$$C \xrightarrow{c_2} E + S$$

$$C \xrightarrow{c_3} E + P$$

many complex formations/dissociations must occur until a product is formed

Multistep simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

Multistep simulation

time var

system state

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

Multistep s

transition rate of type i event (which changes the populations) e.g. chemical reaction, arrival of a customer

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

Multistep simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

random var for number type i events within next τ time units

Multistep simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

realizations of $K_1,\ldots,K_R$

Multistep simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_m$ for $K_1,\ldots,K_m$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

change vector of type i events

Multistep simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_R$ for $K_1,\ldots,K_R$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

direct multistepping: use Poisson distribution (parameter $\alpha_i(x)\tau$) to estimate $k_1,\ldots,k_R$

explicit τ-leaping: choose time step such that rates do not change much (Gillespie 2001)

Hybrid simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;
2. Choose a step size τ according to some appropriate rule;
3. Compute suitable estimates $k_1,\ldots,k_R$ for $K_1,\ldots,K_R$;
4. Set $t := t + \tau$ and update x as $x=x+\sum v_i k_i$.

If the parameter $\alpha_i(x)\tau$ of the Poisson distribution is large ($\alpha_i(x)\tau >> 1$), then it tends to a normal distribution with mean $\alpha_i(x)\tau$ and variance $\alpha_i(x)\tau$ (Gillespie 2002).

If we forget about the variance, we just use $\alpha_i(x)\tau$

=> deterministic approximation

Hybrid simulation

Init $t:=t_0$, $x:=x_0$ and t_{end};

while $t < t_{end}$

1. Compute all $\alpha_i(x)$ and $\alpha(x):=\alpha_1(x)+\cdots+\alpha_m(x)$;

2. Choose a step size τ according to some appropriate rule;

usually the case if reactant populations are large

$k_1,\ldots,k_R$ for $K_1,\ldots,K_R$;

as $x=x+\sum v_i k_i$.

If the parameter $\alpha_i(x)\tau$ of the Poisson distribution is large ($\alpha_i(x)\tau >> 1$), then it tends to a normal distribution with mean $\alpha_i(x)\tau$ and variance $\alpha_i(x)\tau$ (Gillespie 2002).

If we forget about the variance, we just use $\alpha_i(x)\tau$

=> deterministic approximation

Stochastic Hybrid Simulation

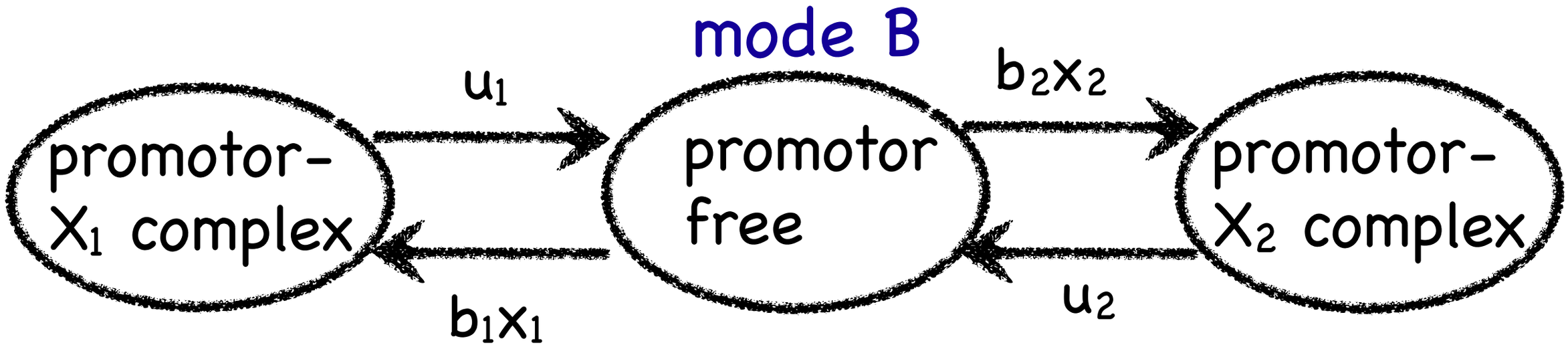
How long do we stay in a mode until we change the mode?

$d/dt\ x_1 = k_1 - d_1 x_1 + u_1$
$d/dt\ x_2 = -d_2 x_2$

for mode A:
exit rate $\lambda = u_1$ is independent of evolution of x_1 and x_2
=> exponential distributed delay with parameter $-u_1$

Stochastic Hybrid Simulation

How long do we stay in a mode until we change the mode?



$d/dt\ x_1=k_1-d_1x_1+b_1x_1$
$d/dt\ x_2=k_2-d_2x_2+b_2x_2$

for mode B: exit rate

$\lambda(s)=b_1x_1(s)+b_2x_2(s)$

=> delay τ such that

$$P(\tau > t') = \exp\left(-\int_t^{t+t'} \lambda(s)ds\right)$$

Stochastic Hybrid Simulation

How long do we stay in a mode until we change the mode?

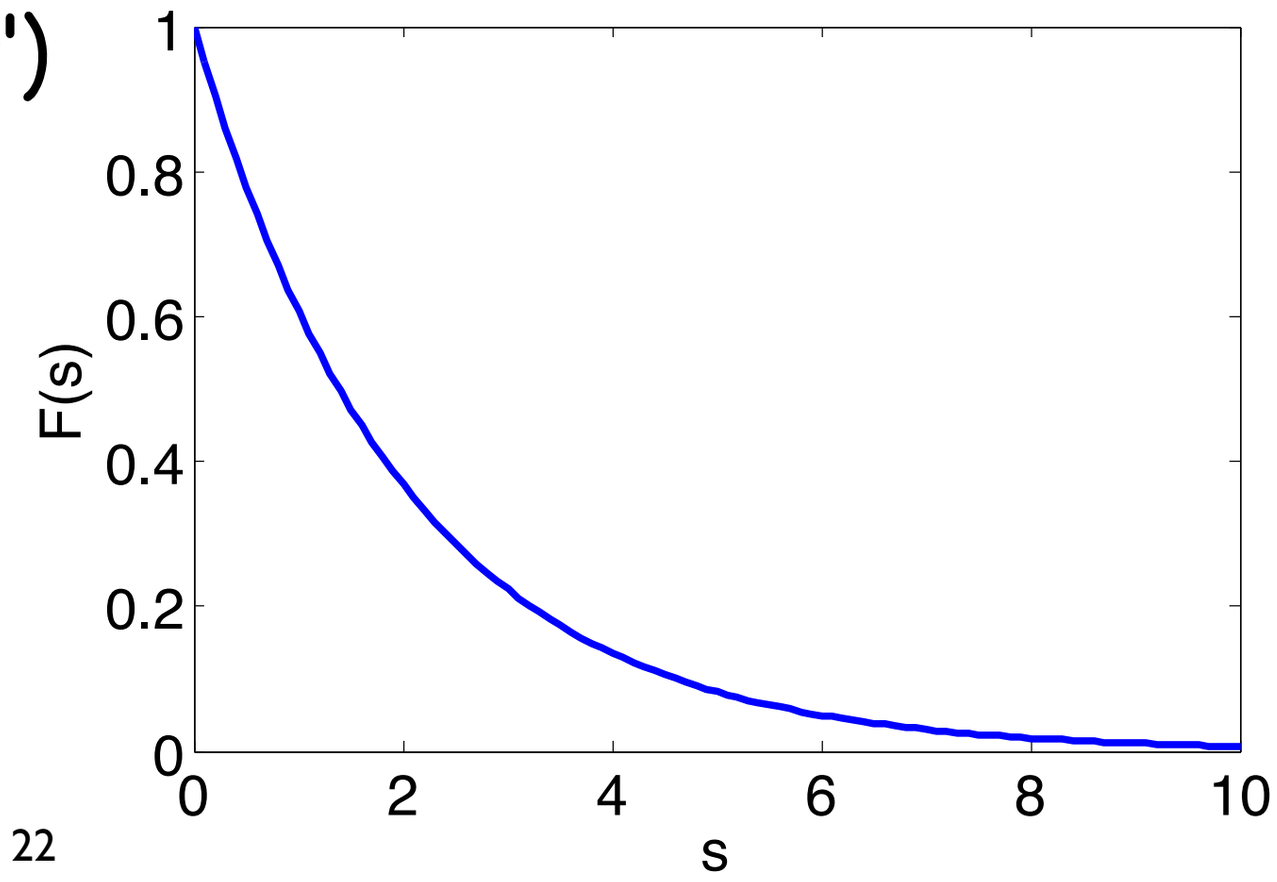
$\lambda(s)=b_1x_1(s)+b_2x_2(s)$

but the evolution of $x_1(s)$ and $x_2(s)$ during $[t,t+\tau]$ is apriori not known =>

exploit that for $F(t')= P(\tau>t')$

$$\frac{d}{ds}F(s) = \lambda(s)F(s)$$

and $F(0) = 1$



Stochastic Hybrid Simulation

How long do we stay in a mode until we change the mode?

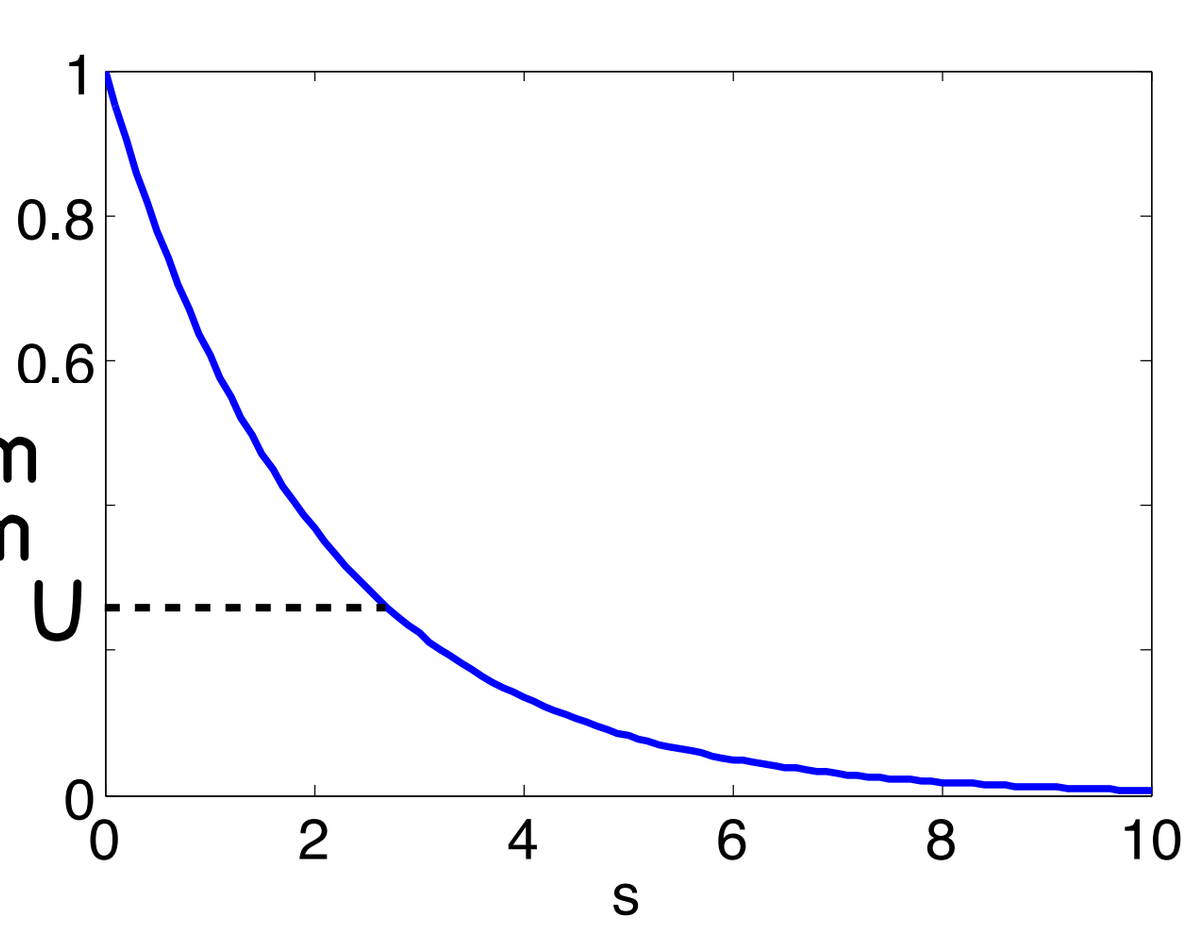
$\lambda(s)=b_1x_1(s)+b_2x_2(s)$

but the evolution of $x_1(s)$ and $x_2(s)$ during $[t,t+\tau]$ is apriori not known =>

exploit that for $F(t')= P(\tau>t')$

$$\frac{d}{ds}F(s) = \lambda(s)F(s)$$

and $F(0) = 1$



Hybrid simulation

Init $t:=t_0$, $x:=x_0$, $m:=m_0$, and t_{end};

while $t < t_{end}$

1. Pick uniformly distributed random number U;
2. Integrate x using ODEs of current mode;

simultaneously integrate F(s) with initial condition F(0)=1;

3. Stopp integration at time τ where $F(\tau)=U$;
4. Decide for next mode accoring to jump rates of current mode m;
5. Set $t := t+\tau$ (and update x according to mode switch)

only of discrete jump rates are not part of ODEs

Transient numerical solution

Transient numerical solution

Why do we care about numerical solutions if Monte-Carlo simulation works well?

- compute the whole probability distribution

- compute probabilities of rare events

- calibrate parameters w.r.t. observations

=> force simulation method to explore certain interesting parts of the state space (even if they are unlikely)!

PDE of the PDF

single continuous variable:

$$p_i(t,x) = \frac{1}{\Delta} \lim_{\Delta \to 0} P\left(M(t) = i, x < X(t) < x + \Delta\right)$$

mode i

continuous variable for protein concentration

$$\frac{\partial}{\partial t} p(x,t) + \frac{\partial}{\partial x} p(x,t) R(x) \quad = \quad p(t,x) Q(x)$$

see "Fluid Stochastic Petri Nets" by Trivedi, Kulkarni, 1998

PDE of the PDF

single continuous variable:

$$p_i(t,x) = \frac{1}{\Delta} \lim_{\Delta \to 0} P\left(M(t) = i, x < X(t) < x + \Delta\right)$$

mode i

continuous variable for protein concentration

$$\frac{\partial}{\partial t} p(x,t) + \frac{\partial}{\partial x} p(x,t) R(x) \quad = \quad p(t,x) Q(x)$$

ODE rates of protein dynamics

jump rates for switching modes

see "Fluid Stochastic Petri Nets" by Trivedi, Kulkarni, 1998

PDE of the PDF

single continuous variable:

$$p_i(t,x) = \frac{1}{\Delta} \lim_{\Delta \to 0} P\left(M(t) = i, x < X(t) < x + \Delta\right)$$

mode i

continuous variable for protein concentration

$$\frac{\partial}{\partial t} p(x,t) + \frac{\partial}{\partial x} p(x,t) R(x) \quad = \quad p(t,x) Q(x)$$

ODE rates of protein dynamics

jump rates for switching modes

see "Fluid Stochastic Petri Nets" by Trivedi, Kulkarni, 1998

Numerical Solution => either discretize continuous part of state space and integrate PDE or ...

Numerical Solution Algorithm (Mateescu, Mikeev, Henzinger, Wolf: CMSB 2010)

In general, split population vector:

- large populations -> deterministic/continuous (DC) dynamics given by ODE (depend on mode)
  (also possible with more moments than just 1st)

Numerical Solution Algorithm (Mateescu, Mikeev, Henzinger, Wolf: CMSB 2010)

In general, split population vector:

- large populations -> deterministic/continuous (DC) dynamics given by ODE (depend on mode) (also possible with more moments than just 1st)

- small populations -> stochastic/discrete (SD) modes; dynamics given by (small) Markov chain

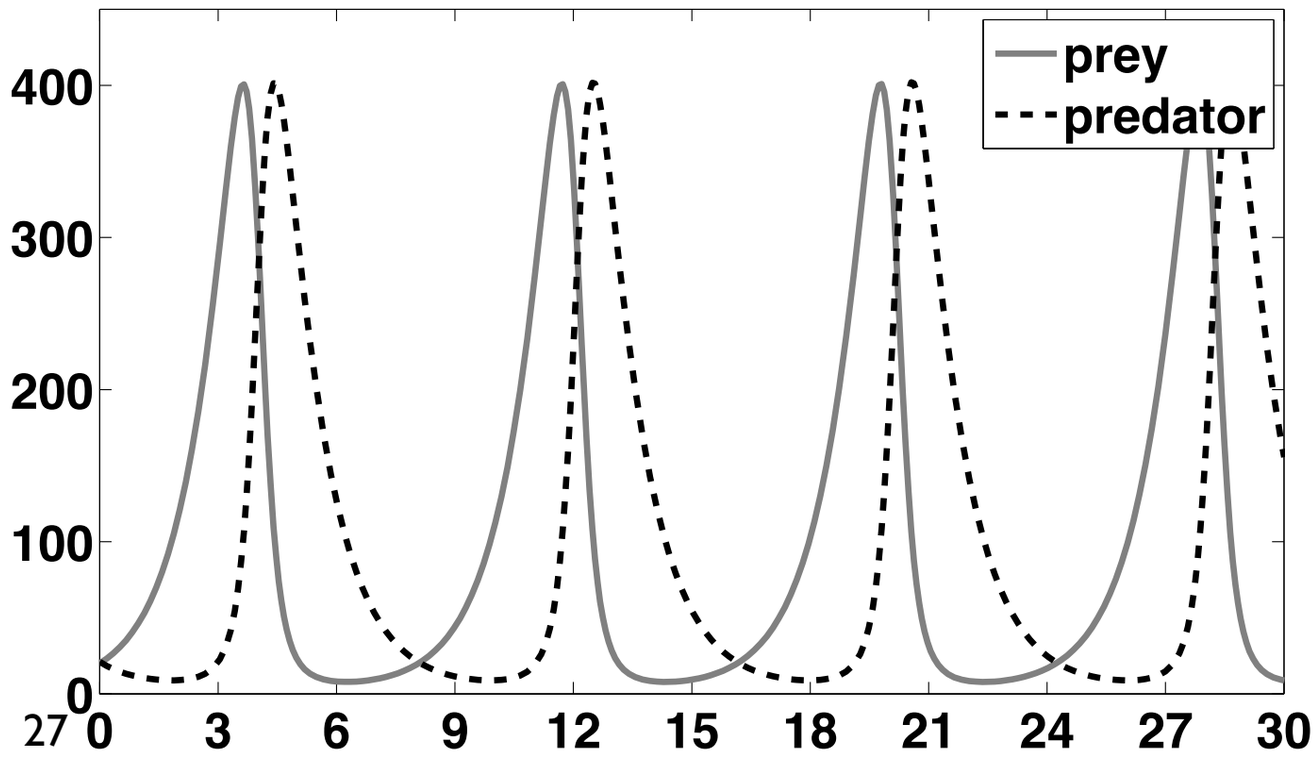
Numerical Solution Algorithm (Mateescu, Mikeev, Henzinger, Wolf: CMSB 2010)

In general, split population vector:

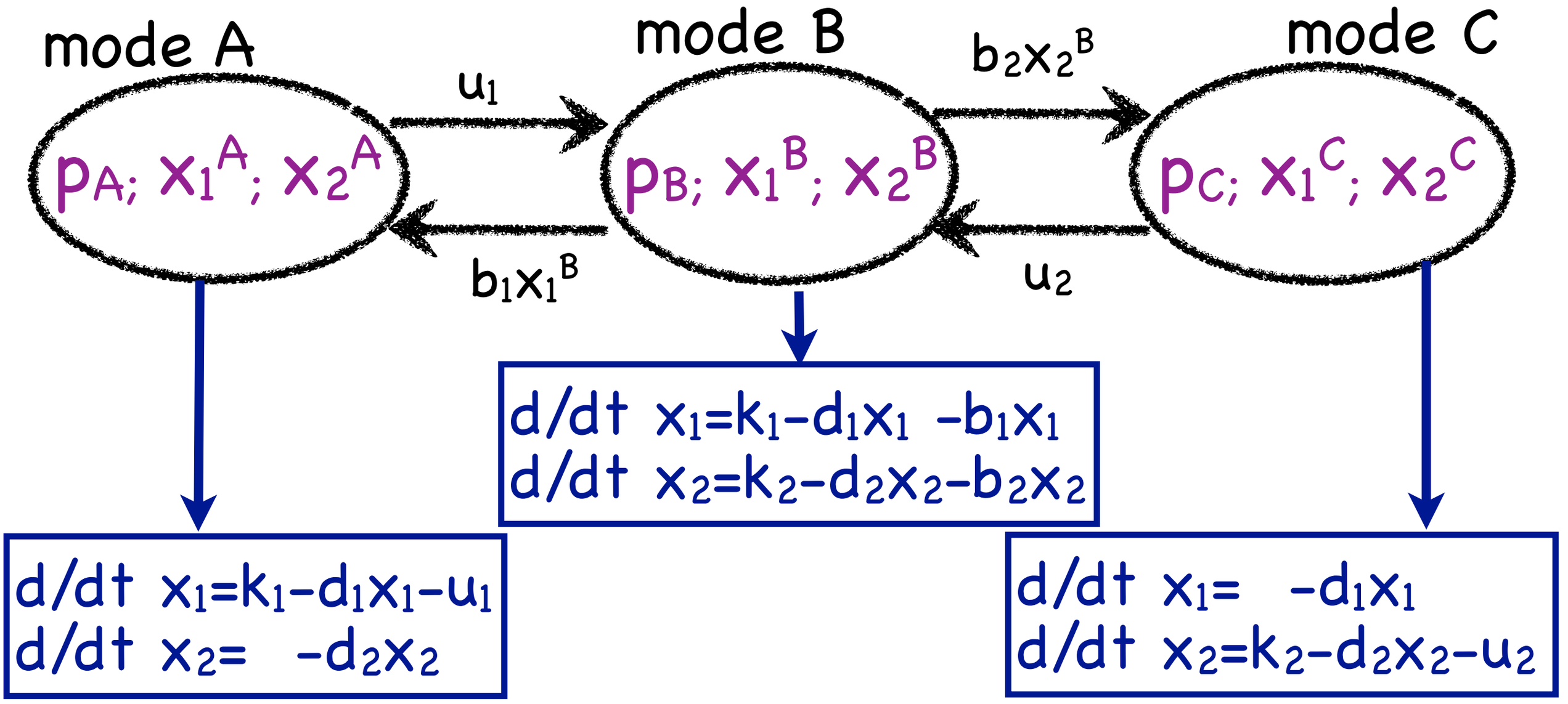
- large populations -> deterministic/continuous (DC) dynamics given by ODE (depend on mode) (also possible with more moments than just 1st)

- small populations -> stochastic/discrete (SD) modes; dynamics given by (small) Markov chain

➡ may switch representations over time



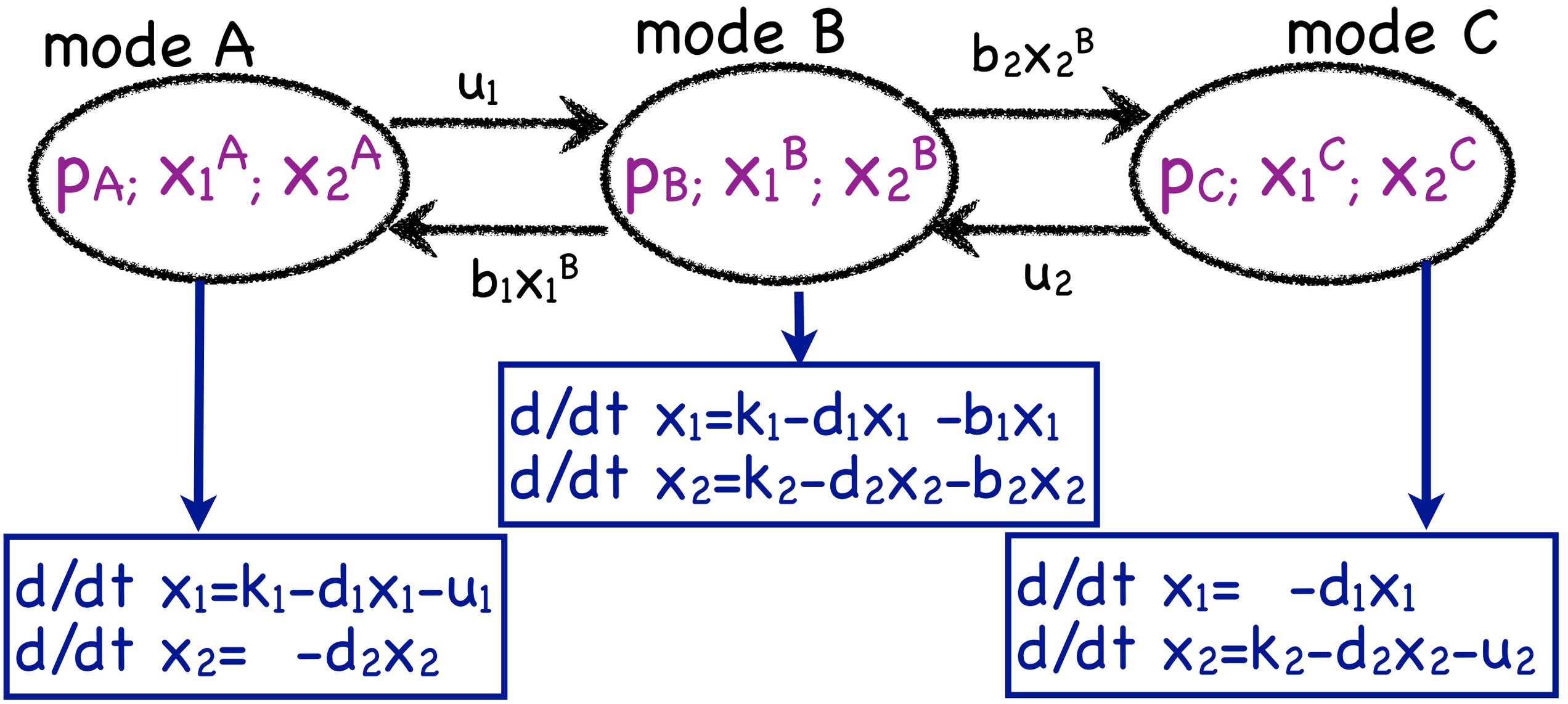
How to integrate over time?



Given at time t: probabilities $p_A+p_B+p_C=1$

and conditional expectations x_i^A, x_i^B, x_i^C (i=1,2)

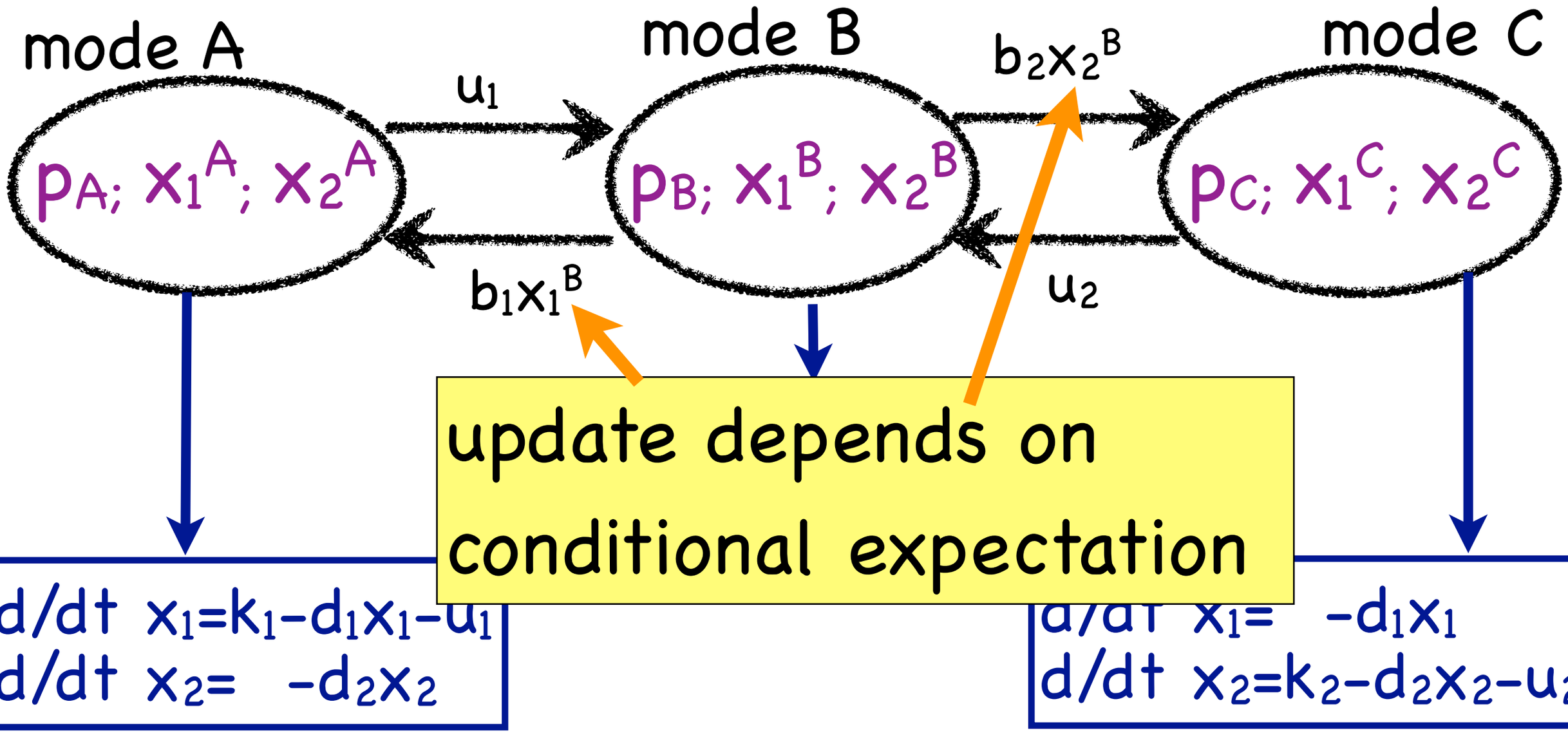
How to integrate over time?



1) integrate <u>probability distribution</u> for small [t,t+h]

$p_A(t) \dashrightarrow p_A(t+h)$ $p_B(t) \dashrightarrow p_B(t+h)$ ....

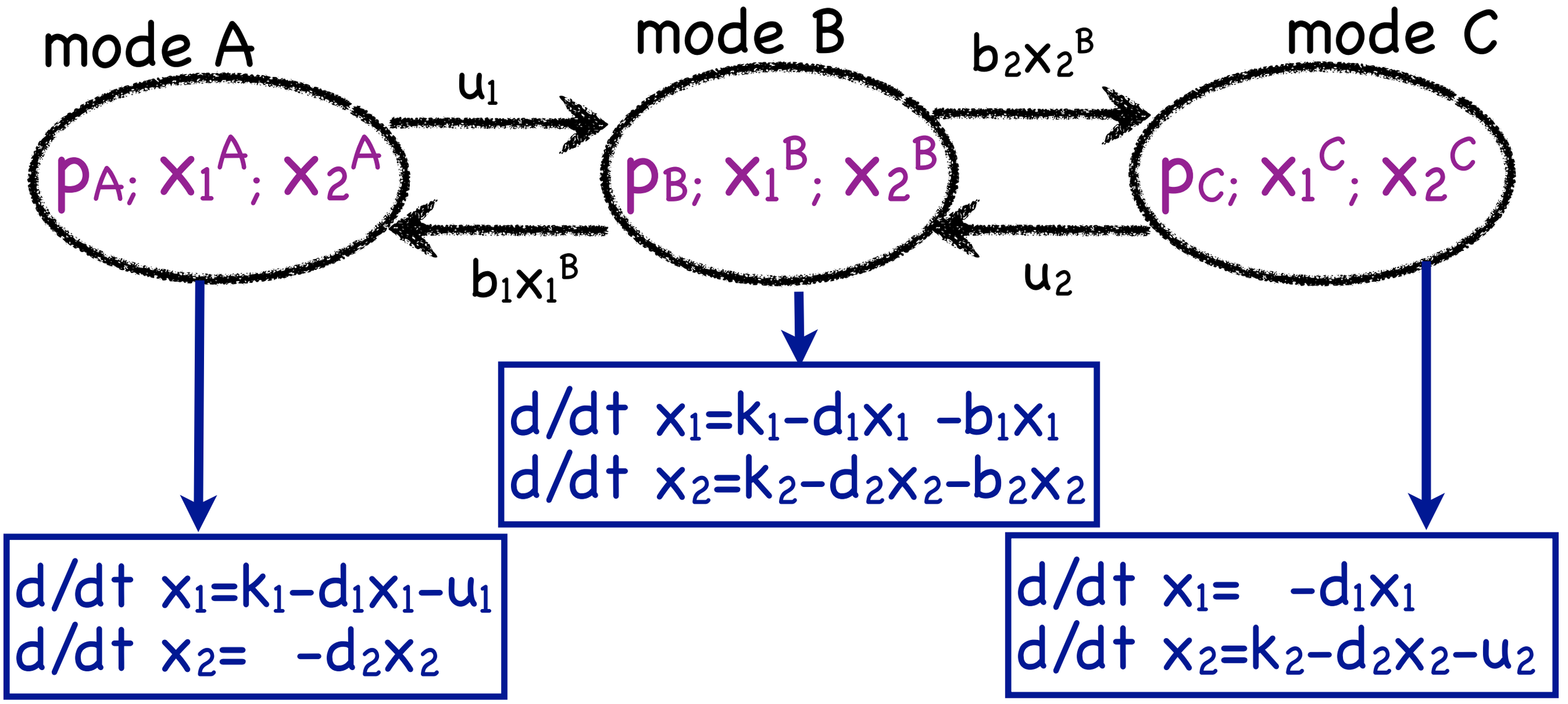
How to integrate over time?



1) integrate probability distribution for small [t,t+h]

$p_A(t) \dashrightarrow p_A(t+h)$ $p_B(t) \dashrightarrow p_B(t+h)$ ....

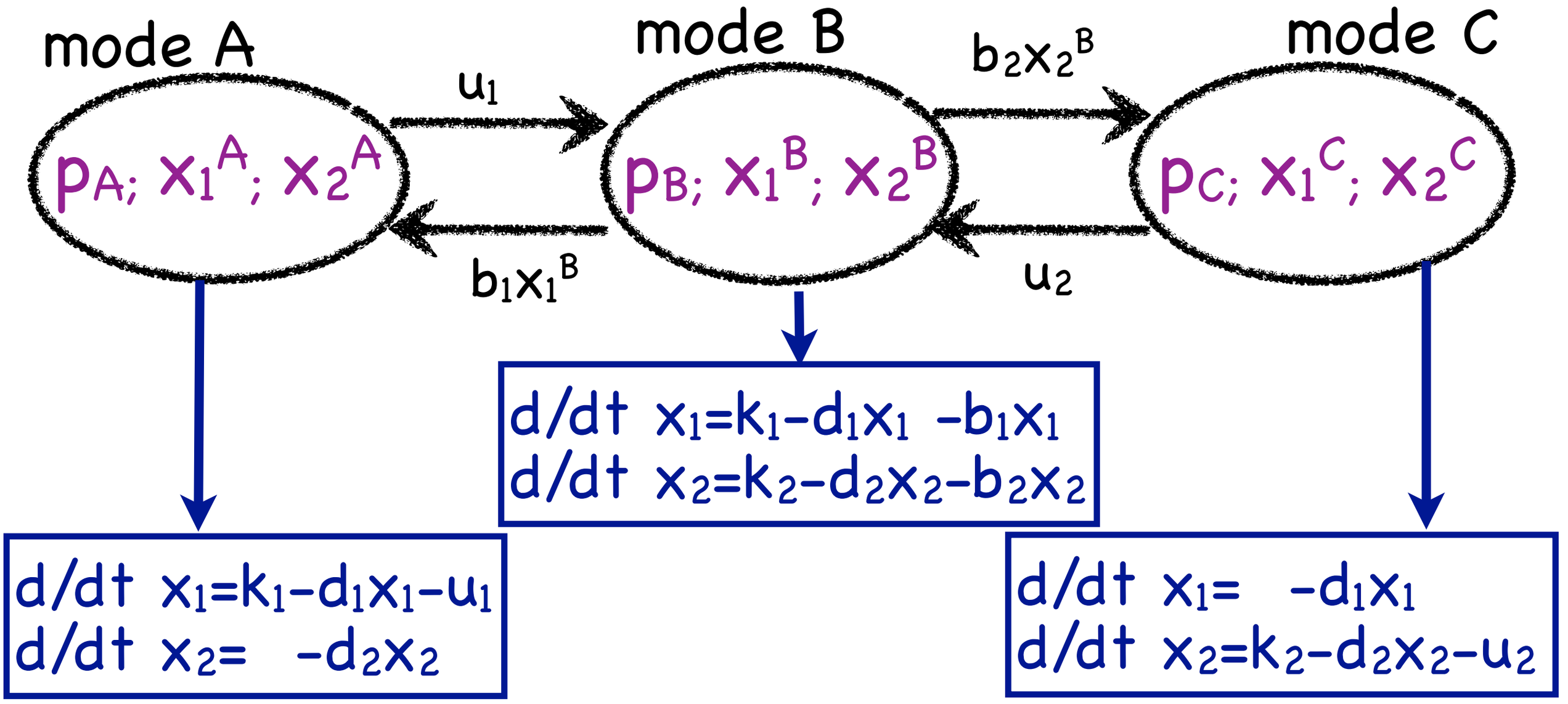
How to integrate over time?



2) integrate <u>conditional expect.</u> for small [t,t+h]

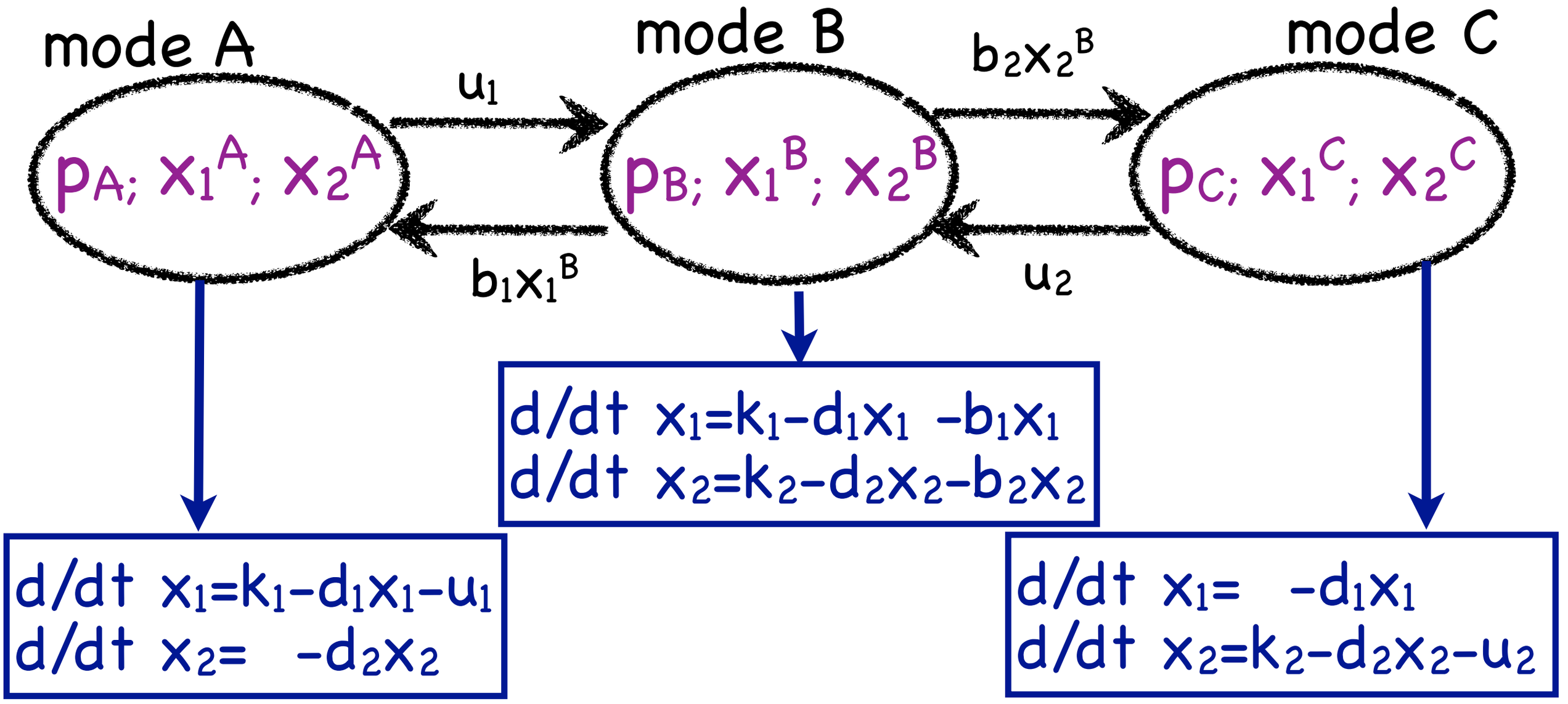
$x_i^A(t) \dashrightarrow x_i^A(t+h)$ $x_i^B(t) \dashrightarrow x_i^B(t+h)$ ....

How to integrate over time?



3) ``correct'' condition in $x_i^A(t+h), x_i^B(t+h), x_i^C(t+h)$ by taking into account that state is left during [t,t+h]

How to integrate over time?



Result at t+h: new probabilities $p_A(t+h), p_B(t+h), \ldots$

and new conditional expect. $x_i^A(t+h), x_i^B(t+h), \ldots$

How to integrate over time?

1) integrate mode probabilities for h time units

2) integrate conditional expectations of all modes for h time units

3) correct values obtained in 2) as follows:

$E[X_i(t+h) \mid \text{in mode A at time } t+h] \approx$

$\Sigma_{\text{mode B}}$ (inflow from B)*(value obtained in 2) for B) / (total inflow to A)

How to integrate

use numerical approachs developed for systems with small populations here

1) integrate mode probabilities for h time units

2) integrate conditional expectations of all modes for h time units

3) correct values obtained in 2) as follows:

$E[X_i(t+h) \mid \text{in mode A at time } t+h] \approx$

$\Sigma_{\text{mode B}}$ (inflow from B)*(value obtained in 2) for B) / (total inflow to A)

How to integrate over time?

1) integrate mode probabilities for h time units

2) integrate conditional expectations of all modes for h time units

3) correct values obtained in 2

value obtained under the assumption of remaining in mode during [t,t+h)

$E[X_i(t+h) \mid \text{in mode A at time } t+h] \approx$

$\Sigma_{\text{mode B}}$ (inflow from B)*(value obtained in 2) for B) / (total inflow to A)

How to integrate over time?

1) integrate mode probabilities for h time units

2) integrate conditional expectations of all modes for h time units

3) correct values obtained in 2) as follows:

$E[X_i(t+h) \mid \text{in mode A at time } t+h] \approx$

$\Sigma_{\text{mode B}}$ (inflow from B)*(value obtained in 2) for B) / (total inflow to A)

Probability flow from B to A during [t,t+h)

Experimental Results

Results for exclusive switch

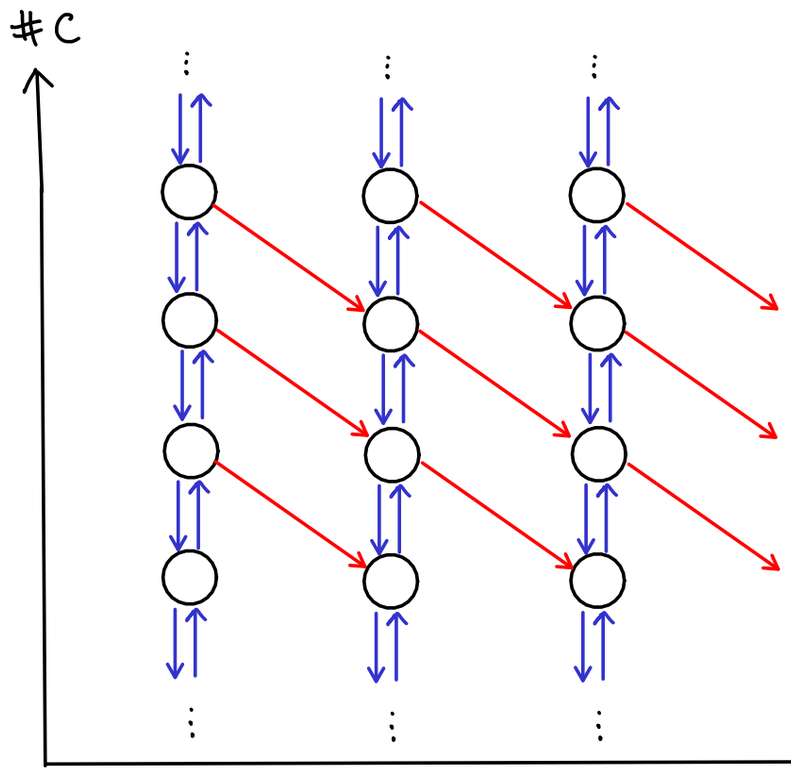
purley discrete ⟷ hybrid ⟷ ODE

pset	ex. time	$\|Sig\|$	error	pop. thres.	ex. time	$\|Sig\|$	m1	m2	m3	ex. time	m1
1	4h 51min	$2 \cdot 10^5$	$4 \cdot 10^{-5}$	50	25sec	$4 \cdot 10^2$	0.06	0.08	0.09	1sec	0.45
				100	28sec	$6 \cdot 10^2$	0.06	0.07	0.09		
2	2min 21sec	$7 \cdot 10^5$	$6 \cdot 10^{-5}$	50	18sec	$6 \cdot 10^3$	0.02	0.08	0.16	1sec	0.05
				100	1min 41sec	$4 \cdot 10^4$	0.01	0.05	0.12		

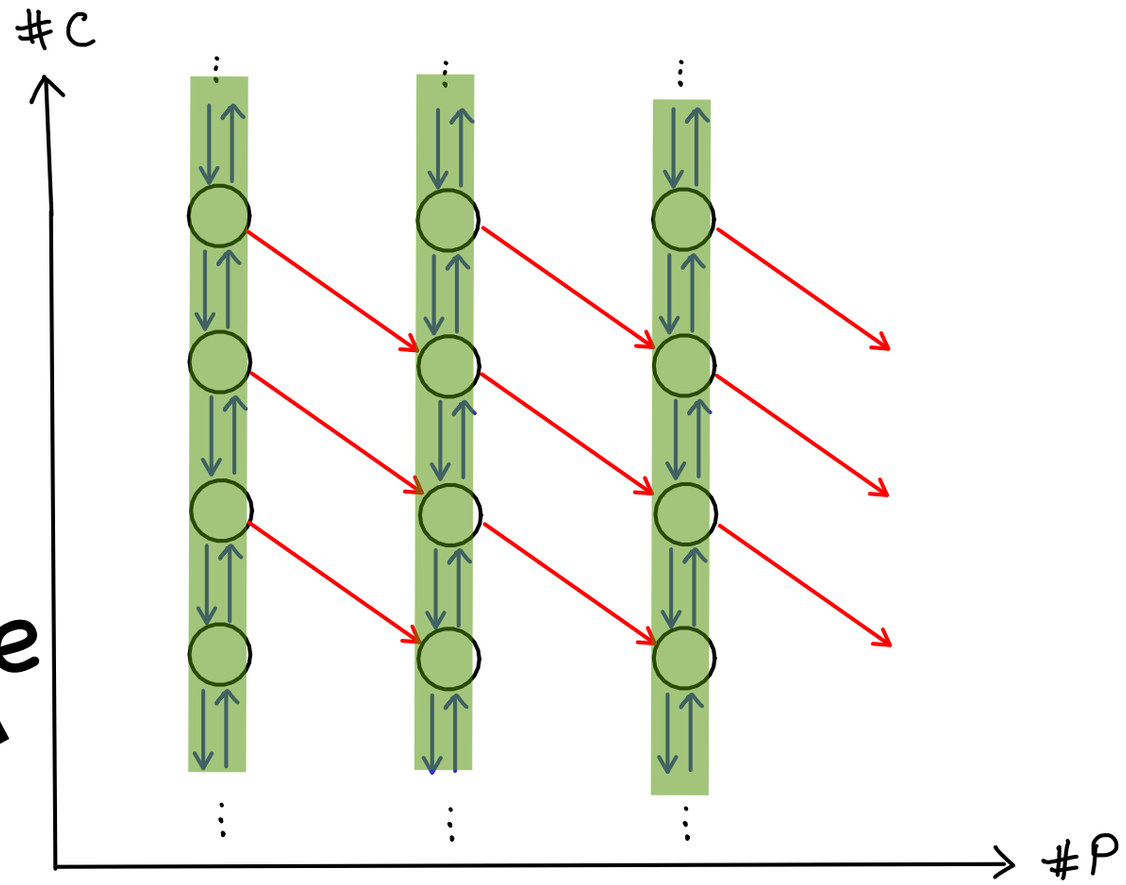
Use moment-based representation for proteins X_1 and X_2 when population reaches 50 or 100.

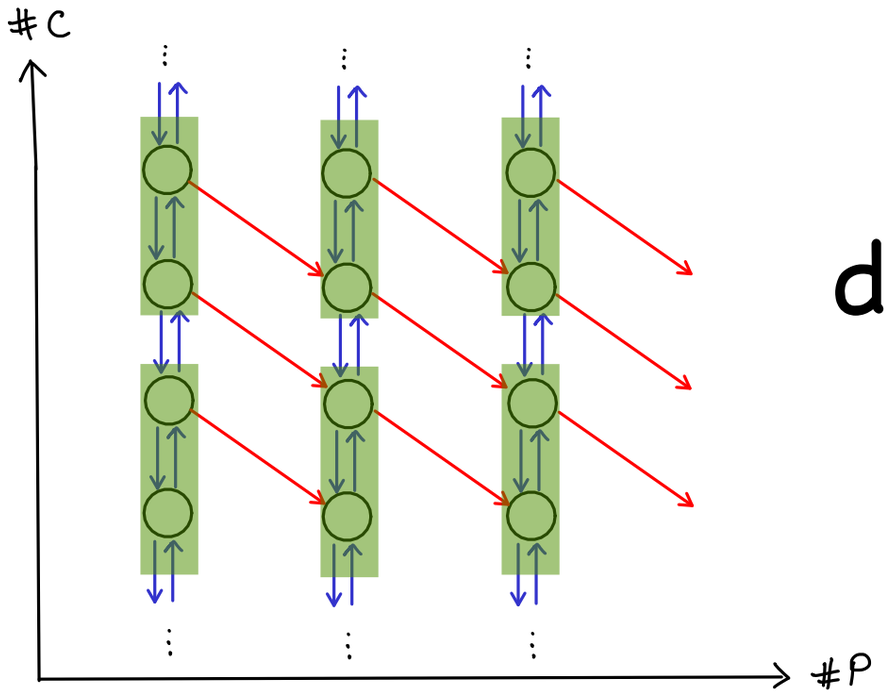
-> SHAVE DEMO

Solving the PDE by discretization



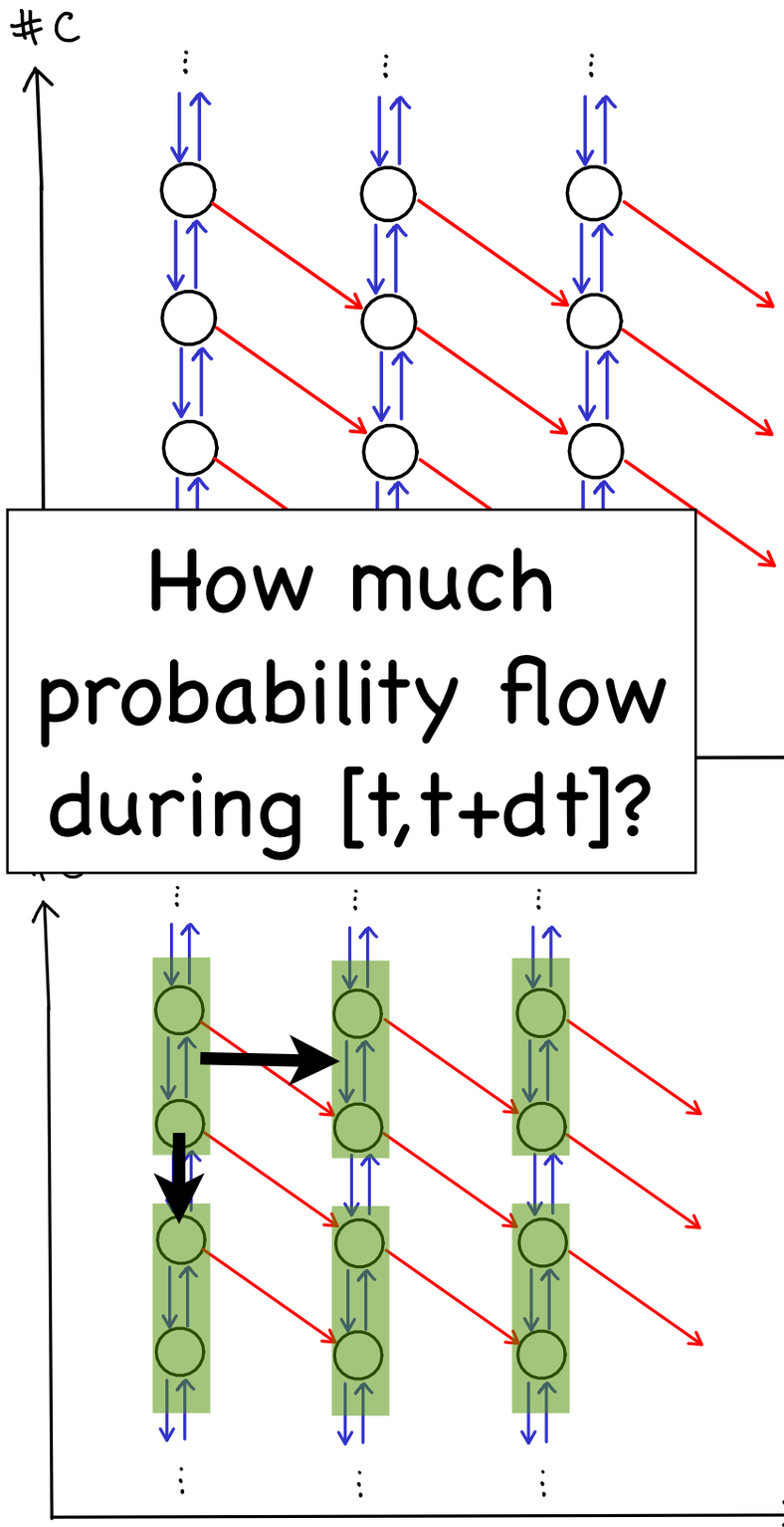
fluidize



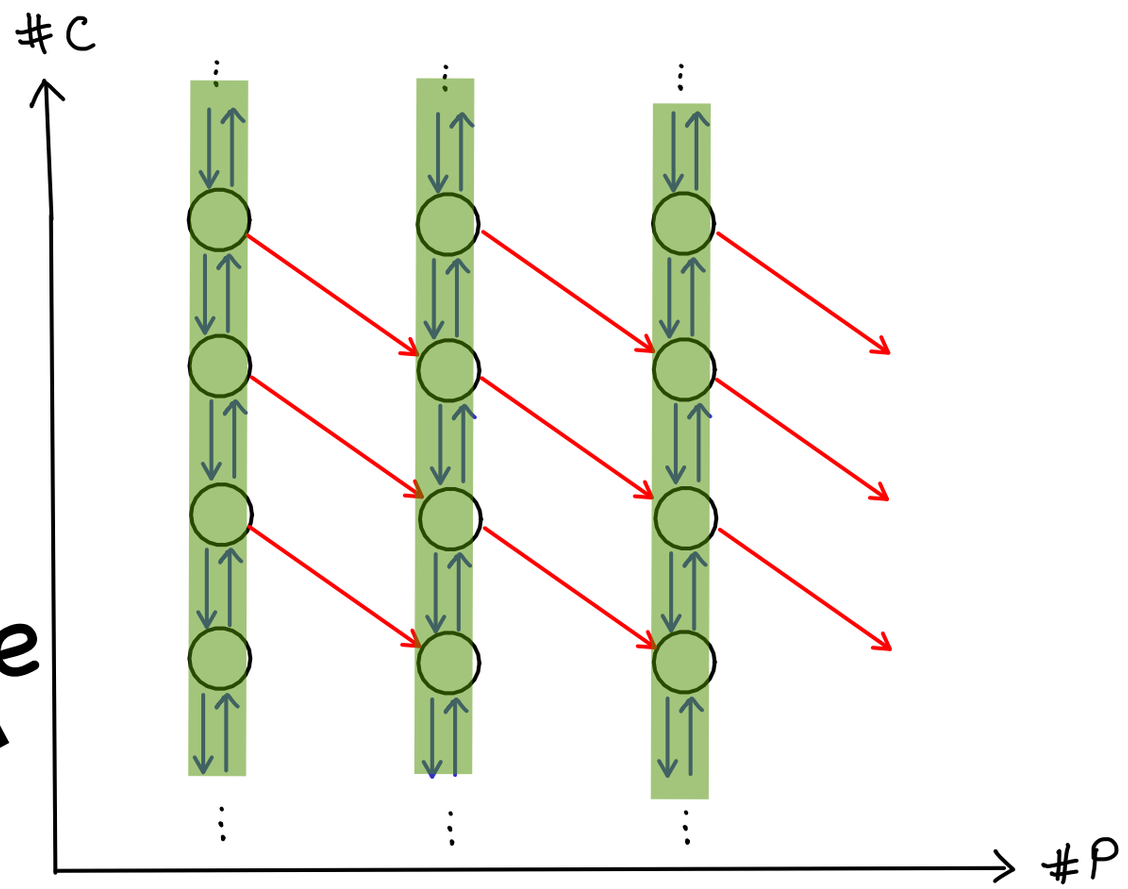


discretize

Solving the PDE by discretization



fluidize



discretize

Aggregation vs. Flow Approximation

assume that cells are (macro) states of a new (reduced) Markov chain

assume exponential distribution for jumps between macro states

true distribution is phase type => in general variance increases if number of phases is reduced to one

=> works only well in certain cases

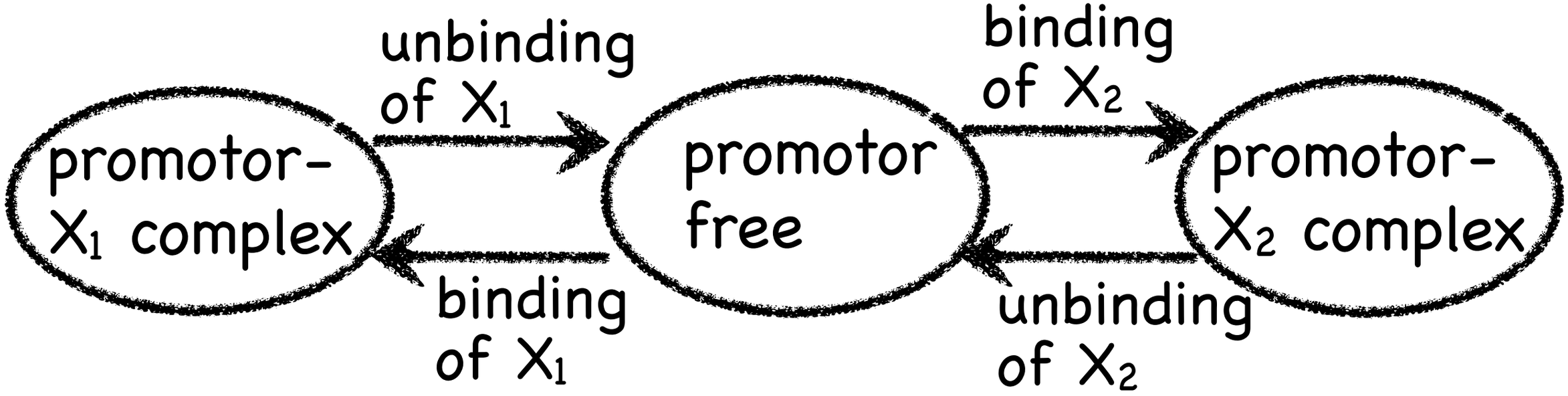
safe way:

approximate probability flow between cells and numerically integrate PDE

see e.g. "Fokker-Planck approximation of the master equation in molecular biology" by Sjöberg, Lötstedt, Elf

Steady-state solutions and stability analysis

Example: Exclusive Switch



Equilibrium points of mode ODEs:

$$\begin{aligned} x_1^A &= \frac{k_1+u_1}{d_1} \\ x_2^A &= 0 \end{aligned}$$

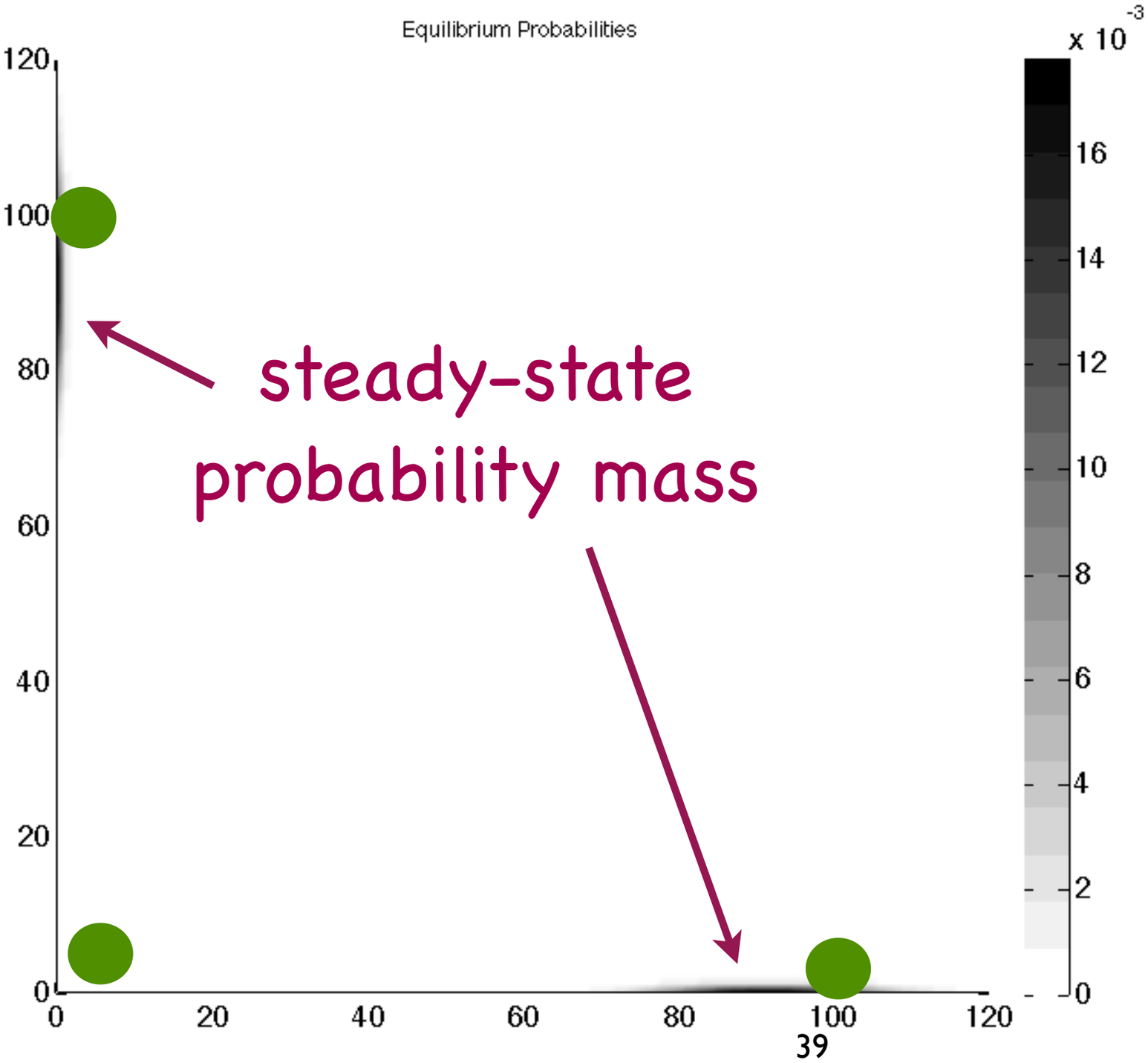
$$\begin{aligned} x_1^B &= \frac{k_1}{d_1+b_1} \\ x_2^B &= \frac{k_2}{d_2+b_2} \end{aligned}$$

$$\begin{aligned} x_1^C &= 0 \\ x_2^C &= \frac{k_2+u_2}{d_2} \end{aligned}$$

Does this help for locating equilibrium probabilities of the Markov chain?

High Binding Rate

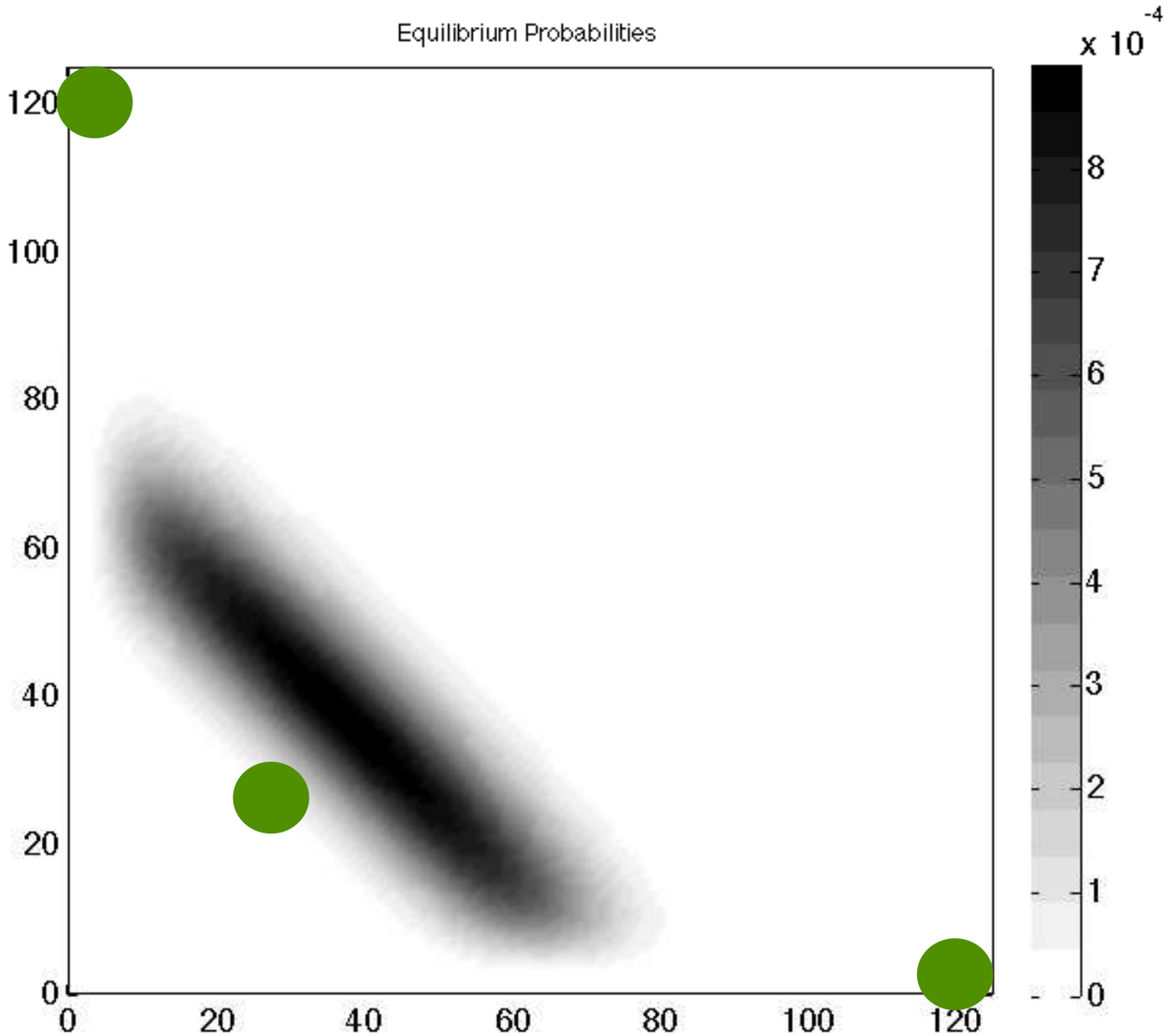
equilibrium of mode A and C at (100,0) and (0,100)



equilibrium point of mode B: (5,5)

Low Binding Rate

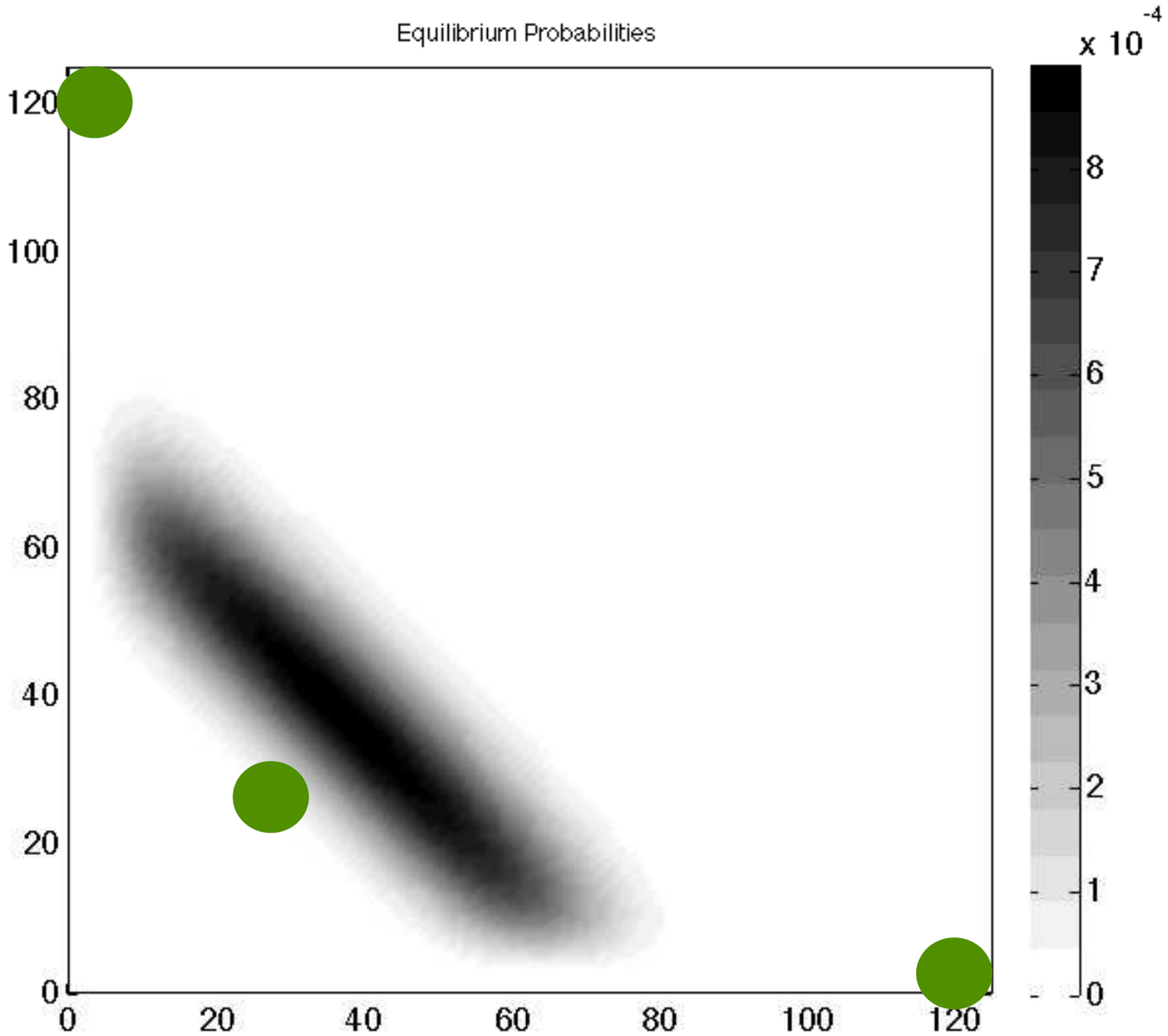
equilibrium of mode A and C at (120,0) and (0,120)



equilibrium point of mode B: (33,33)

Low Binding Rate

equilibrium of mode A and C at (120,0) and (0,120)

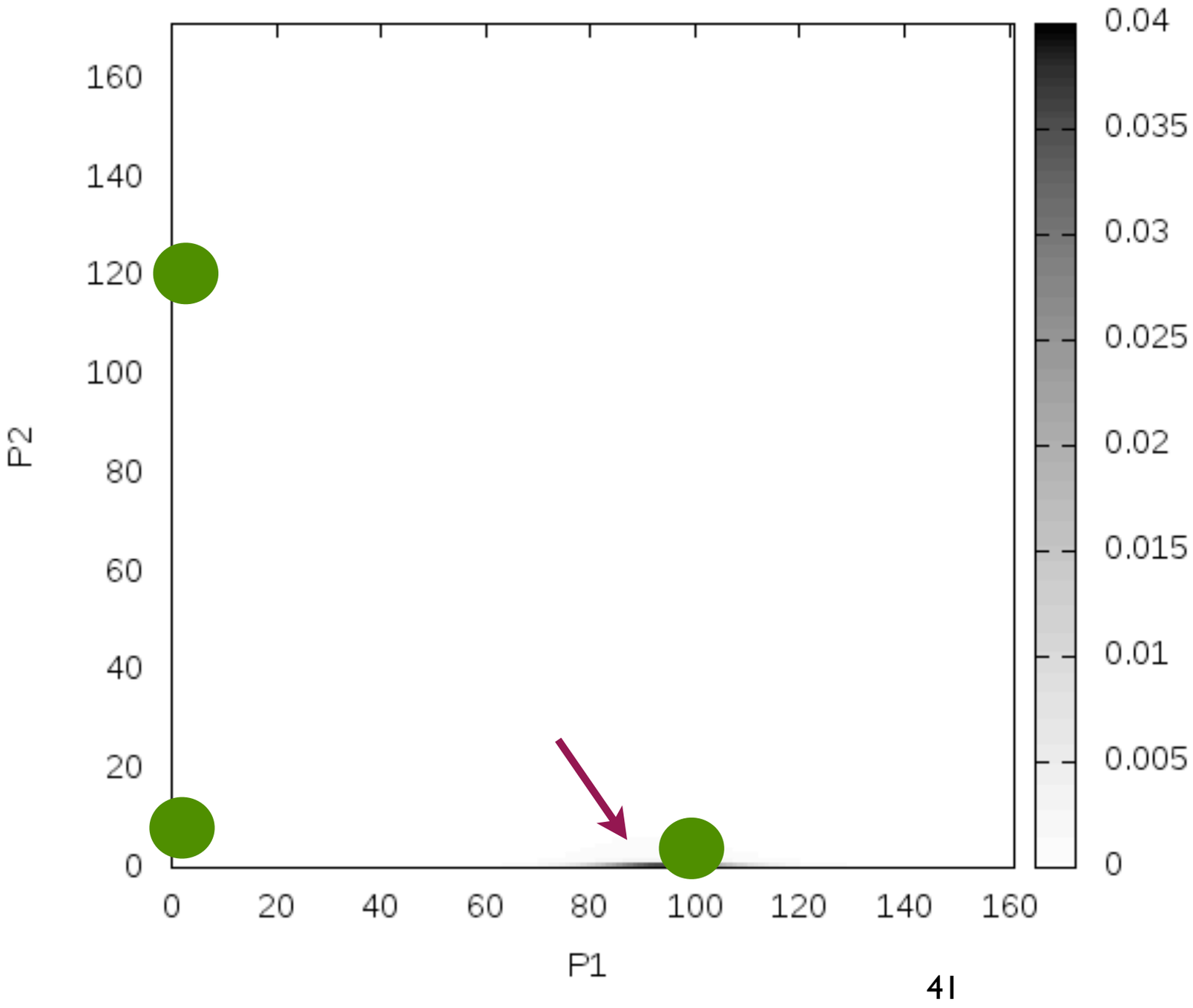


equilibrium point of mode B: (33,33)

Jumps between modes are not adequately taken into account!

Asymmetric Binding Rate

equilibrium of mode A and C at (120,0) and (0,100)



equilibrium of mode B: (5,0.5)

Jumps between modes are not adequately taken into account!

Stability Analysis

In order to decide whether a system is multistable and where the attractors are located:

in general equilibrium points of modes are not enough information

one has to compute/approximate the steady-state probability density

Steady-state probability density

$$\cancel{\frac{\partial}{\partial t} p(x,t)} + \frac{\partial}{\partial x} p(x,t) R(x) \quad = \quad p(t,x) Q(x)$$

Problem: no initial conditions are known

=> find values of x where density is zero!

=> solve PDE w.r.t. these side conditions (derivation of side condtions is still and open problem)

=> alternatively, run the system transiently until convergence of distribution

Conclusions

- for many systems, a hybrid approach is the right way to go (switch variables!)
- fluidization of large populations gives huge computational benefits (both for Monte-Carlo and numerical simulations)
- Efficient approaches for stability analysis are still missing
- Efficient approaches for parameter estimation are still missing